UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
CIÊNCIAS EXATAS E TECNOLÓGICAS,
PROGRAMA INTERDISCIPLINAR DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA - PIPCA
NÍVEL MESTRADO

ANDERSON DA CRUZ

Um Sistema Multiagente para Geração Automática de uma Rede Bayesiana para Análise de Riscos de Tecnologia de Informação

SÃO LEOPOLDO
2011

ANDERSON DA CRUZ

Um Sistema Multiagente para Geração Automática de uma Rede Bayesiana para Análise de Riscos de Tecnologia de Informação

Dissertação submetida à avaliação como requisito parcial para a obtenção do grau de Mestre em Computação Aplicada

Orientador: Prof. Dra. Patrícia Jaques

SÃO LEOPOLDO
2011

C957s Cruz, Anderson da

Um sistema multiagente para geração automática de uma Rede Bayesiana para análise de riscos de tecnologia de informação / por Anderson da Cruz. – São Leopoldo, 2011.

107 f. : il. ; 30 cm.

Dissertação (mestrado) – Universidade do Vale do Rio dos Sinos, Programa Interdisciplinar de Pós-Graduação em Computação Aplicada, São Leopoldo, RS, 2011.

Orientação: Profª. Drª. Patrícia Jaques, Ciências Exatas e Tecnológicas.

1. Proteção de dados. 2. Administração de risco. 3.Sistemas multiagentes. 4.Rede Bayesiana. 5.Tecnologia da informação. I.Jaques, Patrícia. II.Título.

CDU 004.056.53
004.89
004.72

Catalogação na publicação:
Bibliotecária Carla Maria Goulart de Moraes – CRB 10/1252

Dedico este trabalho
aos meus pais e a
minha esposa.

AGRADECIMENTOS

Agradeço em primeiro lugar aos meus pais Normélio Francisco da Cruz e Maria Rejane da Cruz e minha irmã Janaina da Cruz, que sempre me apoiaram e incentivaram minha evolução pessoal, profissional e acadêmica.

A minha esposa Camila Kunrath da Costa que além de apoiar, soube compreender as limitações impostas por este desafio.

A minha orientadora, Prof, Dra Patrícia Jaques, por ter possibilitado e apoiado o desenvolvimento deste trabalho, pelos conselhos dados que contribuíram na minha formação como pesquisador e professor e também pelos conselhos relacionados a carreira acadêmica.

Ao Prof. Dr João Gluz, por contribuir para o trabalho com sua experiência na utilização de Redes Bayesianas e Sistemas Multiagentes.

Aos professores do Programa Interdisciplinar de Pós-Graduação em Computação Aplicada da UNISINOS, que tive o privilégio de conhecer e em alguns casos, tê-los como professores durante mestrado.

Aos colegas de mestrado, que sempre mantiveram uma relação de amizade e ajudando um ao outro nos momentos de tensão e pressão no nivelamento e também durante as disciplinas.

"Não conseguimos segurar uma tocha para
iluminar o caminho de outra pessoa,
sem clarearmos o nosso próprio."
— **Ben Sweetland**

RESUMO

A cada ano, a informação ganha mais importância no meio corporativo e a utilização de sistemas de tecnologia de informação é cada vez mais comum em empresas dos mais diversos segmentos. No entanto, estes sistemas são compostos por aplicações que são sujeitas a vulnerabilidades que podem comprometer a confidencialidade, integridade e disponibilidade, ou seja, a segurança destas informações. Fornecedores de tecnologia estão sempre corrigindo falhas em suas ferramentas e disponibilizando correção para seu produtos para que estes se tornem mais seguros. O processo de correção de uma falha leva um determinado tempo até que o cliente possa atualizar o seu sistema. Muitas vezes este tempo não é suficiente para evitar um incidente de segurança, o que torna necessário soluções de contorno para diminuir riscos referentes aos aplicativos vulneráveis. O processo de análise/avaliação na Gestão de Riscos prioriza as ações que são tomadas para mitigar estes riscos. Este processo é árduo, envolvendo a identificação de vulnerabilidades para as aplicações utilizadas na empresa, sendo que o número de vulnerabilidades aumenta diariamente. Para apoiar na análise de riscos de tecnologia da informação, este trabalho propõe um método para geração automática de uma Rede Bayesiana baseado em sistemas multiagentes. O sistema multiagente conta com quatro agentes, sendo um destinado a monitorar as vulnerabilidades na *National Vulnerability Database*, outro para monitorar os ativos que compõem o negócio da organização, outro para monitorar os incidentes ocorridos no ambiente da organização e outro, destinado a reunir todas estas informações com o intuindo de determinar um fator de risco para os ativos da organização. Este último agente utiliza Redes Bayesianas para determinar o risco dos ativos. O método proposto mostrou-se eficiente para identificar mudanças no ambiente da organização e alterar o risco dos ativos de acordo com os diversos fatores que influenciam no seu cálculo, como o surgimento e/ou alteração de uma vulnerabilidade, surgimento e/ou alteração na base de dados de configuração de ativos da organização e identificação e/ou alteração no relato de incidentes de segurança no ambiente da empresa. Este resultado deve-se a utilização de Redes Bayesianas para calcular o risco dos ativos, visto que esta é capaz de considerar a relação causal entre os ativos da organização.

Palavras-chave: Gestão de Risco, Redes Bayesianas, Sistemas Multiagentes.

TITLE: "A Multi-agent System for Automatic Generation of a Bayesian Network Based Risk Analysis of Information Technology"

ABSTRACT

Every year information gains more significance in the corporative scenario and information technology system use is incresingly more commom on different segment companies. However, these systems are composed by applications that are under vulnerabilities that can compromise the confidentiality, integrity and availability, i.e. infomation security. Technology providers are always correcting flaws in their tools and providing it for their products in order for them to be safer. The flaw correction process considers some time until the client update his system. Many times this window is not enough to avoid a security incident, which turns necessary workarounds for minimizing risks concerning these vulnerable applications. The risks evaluation/analysis process aims primarily actions to mitigate these risks. This process is ardous, involving the identification of vulnerabilities in the used applications of a company, with this number growing each day. For supporting the information technology risks evaluation/analysis, a method for automated generation of a Bayesian Network multiagent system was proposed. This system is composed by four agents, one being destinated to monitoring vulnerabilities in National Vulnerability Database, another one for monitoring actives that compose the organition business, another one is responsible for to monitor the incidents occured in the organization enviroment and another one to gather all these information with the objective of determining a risk factor for the organization actives. The last one uses a Bayesian Network in order to determine the risk factor for the organization actives. The proposed method has shown to be effective in the identification of enviroment changes and in the changing of active risks according with several factors that influence its calculation, such as the emergence or changing of vulnerabilities, emergence or alteration on the business organization actives configuration database or identification and alteration of security incidents report on the company enviroment.

Keywords: Risk Management, Bayesian Network, Multi-agent System.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE SIGLAS

STI - Sistema de Tecnologia da Informação

TI - Tecnologia da Informação

GTI - Governança de Segurança da Informação

ISG - Information Security Governance

ITG - Information Technology Governance

SGSI - Sistema de Gestão de Segurança da Informação

NVD - National Vulnerability Database

SI - Segurança da Informação

CISR - Center for Information System Research

CVE - Common Vulnerabilities and Exposures

CPE - Common Platform Enumeration

CCE - Common Configuration Enumeration

URI - Uniform Resource Identifiers

IETF - Internet Engineering Task Force

RFC - Request for Comments

SMA - Sistema Multi Agente

IDA - Inteligência Artificial Distribuída

IA - Inteligência Artificial

UML - Unified Modeling Language

TPC - Tabela de Probabilidade Condicional

SUMÁRIO

1 INTRODUÇÃO

A introdução apresenta os fatores que serviram como motivação para o desenvolvimento deste trabalho, assim como o problema que é abordado e o objetivo que se deseja alcançar com a realização do trabalho. A Seção 1.1 apresenta a atual realidade das organizações e explica brevemente o motivo pelo qual a gestão de riscos é assunto de grande importância para estas organizações. O problema existente na análise/avaliação de risco, que é uma fase importante da gestão de risco, é abordado na Seção 1.2. Finalizando o Capítulo, é exposto na Seção 1.3, o objetivo que o trabalho busca atingir para contribuir com o problema apresentado na Seção 1.2.

1.1 MOTIVAÇÃO

A informação torna-se cada vez mais vital para uma organização, sendo capturada, armazenada, processada e transmitida através de Sistemas de Tecnologia da Informação (STI). Estes sistemas estão expostos a um vasto número de ameaças, que oferecem um grande risco para as informações, ameaçando a confidencialidade, integridade e disponibilidade (CID)[1] de tais informações (SOLMS e SOLMS, 2009).

Atualmente, o grande desafio para proteger a informação é utilizar uma disciplina que garanta a segurança da informação em meios eletrônicos e que as protejam contra riscos que possam surgir. A quantidade de regulamentações legais as quais as organizações são submetidas torna esse desafio ainda maior (SOLMS e SOLMS, 2009).

Segurança da Informação é uma disciplina que visa garantir a proteção desejada para a informação. A Governança em Segurança da Informação é um ambiente completo destinado a garantir esta proteção e sendo considerado um componente essencial para o sucesso da gestão da organização (SOLMS e SOLMS, 2009). A fragilidade da segurança da informação exige que sejam tomadas medidas imediatas para assegurar que os dados não sejam comprometidos e que os sistemas de informação permaneçam seguros (ISG, 2004).

A Gestão de Risco é apontada pelos guias de melhores práticas em governança como sendo uma das maiores responsabilidades da Governança Corporativa, visto que existem muitos tipos de riscos a serem geridos nas organizações, tais como: riscos financeiros, humanos etc (SOLMS e SOLMS, 2009). Guias de melhores práticas em Segurança da Informação, como a norma ABNT ISO/IEC 27001:2006, mais comumente conhecida como

[1] CID - Confidencialidade, integridade e disponibilidade são propriedades de segurança da informação e serão abordadas na Seção 2.1.1.

ISO 27001, também salientam a importância e a necessidade da Gestão de Risco (ABNT, 2006).

Um dos riscos ou tipos de riscos administráveis mais importantes são aqueles relacionados com a Tecnologia da Informação (TI) baseada em infraestrutura, como redes, banco de dados, etc. Essas infraestruturas geralmente manipulam todos os ativos[2] eletrônicos da organização e se essa viesse a perder esta infraestrutura, isso poderia gerar a paralisação do negócio da organização (SOLMS e SOLMS, 2009).

Pode-se perceber que a idéia de risco e Gestão de Risco é o cerne da Governança Corporativa e também que o uso de qualquer sistema baseado em TI gera sérios riscos para uma organização. Sendo assim, a competência de gerir riscos gerados pelo uso de TI também é o cerne da Governança em Tecnologia da Informação (GTI) (SOLMS e SOLMS, 2009).

O motivo pelo qual a gestão de riscos de TI é um dos componentes mais importantes da Governança em Tecnologia da Informação é o fato de que os sistemas de tecnologia de informação se tornam ativos eletrônicos da organização, incluindo nisto:

- todos os dados e informações armazenadas eletronicamente em arquivos e banco de dados;
- todos os dados e informações transmitidas pela rede; e
- todos os sistemas e aplicações necessárias para armazenar, transmitir e processar dados e informações.

Ativos eletrônicos possuem muitas ameaças, que podem afetá-los de diversos modos, porém, os mais relevantes são os relacionadas ao comprometimento das propriedades da segurança desses recursos. Isso faz com que se torne importante garantir estas propriedades perante um vasto número de ameaças que tentam comprometê-los. Estas ameaças podem ser (SOLMS e SOLMS, 2009):

- ataques externos como ataques maliciosos da internet através de vírus, *malware* etc;
- ataques internos de funcionários descontentes;
- ataques internos de erros gerados por funcionários; e
- ataques físicos como roubo, incêndio etc.

[2] A definição de ativo é apresentada na Seção 2.1.

A concretização das ameaças, os ataques, podem gerar sérios riscos aos ativos de uma organização, podendo levar até mesmo a paralisação de seu processo de negócio. No entanto, o benefício da segurança da informação não é apenas a redução de riscos ou a redução do impacto sobre a organização, se algo prejudicial vier a ocorrer. Outros retornos que a segurança da informação gera à organização são reputação e confiança do cliente e das pessoas que conduzem os negócios, podendo até melhorar sua eficiência, evitando perda de tempo e esforços relacionados a recuperação de um incidente de segurança (ITGI, 2006).

1.2 PROBLEMA

A informação é um ativo tão importante em uma organização quanto qualquer outro, sendo essencial para os negócios de uma organização, necessitando então, de uma proteção adequada. Isso se deve principalmente ao fato que essa informação se encontra em um ambiente de negócio cada vez mais interconectado, que a expõe a um crescente número e a uma grande variedade de ameaças e vulnerabilidades (ABNT, 2005).

Muitos sistemas de informação não foram projetados para serem seguros e a capacidade de se obter segurança através de meios técnicos é limitada. Devido a esta limitação, os sistemas de informação devem ser apoiados por uma gestão e por procedimentos aprimorados (ABNT, 2005).

Para se alcançar tal proteção, normas e/ou manuais de melhores práticas em Segurança da Informação são utilizados, como por exemplo, a ABNT NBR ISO/IEC 27002:2005, que apresenta os controles necessários para um Sistema de Gestão de Segurança da Informação (SGSI), que é especificado pela norma ABNT NBR ISO/IEC 27001:2006 (SOLMS e SOLMS, 2009).

A ABNT NBR ISO/IEC 27002:2005 sugere que uma avaliação de risco seja realizada no início do processo de Gestão de Segurança da Informação, ajudando então a direcionar e a determinar as ações apropriadas e prioritárias para o gerenciamento de riscos de segurança da informação e para a implementação dos controles selecionados para a proteção contra estes riscos (ABNT, 2005). A análise/avaliação de riscos quantifica ou descreve o risco qualitativamente, possibilitando então aos gestores priorizar os riscos de acordo com sua gravidade ou critérios estabelecidos (ABNT, 2008).

Uma boa análise/avaliação de risco é vital para a etapa de tratamento de risco. É possível que o tratamento de risco não resulte em um nível aceitável de risco residual, tornando necessária uma nova iteração na análise/avaliação de risco (ABNT, 2008).

A etapa de análise/avaliação de riscos é composta por duas atividades: (i) a análise

de riscos, que por sua vez possui duas tarefas e (ii) a avaliação de riscos. A análise de risco tem com propósito a identificação de riscos, bem como a estimativa destes mesmos riscos. Esse propósito é obtido através das suas duas tarefas, que são a identificação de risco e a estimativa de risco (ABNT, 2008).

Existem diversos modos de se identificar riscos, no entanto, todas partem da idéia básica de identificar os ativos, ameaças e preferencialmente vulnerabilidades (SOLMS e SOLMS, 2009).

Analisando relatórios gerados pelo *National Vulnerability Database (NVD)*[3] sobre as vulnerabilidades reportadas, é possível perceber o elevado número de vulnerabilidades reportadas todos os anos, como mostra a Figura 1.1. O mesmo ocorre com o número de incidentes reportados ao CERT.br[4], que mostra um crescimento em incidentes de segurança a cada ano, como mostra a Figura 1.2.

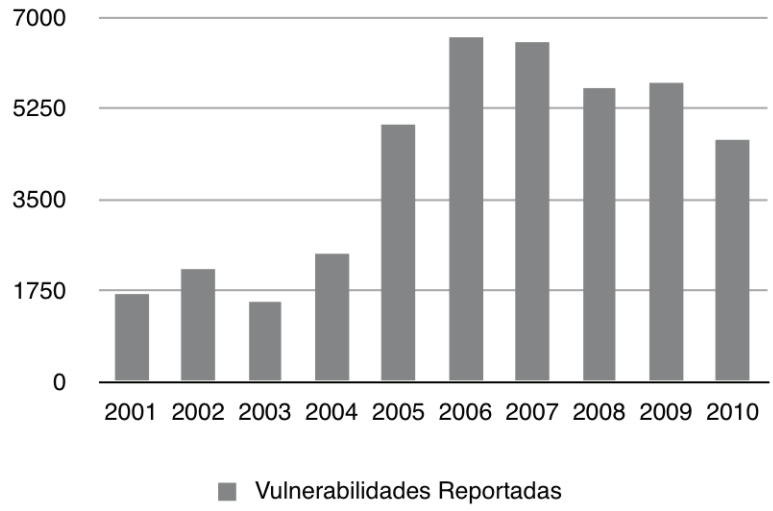


Figura 1.1: Número de vulnerabilidades por ano, segundo o NVD

A Figura 1.1 mostra um aumento significante de vulnerabilidades em ativos eletrônicos reportadas ao NVD entre os anos de 2004 e 2006. Após o ano de 2006, o número de vulnerabilidades encontradas nestes ativos manteram-se entre de 5.632 e 6.608 ao ano. Isso significa que, aproximadamente, dezessete novas vulnerabilidades aparecem a cada dia, aumentando o risco de organizações que utilizam estes ativos.

A Figura 1.2 ilustra o crescimento no número de incidentes reportados ao CERT.br a partir do ano de 2006, chegando a aumentar cerca de 473% no intervalo entre 2006 e 2009. O número de incidentes em 2006 foi 75.722 ao ano e em 2009 de 358.343. Incidentes sempre geram algum prejuízo as organizações, então pode-se afirmar que o prejuízo relacionado

[3]http://nvd.nist.gov/
[4]http://cert.br

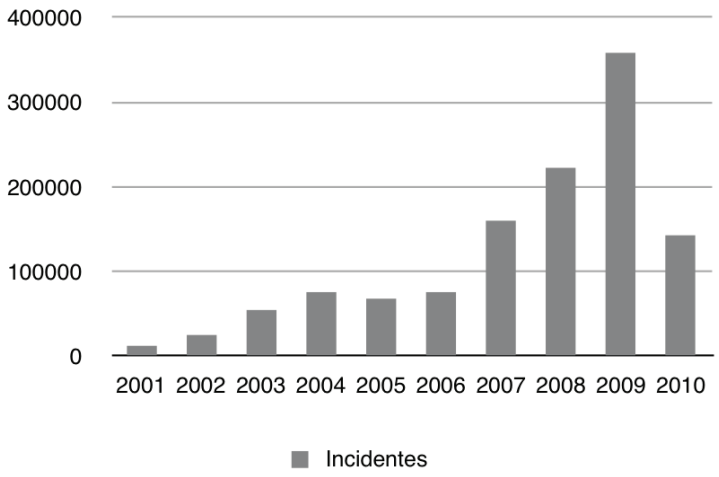


Figura 1.2: Número de incidentes por ano, segundo CERT.br

à segurança da informação também cresceu nos últimos dez anos.

Considerando que os sistemas de informação existentes em organizações estão expostos a diversos tipos de ameaças à segurança da informação, incluindo fraudes eletrônicas, espionagem, sabotagem e vandalismo (ABNT, 2005), o alto número de vulnerabilidades reportadas e o aumento de incidentes de segurança envolvendo estes sistemas, a análise/avaliação de riscos de sistemas de tecnologia da informação se torna cada vez mais complexa e também necessária.

Um dos principais motivos pelo qual a análise/avaliação de riscos é tão complexa é a relação existente entre os ativos da organização. O não funcionamento de ativo pode impactar diretamente no funcionamento de outros ativos. Sendo assim, é necessário considerar o risco de todos os ativos do qual um determinado ativo depende para se obter um valor de risco adequado.

1.3 OBJETIVOS

Este trabalho tem como objetivo implementar um sistema multiagente que auxilie na atividade de análise de risco. O sistema gerará automaticamente uma Rede Bayesiana, com base nos dados obtidos da organização e de uma base de dados de vulnerabilidades. Esta Rede Bayesiana será utilizada no sistema para estimar o risco relacionado à segurança da informação dentro da organização.

Uma abordagem multiagente permite reduzir a complexidade de um sistemas criando componentes modulares que executam sub-tarefas que juntas constituem um objetivo, que neste caso, é auxiliar na atividade de análise de riscos de tecnologia da informação.

O sistema conta com um agente responsável por monitorar a base de dados de vulnerabilidades NVD, identificando novas vulnerabilidades e também a alteração em vulnerabilidades já existentes. Outro agente é responsável por identificar novos ativos na base de dados de configuração de ativos (interna) e também alterações em sua configuração. Os incidentes de segurança da informação ocorridos na organização são monitorados por um terceiro agente, assim como alterações de seus dados.

Todas as informações coletadas por estes agentes são repassadas a um quarto agente que será responsável por criar dinamicamente uma Rede Bayesiana, que será utilizada para calcular o risco dos ativos existentes na organização. Com a utilização de Redes Bayesianas para determinar o riscos dos ativos, é possível considerar a relação entre os ativos, permitindo identificar alterações que podem ser ignoradas por outras formas de cálculo, que dependem da percepção humana para considerar as relações entre os ativos.

1.4 ORGANIZAÇÃO DO TEXTO

O Capítulo 2 apresenta a definição de Segurança da Informação, apresentando os principais termos e nomenclaturas. A disciplina de Segurança da Informação é relacionada com Governança em Segurança da Informação, deixando evidente a necessidade mútua da Gestão de Risco. As atividades necessárias para a Gestão de Riscos, tendo um maior foco na tarefa de identificação de riscos, sendo essa relacionada com o objetivo do trabalho. Bases de dados de vulnerabilidades, que fornecem informações importantes para especialistas em segurança, também são abordadas no Capítulo 2, juntamente com o cálculo de pontuação para vulnerabilidades. O Capítulo 3 introduz Redes Causais e o conceito de *d-separation*, que servem como base para as definições de Redes Bayesianas, também apresentadas neste capítulo. Os conceitos e definições de Agentes e Sistemas Multiagentes são apresentados no Capítulo 4, juntamente com a metodologia *Prometheus* para desenvolvimento de sistemas multiagente, que é utilizada neste trabalho. Os trabalhos relacionados que foram utilizados como idéia inicial para o presente trabalho são apresentados no Capítulo 5. O trabalho proposto é apresentado no Capítulo 6. Nele são apresentados os cenários, base de dados, agentes, percepções, ações e como será gerada a Rede Bayesiana no sistema proposto. Finalizando o trabalho, são apresentadas as considerações finais no Capítulo 7.

2 SEGURANÇA DA INFORMAÇÃO

Como visto no Capítulo 1, a Segurança da Informação é uma disciplina que tem como objetivo garantir a confidencialidade, integridade e a disponibilidade das informações armazenadas em meios eletrônicos, ou seja, garantir a segurança destas informações. Sendo esta disciplina integrante da Governança em Segurança da informação e também relacionada às atividades de Gestão de Risco de ativos de TI, este capítulo se destina a apresentar os principais conceitos de Segurança da Informação para uma melhor compreensão do trabalho proposto.

Este capítulo apresenta, na Seção 2.1, a definição de Segurança da Informação, apresentando os principais termos e nomenclaturas, a família de normas 27000 juntamente com o seu propósito e os controles de segurança que são beneficiados com este trabalho. Na Seção 2.2, a disciplina de Segurança da Informação é relacionada com Governança em Segurança da Informação, deixando evidente a necessidade mútua da Gestão de Risco. Na Seção 2.3, serão apresentadas todas as atividades necessárias para a Gestão de Riscos, tendo um maior foco na tarefa de identificação de riscos, sendo essa relacionada com o objetivo do trabalho. Finalizando o capítulo, a Seção 2.4 apresenta o tema das bases de dados de vulnerabilidades, que fornecem informações importantes para especialistas em segurança.

2.1 DEFINIÇÃO DE SEGURANÇA DA INFORMAÇÃO

Antes de apresentar a definição de Segurança da Informação, é necessário definir o que é um ativo. Um ativo é qualquer coisa que tenha valor para a organização (ABNT, 2006). Podemos dar como exemplos de ativos de uma organização: os funcionários, a imagem da organização perante seus clientes, seu patrimônio, etc. Os ativos podem ser classificados em subconjuntos, como por exemplo, ativos eletrônicos, que aparecem com maior freqüência neste trabalho. Por este fato, quando o termo ativo aparecer no texto, este está se referindo especificamente a um ativo eletrônico.

A forma mais objetiva para se definir Segurança da Informação é utilizando a definição da norma ABNT NBR ISO/IEC 27001:2006, que a apresenta como a preservação da confidencialidade, integridade e disponibilidade da informação e de outras propriedades, como autenticidade, responsabilidade, não repúdio e confiabilidade (ABNT, 2006).

Sêmola (2003) corrobora esta definição e ainda complementa como sendo uma área do conhecimento dedicada à proteção de ativos da informação contra acessos não autorizados, alterações indevidas ou sua indisponibilidade. O autor a conceitua como um fim alcançado

por meio de práticas e políticas voltadas a uma padronização operacional e gerencial dos ativos e processos que manipulam e executam a informação.

A norma ABNT NBR ISO/IEC 27002:2005 define Segurança da Informação como sendo 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).

A definição mais completa, e também complexa, é apresentada por Solms e Solms (2009), em que esta aparece como sendo uma disciplina multi-dimencional direcionada para a Governança em Segurança da Informação. Os autores defendem esta teoria afirmando que Segurança da Informação só pode ser implementada com sucesso e eficientemente em uma organização se todas as suas dimensões forem abordadas de uma forma holística e compreensível (SOLMS e SOLMS, 2009).

As dimensões propostas por Solms e Solms (2009) são:

- Dimensão da governança (corporativa);
- Dimensão organizacional;
- Dimensão de gestão;
- Dimensão de políticas (de SI);
- Dimensão de melhores práticas;
- Dimensão ética;
- Dimensão de certificação;
- Dimensão legal;
- Dimensão de garantia;
- Dimensão humana;
- Dimensão de consciência;
- Dimensão técnica;
- Dimensão métricas;
- Dimensão de auditoria; e
- Dimensão forense em TI.

Analisando estas definições, é possível perceber uma relação entre estas. A base para todas as definições sempre corrobora com a primeira definição apresentada neste trabalho, a da norma ABNT NBR ISO/IEC 27001:2006, que tem como base as três principais propriedades de segurança. Solms e Solms (2009) apenas consideram em sua definição o contexto em que se tenta garantir estas propriedades.

2.1.1 Propriedades de Segurança da Informação

Ao longo do texto, os termos integridade, confidencialidade e disponibilidades aparecem sempre relacionados entre si e vinculados a definições de Segurança da Informação. Isso ocorre porque estes são os pilares da Segurança da Informação.

As definições destas propriedades são simples e objetivas, no entanto, a forma com que estas são relacionadas para se obter a segurança da informação requer um pouco mais de atenção, pois esta depende da utilização dos ativos de informação em que as propriedades estão sendo aplicadas. Devido a simplicidade e objetividade das definições destas propriedades, este trabalho se baseia nas definições da norma ABNT NBR ISO/IEC 27001:2006 (ABNT, 2006).

Integridade é a propriedade que garante que a informação não será alterada ou destruída sem a autorização adequada. Para melhor entender esta propriedade, pode-se imaginar um documento eletrônico contendo um contrato qualquer. Este tem a sua integridade preservada enquanto seu conteúdo não seja alterado por alguém não autorizado (ABNT, 2006).

A **confidencialidade** garante que a informação não será revelada sem autorização adequada. Utilizando ainda o exemplo do documento eletrônico, imagine que este é extremamente sigiloso e não pode nem ao menos ser acessado por uma pessoa não autorizada. Então este tem sua confidencialidade preservada enquanto seu conteúdo não seja acessado por alguém não autorizado (ABNT, 2006).

Por fim, a **disponibilidade** garante que a informação estará acessível aos usuários legítimos quando solicitada. No mesmo exemplo, imagine que as duas propriedades anteriores estão preservadas e que no momento em que uma das partes autorizadas for abrir o documento para imprimir, o computador não liga. Neste caso sua disponibilidade está comprometida até que o computador funcione (ABNT, 2006).

Além destas três propriedades de segurança da informação, existem outras que são complementares a estas, que são autenticidade, não repúdio e confiabilidade.

Autenticidade garante que a informação foi realmente criada ou emitida por um

determinado autor/emissor e é obtida principalmente por infraestrutura de chaves públicas (criptografia). Em um caminho inverso, o **não repúdio** garante que o autor/emissor não possa negar a autenticidade da informação.

Confiabilidade é a propriedade que representa a capacidade de um ativo desempenhar satisfatoriamente a sua função em condições de operação estabelecidas e em um período de tempo predeterminado.

Com base nessas definições, pode-se afirmar que a segurança da informação é violada quando há a quebra de uma ou mais propriedades de segurança. Sendo a violação de segurança relacionada com as propriedades de segurança, pode-se classificá-las em três categorias:

Violação de integridade: modificação não autorizada da informação;

Violação de confidencialidade: revelação não autorizada da informação;

Violação de disponibilidade: negação de serviço.

2.2 SEGURANÇA DA INFORMAÇÃO E GOVERNANÇA EM SI

Termos como Segurança da Informação, Governança Corporativa, Governança de Tecnologia da Informação e Governança de Segurança da Informação aparecem neste trabalho justificando a importância da Gestão de Risco em uma organização. A relação existente entre estes termos é tênue, podendo levar a dúvidas. Esta Seção aborda a relação existente entre essas áreas/disciplinas, propiciando um melhor entendimento sobre o tema.

No meio corporativo, a governança recebe o nome de Governança Corporativa, que é definida como o conjunto de relações estabelecidas entre gestores de uma organização, seus conselhos de administração, acionistas e demais interessados, como: credores, governos, sociedade, fornecedores, funcionários, entre outros (BRUÈRE et al., 2007).

Governança Corporativa também pode ser definida como uma relação entre investidores que regula e controla a direção estratégica e o desempenho das organizações, estando voltada para a identificação de práticas que garantam que as decisões sejam tomadas de forma correta (VEIGA, 2006).

No Brasil, o IBGC - Instituto Brasileiro de Governança Corporativa - lançou em 1990 o primeiro Código de Práticas de Governança Corporativa, fundamentado em quatro princípios básicos: a transparência, a eqüidade[1], a prestação de contas e a responsabilidade corporativa (BRUÈRE et al., 2007).

[1]Segundo o dicionário Michaelis: Disposição para reconhecer imparcialmente o direito de cada acionista. Igualdade, justiça, retidão.

A Governança Corporativa está em processo evolutivo e acredita-se que este processo está apenas começando. Uma iniciativa que demonstra isso é o *Sarbanes-Oxley Act*, uma lei que adiciona mais responsabilidades aos administradores públicos sob a forma de penalidades criminais. A *Sarbanes-Oxley Act* teve impacto sobre companhias abertas que tem seus papéis negociados no mercado americano (BRUÈRE et al., 2007) .

Na Governança Corporativa, podem existir governanças específicas responsáveis por gerir áreas de uma organização, como a Governança de Tecnologia de Informação (TI), Governança de Segurança da Informação (SI) e Governança Ambiental. Dentre as citadas, a Governança de Tecnologia da Informação e a Tecnologia de Segurança da Informação estão mais próximas, sendo que uma influência na outra.

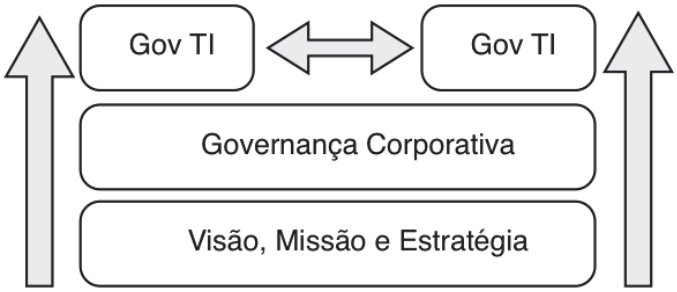


Figura 2.1: Governanças específicas no contexto da governança corporativa

A Figura 2.1 representa o alinhamento da Governança Corporativa com a visão, missão e estratégia da organização, a comunicação existente entre as Governanças de Tecnologia da Informação e Segurança da Informação e o alinhamento destas perante a Governança Corporativa.

Governança de Tecnologia da Informação é uma estrutura de relacionamento entre processos com o intuito de direcionar e controlar uma organização para atingir seus objetivos estratégicos através de agregação de valor e gestão de riscos na utilização da tecnologia da informação e comunicação, visando a fusão da tecnologia da informação e comunicação ao negócio. A Governança de TI determina o comportamento desejável no uso da tecnologia da informação e comunicação no meio corporativo (BERNARDES e MOREIRA, 2006).

Basicamente, a Governança de Tecnologia da Informação pode ser resumida na resposta às três perguntas (MANSUR, 2007):

1. Que decisões devem ser tomadas?
2. Quem deve tomá-las?
3. Como tomá-las e monitorá-las?

O *Center for Information System Research* (CISR), da MIT *Sloan School*, identificou as principais decisões e os arquétipos[2] da Governança de Tecnologia da Informação.

Dentre as tomadas de decisões, foram identificados os (i) princípios de TI, que são basicamente as decisões de alto nível sobre o alinhamento entre TI e o negócio; (ii) arquitetura de TI, que são basicamente as decisões sobre a organização lógica dos dados, aplicações e infraestrutura, definidas a partir de um conjunto de políticas, padronizações e integrações; (iii) infraestrutura de TI, que são basicamente as decisões sobre a capacidade atual e planejada de TI disponível para o negócio, sob a forma de serviço compartilhado; (iv) necessidades de aplicação de negócio, que são basicamente decisões sobre as necessidades do negócio que geram valor, onde deve-se encontrar o equilíbrio entre criatividade e disciplina; e (v) investimentos e priorização de TI, que são basicamente as decisões sobre quanto gastar, em que gastar e como equilibrar a diferentes necessidades (MANSUR, 2007).

As demandas de controle, transparência e previsibilidade deram origem à Governança de TI. Estas demandas se originaram das questões relativas à qualidade, que ganharam importância no cenário mundial no começo da década de noventa.

Na Governança da Tecnologia da Informação, existem alguns guias de melhores práticas de controles que são utilizados para atender o seu objetivo, como o COSO - *The Comitee of Sponsoring Organization*, COBIT - *Control Objectives for Information and Related Technology*, ITIL - *Information Technology Infra-structure Library*, ISO 20000 - Gerenciamento de Serviços, AS 8015:2005 - *Australian Standard for Corporate Governance*, CMM - *Capability Maturity Model*, CMMI - *Capability Maturity Model Integrated* e ValIT - *Value of Information Technology*.

Algumas destas melhores práticas de controle são utilizadas também na Governança da Segurança da Informação para atender ao seu objetivo e ao objetivo da Governança Corporativa.

A Governança de Segurança da Informação é um subconjunto de organizações da Governança Corporativa (ISG, 2004). Governança de Segurança da Informação pode ser definida como o sistema pelo qual a confidencialidade, integridade e disponibilidade dos ativos são mantidos pela organização (SOLMS, 2007).

Também pode-se dizer que Governança de Segurança da Informação é a ação que leva ao controle de uma organização e ao alinhamento com os objetivos estratégicos, garantindo a cultura da segurança da informação, otimizando os processos relacionados e determinando atividades para que pessoas competentes possam desempenhar (ALVES et al., 2006).

[2]Segundo o dicionário Michaelis: Modelo dos seres criados. O que serve de modelo ou exemplo, em estudos comparativos.

Governança de Segurança da Informação é considerada um componente essencial para o sucesso da gestão organizacional. A sua fragilidade exige que sejam tomadas medidas imediatas para assegurar que os dados não sejam comprometidos e que os sistemas de informação permaneçam seguros (ISG, 2004).

A Governança da Segurança da Informação consiste em liderança, estruturas organizacionais e processos que garantam a segurança da informação. Para o sucesso destas estruturas e processos é essencial haver a comunicação eficaz entre todas as partes com base em uma relação construtiva, linguagens comuns e compartilhamento de compromissos para abordar questões (ITGI, 2006).

2.3 GESTÃO DE RISCO

Esta seção apresenta uma visão do processo de Gestão de Riscos e detalhando a atividade de análise de risco. As definições apresentadas nesta seção são fortemente baseadas na norma ABNT NBR ISO/IEC 27005:2008, que aborda a Gestão de Riscos de Segurança da Informação, visto que é a normal sugerida como referência pela norma ABNT NBR ISO/IEC 27001:2006.

Todos os tipos de empreendimentos se deparam com situações (ou eventos) que constituem oportunidades de benefício ou ameaça ao seu sucesso. Oportunidades podem ser aproveitadas ou ameaças podem ser reduzidas por uma gestão ativa (ABNT, 2005b).

Gestão de riscos, em áreas como a do mercado financeiro, trata das flutuações monetárias como uma oportunidade de ganhos ou como uma potencial perda. Na Segurança da Informação em STI, a Gestão de Risco é focada na prevenção e mitigação dos danos em ativos de TI (ABNT, 2005b).

Segundo a norma ABNT NBR ISO/IEC 27005:2008, uma abordagem sistemática de gestão de risco de segurança da informação é necessária para se identificar as necessidades da organização em relação aos requisitos de segurança da informação e para criar um sistema de gestão de segurança da informação (SGSI) que seja eficaz. Convém que os esforços de segurança da informação lidem com riscos de maneira efetiva e no tempo apropriado, onde e quando forem necessários (ABNT, 2008).

É importante que a gestão de segurança da informação seja um processo contínuo que defina o contexto, avalie os riscos e trate os riscos usando um plano de tratamento a fim de implementar as recomendações e decisões. Convém que a gestão de risco analise os possíveis acontecimentos e suas consequências, antes de decidir o que será feito e quando será feito, a fim de reduzir os riscos a um nível aceitável (ABNT, 2008).

O processo de gestão de risco de segurança da informação consiste na definição do

contexto, análise/avaliação de risco, tratamento do risco, aceitação do risco, comunicação do risco e monitoramento e análise crítica do risco (ABNT, 2008). Essas etapas e suas relações podem ser melhor visualizadas no fluxograma apresentado na Figura 2.2.

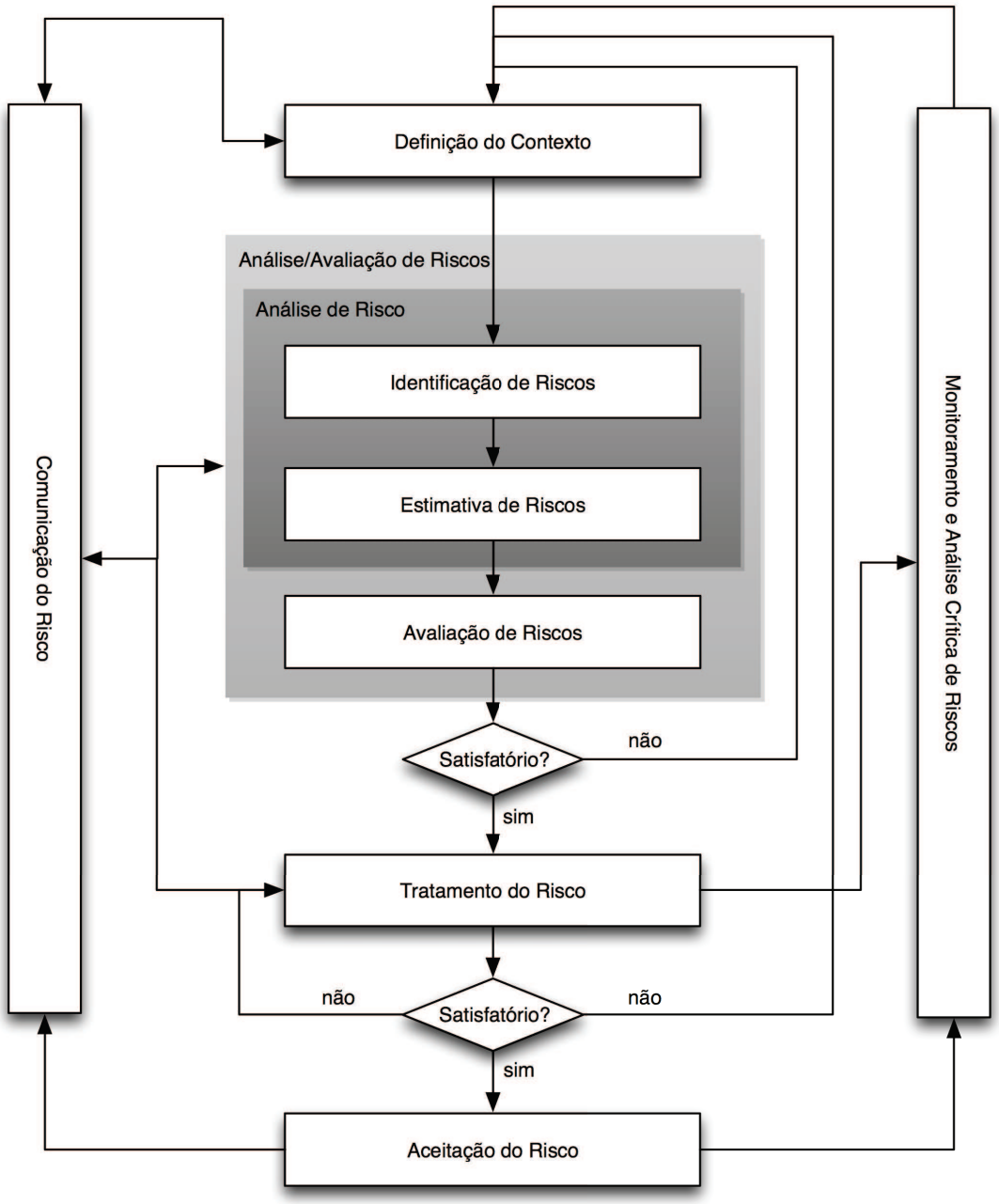


Figura 2.2: Fluxograma de gestão de risco em segurança da informação(ABNT, 2008).

Como pode-se visualizar na Figura 2.2, o processo de gestão de risco de segurança da informação pode ter as atividades de análise/avaliação de risco e/ou tratamento de risco, sendo realizadas mais de uma vez. Um enfoque iterativo na execução da análise/avaliação de risco torna possível aprofundar e detalhar a avaliação em cada repetição e também

permite minimizar o tempo e o esforço despendidos na identificação de controles e, ainda assim, assegurar que os riscos de alto impacto ou de alta probabilidade possam ser adequadamente avaliados (ABNT, 2008).

A primeira etapa do processo de gestão de risco é a definição o contexto, em seguida, executa-se a análise/avaliação de risco, que pode ser refeita até se obter informações suficientes para que se determine de forma eficaz as ações necessárias para reduzir os riscos a um nível aceitável, finalizando então esta etapa. O tratamento de risco sucede a análise/avaliação de risco e tem seus resultados dependentes desta etapa. Existem situações em que o tratamento de risco não chega a um nível de risco residual, sendo então necessária uma nova iteração na análise/avaliação de risco, com mudanças nas variáveis do contexto, seguida de uma fase adicional de tratamento de risco. Ao término da etapa de tratamento de risco, inicia-se a etapa de aceitação do risco, em que esta deve assegurar que os riscos residuais sejam explicitamente aceitos pelos gestores da organização (ABNT, 2008).

A norma ABNT NBR ISO/IEC 27005:2008 salienta que, durante o processo de gestão de risco de segurança da informação, é importante que os riscos e a forma com que são tratados sejam comunicados ao pessoal das áreas operacionais e gestores apropriados. Também é salientado que mesmo antes do tratamento de risco, informações sobre riscos identificados podem ser muito úteis para o gerenciamento de incidentes e ajudar a reduzir possíveis prejuízos (ABNT, 2008).

Até agora, uma visão superficial do processo de Gestão de Riscos em Segurança da Informação foi apresentada. As seções subseqüentes abordam de forma mais detalhada as duas tarefas da análise de risco para sua melhor compreensão, tendo em vista que o trabalho visa auxiliar diretamente nestas duas tarefas.

2.3.1 Identificação de riscos

A finalidade da identificação de riscos é desenvolver uma lista abrangente de fontes de risco e eventos que podem ter um impacto na consecução de cada um dos objetivos identificados nos contextos. Os riscos não identificados podem se tornar uma ameaça à organização ou fazer com que percam oportunidades importantes (CICCO, 2005). O propósito da identificação de riscos, segundo a norma ABNT NBR ISO/IEC 27005:2008, é determinar eventos que possam causar uma perda potencial e deixar claro como, onde e por que a perda pode acontecer. As etapas que servem de entrada para atividade de estimativa de risco são (ABNT, 2008):

- Identificação de ativos;
- Identificação de ameaças;

- Identificação de controles existentes;
- Identificação de vulnerabilidades; e
- Identificação de conseqüências.

Cada uma destas etapas é abordada separadamente e com melhores detalhes a seguir.

2.3.1.1 Identificação de ativos

Como visto na Seção 2.1 deste capítulo, um ativo é algo que tem valor para a organização e que, portanto, requer proteção. Para a identificação dos ativos, convém que se tenha em mente que um sistema de informação compreende mais que *hardware* e *software*. A norma ABNT NBR ISO/IEC 27005:2008 sugere que a identificação dos ativos seja executada com um detalhamento adequado que forneça informações suficientes para a análise/avaliação de riscos. O nível de detalhe usado na identificação dos ativos influência na quantidade geral de informação reunidas durante a análise/avaliação de risco. O detalhamento pode ser aprofundado em cada iteração da análise/avaliação de risco (ABNT, 2008).

Esta atividade tem como resultado uma lista de ativos com risco a serem gerenciados e uma lista dos negócios relacionados aos ativos e suas relevâncias (ABNT, 2008).

2.3.1.2 Identificação de ameaças

Uma ameaça tem o potencial de comprometer ativos (tais como, informação, processo e sistemas), por isso, também as organizações. Ameaças podem ser de origem natural ou humana e podem ser acidentais ou intencionais. É recomendado que tanto as ameaças acidentais, quanto as intencionais sejam identificadas genericamente e por classe (por exemplo, ações não autorizadas, danos físicos, falhas técnicas) e, quando apropriado, ameaças específicas identificadas dentro das classes genéricas. Isso significa que nenhuma ameaça é ignorada, incluindo as não previstas, mas que o volume de trabalho exigido é limitado (ABNT, 2008).

Algumas ameaças podem afetar mais de um ativo. Nesses casos, elas podem provocar impactos diferentes, dependendo de quais ativos são afetados.

Dados de entrada para a identificação das ameaças e estimativas de probabilidade de ocorrência podem ser obtidas dos responsáveis pelos ativos ou dos usuários, do pessoal, dos administradores das instalações e dos especialistas em segurança da informação, de peritos em segurança física, do departamento jurídico e de outras organizações, incluindo organismos legais, autoridades climáticas, companhias de seguros e autoridades governamentais nacionais.

Convém que experiências internas de incidentes e avaliações anteriores das ameaças sejam consideradas à avaliação atual. Pode ser útil a consulta a outro catálogo de ameaças a fim de completar a lista de ameaças genéricas, quando relevantes. Catálogos de ameaças e estatísticas são disponibilizados por organismos setoriais, governos nacionais, organismos legais, companhias de seguros etc (ABNT, 2008).

Esta atividade resulta em uma listagem com as ameaças e sua respectiva identificação do tipo e da fonte das ameaças (ABNT, 2008).

2.3.1.3 Identificação de controles existentes

Convém que a identificação dos controles existentes seja realizada para evitar custos e trabalho desnecessário, por exemplo: na duplicação de controles. Além disso, enquanto os controles existentes estão sendo identificados, convém que seja feita uma verificação para averiguar se estes estão funcionando corretamente. Uma referência aos relatórios já existentes de auditoria do Sistema de Gestão de Segurança da Informação (SGSI) pode reduzir o tempo gasto nesta tarefa.

Deve ser também levado em consideração a possibilidade de um controle selecionado falhar durante sua operação, tornando necessária a existência de controles complementares para tratar efetivamente o risco identificado (ABNT, 2008).

Para identificar controles existentes ou planejados, as seguintes atividades podem ser úteis (ABNT, 2008):

- Analisar de forma crítica os documentos contendo informações sobre os controles (por exemplo: os planos de implementação de tratamento de risco). Se os processos de gestão da segurança da informação estão bem documentados, convém que todos os controles existentes ou planejados e a situação de sua implementação estejam disponíveis;
- Verificar, com as pessoas responsáveis pela segurança da informação e com os funcionários, quais controles, relacionados ao processo de informação ou ao sistema de informação sob consideração, estão realmente implementados;
- Revisar, no local, os controles físicos, comparando os controles implementados com a lista de quais convém que estejam presentes; e verificar se aqueles implementados estão funcionando efetiva e corretamente; ou
- Analisar criticamente os resultados de auditorias internas.

Esta atividade resulta em uma lista de todos os controles implementados ou planejados, sua implementação e o seu nível de utilização (ABNT, 2008).

2.3.1.4 Identificação de vulnerabilidades

As vulnerabilidades podem ser identificadas nas seguintes áreas (ABNT, 2008):

- Organização;
- Processos e procedimentos;
- Rotinas de gestão;
- Recursos humanos;
- Ambientes físicos;
- Configuração de sistemas de informação;
- *Hardware*, *softwares* ou equipamentos de comunicação; e
- Dependência de entidade externa.

A presença de uma vulnerabilidade não causa prejuízo por si só, pois precisa haver uma ameaça presente para explora-lá. Uma vulnerabilidade que não tem uma ameaça correspondente pode não requerer implementação de um controle no presente momento, mas convém que nela seja reconhecida como tal e monitorada, no caso de haver mudanças. Um controle implementado, que funciona incorretamente, ou sendo usado incorretamente, pode, por si só, representar uma vulnerabilidade. Um controle pode ser eficaz ou não, dependendo do ambiente no qual ele opera. Inversamente, uma ameaça que não tenha vulnerabilidade correspondente pode não resultar em um risco.

Vulnerabilidades podem estar ligadas a propriedades dos ativos, as quais podem ser usadas de uma forma ou para um propósito diferente daquele para qual o ativo foi adquirido ou desenvolvido. Vulnerabilidades decorrentes de diferentes fontes precisam ser consideradas, por exemplo: as intrínsecas ao ativo e as extrínsecas.

Esta atividade resulta em uma lista de vulnerabilidades associadas aos ativos, às ameaças e aos controles; assim como uma lista de vulnerabilidades que não se refere a nenhuma ameaça identificada para análise (ABNT, 2008).

2.3.1.5 Identificação de conseqüências

Uma conseqüência pode ser, por exemplo, a perda da eficácia, condições adversas de operação, a perda de oportunidade de negócio, reputação afetada, prejuízo etc.

Essa atividade identifica o prejuízo ou as consequências para a organização que podem decorrer de um cenário de incidente. Um cenário de incidente é a descrição de uma ameaça

explorando uma vulnerabilidade ou um conjunto delas em um incidente de segurança da informação. O impacto dos cenários de incidentes é determinado considerando-se os critérios de impacto definidos durante a atividade de definição do contexto. Esta pode afetar um ou mais ativos ou apenas parte de um ativo. Conseqüências podem ser de natureza temporária ou permanente como no casa da destruição de um ativo (ABNT, 2008).

Esta atividade resulta em uma lista de cenários de incidentes com suas conseqüências associadas ao ativos e processos de negócio (ABNT, 2008).

2.3.2 Estimativa de Riscos

A estimativa de riscos pode seguir uma metodologia qualitativa ou quantitativa ou ainda uma combinação de ambas. Freqüentemente, a estimativa qualitativa é utilizada primeiro para se obter uma indicação geral do nível de risco e para revelar grandes riscos. Depois desta estimativa prévia, poderá ser utilizada a estimativa quantitativa (ABNT, 2008).

A estimativa qualitativa utiliza uma escala de atributos que descrevem a magnitude das conseqüências potencias, como por exemplo, pequena, média e grande, e também a probabilidade destas conseqüências ocorrerem. Por sua vez, a estimativa quantitativa utiliza uma escala numérica tanto para conseqüências quanto para probabilidade (ABNT, 2008).

A estimativa quantitativa pode ser relacionada diretamente com os objetivos de segurança da informação e interesses da organização, no entanto, leva desvantagem sobre a estimativa qualitativa com relação a falta de dados de novos riscos ou sobre fragilidades de segurança da informação (ABNT, 2008).

Os valores atribuídos aos ativos, seja da forma qualitativa ou quantitativa, são utilizados na avaliação das conseqüências, ou seja, do impacto no negócio. Este impacto também pode ser representado de forma qualitativa ou quantitativa. A valorização dos ativos deve levar em consideração sua criticidade e sua importância para a realização dos objetivos de negócio da organização (ABNT, 2008).

2.3.3 Gestão de Riscos e Redes Bayesianas

A Seção 2.3 apresentou uma breve visão sobre Gestão de Risco aplicada a Segurança da Informação. A atual seção tem como objetivo apenas criar uma relação entre Gestão de Riscos e Redes Bayesianas, que serão apresentadas com mais detalhe no Capítulo 3.

Como mencionado na Seção 2.3.2, a estimativa de risco pode ser qualitativa ou

quantitativa. Um possível abordagem para se obter uma estimativa de risco quantitativa é através da utilização de Redes Bayesianas, visto que cada nó desta rede representa uma probabilidade quantitativa de um determinada evento.

Esta abordagem é utilizada em trabalhos como Dantu e Kolan (2005), Fenz e Hudec (2009), Fenz e Neubauer (2009), Lee et al. (2008) e Lucas et al. (2004) para calcular riscos operacionais, em projetos, de saúde, de segurança etc.

Dantu e Kolan (2005) formularam um mecanismo para estimar o nível de risco de recursos críticos através de um grafo de ataques baseados no comportamento do atacante. Este é utilizado para calcular o nível de risco dos recursos.

Fenz e Hudec (2009) propõem um método para geração de Redes Bayesianas com base em ontologia, possibilitando identificar o grau de certeza sobre um determinado evento e sua influência sobre outros componentes da organização.

Fenz e Neubauer (2009) mostram como determinar a probalidade de ameaças com a utilização de ontologias e Redes Bayesianas com o objetivo de possibilitar a gestores de risco uma visualização compreensível e quantitativa do estado da segurança da informação em suas organizações.

Lee et al. (2008) utilizam Redes Bayesianas para construir um diagrama de causa e conseqüência, considerando isso uma metodologia adequada para gestão de risco em projetos com processos integrados e sistemáticos.

Lucas et al. (2004) apresentam o estado da arte de Redes Bayesianas aplicada a medicina, apresentando os problemas de incerteza da biomedicina em que Redes Bayesianas geralmente são empregadas.

Neste seção foi apresentada apenas um pequeno resumo dos trabalhos que utilizam Redes *Bayesianas* aplicada a Gestão de Risco, visto que os trabalhos relacionados serão melhor detalhados no Capítulo 5.

2.4 BASE DE DADOS DE VULNERABILIDADES

Esta seção apresenta a base de dados de vulnerabilidade *National Vulnerability Database (NVD)* e como ela é organizada. Como o trabalho é baseado nesta base de dados, esta seção é fortemente baseada na documentação disponibilizada pela NVD.

Para se manter um sistema de tecnologia de informação seguro, é necessária a utilização das melhores práticas em Segurança da Informação. Para isso, base de conhecimentos de vulnerabilidades, estados de sistema e *checklists* de segurança da informação devem ser criados. Para que isso possa se tornar de fato uma melhor prática,

as descrições de vulnerabilidades de configurações de sistemas devem seguir um padrão comum de nomes (BUTTNER e ZIRING, 2009).

A utilização de um padrão comum para descrever vulnerabilidades e configurações, provê interoperabilidade, melhora da correlação dos resultados e facilita a coleta de métricas de Segurança da Informação. Outra motivação para esta padronização é a tendência para a automação em práticas de segurança, em que um padrão é essencial, sendo então possível identificar claramente uma vulnerabilidade e as plataforma de Tecnologia da Informação as quais elas se aplicam (BUTTNER e ZIRING, 2009).

O padrão de nomenclatura que está se popularizando é o *Common Vulnerabilities and Exposures (CVE)*, que é utilizado para identificar e descrever plataformas de Tecnologia da Informação e Vulnerabilidades. Um padrão similar destinado a configuração de plataformas de Tecnologia da Informação é o *Common Configuration Enumeration (CCE)*. Esta especificação descreve um esquema estruturado de atribuição de nomes para as plataformas informáticas (*hardware*, sistemas operacionais e aplicações), que recebe o nome de *Common Platform Enumeration (CPE)*(BUTTNER e ZIRING, 2009).

2.4.1 Common Platform Enumeration

O *Common Platform Enumeration (CPE)* é baseado na sintaxe genérica para *Uniform Resource Identifiers (URI)*[3] e especifica sintaxe de nomes e convenções para a construção de nomes e informação sobre os produtos, criando um dicionário chamado *CPE Dictionary*, que pode ser atualizado através do endereço eletrônico `http://cpe.mitre.org`.

Esta especificação foi criada com base em três propósitos (BUTTNER e ZIRING, 2009):

1. **Nomes Comuns** - Se uma ferramenta cria uma descrição e/ou resultado aplicável a um certo tipo de plataforma, uma segunda ferramenta deve ser capaz de compreender o que foi descrito pela primeira ferramenta e saber que ação tomar;

2. **Correspondência** - Uma configuração manual pode ser aplicável para todas as versões de um determinado sistema operacional e ser atribuído um nome a este CPE. Portanto, a ferramenta pode consultar um sistema e determinar um nome mais específico, que inclui uma versão exata, ou talvez uma determinação complexa utilizando a plataforma de descrição CPE .

3. **Relatórios** - Quando utilizado em relatórios sobre um sistema específico ou de um grupo de sistemas, um nome CPE permite a correlação dos resultados com fontes

[3]Maiores informações sobre *Uniform Resource Identifiers (URI)* podem ser obtidas em http://labs.apache.org/webarch/uri/rfc/rfc3986.html

adicionais. Se um nome CPE é atribuído para cada sistema, esse identificador pode ser utilizado para pesquisar em outro sistema as possíveis vulnerabilidades que são aplicáveis para esse tipo de plataforma.

Um nome CPE sempre começa com um prefixo URI "cpe:"[4] e é seguido de uma estrutura que identifica qualquer plataforma. A estrutura de um nome CPE é formada por um ou mais componentes separadas por ":". O primeiro componente identifica a parte da plataforma que está sendo especificada; o segundo identifica um vendedor ou fornecedor do item; o terceiro identifica o nome do produto; o quarto indica uma versão do produto; o quinto é utilizado para determinar a atualização ou *service pack*; o sexto identifica a edição e o sétimo é usado para especificar internacionalização das informações. O exemplo abaixo representa a estrutura completa de um nome CPE.

```
cpe:/part:vendor:product:version:update:edition:language
```

Estrutura de um nome CPE.

O CPE permite componentes nulos, o que significa que qualquer aspecto do respectivo componente é válido. Por exemplo, o nome CPE abaixo é designado a todas as edições do *Microsoft Windows XP Professional*, independentemente do *service pack* nível.

```
cpe:/o:microsoft:windows_xp:::pro
```

Para forçar que um determinado componente seja configurado como vazio, é utilizado o "-". Quando empregado o "-", as consultas só retornarão nomes em que o determinado componente é nulo ou vazio, ao contrário do exemplo anterior, em que retornaria qualquer valor para aquele componente. Por exemplo, uma aplicação não pode ter diferentes edições. Um nome CPE para tal aplicativo pode usar o "-" para a edição componente, como visto abaixo.

```
cpe:/a:acme:product:1.0:update2:-:en-us
```

O sinal "-" também é utilizado para determinar a primeira versão de uma plataforma que se identifica com o componente. Como, por exemplo, a primeira liberação de uma aplicação, antes que exista uma atualização, como pode ser visto a seguir:

```
cpe:/a:acme:product:1.0:-
```

[4]O prefixo "cpe" não é um prefixo oficializado pela *Internet Assigned Numbers Authority - IANA*. Maiores informações sobre a IANA em http://www.iana.org.

Após a primeira atualização:

```
cpe:/a:acme:product:1.0:update1
```

Ambas opções seriam selecionadas em uma consulta em que o componente referente à atualização esteja com o valor vazio, ou seja:

```
cpe:/a:acme:product:1.0:
```

2.4.1.1 Componentes de um nome CPE

Da estrutura especificada de um nome CPE, o primeiro componente determina a particularidade da plataforma que está sendo identificada, que são definidas pelos seguintes códigos:

- **h** - *hardware*;
- **o** - sistema operacional; e
- **a** - aplicação.

O segundo componente do nome CPE, que determina o vendedor ou fornecedor do item, pode ser fonte de divergências pelas diversas forma de expressar o nome de uma organização. Para evitar esta divergência, é utilizado o nome do domínio DNS da organização, mesmo que este domínio seja diferente do nome da organização. A tabela 2.1 mostra alguns exemplos que representam esta situação.

Nome da Organização	**Domínio**	**Componente CPE**
Cisco Systems, Inc.	cisco.com	cisco
The Mozilla Foundation	mozilla.org	mozilla
University of Oxford	oxford.ac.uk	oxford

Tabela 2.1: Relação entre nomes de organizações, domínios e descrição do vendedor de um nome CPE

Algumas organizações podem utilizar mais de um domínio, como, por exemplo, hewlett-packard.com e hp.com. Nestes casos, deve ser utilizado o nome que a organização utiliza em publicidade ou documentações. No caso do exemplo anterior, seria utilizando "hp" para referenciar o componente do nome CPE.

Caso o nome específico do domínio de uma organização seja compartilhado por uma segunda organização, diferenciando apenas o sufixo do endereço DNS, deve-se

então utilizar o nome completo do domínio DNS. Por exemplo, se ao registrar um nome `cpe:/a:acme` for identificado que o mesmo é referente a organização com domínio acme.com, então dever utilizar o nome de componente `cpe:/a:acme.org`.

Em casos em que o vendedor ou fornecedor não possui um endereço DNS, deve-se então utilizar o termo pelo qual o vendedor ou fornecedor seja mais conhecido, sempre substituindo os espaços entre palavras por sublinhado ("_").

Quando se está gerando um nome para uma aplicação que não possui um vendedor ou fornecedor, comum em aplicações *open-source*, utiliza-se o nome do desenvolvedor para determinar vendedor ou fornecedor do componente. Neste caso, também se substitui os espaços por sublinhado.

Em situações em que um vendedor ou fornecedor muda o seu nome devido a sua comercialização ou aquisição, o nome CPE original deve permanecer o mesmo, sendo gerado um novo nome CPE contendo o nome do novo vendedor ou fornecedor.

Em tal caso, o nome utilizado no terceiro componente do nome CPE, refere-se ao nome do produto. Utiliza-se o nome mais comum para o produto, considerando materiais de publicidade e documentação. Produtos que possuem um nome composto por mais de uma palavra, devem ser descritos integralmente, sempre substituindo os espaços por sublinhados. Como por exemplo, um produto com o nome de *ZoneAlarm Internet Security Suite version 7.0* deve ter um nome CPE:

```
cpe:/a:zonelabs:zonealarm_internet_security_suite:7.0
```

Em casos em que o produto possui um nome composto por mais de uma palavra e o vendedor ou fornecedor utiliza uma abreviatura oficial para o mesmo, esta última é utilizada. A tabela 2.2 apresenta alguns exemplos deste caso.

Nome do Produto	**Abreviação**
Internet Explorer	ie
Java Runtime Environment	jre

Tabela 2.2: Abreviações de nomes de produtos

O quarto componente do nome CPE, refere-se à versão da plataforma descrita, utilizando a versão de forma idêntica a utilizada pelo produto, empregando como delimitadores, pontos, vírgulas, sublinhados, etc.

O mesmo ocorre com o quinto elemento do nome CPE, que refere-se à atualização da plataforma descrita, em que é utilizado o nome determinado pelo vendedor ou fornecedor. Como, por exemplo, para determinar suas atualizações a *Microsoft* utiliza o formato *Servive Pack 1 (ou sp1)* e a *Red Hat* que utiliza o formato *Update 1*.

O sexto componente, referente à edição da plataforma, é utilizado para determinar o objetivo específico da arquitetura de *hardware* ou aplicação, ou seja, diferentes edições de uma plataforma. Um exemplo de edição para *hardware* pode ser a arquitetura a qual se refere, *i386* e x*64*. Um exemplo de edição para aplicação é o sistema operacional de destino da aplicação, que pode ser *Windows*, *Linux* ou *Macintosh OS X*, ou até mesmo, a edição do sistema operacional, como *Windows Vista Hom*e, *Professional* ou *Ultimate*.

O último componente de um nome CPE é referente à linguagem da plataforma descrita, que é representada pela IETF RFC 4646[5].

Algumas palavras e frases que são comumente utilizadas no meio da Tecnologia da Informação com relação à denominação de produtos, e que possuem abreviaturas já conhecidas, são utilizadas para facilitar a utilização de nomes CPE e também para reduzi-los.

A tabela 2.3 apresenta as principais abreviações utilizadas em nomes de produtos.

Termo Completo	**Abreviação**
advanced	adv
professional	pro
server	srv
standard	std
edition	ed
version 3.4	3.4
patch level 3	pl3
release 3	r3
release candidate 2	rc3
service pack 4	sp4
support pack 2	sup2
service release 2	sr2
security rollup	sru
general availability	ga

Tabela 2.3: Abreviações utilizadas em nomes de produtos.

A lista completa e atualizada de abreviaturas pode ser obtida através do endereço eletrônico `http://cpe.mitre.org`.

2.4.2 Common Vulnerability Scoring System

O *Common Vulnerability Scoring System* (CVSS) é um *framework* aberto que relaciona características de comunicação e impacto de vulnerabilidades de TI para obter

[5]IETF: *Internet Engineering Task Force*; RFC: *Request for Comments*; A IETF RFC 4646 recebe o nome de *Tags for Identifying Languages*. Maiores detalhes podem ser obtidos através do endereço http://www.ietf.org/rfc/rfc4646.txt

uma pontuação. Estas pontuações auxiliam na tarefa dos gestores de Segurança da Informação a priorizar as vulnerabilidades que serão abordadas em seu ambiente, visto que, geralmente, devem monitorar um grande número de *hardware* e aplicativos, que por sua vez, possuem inúmeras vulnerabilidades. A documentação do CVSS é disponibilizada em Mell et al. (2007) e esta seção está totalmente baseada nesta documentação.

O CVSS está dividido em três grupos de métricas que fornecem uma pontuação entre 0 e 10, que são: básicas, temporais e de ambiente.

Métricas básicas: Representam características fundamentais e intrínsecas de uma vulnerabilidade que é constante no tempo e no ambiente do usuário;

Métricas Temporais: Representam as características da vulnerabilidade que mudam no tempo, porém, não no ambiente de usuário;

Métricas de Ambiente: Representam as características das vulnerabilidades que são únicas e relevantes para o ambiente de usuário.

Quando os valores das métricas básicas são definidos, uma equação calcula a pontuação, retornando um valor entre 0 e 10 e um vetor é criado, como pode ser visualizado na Figura 2.3.

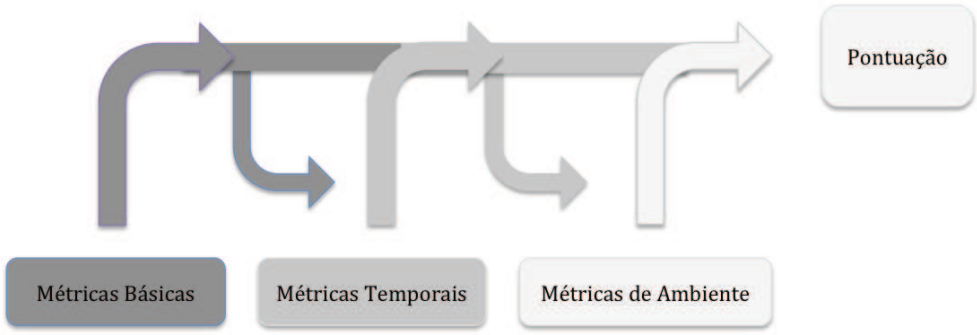


Figura 2.3: Métricas e equações do CVSS (MELL et al., 2007).

Como pode-se observar no fluxo apresentado na Figura 2.3, a utilização dos grupos de métricas temporais e de ambientes são opcionais para se obter a pontuação da vulnerabilidade. As métricas temporais e de ambiente somente são utilizadas, caso seja necessário refinar a pontuação obtida a partir das métricas básicas.

2.4.2.1 Métricas básicas

Este grupo capta as características das vulnerabilidades que são constantes no tempo e nos ambientes de usuários. Estas características são o vetor de acesso, a complexidade de acesso, autenticação e impacto.

O vetor de acesso corresponde a como a vulnerabilidade é explorada e seus possíveis valores são:

***Local*:** Que corresponde às vulnerabilidade que necessitam de acesso local para serem exploradas.

***Adjacent Network*:** Que correspondem às vulnerabilidades que necessitam de acesso a uma rede adjacente para serem exploradas, como por exemplo, acesso a rede local, acesso via Bluetooth, IEEE 802,11 e um segmento da rede.

***Network*:** Que corresponde às vulnerabilidades que podem ser exploradas através da rede.

A complexidade de acesso corresponde ao nível de dificuldade que o atacante tem para explorar a vulnerabilidade e seus possíveis valores são:

***High*:** Existem condições de acesso especializadas, como por exemplo, em casos em que algum momento do ataque é necessário acesso privilegiado ao sistema, o atacante necessita de engenharia social ou a configuração da vulnerabilidade é muito rara;

***Medium*:** Quando as condições de acesso são menos especializadas, como por exemplo, em casos em que o atacante já faz parte de um grupo de usuários do sistema, onde o atacante já tenha alguma informação que facilite o ataque ou que a configuração da vulnerabilidade não é padrão, mas é comumente encontrada;

***Low*:** Quando condições de acesso especializado não existem, com por exemplo, quando o produto afetado requer acesso anônimo e não confiável a uma grande número de sistemas e usuários, a configuração da vulnerabilidade é padrão e ubíqua ou quando o ataque pode ser efetuado de forma manual e com pouco conhecimento ou com poucas informações.

A métrica de autenticação corresponde ao número de vezes que o atacante deve realizar autenticação para conseguir explorar a vulnerabilidade. Seus possíveis valores são:

***Multiple*:** Quando o atacante deve autenticar no sistema duas ou mais vezes para conseguir explorar a vulnerabilidade;

***Single*:** Quando o atacante necessita autenticar apenas uma vez para conseguir explorar o vulnerabilidade;

***None*:** Quando o atacante não necessita autenticação para explorar a vulnerabilidade.

As métricas de impacto são subdivididas em três métricas, referentes ao impacto na CID, ou seja, na Confidencialidade, Integridade e Disponibilidade. As três métricas de impacto podem assumir os seguintes valores:

None Quando a vulnerabilidade não impacta na confidencialidade, integridade ou disponibilidade;

Partial Quando a vulnerabilidade impacta parcialmente na confidencialidade, integridade ou disponibilidade;

Complete Quando a vulnerabilidade impacta totalmente na confidencialidade, integridade ou disponibilidade.

É importante ressaltar que existe uma métrica para confidencialidade, uma segunda para integridade e outra para disponibilidade, sendo que as três possuem as mesmas opções de valores.

2.4.2.2 Métricas temporais

A ameaça representada por uma vulnerabilidade pode mudar ao longo do tempo. Os três fatores que podem ser alterados ao longo do tempo que são utilizados pelo CVSS são: a confirmação dos detalhes de uma vulnerabilidade, o estado de correção da vulnerabilidade e a disponibilidade de códigos para explorar a vulnerabilidade. As métricas que compões este grupo são: explorabilidade, nível de correção e confiança do relatório.

A explorabilidade corresponde ao estado atual de técnicas e códigos para explorar a vulnerabilidade e seus possíveis valores são:

***Unproven*:** Quando nenhum código de exploração encontra-se disponível, ou uma exploração é meramente teórica;

***Proof-of-Concept*:** Quando existe prova de conceito ou uma demonstração de ataque, sendo este não prático para a maioria dos sistemas disponíveis;

***Functional*:** Quando o código de exploração está disponível e funcional para a maioria das situações onde a vulnerabilidade existe;

***High*:** Quando a vulnerabilidade pode ser explorada de forma automática e os detalhes são amplamente disponíveis;

***Not Defined*:** Sinal para ignorar a métrica no resultado final.

O nível de correção de uma vulnerabilidade é um fator importante para a priorização. Um típico caso de vulnerabilidade não corrigida é quando ela é recém publicada. As soluções alternativas ou correções podem gerar uma correção, ou uma atualização oficial. Cada uma destas fases influi na pontuação temporal. Os valores que esta métrica pode assumir são:

***Official Fix*:** Quando uma solução completa do fornecedor está disponível;

***Temporary Fix*:** Quando existe uma solução oficial, no entanto, temporária;

***Workaround* :** Quando existe uma solução de contorno não oficial;

***Unavailable* :** Quando não existe uma solução disponível;

***Not Defined*:** Sinal para ignorar a métrica no resultado final.

Essa métrica mede o grau de confiança na existência da vulnerabilidade e da credibilidade dos detalhes técnicos conhecidos. Em alguns casos, apenas a existência de vulnerabilidades é divulgada, mas sem detalhes específicos. A vulnerabilidade pode ser posteriormente confirmada. A urgência em tratar uma vulnerabilidade é maior quando uma vulnerabilidade é confirmada. Esta métrica também sugere o nível de conhecimento técnico disponível para possíveis atacantes. Seus possíveis valores são:

***Unconfirmed*:** Quando não existe uma única fonte não confirmada ou possivelmente vários relatos conflitantes. Como por exemplo, boatos;

***Uncorroborated*:** Quando existem várias fontes não oficiais, possivelmente incluindo empresas de segurança ou organizações independentes de pesquisa;

***Confirmaded* :** Quando a vulnerabilidade é reconhecida pelo fornecedor ou autor do tecnologia afetada ou quando publicada uma prova de conceito ou código para explorar a vulnerabilidade;

***Not Defined*:** Sinal para ignorar a métrica no resultado final.

2.4.2.3 Métricas de Ambiente

Diferentes ambientes podem ter uma grande influência sobre o risco que representa uma vulnerabilidade para uma organização e seus parceiros. Este grupo de métricas coleta características da vulnerabilidade que estão associadas a um usuário do ambiente de TI.

As métricas que compõem este grupo são dados colaterais em potencial, distribuição do objetivos e requisitos de segurança.

A métrica de danos colaterais em potencial mensura a potencial perda de ativos físicos por meio de danos ou furto de bens ou equipamentos, assim como a perda de produtividade ou da receita. Os possíveis valores para esta métrica são:

***None*:** Quando não existem potenciais dados para perda de vida, ativos físicos, produtividade ou receita;

***Low*:** Quando uma exploração bem-sucedida desta vulnerabilidade pode resultar em danos físicos ou materiais leves;

***Low-Medium*:** Quando uma exploração bem-sucedida desta vulnerabilidade pode resultar em danos físicos ou materiais moderada;

***Medium-High*:** Quando uma exploração bem-sucedida desta vulnerabilidade pode resultar em danos físicos ou materiais ou perda substancial;

***High*:** Quando uma exploração bem-sucedida desta vulnerabilidade pode resultar em danos catastróficos ou danos materiais e perdas;

***Not Defined*:** Sinal para ignorar a métrica no resultado final.

A métrica de distribuição de alvos mensura a proporção do sistema vulnerável. Entende-se como um indicador para cada ambiente a fim de aproximar o percentual do sistema que pode ser afetado pela vulnerabilidade. Os seus possíveis valores são:

***None*:** Quando não existem alvos no ambiente, ou os alvos são altamente especializados que só existem em um ambiente de laboratório;

***Low*:** Quando existem alvos dentro do ambiente, mas em pequena escala.;

***Medium*:** Quando existem objetivos dentro do ambiente, mas em escala média;

***High*:** Quando existem alvos dentro do ambiente em escala considerável;

***Not Defined*:** Sinal para ignorar a métrica no resultado final.

A métrica de requisitos de segurança permite ao analista personalizar a pontuação do CVSS de acordo com a importância do ativo para a organização, mensurando em termos de confidencialidade, integridade e disponibilidade. O pleno efeito sobre a pontuação do ambiente é determinado pela sua métrica básica de impacto. É importante ressaltar que as métricas básicas de impacto não são alteradas. Apenas é atribuído um impacto do CID

as métricas de ambiente. Atribuindo pesos de impacto ao ambiente, é possível realizar uma ponderação do impacto, Por exemplo, a métrica de impacto na confidencialidade (básica) diminui se o peso da exigência de confidencialidade (no ambiente) for baixo. Os valores possíveis para esta métrica são:

***Low*:** Quando a perda de confidencialidade, integridade, disponibilidade é suscetível de ter um efeito limitado sobre a organização ou os indivíduos associados com a organização;

***Medium*:** Quando a perda de confidencialidade, integridade, disponibilidade é susceptível de ter um efeito grave sobre a organização ou os indivíduos associados com a organização;

***High*:** Quando a perda de confidencialidade, Integridade, disponibilidade é susceptível de ter um efeito catastrófico sobre a organização ou os indivíduos associados com a organização;

***Not Defined*:** Sinal para ignorar a métrica no resultado final.

2.4.2.4 Cálculo da pontuação

A equação para obter a pontuação básica é dada pela seguinte formula:

$$BS = 0,6 * I + 0,4 * E - 1,5 * f(I) \tag{2.1}$$

onde: BS é a pontuação básica, I é o impacto, E é a explorabilidade e $f(I)$ é 0 se impacto = 0, caso contrário 1.176.

O impacto é obtido pela equação:

$$I = 10.41 * (1 - (1 - Conf) * (1 - Int) * (1 - Avail)) \tag{2.2}$$

onde: I é o impacto, $Conf$ é 0 se o impacto na confidencialidade for igual *none*, 0,275 se igual a *partial*, 0,66 se igual *complete*, Int é 0 se o impacto na integridade for igual *none*, 0,275 se igual a *partial*, 0,66 se igual *complete*, $Avail$ é 0 se o impacto na disponibilidade for igual *none*, 0,275 se igual a *partial*, 0,66 se igual *complete*.

A explorabilidade é obtida pela equação:

$$E = 20 * AV * AC * AH \tag{2.3}$$

onde: E é a explorabilidade, AV é 0,395 se o vetor de acesso é igual a *required local access*, 0,646 se igual a *adjacent network accessible* e 1 se igual a *network accessible*, AC é 0,35 se a complexidade de acesso igual a *high*, 0,61 se igual a *medium* e 0,71 se igual a *low*, AH é 0,45 se a autenticação igual a *requires multiple instances of authentication*, 0,56 se igual a *requires single instance of authentication* e 0,704 se igual a *requires no authentication*.

A equação para se obter a pontuação temporal é dada pela fórmula:

$$TS = BS * E * RL * RC \tag{2.4}$$

onde: TS é a pontuação temporal, BS é a pontuação básica, E é 0,85 se exploração igual a *unproven*, 0,9 se igual a *proof-of-concept*, 0,95 se igual a *functional*, 1 se igual *high* e 1 se igual a *not defined*, RL é 0,87 se o nível de correção igual a *official-fix*, 0,90 se igual a *temporay-fix*, 0,95 se igual a *wokaround*, 1 se igual a *unavailable* e 1 se igual a *not defined* e RC é 0,90 se o grau de confiança igual a *unconfirmed*, 0,95 se igual *uncorroborated*, 1 se igual a *confirmed* e 1 se igual a *not defined*;

A equação para obter a pontuação de ambiente é dada pela seguinte fórmula:

$$ES = (AT + (10 - AT) * CDP) * TD \tag{2.5}$$

onde: AT é o ajuste temporal recalculado na fórmula da pontuação básica, alterando a sub-equação de impacto pela sub-equação de ajuste do impacto, CDP é 0,1 se potencial dano colateral for igual *low*, 0,3 se igual a *low-medium*, 0,4 se igual a *medium-high*, 0,5 se igual a *high* ou 0 caso contrário, TD é 0.25 se a distribuição de objetivos igual a *low*, 0.75 se igual *medium* e 1 se igual a *high* ou *not defined*.

O ajuste do impacto é dado pela equação:

$$\begin{aligned} AI = \min(10; 10,41 * (1 - (1 - ConfImp * ConfReq) * (1 - IntImp * IntReq) \\ * (1 - AvailImp * AvailReq))) \end{aligned} \tag{2.6}$$

onde: AI é o ajuste de impacto, $ConfImp$ é o impacto na confidencialidade. $ConfReq$ é a confidencialidade necessária, $intImp$ é o impacto na integridade, $intRe$ é a integriadde necessária, $AvailImp$ é o impacto na disponibilidade e $AvailReq$ é a disponibilidade necessária.

3 REDES BAYESIANAS

No Capítulo 2, foi apresentada a definição de segurança da informação, sua relação com Governança em Segurança da Informação o processo de Gestão de Risco em Segurança da Informação, em que técnicas probabilísticas como Redes Bayesianas foram comentadas. Este capítulo aborda de forma breve este assunto.

A Seção 3.1 introduz redes causais e o conceito de *d-separation*, que servem como base para as definições de Redes Bayesianas apresentadas na Seção 3.2.

3.1 REDES CAUSAIS E *D-SEPARATION*

As definições apresentadas nesta seção são baseadas no exemplo de Jensen e Nielsen (2007), que se refere ao raciocínio humano no dia-a-dia.

> "Pela manhã, meu carro não ligará. Posso escutar o motor de arranque, mas ele não liga. Podem existir diversas razões para o meu problema. Eu posso escutar o motor de arranque, então a bateria deve ter carga. Portanto, as causas mais prováveis são que a gasolina tenha sido roubada durante a noite ou que as velas estejam sujas. Pode ser também que o carburador esteja sujo, uma conexão do sistema de ignição solta ou alguma coisa mais séria. Para encontrar uma saída, eu primeiramente olho para o medidor de combustível. Ele indica meio tanque, então eu decido limpar as velas."

Para que um computador tenha o mesmo tipo de raciocínio, é necessário responder questões como: "O que me faz concluir que a causa provável é roubo de combustível e sujeira nas velas?" ou "Por que a decisão de olhar o medidor de combustível e como essa observação pode me fazer concluir que o problema aparente é relacionado às velas?".

Para isso é necessário meios de se representar o problema e também de realizar inferência nesta representação, possibilitando que um computador simule esses tipos de raciocínios e, talvez, fazer isso melhor e mais rápido que humanos.

No raciocínio lógico, são utilizados quatro tipo de conexões lógicas: conjunção, disjunção, implicação e negação. Um exemplo de uma declaração lógica simples pode ser, "se está chovendo, então a grama está molhada". A partir um conjunto de declarações lógicas, é possível deduzir novas declarações. Das duas declarações "se está chovendo, então a grama está molhada" e "a grama não está molhada", então pode-se inferir que não está chovendo.

Quando se lida com eventos de incerteza, utilizar, preferencialmente, conectividades similares com certezas é melhor que usar ligações de valores de verdade. Então deve-se estender os valores de verdade da lógica proposicional para certezas, que são números entre 0 e 1. Quanto maior for o número, maior é a certeza. O valor 0 significa que a certeza é falsa e o valor 1, significa que a certeza é verdadeira.

Isso possibilita o uso de declarações tais como, "se eu tomar uma xícara de café no intervalo, eu permanecerei acordado na próxima aula com 0,5 de certeza" ou "se eu der uma pequena caminhada durante o intervalo, eu permanecerei acordado na próxima aula com 0.8 de certeza". Suponhamos que eu faça uma caminhada e ainda tome um café no intervalo. Qual a certeza de eu permanecer acordado? Para responder isso, é necessário uma função que combine duas certezas, no caso $0,5$ e $0,8$ e retorne um número, que deve ser a certeza resultante da combinação das certezas de duas declarações.

O mesmo é necessário para o encadeamento: "se a então b com x de certeza" e "se b então c com y de certeza". Sabendo a, então qual a certeza de c? Qualquer função para combinação e encadeamento levam a algumas situações que conduzem para conclusões erradas. Outro problema que também pode ser encontrado, que também é um problema de raciocínio lógico, e dedução e pode ser exemplificado da seguinte maneira: eu tenho a regra "uma mulher tem cabelos longos com 0,7 de certeza". Se eu olhar uma pessoa com cabelo longo, o quanto eu posso inferir sobre o sexo desta pessoa?

Um meio de estruturar uma situação para o raciocínio sobre incerteza é construir um grafo representando relações causais entre eventos. Para exemplificar, é utilizado o exemplo do problema do carro.

Assume-se os eventos {sim, não} para "combustível?", {sim, não} para as "velas estão limpas?", {cheio, 1/2, vazio} para "medidor de combustível" e {sim, não} para "o carro liga?". Sabendo qual o estado do "combustível?" e do estado "as velas estão limpas?", tem-se então um impacto causal sobre o estado "o carro liga?". O mesmo acontece para o estado "combustível?", que impacta no estado "medidor de combustível". Essa relação pode ser vista na Figura 3.1, que apresenta em forma de grafo a direção dos impactos entre os eventos.

Sabendo que as velas não estão limpas, a certeza que o carro não ligará aumenta. No entanto, a situação do exemplo é oposta. Como a certeza sobre o evento "o carro liga?" se move em uma direção negativa, é possível encontrar as possíveis causas, que são "combustível?" e "as velas estão limpas?", visto que ambos se relacionam com o evento "o carro liga?" positivamente.

Analisando o "medidor de combustível" é possível obter uma informação relacionada com o problema. Se ele estiver marcando 1/2 tanque, é possível mover-se na direção inversa, ou seja, de forma negativa e determinar a certeza relacionada ao evento

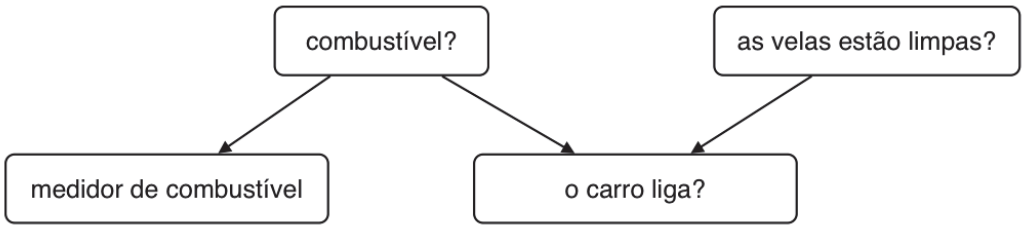


Figura 3.1: Uma Rede Causal para o problema ao ligar o carro (JENSEN e NIELSEN, 2007).

"combustível?". Assim é possível deduzir que provavelmente o motivo do problema não é falta de combustível, dado que o medidor está marcando 1/2 tanque.

Este raciocínio sobre incerteza foi estruturado a partir de um grafo representando relações causais. Grafos também são utilizados em Redes Causais. Essas redes são compostas por um conjunto de variáveis e um conjunto de associações direcionadas, que matematicamente, recebe o nome de grafos direcionados.

Em grafos direcionados, utiliza-se palavras referentes a relações familiares para representar a relação entre as variáveis. Se houver uma associação de A para B, então se diz que B é filho de A e A é pai de B. Em uma rede causal, uma variável representa um conjunto de possíveis estados de casos. Uma variável está exatamente em um destes estados.

Um exemplo simples de uma rede causal é apresentado na Figura 3.2. Neste exemplo, A tem influência sobre B, que por sua vez tem influência sobre C. Uma evidência sobre A influenciará na certeza de B, que tem influência sobre C. Sendo assim, uma influência sobre C influenciará em A através de B. Se o estado de B é conhecido, então o canal é bloqueado e A e C se tornam independentes. Então se diz que A e C são *d-separados* dado B. Quando o estado de uma variável é conhecido, diz-se que a variável está instanciada.

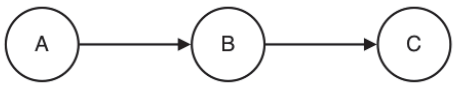
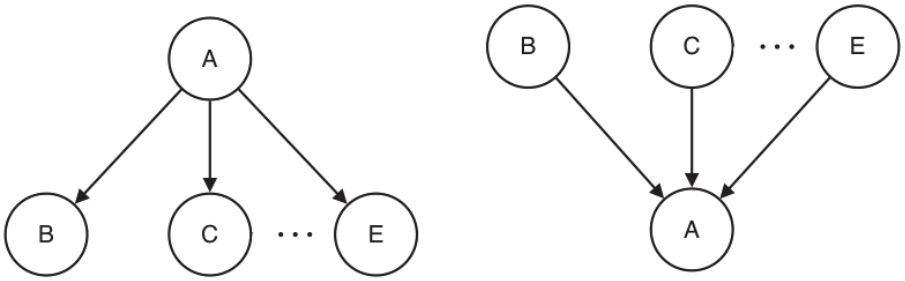


Figura 3.2: Conexão serial

Com base no exemplo da Figura 3.2, conclui-se que a evidência talvez seja transmitida através de uma conexão serial a menos que o estado da variável na conexão seja conhecido.

Existem dois tipos de conexão: (i) de divergência e (ii) de convergência. Estas são exemplificadas nas Figuras 3.3a e 3.3b.



(a) Conexão divergente (b) Conexão convergente

Figura 3.3: Exemplo de conexões em uma Rede Causal.

No exemplo da Figura 3.3a a influência pode passar entre todos os filhos de A, desde que A seja conhecido. Então $B, C, \ldots, E$ são d-separados dado A. A Figura 3.3b mostra um caso que requer mais cuidado. Neste exemplo, se não se sabe nada sobre A, exceto que talvez seja inferido de seus pais $B, \ldots, E$, então os pais são independentes, ou seja, uma evidência sobre um deles não pode influenciar nos demais através de A.

Tanto no exemplo da Figura 3.2, quanto no da Figura 3.3a, foi comentado que um variável é d-separada de outra. Diz-se que uma variável A é d-separada (d para grafo direcionado, do inglês *directed graph*) de outra B quando entre todos os caminos entre A e B, haja uma variável intermediária V, distinta de A e B, de tal forma que:

- a conexão seja serial ou divergente e V seja instanciada; ou
- a conexão seja convergente e nem V, nem qualquer valor descendente de V, tenha recebido evidência.

Se A e B não são d-separados, então são d-conectados.

Os temas abordados até então neste capítulo, servem como base para as definições de redes bayesinas, que serão apresentadas na Seção 3.2.

3.2 DEFINIÇÃO DE REDES BAYESIANAS

Segundo Jensen e Nielsen (2007), uma Rede Bayesiana consiste em:

- Um conjunto de variáveis e um conjunto de arestas direcionadas entre variáveis;
- Cada variável tem um conjunto finito de estados mutuamente exclusivos;
- As variáveis, juntamente com sua aresta direcionada, formam um grafo acíclico direcionado; e

- Para cada variável A com pais $B_1, \ldots, B_n$, uma tabela de probabilidade condicional $P(A|B_1, \ldots, B_n)$ é associada.

A definição de Redes Bayesianas não se refere a causalidade e não tem a necessidade de ter ligações que representem um impacto causal. Quando constrói-se um modelo de Redes Bayesianas, não é necessário insistir em ter as ligações que levam a uma direção causal. No entanto, é necessário verificar as propriedades *d-separation* do modelo e assegurar que elas correspondem a percepção das propriedades independentes condicionais do mundo. Se A e B são d-separados dado um evidência e, o cálculo da probabilidade por Redes Bayesianas deve ser $P(A|e) = P(A|B, e)$ (JENSEN e NIELSEN, 2007).

Pearl (1997) define Redes Bayesianas como um grafo acíclico direcionado, em que os nós representam variáveis e as arestas representam a existência de uma influência causal direta entre variáveis associadas e a força destas influências são expressadas através de propriedades condicionais. Russel e Norvig (2004) complementam afirmando que cada nó é representado por uma probabilidade quantitativa e também apresentam a seguinte especificação:

- Um conjunto de variáveis aleatórias constitui os nós da rede. As variáveis podem ser discretas ou contínuas.
- Um conjunto de vínculos orientados ou arrestas conecta pares de nós. Se houver uma arresta de nó X até o nó Y, X será denominado pai de Y.
- Cada nó X_i tem uma distribuição de probabilidade condicional $P(X_i|Pais(X_i))$ que quantifica o efeito dos pais sobre o nó.
- O grafo não tem nenhum ciclo orientado (e consequentemente é um grafo acíclico orientado).

Para Pearl (1992), Redes Bayesianas proporcionam a sistemas baseados em conhecimento, meios gráficos para representação e manipulação de conhecimentos probabilísticos. Suas propriedades e capacidades básicas são elencadas da seguinte forma (PEARL, 1992):

1. Métodos gráficos são criados facilmente para manter a consistência e completude nas bases de conhecimento probabilístico. Estes também definem os procedimentos modulares para a aquisição de conhecimento que reduz significativamente as análises necessárias;
2. Independências podem ser tratadas de forma explícita. Estas podem ser articuladas por um especialista, graficamente codificadas, lidas fora da rede e raciocinadas e ainda sempre permanecerem robustas para a imprecisão numérica.

3. Oportunidades descobertas para representação gráfica para uma computação eficiente. A atualização distribuída é possível em uma estrutura de conhecimento rica o suficiente para exibir interações intercausais. Quando prorrogados por agrupamento ou condicionamento, os algoritmos de propagação em árvore são capazes de atualizar redes de topologia arbitrária.

4. A combinação de inferências preditivas e abdutivas resolvem muitos problemas encontrados na primeira geração de sistemas especialistas e torna redes de crenças (bayesinas) um modelo viável para funções cognitivas, necessitando inferências *top-down* e *bottom-up*.

Redes Bayesianas são consideradas por Pearl (1992) uma ferramenta de grande versatilidade e poder e também o maior esquema de representação comum de conhecimento probabilístico, sendo utilizadas para: diagnósticos médicos, compreender históricos, interpretar imagens, filtragem, visualização e predição, facilitar planejamento de ambientes de incerteza e estudar causas, não monotonicidade, atividade, alteração e atenção. Esta também pode ser vista como um instrumento de inferência para deduzir novas relações independentes a partir das utilizadas para construir a rede.

A partir do problema utilizado como exemplo ao longo do capítulo, pode-se criar uma Rede Bayesiana, onde o estado Cb representa "combustível?", MC o "medidor de combustível;", Lc "o carro liga?"e VL "as velas estão limpas?"(JENSEN e NIELSEN, 2007).

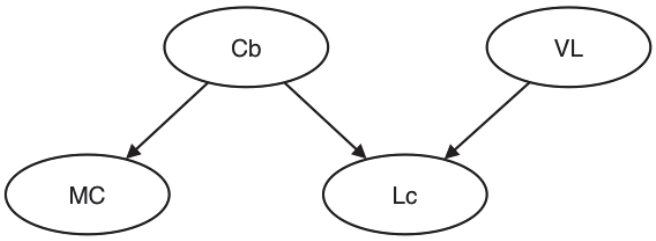


Figura 3.4: Rede causal para o problema de ligar o carro.

Considerando $P(Cb) = (0,98; 0,02)$ e P(VL) = (0,96; 0,04), obtém-se as tabelas de probabilidades condicionais apresentadas na tabelas 3.1.

É possível perceber na Tabela 3.1a reflete o fato de que o medidor talvez não esteja funcionando bem e a Tabela 3.1b deixa espaço para outras causas, no caso, "combustível?" e "as velas estão limpas?", representado por P(Cl = não | Cb = sim, VL = sim) > 0 .

Toda entrada na distribuição de probabilidade conjunta pode ser calculada a partir das informações armazenadas na rede, tendo como entrada genérica a probabilidade de uma

	Cb=sim	Cb=não
MC=Cheio	0,39	0,001
MC=1/2	0,60	0,001
MC=vazio	0,01	0,998

(a) $P(CM|Cb)$

	Cb=sim	Cb=não
VL=sim	(0,99; 0,01)	(0,1)
VL=não	(0,01; 0,99	(0,1)

(b) $P(Cl|Cb, VL)$

Tabela 3.1: Tabelas de probabilidade condicional para Rede Bayesiana do problema ao ligar o carro.

conjunção de atribuições específicas de cada variável, tal como $P(X = x_1 \wedge \ldots \wedge X_n = x_n)$, ou de uma forma resumida $P(x_1, \ldots, x_n)$, o valor desta entrada é dada pela formula (RUSSEL e NORVIG, 2004):

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|pais(X_i)) \tag{3.1}$$

onde os $pais(X_i)$ denotam os valores específicos das variáveis em $Pais(X_i)$.

Certos relacionamentos de independência condicional podem ser utilizados na construção da topologia da rede. O primeiro passo para isso é rescrever a distribuição conjunta em termos de uma probabilidade condicional através da regra do produto (RUSSEL e NORVIG, 2004):

$$P(x_1, \ldots, x_n) = P(x_n|x_{n-1}, \ldots, x_1)P(x_{n-1}, \ldots, x_1) \tag{3.2}$$

Após esta etapa, é repetido o processo, reduzindo cada probabilidade conjuntiva a uma probabilidade condicional e uma conjunção menor, levando a um grande produto:

$$\begin{aligned} P(x_1, \ldots, x_n) &= P(x_n|x_{n-1}, \ldots, x_1)P(x_{n-2}, \ldots, x_1) \ldots P(x_2|x_1)P(x_1) \\ &= \prod_{i=1}^{n} P(x_i|x_{i-1}, \ldots, x_1) \end{aligned} \tag{3.3}$$

Comparado com a Equação 3.1, é possível visualizar que a especificação da distribuição conjunta é equivalente à asserção geral de que, para toda variável X_i na rede, tem-se:

$$P(X_1|X_{i-1}, \ldots, X_1) = P(X_i|Pais(X_i)) \tag{3.4}$$

desde que $Pais(X_i) \subseteq \{X_{i-1}, \ldots, X_1\}$.

A Equação 3.4 demostra que a Rede Bayesiana é uma representação correta do domínio somente se cada nó é condicionalmente independente de seus predecessores na ordenação dos nós, dados seus pais. Por sua vez, para construir uma Rede Bayesiana com a sua estrutura correta para o domínio, é necessário selecionar pais para cada nó de tal forma que essa propriedade seja mantida. Sendo assim, os pais de X_i devem conter todos os nós em $X_1, \ldots, X_{i-1}$ que influenciam diretamente em X_i (RUSSEL e NORVIG, 2004).

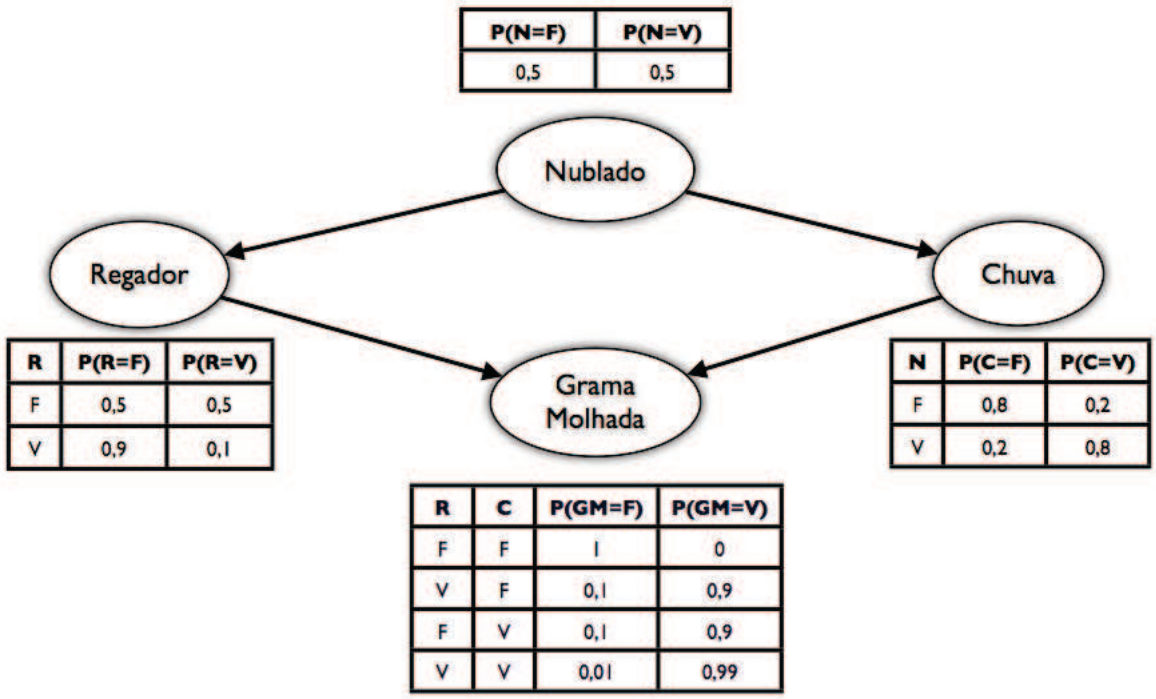


Figura 3.5: Exemplo de uma Rede Bayesiana com as Tabelas de Probabilidade Condicional.

A Figura 3.5 apresenta uma representação gráfica de uma Rede Bayesiana que possui quatro nós, Nublado, Regador, Chuva e Grama Molhada. O Fato do tempo estar ou não nublado, que é o que representa o nó Nublado, tem influência direta sobre o regador estar ligado (nó Regador) e estar chuvendo (nó Chuva). Por sua vez, estes dois nós tem influência direta do o nó Grama Molhada. É possível perceber que cada nó da Rede Bayesiana possui uma Tabela de Probabilidade Condicional composta por todas as possibilidades de resultado possível para o nó, dado evidências ou não de seus pais (nós que o influenciam).

3.3 INFERÊNCIA EM REDES BAYESIANAS

Sistemas de inferência probabilíbilistico tem como objetivo calcular a distribuição de probabilidade posterior para um conjunto de variáveis de consulta, dado um evento

observado. Geralmente, uma consulta busca a distribuição posterior $P(X|e)$ (RUSSEL e NORVIG, 2004).

Tendo como exemplo uma Rede Bayesiana que represente o problema do carro funcionar pela manhã, pode-se ter como exemplo de inferência:

$$P(Lc|Mc = Vazio, Cb = False) = < 0.998, 0.02 > \tag{3.5}$$

Embora existam mais meios de realizar inferências, com as apresentadas em Link e Barker (2010), Darwiche (2009), este trabalho aborda apenas inferência por enumeração, que é abordada de forma direta e objetiva por Russel e Norvig (2004), sendo este, o algoritmo utilizado na implementação do presente trabalho.

O algoritmo de inferência por enumeração é uma adaptação do *ask-enumeration-join*, que realiza inferência por enumeração a partir da distribuição conjunta total. Este algoritmo é definido pela equação:

$$P(X|e) = \alpha P(X, e) = \alpha \sum_{y} P(X, e, y) \tag{3.6}$$

e implementado de acordo com os Algoritmos 3.1 e 3.2.

Algorithm 3.1: ASK-ENUMERATION-JOIN

input : variável de consulta X,
valores observados e para variável E,
uma distribuição conjunta P sobre variáveis $\{ X \} \cup E \cup \mathbf{Y}$
/* $\mathbf{Y}$ = variáveis ocultas */

output: uma distribuição sobre X normalizada

```
1 Q(X) ← uma distribuição sobre X, inicialmente vazia
2 foreach valor x_i de X do
3 |  Q(x_i) = ENUMERATE-JOIN(VARS[P],e)
4 endfor
5 return NORMALIZE(Q(X))
```

Sendo que uma Rede Bayesiana fornece uma representação completa da distribuição conjunta total, a equação

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|pais(X_i)) \tag{3.7}$$

mostra que os termos $P(x, e, y)$ na distribuição conjunta podem ser escritos com os produtos de probabilidades condicionais da rede. Desta forma, uma consulta pode ser respondida com o uso de uma Rede Bayesiana, calculando-se as somas dos produtos das

Algorithm 3.2: ENUMERATE-JOIN

input : x,
evidências e,
variáveis $vars$,
valores $values$,
distribuição conjunta P
output: um número real

if *vars está vazio* **then**
  $P(x, e, values)$
end
$Y \leftarrow PRIMEIRO(vars)$
return $ENUMERATE - JOIN(x, e, RESTO(vars), [y|values], P)$

probabilidades condicionais (RUSSEL e NORVIG, 2004).

Para realizar inferência por enumeração em uma Rede Bayesiana, basta adaptar o algoritmo *ask-enumetation-join* para receber uma Rede Bayesiana ao invés de uma distribuição conjunta total e pesquisar entradas conjuntas multiplicando as entradas da Tabela de Probabilidade Condicional (TPC) correspondentes a partir da Rede Bayesiana. Esta adaptação é apresentada nos Algoritmos 3.3 e 3.4.

Algorithm 3.3: ASK-ENUMERATION

input : variável de consulta X,
valores observados e para variável E,
uma rede bayesiana rb com variáveis $\{ X \} \cup E \cup \mathbf{Y}$
/* $\mathbf{Y}$ = variáveis ocultas */
output: uma distribuição sobre X normalizada

$Q(X) \leftarrow$ uma distribuição sobre X, inicialmente vazia
foreach *valor* x_i *de* X **do**
  estender e com valor x_i para X
  $Q(x_i)$ = ENUMERATE-ALL(VARS[rb],e)
endfor
return NORMALIZE($Q(X)$)

Um fator que deve ser levado em consideração é que inferência em Redes Bayesianas possuem complexidade NP-Difícil, ou seja, mais difícil que problemas NP-Completo, (RUSSEL e NORVIG, 2004). Considerando uma Rede Bayesiana com cinco nós, em uma consulta com uma evidência é necessário somar quatro termos que são obtidos através da multiplicação de cinco números. Sendo assim, a complexidade do algoritmo para uma rede de n nós booleanos é $O(n2^n)$, o que torna a sua execução crítica de acordo com o número de nós da rede.

Existem formas para minimizar este problema de desempenho, como por exemplo, a utilização de um algoritmo de eliminação de variáveis ou então de agrupamentos de nós,

Algorithm 3.4: ENUMERATE-ALL

input : lista com nós da rede bayesiana $vars$
evidências e
output: um número real

if *vars está vazio* **then**
  return 1.0
end
$Y \leftarrow PRIMEIRO(vars)$
if Y *tem valor* y *em* e **then**
  return $P(y|pais(Y))ENUMERATE - ALL(RESTO(vars), e)$
else
  return $\sum_y P(y|pais(Y))ENUMERATE - ALL(RESTO(vars), e_y)$
end

como apresentado por Russel e Norvig (2004). Como o presente trabalho tem foco na geração automática de uma Rede Bayesiana, estes tipos de algoritmo não são abordados neste trabalho.

4 SISTEMAS MULTIAGENTES

No Capítulo 2, foi comentado sobre a utilização de Redes Bayesianas e Sistemas Multiagentes para beneficiar algumas tarefas do processo de Gestão de Riscos. Neste capítulo, as definições de Agentes e Sistemas Multiagentes são apresentadas, assim como sua relação com Redes Bayesianas e a utilização de Sistemas Multiagentes com Redes Bayesianas para beneficiar o processo de Gestão de Riscos.

Agentes, em Sistemas Multiagentes, surgem como um potencial tecnológico para auxiliar na complexidade e variedade de cenários atuais de Tecnologia da Informação. Na indústria, existem experiências de agentes sendo utilizados no processo de produção, *Web Services*, negócios baseados em *Web*, entre outros. No meio acadêmico, estudos apontam para a possibilidade de explorar Agentes e Sistemas Multiagentes como uma tecnologia para uma variedade de futuros cenários, como computação pervasiva, computação em grade, *Web* semântica etc (BERGENTI et al., 2004).

Existe um entendimento geral que Sistema Multiagente é mais que uma tecnologia efetiva, representando na verdade, um novo paradigma de desenvolvimento de *software*. Computação baseada em agentes compreende o projeto e o desenvolvimento de aplicações em termos de entidades de *softwares* autônomos, ou seja, agentes. Este agentes estão situados em um ambiente e podem alcançar, de forma flexível, seus objetivos por meio de interação com outros agentes através de linguagens e protocolos de alto nível (BERGENTI et al., 2004).

A definição de agente será apresentada na Seção 4.1, possibilitando, então, a introdução a sistema multiagente, que será abordada na Seção 4.2.

4.1 AGENTE

Autores como Weiss (1999), Wooldridge (2001), Bergenti et al. (2004) e Rezende (2005) afirmam não existir uma definição amplamente aceita para agentes. No entanto, existe um consenso geral que uma característica essencial em um agente é autonomia. Relacionado ao meio computacional, Wooldridge (2001) define um agente como um sistema que está situado em um ambiente e que é capaz de efetuar ações autônomas neste ambiente, a fim de satisfazer seus objetivos de projeto.

De uma forma mais ampla, Russel e Norvig (2004) definem um agente como qualquer coisa que é capaz de perceber seu ambiente por meio de sensores e de agir sobre este por meio de atuadores. Ferber e Gasser (apud REZENDE, 2005) corroboram com esta definição e complementam que são movidos por um conjunto de inclinações, possuem

recursos próprios e podem, eventualmente, se reproduzir.

Para representar a definição de um agente, Russel e Norvig (2004) apresentam exemplos com relação entre humanos, robôs e *softwares*. Um humano possui olhos, ouvidos e outros órgãos como sensores, e possui mãos, pernas, boca e outras partes do corpo que servem como atuadores. Um robô poderia ter uma câmera e detectores de faixa infravermelho funcionando como sensores e vários motores como atuadores. Um agente de *software* recebe seqüências de teclas digitadas, conteúdo de arquivos e pacotes de rede como entradas sensórias e atua sobre o ambiente exibindo algo na tela, gravando arquivos e enviando pacotes de rede.

As definições apresentadas podem ser representadas com a Figura 4.1, que mostra um agente, composto por sensores e atuadores. Os sensores são responsáveis pelas percepções do ambiente e os atuadores por realizar ações neste ambiente.

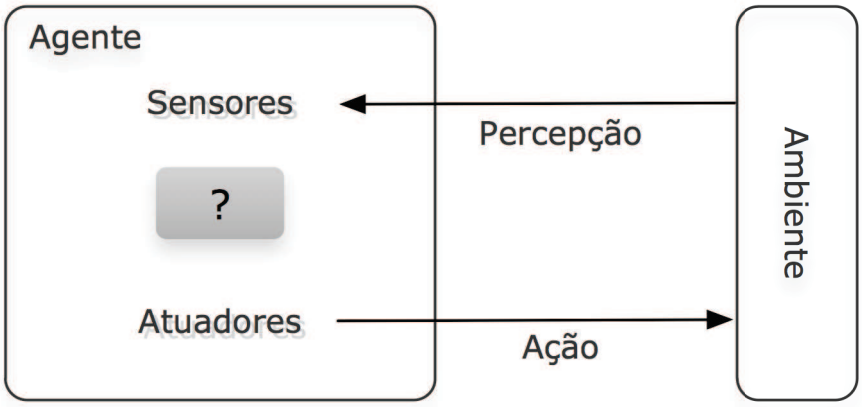


Figura 4.1: Um agente em um ambiente. (RUSSEL e NORVIG, 2004)

Na maioria dos domínios de complexidade razoável, um agente não terá total controle sobre o ambiente e terá no melhor dos casos, um controle parcial. Do ponto de vista de um agente, isso significa que a mesma ação desempenhada duas vezes em circunstâncias aparentemente idênticas, podem, ter efeitos completamente diferentes e deixar de ter o efeito desejado. Desta forma, os agentes devem estar preparados para uma possível falha (WOOLDRIDGE, 2001).

Para evitar falhas, normalmente, os agentes possuem um repertório de ações disponíveis, que representam a capacidade *efetivadora* do agente, que possibilita modificar o ambiente. A escolha de qual ação desempenhar para melhor satisfazer os objetivos de projeto é o problema chave de um agente (WOOLDRIDGE, 2001).

Com base nesta definição, um termostato pode ser considerado um agente. Este agente

possui um sensor que detecta a temperatura de uma sala e gera duas possíveis saídas: (i) a temperatura está baixa e (ii) a temperatura está adequada. As ações disponíveis para este agente são: (i) ligar aquecimento e (ii) desligar aquecimento. Sempre que o sensor identificar que o ambiente está com a temperatura baixa, a ação que ele efetuará é "ligar aquecimento' e quando identificar que a temperatura está adequada, a ação que ele realizará é "desligar aquecimento".

O termostato pode ser visto com um agente, no entanto, não pode ser considerado um agente inteligente. Para um agente ser considerado inteligente, este deve possuir características como: reatividade, pró-atividade, sociabilidade.

Reatividade: Agentes inteligentes são capazes de perceber seu ambiente e responder em tempo hábil a mudanças que ocorram nele, a fim de satisfazer seus objetivos de projeto.

Pró-atividade: Agentes inteligentes são capazes de expor o seu comportamento voltado a sua meta, de forma a satisfazer seus objetivos de projeto.

Sociabilidade: Agentes inteligentes tem a capacidade de interagir com outros agentes de forma a satisfazer seus objetivos de projeto.

4.2 DEFINIÇÃO DE SISTEMA MULTIAGENTE

Existem situações em que agentes podem operar com sucesso isoladamente, no entanto, com o crescimento de interconeções e redes de computadores, este cenário é raro e os agentes comunicam-se com outros agentes, compreendendo e trocando mensagens entre si.

De fato, soluções centralizadas são mais eficientes, mas algumas vezes, sistemas computacionais distribuídos são mais fáceis de compreender e também de se desenvolver, especialmente quando o problema que está sendo resolvido também está distribuído. Também existem casos em que uma abordagem centralizada não é possível devido aos sistemas e dados serem independentes de organização, ou seja, encontram-se em organizações diferentes (WEISS, 1999).

Os cenários em que a informação é distribuída podem ser classificados em: (i) elas são geograficamente distribuídas, (ii) possuem muitos componentes, (iii) possuem um grande conteúdo e (iv) possuem um grande escopo.

As quatro grandes técnicas para lidar com o tamanho e complexidade dos sistemas de informação destas organizações são modularidade, distribuição, abstração e inteligência, caracterizando a Inteligência Artificial Distribuída (IDA) (WEISS, 1999).

Sistemas multiagentes são o melhor meio de categorizar e projetar sistemas computacionais distribuídos, em que o processamento da informação é ubíqua, tendo como principais características (WEISS, 1999):

1. Ambientes multiagentes possibilitam uma estrutura específica de comunicação e protocolos de comunicação;
2. Ambientes multiagentes são, tipicamente, distribuídos e não possuem um projeto centralizado;
3. Ambientes multiagentes possuem agentes que são autônomos e distribuídos, e talvez, auto-interessados ou cooperativos.

Um sistema multiagente possui agentes que interagem um com outros através de comunicação, formando uma relação organizacional. Estes agentes são capazes de agir sobre o ambiente, em que diferentes agentes possuem uma esfera de influência sobre este mesmo ambiente. Estas esferas podem coincidir em alguns casos, podendo aumentar a dependência do relacionamento entre agentes (WOOLDRIDGE, 2001).

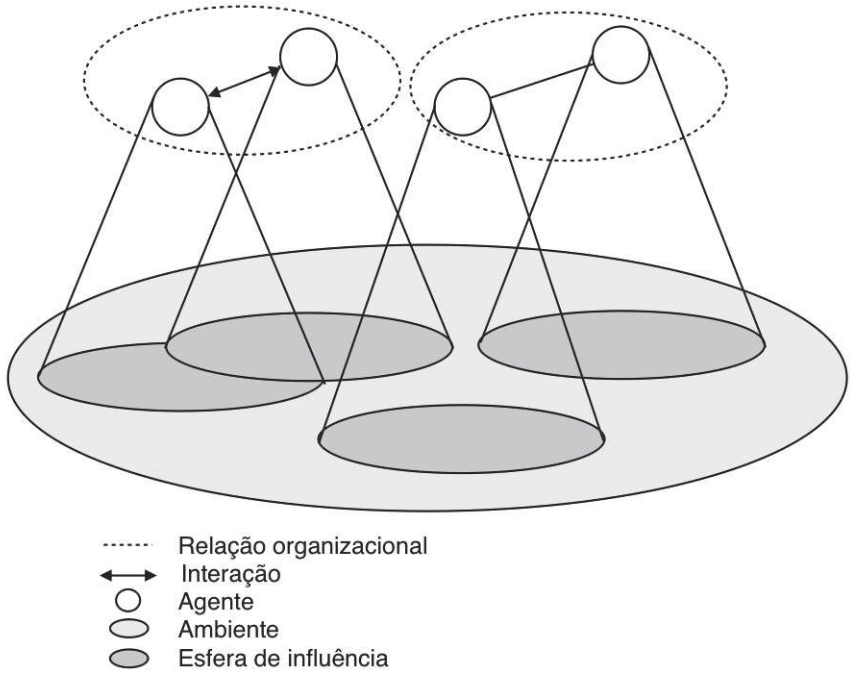


Figura 4.2: Estrutura típica de um sistema multiagente (WOOLDRIDGE, 2001).

A Figura 4.2 apresenta de forma mais clara a definição de Wooldridge (2001). A Figura 4.2 apresenta na parte superior, a comunicação entre agentes, formando então uma relação organizacional. Na base da Figura 4.2 pode ser visto o ambiente em que o

sistema multiagentes atua. Na centro da Figura 4.2, podem-se visualizar as esferas de influência que o agente tem sobre o ambiente.

Sistemas multiagentes são relevantes para solucionar problemas complexos e distribuídos e as aplicações projetadas são cada vez mais complexas. Essa complexidade tem origem e dificuldades como (BERGENTI et al., 2004):

- Identificar as tarefas que o sistema global tem que resolver;
- Identificar qual entidade do sistema é um agente;
- Definir interação entre agentes;
- Especificar protocolos adequados e/ou complexos;
- Definir interações entre o sistema e seu ambientes; e
- Definir o comportamento relevante do agente.

Novas ferramentas e modelos ajudam engenheiros a trabalhar com estas novas noções. Além disso, o software contém uma enorme quantidade de linhas de código e sua distribuição aumenta a complexidade para os projetistas que são obrigados a ter em conta novos problemas, como mobilidade e segurança. Metodologias ajudam os projetistas a lidar com estes problemas e gerenciar a complexidade envolvida (BERGENTI et al., 2004).

Uma metodologia baseada em agente ou multiagente consiste em um processo, uma linguagem de modelagem ou notação e uma ferramenta para auxiliar no processo e na notação, ajudando o projetista. Atualmente, as principais fases deste processo são similares a aquelas usadas em metodologias orientadas a objetos, ou seja, requisitos, análise (ou especificação), projeto, desenvolvimento (ou implementação) e publicação do sistema (BERGENTI et al., 2004).

Algumas metodologias já foram propostas para o desenvolvimento de sistemas multiagente e são abordades em Bergenti et al. (2004), como: Gaia, Tropos, MaSE, ADELFE, MESSAGE, SADDE e Prometheus. A metodologia que será abordada e também utilizada neste trabalho é a *Prometheus*, que é apresentada na Seção 4.3.

4.3 METODOLOGIA *PROMETHEUS*

Prometheus é uma metodologia que tem como objetivo proporcionar a profissionais e a estudantes de graduação um fácil desenvolvimento de sistemas multiagente sem a necessidade de um conhecimento prévio do assunto (BERGENTI et al., 2004).

Mesmo com este objetivo, a metodologia *Prometheus* é uma abordagem consistente para construção de sistemas multiagentes. Esta é composta por três fases distintas: Especificação do Sistema, Projeto Arquitetural e Projeto Detalhado.

Na primeira fase, Especificação do Sistema, são analisados os fatores de entrada (*inputs*), saída (*outputs*), e repositórios de dados (*data sources*), sejam compartilhados ou não. Posteriormente, na fase de *Design* Arquitetural, usam-se os *outputs* para determinar que agentes irão existir no sistema, bem como as suas interações. Tendo isto concluído, a fase de Design Detalhado pode ser iniciada, em que é avaliado internamente como cada agente irá concluir as suas tarefas (PADGHAM e WINIKOFF, 2004).

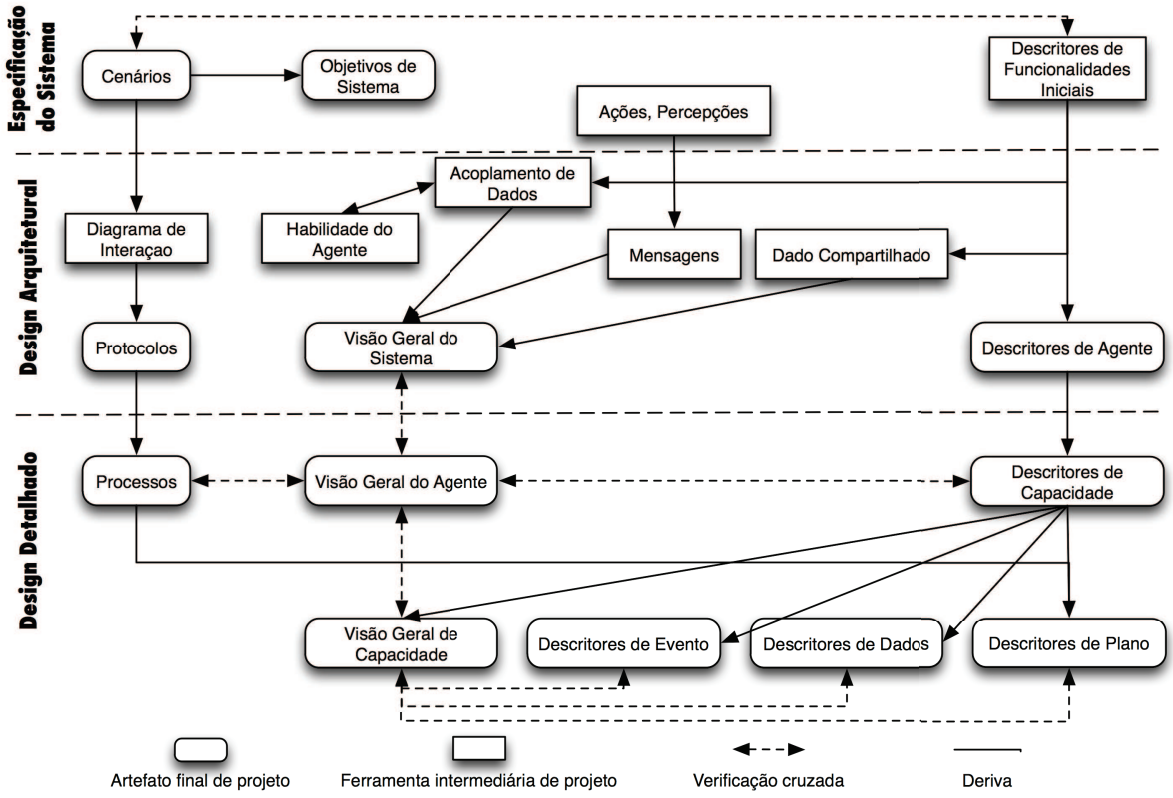


Figura 4.3: Diagrama da metodologia Prometheus

A Figura 4.3 apresenta os principais artefatos de projeto que surgem em cada uma das fases existentes na metodologia, alguns itens intermediários e a relação entre os itens. Esta figura é dividida horizontalmente e verticalmente. Horizontalmente, esta possui três regiões que representam as três fases existentes na metodologia. Verticalmente, esta também possui três regiões, sendo que a região mais à esquerda lida com os comportamentos dinâmicos do sistema, a região central lida com as visões gerais do sistema e a mais à direita, que detalha cada entidade do sistema (BERGENTI et al., 2004).

O desenvolvedor, ao utilizar esta metodologia, deve considerar alguns fatores importantes na modelagem do seu respectivo SMA (PADGHAM e WINIKOFF, 2004):

- Informações provenientes do ambiente do sistema são consideradas percepções do agente;
- Mecanismos que podem afetar este ambiente são considerados ações que o agente pode tomar;
- Um evento não é considerado igual a uma percepção, trata-se de um conceito de maior importância para um agente;
- Nem tudo que ocorre no ambiente faz o agente alterar seus planos para alcançar um objetivo;
- Percepções necessitam de processamento por parte do agente para se tornarem um evento; e
- Tal processamento deve ocorrer no agente para ser considerada uma metodologia *Prometheus*.

4.3.1 Especificação do Sistema

Esta fase consiste em três atividades: (i) determinar a interface do sistema para o ambiente, (ii) determinar os objetivos e funcionalidades do sistema e (iii) capturar os cenários que refletem a utilização do sistema.

A *Prometheus* utiliza a terminologia utilizada por Russel e Norvig (2004), em que o o agente percebe alterações no ambiente através de percepções e age sobre ele por meio de ações. Os dados em forma bruta das percepções podem precisar de um processamento para se obter informações dos eventos significantes para um agente.

A determinação dos objetivos e funcionalidades do sistema são obtidos a partir dos seguintes passos (BERGENTI et al., 2004):

- Identificar e refinar objetivos do sistema - principal e secundário;
- Agrupar objetivos em funcionalidades;
- Preparar descritores de funcionalidades; e
- Verificar se todos os objetivos estão abrangidos pelo cenário;

Um conjunto de objetivos iniciais é obtido de um requisito inicial. Funcionalidades são limitadas do comportamento do sistema, que descrevem em sentido mais amplo que

o sistema precisa ser capaz de fazer. Descritores de funcionalidades capturam o nome de descrição de cada funcionalidade, assim como os eventos que as ativam, quais objetivos são alcançados, quais ações são executadas, quais percepções são recebidas, quais mensagens são enviadas e recebidas e quais dados são utilizados e produzidos (BERGENTI et al., 2004).

4.3.2 *Design* Arquitetural

Basicamente, são três aspectos que são abordados durante o *Design* Arquitetural, que são (BERGENTI et al., 2004):

1. Decidir quais tipos de agentes serão utilizados no sistema. O tipo de agente é formado agrupando um número de funcionalidades. Os diagramas utilizados para auxiliar nesta análise são: *data coupling diagrams* e *agent acquaintance diagrams*;
2. Criar a estrutura global do sistema (com um diagrama de visão geral do sistema, juntamente com descritores); e
3. Descrever interações entre agentes utilizando o diagrama de interações e protocolos de interação.

4.3.3 *Design* Detalhado

Esta fase lida com a parte interna de cada agente, ao invés do sistema como um todo. É utilizado um modelo hierárquico de modo que cada agente é dividido em capacidades e estas podem ser incluídas em mais de uma agente.

Os passos que compõe o *Design* Detalhado são:

1. Desenvolver a visão geral dos agentes (mostrando as interações entre as capacidades) e os descritores de capacidade.
2. Desenvolver o processo interno dos protocolos de interação de um agente, utilizando uma variante de diagramas de atividade *Unified Modeling Language* UML.
3. Desenvolver o projeto interno de cada capacidade em termos de planos, eventos, crenças e possivelmente sub-capacidades.

5 TRABALHOS RELACIONADOS

Este capítulo apresenta os trabalhos relacionadas com o trabalho proposto. Até o presente momento, não foram encontrados trabalhos com o mesmo objetivo do trabalho proposto. Os artigos que são apresentados neste capítulo são os que contribuíram e também serviram como base para a elaboração deste trabalho, mas em alguns casos não possuem relação entre si, impossibilitando a comparação entre eles.

Segurança da Informação é uma área de pesquisa muito recente e também com muitos temas ainda em abertos. Este é o principal motivo por não existirem trabalhos com o mesmo objetivo que o trabalho proposto.

A utilização de Redes Bayesianas para favorecer a Gestão de Riscos em Segurança da Informação é abordada por Dantu e Kolan (2005) e Dantu et al. (2007). Dantu e Kolan (2005) calculam o nível de riscos dos recursos críticos da organização tendo em vista o aparecimento de novas vulnerabilidades e no comportamento do atacante, utilizando Redes Bayesianas para isto. A metodologia proposta possui cinco etapas: (i) criação do perfil do atacante, (ii) criação de um grafo de ataque, (iii) atribuir atributos de comportamento ao grafo de ataque, (iv) cálculo do risco, e (v) otimização do nível de risco.

Criação do perfil do atacante O perfil de um atacante fornece os recursos prescindíveis associados com o atacante. Esses recursos podem ser: qualquer custo, habilidades na informática e no *hacking*, tenacidade, perseverança (como vingança) e reputação. Diferentes perfis de ataque possuem diferentes valores para atributos comportamentais para o recurso do atacante. Uma espionagem corporativa envolve mais dinheiro que um *script kiddie*[1], que realiza o *hack* apenas por diversão. O funcionário de uma empresa tem mais conhecimento que um *hacker* sobre a topologia da rede. Com base nestas relações, é atribuído um custo para cada perfil de atacante.

Criação de uma grafo de ataque Um grafo é criado com base na topologia da rede, interconexões entre *hosts* e várias vulnerabilidades de uma dado *host*. O grafo de ataque é representado por um grafo de causalidade, em que cada nó representa uma causa e seus nós filhos representam um efeito. Cada nó do grafo representa um evento, e um caminho do nó raiz do até a folha representa um ataque bem sucedido.

Atribuir atributos de comportamento ao grafo de ataque Para o perfil de um determinado invasor, os nós do grafo são rotulados com um conjunto de atributos de comportamento, tais como: conhecimento em informática, habilidades de *hacking*, tenacidade, custo do ataque, técnicas para evitar detecção etc.

[1] Termo utilizado para definir atacantes inexperientes, geralmente jovens.

Cálculo do risco O nível de risco para todos os recursos são calculados com base no conjunto de caminhos, atributos e o tipo do atacante. Nesta etapa, uma estimativa baseada em Redes Bayesianas é utilizada para calcular o valor de risco agregado para cada recurso.

Otimização do nível de risco Em uma rede típica, a correção de uma vulnerabilidade pode impactar outros elementos da rede. Por exemplo, depois de corrigir algumas falhas e mudar a configuração da rede (mover o firewall para outro local na topologia, alterar as regras de firewall ou implementar um sistemas de detecção de intrusão), a etapa (iii) desta metodologia necessita ser executada repetidamente para se obter um valor de risco ótimo. Este valor estimado de risco ajudaria em processos como o gerenciamento de *patches* e teste de invasão etc.

Complementando esta abordagem de Dantu e Kolan (2005), o trabalho de Dantu et al. (2007) apresenta uma classificação de atributos e comportamentos em Gestão de Riscos utilizando Redes Baysianas. Neste trabalho, os princípios apresentados em Dantu e Kolan (2005) são mantidos, tendo como foco o perfil de comportamento.

Fenz e Hudec (2009) propoem um método para geração de Redes Bayesiana baseado em uma ontologia de segurança da informação. O método desenvolvido permite, com base na ontologias, gerar de forma semi-automática e alterar a Rede Bayesiana.

O método proposto por Fenz e Hudec (2009) possui quatro fases, que são: Componentes, Relações, Axiomas e Instancias.

Componentes → Nós Os conceitos da ontologia que são considerados importantes para o problema, são selecionados para formar a Rede Bayesiana.

Relação → *Links* As relações da ontologia que começam e terminam entre os conceitos selecionados são utilizados para estabelecer as ligações entre os nós da Rede Bayesiana.

Axiomas → Escala e peso dos nós A escala e o peso dos axiomas relevantes são utilizados para determinar os estados e pesos potenciais dos nós da Rede Bayesiana.

Instâncias → Resultados Instâncias de conceitos representados pelos nós folhas da Rede Bayesiana são utilizados para obter e também inserir resultados concretos na rede.

Este método permite a criação semi-automática de Redes Bayesianas a partir de uma ontologia já existente, reduz a complexidade de modelagem de Redes Bayesianas utilizando conceitos de alto nível e relações relevantes para a integração de sub-conceitos para a Rede

Bayesiana e prevê, através do uso de ontologias, a possibilidade de fácil manutenção do conhecimento subjacente das Redes Bayesianas.

As limitações do método proposto, apresentadas pelos próprios autores são: (i) as funções para calcular tabelas de probabilidade condicional não são fornecidos pela ontologia e necessitam ser modelados externamente, e (ii) a intervenção humana ainda é necessária se a ontologia fornece um modelo de conhecimento que não está ajustado ao domínio de interesse.

A pesquisa sobre o perfil do atacante realizada por Dantu et al. (2007) é a principal contribuição para o trabalho proposto. Os comportamentos classificados pelos autores podem ser utilizados para complementar o modelo proposto.

As etapas sugeridas por Fenz e Hudec (2009) serviram como base para o modelo proposto. No entanto, a forma pelo qual os dados são obtidos são diferentes. Dantu et al. (2007) buscam informações em uma ontologia para montar sua Rede Bayesina. No trabalho proposto, as informações são obtidas a partir de agentes que localizam as informações em ambientes específicos, como a base de dados de configuração de ativos, base de dados de incidentes reportados na organização e em base de dados de vulnerabilidades reportadas no mundo. Desta forma, as limitações do método de Dantu et al. (2007) podem ser minimizadas.

A ferramenta proposta por (EKELHART et al., 2009) utiliza a geração de Redes Bayesianas de Fenz e Hudec (2009). Esta ferramenta, que recebe o nome de *Automated Risk and Utility Management* (AURUM) é destinada a minimizar a interação entre gestores de segurança da informação e sistemas.

Para alcançar este objetivo, o AURUM utiliza uma ontologia de segurança da informação, uma ferramenta de inventários, redes bayesianas e o método para o cálculo do risco. A arquitetura do AURUM é apresentada na Figura 5.1.

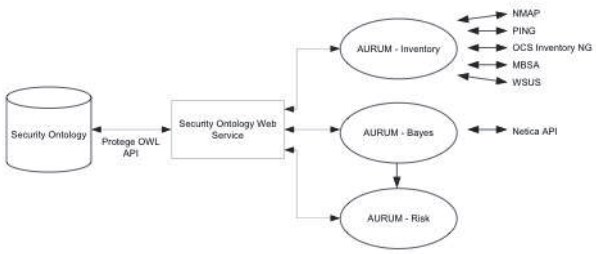


Figura 5.1: Arquitetura do ferramenta AURUM

O AURUM tem como principal característica a descoberta automática de ativos no ambiente da organização através de ferramentas como NMAP, PING, OCS Inventory NG,

etc, que são ferramentas que coletam informações sobre os ativos computacionais através da rede. Outra característica importante é a criação de uma rede bayesiana a partir de uma ontologia do domínio de segurança da informação.

Este trabalho é o mais próximo do trabalho proposto, visto que possui um objetivo similar. No entanto, o trabalho não considera informações pontuais sobre incidentes que aconteceram na organização e também as vulnerabilidades existentes para os ativos que compõem o negócio da organização.

6 TRABALHO PROPOSTO

Este capítulo apresenta o trabalho proposto, assim como sua estrutura, arquitetura, algoritmos e métodos utilizados. Como o método proposto é baseado em sistemas multiagentes, o capítulo inicia com a descrição dos cenários existentes no sistema, seguido de todos os itens que compõem o sistema multiagente. Detalhes específicos do método de geração da Rede Bayesiana são apresentados na Seção 6.4. Detalhes sobre o desenvolvimento do protótipo são apresentados na Seção 6.6. A simulação para teste do protótipo é descrita na Seção 6.7.1 e os resultados obtidos com o protótipo são apresentados na Seção 6.7.2.

O sistema multiagente proposto no presente trabalho é composto pelos seguintes cenários; o *identify asset*, o *identify incedent* e o *identify vulnerabilty*. Nestes três cenários estão presentes quatro agentes de *software* e quatro bancos de dados.

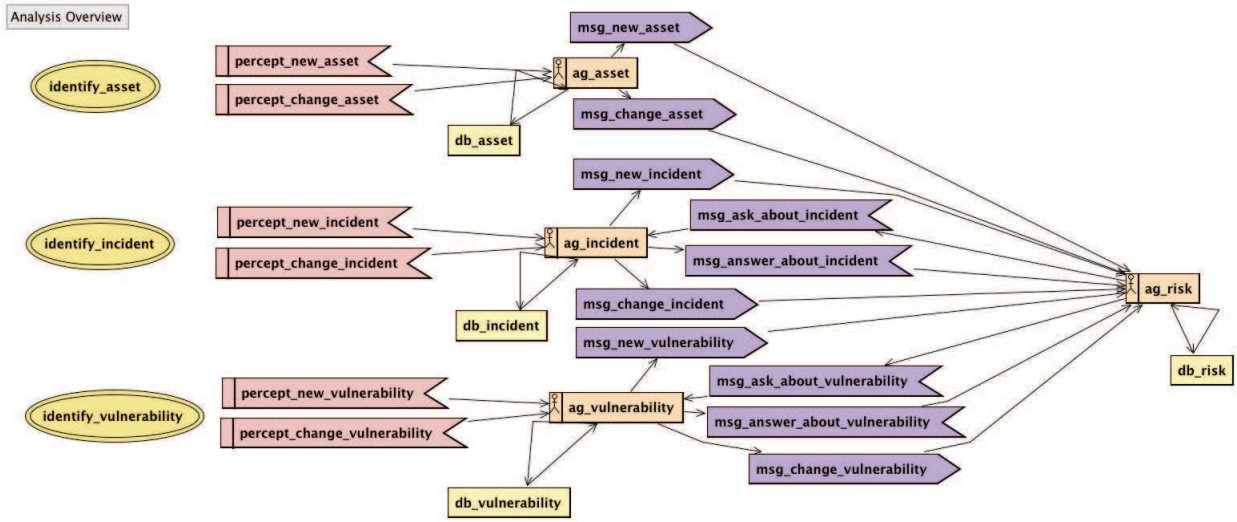


Figura 6.1: Visão geral do sistema multiagente proposto

A Figura 6.1 apresenta uma visão geral do sistema multiagente que está sendo proposto para atender os objetivos do trabalho. O diagrama ilustrado na Figura 6.1 apresenta os três cenários que inicia o sistema multiagente, que são: (i) *identify asset*, (ii) *identify incident* e (iii) *identify vulnerability*. Cada cenário possui um agente, que são eles: (i) *ag Asset*, (ii) *ag Incident* e (iii) *ag Vulnerability*, respectivamente. Estes três agentes comunicam-se com o agente *ag Risk*, que é o principal agente do sistema.

Os três cenários existentes no sistema estão diretamente relacionados aos os agentes *ag asset*, *ag incident* e *ag vulnerability*. O cenário *identify asset* é disparado a partir da alteração realizada pelos responsáveis pela TI da organização na base de dados de configuração de ativos. O cenário *identify incident* é disparado a partir da alteração

na base de dados de incidentes de segurança, que é administrada pelos responsáveis pela segurança na organização. O cenário *identify vulnerability* é disparado a partir da alteração na base de dados de vulnerabilidades da NVD, que é realizada com a colaboração de organizações de apoio a incidentes de segurança.

Ao longo deste capítulo, maiores detalhes sobre o sistema serão abordados. A Seção 6.1 existentes no sistema proposto. As bases de dados utilizadas no sistema são apresentadas na Seção 6.2. Os agentes que compõem o sistema são abordados na Seção 6.3, assim como as mensagens trocadas por estes agentes. Finalizando o capítulo, a Seção 6.4 apresenta os detalhes sobre a geração da Rede Bayesiana pela agente *ag Risk*.

6.1 CENÁRIOS

Esta seção apresenta os três cenários que compõem o sistema multiagentes proposto e introduz o funcionamento do sistema a partir de cada cenário.

***Identify Asset*:** As percepções que caracterizam este cenário são a identificação de um novo ativo na infraestrutura de tecnologia da informação ou alguma alteração nestes ativos. A identificação de um novo ativo ou a identificação da alteração de um novo ativo são as percepções do agente "ag asset". Quando identificado um novo ativo, o agente "ag asset" armazena a informação no banco de dados "db asset" e, posteriormente, envia para o agente "ag risk" as informações do ativo identificado, que irá executar todos os processos necessários para atualizar a Rede Bayesiana de riscos, incluindo este novo ativo na rede.

***Identify Incident*:** Este cenário tem início com a identificação de novos incidentes envolvendo os ativos de tecnologia da informação da organização ou alterações nos incidentes relacionadas a segurança da informação nos ativos de TI da organização, caracterizando então as duas percepções do agente "ag incident". Quando este agente identifica um novo incidente ou a alteração de um incidente, este informa ao agente "ag risk" as alterações do ambiente identificadas para que este execute todos os processos necessários para atualizar a Rede Bayesiana de riscos. Estes incidentes são tratados como evidências na rede.

***Identify Vulnerability*:** Este é o cenário em que o sistema identifica uma nova vulnerabilidade nos ativos de TI da organização ou alguma alteração nas vulnerabilidades já conhecidas. O agente "ag vulnerability" identifica as alterações no ambiente através das suas duas percepções e comunica ao agente "ag risk" sobre estas alterações para que este realize os processos necessários para montar ou alterar a Rede Bayesiana.

6.2 BASES DE DADOS

Este seção descreve a utilização as base de dados utilizadas no sistema proposto no presente trabalho, assim como as estruturas relacionais que foram utilizadas no desenvolvimento do sistema.

6.2.1 DB Vulnerability

Esta base de dados possui as vulnerabilidades reportadas em todo mundo e são disponibilizadas livremente no endereço eletrônico `http://cve.mitre.org`. Os dados são disponibilizados em um formato *eXtensible Markup Language* - XML. Este documento XML utiliza padrões de nomenclatura, conforme subseção 2.4.1, que possibilita a identificação exata dos ativos que estão sujeitos às vulnerabilidades nela reportadas. As informações contidas nesta base são armazenadas em um banco de dados local para que sejam utilizadas para o cálculo do fator de risco dos ativos de TI.

Futuramente, outras fontes de vulnerabilidades também poderão ser utilizadas, visto que estas bases possuem referências entre elas, o que possibilita que vulnerabilidades não sejam duplicadas.

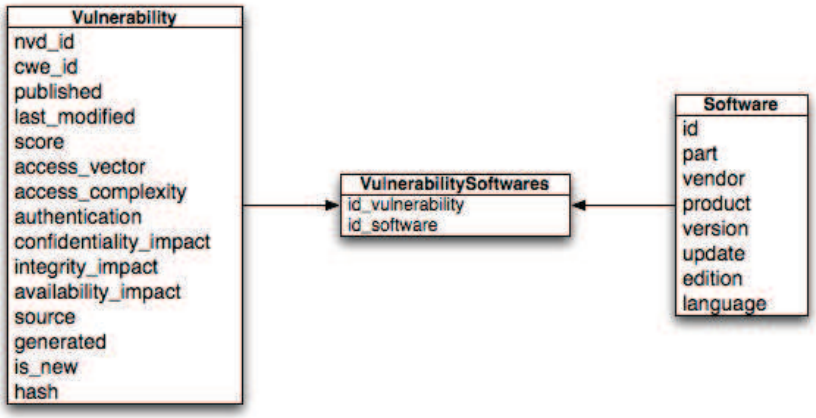


Figura 6.2: Modelo ER da base de dados *DB Vulnerability*

A Figura 6.2 apresenta o modelo relacional utilizado para armazenar as vulnerabilidades importadas da base de dados do NVD. Como pode se perceber, o modelo é composto por três tabelas.

A primeira tabela, chamada *Vulnerability*, é destinada a armazenar os dados relacionados às vulnerabilidades. Esta tabela segue a mesma estrutura da base do NVD, exceto por dois campos, que são: *is_new* e *hash*. O campo *is_new* é utilizado para identificar as vulnerabilidade novas, ou seja, recém adicionadas na base local. Este campo

é alterado para *false* assim que a vulnerabilidade é processada pela primeira vez no sistema. O Campo *hash* é utilizado para armazenar um *hash* dos dados que compõem a vulnerabilidade e é utilizado para identificar qualquer alteração na configuração de uma vulnerabilidade. Quando uma alteração na configuração da vulnerabilidade é identificada, esta alteração é divulgada no sistema e o valor da campo *hash* é atualizado com o *hash* da nova configuração da vulnerabilidade.

A segunda tabela, chamada *VulnerabilitySoftwares* faz a relação das vulnerabilidades com os *softwares* afetados, que são armazenados na terceira tabela, chamada *Softwares*. A tabela *Softwares* segue a mesma estrutura de um nome CPE, apenas separando cada parte de um nome CPE em colunas.

6.2.2 DB Asset

A base de dados de ativos possui todos os ativos pertencentes aos processos de negócio da organização. Estes ativos se relacionam com determinados processos e também possuem uma estrutura que gera uma dependência entre eles.

Este trabalho assume que a organização segue a estrutura proposta por este trabalho para gerenciar seu banco de dados de configuração de ativos.

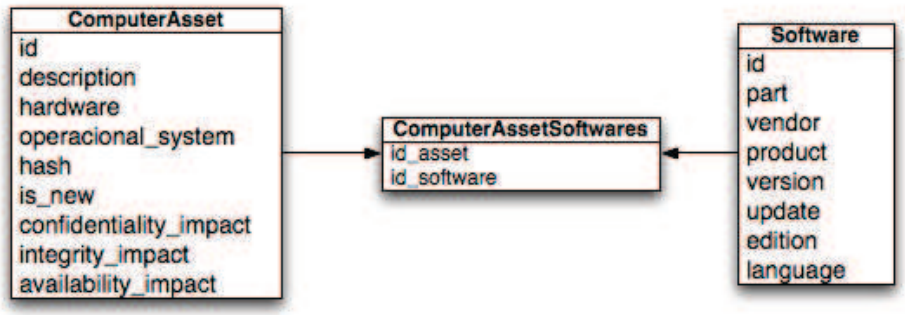


Figura 6.3: Modelo ER da base de dados *DB Asset*

A Figura 6.3 apresenta o modelo relacional proposto para armazenar as informações referentes aos ativos da organização. Como pode-se perceber, o modelo é composto por três tabelas.

A primeira, chamada *ComputerAsset*, que armazena as informações referentes aos ativos computacionais da organização. A estrutura desta tabela é descrita a seguir:

***id*:** coluna utilizada para armazenar o identificador do registro;

***description*:** coluna destinada a armazenar uma descrição a fim de identificar o ativo computacional;

***hardware*:** coluna destinada a armazenar o nome CPE referente ao *hardware* do ativo computacional;

***operational_system*:** coluna destinada a armazenar o nome CPE referente ao sistema operacional do ativo computacional;

***confidenciality_impact*:** coluna destinada a armazenar um valor referente ao impacto na confidencialidade que o ativo computacional possui sobre o negócio da organização;

***integrity_impact*:** coluna destinada a armazenar um valor referente ao impacto na integridade que o ativo computacional possui sobre o negócio da organização;

***availability_impact*:** coluna destinada a armazenar um valor referente ao impacto na disponibilidade que o ativo computacional possui sobre o negócio da organização;

***is_new*:** coluna destinada a identificar os ativos computacionais novos, ou seja, recém adicionado à base local. Este campo é alterado para *false* assim que o ativo computacional é processado pela primeira vez no sistema; e

***hash*:** coluna destinada a armazenar um *hash* dos dados que compõem o ativo computacional, sendo este, utilizado para identificar qualquer alteração na configuração do ativo computacional.

A segunda tabela, camada *ComputerAssetSoftwares*, que relaciona os ativos computacionais aos seus *softwares* (que também são seus ativos), que são armazenados na terceira tabela apresentada, chamada de *softwares*, que segue a mesma estrutura de um nome CPE.

6.2.3 DB Incident

Esta base de dados armazena todos os incidentes de segurança da informação que ocorreram nos ativos de TI da organização. As causas que levaram ao incidente, sempre que identificadas, também são armazenadas nesta base. O mesmo ocorre com os ativos envolvidos.

A estrutura desta base de dados será definida de forma a atender os requisitos do sistema proposto neste trabalho. Assume-se que a organização utilizará esta estrutura para armazenar os incidentes que ocorrem em seus ativos de TI.

A Figura 6.4 apresenta o modelo relacional proposto para armazenar os incidentes de segurança da informação. Como pode-se perceber, o modelo é composto por duas tabelas.

A tabela chamada de *Incident* armazena os dados dos incidentes e sua estrutura é descrita a seguir:

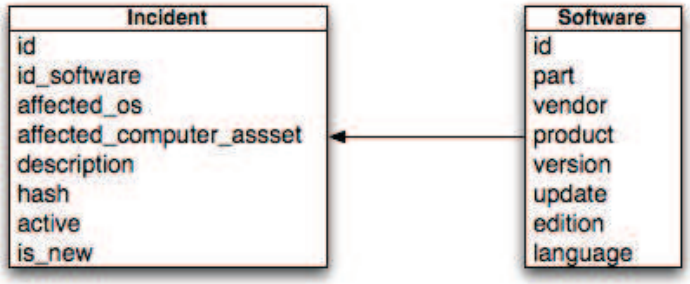


Figura 6.4: Modelo ER da base de dados *DB Incident*

***id*:** coluna utilizada para armazenar o identificador do registro;

***id_software*:** coluna destinada a relacionar o incidente a um software;

***affected_os*:** coluna utilizada para informar se o incidente comprometeu ou não o sistema operacional;

***affected_computer_asset*:** coluna utilizada para informar se o incidente comprometeu ou não o funcionamento do ativo computacional;

***description*:** coluna destinada a armazenar uma descrição a fim de identificar o ativo computacional;

***active*:** coluna destinada a informar se o incidente ainda está ativo ou se alguma solução de contorno foi adotada;

***is_new*:** coluna destinada a identificar novos incidentes, ou seja, recém adicionados à base de dados. Este campo é alterado para *false* assim que o incidente é processado pela primeira vez no sistema; e

***hash*:** coluna destinada a armazenar um *hash* dos dados que compõem o incidente, sendo este, utilizado para identificar qualquer alteração na configuração de um incidente.

6.2.4 DB Risk

É uma base de dados consolidada, onde os fatores de risco dos ativos de TI são armazenados. A partir desta base de dados, os usuários (profissionais de gestão de risco) poderão efetuar consultas e analisar os riscos de segurança existentes na organização.

Por esta ser a base de dados utilizada pelo *Ag Risk*, esta possui um modelo que replica todo o sistema. Esta abordagem é utilizada para melhorar o desempenho do sistema, diminuindo a quantidade de consultas a outros agentes no momento da execução do algoritmo que gera a Rede Bayesiana. As informações desta base de dados são modificadas

de acordo com as informações enviadas pelos outros agentes que compõem o sistema multiagente.

A única tabela que não possui um relação direta com as informações enviadas pelos agentes do sistema é a tabela *RiskHistory*. Esta tabela é utilizada para armazenar o resultado do cálculo do risco dos agentes computacionais, possibilitando assim, a geração de gráficos que demonstram a evolução no tempo do risco relacionado aos ativos computacionais. A geração deste tipo de gráfico está fora do escopo do presente trabalho.

A Tabela *RiskHistory* é composta por três colunas, sendo a coluna *id* destinada a armazenar o identificador do registro na tabela, *id_risk_computer_asset* utilizado para relacionar ao ativo computacional referente ao resgistro, *risk* que armazena o fator de risco do ativo computacional e *date*, que armazena a data e hora que o cálculo foi realizado.

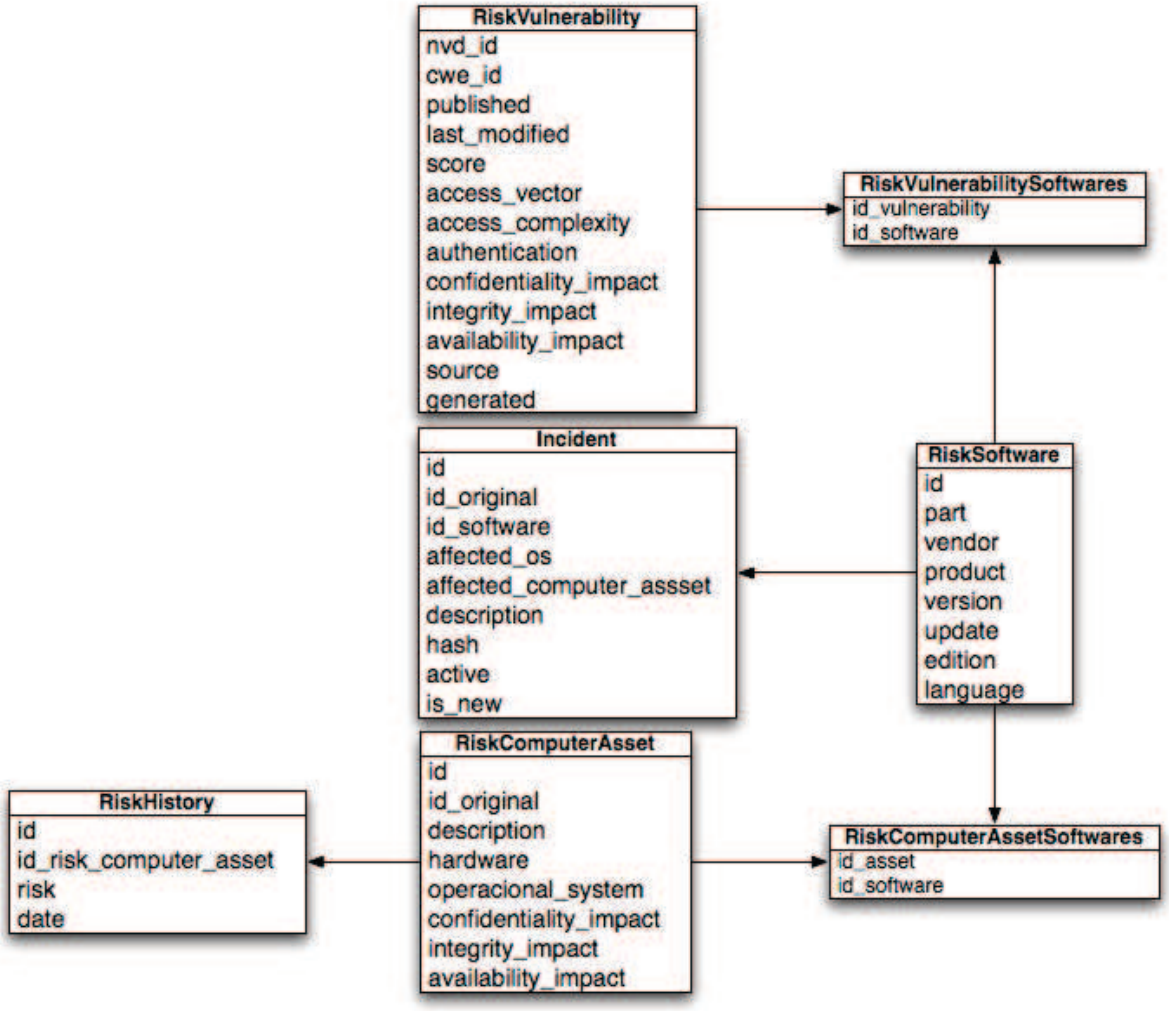


Figura 6.5: Modelo ER da base de dados *DB Risk*

6.3 AGENTES

Esta seção apresenta os agentes que compõem o sistema, assim como suas ligações com os banco de dados, suas percepções, ações, comunicações com outros agentes e o seu papel dentro do sistema.

6.3.1 Ag Asset

O agente “ag asset" lida com as informações dos ativos de TI da organização. O ambiente sobre o qual ele atua é o banco de dados de configuração de ativos, que recebe o nome de “db asset" neste sistema. O papel que este agente tem no sistema é de monitorar alterações na configuração de ativos da organização.

As percepções que este agente tem do ambiente são (i) *“percept new asset"* (perceber um novo ativo), que detecta a existência de um novo ativo na organização e (ii) *“percept change asset"* (perceber alteração na configuração de ativos), que detecta a alteração da configuração de um ativo.

Para este trabalho, assume-se que a organização possui um processo de gestão de ativos e que mantém a base de dados de configuração de ativos sempre atualizada.

O termo configuração é utilizado pelo fato de um ativo poder ser composto por um conjunto de outros ativos. Isso possibilita que um computador (*hardware*), que é um ativo, tenha um sistema operacional *Windows* instalado, que também é um ativo. Caso este computador seja formatado e tenha o sistema operacional *Linux* instalado, uma alteração na configuração de um ativo foi realizada e deve ser identificada.

A partir destas percepções, o agente executa as ações *“act inform new asset"* (informar um novo ativo) e *“act inform change asset"* (informar alteração de um ativo), efetuando então seu papel de identificar alterações no ambiente que recebe o nome de *“rule chance asset environment"*. O outro papel que este agente possui é o *“rule answer about asset"*, que responde a consultas sobre os ativos.

A comunicação deste agente com o agente *“ag risk"* é realizada através das mensagens que são apresentadas na Tabela 6.1.

6.3.2 Ag Incident

O agente *“ag indicent"* lida com as informações relacionadas aos incidentes de Segurança da Informação que ocorreram nos ativos de TI da organização. O ambiente sobre o qual ele atua é o banco de dados com o histórico de incidentes de Segurança da organização, ou seja, o banco *“db incident”*.

Tabela 6.1: Mensagens trocadas com o agente “ag asset".

Nome	**Descrição**	**Origem**	**Destino**
msg inform new asset	Informa a existência de um novo ativo no ambiente	*ag risk*	*ag asset*
msg inform change asset	Informa a existência de uma alteração na configuração de um ativo no ambiente	*ag risk*	*ag asset*
msg ask about asset	Solicita informações com relação aos ativos	*ag asset*	*ag risk*
msg answer about asset	Retorna informações sobre os ativos	*ag risk*	*ag asset*

As percepções do agente *“ag incident”* tem do ambiente são (i) *“percept new incident”* (perceber um novo incidente), que detecta o relato de um novo incidente de segurança e (ii) *“percept change incident"* (perceber alteração em um incidente) que detecta que a descrição de um incidente foi alterada. Incidentes podem afetar mais de um ativo. Existem casos que isto só é identificado em um segundo momento, necessitando então que um relato do incidente seja atualizado. O trabalho assume que a organização possui uma base de dados em que os incidentes de segurança são informados pelos responsáveis pelo setor de TI.

A partir das percepções do agente, este executa as ações *“act inform new incident"* (informar um novo incidente) e *“act inform change incident”* (informar alteração de um incidente), efetuando então seu papel de identificar alterações no ambiente, que recebe o nome de *“rule change incident environment”*. O outro papel que este agente possui é o *“rule answer about incident"*, que responde a consultas sobre os incidentes.

A comunicação deste agente com o agente *“ag risk"* é realizada através das mensagens que são apresentadas na Tabela 6.2.

Tabela 6.2: Mensagens trocadas com o agente “ag incident”.

Nome	**Descrição**	**Origem**	**Destino**
msg inform new incident	Informa a existência de um novo incidente no ambiente	*ag risk*	*ag incident*
msg inform change incident	Informa a existência de uma alteração na configuração de um incidente no ambiente	*ag risk*	*ag incident*
msg ask about incident	Solicita informações com relação aos incidentes	*ag incident*	*ag risk*
msg answer about incident	Retorna informações sobre os incidentes	*ag risk*	*ag incident*

6.3.3 Ag Vulnerability

O agente *"ag vulnerability"* lida com as informações relacionadas a vulnerabilidades de Segurança da Informação de ativos de TI. O ambiente sobre o qual ele atua é o banco de dados com o histórico de vulnerabilidades de ativos de TI em uma escala mundial.

As percepções que o agente *"ag vulnerability"* tem do ambiente são: (i) *"percept new velnerability"* (perceber uma nova vulnerabilidade), que detecta o relato de uma nova vulnerabilidade de segurança e (ii) *"percept change vulnerability"* (perceber alteração em uma vulnerabilidade) que detecta que houve alguma alteração no relato de uma vulnerabilidade. Uma vulnerabilidade pode afetar diversas versões de um ativo (como o *Windows*). Em um primeiro momento, uma vulnerabilidade pode ter sido identificada para o *Window XP*, posteriormente, pode ter sido constatado que o *Windows Vista* também é vulnerável a mesma vulnerabilidade. Neste caso, o relato desta vulnerabilidade terá uma alteração para adicionar o novo ativo vulnerável.

Tabela 6.3: Mensagens trocadas com o agente "ag vulnerability".

Nome	**Descrição**	**Origem**	**Destino**
msg inform new vulnerability	Informa a existência de uma nova vulnerabilidade no ambiente	*ag risk*	*ag vulnerability*
msg inform change vulnerability	Informa a existência de uma alteração na descrição de uma vulnerabilidade	*ag risk*	*ag vulnerability*
msg ask about vulnerability	Solicita informações com relação a vulnerabilidade	*ag vulnerability*	*ag risk*
msg answer about vulnerability	Retorna informações sobre vulnerabilidades	*ag risk*	*ag vulnerability*

A partir das percepções do agente, ele executa as ações *"act inform new vulnerability"* (informar uma nova vulnerabilidade) e *"act inform change vulnerability"* (informar alteração de uma vulnerabilidade), efetuando então seu papel de identificar alterações no ambiente, que recebe o nome de *"rule chance vulnerability environment"*. O outro papel que este agente possui é o *"rule answer about vulnerability"*, que responde a consultas sobre vulnerabilidades.

A comunicação deste agente com o agente *"ag risk"* é realizada através das mensagens que são apresentadas na Tabela 6.3.

6.3.4 Ag Risk

Este agente é o centro do sistema, sendo responsável por administrar a Rede Bayesiana, ou seja, adicionar e remover nós da rede, realizar consultas na rede e consultar

outros agentes do sistema para complementar a rede. O agente ‘*‘ag risk"* troca mensagens com todos os agentes do sistema.

O funcionamento deste agente é descrito na Seção 6.4, que aborda o modo pelo qual é gerada a Rede Bayesiana.

6.4 CRIAÇÃO DA REDE BAYESIANA

Esta seção descreve o método que está sendo proposto para criar uma Rede Bayesiana com base nos ativos de TI da organização, suas vulnerabilidades e o histórico de incidentes de segurança da organização.

A criação da Rede Bayesiana é baseada na idéia do método proposto por Fenz e Hudec (2009). No entando, este trabalho trata as vulnerabilidades de uma forma diferente. Fenz e Hudec (2009) utilizam em sua ontologia vulnerabilidades de processo, como por exemplo *“No Security Audits"* (Sem auditorias de segurança), que indica que não foram realizadas auditorias de segurança na organização. O presente trabalho lida com vulnerabilidades técnicas de ativos eletrônicos que são reportadas em bases de dados públicas, como apresentado na Seção 2.4.

O método proposto para gerar automaticamente uma Rede Bayesiana é composto por três etapas, que são estas:

1. identificar novos nós e relacioná-los;
2. determinar pesos e escalas; e
3. validar a rede.

Cada uma destas etapas é comentada com detalhes nas subseções subsequentes.

6.4.1 Identificar e Relacionar os Nós

Esta etapa utiliza as informações enviadas pelo agente *“ag Asset”*, que são relacionadas a configuração dos ativos da organização. Esta etapa consistem em gerar um grafo representando a relação entre os ativos de acordo com o método proposto.

A relação proposta no método é organizada da seguinte forma:

- Os ativos computacionais tem como pais, todos os ativos de aplicação e de sistema operacional que o compõem; e

- Os ativos de sistema operacional tem como pais todos os ativos de aplicação que estão instalados sobre ele e também o ativo de *hardware* em que está instalado.

Para elaborar esta relação foi considerado que uma vulnerabilidade em um aplicativo pode comprometer o sistema operacional no qual ele está instalado e/ou comprometer a função para qual o ativo computacional é destinado. Por exemplo, a falha de um *hardware* pode comprometer o funcionamento do sistema operacional. E por fim, falhas em aplicativos e sistemas operacionais podem comprometer a função para qual o ativo computacional é destinado.

Esta organização forma um grafo de quatro níveis, cada um representando um tipo de ativo (*hardware*, aplicativo, sistema operacional e computacional). Como as mensagens que são trocadas entre os agentes são baseadas no padrão de nomenclatura CPE e este possui apenas três classificações para plataforma, que são *hardware* (h), sistema operacional (o) e aplicação (a), foi adicionado uma classificação a este padrão, que é o "computacional" (c), sendo este utilizado apenas no método proposto no presente trabalho. Os demais ativos são apenas considerados como parte integrante de um ativo computacional.

Para evitar nós duplicados, são executadas os seguintes passos : (i) verificar se o ativo computacional já existe na rede, caso não exista, este é adicionado como um nó; (ii) verifica se os ativos que compõem o ativo computacional já existem na rede, caso não existam, estes são adicionados; (iii) remover a relação do ativo computacional com ativos que não mais pertencem a sua configuração; e (iv) adicionar relações com novos itens de configuração (ativos). Este fluxo pode ser melhor visualizado na Figura 6.6.

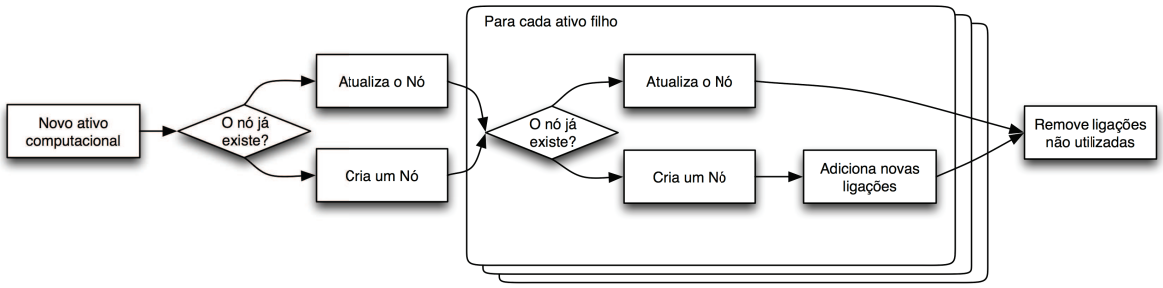
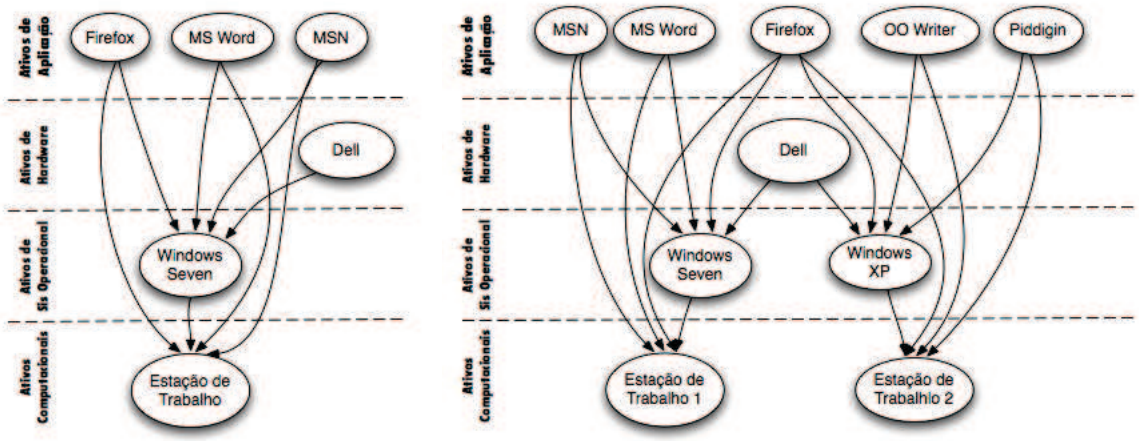


Figura 6.6: Fluxo para adição de um ativo computacional na Rede Bayesiana.

A Figura 6.7 apresenta dois grafos representando as relações de uma Rede Bayesiana. O grafo que pode ser visto na Figura 6.7a representa o relacionamento da Rede Bayesiana com apenas um ativo computacional, que é o ativo que recebe o nome de "Estação de Trabalho 1". O grafo apresentado na Figura 6.7b mostra o estado desta mesma rede após acrescentar um novo ativo computacional, o qual recebeu o nome de "Estação de

Trabalho 2". É possível perceber que além do nó referente ao ativo "Estação de Trabalho 2", surgiram outros nós, que são referentes aos ativos que integram o ativo computacional, que são eles: *OO Writer*, *Piddigin* e *Windows XP*. Note que os nós *Firefox* e *Dell* não foram duplicados, visto que são utilizados nos dois ativos computacionais existentes.



(a) Rede Bayesiana inicial (b) Rede Bayesiana após etapa dois

Figura 6.7: Exemplo de conexões em uma Rede Causal.

Para melhor entender o relacionamento da rede gerada na etapa 2, a Figura 6.7a será utilizada como exemplo. Na Figura 6.7a pode ser visto o nó "Estação de Trabalho 1". A probabilidade de uma vulnerabilidade vir a ser explorada no ativo que este nó representa depende da probabilidade de uma vulnerabilidade vir a ser explorada nos ativos que compõem este ativo, ou seja, seus pais. Na caso do exemplo da Figura 6.7a, os pais do nó "Estação de Trabalho 1" são os nós: *"Firefox"*, *"MS Word"*, *"MSN"* e *"Windows Seven"*. O mesmo ocorre para os nós *"Firefox"*, *"MS Word"* e *"MSN"*, que possuem uma dependência do nós *"Windows Seven"*.

No grafo apresentado na Figura 6.7b é possível visualizar um caso que não é visto no grafo apresentado na Figura 6.7b. Neste grafo, o nó que é referente ao ativo *"Firefox"* possui influência sobre os nós "Estação de Trabalho 1" e "Estação de Trabalho 2" e é influenciado pelo nós referentes aos ativos *Windows Seven* e *Windows XP*.

6.4.2 Determinar Pesos e Escalas

Com os nós da rede já atualizados, tem-se início a segunda etapa, que consiste em determinar pesos e escalas. Nesta etapa as Tabelas de Probabilidade Condicional (TPC) são geradas. Esta etapa é divida em dois passos distintos. O primeiro passo gera todos os estados possíveis, dados os pais do nós. O segundo passo calcula os peso de cada uma das possibilidades geradas.

Estes dois passos são executados pelo Algoritmo 6.1, sendo o primeiro passo executado na linha 6, em que é executado o Algoritmo 6.2, responsável por gerar a estrutura da TPC. A segunda etapa é executada no trecho de código entre as linhas 7 e 15 do Algoritmo 6.1.

Algorithm 6.1: Algoritimo para geração de TPC

```
1  nodes_finalized ← um conjunto com os nós já processos, inicialmente vazio
2  nodes ← fila contendo todos os nós da rede
3  while nodes ≠ ∅ do
4  |  node ← nodes.dequeue()
5  |  if length(node) == 0 or parents(node) ⊂ nodes_finalized then
6  |  |  GENERATE-CPT(node)
7  |  |  if length(node) == 0 then
8  |  |  |  P(node) = probapriore(node)
9  |  |  |  nodes_finalized.add(node)
10 |  |  |  continue
11 |  |  end
12 |  |  foreach valor k_i de node.cpt do
13 |  |  |  node.cpt[k_i] = getstatistic(node)
14 |  |  |  nodes_finalized.add(node)
15 |  |  endfor
16 |  else
17 |  |  nodes.queue(node)
18 |  end
19 end
```

Algorithm 6.2: GENERATE-CPT

```
   input: um nó da rede node
1  if length(node) == 0 then
2  |  P(node) = null
3  |  continue
4  else
5  |  auxctp = null
6  |  lastcpt = node.parents[0]
7  |  foreach valor p_i de node.parents do
8  |  |  row = (True, False)
9  |  |  foreach valor k1_i de p.cpt do
10 |  |  |  foreach valor k2_i de row do
11 |  |  |  |  auxcpt[k1 ∪ k2] = null
12 |  |  |  end
13 |  |  end
14 |  |  lastcpt = auxcpt
15 |  end
16 |  node.cpt = lastcpt
17 end
```

Para gerar a TPC é necessário que a TPC de todos os nós pais já estejam criadas.

Para não utilizar recursividade, que aumenta a complexidade do algoritmo, foi usada um fila para armazenar os nós que já tiveram sua TPC calculada. Pode-se perceber que o Algoritmo 6.1 permanece em uma laço até que todos os nós da rede sejam processados. Na linha 5 do Algoritmo 6.1 é realizado um teste para verificar se o nó que vai ser processado possui pais, caso positivo, se todos os pais pertencem ao conjunto de nós já processados. Caso este teste não seja verdadeiro, o nó é adicionado ao final da fila e será processado futuramente. Caso o teste retornando verdadeiro, o algoritmo continua sua execução e ao seu término, o nó é adicionado no conjunto de nós já processados.

A linha 6 do Algorimo 6.1 gera a estrutura da TPC do nó que está sendo processado através do Algoritmo 6.2. O Algoritmo 6.2 apenas gera a estrutura da TPC, sendo que os valores da TPC serão atribuídos nos próximos passos do Algoritmo 6.1.

Para atribuir os valores a TPC gerada, são considerados dois casos. O primeiro caso é quando o nó não possui pais. Neste caso o valor atribuído a TPC é a probabilidade *a priore* do nó, que neste caso é a pontuação básica do ativo no CVSS. O outro caso é quando o nó possui nós pais. Possuindo nós pais é atribuído um valor obtido através de uma análise estatística na base de incidentes reportados. Esta análise leva em consideração a quantidade de incidentes relacionados ao mesmo tipo de ativo do pai que afetaram o mesmo tipo de ativo do nós filhos. Desta forma é possível obter o impacto que determinado ativo tem sobre outro, caso este não exerça sua função corretamente. Neste caso o valor atribuído utiliza a equação:

$$\frac{NOA}{NO} * BS \tag{6.1}$$

para calcular sua probabilidade *a priori*, onde NOA é o número total de ocorrências de incidentes que afetam o ativo computacional, NO é o número de ocorrências de incidentes e BS é a pontuação básica da vulnerabilidade.

Obtendo os valores da TPC através de estatística real baseada em uma base de incidentes que ocorrem no ambiente de tecnologia da informação da organização leva a uma Rede Bayesiana que reflete a realidade desta organização e que é ajustada de acordo com os acontecimentos neste ambiente. A possibilidade de utilizar dados de outras organizações disponibilizados anonimamente em uma base de dados cooperativa será avaliada em trabalhos futuros.

6.4.3 Validar a Rede

Para garantir que a rede gerada até a etapa dois é realmente uma Rede Bayesina, é necessário validar esta rede, sendo essa a terceira etapa do método. Essa etapa consiste em verificar se o grafo gerado é acíclico, e também, as TPCs geradas.

Para verificar se o grafo é acíclico, é percorrido todos os nós do grafo. Caso o nó inicial seja encontrado durante o percurso, é identificado um ciclo no grafo. Este processo é realizado para todos os nós do grafo. Caso seja identificado um ciclo no grafo, o nó que está sendo adicionado ao grafo é desconsiderado. Quando o nó é desconsiderado por estar gerando um ciclo no grafo, é gerado um alerta para informar a situação ao usuário (administrador), possibilitando que possa ser realizada uma análise na configuração do ativo na base de dados de configuração de ativos para identificar alguma provável falha no processo.

A validação da TPC é realizada em duas etapas. A primeira etapa valida o número de linhas existentes na TPC. O número de linhas deve ser igual 2^n, onde n é a quantidade de nós pais do nó atual, visto que a Rede Bayesiana utilizada no presente trabalho utiliza apenas nós booleanos, isto é, que possuem apenas como valor, verdadeiro ou falso. Esta validação é representado pela equação:

$$rows(x) = 2^n \tag{6.2}$$

onde $rows$ é uma função que retorna o número de itens de uma TPC e n é o número de pais de x.

A segunda etapa valida a distribuição de probabilidade conjunta, que é representada pela equação:

$$\sum_{\omega \in \Omega} P\{\omega\} = 1 \tag{6.3}$$

onde Ω o espaço amostral e ω representa a probabilidade de um evento.

Finalizando as três etapas, uma Rede Bayesiana está criada e apontando os principais riscos de TI existentes no ambiente da organização, favorecendo a identificação de vulnerabilidades e ameaças críticas para a organização.

A Rede Bayesiana gerada será alterada sempre antes que os riscos de TI sejam calculados. O período de tempo em que o risco é calculado é definido pelo administrador do sistema. A possibilidade de alterar a Rede Bayesiana sempre que uma alteração no sistema fosse identificada foi considerada, no entanto, esta possibilidade se mostrou inviável, visto o tempo que o algoritmo leva para gerar e calcular as TPC.

6.5 CÁLCULO DO RISCO DE TI

Possuindo a Rede Bayesiana que reflete o cenário atual da estrutura de TI da organização e conhecendo os ativos que fazem parte desta estrutura, é possível calcular o potencial risco destes ativos computacionais.

Sendo o risco uma medida da extensão em que uma entidade está ameaçada por uma circunstância ou evento em potencial e, normalmente, uma função de: (i) os impactos negativos que se verificariam se a circunstância ou evento ocorresse; e (ii) a probabilidade de ocorrência (LOCKE e GALLAGHER, 2011), cálculo é realizado utilizando a fórmula base para se obter o índice de risco (CICCO, 2005):

$$R = P \text{ x } I \tag{6.4}$$

onde R é o risco, P é a probabilidade de um ativo vir a não executar sua função e I é o impacto do ativo computacional para a organização.

Como o agente *"ag Risk"* conhece todos os ativos identificados pelo agente *"ag Asset"*, além de conhecer todos os ativos que compõem este ativo e possuir acesso a Rede Bayesiana, é possível obter todas as informações necessárias para calcular os riscos associados aos ativos.

O impacto utilizado no cálculo de risco é obtido através da pontuação do CVSS, obtida utilizando as métricas de ambiente, conforme visto na Seção 2.4.2. As métricas temporais foram ignoradas para este cálculo, visto que o sistema não possui os dados necessários para o cálculo.

Para se obter a probabilidade de um ativo vir a não executar sua função, é utilizada a Rede Bayesiana utilizando como evidências, todos os incidentes que estão ativos na base de dados de incidentes. Com estas evidências, é possível realizar a inferência $P(X|e)$, que é exemplificada a seguir:

$$P(WorkStation1|Firefox = True, WindowsXP = True)$$

Esta inferência pode ser interpretada da seguinte forma: qual a probabilidade de um ativo não executar corretamente sua função dado o não funcionamento dos ativos utilizados como evidência.

Desta forma, podemos afirmar que o risco associado a um ativo é definido por:

$$R(x) = P(x|e)I(x) \tag{6.5}$$

onde R é o risco, x é o ativo computacional P é a probabilidade de um ativo não desempenhar corretamente sua função, e são as evidências encontradas e I é o impacto sobre o negócio da organização.

Utilizando esta fórmula, o agente *"ag Risk"* reliza o cálculo de risco de todos os ativos computacionais existentes na organização e armazena estas informações em uma base de dados que possui as informações consolidadas. Com estas informações consolidadas, o agente *"ag Risk"* tem a possibilidade de retornar consultas referentes aos riscos dos ativos de uma forma mais eficiente, sem a necessidade de realizar inferências na Rede Bayesiana sempre que uma informação for solicitada.

6.6 IMPLEMENTAÇÃO

Para avaliar o método proposto, foi desenvolvido um protótipo de Sistema Multiagente. Este protótipo foi desenvolvido nos seguintes linguagens/*frameworks*:

- *Python*
- SPYSE
- *Django*
- *SQLite*

A linguagem *Python*[1] foi adotada por atender as seguintes características:

Simples: Sua sintaxe é simples e de fácil compreensão;

Alto Nível: Manipula automáticamente a memória e converte tipos automaticamente;

Portável: Funciona em um grande número de plataformas;

Orientada a Objetos: Além de suportar o programação procedural (estruturada), tem suporte a programação orientada a objetos, assim como Java e C++; e

Extensível: Possibilita que trechos críticos do código sejam implementados em C/C++ e utilizados a partir do código *Python*.

Não é intuito deste trabalho pesquisar sobre as características da linguagem e nem ao menos fazer apologia, sendo assim, estão sendo descritas apenas as características que levaram a optar por esta linguagem.

[1]Maiores informações sobre a linguagem *Python* pode ser encontradas em http://www.python.org

O *framework Smart Python Simulation Environment* SPYSE foi utilizado por ser implementado em *Python* e por seguir uma estrutura e forma de utilização do *framework* JADE (implementado em Java). Sua implementação é guiada pela documentação do JADE. É um *framework* que suporta troca de mensagens no padrão FIPA, ontologias, entre outras característica que podem vir a ser utilizadas em futuros trabalhos relacionados ao presente trabalho.

O *framework Django*[2], apesar de ser um *framework* destinado ao desenvolvimento de aplicações Web, fornece uma camada de acesso a dados que abstrai o banco de dados relacional, permitindo que o banco de dados seja manipulado através de objetos. Por esta facilidade em acessar o banco de dados, este *framework* foi utilizado.

Apesar da camada de acesso a dados do *Django* possuir suporte aos principais banco de dados existentes atualmente, foi utilizado o *SQLite*[3], que é um banco de dados de acesso local simples. O *SQLite* pode ser utilizado em diversos tipos de ambientes, incluindo navegadores de internet, computadores convencionais, servidores e inclusive dispositivos móveis, visto que é um banco de dados que necessita de poucos recursos para seu funcionamento, característica esta, que é de interesse do presente trabalho.

A implentação da Rede Bayesiana não utilizou nenhum *framework* em especial. O código desenvolvido para a Rede Bayesiana foi baseado no código disponibilizado no livro de Russel e Norvig (2004), que foi desenvolvido em *Python*. O código original não é orientado a objetos, portanto, esta foi a primeira modificação realizada. As demais alterações foram para adicionar as funcionalidades necessárias para que a rede pudesse ser gerada automaticamente.

6.7 AVALIAÇÃO

Como o presente trabalho aborda um tema atual e com poucos trabalhos relacionados publicados, a comparação dos resultados obtidos com outros resultados se torna inviável. Para avaliar a eficiência do sistema desenvolvido, foi elaborada um simulação composta por treze etapas. Levando em consideração que cenários complexos geram uma Rede Bayesiana grande, o que gera um problema de desempenho, a simulação utiliza um cenário controlado e resumido.

Sendo assim, a avaliação se limitou a cinco ativos computacionais, utilizando para teste apenas seus principais ativos, ou seja, os ativos relacionados diretamente à função a qual o ativo computacional é destinado.

[2]Maiores informações sobre o *framework Django* podem ser obtidas em http://www.djangoproject.com
[3]Maiores informações sobre o banco de dados *SQLite* podem ser encontrato em http://www.sqlite.org

6.7.1 Simulação

Para simular um ambiente onde o sistema poderá ser utilizado, foi elaborado um cenário controlado e resumido que represente o mais próximo possível da realidade. Este cenário se resume aos principais ativos que compõem o ativo computacional. Para isso, foi montada uma base de dados de configuração de ativos que simula o ambiente fornecido, porém, deixando alguns ativos reservados para a simulação, atendendo então o cenário da inclusão/alteração de ativos na base de dados de configuração de ativos.

Para gerar os dados para a simulação, foi utilizado uma base de dados de nomes CPE, que tem a função de dicionário para consultas de nomes CPE já cadastrados. Esta base de dados é fornecida livremente pelo NVD[4].

Para simular a identificação de novas vulnerabilidades, a importação das informações da base de dados de vulnerabilidades também deixou registros reservados para testes. Como os dados sobre incidentes não foram cedidos, os eventos são todos simulados.

O seguinte roteiro foi elaborado para simular o funcionamento do sistema e avaliar o método proposto:

1. Realizar carga inicial de dados de vulnerabilidades;
2. Realizar carga inicial da base histórica de incidentes;
3. Adicionar um ativo computacional na base de configuração de ativos;
4. Adicionar um segundo ativo computacional na base de configuração de ativos;
5. Adicionar um terceiro ativo computacional na base de configuração de ativos;
6. Adicionar um quarto ativo computacional na base de configuração de ativos;
7. Adicionar um quinto ativo computacional na base de configuração de ativos;
8. Adicionar uma evidência direcionada para um ativo utilizado em apenas um ativo computacional;
9. Adicionar uma evidência direcionada para um ativo utilizado em mais de um ativo computacional;
10. Remover software de um ativo computacional na base de dados de configuração de ativos;
11. Acrescentar software em um ativo computacional;

[4]Disponível em http://static.nvd.nist.gov/feeds/xml/cpe/dictionary/official-cpe-dictionary_v2.2.xml

12. Atualizar base de vulnerabilidades com os arquivos recentes e modificados; e

13. Alterar um ativo computacional, aumentando o seu impacto perante o negócio da organização.

Ao iniciar o sistema, é executada a primeira etapa do roteiro. Esta é uma etapa que irá ocorrer apenas um vez, visto que ela é destinada a realizar a importação das vulnerabilidades disponibilizadas como base histórica na NVD. Para a simulação do sistema foram utilizadas apenas as vulnerabilidade referentes aos anos 2009 e 2010 para diminuir o tempo de carga dos demais anos. As vulnerabilidades do ano de 2011 são utilizadas durante o roteiro para simular o surgimento de novas vulnerabilidades.

Na segunda etapa do roteiro, é gerada uma base histórica de incidentes de segurança. Esta base é gerada a partir da seleção aleatória de ativos na base de dados de vulnerabilidades do NVD. A base de incidentes de segurança é alterada durante a simulação.

Da etapa três à sete são adicionados ativos à base de dados de configuração de ativos. A cada etapa realizada é verificada a estrutura da Rede Bayesiana gerada e é calculado pelo sistema o risco para os ativos computacionais.

Na etapa três, é adicionado um servidor que recebe o nome de *Zeus*. Este ativo computacional é composto pelos ativos apresentados na Tabela 6.4.

Tabela 6.4: Ativos que compõem o ativo computacional *Zeus*.

Hardware	**Identificador**
h:dell:poweredge:830	PowerEdge
Sistema Operacional	**Identificador**
o:microsoft:windows_2003_server::sp2:standard	
Aplicativos	**Identificador**
a:avg:avg_anti-virus:8.0	AVG 8
a:march-hare:cvsnt:2.5	CVSNT 2.5
a:microsoft:ie:7	IE 7
a:microsoft:sql_server:2005:sp2	SQL 2005
a:microsoft:.net_framework:1.0	.NET FW 1.0
a:microsoft:.net_framework:2.0:sp1	.NET FW 2.0
a:microsoft:.net_framework:3.5	.NET FW 3.5
a:sun:jre:1.6.0:update_1	JRE 1.6
a:sun:jdk:1.6.0:update_10	JDK 1.6
a:cobiam:cobian_backup:8	Cobian 8

Na etapa quatro, é adicionado um servidor que recebe o nome de *Apolo*. Este ativo computacional é composto pelos ativos apresentados na Tabela 6.5.

Tabela 6.5: Ativos que compõem o ativo computacional *Apolo*.

Hardware	**Identificador**
h:dell:poweredge:830	PowerEdge
Sistema Operacional	**Identificador**
o:microsoft:windows_2003_server::sp2:standard	
Aplicativos	**Identificador**
a:avg:avg_anti-virus:8.0	AVG 8
a:microsoft:ie:7	IE 7
a:microsoft:.net_framework:1.0	.NET FW 1.0
a:microsoft:.net_framework:2.0:sp1	.NET FW 2.0
a:microsoft:.net_framework:3.5	.NET FW 3.5
a:vmware:server:1.0.4	VMWare

Na etapa cinco, é adicionada uma estação de trabalho que recebe o nome de *W140*. Este ativo computacional é composto pelos ativos apresentados na Tabela 6.6.

Tabela 6.6: Ativos que compõem o ativo computacional *W140*.

Hardware	**Identificador**
h:dell:vostro:200	Vostro
Sistema Operacional	**Identificador**
o:microsoft:windows_xp::sp2	WinXP
Aplicativos	**Identificador**
a:avira:antivir	Avira
a:openoffice:openoffice.org :3.1	OO 3.1
a:7-zip:7-zip:9.11	7-Zip 9.11
a:sap:crystal_reports:2008	Crystal 2008
a:foxitsoftware:foxit_reader:3.0	Foxit
a:microsoft:ie:7	IE 7
a:microsoft:.net_framework:1.0	.NET FW 1.0
a:microsoft:.net_framework:2.0:sp1	.NET FW 2.0
a:microsoft:.net_framework:3.5	.NET FW 3.5
a:microsoft:sql_server_express:2005	SQLx 2005
a:tigris:tortoisecvs1.1.1	TortoiseCVS
a:microsoft:visual_basic_sdk:6.3	VB 6.3

Na etapa seis, é adicionada uma estação de trabalho que recebe o nome de *W265*. Este ativo computacional é composto pelos ativos apresentados na Tabela 6.7.

Tabela 6.7: Ativos que compõem o ativo computacional *W265*.

Hardware	**Identificador**
h:dell:vostro:200	Vostro
Sistema Operacional	**Identificador**
o:microsoft:windows_vista:sp1	WinVista
Aplicativos	**Identificador**
a:adobe:acrobat_reader:8.0	Acrobat
a:winrar:winrar_archiver	WinRAR
a:avg:avg_anti-virus:8.0	AVG 8
a:sap:crystal_reports:2008	Crystal 2008
a:microsoft:.net_framework:1.0	.NET FW 1.0
a:microsoft:.net_framework:2.0:sp1	.NET FW 2.0
a:microsoft:.net_framework:3.5	.NET FW 3.5
a:microsoft:sql_server_express:2005	SQLx 2005
a:tigris:tortoisecvs1.1.1	TortoiseCVS
a:microsoft:visual_basic_sdk:6.3	VB 6.3
a:microsoft:visual_studio:2008	VS 2008
a:microsoft:ie:7	IE 7
a:mozilla:firefox:3.6	Firefox 3.6

Na etapa sete, é adicionada uma estação de trabalho que recebe o nome de *W126*. Este ativo computacional é composto pelos ativos apresentados na Tabela 6.8.

Tabela 6.8: Ativos que compõem ao ativo computacional *W126*.

Hardware	**Identificador**
h:dell:vostro:200	Vostro
Sistema Operacional	**Identificador**
o:microsoft:windows_xp::sp2	WinXP
Aplicativos	**Identificador**
a:adobe:acrobat_reader:8.0	Acrobat
a:avira:antivir	Avira
a:sap:crystal_reports:2008	Crystal 2008
a:tigris:tortoisecvs1.1.1	TortoiseCVS
a:microsoft:.net_framework:1.0	.NET FW 1.0

a:microsoft:.net_framework:2.0:sp1	.NET FW 2.0
a:microsoft:.net_framework:3.5	.NET FW 3.5
a:microsoft:sql_server_express:2005	SQLx 2005
a:microsoft:visual_studio:2008	VS 2008
a:microsoft:visual_basic_sdk:6.3	VB 6.3
a:microsoft:ie:7	IE 7
a:winrar:winrar_archiver	winRAR

Na etapa oito, é adicionado, à base de dados de incidentes, um novo incidente relacionado ao ativo *a:7-zip:7-zip:9.11*. Este ativo pertence apenas ao ativo computacional *W140*.

Na etapa nove, é adicionado, um aplicativo base de dados de incidentes, um segundo incidente relacionado ao ativo *a:tigris:tortoisecvs1.1.1*. Este ativo pertence a todas as estações de trabalhos existentes no sistema.

Na etapa dez, é removido o aplicativo *a:microsoft:office:2007:sp1::* do ativo computacional *W265*, validando então a capacidade de alterar a Rede Bayesiana gerada e também a mudança no risco relacionado ao ativo computacional *W265*.

Na etapa onze, é acrescentado o software *a:google:chrome:8.0.552.344* no ativo computacional *w140*. Sendo que o aplicativo *a:google:chrome:8.0.552.344* possui registros de vulnerabilidades no NVD.

Na etapa doze, é realizada a importação de novas vulnerabilidades e também das alteradas, utilizando para isto o período de vulnerabilidades reservado na etapa inicial para a simulação.

Na etapa treze, o ativo computacional *w126* tem seu impacto alterado, aumentando seu valor perante o negócio da organização. Com esta última etapa, todas as características propostas para o presente trabalho foram simuladas.

Os resultados obtidos durante a simulação são apresentados na subseção 6.7.2, assim como os comentários relacionados aos resultados.

6.7.2 Análise dos Resultados

Esta subseção apresenta os dados obtidos na simulação descrita na subseção 6.7.1.

Os resultados obtidos a partir do sistema proposto estão normalizados em valores entre 0 a 1, inclusive, com o intuito de manter a compatibilidade entre os valores utilizados na inferência da rede bayesiana e no cálculo do CVSS.

A Tabela 6.9 apresenta os fatores de risco de cada ativo computacional ao longo da simulação. Os campos em branco na Tabela 6.9 são referentes a inexistência do ativo computacional na etapa da simulação.

Tabela 6.9: Risco obtidos durante a simulação.

Etapa	**Zeus**	**Apolo**	**w140**	**w265**	**w126**
1					
2					
3	0,283144919				
4	0,283566426	0,297289929			
5	0,283566426	0,297289929	0,235670548		
6	0,283566426	0,297289929	0,235670548	0,307948209	
7	0,283566426	0,297289929	0,23674143	0,307948209	0,224901298
8	0,283611221	0,297367254	0,266900968	0,307972788	0,225997108
9	0,295356363	0,31302739	0,274954698	0,315634269	0,232222123
10	0,295356363	0,31302739	0,274954698	0,306513963	0,232222123
11	0,295356363	0,31302739	0,257630576	0,306513963	0,231606591
12	0,289427769	0,312866358	0,266537442	0,306513963	0,231981113
13	0,289427769	0,312866358	0,266537442	0,306513963	0,311626667

As probabilidades obtidas através de inferência na Rede Bayesiana, utilizadas para calcular os fatores de risco apresentados na Tabela 6.9, são apresentadas na Tabela 6.10. Os campos em branco na Tabela 6.10 são referentes a inexistência do ativo computacional na etapa da simulação.

Tabela 6.10: Probabilidades obtidas durante a simulação.

Etapa	**Zeus**	**Apolo**	**w140**	**w265**	**w126**
1					
2					
3	0,300722957				
4	0,301170632	0,3381885			
5	0,301170632	0,3381885	0,282151536		
6	0,301170632	0,3381885	0,282151536	0,334536658	
7	0,301170632	0,3381885	0,283433627	0,334536658	0,32359194
8	0,301218208	0,338276463	0,319541491	0,33456336	0,325168612
9	0,313692504	0,356090984	0,329183647	0,342886338	0,334125273
10	0,313692504	0,356090984	0,329183647	0,333799764	0,334125273

11	0,313692504	0,356090984	0,308442711	0,333799764	0,333239634
12	0,310486515	0,355907799	0,320707178	0,333799764	0,333778503
13	0,310486515	0,355907799	0,320707178	0,333799764	0,333778503

Outro item que compõe o cálculo do risco é o impacto que o ativo tem sobre o negócio da organização. Os impactos utilizados para calcular os fatores de risco apresentados na Tabela 6.9 são apresentados na Tabela 6.11. Os campos em branco na Tabela 6.11 são referentes a inexistência do ativo computacional na etapa da simulação.

Tabela 6.11: Impactos obtidos durante a simulação.

Etapa	**Zeus**	**Apolo**	**w140**	**w265**	**w126**
1					
2					
3	0,941547403				
4	0,941547403	0,879065756			
5	0,941547403	0,879065756	0,835262325		
6	0,941547403	0,879065756	0,835262325	0,920521568	
7	0,941547403	0,879065756	0,835262325	0,920521568	0,69501514
8	0,941547403	0,879065756	0,835262325	0,920521568	0,69501514
9	0,941547403	0,879065756	0,835262325	0,920521568	0,69501514
10	0,941547403	0,879065756	0,835262325	0,918256978	0,69501514
11	0,941547403	0,879065756	0,835262325	0,918256978	0,69501514
12	0,932175005	0,879065756	0,83109285	0,918256978	0,69501514
13	0,932175005	0,879065756	0,83109285	0,918256978	0,933633125

As duas primeiras etapas da simulação descrita na subseção 6.7.1 não geram resultados, visto que são etapas destinadas a inicializar o sistema, que utilizam dados históricos da organização.

A evolução da simulação pode ser melhor visualizada nas Figuras 6.13, 6.14 e 6.15. A Figura 6.13 apresenta a evolução dos fatores de risco ao longo da simulação. A Figura 6.14 apresenta a evolução das probabilidades obtidas através de inferência na Rede Bayesiana gerada e a Figura 6.15 apresenta as pontuações de impacto ao negócio.

O primeiro detalhe que deve-se atentar é que os resultados obtidos até a etapa sete da simulação são estáveis, ou seja, variam pouco. Essa característica é esperada, visto que até a etapa sete da simulação, apenas são acrescentados novos ativos no sistema. Estas etapas avaliam a capacidade do sistema em perceber novos ativos no ambiente.

A capacidade do sistema identificar novos ativos e modificar a Rede Bayesiana pode ser percebida a partir dos gráficos apresentados nas Figuras 6.13, 6.14 e 6.15. Nestes gráficos, é possível perceber que até a etapa sete, é incluído um novo ativo computacional. Esta capacidade também pode ser percebida através das Figuras 6.8, 6.9, 6.10, 6.11 e 6.12, que representam a evolução da Rede Bayesiana até a sétima da simulação.

Os grafos apresentados nas Figuras 6.8, 6.9, 6.10, 6.11 e 6.12 referenciam os ativos utilizando um identificador. Este identificador é relacionado aos ativos através da coluna **identificador** nas Tabelas 6.4, 6.5, 6.6, 6.7 e 6.8.

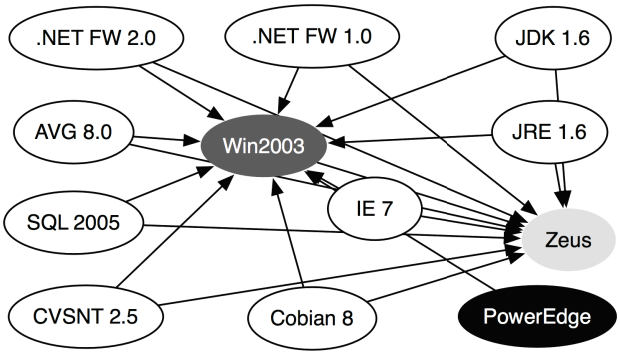


Figura 6.8: Grafo representando a Rede Bayesiana gerada até a etapa três da simulação.

A Figura 6.8 apresenta um grafo representando a forma da Rede Bayesiana gerada após a execução da terceira etapa da simulação. Neste grafo é possível perceber a existência de um ativo computacional, um ativo de *hardware*, um ativo de sistema operacional e nove ativos de aplicação.

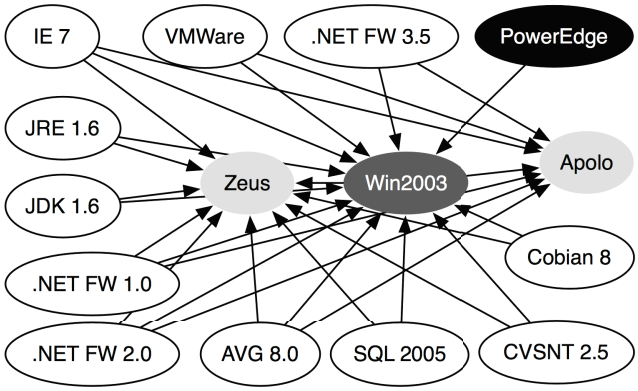


Figura 6.9: Grafo representando a Rede Bayesiana gerada até a etapa quatro da simulação.

A Figura 6.9 apresenta um grafo representando a forma da Rede Bayesiana gerada após a execução da quarta etapa da simulação. Neste grafo é possível perceber que o

número de ativos computacionais aumentou para dois e o número de ativos de aplicação aumentou para onze, no entanto, a quantidade de ativos de *hardware* e sistema operacional se manteve igual, visto que os dois ativos computacionais utilizam o mesmo tipo de *hardware* e sistema operacional.

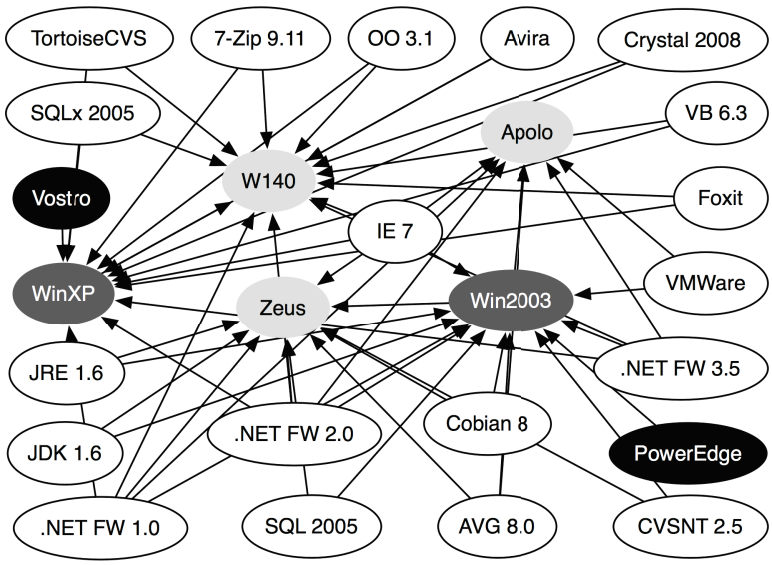


Figura 6.10: Grafo representando a Rede Bayesiana gerada até a etapa cinco da simulação.

A Figura 6.10 apresenta um grafo representando a forma da Rede Bayesiana gerada após a execução da quinta etapa da simulação. Neste grafo é possível perceber o surgimento de mais um ativo computacional, um ativo de *hardware* e um de sistema operacional e que a quantidade de ativos de aplicação subiu para dezenove.

A Figura 6.11 apresenta um grafo representando a forma da Rede Bayesiana gerada após a execução da sexta etapa da simulação. Neste grafo é possível perceber o surgimento de mais um ativo computacional, no entanto, a quantidade de ativos de *hardware*, de aplicação e sistema operacional se mantiveram.

A Figura 6.12 apresenta um grafo representando a forma da Rede Bayesiana gerada após a execução da sétima etapa da simulação. Neste grafo é possível perceber o surgimento de mais um ativo computacional e que a quantidade de ativos de aplicação subiu para vinte e quatro, no entanto, a quantidade de ativos de *hardware* se manteve.

Com o cadastro de um incidente envolvendo o aplicativo *a:7-zip:7-zip:9.11*, na etapa oito da simulação, pode-se perceber no gráfico que apresenta as probabilidades obtidas, apresentado na Figura 6.14, que a probabilidade do ativo computacional *w140* aumentou, visto que este ativo computacional utiliza o aplicativo *a:7-zip:7-zip:9.11*, que possui um relato de incidente ativo no sistema. O incidente simulado na etapa oito afeta indiretamente o ativo computacional *w126*. Isto ocorre pelo fato dos dois ativos

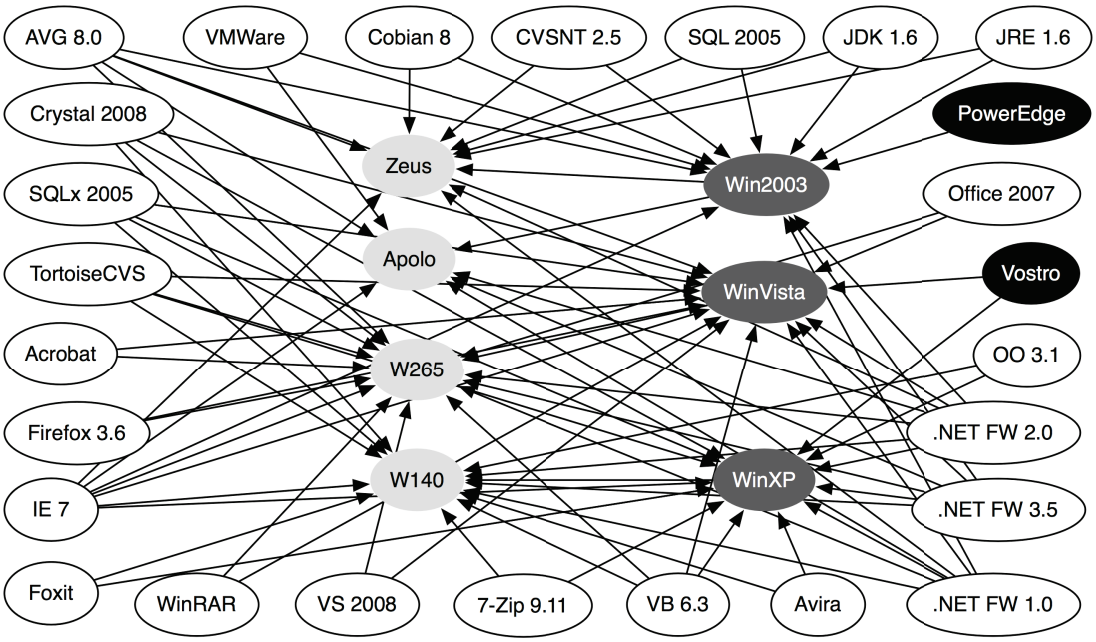


Figura 6.11: Grafo representando a Rede Bayesiana gerada até a etapa seis da simulação.

computacionais utilizarem a mesma versão de sistema operacional.

A mudança no risco dos ativos computacionais *w140* e *w126* pode ser percebida visualmente no gráfico apresentado na Figura 6.13. No entanto, esta etapa afeta praticamente todos os ativos computacionais existentes no sistema, como pode ser percebido na Tabela 6.9 comparando os fatores de risco da etapa sete e oito. O novo incidente afeta todos os ativos computacionais por que essa informação, além de ser utilizada como evidência no momento da inferência na Rede Bayesiana, é utilizada na estatística que determina o peso entre os nós da Rede Bayesiana.

A execução da etapa nove, acrescenta mais um incidente ao sistema, desta vez referente ao aplicativo *a:microsoft:ie:7*, que é utilizado por todos os ativos computacionais existentes no sistema. Nesta etapa pode-se perceber no gráfico apresentado na Figura 6.13 que o risco de todos os ativos computacionais aumenta. Mesmo o aplicativo sendo utilizado em todos os ativos computacionais, o risco não aumenta proporcionalmente para todos os ativos computacionais. Isso ocorre por dois motivos principais: (i) A inferência na Rede Bayesiana considera todos os relacionamentos existentes, sendo assim, a quantidade de aplicativos instalados em um ativo computacional afeta o resultado a inferência; e (ii) O cálculo do risco utiliza uma pontuação de impacto que pondera os requisitos de impacto do ativo computacional.

Com a remoção do aplicativo *a:microsoft:office:2007:sp1* do ativo computacional *w265*, na etapa dez, pode-se perceber que o risco referente ao ativo computacional

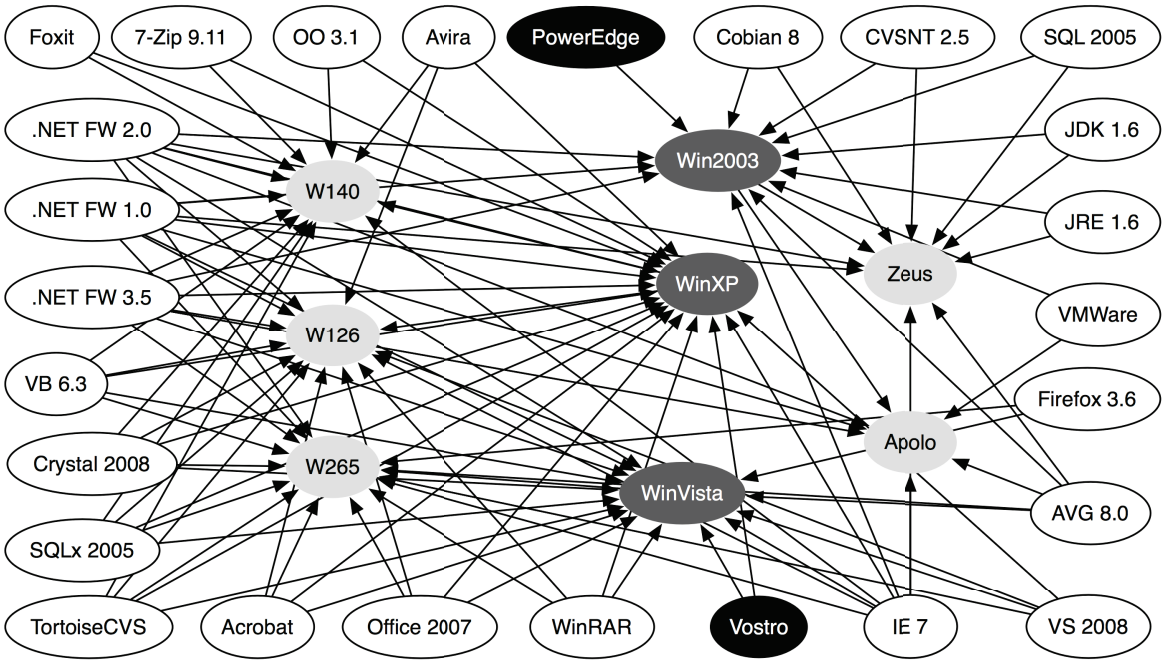


Figura 6.12: Grafo representando a Rede Bayesiana gerada até a etapa sete da simulação.

diminuiu, como pode-se perceber no gráfico apresentado na Figura 6.13.

A etapa onze realiza a troca do aplicativo *a:microsoft:ie:7* pelo *a:google:chrome:8.0.552.344* no ativo computacional *w140*, o que reflete diretamente no seu fator de risco. A queda no fator de risco do ativo computacional está ligada diretamente ao fato do aplicativo não possuir registro de vulnerabilidades na base de dados.

Na etapa doze, a situação do aplicativo *a:google:chrome:8.0.552.344* muda, visto que na etapa doze é realizada a importação dos registros do ano de 2011 da base de dados do NVD e esta possui registro de vulnerabilidade para este aplicativo. O efeito disto é o aumento do fator de risco relacionado ao ativo *w140*. Um fato que merece atenção nesta etapa é a redução do fator de risco referente ao ativo computacional *Zeus*. Isso ocorre porque a importação da base de vulnerabilidades da NVD traz novos registros e também as modificações em registros antigos. O que pode ter ocorrido neste caso, é que alguma vulnerabilidade estava relacionada com o sistema operacional que o ativo computacional utiliza e depois constatou-se que a vulnerabilidade não afetava este sistema operacional, atualizando então o registro. Como as informações sobre vulnerabilidades são amplamente utilizadas no método proposto, a atualização pode ser percebida inclusive no gráfico apresentado na Figura 6.15, que apresenta o impacto do ativo computacional no negócio da organização. Neste gráfico, pode-se perceber uma leve queda no impacto no negócio do ativo computacional *Zeus*. Esta situação também é importante para avaliar

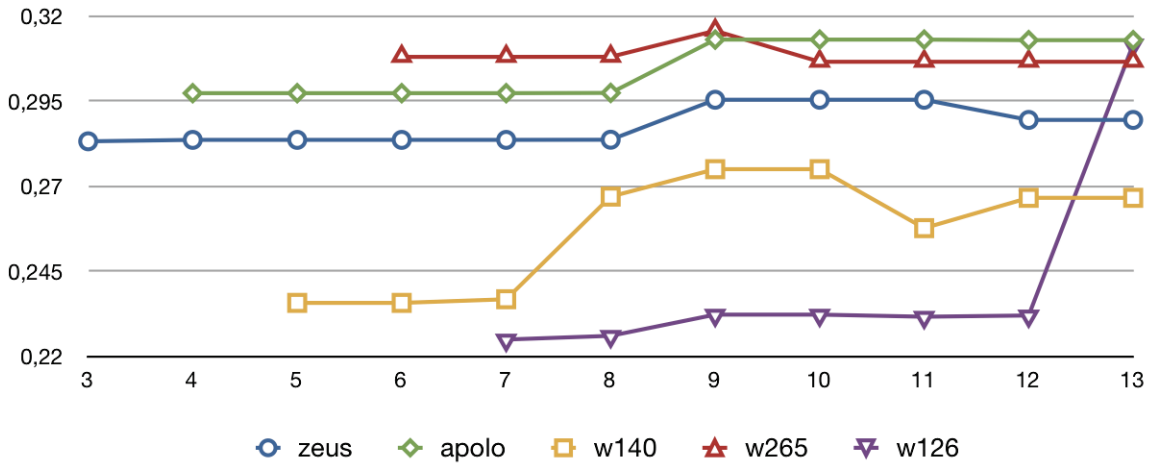


Figura 6.13: Evolução do risco durante a simulação

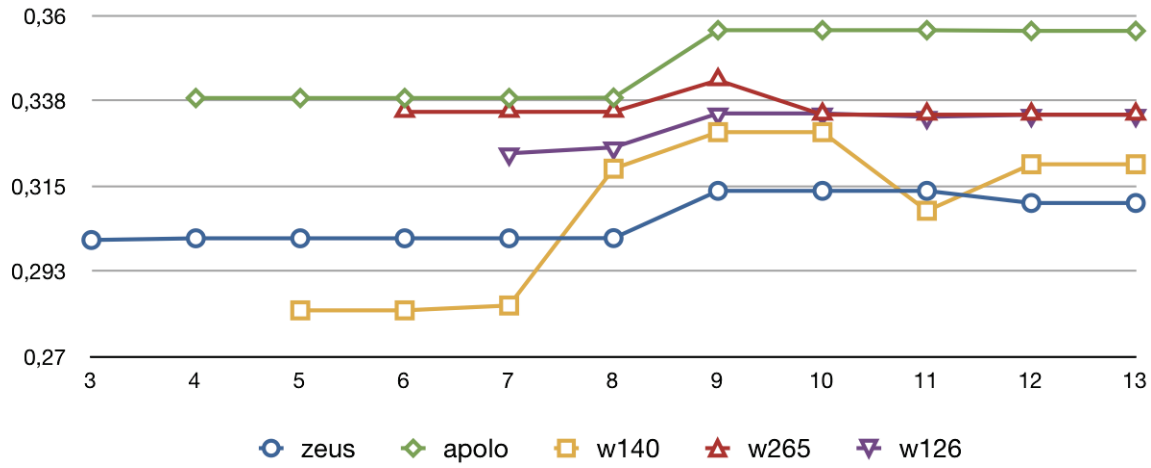


Figura 6.14: Evolução do impacto durante a simulação

a capacidade do sistema de perceber estas alterações no NVD.

A etapa treze, que simula a troca de função do ativo computacional *w126* para uma função que é crítica para o negócio da organização, é o último caso simulado na avaliação do sistema. Esta alteração reflete diretamente no impacto no ativo computacional perante o negócio, que pode ser percebido facilmente no gráfico apresentado na Figura 6.15. No momento que a etapa treze é executada, o impacto do ativo computacional *w126* e igual ao impacto de *Zeus*, que até o momento, era o ativo computacional mais importante. Com o aumento do impacto relacionado ao ativo computacional *w126*, o fator de risco atribuído a ele aumentou consideravelmente, chegando bem próximo do risco referente ao ativo computacional *Apolo*, que até a etapa doze era ativo com maior risco na organização.

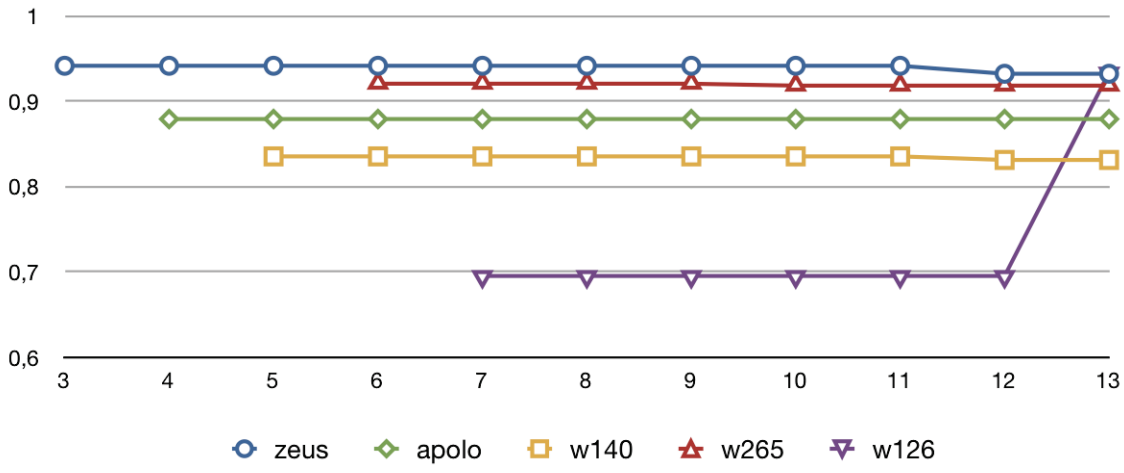


Figura 6.15: Evolução do risco durante a simulação

7 CONCLUSÃO

Ao longo do trabalho, é possível perceber o valor que a informação tem e o motivo pelo qual esta é considerada o ativo mais importante das organizações. As ameaças que envolvem a informação também foram apresentadas, assim como os métodos utilizados para diminuir os riscos relacionados aos ativos eletrônicos que compõem os processos de negócio da organização, momento este que torna clara a importância da Gestão de Risco para a Segurança da Informação, sendo esta para um Sistema de Gestão de Segurança da Informação ou para a Governança de Segurança da Informação.

Considerando a forma com que os ativos se relacionam para formar um processo de negócio, a possibilidade de utilização de Redes Bayesianas para a estimativa de risco foi identificada. Em um processo de negócio, os ativos possuem uma relação causal, em que um ativo tem um impacto direto sobre o bom funcionamento de outros ativos.

Um trabalho abordando a geração automática de Redes Bayesianas para estimar riscos de segurança foi publicado por Fenz e Hudec (2009). No entanto, este trabalho aborda riscos de processos/controles de segurança de uma forma genérica, como por exemplo, "sem auditorias de segurança", que apenas indica que não foram realizadas auditorias de segurança na organização e não o motivo específico que gerou a ameaça. A idéia de gerar uma Rede Bayesiana automaticamente prosposta por Fenz e Hudec (2009) foi agregada ao trabalho atual para melhor atender o objetivo proposto, que é auxiliar na atividade de análise de risco.

Um modelo para geração automática da Rede Bayesiana baseado em Sistemas Multiagentes foi formulado e um protótipo foi desenvolvimento para que o modelo pudesse ser avaliado.

Apesar não ser possível comparar os resultados obtidos com o modelo proposto com os modelos dos trabalhos relacionados, visto que o presente trabalho considera fatores que os trabalhos relacionados não consideram, como por exemplo, a relação causal entre os ativos da organização, foi possível constatar que os resultados obtidos refletem as situações que requerem atenção no dia-a-dia de profissionais de Segurança da Informação e que conseguem identificar alterações no cenário de segurança a partir de toda a relação existente entre ativos, vulnerabilidades e incidentes. A qualidade dos dados obtidos pode ser avaliada com a utilização do sistema em um ambiente real, no entanto, o protótipo requer otimizações de desempenho para que isto seja possível.

7.1 TRABALHOS FUTUROS

Durante o período de pesquisa e desenvolvimento do protótipo para avaliar o modelo proposto, alguns temas foram identificados como trabalhos futuros. Estes temas são apresentados na ordem cronológica que surgiram no presente trabalho.

O primeiro tema, que surgiu no início da pesquisa e não foi incorporado no trabalho por limitação de tempo é a utilização de informações referentes aos controles de segurança no cálculo do risco. Mesmo existindo vulnerabilidades em aplicativos em um ativo computacional, alguns controles/medidas de segurança podem e são utilizados para diminuir o risco sobre estes ativos. Sendo assim, para obter-se um resultado mais preciso, considerar estes controles é necessário. Para isso, pode-se utilizar como base os trabalhos de Ekelhart et al. (2009) e Fenz e Hudec (2009), que foram citados nos trabalhos relacionados.

Uma das limitações do protótipo desenvolvido é com relação ao número de nós gerados na Rede Bayesiana, visto que o algoritmo de inferência utilizado tem complexidade NP-Difícil, o que torna sua execução lenta. Uma pesquisa referente a este problema pode possibilitar a utilização do protótipo em ambientes mais complexos e que leva a resultados mais reais. A utilização de meganós, existente em Russel e Norvig (2004), chegou a ser considerada no presente trabalho, no entanto, o tempo necessário para pesquisa e desenvolvimento não era viável para o presente trabalho.

A integração do protótipo desenvolvido com ferramentas de inventario já existentes contribui com a eliminação do processo de atualização da base de dados de configuração de ativos, visto que este processo seria automático, como no sistema AURUM.

Visto que o presente trabalho possui uma limitação quanto a sua avaliação, a utilização do método proposto por Fenz e Ekelhart (2010) para verificar e avaliar a gestão de risco em segurança da informação é o trabalho futuro mais próximo. Este método não foi utilizado na avaliação do trabalho porque foi publicado no ano de 2010 e disponibilizado apenas em fevereiro de 2011.

BIBLIOGRAFIA

ABNT. *Tecnologia da informação - Técnicas de segurança - Sistema de gestão de segurança da informação - Requisitos*. http://www.abnt.org.br, 2005. Técnicas de segurança - Código de prática para a gestão da segurança da informação.

ABNT. *Gestão de risco - Vocabulário - Recomendações para uso em normas*. http://www.abnt.org.br, 2005b.

ABNT. *ABNT NBR ISO/IEC 27001*. http://www.abnt.org.br, 2006.

ABNT. *Tecnologia da informação - Técnicas de segurança - Gestão de risco de segurança da informação*. http://www.abnt.org.br, 2008.

ALVES, G. A. de O.; CAMARGO, L. F. R. da C.; ALMEIDA, A. C. R. D. de. Enterprise security governance. 2006.

BERGENTI, F.; GLEIZES, M.-P.; ZAMBONELLI, F. *Methodologies and Software Engeneering For Agent Systems*. : Frederico Bergenti and Marie-Pierre Gleizes and Franco Zambonelli, 2004.

BERNARDES, M. C.; MOREIRA, E. dos S. Um modelo para inclusção da governaça da segurança da informação no escopo da governança organizacional. *Instituto de Ciências Matemáticas e de Computação - ICMC*, 2006.

BRUÈRE, A. J.; SILVA, W. M. da; SANTOS, J. F. dos. Aspectos da governança corporativa de empresas listadas na bovespa: Um estudo exploratório sobre a composição e perfil dos conselhos de administração. UNISINOS, 2007.

BUTTNER, A.; ZIRING, N. *Common Plataform Enumeration (CPE) - Specification*. 2.2. ed. Março 2009.

CICCO, F. D. *Gestão de risco: Diretrizes para a implementação da AS/NZS 4360:2004*. : Risk Tecnologia Editora, 2005. (Risk Management).

DANTU, R.; KOLAN, P. Risk management using behavior based bayesian networks. *ISI (Springer-Verlag Berlin Heidelberg)*, p. 115–126, 2005.

DANTU, R.; KOLAN, P.; AKL, R.; LOPER, K. Classification of attributes and behavior on risk management using bayesian networks. *IEEE Xplorer*, 2007.

DARWICHE, A. *Modeling and Reasoning with Bayesian Networks*. : Cambridge University Press, 2009.

EKELHART, A.; NEUBAUER, T.; FENZ, S. Automated risk and utility management. In: *Proceedings of the 2009 Sixth International Conference on Information Technology: New Generations*. Washington, DC, USA: IEEE Computer Society, 2009. p. 393–398. ISBN 978-0-7695-3596-8.

FENZ, A. M. T. S.; HUDEC, M. Ontology-based generation of bayesian network. *International Conference on Complex, Intelligent and Software Intensive System (IEEE Computer Society)*, 2009.

FENZ, S.; EKELHART, A. Verification, validation, and evaluation in information security risk management. *IEEE Security and Privacy*, IEEE Computer Society, Los Alamitos, CA, USA, v. 99, n. PrePrints, 2010. ISSN 1540-7993.

FENZ, S.; NEUBAUER, T. How to determine threat probabilities using ontologies and bayesian networks. *CSIIRW'09 (ACM)*, April 2009.

FERBER, J.; GASSER, L. Intelligence artificiele distribuée. *The 11th Conference on Expert Systems and their Applications (Avignon'91), FR*, 1991.

ISG, I. S. G. *Information Security Governance: A Call To Action*. 2004.

ITGI. *Information Security Governance: Guidance for Boards of Directors and Executive Management*. 2nd edition. ed. 3701 Algonquin Road, Suite 1010 Rolling Meadows, IL 60008 USA: Printed in the United States of America, 2006. ISBN 1-933284-29-3. Disponível em: <www.itgi.org>.

JENSEN, F. V.; NIELSEN, T. D. *Baysian Networks an Decision Graphs*. second. : Springer Science+Business Media, LLC, 2007. (Information Science e Statistic).

LEE, E.; PARK, Y.; SHIN, J. G. Large engineering projetc risk management using a bayesian belief network. *Expert System with Applications*, Elsevier Ltd, 2008.

LINK, W. a.; BARKER, R. J. *Bayesian Inference with Ecologial Applications*. First. : Elsevier Ltd, 2010.

LOCKE, G.; GALLAGHER, P. D. *Managing Information Security Risk: Organization, Mission, and Information System View*. 800-39. ed. March 2011.

LUCAS, P. J.; GAAG, L. C. van der; ABU-HANNA, A. Bayesian networks in biomedicine and health-care. *Artificial Intelligence in Medicine (Elsevier)*, 2004.

MANSUR, R. *Governança de TI: metodologia, framework e melhores práticas*. : Brasport, 2007. ISBN 978-85-7452-322-4.

MELL, P.; SCARFONE, K.; ROMANOSKY, S. *A Complete Guide to the Common Vulnerability Scoring System Version 2.0*. 2007.

PADGHAM, L.; WINIKOFF, M. The prometheus metodology. April 2004.

PEARL, J. Plausible reasoning: From numerical probabilities to qualitative beliefs. In: *The 2nd Pacific Rim International Conference on Artificial Intelligence (PRICAI '92)*. 1992.

PEARL, J. *Probabilistic reasoning in intelligent systems : networks of plausible inference*. : Morgan Kaufmann, 1997. Paperback. ISBN 1558604790.

REZENDE, S. O. *Sistemas Inteligentes: Fundamentos e Aplicações*. : Solange Oliveira Rezende, 2005.

RUSSEL, S.; NORVIG, P. *Inteligência Artificial: tradução da segunda edição / Stuart Russel, Peter Norvig*. : Stuart Jonathan Russel, 2004.

SÊMOLA, M. *Gestão da segurança da informação: visão executiva da segurança da informaçãao : aplicada ao Security Officer / Marcos Sêmola e Módulo Security Solutions S.A.* : Campus, 2003.

SOLMS, B. von. *The Relationship between Corporate Governance, IT Governance and Information Security Governance*. 2007.

SOLMS, S. von; SOLMS, R. von. *Information Security Governance*. : Springer Science+Business Media, LLC, 2009.

VEIGA, L. R. da. *A controladoria como um mecanismo interno de Governança Corporativa: Um estudo envolvendo empresas de países relacionados aos modelas de Governança Corporativa anglo-saxão, alemão e latino-europeu*. 2006.

WEISS, G. *Multiagent System: A Modern Approachn to Distribuct Artificial Intelligence*. : Gerhard Weiss, 1999.

WOOLDRIDGE, M. J. *An Introduction to Multiagent System*. : Michael J. Wooldridge, 2001.