Universidade de Évora - Instituto de Investigação e Formação Avançada

Programa de Doutoramento em Informática

Tese de Doutoramento

Um modelo adaptativo para gestão de riscos de segurança em IoT

Luiz Otávio Botelho Lento

Orientador(es) | Pedro Patinho

Salvador Abreu

Évora 2024

Universidade de Évora - Instituto de Investigação e Formação Avançada

Programa de Doutoramento em Informática

Tese de Doutoramento

Um modelo adaptativo para gestão de riscos de segurança em IoT

Luiz Otávio Botelho Lento

Orientador(es) | Pedro Patinho

Salvador Abreu

Évora 2024

A tese de doutoramento foi objeto de apreciação e discussão pública pelo seguinte júri nomeado pelo Diretor do Instituto de Investigação e Formação Avançada:

Presidente	Luís Rato (Universidade de Évora)
Vogais	Jacques Robin (ESIEA - École Supérieure d'Informatique Électronique Automatique) Pedro Abílio Duarte de Medeiros (Universidade Nova de Lisboa) Pedro João Valente Dias Guerreiro (Universidade do Algarve) Rui Miguel Soares Silva (Instituto Politécnico de Beja) Salvador Abreu (Universidade de Évora)

Évora 2024

A todos aqueles que me inspiraram a ser o que sou.

Prefácio

"A tecnologia move o mundo." (Steve Jobs)

A pesquisa científica na área tecnológica pode proporcionar a resolução de problemáticas relevantes nos mais diversos setores da sociedade em que vivemos. Disto consciente, espero que este trabalho apresente contributos à elaboração de futuros trabalhos acadêmicos na área e, sobretudo, ao bem comum, como foi em outro projeto do qual tive a honra de fazer parte, a criação da primeira urna eletrônica no Brasil, um trabalho que colocou a tecnologia inteiramente a serviço do eleitorado brasileiro.

O Autor

Agradecimentos

AOS MEUS PAIS, referências maiores da minha vida, pelo amor e cuidado, que me nortearam no caminho do bem e da superação.

A MINHA ESPOSA, amiga e companheira de jornada, por estar ao meu lado e me apoiar em todos momentos, sendo fonte inspiradora por sua garra, determinação e competência; e pela parceria, afeição e lealdade demonstradas. Sem dúvida, a minha companheira de trincheira nesta vida.

AOS MEUS FILHOS, razão de toda uma existência, pela compreensão e manifestação de apoio em todos os momentos.

AOS PROFESSORES DOUTORES SALVADOR PINTO DE ABREU E PEDRO PATINHO, pela orientação pontual, abertura, generosidade e disponibilidade; por me colocar no caminho certo quando em alguns momentos eu me fiz perdido, me orientando e possibilitando que eu alcançasse as perspectivas de estudo; por ler, analisar e corrigir os artigos por ora trabalhados.

AOS DOCENTES DO CURSO DE DOUTORAMENTO EM INFORMÁTICA E INSTITUTO DE INVESTIGAÇÃO E FORMAÇÃO AVANÇADA (IIFA) DE ÉVORA, pelo tratamento como igual; pela aprendizagem e partilha.

AO POVO PORTUGUÊS, pelo acolhimento respeitoso e cordial.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Acrónimos

URL *Uniform Resource Locator*

IIFA Instituto de Investigação e Formação Avançada

ECT Escola de Ciências e Tecnologia

IoT *Internet of Things*

ISO/IEC *International Organization for Standardisation/International Electrotechnical Commission*

UE Universidade de Évora

ML Machine Learning

FLC Fuzzy Logic Control

CVE Common Vulnerabilities and Exposures

GNE Gestalt Nash Equilibrium

GAN Generative Adversarial Network

Sumário

Com o mundo mais dinâmico e competitivo, as pessoas vivem cada vez mais conectadas, respirando uma realidade cibernética em suas vidas. Os sistemas IoT também não escapam a esta realidade, estão espalhados por todo o mundo, disponibilizando uma vasta gama de serviços aos seus utilizadores, e aumentando a sua qualidade de vida, via utilização de dispositivos inteligentes. Paralelamente a toda esta tecnologia, os problemas relacionados à segurança da informação também fazem parte desta evolução dos sistemas IoT. Um grande problema nestes ambientes é garantir a segurança em todos os serviços e dispositivos, pois a diversidade de ameaças, em conjunto com a falta de preocupação da maioria de seus administradores e projetistas dos dispositivos, tornou o ambiente de rede IoT vulnerável.

Desta forma, este trabalho apresenta o RTRMM, um modelo de gerenciamento de riscos de segurança baseado em lógica para ambientes IoT, com novas estratégias para detectar, analisar e avaliar riscos, possibilitando assim prever riscos e, de forma objetiva gerenciá-los em tempo real, tornando o ambiente IoT mais confiável. O RTRMM faz uso de estratégia de probabilidade, tecnologias como Lógica Difusa (Fuzzy Logic), Cadeia de Markov, Teoria dos Jogos no desenvolvimento da sua estrutura lógica, e programação lógica para testar e validar suas funcionalidades.

Palavras chave: Gestão de Riscos, IoT, Reduzir Ameaças, Lógica Probabilística, Lógica Difusa

Abstract

An Adaptive Model to Reduce IoT Security Risks

With the world becoming more dynamic and competitive, people are increasingly connected, breathing a cyber reality into their lives. IoT systems also do not escape this reality, they are spread throughout the world, providing a wide range of services to their users, and increasing their quality of life, through the use of smart devices. In parallel with all this technology, problems related to information security are also part of this evolution of IoT systems. A major problem in IoT environments is ensuring security across all services and devices. The diversity of threats, together with the lack of concern on the part of most administrators and device designers, has made the IoT network environment vulnerable.

In this way, this work presents RTRMM, a logic-based security risk management model for IoT environments, with new strategies to detect, analyze and evaluate risks, thus making it possible to predict risks, and objectively manage them in real time. , making the IoT environment more reliable. RTRMM makes use of probability strategy, technologies such as Fuzzy Logic, Markov Chain, Game Theory in the development of its logical structure, and logic programming to test and validate its functionalities.

Keywords: Risk Management, IoT, Threat Mitigation, Probabilistic Logic, Fuzzy Logic

1

Introdução

Vivemos hoje em um mundo globalizado, interligado via redes de computadores, onde as distâncias desapareceram. Neste mundo virtual e cada vez mais competitivo, a informação é o maior patrimônio que uma organização pode ter para se manter competitiva, pois é considerada um ativo estratégico para o negócio da organização. A informação deve receber uma maior atenção no que tange ao seu tratamento, para ser protegida e gerenciada quanto ao seu armazenamento e tramitação, evitando que pessoas indesejadas a acessem.

A agilidade na composição e realização de negócios, bem como a necessidade de entregar produtos com qualidade e rapidez a clientes sempre mais exigentes, forçou as organizações a mudarem a sua estratégia. O uso de sistemas de informação e comunicação passou a constituir uma necessidade para atender a esta demanda, trazendo maior dinâmica aos processos do negócio da organização. Com a evolução da Internet, muitos recursos, tecnologias e oportunidades surgiram, desde a simples troca de informações até a possibilidade de transações financeiras e comerciais, gerenciamento remoto de ambientes, telemedicina, entre outros, tornando o IoT uma tecnologia real que melhora e facilita as redes de comunicação de dados.

A IoT, uma tecnologia de informação e comunicação, propicia uma maior rapidez e flexibilidade nas atividades executadas e nas tomadas de decisão. Ela é uma tecnologia que proporciona a conectividade de dispositivos inteligentes, sensores ou qualquer componente que possa ser considerado um computador (dispositivos IoT). A tecnologia IoT é uma realidade, que via conectividade de redes e capacidade computacional permite que os dispositivos interconectados gerem, troquem e processem dados com o mínimo de intervenção humana, oferecendo assim um amplo conjunto de funcionalidades ao universo computacional. [REC15]

Este universo de opções proporcionado pela tecnologia IoT, todavia, traz consigo um universo de preocupações quanto a segurança da informação. Considerada como uma das grandes preocupações de fornecedores e usuários, a segurança da informação tem um papel estratégico nessa tecnologia. Essa preocupação vai ao encontro da diversidade de tipos de dispositivos e fabricantes, dificultando a aplicação ou implantação de uma solução de segurança única, aplicável a qualquer solução/sistema IoT. Fatores como baixa capacidade de processamento e armazenamento são componentes que também dificultam que os dispositivos possam apresentar soluções embarcadas em seu software ou hardware [ARC18, RPIKR18]

Além destas variáveis apresentadas, a privacidade de dados é um fator crítico no uso de dispositivos IoT. Por exemplo, se pensar em sistemas de monitoramento de sistemas humanos vitais (batimentos cardíacos, comportamento cerebral, dentre outros) ou mesmo para monitoramento e aplicação de medicamentos, a privacidade dos dados trafegados e armazenados, composta pela confidencialidade e integridade destes é primordial para que problemas não sejam causados aos seus usuários (ex: alteração de dosagem de medicação). Isto porque a cada momento surgem novas ameaças à segurança da informação, proporcionando um universo de opções para os atacantes de plantão (hackers).

Apesar de todos estes componentes contrários à segurança da informação, os sistemas IoT vêm em uma crescente no universo computacional. Este crescimento se deve a diversos fatores, como prover uma maior economia de recursos, a evolução da Internet, a flexibilidade e diversidade de funcionalidades, de prover maior qualidade de vida aos seus usuários, entre vários outros fatores que poderiam ser citados. Os sistemas IoT evoluem em conformidade com as novas necessidades que o mundo moderno exige.

Sendo assim, este trabalho, preocupado com a segurança da informação em sistemas IoT, visa apresentar uma nova estratégia em relação a melhorar a segurança em sistema IoT. A estratégia é gerenciar os riscos, em sistemas IoT, de forma dinâmica, possibilitando o monitoramento e controle, destes em tempo real. Neste sentido, os riscos serão analisados e avaliados, dinamicamente, possibilitando que mecanismos de segurança possam ser aplicados antes que danos maiores sejam causados aos sistemas IoT.

A estratégia de gerenciar riscos em sistemas IoT foi escolhida por ser, talvez, a forma mais eficiente de reduzir os riscos, pois se tem condições de estar monitorando e controlando os riscos sempre que necessário em tempo real. Logo, a probabilidade de se prover um sistema IoT mais seguro e confiável para os seus usuários.

1.1 Fatores motivadores

Os sistemas IoT podem ser considerados sistemas cooperativos complexos [ZBSA12, Fil16]. São complexos, porque são constituídos por peças e dispositivos, distribuídos fisicamente, que funcionam em conjunto, via enlace físico e lógico, fornecendo novas habilidades/funcionalidades para seus usuários. A questão dos sistemas IoT serem cooperativos é porque interagem entre si de forma cooperativa, possibilitando que a solicitação de um usuário possa ser atendida. Por conseguinte, para manter seus serviços e dispositivos ativos, é necessário ter mecanismos de segurança habilitados para prevenir ou proteger contra ameaças capazes de interromper um serviço crítico, por exemplo.

Os sistemas IoT são, numa visão geral, sistemas que apresentam risco de ataques cibernéticos [Fil16]. Fornecer segurança aos sistemas IoT é um grande desafio, por serem sistemas normalmente compostos por dispositivos que possuem pouca capacidade de armazenamento, processamento e, também, baixo consumo de energia, dificultando embarcar mecanismos de segurança (IDS/IPS, criptografia forte), robustos, que possam melhorar, de forma mais eficaz, a proteção desses dispositivos e do sistema IoT como um todo.

Outros fatores também reforçam a importância da segurança da informação nos sistemas IoT, como o dinamismo e a diversidade na composição de sua topologia. Esta afirmação se deve ao fato de que a flexibilidade e a mobilidade dos dispositivos IoT são reais, principalmente em razão das funcionalidades que possuem. Outra questão é a diversidade de técnicas de ameaças utilizadas pelos invasores, que estão, algumas vezes, mais elaboradas e qualificadas tecnicamente, dificultando a criação de uma arquitetura padronizada de soluções de segurança.

As redes de comunicação de dados que provêm suporte à comunicação nos sistemas IoT, tem vulnerabilidades, como o tráfego de dados não criptografado ou a falta de identificação e autenticação adequada entre os dispositivos, podendo ser exploradas por atacantes que buscam coletar ou alterar informações. Os provedores de serviços de comunicação ou os servidores, onde os dados são armazenados, por exemplo, podem sofrer ataques de Negação de Serviço (DoS) ou mesmo invasões buscando violações de privacidade, tornando indisponíveis ou não confiáveis as funcionalidades que são oferecidas aos usuários de sistemas IoT. [ARC18, RPIKR18, Xe19].

Neste sentido, prover e garantir a confiança do usuário é fundamental para a utilização e evolução dos sistemas IoT, pois se ela não existir haverá um problema quanto aos relacionamentos de usuários e sistemas IoT. Desta forma, melhorar e garantir a privacidade dos dados de utilizadores de sistemas IoT é talvez uma das questões mais importantes a resolver em relação à confiança. A gestão de confiança é um dos componentes da gestão de riscos em sistemas IoT, pois ajuda a resolver o problema da privacidade dos dados, uma vez que irá gerir e garantir que a confiança e segurança do sistema IoT possa ser trabalhada até atingir níveis satisfatórios para seus usuários. Logo, parâmetros como nível de confiança, integridade, confidencialidade dos dados dos usuários devem ser gerenciados em tempo real, para permanecerem em níveis aceitáveis, permitindo ter conhecimento de quando e para quem as informações sobre um usuário devem e são divulgadas. [Ye14, Pe19]. Desta forma, é possível afirmar que não há privacidade de

dados sem que exista segurança adequada.

Implementar segurança em sistemas IoT é uma tarefa difícil e complexa. Alguns fatores aqui descritos reforçam esta afirmação e, por isso, incorporar novos mecanismos de segurança em dispositivos IoT ou encontrar uma solução de segurança única para resolver problemas de segurança não é uma solução simples. Uma forma de buscar melhorar a segurança em sistema IoT seria a de criar perímetros de segurança em torno de cada grupo de dispositivos e serviços. Cada perímetro estaria atrelado a regras de segurança, a fim de prover formas de reduzir os riscos ali existentes. Porém, é necessário conhecer as ameaças, vulnerabilidades e riscos para decidir quais medidas de segurança serão aplicadas. Uma análise e avaliação de riscos é fundamental, pois serão conhecidos os riscos e o nível de criticidade de cada um, possibilitando a criação e implantação das regras de segurança.

Por fim, mais um componente motivador pode ser citado para o desenvolvimento deste trabalho: a diversidade e quantidade de novas ameaças que surgem a todo instante. As ameaças no mundo cibernético são diversas e dinâmicas, principalmente em termos de técnica de ataque, o que torna um sistema IoT um alvo em potencial, sofrendo diversos ataques de diferentes locais. Reconhecer se um fluxo de dados tem ou não uma ameaça, em tempo real, é fundamental nesta "batalha"; analisar/avaliar se existe ou não uma ameaça e fornecer uma solução para minimizar um possível ataque, aplicando esta medida de segurança, em tempo real, é fator de sucesso neste processo de assegurar a segurança IoT. Portanto, em todas as situações apresentadas, percebe-se que a segurança em sistemas IoT é uma tarefa muito difícil, na qual uma única solução de segurança não resolverá o problema. Todavia, gerenciar riscos em tempo real talvez seja a forma mais desejável de melhorar a segurança em sistemas, dando assim aos seus usuários confiança na sua utilização.

1.2 Objetivos do Trabalho

O objetivo geral do trabalho é apresentar um modelo de gestão de riscos de segurança em tempo real para sistemas IoT, capaz de melhorar a confiabilidade destes.

Objetivos Específicos

1. Estudar e analisar as metodologias e modelos de gestão de riscos desenvolvidas e implementadas.
2. Apresentar um conjunto de novas estratégias para gerir riscos em sistemas IoT em tempo real.
3. Recorrer a ferramentas declarativas, como a Programação em Lógica.
4. Projetar e especificar um modelo de gerenciamento de riscos de segurança para sistema IoT.
5. Projetar e implantar uma arquitetura de detecção de ameaças em tempo real.

6. Projetar e prover uma solução que possibilite analisar e avaliar os riscos em sistemas IoT.

7. Projetar uma arquitetura de solução que possibilite selecionar o controle de segurança apropriado para reduzir o risco em sistemas com componentes IoT.

1.3 Metodologia de Pesquisa

A metodologia teve por base numa pesquisa exploratória, que tem como objetivo explorar um tema não muito explorado ou mesmo investigado. adotando uma abordagem qualitativa como método de melhor entendimento do problema a ser pesquisado.

A abordagem qualitativa se preocupou com a qualidade dos dados pesquisados, e a técnica de pesquisa, com o intuito de compreender as necessidades e comportamentos relacionados a gestão de riscos em sistemas IoT.

Para isso, foram obtidas informações sobre a área pesquisada, formulação de problemas e hipóteses, no qual o método de pesquisa adotado foi bibliográfico, se valendo de artigos acadêmicos, normas internacionais e livros sobre o tema.

1.4 Organização da Tese

O trabalho teve início via introdução, apresentando o problema a ser resolvido, suas características, como pode ser resolvido e os fatores motivadores que levaram o projeto a ser desenvolvido.

Na sequência, foram apresentados os objetivos gerais e específicos, que nortearam o desenvolvimento do trabalho, a metodologia aplicada, e o formato do trabalho, que compreende o esquema conceptual e a estrutura seguidos na investigação.

A dissertação está organizada em duas partes. A primeira, fundamentação teórica, possui o enquadramento relevante ao desenvolvimento do modelo apresentado, no qual foi realizada uma pesquisa bibliográfica. Neste capítulo, também foram realizadas comparações de técnicas que poderiam ser aplicadas ao trabalho.

A segunda parte é dividida em dois capítulos. O primeiro apresenta e especifica o RTRMM. Este capítulo apresenta o modelo funcional, discorrendo as suas características e como elas foram especificadas. Este capítulo também inclui uma ilustração por exemplos.

O segundo capítulo teve como objetivo apresentar os modelos de gestão de riscos estudados, realizando uma comparação destes com o RTRMM, constituindo uma validação conceptual do modelo.

1.4.1 Desenho do Trabalho

Para dar uma resposta aos motivos que nortearam o desenvolvimento deste trabalho, apresentamos um mapa (figura 1.1) com o desenho das principais etapas do desenvolvimento do trabalho, com delineamento do processo investigativo, de modelação e especificação do RTRMM.

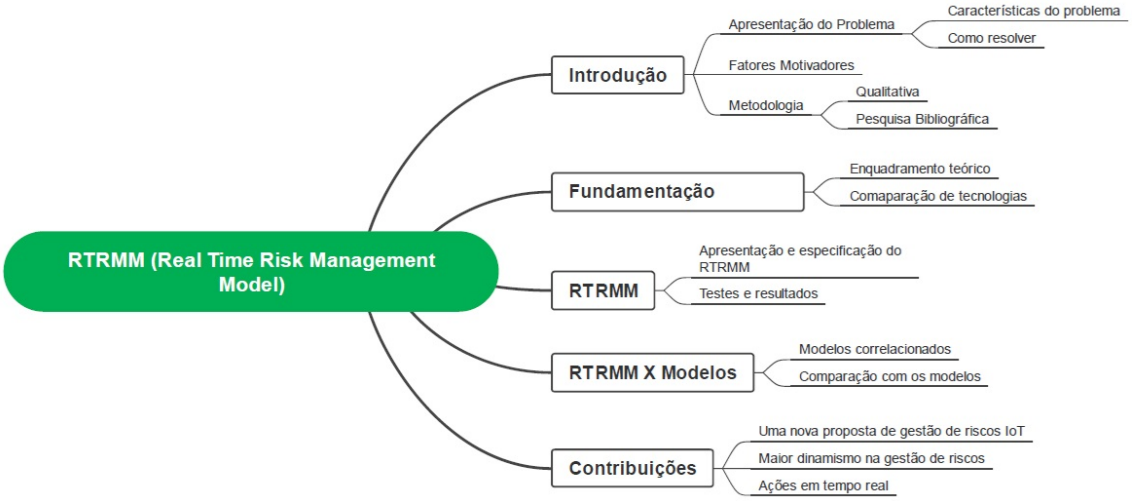

Figura 1.1: Desenho do Trabalho

Este trabalho deu origem a três artigos publicados em conferência com revisão por pares, em fases diferentes de seu desenvolvimento. As referências bibliográficas são:

1. Combining IoT Risk Management with Logic Programming - 2023 18th Iberian Conference on Information Systems and Technologies (CISTI) [LAP23a]
2. Um Modelo Declarativo para Gestão de Riscos em IoT - XXIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSEG2023) [LAP23b]
3. A Logic-Based Model to Reduce IoT Security Risks - 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) [LPA24]

2

Segurança da Informação

Neste capítulo apresentamos os conceitos de segurança da informação, um dos requisitos básicos para o desenvolvimento deste trabalho. O capítulo está divido em seis seções que tratam os conceitos de segurança da informação; a relação de processos de negócio com a segurança da informação; quais são os componentes que fazem parte da segurança da informação na Organização; o que são riscos; a segurança em ambientes IoT; e uma abordagem para compor uma solução de segurança da informação.

2.1 O que é Segurança da Informação?

A grande maioria da humanidade está conectada ao mundo 24 horas, 7 dias da semana e 365 dias ao ano, pois a utilização de recursos computacionais agilizou e facilitou o seu dia a dia. O aparelho celular passou a fazer parte da indumentária da maioria das pessoas no mundo, a Internet passou a ser vista como o "oxigênio" para sobrevivência do mundo, dos negócios e da troca de informações.

Com esta integração, composta por uma capilaridade de vias de comunicação entre todos os pontos no mundo, é de esperar que vulnerabilidades de segurança apareçam em paralelo. O uso indiscriminado da Internet, e a falta de mecanismos de resposta às novas ameaças, além da evolução constante das tecnologias, são elementos que colaboram para que o mundo virtual não seja seguro. Em paralelo a todo este universo de facilidades, existe um mundo obscuro, cercado de ameaças e agentes mal intencionados, que tornaram a vida virtual um tanto quanto "perigosa", principalmente para utilizadores pouco técnicos.

A segurança da informação consiste em garantir que as informações (em qualquer formato – mídias eletrônicas, papel e até mesmo em conversações pessoais ou por telefone, redes de computadores) estejam protegidas contra o acesso de pessoas não autorizadas (confidencialidade); estejam sempre disponíveis quando necessárias; e, que sejam confiáveis (integridade – não corrompidas ou adulteradas por atos de pessoas mal intencionadas), a preocupação com a proteção de dados, e recursos computacionais é uma constante (figura 2.1 - segurança em seu contexto). Esta preocupação vai ao encontro das diversidades de ameaças e a sua crescente evolução, principalmente pela variedade de tecnologias ou falta de padronização de protocolos dos dispositivos.[Len18]

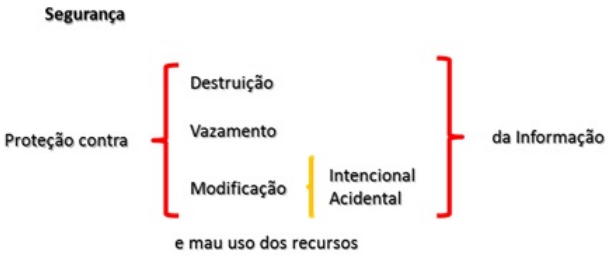

Figura 2.1: Segurança contextualizada

Toda esta diversidade torna a implantação e manutenção da segurança da informação no dia a dia em uma organização, mantendo os riscos a um nível aceitável, um desafio significativo. A segurança passou a ser fator crítico de sucesso para qualquer segmento de negócio, tornando a incessante busca pela eficácia dos controles, manutenção de soluções, e a necessidade de conscientização de usuários alguns dos fatores que contribuem para alcançar essa eficácia.

O sucesso de uma solução segurança está diretamente relacionada ao grau de confiança que é transmitido ao usuário, tendo em mente sempre que a informação é o legado de uma organização, e o seu vazamento e/ou roubo poderá causar danos irreparáveis.

A preocupação com a segurança na área de redes de computadores está diretamente relacionada à provisão e políticas aplicadas pelo seu gerente. Essas ações tem como objetivo monitorar e controlar o acesso não autorizado e o uso incorreto de recursos de rede, além de evitar qualquer tentativa de alterar, modificar ou mesmo negar acesso de serviços oferecidos pela rede de uma organização.

2.2 Segurança X Processos

A dinâmica dos processos de negócios nas organizações tem levado os gerentes a buscar qualidade no fornecimento de seus produtos. A tecnologia da informação (TI) hoje é parte deste contexto, sendo os recursos computacionais ferramentas de suporte aos processos de negócio, dando dinâmica e rapidez na sua execução. Em paralelo a este "leque" de opções dado pela TI, que disponibiliza informações de forma simples e rápida, vieram as vulnerabilidades e ameaças. Garantir, por exemplo, a confidencialidade, integridade e disponibilidade dessas informações passou a ser um processo integrante na gestão de TI.

A figura 2.2 apresenta parte da segurança da informação na relação entre a dinâmica dos processos de negócios nas organizações. Pode-se observar que não basta apenas prover segurança, mas gerir este processo de manter a segurança numa organização em nível aceitável para que os processos de negócio possam decorrer.

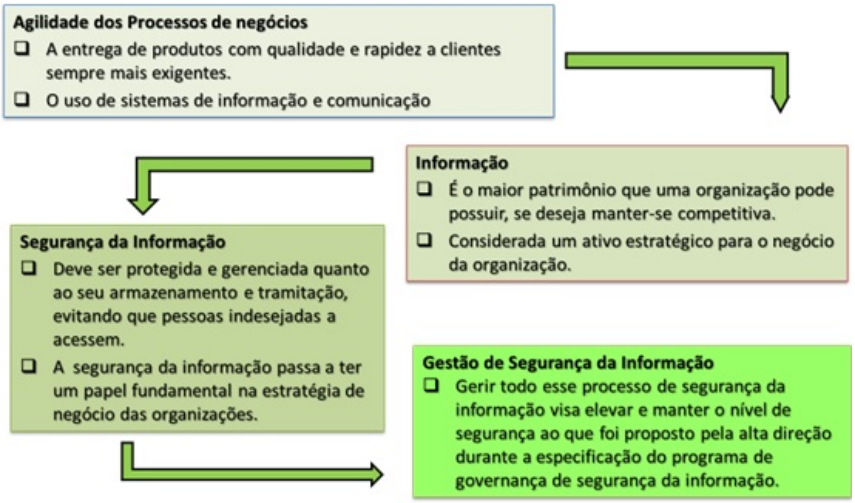

Figura 2.2: Gestão de Segurança no seu contexto - [Len12]

A segurança da informação busca sempre ser parceira quanto as necessidades dos processos de negócio da organização. Ela busca trazer benefícios como: melhorar a disponibilidade dos serviços de negócio ou de infra-estrutura e/ou reduzir-lhes os prejuízos e interrupções. A figura 2.3 apresenta um conjunto desses benefícios provenientes de implementar segurança com eficácia em uma organização.

2.3 Componentes da Segurança em uma Organização

A implementação da segurança da informação em uma organização tem como objetivo garantir as suas propriedades básicas como[Bis18]:

- Confidencialidade – a segurança de um sistema computacional não deve admitir que informações sejam descobertas por qualquer pessoa não autorizada. A

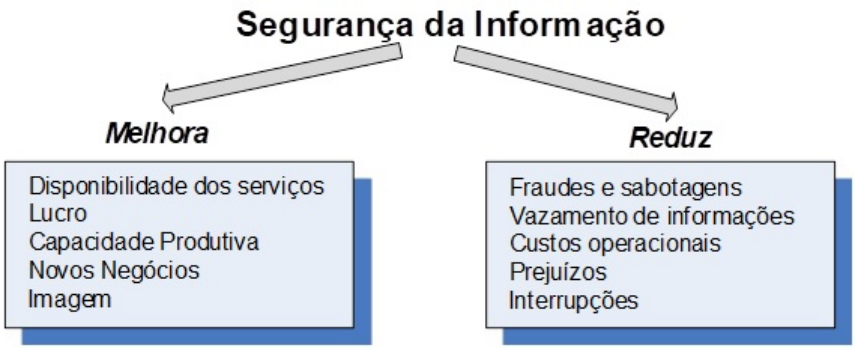

Figura 2.3: Benefícios da Segurança da informação - [Len18]

confidencialidade garante a privacidade das informações sensíveis em ambientes computacionais;

- Integridade – manter a integridade dos dados de um sistema significa que estes não terão as suas informações corrompidas, sejam de forma acidental ou intencional via pessoas não autorizadas. Logo, visa assegurar a não modificação indevida, seja acidental ou intencional, das informações e recursos naquele sistema;
- Disponibilidade – consiste na capacidade de manter disponível para os usuários do sistema os dispositivos de software e hardware. Logo, garante que as informações e recursos num sistema computacional estarão desimpedidos e prontos para serem usados, quando requisitados por pessoas autorizados.
- Autenticidade - garante a legitimidade de informações e de principais (origem e destino), sendo uma forma de verificar a veracidade da origem e do destino do dado.
- Não-repúdio - garante, em protocolos e transações, as proteções contra comportamentos omissos ou maliciosos onde participantes neguem ações realizadas.

Além dessas propriedades, outro componente importante da segurança da informação é o controle de acesso. Fator estratégico para o sucesso da segurança da informação, o controle de acesso é responsável por garantir o acesso aos recursos e serviços computacionais somente a usuários autorizados.

Quando se fala de segurança da informação, deve-se trabalhar em cima de três universos distintos:

1. Segurança física – pode ser alcançada através da criação de diversas barreiras físicas/controles físicos ao longo da propriedade física do negócio e das facilidades de processamento da informação.
2. Segurança lógica – consiste na habilidade de aplicar controles lógicos, isto é, barreiras lógicas/controles lógicos que impedem ou limitam o acesso a informação, que está em ambiente controlado, geralmente eletrônico, e que, de outro modo, ficaria exposta a alteração não autorizada por elemento mal intencionado.

3. Segurança de pessoal/organizacional – busca criar uma conscientização de segurança da informação dos colaboradores da organização fazendo uso de políticas e procedimentos. Além disso, gerenciar e prover treinamentos a colaboradores e terceiros quanto a segurança da informação.

A junção destes 3 universos viabiliza a composição de uma solução global de segurança da informação, possibilitando garantir as suas propriedades.

2.4 Riscos

Em qualquer transação virtual, acesso à Internet, existe risco, mas a grande questão é conhecê-lo quanto à criticidade para uma organização. A necessidade de analisar e avaliar os riscos existentes em uma organização é um fator crítico de sucesso para o seu negócio. O risco é simples de se entender e definir, conforme citado na ISO/IEC 27005:2022 [ISO22b] (talvez a definição mais simples e completa sobre risco), ele pode ser visto como a relação entre uma ameaça explorar uma vulnerabilidade de um ativo de informação e o impacto que esta ameaça irá causar ao sistema computacional. Uma outra forma de entender o que é risco, é conhecer a possibilidade de um ativo de informação estar sujeito a incidentes/fatores (vulnerabilidades, ameaças, ...) que possam vir a causar perdas ou danos ao sistema computacional [Len18].

Existem ainda outros entendimentos do que é risco, como o impacto negativo da ação de uma vulnerabilidade, considerando tanto a probabilidade e o impacto da ocorrência, isto é, risco é a função que relaciona a probabilidade de uma determinada ameaça explorar uma vulnerabilidade em potencial e o impacto resultante desse evento adverso sobre a organização[Sto02]. Outra definição interessante é a apresentada em [ISO04], quando são relacionados eventos com riscos e oportunidades. Os eventos podem ter um impacto negativo, positivo, ou ambos. Eventos com um impacto negativo representam riscos que podem impedir a criação de valor ou reduzir o valor existente. Eventos com impacto positivo podem compensar os impactos negativos ou representar oportunidades. As oportunidades são as possibilidades de um evento ocorrer e afetar positivamente a realização dos objetivos, apoiar a criação de valor e de manutenção.

Desta forma, ao se trabalhar com tecnologia da informação (TI) sempre existirão riscos, pois a TI é utilizada com o intuito de agregar valor aos processos de negócio. Logo, o risco do negócio está diretamente ligado ao uso, operação e adequação dos recursos aos seus processos, criando desafios na busca pela empresa de suas metas e objetivos estratégicos.

A figura 2.4 apresenta uma dinâmica de como o risco se relaciona com os ativos de TI, representados pelos dispositivos IoT e os mecanismos de segurança da informação, que vai ao encontro do risco que qualquer processo de negócio de uma organização pode apresentar. Por exemplo, todo dispositivo IoT tem o seu valor para o sistema como um todo. Logo, quanto maior este valor mais importante ele é, e mais riscos ele possui. Todo dispositivo IoT possui vulnerabilidades que junto com as ameaças ao seu redor também aumentam os riscos. Em contrapartida, existem os mecanismos de segurança

que visam reduzir os riscos, que são definidos os tipos e como serão implementados em função dos riscos existentes.

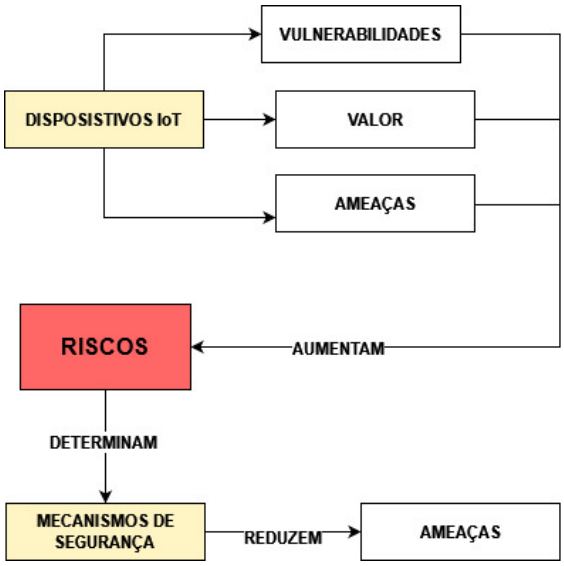

Figura 2.4: Dinâmica dos Riscos

2.5 Segurança em IoT

A Internet das Coisas (IoT) pode ser interpretada como um conjunto de dispositivos interligados em rede trocando dados, normalmente controlados por sistemas ou mesmo servidores. O IoT possibilitou uma maior interação social e econômica em diversas áreas de atuação, como medicina, mídia social entre outros [RRM+19]. Todavia, toda esta tecnologia e estes recursos vêm acompanhados de um conjunto de riscos, definidos como a possibilidade de um ativo estar sujeito a vulnerabilidades e incidentes (ameaças explorando essas vulnerabilidades), comprometendo a continuidade das atividades de uma organização (impacto) [ISO17].

Se for pensar que muitos negócios estão adotando novas formas de tecnologia (como IoT, Blockchain e Inteligência Artificial) para aumentar a eficiência e eficácia de seus serviços, os riscos acompanham estas tecnologias. Isto porque estas tecnologias estão expostas a riscos técnicos (falha de hardware ou software), éticos (dano não intencional causado por uma ação involuntária), de segurança (vulnerabilidades que podem ser exploradas) e de privacidade (dados causando danos a um usuário ou organização) [RRM+19].

Implantar segurança em um sistema IoT também é um processo árduo e contínuo. IoT é uma tecnologia a mais que cresce de forma rápida e dinâmica, que manipula informações que precisam ser protegidas. Logo, o processo é árduo porque é composto por um conjunto amplo de etapas, e contínua porque a sua gestão (monitoramento e controle) deve ser realizado periodicamente em função das mudanças que o sistema pode sofrer, além das constantes novas ameaças que se fazem presentes no universo cibernético. Para facilitar a tarefa de implantação de segurança, entender os processos de negócio da organização é fundamental, pois este entendimento facilita a tomada de

decisão sobre quais controles de segurança serão aplicados, bem como a forma mais adequada de serem implementados, a fim de reduzir os riscos que um dispositivo IoT pode sofrer. Desta forma, se o entendimento adequado dos processos de negócio for insuficiente, possíveis tomadas erradas de decisão levarão à implantação de controles de segurança inadequados ou controles implantados de forma errônea, causando brechas de proteção de um perímetro de segurança [Len18].

2.5.1 Problemas e Desafios de Segurança em IoT

Os desafios e problemas em sistemas IoT são, em parte, semelhantes à maioria dos sistemas computacionais existentes, diferindo em suas especificidades, como limitação de memória, ou capacidade de processamento, entre outros. Sistemas IoT representam uma grande diversidade de tecnologias interconectadas, comunicando e compartilhando dados continuamente. Logo, riscos de segurança são criados em torno deste universo, podendo causar sérios problemas aos seus usuários, como aqueles que fazem uso de dispositivos que armazenam ou informam dados médicos vitais. Imagina-se, hipoteticamente, uma situação no qual um remédio (droga) aplicado em um paciente, controlado por um dispositivo IoT, em um sistema de cuidados médicos (Healthcare system), e este dispositivo recebe dados alterados porque foram manipulados de forma maliciosa (ou não, por erro de transmissão). Quais danos este paciente poderá vir a sofrer? Como monitorar e controlar para que este tipo de dano não venha a acontecer? Um outro bom exemplo seria um dispositivo IoT encaminhando dados pessoais, manipulados ou mesmo errados, para serem processados em um sistema computacional. Quais danos podem vir a causar para este usuário IoT. Poderiam ser citados vários outros exemplos, como câmeras invadidas e imagens sendo redirecionadas ou mesmo alteradas; o universo de riscos é imensurável. [SAH+21]

Pode-se afirmar em sistemas IoT que é fundamental a confidencialidade, integridade e autenticidade dos dados trocados, processados ou mesmo armazenados por importantes dispositivos em sistemas IoT. Questões como disponibilidade e tempo de resposta também são aspectos para determinados sistemas IoT, que também devem ser tratados. Vale lembrar que um ambiente IoT possui uma grande diversidade de dispositivos e tecnologias de comunicação em seu domínio, o que traz dificuldade para seus projetistas e seus usuários, pois pode dificultar ainda mais a busca por uma solução de segurança IoT [HMW+16].

Desafios

O desejo de manter flexibilidade, escalabilidade e conexões de rede em tempo real pela Internet acarreta uma série de vulnerabilidades de dados. Proteger, por meio soluções de software os aplicativos em IoT, é totalmente viável, mas nem tudo é viável, alguns desafios são uma realidade. Os principais desafios são:

I - Proteger os dispositivos e protocolos de comunicação proprietários

Monitorar todos os dispositivos e protocolos de comunicação é uma tarefa complexa, principalmente porque cada fabricante possui a sua própria padronização. Com isso existe um problema complicado de ser minimizado, pois sem uma padronização pré-estabelecida do comportamento de execução de um dispositivo, e dos protocolos de comunicação, detectar se um comportamento é normal ou anormal é uma tarefa difícil, aumentando o desafio de detecção de possíveis ataques cibernéticos. Em [RPIKR18], é citado que a arquitetura aberta dos sistemas IoT aumenta ainda mais o desafio de proteger dispositivos, pois aumenta a diversidade de funcionalidades e arquiteturas.

II - Diversidade de dispositivos e fabricantes

Prover segurança em sistemas IoT é um desafio e exige muito esforço, principalmente se o ambiente IoT for composto por uma diversidade de dispositivos e fabricantes. Atender a esta diversidade de dispositivos demanda também uma diversidade de soluções de segurança integradas para atender às necessidades do sistema. Por exemplo, se a IoT for aplicada à indústria como em [HMW+16], os desafios estão relacionados à sua cadeia de suprimentos, desde a necessidade de armazenar uma grande quantidade de dados até os sistemas de controle. Sendo assim, conhecer bem os objetivos do negócio, analisar e avaliar os riscos existentes é fundamental, porque atender a demanda dos principais componentes reduz o esforço e atende, em parte, a demanda de segurança do sistema. Este, talvez, seja o maior desafio que se tenha para resolver.

Apresentados alguns desafios, fica a pergunta: como resolver ou só mesmo mitigar os riscos que um ambiente IoT pode sofrer? Atualmente, existem países e organizações discutindo sobre como resolver esse problema. A pesquisa de soluções de segurança IoT é uma realidade, sendo um problema a ser enfrentado [Xe19]. Uma estratégia que pode ser adotada, é pensar como ordenar a tomada de decisão para se obter a melhor opção de solução de segurança em uma determinada situação. A Figura 2.5 apresenta as etapas de como uma solução de segurança IoT pode ser alcançada [Len18].

Figura 2.5: Etapas de um Solução de Segurança - [Len18]

1. O que proteger – definir o que é crítico a ser protegido.

   (a) Quais segmentos de negócios do ambiente IoT são críticos.

(b) Para o segmento crítico de IoT, defina quais serviços são necessários para que ele não pare de funcionar.

(c) Determinar que ativos (Hardware, Software, Peopleware, Informação, ...) são essenciais para que os serviços do segmento IoT permaneçam operacionais.

2. Como proteger – avaliar e analisar os riscos que cada segmento crítico de IoT apresenta. Inicialmente definir e analisar quais os possíveis riscos, para depois avaliar, determinando a prioridade de tratamento para cada um.

   Conhecidos os riscos, existem condições para determinar como proteger o ambiente IoT. Portanto, o planejamento, a implantação e implementação de soluções de segurança será o próximo passo.

3. Garantir Proteção – após a implantação da solução de segurança é necessário um monitoramento constante para adequar a solução de segurança a novas ameaças. Além disso, é preciso alertar os utilizadores e equipe de TI, e realizar uma auditoria periódica do que foi implementado com o que foi planejado.

Como dito em [MS19], mais desafios podem ser apresentados, como privacidade do usuário, no qual a proteção de dados é fundamental quando trocados pela Internet, garantindo a confidencialidade e integridade, além da privacidade do usuário e/ou dispositivos que estão manipulando esses dados. A utilização de protocolos de criptografia é uma solução plenamente aplicável, mas existe o problema de recursos de armazenamento e processamento dos dispositivos IoT. Este problema vem acarretar a redução do tamanho da chave a ser processada, ou mesmo a utilização de algoritmos criptográficos mais robustos.

O desafio de dispositivos e/ou usuários de identificar e autenticar pode ser resolvido com a implantação de protocolos de criptografia e a estratégia de gerenciamento de identidade[Osm14]. O desafio de prover segurança ponta a ponta, entre dois dispositivos, pode ser resolvido com algoritmos de criptografia (simétricos ou assimétricos [BP02]), garantindo a confidencialidade e integridade dos dados transmitidos às informações em um dispositivo, antes que as informações sejam enviadas para outro dispositivo e descriptografadas. Todavia, vale a pena ressaltar que o nível de segurança desses dados, por exemplo, dependem do algoritmo e do tamanho da chave utilizados.

Portanto, implantar uma solução de segurança não é apenas escolher controles e/ou mecanismos de segurança (Firewalls, IDS/IPS), mas também é uma questão de estratégia e equilíbrio entre as possíveis ameaças existentes e a eficiência do Negócio. Desta forma, é certo dizer que para sistema IoT melhorar a segurança, é necessário realizar uma gestão eficaz de riscos existentes.

Problemas

Um problema pode ser definido como a causa de um incidente.

Apesar de existirem soluções de segurança para sistemas IoT, problemas de segurança considerados críticos persistem, como garantir a confidencialidade e integridade de dados. A problemática da dependência da capacidade dos dispositivos IoT limitam o tamanho das chaves e algoritmos criptográficos. Mecanismos de segurança tradicionais também não são eficientes, eles sofrem do mesmo problema: recursos computacionais. Por exemplo, a utilização de mecanismos de defesa de ponto (HIDS - Host Intrusion Detection System, firewall de defesa de ponto) em cada dispositivo IoT é um problema, pois estes mecanismos necessitam de recursos computacionais para processamento e armazenamento de informações, e energia suficiente para alimentar esses mecanismos [Zha14].

Uma outra forma de tentar reduzir os problemas de segurança seria adotar a estratégia de defesa de área. Todavia, esta estratégia como opção de solução de segurança para ambiente IoT não é a mais adequada. Normalmente, as soluções que adotam esta estratégia não visam garantir a privacidade, confidencialidade e integridade dos dados entre dispositivos IoT: a área de defesa preocupa-se com o controle de entrada e saída de dados (autenticação de dados, ...) e não com o tráfego ponto a ponto entre dispositivos IoT.

Um fator importante que influencia os problemas de implantação de soluções de segurança para ambiente IoT é a diversidade de ameaças existentes no universo cibernético. Em [SRJ18], são exibidos um conjunto dessas ameaças como: clonagem de dispositivos IoT, ataque *man-in-the-middle* durante autenticação e autorização entre dois dispositivos, ameaça à privacidade contra dados sensíveis, ataque de negação de serviço (DoS), espionagem, quebra de sigilo, entre outros.

Desta forma, conclui-se que ter uma solução que minimize o máximo de ameaças a um nível considerado satisfatório é um tanto complicado e difícil. É importante lembrar que um ou outro dispositivo pode estar vinculado a uma ou mais ameaças, necessitando de mais controles de segurança para que possa operar de forma satisfatória, o que torna ainda mais difícil encontrar soluções de segurança adequadas.

2.6 Uma Visão para uma Solução de Segurança

Para implantar uma solução de segurança é necessário conhecer como funcionam os processos de negócio da organização. Conhecendo os processos, tem-se conhecimento da importância do sistema IoT para a organização e de quais dispositivos são fundamentais para o seu funcionamento. Conhecendo estes fatores, têm-se condições de determinar o que deve ser protegido.

O objetivo é identificar maneiras de mitigar vulnerabilidades reduzindo os riscos de uma ameaça ser concretizada. Para isso, o conhecimento das vulnerabilidades que um dispositivo possui, além das possíveis ameaças que ele poderá sofrer são fundamentais. Conhecidas as vulnerabilidades e ameaças, pode-se explorar as estratégias e controles de segurança.

Por exemplo, o uso de técnicas de autenticação pode minimizar a probabilidade de

um ataque *man-in-the-middle* entre dispositivos IoT. As técnicas de controle de acesso podem estar minimizando o ataque à privacidade de dados em ambientes IoT. Outra técnica, tão importante quanto as citadas, é o uso de criptografia entre as comunicações dos dispositivos IoT, reduzindo a chance de ataques que prejudiquem a confidencialidade e integridade, ataques de espionagem [ARC18].

Implantada a estratégia e os controles de segurança, deve-se realizar uma auditoria do que foi implementado com o que foi planejado. Sendo assim, a implantação de uma solução de segurança não significa que o processo está encerrado, principalmente porque a dinâmica do mundo virtual provê novas ameaças periodicamente, obrigando a uma revisão esporádica de todo o processo de segurança adotado. O processo de gerenciamento de riscos é uma metodologia que engloba todas as etapas mencionadas, e, realizando esse processo de forma dinâmica e em tempo real, proporcionará a análise e avaliação de segurança [Has19].

2.7 Considerações

Este capítulo teve como objetivo apresentar os conceitos básicos em segurança da informação, tratando a sua relação com os processos de negócio da organização, os conceitos de riscos e segurança em IoT e, por fim, uma abordagem para se desenvolver uma solução de segurança da informação.

No próximo capítulo serão abordados os conceitos de gestão de riscos de segurança da informação e a sua relação com sistemas IoT.

3

Gerenciamento de Riscos de Segurança da Informação

Neste capítulo apresentamos o que é gerenciamento de riscos de segurança da informação, suas características, funcionalidades e como aplicá-lo em sistemas IoT. O capítulo está dividido em 5 seções que tratam dos fundamentos sobre gerenciamento de riscos; características de gerenciamento reativo e proativo; gerenciamento de riscos em sistemas IoT; modelos de gerenciamento de riscos e trabalhos relacionados ao tema do capítulo.

3.1 Fundamentos

A proteção da informação contra os riscos que ela possui no dia a dia é fundamental, principalmente pela crescente evolução tecnológica e a dependência das empresas da TI, tornando a segurança da informação fator estratégico para qualquer organização. Gerir riscos, isto é, manter os riscos a um nível considerado aceitável, visa elevar e manter

o nível de segurança. Porém, vale a pena ressaltar que, para gerir riscos, é necessário conhecer quais riscos e como eles se comportam dentro da organização.

O gerenciamento de riscos de segurança visa planejar, executar, monitorar e realizar a manutenção da melhor, de forma contínua, com o objetivo de sempre otimizar ou manter o nível de riscos aceitável em uma organização. A gestão de riscos de segurança deve possuir um planejamento adequado, e a execução e evolução em conformidade com as necessidades do negócio da organização. Desta forma, para que a gestão de riscos possa ajudar no negócio, ela deverá ser governada pelos níveis hierárquicos que compõem a organização, agregando valor estratégico e, com isso, sendo compreendida e reconhecida como fundamental para o negócio da organização [Len15].

Qualquer processo de gestão de riscos é basicamente composto por 3 etapas:

1. Análise / Avaliação de riscos - esta etapa vai desde identificar, calcular e avaliar os riscos;

2. Tratar os riscos - nesta etapa são especificados quais riscos deverão ser tratados, bem como a prioridade de tratamento. São definidos quais ´mecanismos de segurança serão aplicados e como serão implementados;

3. Manutenção - não basta somente implementar os mecanismos de segurança, existe a necessidade de realizar um monitoramento constante a fim de verificar a eficiência dos controles implementados.

A Importância do Gerenciamento de Riscos

O gerenciamento de riscos em sistemas IoT possui grande importância dentro das organizações, pois tem como objetivo estar alinhado e direcionado aos diversos desafios proporcionados pelo seu negócio. Isto porque a gestão de riscos agrega valores e elementos estratégicos para evolução e extensão prática da organização.

O gerenciamento de riscos é estratégico para o negócio. Ele é estratégico via monitoramento e controle destes riscos, tem-se uma visão se os mecanismos de segurança apicados estão em conformidade com as necessidades de segurança ou não. Desta forma, existe a possibilidade de se reduzir ou manter os riscos em um nível considerado aceitável ao negócio da organização, minimizando os prejuízos financeiros que possam ser decorrentes de uma interrupção do sistema de informação, vazamento ou roubo de dados.

Os incidentes de segurança normalmente comprometem de forma direta ou indiretamente as operações de negócio e, por conseguinte, a imagem junto a clientes, parceiros e fornecedores. Logo, uma visão holística de gestão de segurança seria a busca de resultados compatíveis às necessidades do negócio de uma organização [Len15].

3.2 Gestão de Riscos Reativo / Proativo

Apesar do gerenciamento de riscos de segurança da informação possuir o mesmo objetivo de identificar, analisar/avaliar e mitigar os riscos, pode ser diferenciado em 2 (duas) formas de atuação: gerenciamento de risco reativo e gerenciamento de riscos proativo.

Gerenciamento Reativo

O gerenciamento reativo é ativado após o acontecer um incidente de segurança da informação ou mesmo se algum evento ocorra. O seu objetivo é reduzir o impacto dos eventos e as ameaças já acorridas [KJ14]. Sendo assim, o incidente é analisado e avaliado, medidas de contenção, por exemplo, são tomadas com o objetivo de limitar a expansão do ataque pela organização, evitando que os processos de negócio sofram grandes perdas e prejuízos.

Após uma análise e avaliação detalhada das brechas de segurança, mecanismos de segurança são implementados para reduzir a probabilidade de que aquela ameaça explore, com sucesso, tais brechas em outras ocasiões. Todos os incidentes são registrados, criando um histórico de incidentes, objetivando que medidas de segurança, preventivas, possam ser aplicadas, monitoradas e controladas para garantir que os riscos se mantenham em um nível aceitável.

Gerenciamento Proativo

O gerenciamento de riscos proativo, ao contrário do gerenciamento reativo, tem como objetivo prever e identificar os riscos antes que um incidente aconteça. Desta forma, medidas de segurança poderão ser acionadas antes que o incidente realmente se concretize. Para que isso seja possível, normalmente o gerenciamento proativo se baseia em uma base de informações, previamente cadastradas [KJ14]. Esta base de conhecimento pode estar relacionada ao comportamento dos ativos do sistema computacional que constituem, numa visão macro, o perfil do sistema computacional, ou parte específica dele (ex: rede de comunicação de dados).

Uma forma simples de melhor entender é analisando a capacidade de processamento e armazenamento da memória principal de um dispositivo de rede (roteador, switch). Após o monitoramento durante um ciclo de vida da organização (por exemplo: 30 dias), tem-se o perfil de processamento e memória deste dispositivo. O monitoramento deste possibilita analisar se o comportamento está dentro do perfil estabelecido. Caso as informações fujam ao perfil determinado, pode o dispositivo estar sofrendo um incidente de segurança. Desta forma, mecanismos ou mesmo ações focadas em segurança da informação são acionadas pela equipe de gestão de segurança da Organização. Com essas ações de antecipação ao problema, espera-se mitigar a probabilidade de que o incidente de segurança da informação se concretize.

A postura proativa do gerenciamento de riscos é uma forma de se antecipar ao problema, evitar ou mitigar danos aos processos de negócio de uma organização. Todavia, a capacidade e destreza de suas ferramentas de gestão de riscos, bem como a da sua

equipe são fundamentais para o sucesso deste formato de gerenciamento.

3.3 Gerenciamento de Riscos de Segurança em Sistemas IoT

Uma estratégia interessante a ser adotada para melhorar a segurança em sistemas IoT é gerenciar os seus riscos. Porém, devido à amplitude da possibilidade da existência de riscos em uma organização, é necessário contextualizar o que é risco. Existem algumas definições sobre risco, mas em [ISO17] risco é definido como a possibilidade de um ativo estar sujeito a vulnerabilidades e incidentes (ameaças explorando essas vulnerabilidades), comprometendo a continuidade das atividades de uma organização (impacto).

Gerenciar riscos consiste em monitorar e controlar um sistema IoT de forma que seja possível manter os riscos a um nível considerado adequado. Para isso, é necessário o conhecimento das atividades que o ativo irá exercer no sistema IoT, as suas vulnerabilidades, o seu ativo, grau de importância para o sistema IoT, analisar e avaliar se os controles aplicados conseguem manter o nível do risco no nível considerado aceitável para sua operação [ISO22b]. Desta forma, a gestão de riscos visa agregar valores e elementos estratégicos para evolução e extensão prática da organização.

Desafios do Gerenciamento de Riscos em Sistemas IoT

O Gerenciamento de riscos de segurança já é um grande desafio por si só, em sistemas IoT este desafio ainda pode ser maior. Os desafios de gerenciar sistemas IoT podem ser divididos em 3 (três) grupos que englobam: quais dispositivos IoT ou dados de usuários que serão protegidos (o que proteger); como realiza a proteção desses componentes (como proteger); e após aplicar os mecanismos de segurança nos dispositivos IoT, como garantir e melhorar (manter/melhorar) a proteção existente.

O que proteger é talvez o mais complexo e, quem sabe, o mais difícil de ser sobrepujado. Imagine se tivesse que proteger todos os dispositivos IoT e os dados do sistemas IoT, o quanto seria custoso. Logo, deve-se identificar os diversos segmentos do sistema IoT, como base no negócio que está sendo aplicado, analisar de forma detalhada cada um deles e escolher quais dispositivos IoT serão considerados mais críticos para que o negócio da organização.

Selecionado o que será protegido, deve-se analisar a necessidade de dos dispositivos IoT para verificar se os mecanismos de segurança (controles de segurança) existentes e aqueles a serem aplicados, serão eficientes, bem como os recursos necessários para a implementação e manutenção desses mecanismos de segurança. Este processo envolve várias etapas, dentre elas o processo de análise e avaliação dos riscos existentes, no qual serão analisados e avaliados os controles de segurança existentes e a necessidade de novos controles.

Por fim, manter/melhorar os mecanismos existentes consiste em estar sempre monitorando e controlando os riscos no sistema IoT. Isto envolve um processo contínuo, onde são verificados se os controles implementados estão em conformidade com o que foi

planejado. Outrossim, no caso de introdução ou mudança de novas tecnologias, uma nova análise e avaliação de riscos deverá ser realizada, com o objetivo de verificar se os controles de segurança ainda são eficientes e a necessidade de se implantar novos controles.

3.3.1 Quantificação dos Riscos

I - Medir os riscos

Os riscos, para serem medidos, necessitam estar claros e definidos [Rai22]. Conhecer bem os processos de negócio e os ativos necessários para que estes processos funcionem é básico. Isto porque pode-se conhecer as vulnerabilidades, e por conseguinte, os riscos que cada ativo possui. A análise de riscos pode ser realizada de várias formas, de acordo com o nível de criticidade dos ativos, conhecimento das vulnerabilidades e os incidentes.

Existem duas estratégias de análise de riscos, qualitativa ou quantitativa, podendo ser aplicadas de forma individual ou de forma combinada. A análise qualitativa é normalmente utilizada para se ter uma visão ampla do nível de risco, revelando os principais riscos. A análise de risco qualitativa pode ser usada:

- Como uma atividade de triagem inicial para identificar os riscos que necessitam uma análise mais detalhada;
- Utilizada para tomada de decisões;
- Quando valores numéricos ou recursos são inadequados.

A análise quantitativa de riscos é mais específica, podendo complementar a análise sobre os principais riscos. A análise quantitativa de riscos, normalmente, faz uso de históricos de incidentes, tendo como vantagem a relação existente entre os objetivos a serem alcançados e as preocupações de segurança da informação da organização. Todavia, tem como desvantagem que dados relacionados a fatos e auditoria normalmente não estão disponíveis, dando uma ilusão de valor e precisão da avaliação de risco. Qual das estratégias seria a mais adequada? Qualquer uma pode ser utilizada, mas a análise qualitativa é menos complexa e custosa de realizar que a quantitativa [ISO22b].

II - Métricas Confiáveis

A medição de riscos requer métricas que, às vezes, podem não ser as mais adequadas, causando impactos negativos nos cálculos de risco. As abordagens de medição podem ser simplificadas, manipuladas; serem confiáveis de maneiras inesperadas ou mesmo não levar em conta diferenças e contextos em um incidente. Isto porque os danos de um incidente podem ser variados, conforme os processos de negócio e as comunidades afetadas. Logo, a métrica deverá satisfazer todos esses elementos, para que se aproxime de uma realidade [Rai22].

Quando se trabalha com análise de risco qualitativa, pode-se utilizar como métrica uma escala de atributos de qualificação para medir o impacto de um incidente e a probabi-

lidade desta ameaça ser efetivada. Essa escala é um tanto subjetiva, e a quantidade dos níveis de medição podem variar, por exemplo: baixo, médio ou alto. A vantagem de se utilizar a análise qualitativa é a facilidade de compreensão, em função da métrica adotada, por toda a equipe envolvida, e a desvantagem é a dependência da escolha subjetiva da escala (ex: muito baixo, baixo, médio, alto e crítico).

A escala adotada nessa métrica é ajustada conforme as diferentes situações presentes e os diferentes riscos existentes. Essas escalas podem ser adaptadas ou ajustadas para atender às circunstâncias e diferentes descrições podem ser usadas para diferentes riscos.

Na análise de risco quantitativa a métrica é baseada em valores numéricos, tanto para medir o impacto de um incidente quanto para a probabilidade desta ameaça ser efetivada. A qualidade da análise depende da precisão e abrangência dos valores numéricos e da validade dos modelos utilizados.

A forma como o impacto e a probabilidade são expressos e as formas como são combinados para fornecer um nível de risco variam de acordo com o seu tipo e a finalidade para a qual o resultado da avaliação de risco será usado. Uma função pode representar essa combinação, onde a incerteza e a variação do valor do impacto e da probabilidade devem ser consideradas na análise e comunicadas de forma eficaz [ISO22b].

III - Risco em diferentes estágios do ciclo de vida

Alguns riscos podem estar latentes em um determinado momento, podendo aumentar conforme a evidência da situação. Desta forma, o monitoramento constante das atividades é fundamental para analisar se a probabilidade de uma ameaça, ou mesmo outra ameaça se tornar efetiva aumentou ou mesmo reduziu. Para isso, o estabelecimento de um perfil de comportamento do ativo durante a sua atividade é fundamental. A análise pode ser realizada com base em comparações com este perfil que possibilita uma verificação constante de mudança, possibilitando estabelecer um novo valor para o risco.

3.4 Modelos de Gestão de Riscos em Segurança da Informação

Durante a realização do desenvolvimento do RTRMM, algumas abordagens de gestão de riscos foram pesquisadas e analisadas, com o objetivo de se ter uma base para a sua composição e, dentre as diversas abordagens, as seguintes foram selecionadas:

3.4.1 OCTAVE - Operationally Critical Threat, Asset, and Vulnerability Evaluation

O Octave [Alb01] é uma abordagem para avaliações de risco de segurança da informação, baseada em ativos, isto é, as tomadas de decisão são com base nos riscos aos ativos de informações. A abordagem é dividida em 3 (três) etapas :

1. Determinar quais ameaças baseadas nos ativos - a análise determina quais ativos são mais importantes para a organização (ativos críticos) e identifica o que está sendo feito atualmente para proteger esses ativos;

2. Identificar as vulnerabilidades – identifica e analisa os principais componentes operacionais (ativos) do sistema visando identificar as vulnerabilidades que podem ser exploradas;

3. Desenvolver planos e estratégias de segurança – identificar os riscos para os ativos críticos da organização, decidindo quais ações deverão ser realizadas.

3.4.2 Risk COBIT

O COBIT 5 [Ahm17] é uma abordagem de gerenciamento de riscos no domínio da TI e, além de exercer atividades de governança e gerenciamento de TI corporativa. O COBIT 5 pode ser aplicado sob duas visões:

- Função de risco - especifica o que é necessário para uma empresa construir e manter atividades de governança e gerenciamento de riscos;

- Gerenciamento de riscos—especifica o processo de gerenciamento de riscos no qual são executadas as funções de identificar, analisar, tratar e informar os riscos.

O COBIT 5 tem a otimização do risco como seu principal objetivo, e a governança e o gerenciamento de riscos como parte da governança e gerenciamento geral da TI (Tecnologia da Informação) corporativa. Na governança de riscos existe um processo que visa garantir que o risco corporativo não exceda um valor considerado aceitável. O processo de gestão de riscos procura integrar a gestão de riscos corporativos relacionados à TI com a gestão de riscos da empresa como um todo. Sendo assim, as organizações podem obter resultados: uma estratégia de gerenciamento de risco; um plano de comunicação de gerenciamento de risco; e requisitos financeiros e orçamentários para responder e mitigar o risco.

3.4.3 National Institute of Standards and Technology NIST 800-30

A norma NIST 800-30 [Bla12] é responsável por prover orientação para a realização de avaliações de risco de sistemas e organizações federais de informação nos EUA. As avaliações de risco, realizadas em todos os níveis da hierarquia de gerenciamento de risco, fazem parte de um processo geral de gerenciamento de risco – fornecendo aos líderes/executivos seniores as informações necessárias para determinar os cursos de ação

apropriados em resposta aos riscos identificados. A NIST fornece uma direção para a realização das etapas do processo de avaliação de risco.

Preparação para a avaliação de risco - tem como objetivo estabelecer um contexto para a avaliação de risco, por exemplo, identificar as informações organizacionais sobre políticas e requisitos para conduzir avaliações de risco, metodologias de avaliação específicas a serem empregadas, procedimentos para selecionar fatores de risco a serem considerados, escopo das avaliações, rigor das análises, ...

Realizar a avaliação de risco - tem como objetivo produzir uma lista de riscos de segurança da informação que podem ser priorizados por nível de risco e usados para informar as decisões de resposta. Para atingir esse objetivo, são analisados ameaças, vulnerabilidades, impactos e probabilidades, além da incerteza associada ao processo de avaliação de riscos. Esta etapa tem como abranger, de forma adequada, todas as ameaças definidas durante a etapa de preparação.

Comunicar e compartilhar as informações de avaliação de risco - esta etapa tem como objetivo garantir que os tomadores de decisão em toda a organização tenham as informações apropriadas relacionadas ao risco e necessárias para informar e orientar as decisões de risco. Para isso deverão ser comunicados os resultados da avaliação de risco; e compartilhada as informações desenvolvidas na execução da avaliação de risco, para apoiar as atividades de gestão de outros riscos.

3.4.4 ISO 27005 (Tecnologia da informação — Técnicas de segurança — Gerenciamento de riscos de segurança da informação)

A norma ISO 27005 [ISO22b] tem como objetivo fornecer um conjunto de diretrizes para o gerenciamento de riscos de segurança da informação. Trabalha com uma abordagem sistemática para o gerenciamento de riscos, no qual são identificadas as necessidades organizacionais em relação aos requisitos de segurança da informação. A sua diretriz é prover um processo de gerenciamento de riscos dinâmico, sendo parte integrante de todas as atividades de gerenciamento de segurança da informação da organização.

A abordagem da ISO 27005 está dividida nas seguintes etapas:

I - Estabelecimento de contexto - todas as informações sobre a organização, relevantes para o estabelecimento do contexto de gerenciamento de riscos de segurança da informação, são necessárias. De posse dessas informações, são definidos os critérios básicos necessários para o gerenciamento de riscos de segurança da informação, o escopo de abrangência. É definido também o objetivo do gerenciamento de riscos de segurança da informação, pois isso afeta o processo geral e o estabelecimento do contexto em particular.

II - Avaliação de Risco – nesta etapa, os riscos serão identificados, quantificados ou descritos qualitativamente e priorizados em relação aos critérios de avaliação de riscos e

objetivos relevantes para a organização. Algumas etapas são executadas:

- identificar os riscos – identificar os ativos a serem protegidos; estabelecer o valor de cada; identificar as vulnerabilidades de cada ativo; e identificar as ameaças relacionadas a cada ativo;

- analisar os riscos – identificar os controles atrelados a cada ativo; estabelecer a probabilidade de cada ameaça; calcular o impacto para a organização de cada ativo; e calcular o risco;

- avaliar os riscos – estabelecer a prioridade de tratamento dos riscos calculados.

III - Tratar o Risco – esta etapa tem como objetivo estabelecer os controles para reduzir, reter, evitar ou compartilhar os riscos, além de criar um plano de tratamento de risco.

IV - Aceitação do Risco – todo o processo de tratamento de risco não é pleno, existe sempre um risco chamado de residual. Esse risco residual poderá ser aceito ou não, dependendo das regras estabelecidas na fase de definição de contexto. A decisão de aceitar os riscos e responsabilidades pela decisão deve ser tomada e registrada formalmente

V - Comunicar o risco – esta etapa tem como objetivo trocar e compartilhar informações sobre risco entre o responsável pela tomada de decisão e outras partes interessadas. As informações incluem, mas não se limitam à existência, natureza, forma, probabilidade, gravidade, tratamento e aceitabilidade dos riscos. A comunicação eficaz entre as partes interessadas é importante, pois pode ter um impacto significativo nas decisões que precisam ser tomadas.

A tabela 3.1 apresenta um breve resumo das abordagens estudadas e analisadas durante todo o processo de definição e escolha de um modelo mais adequado para ser utilizado em ambientes IoT. Se for analisar de forma direta, nenhuma das abordagens são direcionadas a sistemas IoT. Todavia, as abordagens OCTAVE e COBIT5, por serem mais voltadas a aplicação em ambientes corporativos, não se aplica aos sistemas IoT, devido ao grau de complexidade para implanatar as suas funcionalidades. As abordagens especificadas nas normas NIST 800-30 e ISO 27005 são as mais próximas à aplicação em sistemas IoT, por possuírem processos dinâmicos e que podem ser aplicados a dispositivos de forma independente e como um todo, o que faz dessas abordagens as mais adequadas a sistemas IoT.

3.5 Trabalhos Relacionados à Gestão de Riscos

Durante o desenvolvimento do RTRMM foram analisados alguns trabalhos relacionados à pesquisa desenvolvida.

Nome	Característica	Processos	Risco
OCTAVE	- avaliação de risco baseado em ativos; - as tomadas de decisão baseadas nos riscos dos ativos	- determinar ameaças baseadas nos ativos. - identifica vulnerabilidades - desenvolver planos e estratégias de segurança	Qualitativo
Risk COBIT	- uma abordagem de gerenciamento de riscos no domínio de TI	- Função de risco - Gerenciamento de riscos	Qualitativo
NIST 800-30	- prover orientação para a realização de avaliações de risco de sistemas e organizações - estrutura baseada no uso extensivo de siglas.	- caracterizar o sistema - identificar ameaças e vulnerabilidades - cálculo de probabilidades - análise de impacto - cálculo do risco - recomendações e resultados	Qualitativo
ISO 27005	- fornecer um conjunto de diretrizes para o gerenciamento de riscos de segurança da informação em uma organização	- estabelecer o contexto - identificar os riscos -analisar e avaliar os riscos - Tratar os riscos - Aceitar e comunicar os riscos	Qualitativo

Tabela 3.1: Resumo das Abordagens

3.5.1 Gerenciamento de riscos em ambientes IoT

Em [MS19], é realizada uma pesquisa que faz uso de um software inteligente para encontrar e reduzir vulnerabilidades de segurança. Este software possui uma base de dados CVE (enumeração de vulnerabilidades comuns) e verifica se a vulnerabilidade detectada está disponível nesta base de dados.

Caso a vulnerabilidade exista na base dados, é atribuído um nível para analisar o grau de criticidade da vulnerabilidade, baixo, médio, alto e crítico, para definir a relevância dos riscos. É verificado se existe um caminho de mitigação desta vulnerabilidade e, caso exista ela é aplicada.

3.5.2 Modelo de gerenciamento de risco de segurança IoT para ambientes de saúde com segurança

Em [ZAHS19], trata-se de IoT na área da saúde, existe uma preocupação com a segurança e privacidade dos dados relacionados, compartilhados por meio de dispositivos IoT. É necessário garantir que os dados dos pacientes não vazem, possibilitando a utilização dessas informações de forma inadequada como : a compra de medicamentos e a alteração de informações em dispositivos IoT que realizam o monitoramento de pacientes.

Um modelo de gerenciamento de riscos de segurança IoT para profissionais de saúde foi apresentado como um princípio básico para o uso seguro de sistemas IoT em um ambiente de saúde. O modelo tem três partes: Healthcare IoT Risk Management, alinhamento HPIA (Hospital Performance Indicator for Accountability) e fases de implementação do COBIT 5. A primeira parte do modelo é o COBIT IoT Risk Management, formulado com base na categoria de risco IoT, como dados e aplicativos, gerenciamento de alterações do usuário, segurança e privacidade, ambiente físico, e infra-estrutura. A outra parte são as áreas de gerenciamento do COBIT 5, que consistem em gerenciamento de leis, de infraestrutura, de operações, de portfólio, fornecedores terceirizados e fornecedores e gerenciamento de mercado.

O modelo, por fim, incorpora as categorias HPIA, que são processos internos de negócios, foco no cliente, satisfação do funcionário, aprendizado e crescimento, gerenciamento financeiro e de escritório e, finalmente, suporte ambiental. O modelo proposto é implementado em etapas, provenientes do COBIT5, que vão desde o escopo de abrangência, necessidade, objetivos, execução e avaliação do que foi realizado.

3.5.3 Gerenciamento estratégico interdependente de riscos de segurança com racionalidade limitada na IoT

Em [Che19], se cada dispositivo IoT necessitar alocar recursos para proteger seus aplicativos, e se a política de segurança relacionada a um dispositivo venha causar um impacto no risco de segurança dos nós que estão conectados a ele, é complicado realizar um gerência efetiva de segurança. Sendo assim, é proposto um modelo cujo desafio é gerenciar o risco de segurança de forma descentralizada, no qual uma tomada de decisão pode ser modelada conforme a situação. Desta forma, cada usuário aloca estrategicamente seus recursos para proteger os dispositivos e os riscos são reduzidos quando seus dispositivos estão conectados a vizinhos com alto nível de segurança.

O modelo de gestão é baseado na decisão de segurança do usuário. A decisão usa o conceito de risco seguro onde o usuário percebe, observando um número selecionado de nós, qual é a sua decisão de segurança. A IoT possui conexões complexas e massivas e os usuários não podem estar cientes das políticas de segurança adotadas por todos os seus vizinhos conectados. Assim, o modelo de gestão precisa levar em conta a racionalidade limitada dos dispositivos.

A estrutura de gestão é um vetor que representa a estrutura de observação de cada

usuário de IoT. Um vetor de cognição mais esparso representa um usuário com capacidade de cognição mais fraca, e ele observa menos comportamentos de outros usuários ao decidir sua estratégia. No modelo de gestão racional, os usuários precisam tomar decisões de gerenciamento de segurança, bem como projetar suas redes de cognição de maneira holística. Os autores tomam essas decisões, utilizam o conceito de solução chamado equilíbrio de Nash (GNE) para capturar a formação da rede cognitiva e o gerenciamento de segurança sob a racionalidade limitada simultaneamente. O objetivo do GNE é fornecer um método quantitativo para entender o risco de IoT e fornecer políticas de gerenciamento de segurança tratáveis. O modelo de gerenciamento de segurança possui uma estrutura para avaliar os riscos de segurança IoT relacionados aos aplicativos IoT.

3.5.4 Modelo de gerenciamento de riscos de segurança IoT para o setor de saúde

O modelo descrito em [SBH+19] baseia-se em uma análise que foi feita junto a um hospital, no qual foram realizadas entrevistas e, por conseguinte, coletadas as suas necessidades. Foi usado como referência o modelo de gerenciamento de riscos do Decision-Making Trial and Evaluation Laboratory (DEMATEL), no qual quatro etapas foram aplicadas: definição de metas; avaliação de riscos; análise e cálculo de risco; e gestão de riscos.

O modelo mostrou-se adequado ao propósito que se fez, no qual cada uma das etapas das atividades realizadas atendeu às necessidades de gestão de segurança de hospitais.

3.5.5 Uma arquitetura segura para IoT com gerenciamento de riscos na cadeia de suprimentos

Em [HHV17] foi proposta uma arquitetura para reduzir as vulnerabilidades dos riscos de uma cadeia de suprimentos, aplicando técnicas de aprendizado de máquina, monitoramento de hardware criptográfico e coordenação de sistema distribuído para mitigar as consequências das ameaças. Esta arquitetura depende de ferramentas de reconhecimento de padrões para aprender e identificar o comportamento 'normal' e 'anormal' dos componentes da IoT sob hardware e falhas de software e ataques cibernéticos. Ela também é capaz de responder a possíveis invasões de segurança por meio de um sistema de transmissão de dados. Por fim, a arquitetura provê um sistema que requer uma resposta de verificação do operador para contestar ou aprovar as ações a serem tomadas.

A arquitetura visa inicialmente aprender os comportamentos normais e anormais de um dispositivo IoT. Utiliza uma estratégia de correlação cruzada no qual a classificação aprendida por máquina dos estados de saída de um dispositivo é correlacionada com uma aprovação ou rejeição interativa do operador no loop do estado aprendido por máquina. Utiliza um algoritmo de aprendizado de máquina que emprega uma rede neural profunda para análise do comportamento do sistema, consistindo em duas sub-redes: uma para aprender a representação de recursos de dados normais e uma segunda sub-rede para a

detecção de padrões de recursos anormais causados por tentativas de invasão, malware, bugs de código-fonte não detectados ou falhas de hardware.

3.6 Considerações

Este capítulo teve como objetivo apresentar os fundamentos de gerenciamento de riscos de segurança da informação, apresentando as suas duas formas de execução, reativa e proativa. Foram apresentados também alguns modelos de gestão de riscos e trabalhos relacionados ao tema abordado.

No próximo capítulo, será descrito o que é programação lógica, quais são as suas características, linguagens e, também, o que é programação probabilística.

4

Programação em Lógica

Neste capítulo apresentamos os fundamentos de programação em lógica, as características das linguagens PROLOG e ASP, bem como os fundamentos da programação probabilística e as linguagens PROBLOG e CPLINT.

4.1 Fundamentos de Programação em Lógica

A Lógica tem início com o filósofo grego ARISTÓTELES, cuja essência era a teoria do silogismo, uma estrutura básica de um argumento ou um raciocínio dedutivo. Aristóteles definiu o silogismo como a conclusão deduzida de premissas, a argumentação lógica perfeita, constituído de três proposições declarativas (duas premissas e uma conclusão) que se conectam de tal modo que, a partir das duas primeiras (as premissas), é possível deduzir uma conclusão.

Segundo Palazzo [Pal97], a lógica é a ciência do pensamento correto, isto é, ela pode ser considerada a ciência da verdade. A lógica, como tudo na ciência sofreu evolução,

sendo que a representação dos processos lógicos originários vem desde em Boole e de De Morgan, nos anos 1800 até os dias de hoje [Hit16].

Todavia, somente em 1974 o termo programação em lógica, via Robert Kowalski em [Kow74] apareceu, no qual ele representou um predicado lógico como uma linguagem de programação, em interpretações de implicações como declarações de procedimentos. Programas lógicos são utilizados para representar um determinado problema via a utilização de frases lógicas, conhecidas como cláusulas. Os programas lógicos também são uma base de dados, no qual regras são declaradas.

Logo, pode-se citar que um paradigma de programação em lógica é um conjunto de princípios e técnicas que orientam o projeto e a implementação de programas lógicos, no qual programa lógico pode ser visto como uma coleção de fatos e regras que descrevem os relacionamentos e propriedades das entidades, e uma linguagem de consulta que permite fazer perguntas e obter respostas do programa. Desta forma, uma paradigma de programação em lógica define a sintaxe e a semântica dos fatos, regras e consultas, bem como o mecanismo de inferência que deriva novos fatos e regras dos existentes [Pal97, Hit16].

Desta forma, pode-se afirmar que a programação em lógica é uma abordagem de programação bem trabalhada e sedimentada na comunidade computacional. Os programas lógicos possuem um formalismo natural (linguagem declarativa) para criar sistemas baseados em regras, além disso estes sistemas quando fazem uso de lógica probabilística da informação, os tornam ainda mais capazes [RS92]. Isto porque trabalhar com raciocínio probabilístico possibilita que decisões possam ser tomadas mesmo quando não existem evidências suficientes para determinar se um evento, por exemplo, aconteceu. Logo, trabalhar com programas de lógica probabilística possibilita que situações/regras possam ser analisadas e avaliadas quanto a sua probabilidade de ser real ou não.

Conforme cita em [Spi08], o contraste entre programação imperativa e declarativa pode ser constatado quando se deseja uma solução para um problema. Quando se trabalha com linguagem imperativa, o programa trabalha com instruções específicas na resolução do problema, isto é, representar no formato de código as instruções para que aquele problema possa ser resolvido.

Ao contrário da programação imperativa, a programação em lógica não utiliza de instruções explícitas para solucionar um problema, corresponde ao estilo de programação declarativa. A programação em lógica faz uso de estratégias para encontrar soluções para problemas que foram expressos como programas lógicos. Cada implementação de programação em lógica inclui uma estratégia como um componente central, como Prolog que usa uma estratégia conhecida como 'resolução SLD (Seletiva Linear para cláusula Definida) com pesquisa em profundidade', considerada de fácil de implementar e eficiente. Sendo assim, pode-se dizer que:

Programação Procedural / Imperativa (procedimental)

Se baseia em procedimentos, que são executados numa sequência, derivada da programação imperativa, mas com procedimentos, onde o seu código é organizado em blocos, que possibilita uma reutilização. Desta forma, trabalha com procedimentos que irão de forma direta executar tarefas, especificar o que será computado.

$$\text{Programa} = \text{Algoritmo} + \text{Estruturas de Dados}$$

Programação em Lógica

Não trabalha com procedimentos, a programação é realizada de forma declarativa, especificando o que deve ser computado ao invés de como deve ser computado.

$$\text{Algoritmo} = \text{Lógica} + \text{Controle}$$

$$\text{Programa} = \text{Lógica} + \text{Controle} + \text{Estruturas de Dados}$$

A programação em lógica possui algumas vantagens sobre outras técnicas de programação: ela é declarativa, expressiva e flexível. Ser declarativa significa que os programas lógicos se concentram no que o programa deve fazer, e não em como deve fazê-lo, tornando-os mais fáceis de compreender, manter e modificar. Ser expressivos, significa que os programas lógicos permite representar conceitos complexos e abstratos de forma resumida e natural, sendo adequados para domínios que envolvem raciocínio, representação de conhecimento e inteligência artificial. Os programas lógicos são flexíveis e podem trabalhar com diferentes modos de execução, como encadeamento direto, encadeamento reverso ou consulta interativa, possibilitando trabalhar com diferentes estratégias de resolução de problemas e necessidades do usuário[RS18].

Apesar de vantagens, a programação em lógica também tem as suas desvantagens ou desafios, que aumenta o seu custo computacional, não determinística e difícil de depurar. Necessita de bastante memória e boa capacidade de processamento para realizar inferência e pesquisa, o que o torna mais lento e menos escalável que os demais programas. Outro fator importante é que esta técnica pode produzir múltiplas ou nenhuma solução para uma determinada consulta, tornando-a menos previsível e confiável, bem como difícil localizar e identificar as origens dos erros ou resultados não esperados, tornando-os mais difíceis de testar e verificar do que outros programas.

Segundo Fabrizzio [Rig19, RS18], existem perspectivas que integram programação em lógica com probabilidade. Essas perspectivas são divididas e classificadas em semântica de distribuição (Distribution Semantics - DS) [Sat95] e construção de modelo de base de conhecimento (Knowledge Base Model Construction - KBMC).

Na semântica de distribuição, visa criar uma abordagem no qual quantifica e categoriza semelhanças semânticas entre itens de linguagem, com base em suas propriedades distributivas em amostras de dados de linguagem [Sat95]. Esta abordagem pode ser vista como uma estratégia que possuiu itens linguísticos com distribuições semelhantes

possuem significados semelhantes. Logo, esta distribuição levará em consideração situações (mundos) semelhantes, que deverá levar em consideração a somatória destes. A distribuição sobre programas é definida pela codificação de escolhas aleatórias para cláusulas, onde cada escolha gera uma versão alternativa da cláusula, e o conjunto de escolhas está associado a uma distribuição de probabilidade (descreve o comportamento aleatório de um fenômeno que depende do acaso).

A estratégia de construção de modelo de base de conhecimento (KBMC), um programa de lógica probabilística é uma forma compacta de codificar um modelo gráfico: rede Bayesiana (Bayesian Network) [D 99, NBBW06] ou uma rede de Markov (Markov Network) [Rig19, Won94]. A semântica de um programa nesta abordagem é definida pela forma de construção do modelo gráfico do programa.

4.2 PROLOG (Programming in Logic)

O PROLOG (LOGical PROgramming) é uma aplicação de programação em lógica, pois é uma linguagem de programação declarativa (descrevem o que fazem e não exatamente como suas instruções funcionam) e lógica. Ela foi desenvolvida em função de um projeto que não tinha como objetivo criar uma linguagem de programação, mas sim o processamento de linguagem natural. O projeto teve uma versão inicial em 1971, e uma versão mais robusta em 1972, no qual em 1973 teve a sua versão final, e nos anos de 1974 e 1975 a sua distribuição [CR93].

Por ser uma linguagem de programação declarativa de primeira ordem, o Prolog possui uma extensão da lógica proposicional que contém predicados, quantificadores e variáveis. Uma linguagem lógica de primeira ordem é definida por um alfabeto que consiste nos seguintes conjuntos de símbolos: variáveis, constantes, símbolos de funções, símbolos de predicados, conectivos lógicos, quantificadores e símbolos de pontuação. O Prolog é utilizado para resolver problemas envolvendo objetos (são coisas sobre as quais se deseja raciocinar) e relações entre objetos [Rig19, S.20].

Esta linguagem pode ser entendida como um conjunto de axiomas e de regras de inferência, descrevendo um determinado problema (base de conhecimento), isto é, o programa em si expressa os fatos e regras sobre diferentes problemas dentro de um sistema de lógica formal. Falando de forma mais assertiva, um programa Prolog consiste em um ou mais predicados, e cada predicado consiste em uma ou mais cláusulas lógicas (expressam as regras). Logo, definido o programa, pode-se realizar consultas para ver se uma proposição é verdadeira ou não. Sendo assim, para que um programa em Prolog possa ser executado, são realizadas deduções de consequências lógicas da base de conhecimento. Para que o Prolog seja acionado, questões são realizadas, onde o motor de inferência busca na base de conhecimento axiomas e regras que possibilitem apresentar uma resposta [S.20, Fra06].

Algumas linguagens de programação em lógica como ASP (Answer Set Programming) são conhecidas como linguagem puramente declarativa. No caso do ASP, ela permite declarações sobre o que o programa deve realizar, onde não existe instrução passo a passo sobre como executar a tarefa. Todavia, o Prolog, possui propriedades declara-

tivas e também imperativas, que possibilitam incluir declarações procedimentais como: para se resolver o problema (Header), execute as ações X,Y e Z. O Prolog tem como vantagem implementar com precisão redes neurais, algoritmos genéticos, sociedades de agentes inteligentes, sistemas concorrentes e paralelos. Ele libera o programador da preocupação com o controle das rotinas de um programa, mantendo a concentração nas questões lógicas do problema [Pal97].

4.3 ASP (Answer Set Programming)

O ASP é uma abordagem de programação declarativa, decorrente de uma representação de conhecimento e formalismo de raciocínio baseado na semântica de conjuntos de respostas de programas lógicos [MLS13]. Ele é um paradigma de resolução de problemas baseado no cálculo dos conjuntos de respostas de um programa [Rig19, HS14]. O ASP é uma forma de programação orientada para problemas de pesquisa considerados difíceis, principalmente NP-difíceis, trabalhando com representação e raciocínio do conhecimento baseado em lógica e resolução de restrições.

Esta linguagem é baseada na semântica do modelo estável, conjunto de respostas, de programação em lógica de [GL88]), que aplica ideias de lógica auto-epistêmica (uma lógica formal para a representação e raciocínio do conhecimento sobre o conhecimento)[Moo85] e lógica não monotônica (trabalha tanto com premissas positivas possui, quanto com premissas negativas.) à análise da negação como falha. O uso de regras de inferência não monotônicas foi a maneira encontrada, em Inteligência Artificial, para representar o que se chama de padrão evidencial de inferência, que apesar de não garantir a certeza da conclusão, há uma grande evidência de que é a melhor opção. Sendo assim, o ASP é uma programação de conjunto de respostas, e teve o seu talvez o seu primeiro caso de aplicação em [DNK97], onde são apresentados e discutidas situações de planejamento em programação em lógica relacionadas a desempenho e exploração de estruturas lógicas.

O ASP, como citado, surgiu como um paradigma declarativo de resolução de problemas que tem suas raízes na programação em lógica e no raciocínio não-monotônico. As vezes é chamada de A-Prolog (AnsProlog), utilizada para modelar e resolver problemas de senso comum. Isto significa que a solução de um problema será representada por modelos estáveis (conjunto de respostas), onde as regras e descrições são utilizadas ao invés de algoritmos clássicos [EIK09]. A programação por conjunto de respostas restritas (CASP) é uma outra linha de trabalho do ASP, onde se integra programação de conjunto de respostas com programação de restrição (lógica), que permite a aplicação de técnicas de processamento de restrição para raciocínio eficaz sobre construções não booleanas [MLS13].

Vale a pena ressaltar que o ASP tem sido aplicado com sucesso em diferentes áreas de representação do conhecimento e ciência da computação, como a configuração automatizada de produtos, levando a criação de um configurador de produto baseado na web. Também tece sucesso com o ônibus espacial, sendo uma ferramenta de apoio à decisão, sendo um sistema capaz de resolver algumas tarefas de planeamento e diagnóstico rela-

cionadas com o funcionamento da nave espacial. Por fim, um método baseado em ASP para reconstruir uma filogenia para um conjunto de táxons foi aplicado à análise histórica de línguas e à análise histórica de sistemas parasita-hospedeiro. O grupo de autores inclui um zoólogo, dois linguistas e dois especialistas em programação de conjuntos de respostas [MLS13, Lif08].

Os programas em ASP são semelhantes aos programas em Prolog, mas os mecanismos computacionais do ASP são diferentes, são baseados em solucionadores de satisfação para lógica proposicional. Uma diferença sintática é que o ASP possui regras de escolha, o que permite usar a disjunção no início de uma regra. Imagine um programa que declara p, podendo ser verdadeiro ou não, produz dois conjuntos, um vazio e o outro p. O ASP também trabalha com regras de escolha restrita como 1 p, q, r 2. Esta regra indica que deve ser escolhido pelo menos um do conjunto p, q, r, mas não mais do que 2 [S.20].

A tabela 4.1 apresenta uma breve comparação entre ASP e PROLOG.

	ASP	PROLOG
1	Trata problemas NP-Completo, problemas difíceis	Trabalha com problemas que possuem objetivo diretos.
2	Usados para problemas com abordagem Bottom-Up	Usados para problemas com abordagem Top-Down
3	Formato básico da regra: Instanciação e termos fixos	Formato básico da regra: Unificação e termos aninhados
4	Linguagem de modelagem: uma linguagem puramente declarativa. Possibilita a otimização do código.	Linguagem de programação: uma linguagem declarativa e descritiva. Provê a representação de um problema.
5	O programa como todo é relevante na resolução de um problema.	Uma análise de todo o programa é realizada, mas somente parte do programa pode ser processado.
6	Trabalha com regras de escolha.	A solução é dada por a ativação de uma consulta.

Tabela 4.1: ASP vs PROLOG

4.4 Programação Probabilística

Esta seção apresenta um estudo sobre a teoria da probabilidade, lógica de probabilidade, o que vem a ser programação em lógica e a apresentação das linguagens PROBLOG e CPLINT.

4.4.1 Teoria da Probabilidade

Explicar ou justificar a ocorrência de um fenômeno no mundo real ou mesmo de um experimento realiza n vezes sob diversas condições, pode ser um tanto complexo ou não.

Se for analisar que a ocorrência de um evento com base em fatos que são conhecidos ou mesmos previstos, sabe-se que a ocorrência do fenômeno é determinada, sendo um fenômeno determinístico. Agora suponha que um fenômeno é baseado em fatos aleatórios, isto é, não se sabe se irá acontecer ou mesmo quando irá acontecer. Determinar se este fenômeno acontecerá e quando acontecerá torna-se complexo, podendo dizer que são fenômenos estocásticos, com origem em fatos aleatórios. Sendo assim, esta seção visa tratar estas questões da teoria da probabilidade, responsável em negociar com eventos que ocorrem de forma randômica.

A teoria da probabilidade trabalha na chance de um evento (fenômeno) ocorrer ou não. Suponha um conjunto de eventos

$$E = A_1, A_2, A_3, ...$$

, sendo que para todo evento A existe um não A, isto é, o evento A não ocorreu se somente se A não ocorreu [Lo 17].

$$\forall A_i \exists A_i \wedge \neg A_i | \exists \neg A_i \Leftrightarrow A_i$$

Um evento pode implicar em outro evento, como $A \subset B$, isto é, quando o evento A acontece, então B acontecerá. Se $A \subset B$ e $B \subset A$, A = B, são equivalentes.

Se $A \wedge B$ é um evento $(ABouA \cap B)$ se somente se os eventos A e B acontecerem, isto significa se A ocorrer e B não ou vice-versa $A \cap B$ não acontecerá.

$$A_1 \cap A_2 \cap \ldots \cap A_n ou A_1 A_2 \ldots A_n ou \bigcap_{k=1}^{n} A_k$$

Se $A \vee B$ é um evento $(A \cup B)$ se somente se ao menos um dos eventos A ou B acontecerem.

$$A_1 \cup A_2 \cup \ldots \cap A_n ou A_1 A_2 \ldots A_n ou \bigcup_{k=1}^{n} A_k, A1 + A2 + \ldots + A_n or \sum_{k=1}^{n} A_k$$

Existe no mundo vários fenômenos apropriados em um esquema determinístico, isto é, dado um número X de circunstâncias, um evento A necessariamente irá acontecer. Porém, de outra forma, existem fenômenos que não são determinísticos, isto é, dado um número X de circunstâncias o evento A poderá acontecer ou não. Estes eventos são chamados de eventos randômicos, tal esquema é dito como estocástico, a chance do evento A ocorrer é indeterminado [Ré07].

Um evento randômico pode ser interpretado como um fenômeno que repetido várias vezes da mesma forma irá apresentar resultados não previsíveis. Pode-se citar como exemplo o o evento de um lançamento de uma moeda, no qual independente da quantidade de vezes que for lançada, não se pode afirmar quantas vezes cairá um lado ou outro

da moeda. A frequência de saída desse evento, executado de forma repetida n vezes pode ser representado na razão $\frac{n_a}{n}$, no qual n_a é a quantidade de vezes que um lado da moeda ocorreu [Rig19, Lo 17]. A teoria da probabilidade visa o estudo de fenômenos randômicos, explicando a sua importância.

As saídas de tentativas aleatórias são chamados de eventos randômicos, e o valor numérico, medido pela frequência de um evento randômico acontecer é chamado de probabilidade P_A, sendo que $0 \geq P_A \leq 1$. Seja n_A, n_B e n_AB os quantidade de ocorrências dos eventos A, B e AB em n repetições de tentativas aleatórias. A frequência do resultado B nas tentativas na em que A ocorre é

$$\frac{n_{AB}}{n_A} = \frac{n_{AB}}{n} : \frac{n_A}{n}.$$

Medindo a razão PAB/PA, chamada a probabilidade de B dado A (P_AB), que corresponde a PAB=PA.PB. Adicionando o fato de que A ocorre, a probabilidade de B, PB, é transformada em P_AB, definindo B como estocástico (ie, o estado é indeterminado, com origem em eventos aleatórios) de A se $P_AB =$ PB ou PAB=PA.PB. Sendo assim, pode-se dizer que qualquer sistema analisado e que faz uso da teoria probabilística é estocástico.

Imagine A um evento e um experimento repetido várias vezes sob diversas condições, o evento A irá ocorrer em algumas situações e em outras não. No caso de n experimentos A ocorreu k vezes, onde k é chamado de frequência, e $\frac{k}{n}$ é chamado de frequência relativa do evento A em uma sequência de n experimentos. A frequência relativa, chamada de probabilidade, normalmente não é constante em eventos randômicos [Ré07]. A probabilidade pode ser entendida como um segmento da matemática que estuda a chance de eventos acontecerem. Segundo Raymond [RS92], a probabilidade possui uma importante função na compreensão do mundo e na maneira como se raciocina sobre ele. Por exemplo, quando da pandemia da COVID chegou-se a falar que cerca de 50% da população mundial poderia ser infectada. Isto significava que para cada 2 (dois) pessoas uma seria infectada. Este simples exemplo apresenta que a utilização de raciocínio sobre informações probabilísticas e estatísticas é bastante comum em diversas situações do mundo real. Ao atribuir um significado a valores, os problemas podem ser chamados de:

- Diagnóstico - P(causa|sintoma).
- Predição - P(sintoma|causa).

4.5 Lógica Probabilística

A inferência possui o papel principal na lógica como um todo, ela é fundamental para grande parte da lógica computacional e também para boa parte da programação em lógica, pois absorve o raciocínio não monotônico, logica difusa e possibilística, com também as combinações de lógica e probabilidade. A inteligência artificial faz uso dessa interação entre inferência probabilística e lógica, além de outras áreas de atuação (medicina,

análise de linguagem natural, ...). Estas questões levaram a pesquisas de formalismos probabilísticos (redes bayesianas, redes de Markov, ...), refletindo assim nas diversas abordagens da Programação em Lógica Probabilística (PLP) existentes (PRISM, Programas de Lógica Bayesiana, ...) [RS18].

Desta forma, a lógica probabilística pode ser vista como uma lógica proposicional ou de primeira ordem, que visa manipular não apenas de asserções verdadeiras ou falsas, mas também proposições de caráter probabilístico, não afirmando que sejam verdadeiras ou falsas, mas tenham uma probabilidade destes valores acontecerem [Glu08].

Em [Hal90], Halpern cita que para cada interpretação de probabilidade seria necessário uma linguagem lógica para o seu tratamento, isto é, seria necessário definir um conjunto de linguagens lógicas capazes de lidar tanto com probabilidades relacionadas a um domínio de discurso chamadas de probabilidades objetivas, quanto para probabilidades relacionadas aos estados de um agente sobre o domínio [Glu08].

Desta forma Halpern [Hal90] criou 3 (três) linguagens para tratar as probabilidades, que expressam afirmações lógicas probabilísticas.

- A primeira linguagem tem como objetivo construir afirmações lógicas sobre os elementos de um domínio, basicamente informações do tipo estatístico. Por exemplo:

  A probabilidade de um tráfego IoT qualquer possuir uma ameaça é maior que 0.8, mais que 80% de chances.

- A segunda linguagem tem como objetivo apresentar afirmações relacionadas a um agente, por exemplo:

  Existe uma certeza (85%) que o tráfego proveniente do dispositivo IoT k1 possui uma ameaça.

- A terceira linguagem é uma combinação das 2 linguagens anteriores, que possibilita apresentar expressões probabilísticas que envolvam tanto informações estatística sobre os elementos de um domínio quanto das crenças de um agente. Vale a pena ressaltar que se pode combinar os 2 tipos de probabilidade em uma mesma expressão.

  A chance de existir uma ameaça no tráfego IoT vindo do dispositivo K1 é menor que qualquer outro dispositivo IoT pertencente ao sistema analisado.

Concluindo, se pode afirmar que a principal vantagem de se trabalhar com raciocínio probabilístico sobre raciocínio lógico, consiste no fato de que decisões podem ser tomadas de forma racional mesmo que não exista informação suficiente para se provar que uma ação realmente irá funcionar de forma adequada. Isto porque não existem informações em uma base de conhecimento, seja por ignorância teórica ou por não se conseguir coletar informações.

4.5.1 Aspectos sobre Programação Probabilística

A programação probabilística pode ser considerada como uma ferramenta útil para a construção de modelos probabilísticos complexos, com o intuito de realizar inferências e aprender sobre estes. A programação em lógica probabilística (PLP - Probabilistic Logic Programming) é a programação probabilística baseada em programação em lógica que permite modelar domínios caracterizados por relacionamentos complexos e incertos entre entidades de domínio [ACRZ16].

A programação probabilística negocia com a incerteza, aplicando a teoria da probabilidade e modelos gráficos (redes bayesian e de markov), raciocina com dados relacionais, base de dados lógica e programação. Possibilita um aprendizado de valores lógicos e estrutura, por exemplo. A programação probabilística possibilita encontrar formas que liga as causas aos efeitos, no qual via instâncias de diferentes eventos, pode -se citar o grau de incerteza dele ter acontecido.

O campo da programação em lógica probabilística (PLP) tem visto muitas propostas diferentes para integrar a programação em lógica e a teoria das probabilidades. A Semântica de Distribuição (DS), uma das técnicas mais utilizadas, tem como base que um programa lógico probabilístico define uma distribuição de probabilidade sobre um conjunto de programas lógicos normais, chamados mundos, sendo estendida a uma probabilidade conjunta de programas e valores verdade de uma consulta básica. Desta forma, a probabilidade da consulta é então obtida a partir da distribuição conjunta por marginalização [Rig17]. Logo, a distribuição semântica da lógica probabilística pode ser visto desta forma [Sat95]:

escolha probabilísticas + programa lógico → distribuição sobre mundo possíveis

Apresentamos o DS para Programa Lógico com Disjunções Anotadas (LPADs) pela sua sintaxe geral, no qual o LPADs são conjuntos de cláusulas disjuntivas em que cada átomo no cabeçalho é anotado com uma probabilidade. Formalmente, um LPADs consiste em um conjunto finito de cláusulas disjuntivas anotadas. Uma cláusula disjuntiva anotada C_i é da forma: [Rig17]

$$h_{i1} : \prod_{i1}; \ldots; h_{in_i} : \prod_{in_i} \leftarrow b_{i1}, \ldots, b_{im_i}$$

Onde o ponto e vírgula significa disjunção, $h_{i1}, \ldots, h_{in_1}$ são átomos lógicos e $b_{i1}, \ldots, b_{im_i}$ são literais lógicos, $\prod_{i1}, \ldots, \prod_{im_i}$ são números reais no intervalo [0,1], tal que $\sum_{k=1}^{n_i} \prod_{ik} \leq 1$. Se $\sum_{k=1}^{n_i} \prod_{ik} < 1$, o cabeçalho da cláusula disjuntiva anotada contém implicitamente um átomo extra nulo que não aparece no corpo de nenhuma cláusula e cuja anotação é $1 - \sum_{k=1}^{n_i} \prod_{ik}$.

4.6 PROBLOG

O projeto do Problog, foi desenvolvido por Raedt em [Rae07], motivado pelo desejo de fazer uma simples extensão probabilística do Prolog [Rig19]. O ProbLog é essencialmente o Prolog, no qual todas as cláusulas são rotuladas com a probabilidade de serem verdadeiras e são mutuamente independentes. A motivação do Problog foi a aplicação na vida real de mineração de grandes redes biológicas onde as arestas são rotuladas com probabilidades. Estas redes são extraídas de grandes bases de dados, e as ligações probabilísticas podem ser obtidas por diversas técnicas de predição [Rae07].

Um programa ProbLog especifica uma distribuição de probabilidades sobre todos os subprogramas não probabilísticos do programa ProbLog, isto é, ele define uma distribuição sobre programas lógicos atribuindo para cada cláusula existente valor de probabilidade, sendo que estas probabilidades são mutuamente independentes. O Problog possibilita que o programador crie programas que não apenas codifiquem interações em um universo de eventos heterogêneos, mas que também permita trabalhar com as incertezas existentes nesse universo de eventos [Rae07].

Sendo uma linguagem com uma simples sintaxe, um programa em Problog é composto por duas partes, uma parte probabilística que define uma distribuição de probabilidade sobre os valores verdade de um subconjunto de átomos do programa e uma segunda parte lógica que deriva os valores verdade dos átomos restantes usando um mecanismo de raciocínio semelhante ao Prolog. Enquanto a segunda parte contém simplesmente cláusulas Prolog, a primeira é especificada por fatos probabilísticos p :: fato, o que significa que o fato é verdadeiro com probabilidade p. Todos os fatos são probabilisticamente independentes; caso contenham variáveis, todas as instâncias básicas também são independentes [Rae07, DKM$^+$15].

Os fatos probabilísticos são expressos na forma $P_i :: F_i$, sendo que $P_i \in [0, 1]$, e F_i é um átomo, o que significa que cada instanciação fundamental $f_i de F_i$ é verdadeira com probabilidade P_i e falsa com probabilidade $1 - \prod_i$. Cada universo de eventos é obtido selecionando ou rejeitando a fundamentação de cada fato probabilístico. Uma escolha atômica indica se a fundamentação de um fato probabilístico F = p::f, no qual p0,1, o fato é escolhido se p=1 ou não quando p=0 [Rig19].

Para melhor compreensão, imagine uma breve comparação entre programação em lógica e programação probabilística, fazendo uso do PROLOG e PROBLOG.

A - Programação em Lógica: PROLOG – apresenta a realidade de um mundo, representado por fatos e regras. No exemplo que se segue:

Fatos – representam um mundo

```
pacote(tcp,'192.168.2.1','192.168.2.3',25551,53).
pacote(tcp,'192.168.2.2','192.168.2.3',25555,53).
```

Regras – correlacionam estes fatos

```
anomaly(A,B,C,D,E):- pacote(A,B,C,D,23).
anomaly(A,B,C,D,E):- pacote(tcp,B,C,D,E).
anomaly(A,B,C,D,E):- pacote(A,B,C,'192.168.2.3',E).
```

B - Programação probabilística: PROBLOG – apresenta vários possíveis fatos dentro de um mesmo mundo, todos representados na forma $P_i :: F_i$, no qual a probabilidade é estendida a uma regra de Prolog. Logo, um átomo será verdadeiro se aquela regra for verdadeira.

```
0.3::anomaly(A,B,C,D,E):- pacote(A,B,C,D,23).
0.6::anomaly(A,B,C,D,E):- pacote(tcp,B,C,D,E).
0.7::anomaly(A,B,C,D,E):- pacote(A,B,C,'192.168.2.3',E).
```

O ProbLog2 é o sucessor do ProbLog1, que foi completamente integrado ao YAP Prolog[1] e realizou inferência probabilística baseada em BDD (Diagrama binário de decisão)[2] [DKM+15].

A inferência ProbLog2 consiste em uma série de etapas de transformação como:

- A etapa em transformar o programa lógico no formato do Problog, que pode ser representado como uma fórmula lógica com alguns ciclos.
- Nas próximas etapas, o programa criado é convertido em uma fórmula em lógica proposicional[3] que envolve o tratamento de ciclos.

O Problog2 proporciona diferentes opções estão disponíveis a conversão, utilizando diferentes técnicas de raciocínio lógico. A compilação direta compila diretamente em diagramas de decisão sentenciais (SDD). Alternativamente, os ciclos podem ser removidos, e o programa fundamental resultante pode ser transformado na forma normal conjuntiva e compilado em um DNNF, ou compilado diretamente em um SDD. Ambas as formas normais suportam uma contagem eficiente de modelos ponderados para obter as probabilidades finais de interesse.

O ProbLog2 também oferece suporte à amostragem baseada em consulta, na qual atribui um valor verdadeiro a cada uma das consultas com base em sua probabilidade. Este algoritmo opera diretamente durante a fase de criação do programa em Problog e não requer compilação de conhecimento.

4.7 CPLINT

Segundo Riguzzi em [Rig17], cplint é um conjunto de programas para raciocínio e aprendizagem com linguagens de programação em lógica probabilística que seguem a semântica

[1] https://www.dcc.fc.up.pt/ vsc/yap/
[2] https://www.inf.ufrgs.br/ MRPRITT/lib/exe/fetch.php?media=inf05508:t-a-bdds.pdf
[3] https://www.facom.ufu.br/gustavo/Logica/ApostilaLogicaProposicional

de distribuição. O cplint trabalha com variáveis aleatórias contínuas com o módulo de inferência de amostragem. Desta forma, o usuário pode especificar uma densidade de probabilidade para um argumento Var (variável) de um átomo a com regras [Rig19]:

$$a : Densidade \leftarrow Corpo$$

Para um melhor entendimento, a Densidade é um átomo que identifica a probabilidade na variável Var e Corpo é uma de cláusula, que poderá ser opcional. Seguem algumas poucas formas como os átomos de densidade podem ser vistos:

- Uniforme (Var, L, U) - a variável Var é uniformemente distribuído em $[L, U]$.
- Gaussian (Var, Média, Variância) – é a distribuição gaussiana com parâmetros Média e Variância, que pode ser multivariada se a Média for uma lista e Variância uma lista de listas representando o vetor médio e a matriz de covariância. Neste caso, os valores da variável Var são listas de valores reais com o mesmo comprimento da Média.

Suponha que $g(X) : gaussian(X, 0, 1)$, onde o argumento X de g(X) segue a distribuição com a média 0 e a variância 1, sendo que:

$$g(X) : gaussian(X, [0, 0], [[1, 0], [0, 1]])$$

Logo, afirma que o argumento X de g(X) segue uma distribuição multivariada gaussiana com vetor médio [0,0] e matriz de covariância:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

O operador ST_p do cplint - seja P um programa e base (P) um conjunto de todas as instâncias das cláusulas em P com todas as variáveis no corpo substituídas por constantes, e partindo de um conjunto de fatos básicos I, o operador STP retorna:

$ST_p(I) - \{h | h : Densidade \leftarrow b_1, , b_n \in base(P) \wedge \forall b_i : b_i \in I \vee h = h\{Var/v\}$ com Var a variável contínua de h e v as amostras de Densidade$\}$

$\cup \{h | Dist \leftarrow b_1, ..., b_n \in base(P) \wedge \forall b_i : b_i \in$ I com h uma amostra da distribuição discreta Dist$\}$

Não existe a necessidade de tratamento especial para os átomos do corpo, todos são átomos lógicos. Para cada cláusula probabilística $h : Densidade \leftarrow b_1, ..., b_n$ sempre que o corpo $b_1, ..., b_n$ é verdadeiro em I, um valor v para a variável contínua Var de h é uma amostra da distribuição Densidade e $h\{Var/v\}$ é adicionado à interpretação. O mesmo é realizado para as cláusulas discretas e determinísticas.

4.8 Considerações

Este capítulo apresentou os fundamentos de programação lógica e programação probabilística, em conjunto com as linguagens PROLOG, ASP, PROBLOG e CPLINT. O próximo capítulo terá como objetivo apresentar o que é rede bayesiana e cadeias de markov.

5

Rede Bayesiana X Cadeias de Markov

Neste capítulo apresentamos o que é uma rede Bayesiana, sua aplicação além de um exemplo de sua utilização. Também foi discutido o que são cadeias de Markov, e apresentadas as cadeias em tempo discreto e tempo contínuo. O capítulo foi dividido em duas seções, uma para rede bayesiana e a outra para cadeia de markov.

5.1 Rede Bayesiana

Trabalhar com probabilidades (raciocínio probabilístico) é trabalhar com situações que não se tem conhecimento de todo o escopo de um problema. O raciocínio probabilístico tem como principal característica/vantagem é que decisões podem ser tomadas mesmo sem possuir todas as informações, seja por falta de conhecimento (ignorância dos fatos) ou impossibilidade de obter tais informações [Rig19, Sat95].

Dentro deste universo da probabilidade, existe o conceito de probabilidade incondicional (PI) e condicional (PC). No caso da PI (probabilidade incondicional), a probabilidade de um evento P(e) possui um valor no intervalo de [0,1]. Se 2 (dois) eventos (a,b) são mutuamente exclusivos, então $P(a \vee b) = P(a) + P(b)$. Porém, na PC (probabilidade condicional), P(a|b) = x, pode ser vista como dado o evento b, a probabilidade do evento a é x [D 99, NBBW06, Cha91].

O teorema de Bayes tem como seu objetivo calcular as chances de um evento acontecer com base em dados e um conhecimento prévio, isto é, calcular a probabilidade de um evento com base em outro evento que ocorreu, probabilidade condicional. Desta forma, para que o teorema de Bayes funcione, é necessário conhecer a probabilidade de um evento que ocorreu anteriormente. A fórmula do teorema de Bayes é:

$$\mathbf{P(b|a) = P(a|b)P(b)/P(a)}$$

P(b|a) é a probabilidade do evento b ocorrer quando o evento a ocorreu; P(b) é a probabilidade do evento b ocorrer; e P(a) é a probabilidade do evento a ocorrer [D 99, NBBW06].

Uma rede bayesiana pode ser definida como uma representação de relacionamentos incertos entre parâmetros em um domínio [D 99], são relações de probabilidade condicional, isto é, como que a ocorrência de certas variáveis depende do estado de outra. Um outra forma de entender, ela é uma representação compacta de uma tabela de conjunção de probabilidades de um problema, representado por um modelo gráfico apresentando as relações de causalidade das variáveis de um sistema.

As redes bayesianas são representadas por grafos acíclicos direcionados a uma conclusão, o qual representam dependências entre variáveis em um modelo probabilístico. Ela pode ser construída com base em um conjunto de parâmetros que expressem o relacionamento ente eles [NBBW06]. Desta forma, ela consiste de um conjunto de variáveis e um conjunto de arcos ligando estas variáveis. Cada uma das variáveis possui um conjunto limitado de estados que são mutuamente exclusivos. As variáveis e arcos formam um grafo dirigido sem ciclos, e cada variável A que possui como pais B1, ..., Bn, existe uma tabela P(a| b1, ..., bn). Redes bayesianas trabalham com probabilidade condicional P(a|b) = x, no qual dado o evento b, a probabilidade do evento a é x.

Sendo assim, uma rede bayesiana possibilita uma forma para estruturar informações probabilísticas sobre uma situação. As informações derivadas desta servem como base para conclusões e decisões de uma situação correspondente [Rig19]. Desta forma, uma rede bayesiana pode ser considerada como uma de um conjunto de probabilidades, normalmente amplo para ser manipulado de forma simples e única.

5.1.1 Aplicação Rede Bayesiana

A necessidade de escolher ou dar relevância a determinadas informações, com base em fatos/eventos ocorridos, no cálculo da probabilidade de uma determinada situação, o uso

de rede bayesiana é uma opção. Para isso 3 (três) termos são necessários, a probabilidade condicional e 2 (duas) incondicionais.

Para melhor entender, suponha necessite determinar a existência ou não de uma ameaça em um fluxo de dados IoT, no qual inferir a probabilidade da existência ou não dessa ameaça é uma questão complexa. Utilizar rede bayesiana como uma técnica para tratar esta questão pode ser considerada uma boa estratégia, pois esta é capaz de tratar problemas de incerteza no qual os resultados desses problemas não podem ser obtidos sem antes possuir informações prévias sobre ele.

Como citado, inferir um valor inicial para a probabilidade de que exista uma ameaça em um tráfego IoT é um tanto difícil, pois um conjunto de elementos devem ser levados em consideração. Fazer uso de rede bayesiana para inferir o valor da probabilidade, consiste em avaliar eventos em conformidade com a semelhança de ocorrência dos mesmos, possibilitando extrair diversas interpretações do valor da probabilidade. Isto porque se trabalha com valores de probabilidade previamente conhecidos, além de diversos outros valores de probabilidade possibilitando estabelecer o valor da probabilidade de um pacote IoT possuir uma ameaça

Com o objetivo de melhor entender o uso de redes bayesianas como uma das possíveis soluções de inferir probabilidade, criou-se um exemplo. A rede bayesiana apresentada na figura 5.1 tem como objetivo de inferir o valor da probabilidade de um pacote IoT possuir uma ameaça. Inicialmente foi considerada que a probabilidade é igual tanto para ser considerado um pacote normal (sem ameaça) ou com ameaça. A composição da tabela de probabilidade condicional(CPT - Conditional Probability Table) do nó filho se baseia que o pacote pode ter sido alterado ou não.

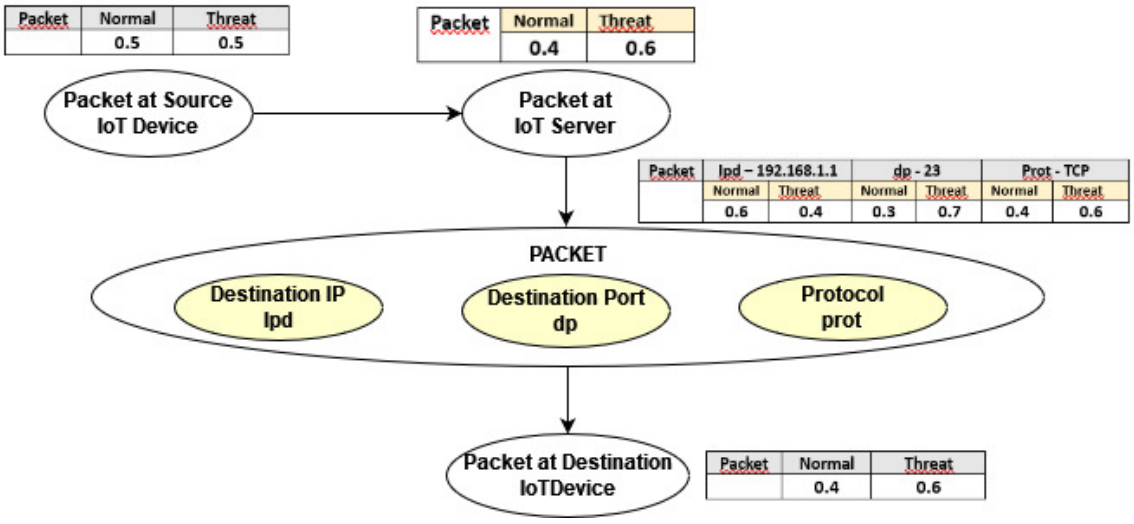

Figura 5.1: Bayesian Network

O pacote quando analisado no próximo nó, foram levados em consideração os parâmetros de análise, como endereço IP de destino, porta de destino e protocolo. Os valores atribuídos foram com base em observações do tráfego existente, no IoT-23, entre dois dispositivos IoT. A composição da tabela de probabilidade condicional teve 60% a probabilidade do pacote não possuir ameaça se for endereçado ao Ip 192.168.1.1 e 40% de

probabilidade possuir ameaça. Se o pacote tiver como destino a porta 23 o percentual é de 30% para que não possua ameaça e 70% para que possua ameaça. Por fim, se o protocolo de rede for TCP, o percentual é de 40% para não possuir ameaça e 60% para ter ameaça.

Sendo assim, o resultado da rede bayesiana, isto é, a tabela de probabilidade condicional apresentou a probabilidade de 40% para que o pacote não possua ameaça e 60% de probabilidade que ele possua ameaça. Valores calculados com base na somatória de todos os elementos da tabela de probabilidade condicional, isto é, valor de cada parâmetro (ipd, dp e prot), dividido pela quantidade destes. O valor de 60% (0.6) será aplicado na probabilidade incial para que o primeiro pacote seja analisado se existe uma ameaça no mesmo.

Para o cálculo das demais probabilidades dos pacotes do tráfego IoT será utilizada programação em lógica (Problog) [Rae07], onde o resultado de cada programação será utilizado como uma nova entrada para os próximos cálculos de probabilidade. Isto significa que P poderá assumir o valor dado aos parâmetros (dp, prot, sp e ipd, por exemplo) forem verdadeiras, ie, todas as proposições assumirem os valores considerado como ameaças. A programação se encerra quando os valores de probabilidade alcançarem um valor estável.

$$\text{“P(dp} \wedge \text{prot} \wedge \text{ipd)} = (\text{(Ipd|threat)} + \text{P(dp|threat)} + \text{P(prot|threat))/P(threat)} = (0.4 + 0.7 + 0.6)/3 = 0.6\text{”}$$

Por fim, trabalhar com redes Bayesianas permitem representar, de forma quantitativa, o grau de certeza e por conseguinte manipular as representações conforme as leis da probabilidade, apresentando vantagens como:

- Possibilita manipular situações onde não se conhece todo o escopo do problema;
- É possível realizar a inferência aproximada da probabilidade de um evento por métodos empíricos;
- Possui custo computacional reduzido;
- Possibilitar a coleta de conhecimento de uma amostra de dados proporcionando reações aos problemas, possibilitando a prevenção de eventos;
- Trabalhar com redes Bayesianas como classificadoras tem como problema: se o número de configurações possíveis é grande, o erro de estimação será grande;
- Se a amostra for o custo será alto para estimar os parâmetros; e
- Cada classe em redes Bayesiana possui independência de características em relação a outras classes.

5.2 Cadeias de Markov

Mentor das cadeias de Markov, Andrei Andreevich Markov nasceu na Rússia em 1856, vindo a falecer em 1922. Os escritos de Markov sobre cadeias ocorrem dentro de seu interesse pela teoria da probabilidade, professor universitário, assumiu o ensino do curso de teoria da probabilidade e continuou a ensiná-lo anualmente, mesmo após sua própria aposentadoria da universidade como professor emérito em 1905. As publicações de probabilidade de Markov foram motivadas por inadequações no tratamento de Chebyshev do Problema do Limite Central [Sen06].

A cadeia de Markov é uma classe de processos estocásticos $\{X_t | t \in T\}$, processos aleatórios, que pode ser definido como uma família ou agrupamento de variáveis randômicas X_t, indexadas por um parâmetro t e pertencente a um conjunto T, definidas no mesmo espaço de probabilidade e assumindo valores em um conjunto S, geralmente referido como o espaço de estados de um processo [Her02].

O conjunto de parâmetros T é frequentemente interpretado como tempo e às vezes é chamado de intervalo de tempo. Cada variável aleatória X_t descreve uma distribuição aleatória instantânea no espaço de estados S do processo no tempo t. O intervalo de tempo pode ser discreto ou contínuo, que proporciona uma separação em duas classes de processos estocásticos: de tempo discreto e de tempo contínuo. Normalmente T representa um conjunto de valores inteiros não negativos, sendo que reais não negativos em processos estocásticos contínuos e números naturais no caso de processos estocásticos discretos. [Llo74, Her02]

Quanto ao estado o processo estocástico pode ser dividido em processo de estado discreto ou contínuo. No estado discreto, o conjunto de variáveis aleatória é definida de forma enumerável ou finita, e no estado contínuo o conjunto é uma sequência e não finita. Em relação ao tempo, ele pode ser dividido em tempo discreto (finito ou enumerável) e contínuo (tempo não finito).

Exemplo de Processo Estocástico - Evolução da temperatura em um determinado local

- O espaço de estados S deste processo ficará em uma faixa de temperatura razoável.
- Cada variável aleatória Xt terá uma probabilidade para qualquer temperatura possível no tempo t. Por exemplo, a temperatura máxima ou média por dia, o intervalo de tempo pode ser discreto.
- Isto leva a um processo estocástico em tempo discreto onde X0 descreve uma distribuição de probabilidade na faixa de temperatura do primeiro dia, X1 do dia seguinte e assim por diante.
- Alternativamente, pode descrever a evolução contínua da temperatura por meio de um processo estocástico de tempo contínuo (escolhe um intervalo de tempo contínuo).

Desta forma, um processo de Markov é um processo estocástico que satisfaz um requisito adicional. Esta propriedade de Markov exige que, para qualquer instante de tempo t_n, o comportamento futuro, o valor de $X_{t_{n+1}}$, seja totalmente independente de sua história, isto é, os valores de $X_{t_{n1}}$, $X_{t_{n2}}$, e assim por diante, dependendo apenas do estado ocupado no instante atual t_n, dado pelo valor de X_{t_n}.

Logo, um processo estocástico (aleatório) com base na propriedade de Markov tem como premissa trabalhar com fatos futuros com base somente no estado atual do processo, sem levar em consideração o que aconteceu até aquele momento. Desta forma, não existe relação entre o estado atual do processo, estados futuros e passados, todos são independentes. Eles podem ser classificados em relação ao estado e ao tempo. [Her02]

Matematicamente falando, as propriedades de Markov necessitam que para cada sequência de estados do tempo $t_{n+1} > t_n > tn - 1 > ... > t_0$ existe para cada subconjunto A de estados:

$$Prob\{X_{tn+1} \in A | X_t n = P_n, X_{tn-1} = P_{n-1}, ..., X_0 = P_0\} = Prob\{X_{tn+1} \in A | X_t n = P_n\}$$

Assim, todos os estados $Pn - 1$ até P_0 são totalmente irrelevantes, pois o estado Xtn contém todo o histórico de informação necessária para determinar a distribuição randômica em S no tempo t_{n+1}. Logo, um processo estocástico (aleatório) com base na propriedade de Markov tem como premissa trabalhar com fatos futuros com base somente no estado atual do processo, sem levar em consideração o que aconteceu até aquele momento. Desta forma, não existe relação entre o estado atual do processo, estados futuros e passados, todos são independentes.

A definição anterior é adaptada para processos de Markov de tempo contínuo, mas o caso de tempo discreto é um pouco mais simples, pois não necessita se preocupar com sequências arbitrárias de instâncias de tempo. É considerada a sequência (única) que contém todas as instâncias de tempo anteriores. Identifica T com números naturais N, $t \in N$ arbitrário.

$$Prob\{X_{t+1} \in A | X_t = P_t, X_{t-1} = P_{t-1}, ..., X_0 = P_0\} = Prob\{X_{t+1} \in A | X_t = P_t\}$$

5.2.1 Uma cadeia de Markov simples - Tempo Discreto

Considera-se cadeias finitas somente aquelas cadeias em que as variáveis Xt estão restritas a um conjunto finito de valores ou estados 0,1,...,n, de forma que uma cadeia é uma simples cadeia de Markov se somente se: [Llo74]

$P(X_t = i | X_{t-1}, = j, X_{t-2} = k, ..., X_0 = w) = P(X_t = i | X_{t-1} = J), i, j = 0, 1.....n, t = 1, 2, ...$

Sendo que a variável X_t somente será influenciada pelo estado atual, o valor X_{t-1}, e não por valores anteriores.

Uma cadeia de Markov discreta consiste de [Beh00]:

- um conjunto finito não vazio T, o espaço de estados; os elementos de T são chamados de estados, normalmente identificados T com um conjunto 1, ... ,n;

- um vetor de probabilidade, ou seja, números $(P_i)i \in scomP_i \geq 0$, para todo $ie \sum p_i = 1$, no qual esses números determinam o gerador aleatório para a posição inicial i, com probabilidade P_i;

- a matriz estocástica, todo p_{ij} são não negativos e para todo i, a matriz P é uma abreviação conveniente de uma descrição dos geradores aleatórios associados aos estados.

Todavia, a distribuição conjunta das variáveis em uma cadeia de Markov simples é o produto dos termos decorrentes das distribuições bivariadas de pares consecutivos sobrepostos, terminados pela distribuição univariada da primeira variável no segmento. Com isso, uma cadeia de Markov simples pode ser especificada pela distribuição inicial $distr(X_0)$ e as probabilidades condicionais $P(X_t = r|X_{t-1} = s), t = 1, 2, ...$; ou equivalentemente pelas distribuições bivariadas $distr(X_t, X_{t-1}), t = 1, 2, ...$

Um simples exemplo de uma cadeia apresentado em [Llo74] é a de soma cumulativa. Seja $X_t = Y_1 + Y_2 + ... + Y_t, t + 1, 2, ...$

Sabe-se que todos os valores de Y são mutuamente independentes, e modelado da seguinte forma:

$X_1 = Y_1,$

$X_t = Xt - 1 + Y_t, t = 2, 3, ...$

O valor de Y_t será inicialmente independente de $Yt - 1, Y_{t-2}, ..., Y_1 edeX_{t-1}, Xt - 2, ..., X_1$, no qual o processo X_t é visto como um modelo autorregressivo, conhecido uma representação de um tipo de processo aleatório, usado para descrever certos processos variantes no tempo, por exemplo.

Sendo assim, obtém-se que:

$P(X_t = r, X_{t-1} = s) = P(X_t = r|Xt - 1 = s)P(Xt - 1 = s) = P(Xt - 1 = s)P(Y_t = r - s)$

Desde que:

$P(X_t = r|Xt - 1 = s) = P(Y_t = r - s)$

Se os valores de Y forem distribuídos de forma idêntica, as probabilidades de transição são independentes de t.

5.2.2 Cadeias de Markov em Tempo Contínuo

As cadeias de Markov de tempo contínuo são mais complexas compradas com as de tempo discreto. As cadeias de tempo contínuo constituem a base da metodologia contemporânea de avaliação de desempenho. A cadeia de tempo contínuo é um processo de Markov com espaço de estados discreto, mas com intervalo de tempo contínuo. [Her02]

A propriedade de Markov, para $t_n + \triangle t > t_n > t_{n1} > ... > t_0$ é:

$Prob\{X_{tn+\triangle t} = P' | X_{tn} = P, X_{tn-1} = P_{tn-1}, ..., X_{t0} = P_{t0}\}$

$= Prob\{X_{tn+\triangle t} = P' | X_{tn} = P\}$

$= Prob\{X_{\triangle t} = P' | X_0 = P\}$

5.3 Considerações

Este capítulo teve como objetivo apresentar os fundamentos de rede bayesiana e sua aplicação, como também os conceitos de cadeia de markov e a descrição de uma cadeia em tempo discreto e de tempo contínuo.

O próximo capítulo descreverá os conceitos de lógica difusa e seus componentes.

6

Lógica Difusa

Neste capítulo apresentamos a lógica difusa, seus conceitos e o controlador lógico difuso (FLC), componente central na composição da arquitetura do Threat Analyzer, módulo do RTRMM.

O capítulo foi dividido em três seções, fundamentos, controlador lógico difuso e os seus tipos.

6.1 Fundamentos

O conceito de "fuzzy sets" (conjuntos nebulosos) foi apresentado por Zadeh [Zad65] em 1965, e em 1978 desenvolveu a Teoria das Possibilidades [Zad78] uma opção menos restrita que a noção de probabilidade. O nascimento da "Fuzzy Logic" (Lógica Difusa) deve-se ao fato de que foi observado que não se consegue descrever uma situação real,

de forma ampla, baseado em lógica booleana, isto porque, tradicionalmente, uma proposição lógica trabalha com dois extremos, ou é falso ou o verdadeiro (0 ou 1). Não existe um meio termo nas tomadas de decisão, tornando as decisões imprecisas e dificultando a modelagem de inferência.

Logo, a lógica difusa pode ser vista como uma extensão da lógica multivalorada, mas a sua utilização e objetivos são diferentes. O fato da lógica difusa trabalhar com raciocínios aproximados e não precisos implica que, de uma forma geral as suas sequências de raciocínio na são curtas e rigorosas quanto nos sistemas lógicos clássicos. Desta forma, pode-se dizer que na lógica difusa tudo, inclusive a verdade, é uma questão de grau. [Zad88]

A grande virtude da lógica difusa é o fato de que ela trabalha com situações especiais, não apenas os casos clássicos de dois valores e sistemas lógicos multivalorados, mas também teoria das probabilidades e lógica probabilística. As suas principais características, as quais diferenciam dos sistemas lógicos tradicionais são: [Zad88]

- Nos sistemas lógicos de dois valores, uma proposição é verdadeira ou falsa, mas nos sistemas lógicos multivalorados uma proposição pode ser verdadeira, falsa ou ter um valor verdade intermediário, que pode ser um elemento de um conjunto de valores verdade finito ou infinito X. Na lógica difusa, os valores verdade podem variar nos subconjuntos difusos de X. Por exemplo, se X for o intervalo entre [0,1], o valor verdade na lógica difusa, como "muito verdadeiro" pode ser interpretado como um subconjunto difuso deste intervalo. Neste sentido, um valor de verdade difuso pode ser visto como uma caracterização imprecisa de um valor verdade numérico.

- Os predicados na lógica de dois valores são limitados a serem nítidos no sentido de que a denotação de um predicado deve ser um subconjunto não difuso do universo do discurso. Na lógica difusa, os predicados podem ser nítidos como: alto, baixo, grande, pequeno, muito largo.

- As lógicas de dois valores e múltiplos valorada permitem apenas dois quantificadores: todos e alguns. Todavia, a lógica difusa permite também o uso de quantificadores difusos como: a maioria, vários, muitos, às vezes, ... Sendo assim, estes quantificadores são vistos como números difusos que fornecem uma caracterização imprecisa da cardinalidade de um ou mais conjuntos difusos. Logo, os quantificadores difusos podem ser usados para representar o significado de proposições contendo as probabilidades difusa e possibilitar a manipulação de probabilidades dentro da lógica difusa.

- Em sistemas lógicos de dois valores, uma proposição pode assumir um valor verdade (verdadeiro ou falso); um operador modal (possível ou necessário); e um operador intencional (saber ou acreditar).

  A lógica fuzzy tem três modos principais de qualificação:

  - Qualificação de verdade - Maria é jovem, não é totalmente verdadeira, em que a proposição qualificada é (Maria é jovem) e o valor de verdade qualificativo é não totalmente verdadeiro;

  - Qualificação de probabilidade - Maria é jovem é improvável, em que a probabilidade difusa qualificativa é improvável;
  - Qualificação de possibilidade - Maria é jovem é quase impossível, em que a possibilidade difusa de qualificação é quase impossível.

Concluindo, pode-se afirmar que a lógica difusa foi criada com base na teoria dos conjuntos difusos, no qual uma premissa não é totalmente verdadeira ou falsa, ela varia em grau de verdade de 0 a 1, o que leva a ser parcialmente verdadeira ou parcialmente falsa. Trabalhar com lógica difusa proporciona algumas vantagens: [Joh22, Ans88]

- Possibilita melhor tratar as imprecisões;
- Existe uma maior facilidade em especificar as regras de controle com a linguagem próxima à natural, isto é, reflete o que as pessoas pensam;
- Possibilidade de trabalhar com variáveis linguísticas, aproximando assim do pensamento humano;
- Simplifica a aquisição da base do conhecimento e a solução de problemas;
- Faz uso de poucas regras, valores e tomadas de decisão.

Como citado, a lógica difusa é uma técnica que se baseia em graus de pertinência (verdade), tendo os valores 0 e 1 como extremos e incluindo estados de verdade entre estes valores. Sendo assim, suponha que se tenha as seguintes proposições de altura:

- Alta $\geq 1,85m$;
- Baixa: $< 1,85m$.

Existem 3 pessoas com alturas diferentes: pessoa A com 1,55 m de altura, pessoa B com 1,84 m e a pessoa C com 1,85 m.

Se comparar as alturas das pessoas com as proposições, a pessoa B é considerada baixa por ter 1cm a menos. Esta análise deve-se ao fato de que o processo de inferência considera que 2 pessoas com uma diferença de 1cm sejam classificadas de forma diferente em relação a sua estatura. Uma forma de resolver esta questão é modelar o raciocínio real considerando a pertinência da pessoa B não em apenas um conjunto, mas sim nos dois ao mesmo tempo, estabelecendo um grau diferente para cada um destes conjuntos.

Para estabelecer o grau de quanto a pessoa B pertence a cada um desses conjuntos:

- Conjunto de pessoas altas - 0,95
- Conjunto de pessoas baixas – 0,05

Se analisar os valores acima, pode-se citar que a pessoa está para mais alta que baixa, mas ao mesmo tempo não se pode dizer que ela é totalmente alta ou totalmente baixa. Desta forma, os extremos não foram aplicados, aplicando a ideia básica de "Fuzzy Logic" no qual o resultado ficou entre o intervalo [0,1].

6.1.1 Fuzzy Sets – Conjuntos Difusos

Conjuntos difusos são aqueles que possuem limites imprecisos. Por exemplo, um conjunto difuso A definido em um universo X é definido por uma função de pertinência μ_A, que mapeia os elementos de deste universo X em um intervalo [0,1], isto é, todos os valores que estiverem na range de $0 \leq \mu \leq 1$, representada por: [Jan07, Zad65, Zad78]

$$\mu_{A:X} \rightarrow [0, 1]$$

A função de pertinência associada a cada elemento y, pertencente ao universo X, é um número inserido intervalo [0,1], representa o grau de pertinência do elemento y ao conjunto A, isto é, o quanto é verdadeiro que o elemento y pertença ao conjunto A. Logo, pode-se dizer que uma sentença pode ser dita parcialmente verdadeira e parcialmente falsa. A função de pertinência $\mu_{A(X)}$ indica o grau de compatibilidade entre o universo X e os valores que representam A. [Ans88]

- $\mu_{A(x)} = 1$ – este caso indica que x é plenamente compatível com A;
- $\mu_{A(x)} = 0$ - este caso indica que x é plenamente incompatível com A;
- $0 < \mu_{A(x)} < 1$ - este caso indica que x é parcialmente compatível com A com o grau de pertinência $mu_{A(x)}$.

Desta forma, pode-se dizer que um conjunto difuso é caracterizado pela sua função de pertinência, e sendo definido formalmente como conjunto A em um universo X expresso como um conjunto de pares ordenados conforme se segue:

$$A = \{(x, \mu_{A(x)}) | x \in X\}$$

6.1.2 Crisp Sets - Conjunto Crisps

Difuso sugere a visão de uma zona limite ao invés de uma fronteira brusca ou precipitada. Na verdade, os lógicos difusos falam de conjuntos clássicos como conjuntos crips, para distingui-los dos conjuntos difusos. Tal como acontece com os conjuntos crisps, não existe uma base para decidir quais objetos são membros e quais não são. [Jan07, Ans88]

Suponha que X seja uma coleção de objetos denotados genericamente por x_i, dados discretos, no qual X é chamado de universo de entrada crisps e x_i representa uma entrada crisp deste conjunto. Sendo assim, pode-se afirmar que um conjunto de entradas nítidas W no universo X é caracterizado por sua função de pertinência $\mu W : X \rightarrow \{x_1, x_2, ..., x_n\}$., sendo que o conjunto dessas entradas nítidas pode ser representado por números ou letras. [Jan07, Ans88]

A figura 6.1 apresenta bem essa relação entre de fronteira entre abrupta existente em conjuntos crisps e não em conjunto difusos, além de visualizar que os valores crisps são valores discretos, enquanto que os valores difusos assumem valores entre [0,1].

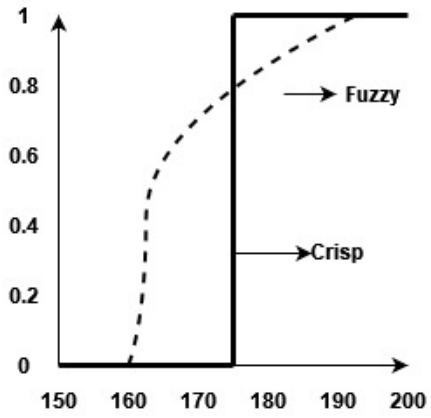

Figura 6.1: Fuzzy X Crisp [Jan07]

Sabe-se que a definição de um conjunto fuzzy estende a definição de um conjunto clássico, porque os valores de pertinência μ são permitidos no intervalo $0 \leq \mu \leq 1$, e quanto maior o valor maior será a pertinência. Um conjunto clássico é consequentemente um caso especial de conjunto fuzzy, com valores de pertinência restritos a $\mu \in \{0, 1\}$, e por pares $(x, \mu(x)$.

Seguem alguns exemplos de conjuntos que podem ser descritos por funções de pertinência difusas.[Jan07]

- O conjunto de altas temperaturas, o conjunto de ventos fortes ou o conjunto de dias agradáveis são conjuntos difusos nos boletins meteorológicos.
- O conjunto dos jovens. Um bebê de um ano será claramente (crisp) um membro do conjunto dos jovens, e uma pessoa de 100 anos não será um membro deste conjunto. Uma pessoa de 30 anos pode ser jovem na proporção de 0,5 (difuso).
- As estradas de ferro dinamarquesas permitem que crianças com menos de 15 anos viajem pela metade do preço. É assim definido como adulto o conjunto de passageiros com idade igual ou superior a 15 anos. Pela sua definição, o conjunto de adultos é um conjunto crisp.
- Um predicado pode ser crisp, mas percebido como difuso: alguns motoristas consideram um limite de velocidade de 60 quilômetros por hora como uma faixa elástica de velocidades mais ou menos aceitáveis, digamos, entre 60 e 70 quilômetros por hora. A lei de trânsito é clara (crisp), mas a compreensão desta por parte dos motoristas é difusa.

6.1.3 Variáveis Linguísticas

São variáveis cujos valores são nomes de conjuntos fuzzy, como por exemplo, a temperatura de um determinado processo, no qual uma variável linguística poderá assumir

valores como: baixa, média, e alta. Sua principal função é prover uma forma aproximada de representar valores de processos considerados complexos ou mesmo não bem definidos.

A essência das variáveis linguísticas e aplicar a linguagem usada pelo ser humano, permitindo um melhor entendimento dos sistemas os quais serão manipulados. As variáveis linguísticas podem ser representadas de formas específicas a partir de termos primários, alto, baixo, médio, ...; conectivos lógicos: não, e, ou, ...; ou por modificadores: muito, pouco, extremamente, ... Vale a pena ressaltar que estes valores são descritos por intermédio de conjuntos fuzzy, representados por funções de pertinência [Tan80]

Uma variável linguística pode ser representada da seguinte forma: [Tan80]

(N, T(N), X, G, M)

- N: nome da variável (ex: ameaça)
- T(N): conjunto de termos de N, ou seja, o conjunto de nomes dos valores linguísticos de N (ex: baixa, média, alta)
- X: universo de discurso (ex: 1 a 5 – quantificação da ameaça)
- G: regra sintática para gerar os valores de N como uma composição de termos de T(N), conectivos lógicos, modificadores e delimitadores (ex: muito alta, muito baixa)
- M: regra semântica, para associar a cada valor gerado por G um conjunto fuzzy em X - associa o valor a um conjunto fuzzy cuja função de pertinência exprime o seu significado.

6.1.4 Membership Function – Função de Pertinência

A função de pertinência visa apresentar o conhecimento que se tem em relação a veemência com que o objeto pertence ao conjunto fuzzy. A função de pertinência tem como características medidas subjetivas; são funções não probabilísticas monotonicamente crescentes, decrescentes ou subdividida em parte crescente e parte decrescente. [Jan07, Ian12]

As funções de pertinência podem possuir diferentes formas, dependendo do que se deseja representar e onde serão utilizadas. Normalmente elas são definidas a partir da experiência e da perspectiva do usuário, mas são utilizadas funções de pertinência padrão como: triangular, trapezoidal, sino generalizada e gaussiana. [Jan07, Tan80]

I - Função de Relevância Triangular (Figura 6.2).

$$\mu S \begin{cases} 0, & if\ \mathsf{x} \leq \mathsf{a} \\ (x-a)/(m-a), & if\ a < x \leq m \\ (b-x)/(b-m), & if\ m < x \leq b \\ 0, & if\ x > b \end{cases}$$

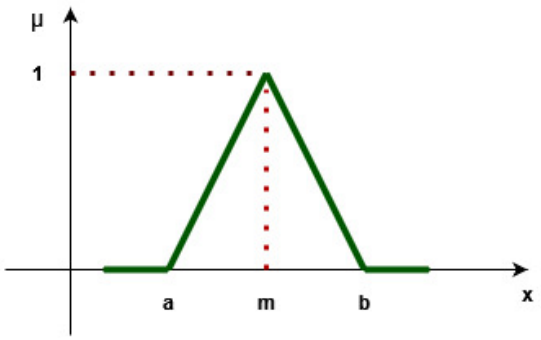

Figura 6.2: Função Triangular - Adaptado de [Jan07]

II - Função de Relevância Trapezoidal Relevance Function (Figura 6.3).

$$\mu S \begin{cases} 0, & if\ \mathsf{x} \leq \mathsf{a} \\ (x-a)/(m-a), & if\ a < x \leq m \\ 1, & if\ m < x \leq b \\ (b-x)/(b-m), & if\ m < x \leq b \\ 0, & if\ x > b \end{cases}$$

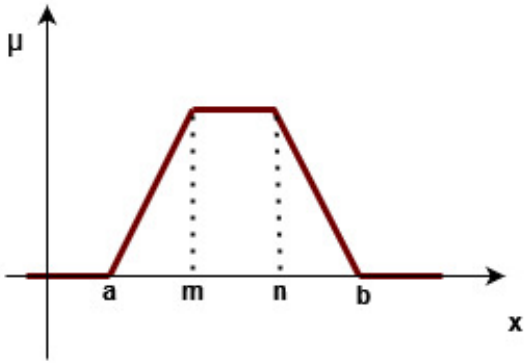

Figura 6.3: Função Trapezoidal - Adaptado de [Jan07]

III - Função de Relevância Gaussiana (Smooth Triangular) (Figura 6.4).

$$f(x) = e^{-\frac{1}{2}(\frac{x-c}{a})^2}$$

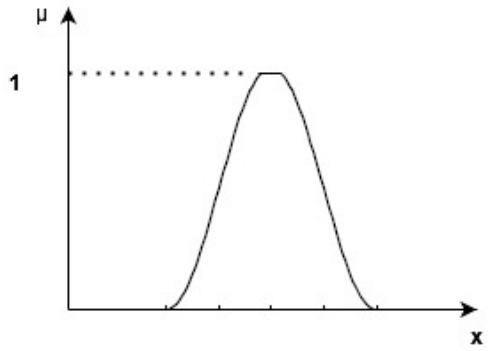

Figura 6.4: Função Gaussiana [Jan07]

IV - Função de Relevância Sino - Smooth Trapezoidal (Figura 6.5).

$$\mu_S T(x,a,b,c,d) = \begin{cases} 0 & ,x \leq a \\ \frac{1}{2} + \frac{1}{2}\cos(\frac{x-b}{b-a}\pi) & ,a \leq x \leq b \\ 1 & ,b \leq x \leq c \\ \frac{1}{2} + \frac{1}{2}\cos(\frac{x-c}{d-c}\pi) & ,c \leq x \leq d \\ 0 & ,d \leq x \end{cases}$$

Sendo que $X \in \mathbb{R}$

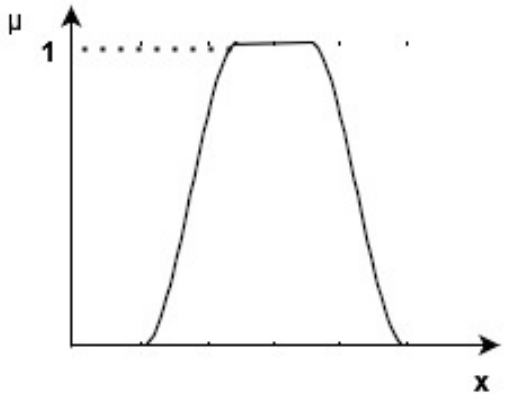

Figura 6.5: Função Sino [Jan07]

6.2 Controlador Lógico Difuso - FLC (Fuzzy Logic Controller)

Como é de conhecimento, o controle difuso apresenta uma técnica para representar, manipular e implementar o conhecimento humano de como controlar um sistema. O FLC possibilita a conversão da estratégia de controle linguístico, baseada em conhecimento, em regras de controle e da combinação da teoria da lógica difusa com processos de inferência. O FLC é de grande utilidade principalmente quando os modelos não são conhecidos ou muito complexos para análise com técnicas quantitativas convencionais [IC09].

O FLC é derivado da teoria de controle baseada em modelos matemáticos do processo em malha aberta (sinal de entrada é um sinal pré-setado, baseado em experiências passadas, de forma que o sistema forneça o sinal de saída desejado [DM08]) a ser controlado. O FLC vem sendo, de forma efetiva, aplicados em processos considerados complexos e mal definidos, especialmente aqueles que podem ser controlados por um operador humano qualificado sem o conhecimento de sua dinâmica subjacente [Ian12].

Comparados por Kosko em [Kos92] o FLC aprende por amostragem, sendo que possibilita estimar uma função sem a necessidade de uma descrição matemática de como a saída depende da entrada. Os sistemas FLC são considerados com bom desempenho, analisados por Wang em [Wan92], e que são capazes de aproximar uma função real contínua num conjunto compacto com uma determinada precisão.

Um FLC possui um conjunto de regras da forma SE (um conjunto de condições é satisfeito) ENTÃO (um conjunto de consequências pode ser inferido), que compõe o seu conhecimento. Como os antecedentes e os consequentes dessas regras SE-ENTÃO estão associados a conceitos difusos (termos linguísticos), eles são frequentemente chamados de declarações condicionais difusas [Ian12, IC09].

Sendo assim, uma regra de controle de controle difusa é uma declaração condicional difusa no qual o antecedente é uma condição em seu domínio de aplicação e o consequente é uma ação de controle para o sistema. Suponha que uma regra difusa tenha a condição que a porta de destino (P_D) possua uma condição de possível ameaça (Almost) e o endereço Ip de destino (Ip_D) é um provável destino de ataque (Really), a regra pode ser representada como:

IF P_D is Almost $\vee$ Ip_D is Really THEN Threat is Really

O sistema FLC é composto por 4 (quatro partes), interface de fuzificação, base de regras fuzzy, mecanismo de inferência fuzzy e interface de defuzificação como apresentado na figura 6.6.

As entradas de sistemas baseados em regras fuzzy devem ser dadas por conjuntos fuzzy, devendo fuzificar as entradas "crisps". A saída de um sistema fuzzy é sempre um conjunto fuzzy, devendo defuzificar para se obter valores "crisps". Desta forma, o processo de fuzificação tem como objetivo transformar entradas "crisps"em conjuntos fuzzy. A figura 6.7 apresenta a fuzificação, no qual $X_0 \in U$ é fuzificado em X_0 [IC09].

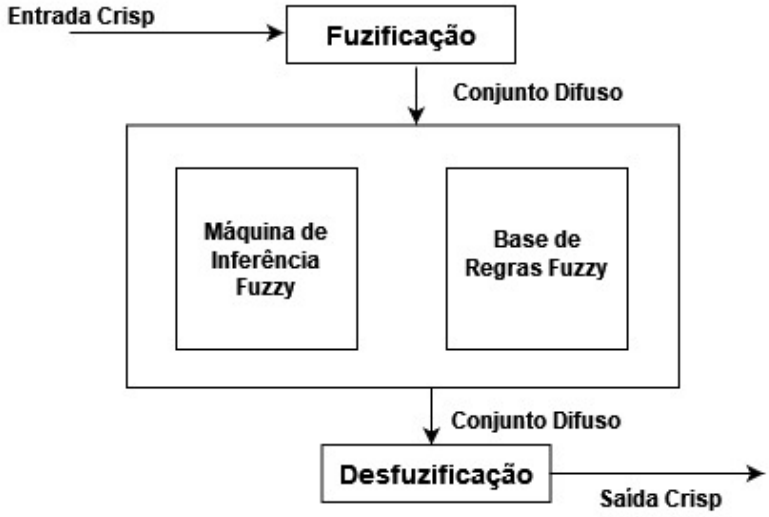

Figura 6.6: Controlador Lógica Difusa [Ian12]

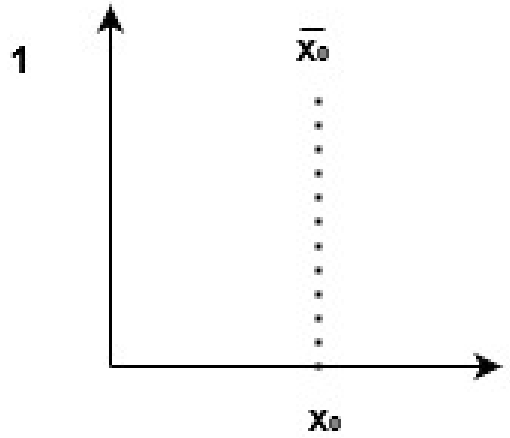

Figura 6.7: Processo Fuzificação [IC09]

Uma outra forma de visualizar o processo de fuzificação [Ian12]:

$$\text{fuzificação } (X_0) = \overline{X_0}$$

$$\mu_{\overline{X_0}}(X) = \begin{cases} 1, & for\ X = X_0 \\ 0, & for\ X \neq X_0 \end{cases}$$

O procedimento utilizado pela máquina de inferência fuzzy com a finalidade de obter uma saída fuzzy possui as seguintes etapas:

- Encontre o nível de disparo de cada regra;
- Encontre a saída de cada regra;

- Agregar os resultados das regras que correspondem a uma implicação, com o objetivo de se ter o resultado global; e
- Combinar os valores anteriores com o objetivo de obter o resultado do sistema.

Por fim, vem o processo de defuzificação, no qual é necessário quando se espera que o sistema retorne um número, por exemplo, e não o conjunto fuzzy criado para cada consequente. Nesta etapa, a saída da inferência fuzzy (que é a região resultante do consequente), baseada em funções e regras de pertinência, é convertida para um valor "crisp". Os processos de defuzificação mais utilizados para um conjunto fuzzy em um universo são [Ian12, IC09]:

- Center-of-Gravity (Centroide) - o valor numérico obtido representa o contro de gravidade da distyribuição de possibilidade de saída do sistema fuzzy.
  1. Determinar abcissa do ponto centroide para cada saída ativada na referência.
  2. Calcular a área entre o grau de pertinência e o eixo das ordenadas para cada saída ativada.
  3. Calcular a média ponderada dos pontos centrados pelas respectivas áreas.

$$Z_0 = \frac{\sum_{j=1}^{N} Z_j \mu_C(Z-j)}{\sum_{j=1}^{N} \mu_C(Z_j)}$$

- Middle-of-Maxima - o valor que sofreu defuzificação é definido como o meio de todos os valores de um universo.

$$Z_0 = \frac{1}{N-1} \sum_{j=1}^{N-1} Z_j, N_1 \leq N$$

- Max-Criterion - este método escolhe um valor arbitrário do conjunto de maximização C.

$$Z_0 \in \{\frac{Z}{\mu_C(Z)} = max\mu_C(v)\}, v \in V$$

Sendo que $Z = \{z_1, ..., z_N\}$ é um conjunto de elementos de um universo V.

6.3 Tipos de FLCs

Os controladores fuzzy, de forma geral, são classificados em dois grandes grupos com base no processo de tomada de decisão: aqueles que se baseiam nas funções de implicação fuzzy e em operadores de composição de composição de definição de saída fuzzy; e aqueles que não trabalham que não trabalham com a definição de funções de implicação e operadores para a inferência [AJ08].

Os controladores do tipo Mamdani [MA75] e Tsukamoto [GT80] tem o seu processo de decisão com base em implicação fuzzy, primeiro grupo, e Sugeno [VA16] tem o seu processo de decisão baseado no segundo grupo. Em ambos tipos de controladores a ação é realizada por um conjunto de regras de controle fuzzy, um algoritmo fuzzy. As regras são generalizadas e cada um pode ser representada por um antecedente SE "x é A e y é B", e um consequente, ENTÃO "z é C".

6.3.1 Sistema Mamdani - FLC

O sistema Mamdani, é uma técnica de inferência difusa, proposto por Mamdani e Assilian em [MA75], com o objetivo inicial de imitar o desempenho de operadores humanos encarregados de controlar alguns processos industriais, resumindo um conjunto de controles linguísticos em regras. A ideia de resumir a experiência do operador em um conjunto de regras (linguísticas) SE-ENTÃO possibilitaria a sua utilização por uma máquina que controlasse automaticamente o processo [GT80].

Desta forma, o uso de um conjunto de regras SE-ENTÃO, o sistema fuzzy Mamdani define uma função f que gera saídas crisps a partir de valores de entrada (entradas crisps). Sendo as regras SE-ENTÃO da forma "SE X é A ENTÃO Y é B", como "SE A PORTA DE DESTINO É 38000 ENTÃO O FLUXO DE DADOS IoT PODE TER UMA AMEAÇA". A parte SE "X é A" é chamada de antecedente da regra, e a parte ENTÃO "Y é B" é chamada de consequente da regra, Y=F(X).

Suponha que X e Y sejam número da porta de destino e o valor numérico para possível ameaça, respectivamente. Sendo assim, a variável X é definida como um intervalo real, intervalo de entrada, e a variável Y definida como intervalo de saída. As letras x e y minúsculas são valores específicos das variáveis X e Y [SI18].

Os símbolos A e B são termos linguísticos modelados como conjuntos fuzzy definidos nos intervalos de entrada e saída. O conjunto fuzzy A é definido por uma função de pertinência μA que atribui um valor real $\mu A(x)$ no intervalo entre 0 e 1 para cada valor x no intervalo de entrada. Desta forma, o valor $\mu A(x)$ é conhecido como o grau de pertinência do elemento x no conjunto fuzzy A. Logo, se o conjunto fuzzy A representa um conceito (suspeita), então $\mu A(x)$ é interpretado como um valor verdade quando X é A, isto é, uma porta de destino é suspeita, sempre que X=x. A mesma sistemática também é aplicada conjunto fuzzy B, definido por uma função de pertinência μB que atribui um valor real $\mu B(y)$ entre 0 e 1 a cada valor real y no intervalo de saída [GT80].

Os sistemas Mamdani, na sua maioria são compostos por várias regras SE-ENTÃO, no qual cada uma pode usar diferentes conjuntos fuzzy A e B. Os antecedentes e consequentes também podem ser proposições combinadas que incluem os conectivos lógicos E ou OU. Um sistema Mamdani padrão usa as seguintes operações para calcular o valor verdade de proposições combinadas [SI18, Ian12, GT80]:

$T(A \wedge B) = \min(T(A),T(B)) \Rightarrow \mu_{A \cap B} = \min\{\mu_A(x), \mu_B(x)\}$; and

$T(A \vee B) = \max(T(A),T(B)) \Rightarrow \mu_{A \cup B} = \max\{\mu_A(x), \mu_B(x)\}$

6.4 Considerações

Este capítulo descreveu o que vem a ser lógica difusa e seus componentes, ainda apresentando o controlador lógico difuso, componente básico do threat analyser, módulo do RTRMM. O próximo capítulo irá abordar o que é teoria dos jogos.

7

Teoria dos Jogos

Neste capítulo apresentamos os conceitos de teoria de jogos, além de como um jogo se comporta e a sua adequação ao processo de análise e avaliação de riscos. O capítulo está dividido em quatro seções, que descreve o que é teoria dos jogos; o que é um jogo; a solução de um jogo; e a análise e avaliação de riscos com base em teoria dos jogos.

7.1 O que é Teoria dos Jogos?

A Teoria dos Jogos pode ser vista como uma teoria Matemática aplicada na construção de modelos que descrevem fenômenos ou situações de conflito ou de cooperação entre dois ou mais agentes / tomadores de decisão interagem entre si, isto é, ela estuda a escolha de decisões ótimas sob condições de conflito. A teoria dos jogos apresenta uma linguagem para descrever processos de decisão que envolvam dois ou mais elementos [SGB+04, Pim21].

7.2 Um Jogo

Um jogo, por si possui objetivos, jogadores, regras e ganhadores e perdedores, por exemplo. Todavia, na teoria dos jogos um jogo pode ser representado formalmente por 3 (três) elementos: [Pim21]

- Um conjunto finito de jogadores $G = \{g_1, g_2, ..., g_n\}$;
- Conjunto de estratégias $S_1, S_2, ..., S_n$ no qual cada jogador $g_i \in G$ possui um conjunto de estratégias $S_i = \{s_{i1}, s_{i2}, ..., s_{in}\}$;
- funções de ganho / recompensa (payoff) $u_1, u_2, ..., u_n$, no qual cada jogador g_i possui uma função de ganho $u_{gi} : S_1 X S_2 X ... X S_n \rightarrow \mathbb{R}$. Para cada configuração de estratégias dos jogadores $s = (s_1, s_2, ..., s_n \in S_1 X S_2 X ... X S_n$ um valor u_{gi} é associado a um jogador j.

Cada jogador g_i, participante do jogo, possui o seu conjunto de estratégias, sendo que com a estratégia escolhida, passa a existir uma situação ou perfil no espaço de todas as situações possíveis. Cada jogador possui interesses (preferências) para cada situação, e tem uma função que é atribuído um número real, chamado de ganho (payoff) para cada situação de jogo.

7.2.1 Ação, Resultado e Preferência

Para melhor esclarecer a atitude de um jogador, precisa inicialmente entender o que é um problema de decisão consiste. Falando de forma simples, um problema de decisão consiste em realizar uma escolha em 2 ou mais opções, por exemplo: sair para correr na chuva ou ficar em casa e beber um chocolate quente. Claro que este problema é simples de resolver, mas um jogador (indivíduo) está frente a uma situação, uma escolha (ação) deverá ser realizada, proporcionando um resultado (satisfação, preferência) para cada escolha realizada. [SGB$^+$04]

Para modelar um problema, 2(duas) perguntas são interessantes d serem realizadas:

- Quais as ações são possíveis - ações são as escolhas (estratégias) de um jogador; e
- O quanto o resultado (preferência) irá satisfazer o jogador em de cada ação - descreve o quanto um jogador está satisfeito com o resultado da ação. Por exemplo, sejam as ações A e B, se $A \geq B$, então a ação A é ao menos tão boa quanto B.

7.2.2 Função de Ganho / Recompensa (Payoff)

A função de Payoff representa a a relação de preferência de um jogador para um par de ações A e B. Se uma função de ganho $u(A) \geq u(B)$ se somente se a ação $A \geq B$. Se usar funções payoff as análises serão mais operacionais, onde um jogador optará sempre por uma ação que maximize a função de payoff.[SGB+04]

7.2.3 Modelando um Problema

Um exemplo interessante de jogo, apresentado em [SGB+04], é a batalha dos sexos, onde um homem deseja ver um jogo de futebol e sua esposa ir ao cinema. Se ambos forem ver o jogo de futebol, o homem possui uma maior satisfação que a esposa. Caso contrário, ambos forem ao cinema, a esposa terá uma satisfação maior que o homem. Finalmente, se não forem a nenhum lugar ambos ficarão insatisfeitos.

G = {homem, mulher}, $S_{homem} = \{futebol, cinema\}$, $S_{mulher} = \{futebol, cinema\}$

S = {(futebol.futebol), (futebol,cinema), (cinema,futebol), (cinema,cinema)}

$$u_{homem} : S \to \mathbb{R} \text{ e } u_{mulher} : S \to \mathbb{R}$$

$u_{homem}(futebol, futebol) = 10$, $u_{homem}(futebol, cinema) = 0$,

$u_{homem}(cinema, futebol) = 0$, $u_{homem}(cinema, cinema = 5)$

$u_{mulher}(futebol, futebol) = 5$, $u_{mulher}(futebol, cinema) = 0$,

$u_{mulher}(cinema, futebol) = 0$, $u_{mulher}(cinema, cinema = 10)$

HOMEM	MULHER		
		FUTEBOL	CINEMA
	FUTEBOL	(10,5)	(0,0)
	CINEMA	(0,0)	(5,10)

Tabela 7.1: Matriz Homem X Mulher
[SGB+04]

7.3 Solução do Jogo

Um solução de um jogo visa apresentar uma previsão do resultado do jogo, no qual dois conceitos comuns de solução são conhecidos: dominância e equilíbrio de Nash [Cam15].

7.3.1 Dominância

Se dois jogadores possuem estratégias dominantes, isto significa que todas as estratégias, menos uma é dominada. Desta forma,o jogo é resolvido por dominância, encerrando o jogo com uma solução que é um equilíbrio de estratégia dominante.

Uma estratégia $sik \in S_i$ de um jogador $g_i \in G$ é dominada por outra estratégia $s_{ik'} \in S_i$ se $u_i(S_{ik'}, S_i) > u_i(Sik, S_i)$. A dominância consiste somente no processo de eliminar as estratégias que são dominadas. [SGB$^+$04]

Uma forma de enxergar a estratégia dominante é o dilema do prisioneiro, onde é um jogo não cooperativo, com a estratégia dominante para os 2 (dois) jogadores, no qual a solução é conhecida como equilíbrio de estratégia estritamente dominante. Este equilíbrio de estratégia é também chamada de ótimo de Pareto, de forma que a estratégia definida para os 2 (dois) jogadores não é dominada por qualquer uma das partes. [Cam15]

7.3.2 Equilíbrio de Nash

A maioria dos jogos não se limita a um equilíbrio de estratégia estritamente dominante (equilíbrio de um jogo não-cooperativo), isto é, a melhor estratégia de um jogador é diferente do outra. Existe uma outra forma de equilíbrio que não se limita somente a uma estratégia, o equilíbrio de Nash. O equilíbrio de Nash em um jogo consiste em um ponto no qual cada jogador não tem incentivo de mudar a sua estratégia caso os demais não a façam. [Cam15, Mye99]

Considere um jogo formado por um conjunto de jogadores $G = \{g_1, g_2, ..., g_n\}$, no qual cada jogador g_i com um conjunto de estratégias S_i e função payoff $u_i : S_1 X ... X S_n \to \mathbb{R}$. [SGB$^+$04, Pim21]

Desta forma, o perfil de estratégia $s = (s_1, s_2, ..., s_n) \in S$ é um equilíbrio de Nash se somente se s_i for a melhor resposta de $s{-}i$ para qualquer $g_i \in G$:

$$u_i(s_i, s{-}i) \geq u_i(s_i, s{-}i) \text{ para qualquer } s_i \in S_i \text{ e qualquer } g_i \in G.$$

A maioria dos jogos não se limita a um equilíbrio de estratégia estritamente dominante (equilíbrio de um jogo não-cooperativo), isto é, a melhor estratégia de um jogador é diferente do outra, existe também uma outra forma de equilíbrio que não se limita somente a uma estratégia, o equilíbrio de Nash. O equilíbrio de Nash em um jogo consiste em um ponto no qual cada jogador não tem incentivo de mudar a sua estratégia caso os demais não a façam. [SGB$^+$04, Cam15]

7.4 Análise e Avaliação de Riscos e a Teoria dos Jogos

A maioria das metodologias de uma análise e avaliação de riscos, como a ISO 27005 [ISO22b], faz uso de relacionados somente ao defensor, vulnerabilidades, controles, his-

tórico de ataques, por exemplo e fatores relacionados ao atacante, como as ameaças que ele pode proporcionar. O risco é calculado com base nesses fatores e na sua maioria de forma qualitativa.

A ideia de trabalhar com análise e avaliação de riscos com teoria de jogos se deve ao fato de ambas serem complementares. Em [Cox09] cita que as formulações da teoria dos jogos de conflitos de ataque e defesa podem melhorar as técnicas de análise de risco existentes que visam modelar as decisões dos atacantes como variáveis aleatórias ou atributos incertos dos alvos (ameaças) e que procuram obter valores via avaliação de dados fornecidos pelo defensor.

A melhora na técnica de análise e avaliação de riscos se deve ao uso de modelos de teoria dos jogos que visam esclarecer a natureza das decisões interativas tomadas por atacantes e defensores. Esta técnica possibilita a distinção entre escolhas as estratégicas (as decisões a serem tomadas) e as variáveis aleatórias (ameaças e vulnerabilidades, fatores do acaso, as vezes não controlados pelo atacante ou pelo defensor), podendo produzir resultados mais sensatos e recomendações mais adequadas e eficazes de mecanismos de defesa no processo de gestão de risco.

7.4.1 Tomadas de Decisão

Segundo [Cox09], um dos grandes problemas na análise/avaliação de riscos consiste em definir as diferentes probabilidades referentes às consequências de estratégias de atacantes e defensores aplicadas. Esta informação é fundamental no processo para quem está defendendo, pois afeta diretamente na decisão do que fazer para proteger os ativos. A aplicação da teoria dos jogos neste processo facilita analisar abordagens consideradas simples ao processo de gestão de riscos, como àquelas que ignoram a capacidade dos atacantes responderem às ações dos defensores. Utilizar técnicas para analisar os riscos via uma avaliação probabilística de riscos como em [Hau02], é de extrema validade, principalmente para informações relacionadas durante um ataque e o resultado de suas consequências.

Montagem da Matriz Para montar a matriz de payoff, o par de estratégia pura do atacante e defensor deve ser baseada em modelos de simulação de riscos que preveem os resultados das estratégias aplicadas (por exemplo: mecanismos de segurança implantados), além de fatores que não são controlados diretamente (por exemplo: desempenho da equipe de defesa e ataque). Para estruturar uma matriz payoff deve:

- Conhecer bem o seu ambiente computacional, para que possa determinar as vulnerabilidades que ele possui, e também conhecer as consequências (impacto) de quando ocorrer a efetivação de ameaças (ataques);

- Saber a eficácia de todos os mecanismos de segurança implementados, informações obtidas por monitoramento constante e testes de penetração em seu ambiente;

- Manter uma base de conhecimento atualizada das possíveis ameaças que seu ambiente possa vir sofrer; e
- Manter a equipe de defesa e ataque bem treinada e atualizada quanto às técnicas de atacantes. Realizar simulados com o objetivo de testar a destreza e eficiência de sua equipe.

De posse dessas informações, a distribuição de probabilidades de que um atacante possa ter sucesso, isto é, que uma ameaça se efetive em um ataque com sucesso, pode ser mais bem definida. É claro que não é simples formular na prática o universo atacante/defensor, os diversos problemas do mundo real, como alguns já citados, surgem como dificuldades na análise de riscos, determinar e quantificar os riscos existentes. A dificuldade também se estende para avaliação de riscos, pois determinar o que será tratado, e quais mecanismos poderão ser aplicados é fator crítico de sucesso.

Desta forma, determinar de forma explícita quais estratégias que cada jogador irá adotar acaba sendo um tanto complexo, pois todos estes elementos citados deverão ser levados em consideração. Todavia, trabalhar com elementos básicos como: o impacto de um dispositivo IoT ficar inoperante em função do ataque; analisar as probabilidades de uma ameaça ter sucesso; e as vulnerabilidades que o sistema IoT possui podem ser elementos que venham facilitar estabelecer os pares de estratégias puras do defensor e atacante. Para isso, os métodos de estimar e avaliar estes valores ficam a encargo dos analistas de risco.

A matriz payoff genérica, representada na tabela 7.2, apresenta uma estratégia mista: ataque A com probabilidade

$$(d1 - b1)/[(d1 - b1) + (a1 - c1)],$$

e defenda A com probabilidade

$$(d2 - c2)/[(d2 - c2) + (a2 - b2)]$$

Assumindo que os valores se encontram entre [0 e 1].

		ATACANTE	
		ATACANTE A	**ATACANTE B**
DEFENSOR	**DEFENSOR A**	a_1,a_2	b_1,b_2
	DEFENSOR B	c_1,c_2	d_1,d_2

Tabela 7.2: Matriz Payoff Genérica, [Cox09]

Sendo assim, os modelos e métodos da teoria dos jogos podem ajudar na análise e avaliação de riscos, possibilitando que se trabalhe de forma mais clara e eficaz sobre os riscos existentes, esclarecendo o que deve ser modelado como variáveis de decisão para diferentes jogadores e o que deve ser modelado como variáveis de acaso ou consequência.

7.5 Considerações

Este capítulo abordou os fundamentos sobre teoria dos jogos; o que é um jogo e como se obtém uma solução do jogo. Também foi descrita como a análise e avaliação de riscos pode ser realizada via a teoria dos jogos.

8

RTRMM – Real Time Risk Management Model

Sistemas IoT crescem e se tornam cada vez mais dinâmicos. A diversidade e as mudanças tecnológicas de dispositivos, bem como sua estrutura física e lógica, provêm a todo momento novas topologias e soluções, dando uma ampla mobilidade aos seus usuários. Em contrapartida a toda esta evolução vem a segurança da informação, pois esta ainda não consegue acompanhar com a mesma velocidade todo este crescimento IoT [RRM+19].

A necessidade de manter as propriedades de segurança (ex: confidencialidade e integridade) durante a trâmite de dados entre dispositivos e/ou seu armazenamento, além da confiança via a disponibilidade dos dispositivos e dados do sistema IoT passou a ser um fator sucesso para qualquer implementação IoT. Porém, existem dificuldades em especificar e implementar uma solução de segurança que atenda à toda essa realidade, podendo destacar as características físicas e lógicas dos dispositivos IoT, a diversidade de tecnologias, e da capilaridade que se altera conforme a demanda dos serviços.

Durante o desenvolvimento e confecção deste trabalho, foram constatados que alguns aspectos que dificultam a implantação de uma solução de segurança capaz de atender de forma plena às necessidades de um sistema IoT, pois possuem:

- Baixa capacidade de processamento e armazenamento interna.
- Normalmente fazem uso de criptografia fraca para garantir a confidencialidade e integridade de dados.
- Existe a necessidade de autenticação dos dispositivos IoT para comunicarem entre si ou com suas bases de dados.
- Diversidade de fabricantes. Esta caraterística faz com que cada fabricante possua os seus próprios protocolos de comunicação e mecanismos proprietários de segurança.

No âmbito destes fatores, em conjunto com a diversidade de tecnologias, dinâmica e outros mais, este trabalho procura apresentar uma estratégia que pudesse tornar os sistemas IoT mais confiáveis para seus usuários, minimizando vulnerabilidades e possibilitando que as propriedades de segurança (confidencialidade, integridade e disponibilidade) sejam mais bem atendidas.

Durante o nosso estudo, constatamos que uma solução única de segurança da informação não atende a todas as necessidades de um sistema IoT. Desta forma, viu-se a necessidade de buscar um novo formato, uma estratégia de segurança compatível com as necessidades de ambientes IoT.

Uma estratégia, que de forma ampla, conhecesse os riscos que um ambiente IoT possui, proporcionando o conhecimento necessário para determinar quais mecanismos de segurança possam ser aplicados, tornando o ambiente adequado às suas necessidades e de seus usuários. Sendo assim, este trabalho adotou a estratégia de gerir os riscos de segurança da informação, com o objetivo de reduzir e manter os riscos a um nível considerado adequado para o funcionamento de um ambiente IoT.

Gerir os riscos de segurança consiste em um monitoramento e controle dos riscos que sejam detectados e calculados, possibilitando então que mecanismos de segurança, mais adequados, sejam aplicados, reduzindo a probabilidade de ameaças explorarem as vulnerabilidades atualmente existentes. O processo de gestão de riscos pode ser realizado de duas formas básicas: reativa e proativa. A gestão de riscos reativa executa ações defensivas após os incidentes de segurança acontecidos, atendendo as necessidades de reduzir os riscos, mas mantendo o problema do tempo de resposta ao incidente de segurança, além de não prever um incidente antes que ele seja efetivado.

Foi observado que reduzir riscos a um nível aceitável, em sistemas IoT de forma reativa não seria o mais adequado. Logo, optou-se pela estratégia de gerir riscos de forma proativa. A decisão de aplicar esta estratégia vem ao encontro de reduzir a probabilidade de uma ameaça explorar uma vulnerabilidade antes do sistema IoT sofrer maiores danos. Isto porque o sistema IoT terá chances de se defender antes do dano ser causado ou mesmo reduzir os estragos que poderiam ocorrer.

Outro aspecto importante analisado para especificar uma estratégia de reduzir os riscos em sistemas IoT, foi onde e como realizar esta gestão de riscos de segurança. Para que evite que uma ameaça venha a explorar alguma vulnerabilidade em um dispositivo IoT, optou-se realizar a análise junto ao tráfego de dados existente entre os dispositivos. Esta decisão vem ao encontro do fato de detectar a ameaça antes que ela se efetive.

A segunda questão, como especificar e implementar, foi talvez a mais fácil de responder. A opção foi por desenvolver um novo modelo de gestão de riscos, que trabalhasse em tempo real, e que atendesse as necessidades de segurança dos sistemas IoT, como um todo. Sendo assim, com o objetivo de melhor desenvolver este modelo, foi realizado um levantamento do estado da arte, estudo e análise de metodologias como ISO 27005 [ISO22b], NIS 800-30 [KJ14], ABNT NBR ISO 31000 [ABN09], além de modelos de gestão de risco como [MS19, ZAHS19, Che19, SBH+19, HHV17, QFG12]. Estes modelos / metodologias de gestão de riscos, não são plenos quanto às suas funcionalidades, não trabalham de forma proativas, e nem em tempo real, como s funcionalidade de análise e avaliação de riscos.

Sendo assim, este trabalho vem apresentar o "Real Time Risk Management Model" (RTRMM), um novo modelo de gestão de riscos de segurança em tempo real para ambientes IoT, que visa minimizar os riscos que forem por ele detectados. O modelo apresenta uma nova proposta de gestão de riscos, com o objetivo principal de tornar os sistemas IoT mais confiáveis aos seus usuários. O RTRMM possui uma arquitetura simples e ao mesmo tempo eficiente que busca analisar / avaliar os riscos de segurança de forma proativa, em tempo real, e propor mecanismos de segurança reduzindo assim a probabilidade das ameaças explorarem as vulnerabilidades de forma efetiva (ataque).

Desta forma, o RTRMM tem como objetivo ser um modelo de gestão de riscos, com funcionalidades e tecnologias inovadoras, podendo vir a ser um modelo de referência para sistemas IoT, propondo uma nova forma, ou mesmo estratégia de gerir riscos em sistemas IoT. Este capítulo irá apresentar a estrutura lógica do RTRMM, suas funcionalidades e uma avaliação experimental inicial, sobre um protótipo.

8.1 RTRMM vs IPS

Quando se fala em detectar ameaças em tempo real ou mesmo proteger sistemas, se pensa em sistemas de detecção ou prevenção de intrusos (IDS/IPS). Contudo, para dirimir quaisquer dúvidas de comparação do RTRMM com um IPS (Intrusion Prevention System), esta seção tem como objetivo apresentar as diferenças de funcionalidades entre os componentes e assim deixar claro que o objetivo do RTRMM é gerir riscos.

O IPS é um dispositivo de segurança da informação, que examina o tráfego de rede, com o objetivo de detectar e prevenir acessos não autorizados ao sistema computacional, protegendo de possíveis exploração das vulnerabilidades. Este dispositivo de segurança funciona de forma proativa, sendo um sistema de defesa de sistemas de rede, que trabalha combinando as técnicas de firewalls e detecção de intrusão adequadamente. O sistema IPS analisa os dados e trabalha com um sensor de reconhecimento de padrões. Ao detectar um comportamento anormal, o sistema IPS bloqueia o fluxo de dados e registra

o incidente [SAI10].

O sistema IPS pode trabalhar em uma máquina (Host), chamado de sistema de prevenção de intrusão baseado em host (HIPS) ou em rede, Sistema de prevenção de intrusão baseado em rede (NIPS) [SAI10]. O HIPS é instalado em uma máquina, tendo acesso ao sistema operacional e ao kernel, controlando os acessos ao sistema de arquivos, configuração e registros do sistema. Uma característica deste dispositivo é a possibilidade de identificar comportamentos suspeitos no sistema operacional, ao invés de comparar assinaturas, bem como a capacidade de decifrar o tráfego de pacotes, possibilitando a detecção de ataques. O NIPS pode ser instalado em um dispositivo de rede, por onde trafegam os pacotes de rede. Normalmente, quando um ataque é identificado, são realizadas decisões baseadas em regras pré-definidas. Este dispositivo pode realizar "drop" de conexões evitando que os pacotes cheguem ao seu destino.

O RTRMM, também age proativamente, mas ao contrário do IPS não é um sistema de defesa de rede com o objetivo de proteger ou mesmo evitar um ataque. O RTRMM tem como objetivo gerenciar o risco em tempo real de sistemas IoT, e para isso ele necessita detectar possíveis ameaças em tempo real. O risco gerenciado pelo RTRMM, vai desde a sua detecção, realizando a sua análise/avaliação, possibilitando posteriormente o seu tratamento. O RTRMM, ao contrário do IDS, não trabalha com base em comportamentos de tráfego, mas sim com inferência de probabilidades; ele independe de qualquer mecanismo de segurança aplicado em sistemas computacionais, podendo ser implementado como um serviço do sistema IoT, onde realiza o gerenciamento dos riscos existentes no fluxo de dados entre os clientes e os dispositivos IoT, por exemplo. O RTRMM aplica mecanismos de segurança, não ficando restrito a bloquear conexões.

8.2 Composição do RTRMM

O RTRMM é um modelo de gestão de riscos de segurança para ambientes IoT, que tem como objetivo tratar os desafios (riscos) de segurança que os sistemas IoT oferecem em tempo real. A estrutura lógica deste modelo é baseada na ISO 27005 [ISO22b], uma norma internacional de gestão de riscos, amplamente aplicada por gestores de segurança em suas Organizações.

A escolha da ISO 27005 se deve ao fato da norma possuir uma estrutura robusta de funcionalidades e especificações, e porque trata todas as etapas do processo de gestão de riscos de forma clara e objetiva, proporcionando uma arquitetura aberta a todos os ambientes computacionais, e também criando a possibilidade de incluir novas funcionalidades. Durante o estudo e análise da ISO 27005 foi observado que apesar da mesma ter sido desenvolvida, inicialmente, para sistemas computacionais diversos, ela se aplica plenamente para sistemas IoT.

Em função da necessidade de se ter resultados de análise e avaliação de riscos, além de implantar mecanismos de segurança em tempo real, a estrutura lógica do RTRMM procurou ser simples e eficiente. Para isso, fez uso somente de funcionalidades consideradas críticas para a realização destas tarefas, a fim de prover tempo de respostas compatíveis com as necessidades dos sistemas IoT.

O RTRMM foi implementado com base em programação lógica probabilística, em razão da possibilidade de que todas as informações, que serão processadas, possam ser representadas por um conjunto de regras de restrição ou aceitação ao tráfego IoT. O RTRMM viabiliza que valores de probabilidade de um tráfego IoT possuir ou não uma ameaça possam ser inferidos. Desta forma, o processo de detecção de ameaças e gerenciamento de riscos é interativo e contínuo, usando um método com base em probabilidades. Ao mesmo tempo se pode dizer que o processo também é sistemático, porque detecta e gerencia os riscos de segurança, de forma contínua e sistemática, a fim de mantê-los em níveis considerados aceitáveis (minimizem perdas e maximizem ganhos).

A composição lógica do RTRMM é calcada por um conjunto de 4 (quatro) módulos (figura 8.1): Threat Analyzer; Risk Management, Threat Category e Controls DB. Todos estes módulos são integrados e interconectados entre si, com o objetivo de detectar possíveis ameaças, analisar / avaliar riscos e prover medidas de segurança a fim de reduzir a probabilidade de incidentes que venham afetar as funcionalidades de sistemas IoT.

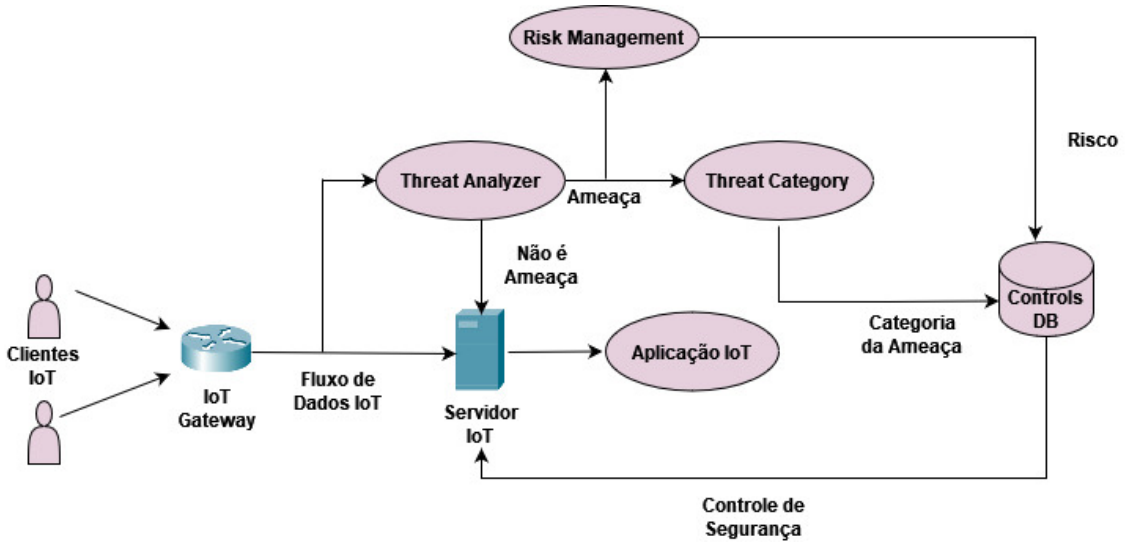

Figura 8.1: Modelo Lógico RTRMM

8.2.1 Dinâmica do RTRMM

A dinâmica do RTRMM começa no fluxo de dados proveniente dos dispositivos IoT (figura 8.1). O fluxo de dados é coletado e encaminhado ao módulo Threat Analyzer, que tem o objetivo de analisar este fluxo e verificar a existência ou não de possíveis ameaças. A análise apresenta como resultado a probabilidade de uma ameaça estar presente no fluxo de dados. A sua estratégia adotada para a análise do fluxo de dados é baseada em incertezas e probabilidades como será discutido posteriormente. O Threat Analyzer, via as informações provenientes do tráfego do dispositivo IoT, realiza os cálculos de probabilidade deste tráfego possuir ou não uma ameaça. Os valores de probabilidade calculados são analisados com o intuito de confirmar a existência ou não de uma ameaça.

Após a avaliação realizada pelo Threat Analyzer, o valor da probabilidade calculado da ameaça é encaminhado ao módulo de Risk management, responsável pela análise e avaliação dos riscos que esta ameaça poderá causar ao sistema IoT. Neste módulo é calculado, durante a fase de análise de riscos, o impacto que poderá causar ao sistema, caso venha a sofrer um incidente de segurança. Nesta fase de análise de riscos também é utilizado o valor da probabilidade proveniente do módulo Threat Analyzer, realizando o cálculo do risco existente, com base em uma função. Calculado o risco, este módulo também irá proporcionar a avaliação destes riscos, informando se os mesmos serão ou não tratados, isto é, se medidas de segurança serão aplicadas para reduzir os riscos existentes no Sistema IoT, além da prioridade de tratamento.

No caso do risco calculado pelo módulo Risk Management vir a ser tratado, ele é encaminhado ao módulo Controls DB. Este módulo possui uma base de dados com os riscos calculados correlacionados com os seus respectivos mecanismos de segurança a serem tratados. Esta base de dados é alimentada simultaneamente pelo módulo Threat Category, que informa os tipos de riscos por classe e o módulo Risk Management. Este módulo realiza a equalização e o armazenamento dessas duas informações.

Por fim, o módulo Threat Category é o responsável em classificar as ameaças em categorias, previamente estabelecidas, e as envia para o módulo Controls DB. O objetivo de classificar e por conseguinte agrupar as ameaças é facilitar e agilizar o processo de seleção das medidas de segurança a serem selecionadas pelo RTRMM.

8.2.2 Módulo Threat Analyser

O Threat Analyzer é um módulo que tem como objetivo analisar um fluxo de dados IoT a fim de verificar a possibilidade de existir ou não uma ameaça. A técnica de detecção aplicada pelo Threat Analyzer se baseia na premissa da incerteza e nas teorias da probabilidade, e na lógica difusa (fuzzy logic). Além desses aspectos, dois fatores críticos de sucesso são primordiais para validar as informações por ele fornecidas:

1. Como confiar que existe uma ameaça - esta questão pode ser resolvida com base nas evidências coletadas, considerando parâmetros pré-estabelecidos. Os valores destes parâmetros foram baseados, inicialmente, em incidentes já ocorridos.

2. Como representar essa confiança para agentes de software e usuários - analisando os resultados obtidos, tem-se condições de determinar qual ou quais mecanismos de segurança poderão se adotados, e assim propor que aquele tipo de risco terá reduzida a probabilidade de ocorrer.

Programação em Lógica Difusa (PLD) x Programação Lógica Probabilística (PLP)

- A programação lógica difusa integra Lógica Fuzzy e Programação em Lógica pura, a fim de lidar com a imprecisão essencial de alguns problemas usando técnicas

declarativas[JIe20]. A PLD apresenta resultados que o sistema produz apenas respostas corretas e todas as respostas corretas possíveis [Ebr01]. Ela permite a representação de crenças certas (verdadeiro ou falso) sobre conceitos e proposições de definições ontologicamente qualitativos e vagos por operadores de teoria dos conjuntos difusos, que são usados naturalmente em alguns domínios de perícia, como a cibersegurança, por especialistas humanos, e combinação destes com conectivas lógicas (embora de semântica distinta da lógica clássica).

- a PLP permite a representação de crenças epistemologicamente incertas e quantitativas sobre conceitos e proposições, tanto precisos em lógica clássica, quanto vagos em PLD

Arquitetura do Threat Analyzer

A arquitetura do Threat Analyzer foi baseada na técnica de lógica difusa (fuzzy logic), pois nela se trabalha com um grau de incerteza, mas ao mesmo tempo oferece suporte para decidir se uma ameaça ocorreu ou não [Joh22, San07]. O grau de incerteza possibilita um raciocínio não exato, isto é, nada está totalmente adequado ou correto e nada está totalmente inadequado ou errado. Tudo pode ser possível, desde que as proposições estejam dentro do universo a ser tratado.

Sendo assim, ao utilizar os conceitos de lógica difusa, viabiliza que "tudo seja possível ou permitido", possibilitando que com base na aproximação, uma abordagem de percepção e visão do problema a ser tratado, o problema poderá ser mapeado e por conseguinte tratado. Desta forma, pode-se afirmar que a lógica fuzzy: [Zad88, RN95].

- Permite modelar um problema de forma aproximada, ao invés de precisa, aproximando-se do raciocínio humano em várias situações. Este raciocínio desenvolvido levou à proposição de uma técnica formal para tratar o raciocínio inexato, denominado teoria da possibilidade
- Possibilita a resolução de problemas complexos e tomadas de decisão e controle por meio de questionamentos. Não existe a visão de certo ou errado, mas os questionamentos levam a proposições que possibilitam uma possível solução ao problema.
- Possibilita trabalhar com um universo de opções adequadas ao problema a ser tratado. As opções vão ao encontro de observações, situações que se foram apresentando durante a solução do problema.

Ao pensar na arquitetura do módulo Threat Analyzer, viu a necessidade de se ter uma arquitetura que além de possibilitar que dados fossem capazes de ser mapeados e trabalhados, tivesse a capacidade de realimentar os dados de entrada com os dados processado para prover uma maior aproximação da realidade. Com base nesta premissa, a arquitetura do Threat Analyzer, apresentada na figura 8.2, foi baseada na arquitetura do controle de lógica difusa (FLC – Fuzzy Logic Control) [Ian12].

A escolha de trabalhar com a arquitetura do controle de lógica difusa se deve ao fato da capacidade dela trabalhar com processos considerados complexos e mal definidos, especialmente aqueles que podem ser controlados por um operador humano qualificado sem o conhecimento de sua dinâmica de avaliação. Se for pensar como representar a lógica do raciocínio humano na solução de um problema, existe toda uma parametrização de dados (processo de fuzzificação) para que possam então ser processados e posteriormente convertidos (processo de defuzzificação) para a leitura humana.

Ao escolher qual seria a melhor arquitetura FLC para o Threat Analyzer, foram analisados alguns sistemas FLC. Dentre os analisados, destacam-se o Mamdani [MA75], Tsukamoto [GT80], Sugeno [VA16] e Larsen [Lar80]. Todas essas arquiteturas trabalham com dados nítidos (dados crisp) como entradas, o que facilita a compreensão de informações provenientes de raciocínio humano.

Todavia, a escolha foi pelo sistema Mamdani, sistema que possibilita mapear os controles linguísticos em regras SE-ENTÃO, se baseiam nas funções de implicação fuzzy e em operadores de composição de saída fuzzy [AJ08]. O sistema Mamdani, também provê a clareza de encontrar uma ação de controle "crisp" a partir da base de regras difusas e de um conjunto de entradas "crisp" para representação dos dados eternos [IC09].

Dinâmica do Threat Analyzer

A arquitetura desenvolvida é composta por um conjunto de funcionalidades responsáveis em analisar o fluxo de dados IoT a fim de verificar se existe alguma ameaça. Para isso, é apresentando como resultado o valor da probabilidade de um fluxo de dados possuir uma ameaça. Como funciona o Threat Analyzer?

Coletados os pacotes do fluxo de dados IoT, são selecionados quais campos irão compor as entradas crisp (protocolo, endereço IP, portas, ...), e os parâmetros que irão compor as regras do sistema fuzzy. Para cada uma das regras, em sua composição, é atribuído um valor de probabilidade, que define a chance de uma ameaça se fazer presente via um dos campos do pacote selecionado. O valor da probabilidade é atualizado pelo Threat Analyzer sempre que novos cálculos são necessários, conforme o fluxo de dados IoT.

Quando o fluxo é analisado pela primeira vez, é realizado o processo de inferência para calcular o valor da probabilidade com base em um ou mais campos (protocolo, endereço IP, portas, ...) dos pacotes coletados. O valor inicial a ser inferido, entendido como a entrada (input) para o sistema, é calculado pelo administrador do sistema IoT, baseado em fatores externos como: histórico de ataques, tipo e volume de tráfego, contexto de aplicação do dispositivo no sistema IoT, entre outros. O valor da probabilidade inferida em conjunto com os campos selecionados irá compor as regras que servirão como referência no motor de inferência.

As entradas "crisp" passarão pelo processo de fuzzificação para serem utilizadas pelo motor de inferência, responsável em comparar os valores que sofreram o processo de fuzzificação como as regras criadas. O resultado deste processo é encaminhado ao processo de defuzzificação que pode encaminhar ao módulo Risk Management ou ser reavaliado

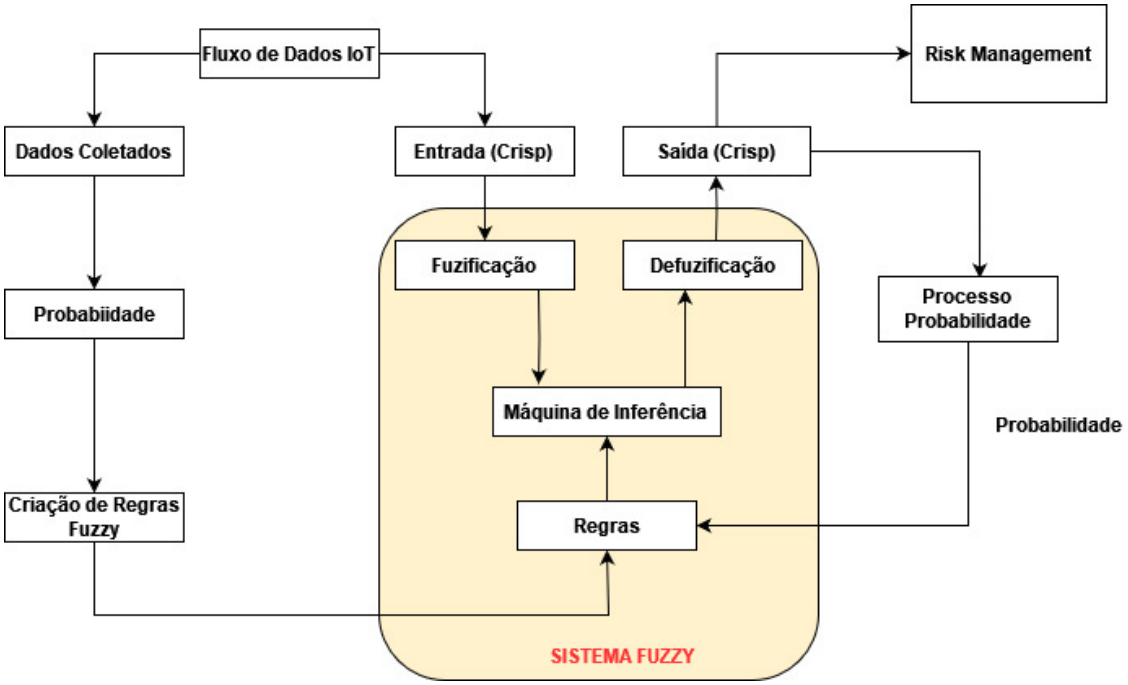

Figura 8.2: Arquitetura do Threat Analyzer - Adaptado de [Ian12]

com o objetivo de melhor aproximar a probabilidade (realimentação do processo) de um rastro de ameaça ser ou não uma real ameaça ao sistema.

O aprendizado do sistema ocorre atualizando os valores de probabilidade de uma detecção, isto é, todos os valores de probabilidade calculados são inseridos na composição de novas regras para serem avaliadas pelo sistema. Estas novas regras serão aplicadas a um novo conjunto de fluxo de dados, provenientes do mesmo dispositivo IoT, buscando uma maior aproximação realidade da detecção de ameaças.

Valor de Probabilidade

O projeto do RTRMM é baseado em programação lógica, tendo o Threat Analyzer, um dos seus módulos, implementado com base nos conceitos de programação em lógica probabilística. Os valores da probabilidade que devem ser atribuídos, tanto no início do processo de análise de uma vulnerabilidade, quanto os demais, durante o processo de aprendizagem, devem ser inferidos / calculados com base em uma tecnologia adequada.

Duas tecnologias foram analisadas para determinar os valores de probabilidade, Redes Bayesianas [D 99] e Cadeia de Markov [Beh00]. As redes Bayesianas trabalham com base em informações anteriores, o estado anterior ao atual. A probabilidade de um evento é calculada com base na probabilidade de um evento acontecido anteriormente ao atual. Reforça que uma rede Bayesiana é criada com base na dependência dos eventos anteriores. Todavia, quando se trabalha com cadeia de Markov, não existe preocupação com o evento anterior, mas sim com o estado atual. O cálculo da probabilidade do próximo evento é baseado na probabilidade do estado atual.

A decisão de projeto pela tecnologia de cadeia de Markov se deve ao fato de ser mais simples de ser implementada, além de ser menos expressivo que redes bayesianas dinâmicas (redes tenporais). Apesar de cadeia de Markov serem menos custosas quanto a carga computacional, ela não possibilita detectar ameaças persistentes avançadas (Advanced Persistent Threats) em função da incapacidade de trabalhar com um histórico de ameaças o que inviabiliza representar dependências temporais a médio prazo.

Definida a tecnologia de cadeia de Markov, optou-se por fazer uso de uma cadeia de Markov simples. A escolha teve como objetivo não propor sobrecarga de tarefas, tendo em vista que o objetivo do projeto é trabalhar em pequenos intervalos de tempo útil a fim de proporcionar reais decisões em tempo capaz de realizar uma análise antes que a ameaça seja efetivada ou mesmo que os danos por ela causados possam ser minimizados.

Aplicação Markov no Threat Analyzer

O Threat Analyzer, usa no cálculo de suas probabilidades o princípio de cadeia de Markov, conforme definido. Sendo assim, previsões de probabilidades em seu futuro com base é baseado somente no seu estado atual independentemente do que aconteceu no passado até o estado atual. Desta forma, o valor da probabilidade de cada estado (probabilidade condicional) é calculado com base na expressão, no qual X são valores aleatórios, t representam pontos sucessivos no tempo. Logo, dado todo o histórico anterior, a distribuição de probabilidade para a variável aleatória na próxima etapa de tempo depende apenas da variável atual.

$$P(X(t_{k+1}) = x_{k+1} \mid X_{(t_k)} = x_k)$$

Por exemplo, suponha que tenhamos três fluxos de dados IoT, no qual serão analisados se possuem ou não ameaças (inicialmente consideradas anomalias) com base nos seguintes parâmetros: porta de destino, protocolo de comunicação e endereço IP de destino. A tabela 8.1 representa o estado inicial das anomalias nos fluxo de tráfego IoT, no qual foi atribuído um valor de probabilidade para cada um desses parâmetros existente em cada fluxo IoT.

Anomalias			
Parametros	Anomalia	Anomalia1	Anomalia2
Port	0.3	0.5	0.4
Protocol	0.6	0.7	0.6
IP	0.7	0.5	0.8

Tabela 8.1: Anomalias em Fluxos IoT

Os valores da tabela 8.1 podem ser dispostos em um vetor X, denominado Vetor de Estados:

X = [Port Protocol IP]

As probabilidades de cada estado (probabilidade não condicional) podem também ser dispostas em 3 vetores distintos $p_1, p_2 e p_3$, denominados de vetores de probabilidade de estado, utilizados para distinguir das probabilidades de transição, que é a probabilidade que caracteriza a transição para o estado em um instante de tempo T, dado que o estado anterior foi $(T-1)$. Os valores Anomalia, Anomalia1 e Anomalia2 são vetores de estado, onde em Anomalia, por exemplo, 0.3 representa a probabilidade para o parâmetro Port; 0.6 a probabilidade para o parâmetro Protocol; e 0.7 a probabilidade para o parâmetro IP.

$$\text{Anomalia} = [0.3\ 0.6\ 0.7]\ \text{Anomalia1} = [0.5\ 0.6\ 0.7]\ \text{Anomalia2} = [0.4\ 0.6\ 0.8]$$

Assumindo que as probabilidades de transição para intervalos de 15 segundos são dadas pela tabela 8.2:

Tabela de Probabilidade de Transição			
Anomalia	I	II	III
I	0.3	0.4	0.3
II	0.6	0.2	0.2
III	0.7	0.2	0.1

Tabela 8.2: Tabela de Probabilidade de Transição

A tabela de probabilidade de transição somente foi aplicada a Anomalia, mas também deveria ser executada tanto para Anomalia1 e Anomalia2. A aplicação somente em um dos casos visa explanar como o processo de cálculo da probabilidade ocorre.

As probabilidades condicionais, no qual a mudança do estado I para os demais estados, usando, por exemplo, 15 segundos como o tempo entre a mudança de um estado para o outro, foi apresentadas como:

1. Estado I para I - a probabilidade do estado ser I após 15 segundos, dado que o estado atual é 0.3 - $P(X(t+15) = I|X(t) = I) = 0.3 \rightarrow P(X(15) = IX(1) = I) = 0.3$

2. Estado I para II - a probabilidade do estado ser II após 15 segundos, dado que o estado atual é 0.3 - $P(X(t+15) = I|X(t) = I) = 0.3 \rightarrow P(X(15) = IX(1) = I) = 0.3$

3. Estado I para III - a probabilidade do estado ser III após 15 segundos, dado que o estado atual é 0.3 - $P(X(t+15) = I|X(t) = I) = 0.3 \rightarrow P(X(15) = IX(1) = I) = 0.3$

A partir dos valores da tabela de probabilidade de transição é criada uma matriz chamada de Matriz de Transição com todos os valores da tabela.

Matriz de Transição

$$\begin{bmatrix} 0.3 & 0.4 & 0.3 \\ 0.6 & 0.2 & 0.2 \\ 0.7 & 0.2 & 0.1 \end{bmatrix}$$

De posse de todas estas informações, tem-se condições de calcular a probabilidade a cada 15 segundos. Com a matriz de transição e o vetor de probabilidade de estado para t, p(0), calcula-se o vetor de probabilidade de estado para t+15, p(1).

$$[30\ 60\ 70] \begin{bmatrix} 0.3 & 0.4 & 0.3 \\ 0.6 & 0.2 & 0.2 \\ 0.7 & 0.2 & 0.1 \end{bmatrix} = [94\ 38\ 28]$$

Para se conhecer a próxima probabilidade a ser aplicado pelo Threat Analyzer é realizado o cálculo do produto entre o vetor de probabilidade e a matriz de transição, os valores de probabilidade em t+15 serão 0.94, 0.38 e 0.28. Logo, a cada 15 segundos novos valores de probabilidades são calculados e atribuídos ao Threat Analyzer.

Processo de Fuzzificação

O processo de fuzzificação irá tratar os valores de entrada como por exemplo: número de porta, endereço IP, protocolo (entradas Crisp), para valores de pertinência, valores entre 0 e 1. Definição: Definidos os universos das variáveis crisp X, cada uma das variáveis x_i deve estar contida em seu respectivo conjunto de domínios:

$$x_i \Rightarrow \begin{cases} if, \exists X \mid X \subset P_S \Rightarrow \text{Porta de Origem.} \\ if, \exists Y \mid Y \subset P_D \Rightarrow \text{Porta de Destino.} \\ if, \exists Z \mid Z \subset Ip_S \Rightarrow \text{Endereço de Origem.} \\ if, \exists W \mid W \subset Ip_D \Rightarrow \text{Endereço de Destino.} \\ if, \exists P \mid P \subset Prot \Rightarrow \text{Protocolo de Comunicação.} \end{cases}$$

I - Termos do Threat Analyzer - Os termos são definidos como um conjunto de valores discretos L representando o universo de variáveis difusas, em que l_i é um dos valores deste conjunto L $\rightarrow \{0, 1\}$.

Definição: Com base no universo de variáveis difusas, os termos definem S onde existem valores μl_i, em que estes valores de S são: S = $\{(l_i, \mu S(l_i))/l_i \in L\}$, de forma que $\mu S(l_i)$ é a função de relevância, e está contida no intervalo entre 0 e 1: μS: L $\rightarrow$ [0, 1].

Foram especificados 4 (quatro) grupos para representarem os termos no processo de fuzzificação, possibilitando o cálculo do valor da variável fuzzy. Os grupos foram estabelecidos com base na metodologia de uma análise de riscos qualitativa [ISO22b]. A escolha de quatro (4) níveis deve-se ao fato de tornar a análise mais simples e prática, otimizando tempo no seu processo.

1. Likely - há uma grande probabilidade de que possa ser uma ameaça. O grau de relevância é: $\mu S \geq 0.9$
2. Almost - existe a probabilidade de ser uma ameaça. O grau de relevância é: $\mu S \geq 0.7 \wedge \mu S < 0.9$
3. Sometimes - há uma probabilidade média de ser uma ameaça. O grau de relevância é: $\mu S \geq 0.4 \wedge \mu S < 0.7$
4. Unlikely - a probabilidade de ser uma ameaça é mínima. O grau de relevância é: $\mu S < 0.4$

II - Função de Pertinência - A função de Pertinência reflete o conhecimento que se tem em relação à intensidade com que o objeto pertence ao conjunto fuzzy. A maioria das aplicações práticas com lógica de fuzzy trabalha com distribuições trapezoidais ou triangulares [SI18]. Desta forma, no Threat Analyzer a opção foi trabalhar com a função trapezoidal, por ser uma função parcialmente linear e contínua.

III - Motor de Inferência (Inference Engine) e Regras de Inferência - O motor de inferência do Threat Analyzer é o responsável por aplicar as regras de inferência à entrada difusa (fuzzy) para gerar a saída difusa (fuzzy). As regras são definidas juntamente com as entradas fuzzy de acordo com as funções de pertinência (reflete o conhecimento que se tem em relação à intensidade com que o objeto pertence ao conjunto difuso (fuzzy) [SI18].

Para determinar a região resultante, o processo de inferência usou a técnica de Mandami [MA75], pois ele é intuitivo, mais adequado para entrada humana, por possuir uma base de regras facilmente interpretável (IF-ELSE: IF TERMO is X ELSE Y) e por ter uma ampla aceitação.

As regras avaliam a verdade fuzzy, T, de uma sentença complexa, e são usadas para ligar as diferentes variáveis fuzzy [MA75]. Desta forma, no Threat Analyzer, uma regra representa como uma anomalia pode ser representada em um fluxo de dados IoT, realizando a combinação entre os elementos que compõem o tráfego e a probabilidade desta anomalia acontecer neste fluxo.

Regras de Inferência do Threat Analyzer

As regras de inferência do Threat Analyzer foram criadas utilizando os mesmos termos utilizados no processo de fuzzificação. Vale a pena citar que os valores de probabilidade estarão em conformidade com os termos definidos. Seguem alguns exemplos de aplicação:

1. IF P_D é Sometimes THEN Threat é Sometimes ⇒ esta regra especifica que há uma possibilidade média de que possa ser uma ameaça (figura 8.3). A primeira imagem da figura 3.3 apresenta um gráfico tendo como coordenadas o grau de relevância (μ_s) para porta de destino (p_d), usando a função de pertinência trapezoidal para representar os níveis de ameaças. A linha vermelha representa o grau de pertinência da porta 80 estar recebendo tráfego com ameaça.

   A imagem 2 representa a junção das funções de pertinência, no qual pode-se constatar que a maior concentração encontra-se em Almost, nível médio de existir uma ameaça no tráfego dos dispositivo IoT direcionado para a porta 80.

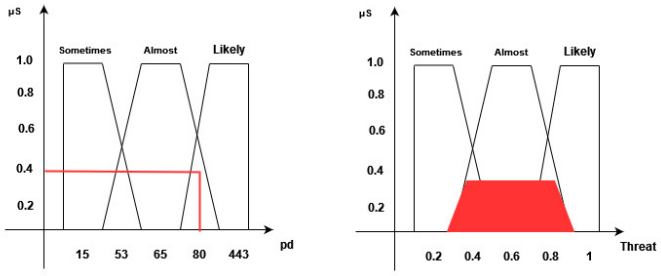

Figura 8.3: Regra1

2. IF P_D é Almost ∨ Ip_D é Likely THEN Threat é Likely ⇒ esta regra especifica que há uma grande chance de que possa ser uma ameaça (figura 8.4). Esta regra leva em consideração o grau de relevância (μ_s) para porta de destino (p_d), na primeira imagem e o grau de relevância (μ_s) para o IP de destino (ip_d), na segunda imagem. Em ambas as imagens é utilizada a representação trapezoidal para representar os níveis de ameaças. A linha vermelha representa o grau de pertinência da porta 80 estar recebendo tráfego com ameaça, na primeira imagem. Na segunda imagem, a linha vermelha representa que o IP 220 com o grau de pertinência 0.9 pode possuir uma ameaça.

   AA terceira imagem junção das funções de pertinência, no qual pode-se constatar que a maior concentração encontra-se em Likely, nível alto de existir uma ameaça no tráfego dos dispositivo IoT direcionado para a porta 80, no IP 220

3. IF P_D é Almost ∨ Ip_D é Likely ∨ Prot é Likely THEN Threat é Likely ⇒ esta regra especifica que há uma grande possibilidade de que possa ser uma ameaça (figura 8.5). Esta regra leva em consideração o grau de relevância (μ_s) para porta de destino (p_d), na primeira imagem e o grau de relevância (μ_s) para o IP de destino (ip_d), na segunda imagem, na terceira imagem o protocolo TCP. Em ambas as imagens é utilizada a representação trapezoidal para representar os níveis de ameaças. A linha vermelha representa o grau de pertinência da porta 80 estar recebendo tráfego com ameaça, na primeira imagem. Na segunda imagem, a linha vermelha representa que o IP 220 com o grau de pertinência 0.9 pode possuir uma ameaça. Na terceira imagem, a linha vermelha representa o protocolo TCP com o grau de relevância 0.9.

   A quarta imagem junção das funções de pertinência, no qual pode-se constatar que a maior concentração encontra-se em Likely, nível alto de existir uma ameaça no

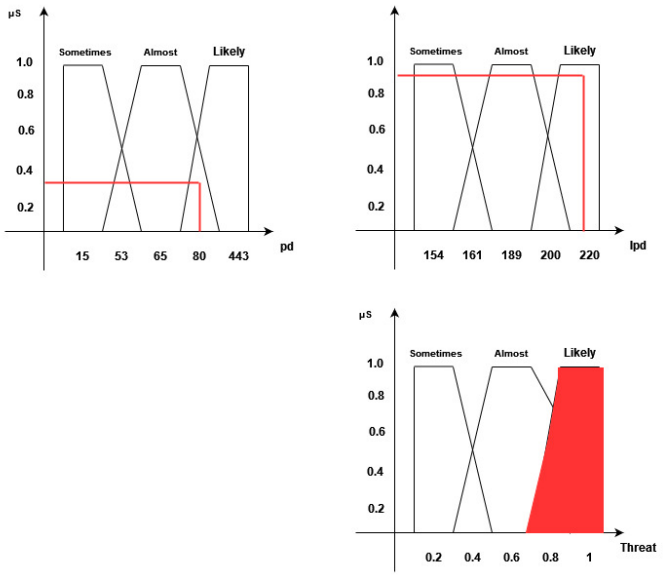

Figura 8.4: Regra2

tráfego dos dispositivo IoT direcionado para a porta 80, no IP 220, com protocolo TCP.

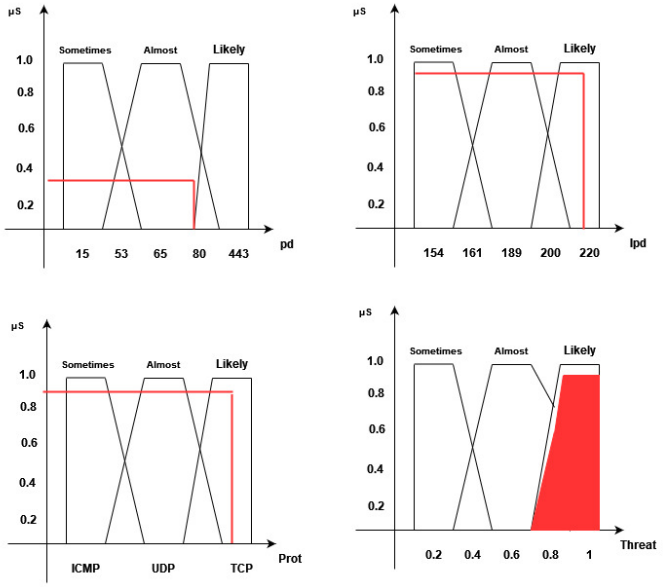

Figura 8.5: Regra3

Processo de Defuzzificação (Defuzzification)

O processo de defuzzificação é necessário quando se espera que o sistema retorne um número e não o conjunto fuzzy criado para cada evento. A técnica de defuzzificação adotada foi a centróide, onde seu cálculo varia em função da saída, podendo assumir

valores discretos ou contínuos. No caso do Threat Analyzer, trabalhamos como valores discretos, onde o valor da probabilidade de uma anomalia determina se é ou não uma ameaça. A escolha adotada deve-se ao fato de que as possibilidades são discretas e o cálculo é baseado em uma média ponderada, de acordo com o grau de relevância para a distribuição de possibilidades da saída do modelo. [Jan07]

8.2.3 Módulo Risk Management

O módulo Risk Management tem como objetivo analisar e avaliar as ameaças detectadas pelo Threat Analyzer direcionadas aos sistemas IoT, de forma dinâmica em tempo real, tendo como resultado o risco calculado, analisado e avaliado, informando ao módulo DB Controls quais riscos avaliados serão tratados.

A decisão de estruturar o módulo Risk Management no formato citado deve-se ao fato de que o mesmo foi baseado no processo de gestão de riscos especificado na norma ISO 27005 [ISO22b]. Esta norma possui um conjunto de funcionalidades que se mais adequa às necessidades e especificidades do RTRMM. Dentre as funcionalidades da ISO 27005, 3 foram selecionadas: calcular o impacto do dispositivo para o sistema IoT,; analisar e calcular os riscos existentes; e avaliar os riscos.

Como citado, 0 módulo está dividido em 3 (três) funcionalidades básicas, conforme apresentado na figura 8.6:

1. Cálculo do impacto (Calculate Impact) – que tem como base o valor da importância deste dispositivo para o funcionamento do sistema. Possui o objetivo calcular o valor do impacto para o sistema IoT no caso de um dispositivo IoT estar comprometido por uma ou mais ameaças.

2. Análise de riscos (Risk Analyzer) - tem o objetivo analisar e calcular os riscos existentes no tráfego existente entre componentes do sistema IoT, com base na probabilidade das ameaças e no impacto calculado.

3. Avaliar riscos (Risks Assessment) – tem como objetivo avaliar os riscos determinados pela função de análise de riscos, a fim de determinar aqueles que serão tratados e a prioridade de tratamento, ordenando-os conforme o seu grau de criticidade e o tipo de dispositivo IoT.

O resultado do Threat Analyzer, a probabilidade de existir uma ameaça, junto com o valor de impacto, são parâmetros que foram analisados e que possibilitam o cálculo do risco. Na figura 8.6 foram estabelecidos valores para probabilidade e impacto, levando em consideração tratar somente ameaças que realmente possam existir e venham causar um real impacto no sistema IoT. O valor do risco é encaminhado para o módulo de avaliação de riscos que em conjunto com o custo operacional do dispositivo é analisado se o mesmo será tratado e a sua prioridade de tratamento.

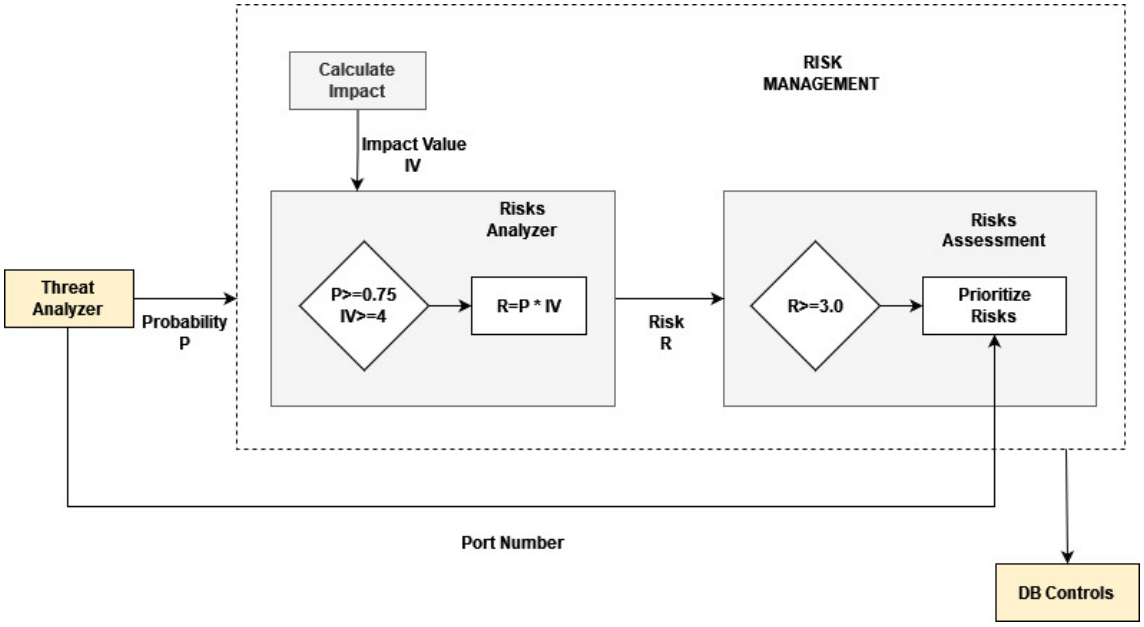

Figura 8.6: Módulo Risk Management

Cálculo do Impacto

O impacto é um dos parâmetros utilizados pelo RTRMM para calcular o risco no processo de gestão de riscos. Todo dispositivo IoT possui um "valor" de importância para o sistema, que está relacionado ao seu valor estratégico no sistema IoT. Quando este dispositivo é afetado em função de um ou mais incidentes de segurança (uma ameaça), a sua importância para o sistema IoT está diretamente relacionado o quanto irá prejudicar as funcionalidades do sistema como um todo. Logo, o valor de importância está diretamente ligado ao montante dos danos causados ou custos à organização.

O nível de danos causados por uma ameaça efetiva pode ser determinado por um conjunto de fatores, como:

- O nível de classificação do dispositivo IoT – cada dispositivo possui uma ou mais funcionalidades exercidas. Separar os dispositivos por classes de funcionalidade facilita determinar o grau de importância para o sistema IoT.
- As ocorrências de violação da segurança da informação – em conformidade com as funções executadas por um dispositivo IoT, este pode ser mais ou menos visado pelos hackers. Existe assim, a necessidade de se manter um histórico de ataques a fim de se ter conhecimento do tipo e da quantidade de incidentes que o dispositivo sofreu. Desta forma, se tem uma melhor noção do grau de importância do dispositivo para o sistema como todo.
- Danos graves no sistema em função desse ataque – dependendo do tipo de incidente e qual dispositivo IoT for afetado, se tem condições de saber as consequências do evento. Desta forma, existe uma real noção da importância do dispositivo para o sistema IoT.

- Outros danos à reputação e violações de requisitos legais ou contratuais – como citado, todo incidente provê consequências negativas para uma Organização. Se um sistema IoT, normalmente trabalha em tempo real, não atende as necessidades de seus clientes, a Organização está sujeita a perda de confiança, processos, sejam pessoais ou mesmo contratuais.

No projeto do RTRMM, os parâmetros selecionados para determinar o valor de importância é estabelecido pelo administrador do sistema. A definição do cálculo do impacto está baseada no nível de classificação do dispositivo IoT, isto é, o quanto este é importante para a manutenção das funcionalidades do sistema. A decisão por este parâmetro foi a possibilidade de se automatizar o processo de valorização dos ativos, tendo em vista que o administrador do sistema tem a capacidade de parametrizar todos os valores antes do seu início.

Todavia, existe a ideia de se utilizar os demais parâmetros, e serem alterados em tempo de execução, com inferência externa (fator humano) ou mesmo via aprendizado em tempo real. Foram analisadas as duas maneiras de reconhecer e classificar um padrão [CJ01], que possibilitariam classificar e estabelecer valores para os demais parâmetros para o cálculo do impacto. São eles:

1. Classificação supervisionada – neste formato, existe uma classe já pré-definida, em tempo de projeto, e o padrão de entrada é identificado com base nesta classe.
2. Classificação não supervisionada - o padrão é identificado por um software, capaz de realizar esta tarefa dentro de um conjunto de classes, dentro de um conjunto de dados não rotulados, dados que não estão marcados com a resposta correta.

Optou-se inicialmente, em trabalhar com um parâmetro pois facilitaria a implementação da função do cálculo do impacto. Todavia, existe a possibilidade de implementar todos os parâmetros fazendo uso da estratégia não supervisionada, o que em parte tornaria o cálculo mais confiável e preciso em função do cruzamento de vários parâmetros.

Sendo assim, os critérios adotados para estabelecer o valor da importância do ativo IoT foram: a quantidades de dispositivos executando a mesma tarefa para determinada funcionalidade; e a importância do serviço IoT oferecido. O projeto está inicialmente parametriza estes valores em tempo de projeto para cada um dos itens. Para isso são analisados:

- O alcance do sistema - a quantidade de dispositivos responsáveis em executar a mesma funcionalidade. Essa análise está baseada na topologia da rede, onde é analisado o grau de redundância de dispositivos para executar uma mesma funcionalidade
- As funcionalidades – o grau de importância da funcionalidade exercida pelo dispositivo para o sistema e seus usuários. Essa análise é realizada pelo próprio administrador do sistema. O valor está diretamente relacionado ao valor do nível de impacto; e

- O alcance aos usuários do sistema - a quantidade de dispositivos que atuam junto ao usuário na execução de uma tarefa.

Com base nos critérios de importância dos ativos (dispositivos IoT), existem condições de se estipular o impacto do dano ao sistema IoT. Para isso, foram adotados 4 (quatro) níveis de impacto, "critical", "high", "medium" e "low", no qual cada nível recebe um valor numérico de 2 a 5, sendo 5 "critical" e os demais valores são subsequentes em ordem decrescente. O valor a ser atribuído ao nível de impacto está diretamente relacionado aos critérios citados.

- A capilaridade do sistema - Suponha a quantidade de dispositivos IoT ligados a uma mesma funcionalidade;

    1. Critical - $\exists d_i | i \geq 2$, o sistema deverá ter pelo menos 2 dispositivos disponíveis para suprir as necessidades do sistema. Isto significa que caso um dispositivo seja afetado em um incidente de segurança, existirão pelo menos outros 2 dispositivos que irão suprir a demanda de transferência de dados no sistema.
    2. High - $\exists d_i | i \geq 1$, o sistema deverá ter 1 ou mais dispositivos disponíveis para suprir as necessidades do sistema.
    3. Medium - $\exists d_i \mid$ sse $i = 1$, o sistema deverá exatamente 1 dispositivo disponível para suprir as necessidades do sistema.
    4. Low – o dispositivo não possui importância para o sistema.

- O alcance aos usuários do sistema - segue a mesma premissa dos dispositivos apresentados em capilaridade do sistema. Isto significa que, por exemplo, para não ser "critical", deve existir pelo menos 2 dispositivos que provem a mesma funcionalidade ao usuário no caso de um outro dispositivo sofrer algum dano por um incidente de segurança.

- O grau de importância da funcionalidade exercida pelo dispositivo:

    1. Critical – um dispositivo pode ser considerado como "Critical" quando este exerce a plena funcionalidade do sistema ou de um módulo deste, afetando diretamente as atividades de seus usuários, vindo a causar sérios danos aos mesmos (integridade física), sejam pela sua ausência ou mesmo pela manipulação de suas funcionalidades, como: dispositivo que controla a inserção de medicamentos em um paciente remoto; manipulação de dados coletados em treinos ou competições de atletas. O valor da probabilidade $0.9 \leq p \leq 1$.
    2. High - um dispositivo pode ser considerado "High" quando este exerce atividades que venham afetar diretamente ou não as funcionalidades do sistema ou de módulo deste, afetando as atividades de seus usuários (perda ou alteração de informações consideradas sensíveis), podendo causar danos sejam pela sua ausência ou mesmo pela manipulação de suas funcionalidades, como: manipulação de dados relacionados a informações pessoais, informações trafegadas entre sistema de monitoramento. O valor da probabilidade $0.75 \leq p < 0.9$.

3. Medium - um dispositivo pode ser considerado "Medium" quando este exerce atividades que não afetam diretamente as funcionalidades do sistema ou de módulo deste, afetando somente em parte as atividades de seus usuários (falta de acesso a um dispositivo IoT), podendo causar danos, mas considerados reparáveis, pois não afetam a sua integridade, informação ou mesmo a execução de alguma atividade que possa ser considerada crítica, como manipulação de dispositivos de manutenção de empresas. O valor da probabilidade $0.6 \leq p < 0.75$.

Análise de Riscos

O processo de análise de riscos (Risks Analyzer) tem como objetivo conhecer, analisar e calcular os riscos, com base em todas as ameaças fornecidas pelo Threat Analyzer. O valor da probabilidade da ameaça calculada, junto com o valor de impacto (Impact Value - IV) fornecido pelo cálculo de impacto, possibilitará calcular o valor do risco e assim determinar o quanto o dispositivo/sistema IoT está vulnerável.

Para analisar e avaliar os riscos foi adotada a técnica da teoria dos jogos [Cox09, Hau02]. A escolha por esta técnica se deve ao fato dela possibilitar uma melhor estratégia de análise dos riscos, pois apresenta uma visão tanto do defensor quanto do atacante, proporcionando assim um universo maior e mais próximo a realidade. A figura 8.3 apresenta uma tabela "payoff", genérica, com dois jogadores atacante e defensor, no qual o defensor possui duas estratégias de defesa possíveis - defender o alvo A ou defender o alvo B, levando em consideração que o atacante pode atacar o alvo A ou o alvo B.

A estratégia é que o defensor somente pode defender um dos alvos, e sendo eficaz não existe dano. Caso os dois jogadores façam as suas escolhas simultaneamente, cada um ignorando o que o outro fez, acontece o equilíbrio único de Nash [Cam15] de estratégia mista, igualando os resultados esperados de cada estratégia para cada jogador. Vale a pena ressaltar que nesse formato de matriz payoff não é levado em consideração a questão do ambiente onde se encontra o sistema IoT. É levado em consideração somente as estratégias do atacante e defensor.

DEFESA	ATAQUE		
		ATAQUE 1	ATAQUE 2
	ATIVO 1	-20	-170
	ATIVO 2	-140	-40

Tabela 8.3: Matriz Payoff, [Cox09]

Na tabela 8.3 tem-se um custo de 20 reais para defender Ativo 1 e 40 reais para defender o Ativo 2. Ao mesmo tempo se sabe que existe um custo de falha para defender o Ativo de 150 reais e 100 reais para o Ativo 2.

De posse dessas informações existe a possibilidade de:

1. Conhecer a probabilidade do atacante maximizar dano ao defensor atacando o ativo 1.

$$\frac{(40-170)}{(40-170)+(20-140)} = \frac{130}{130+120} = \frac{130}{250} = 0.52$$

2. Conhecer a probabilidade do defensor minimizar a perda defendendo o ativo 1.

$$\frac{(40-140)}{(40-140)+(20-170)} = \frac{100}{100+150} = \frac{100}{250} = 0.4$$

Análise da Situação

Desta forma, durante a avaliação realizada da situação, se levou em consideração como foi calculada a probabilidade de danos que podem ser causados a um ativo, bem como a probabilidade de minimizar o sucesso de um ataque.

O resultado da análise dessa situação mostra que o atacante possui mais de 50% de efetividade de causar danos ao ativo1, em contrapartida, a probabilidade de minimizar a perda é menor (40%), o que demonstra uma maior probabilidade de danos que chances de defesa. Desta forma, numa análise inicial, a solução seria tratar estes riscos adequando um mecanismo de segurança com o intuito de reduzir a probabilidade de causar danos ao ativo.

Esta análise inicial está baseada em valores, o que não a torna completa, precisando ainda levar em consideração a necessidade de conhecer se o risco realmente necessita ser tratado. Logo, deve-se verificar se a ameaça existente ao ativo realmente é factível a ponto de ser levada em consideração, isto é, se o risco que ela vem a causar está em um nível alto ou crítico, de forma que deva ser tratado.

O risco para ser calculado deverá ter como parâmetros a ameaça (A), o impacto (I) ao dispositivo Iot e/ou ao sistema IoT, bem como o quanto está vulnerável (V). O resultado do seu cálculo será representado por probabilidade, e todo e qualquer risco que tiver uma probabilidade $p \geq 7$ deverá ser tratado.

A fórmula de cálculo do risco é representada pela expressão:

$$\text{Risco} = \text{A x V x I}$$

Sendo assim, o valor da probabilidade da ameaça é conhecida com base na matriz payoff da figura 8.3, no qual o valor da probabilidade é o valor obtido pelo atacante ter sucesso ao causar danos a um dispositivo IoT. O valor do impacto está relacionado à importância do dispositivo exerce sobre o sistema IoT (critical, high, medium). O valor da vulnerabilidade é atribuído pelos gestores de riscos, pois são eles que possuem total conhecimento do sistema IoT, o seu contexto dentro da organização e seu ambiente em que encontra.

Calculado o valor do risco e este O valor do risco encontrado é encaminhado ao processo de avalição de riscos, componente do módulo Risk Assessment, a fim de classificar e priorizar o que será tratado.

Avaliação de Riscos

O processo de avaliação de riscos (Risk Assessment) do módulo Risk Managent tem como funcionalidades determinar quais riscos serão tratados e a sua ordem de prioridade. O processo é baseado no valor do risco calculado pelo processo de análise de riscos, e pelo tipo do dispositivo IoT que sofreu o incidente, elemento que irá determinar o custo operacional deste dispositivo para o sistema IoT.

Para resolver o que será tratado, aplicamos as técnicas utilizada na Teoria dos Jogos, considerado como uma teoria Matemática aplicada na construção de modelos que descrevem fenômenos ou situações de conflito ou de cooperação entre dois ou mais agentes / tomadores de decisão interagem entre si. Desta forma, a teoria dos jogos apresenta uma linguagem para descrever processos de decisão que envolvam dois ou mais elementos [SGB+04, Pim21].

Dilema Tratar ou Não O problema na fase de avaliação de riscos consiste em definir o que será tratado (Tratar - T) e o que não será tratado (Não Tratar - NT), chamados de jogadores, na teoria de jogos. Em paralelo são estabelecidas duas estratégias, RISCO e CUSTO, que pertencem a um conjunto de estratégias pura para definir o que tratar e não tratar.

I - Estratégia Riscos

A estratégia de riscos a serem tratados é baseada na classificação dos riscos, isto é, no valor do risco calculado na fase de análise de riscos. Foram especificados 4 (quatro) níveis para classificar o risco, no qual os valores foram atribuídos exclusivamente para representar a diferença entre os níveis de risco.

- Critical - o risco (r) é considerado crítico quando $4 \leq r \leq 5$;
- High - o risco (r) é considerado alto quando $3 \leq r < 4$;
- Medium - o risco (r) é considerado alto quando $2 \leq r < 3$;
- Low – o risco (r) é considerado alto quando $r < 2$;

Todavia, por via de regra, o projeto do RTRMM decidiu tratar somente os riscos que possuam valores $r \geq 3$. Esta decisão vai ao encontro de tratar os riscos que venham de uma forma direta afetar as funcionalidades do sistema IoT, riscos "Critical"e "High". Estes níveis de riscos vem afetar diretamente as funcionalidades dos dispositivos IoT, inviabilizando que o sistema IoT trabalhe de forma adequada e plena as suas funcionalidades, o que significa um alto impacto para a operação do sistema.

II – Estratégia Custo

A estratégia custo, que foi apelidada de custo operacional, está diretamente relacionada ao impacto que um dispositivo IoT poderá causar ao sistema IoT, quando este foi afetado por uma ameaça. Esta afirmação é válida porque retirar ou interromper o funcionamento deste dispositivo virá ocasionar problemas no funcionamento do sistema IoT.

Para analisar o custo operacional para um sistema IoT quando um ou mais dispositivos são afetados por uma ou mais ameaças, fatores como análise de impacto, capilaridade do sistema, e dispositivos interligados são levados em conta na avaliação. Sendo assim, o custo foi classificado em 2(duas) categorias, tendo como premissa que somente riscos críticos e altos serão tratados em tempo real pelo RTRMM.

- Critical – o custo operacional de um dispositivo IoT é considerado crítico quando este afeta diretamente a funcionalidade do sistema IoT (esta análise é realizada previamente em tempo de projeto), não possui uma capilaridade que alcance recursos alternativos para a realização das atividades de dispositivo, não possui alternativas em tempo real para substituí-lo, e afeta as funcionalidades de outros dispositivos dependentes deste.
- High – semelhante ao custo operacional crítico, ele também irá afetar as funcionalidades do sistema IoT em caso de não puder exercer as suas funcionalidades. Todavia, o sistema não deixa de exercer as suas atividades de forma plena, apesar de ter suas funcionalidades afetadas, por exemplo quanto ao desempenho, segurança ou mesmo disponibilidade.

Descrição do Dilema

O dilema da tratar ou não tratar está disposto da seguinte forma, 2 (dois) jogadores T (tratar) e NT (não tratar) e as estratégias S_T e S_{NT} baseadas nas estrategias puras de RISCO e CUSTO.

$$G = \{T, NT\}$$

$$S_T = \{RISCO, CUSTO\}, S_{NT} = \{RISCO, CUSTO\}$$

As combinações de estratégias podem ser representadas da seguinte forma:

$$S = \{(RISCO, RISCO), (RISCO, CUSTO),$$
$$(CUSTO, RISCO), (CUSTO, CUSTO)\}$$

As duas funções utilidade

$$u_T : S \to \mathbb{R} \text{ e } u_{NT} : S \to \mathbb{R}$$

são dadas por (representam os ganhos (payoffs) de N e NT). Os valores numéricos atribuídos foram obtidos do cálculo do risco e do custo operacional para o sistema IoT calculado com base na importância do dispositivo IoT para o sistema.

$$u_T = (RISCO, RISCO) = 3 \text{ , } u_{NT} = (RISCO, RISCO) = 3$$
$$u_T = (CUSTO, CUSTO) = 1 \text{ , } u_{NT} = (CUSTO, CUSTO) = 1$$
$$u_T = (RISCO, CUSTO) = 5 \text{ , } u_{NT} = (RISCO, CUSTO) = 1$$
$$u_T = (CUSTO, RISCO) = 1 \text{ , } u_{NT} = (CUSTO, RISCO) = 5$$

Uma forma de representar os payoffs é em matriz (tabela 8.4), no qual cada célula desta matriz está representa os payoffs de tratar (T) e não tratar (NT).

T	NT		
		RISCO	CUSTO
	RISCO	(3,3)	(5,4)
	CUSTO	(4,5)	(1,1)

Tabela 8.4: Matriz Payoff

Solução

A solução é a previsão sobre o resultado do jogo. Ao analisar a avaliação de riscos sob o prisma de tratar, pode ser realizado tanto em função do risco quanto do custo operacional. Da mesma forma pode se pensar quanto a não tratar.

Em ambas as situações foi aplicada a estratégia de dominância, ao invés de equilíbrio, pois os valores são dominantes perante aos outros quando comparado. Logo, aplicando a estratégia de dominância o resultado tanto para tratar ou não tratar é:

(RISCO,CUSTO)

Como pode ser analisado, tanto o custo quanto o risco afetam no tratamento ou não do risco calculado, mas a estratégia de risco é dominante sobre o custo. Isto significa que independente do custo operacional ser alto, o tratamento do risco deverá ser realizado caso ele seja crítico ou alto para o sistema o sistema IoT, aplicando a mesma estratégia de dominância do RISCO sobre o CUSTO.

Priorização de Tratamento

Desta forma, a priorização de tratamento se torna fácil, com base no que foi citado. Apesar de aplicar as estratégias de RISCO e CUSTO, a predominância de tratamento será sempre aplicada aos valores de maior risco. Isto significa que serão tratados os riscos com maiores valores, comparado aos valores de custo inicialmente, e posteriormente os riscos que possuam custos maiores que os valores de riscos.

Se RISCO $RISCO \geq CUSTO \equiv$ PRIORIZA TRATAMENTO

8.2.4 Módulo Threat Category

O módulo Threat Category tem como objetivo classificar as ameaças em categorias de forma que possam ser facilmente pesquisadas quando for selecionado o melhor mecanismo de segurança.

A estratégia de classificação adotada foi vincular categorias de ameaças semelhantes, visando reduzir o número o número de mecanismos de segurança a serem aplicadas no sistema IoT, bem como o número de entradas na base de dados agilizando o processo de pesquisa. Apesar de ser uma estratégia um tanto genérica, ela se faz útil devido a necessidade do RTRMM, gerenciar riscos em tempo real.

No caso do RTRMM, a definição das categorias, bem como o agrupamento é realizado em tempo de projeto, pelo administrador do sistema, refinando-as sempre que achar necessário. O administrador do sistema pode trabalhar com um histórico de ataques que o sistema IoT sofreu, características de funcionalidade do sistema, ou mesmo os diversos tipos de dispositivos IoT existentes e suas funcionalidades dentro do sistema. Estas informações servem como referência para o gestor de riscos estruturar uma base de dados com tais informações.

A estratégia de similaridade agrupa um conjunto de ameaças em uma mesma categoria, como DoS (Deny of Service), invasão, privacidade de dados, entre outras. Sendo assim, para categorizar uma ameaça, foram estabelecidas um conjunto de parâmetros para cada uma dessas categorias. Isto é possível porque cada tipo de ameaça possui características que são, na sua maioria independente entre si, ou mesmo integra mais de uma ameaça para ser efetivo em um ataque. Um bom exemplo é o ataque DDoS (Distributed Denial of Service), que utiliza 2 (duas) estratégias básicas: invasão e inundação. Vários hosts são invadidos (zumbis) e controlados por um host mestre. Em um determinado momento, todos os zumbis (conectados à mesma rede do alvo) acessam o mesmo recurso do mesmo servidor alvo. Os servidores possuem um número limitado de recursos, que se esgotam ao tentar atender simultaneamente todas as solicitações feitas por zumbis (estratégia de inundação).

Com o intuito de viabilizar a implementação desta estratégia, o RTRMM faz uso dos parâmetros utilizados pelo módulo Threat Analyzer, como portas de origem e destino, endereços de origem e destino e protocolo. A especificação de uma categoria de ataque pode ser bem definida com base, por exemplo, porta de destino, endereço de destino, protocolo, além de volume de tráfego ou mesmo randomização do endereço de origem.

Dinâmica do Threat Category

Ao receber os parâmetros da ameaça do Threat Analyzer, o Threat Category analisa os parâmetros, pré-estabelecidos pelo administrador do sistema, e a classifica em um categoria (figura 8.7). Porém, ainda fica uma questão, como analisar esses parâmetros e classificar a ameaça.

Sendo assim, foi adotada uma solução que faz uso de processo estocástico de tempo

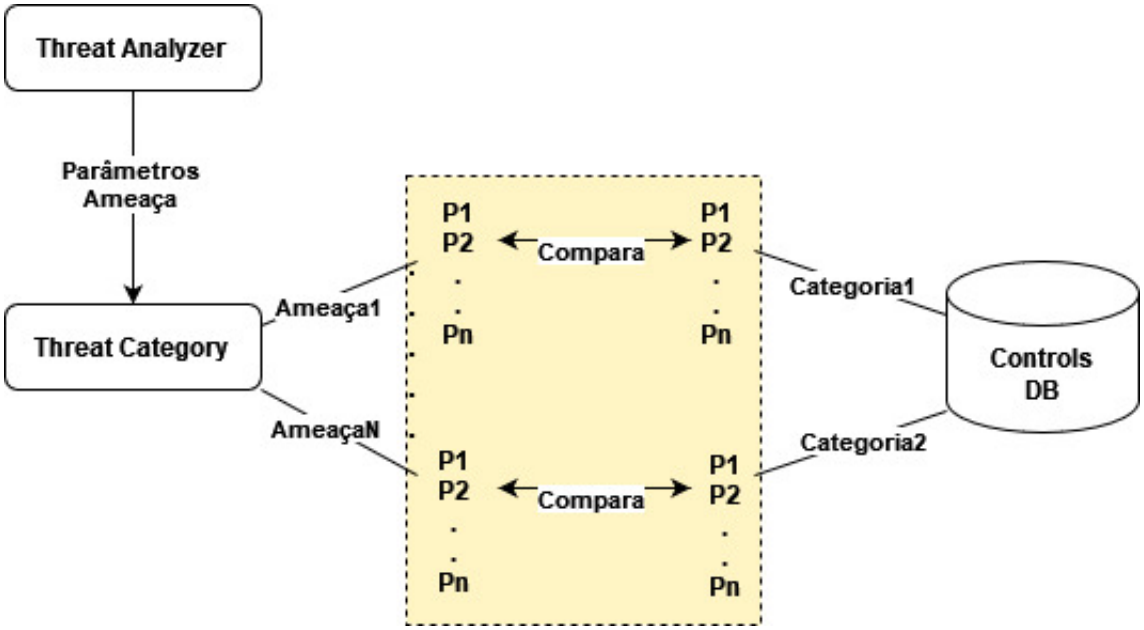

Figura 8.7: Dinâmica Threat Category

discreto, onde um grafo acíclico G = (V,E) [Wai02], possui um conjunto finito, não vazio, de vértices V e um conjunto de arestas A. Cada vértice deste grafo representa um parâmetro que compõe a ameaça detectada pelo Threat Analyzer, e o caminho entre dois ou mais vértices sequenciais definem uma categoria de ameaça.

Definição: para cada par de vértices sequenciais (u,v), em um grafo acíclico $G = (V, E)$, onde u e v são parâmetros de tráfego IoT, formam uma aresta de um caminho que representa uma característica da ameaça (ex: porta de destino).

Existe uma aresta apenas se: $e_v = \{e_1, e_2, ..., e_n\}$, onde e_v é um conjunto de eventos de segurança.

Se $(e_i \wedge e_n)\ \epsilon\ e_v \rightarrow \forall e_i, e_{i+1}\ \exists\ e_d$, onde e_d é uma aresta composta por (e_i, e_{e+1})

Definição: Existe um caminho em um grafo acíclico $G = (V, E)$ se somente se existir um conjunto de arestas (u,v).

$e_d = \{e_{d1}, e_{d2}, ..., e_{dn}\}$, onde e_d é um conjunto de arestas e_{di}.

Se $(e_{di} \wedge e_{di..n})\ \epsilon\ e_d \rightarrow \exists$ PATH | PATH $= \{e_{d1}, e_{d2}, ..., e_{dn}\}$

A construção do grafo é baseada nas informações coletadas no fluxo de dados IoT, ou seja, nos parâmetros de ameaça detectados. Para cada parâmetro de ameaça do tráfego IoT corresponde um vértice do grafo $G = (V, E)$. As arestas serão criadas interligando esses vértices formam vários caminhos. Cada um destes caminhos corresponderá a uma

categoria de ameaça. Todas as categorias de ameaças poderão ser definidas tanto previamente, armazenadas em um banco de dados ou em tempo real, possibilitando que categorias não previamente definidas possam assim se adequar a diversidade de ameaças existentes no mundo real.

A definição de categorias em tempo real, se baseia também nos parâmetros que foram obtidos pelo Threat Analyzer. Todavia, ao se constatar que os parâmetros não se adequam a qualquer uma das categorias, o administrador do sistema atribui um nome a esta e assim a mesma fará parte de categorias já existentes.

Por exemplo, suponha um gráfico acíclico, como a figura 8.8, com $n = |N|$ e $m = |E|$, onde n é o número dos vértices e m o número das arestas.

$t \epsilon T = \{1, 2, 3, 4, 5, ...\}$.

$$X_t = \begin{cases} Threat(Cat), & \text{if s and t in V(G) there is a path} \\ No - threat, & \text{not exist a path} \end{cases}$$

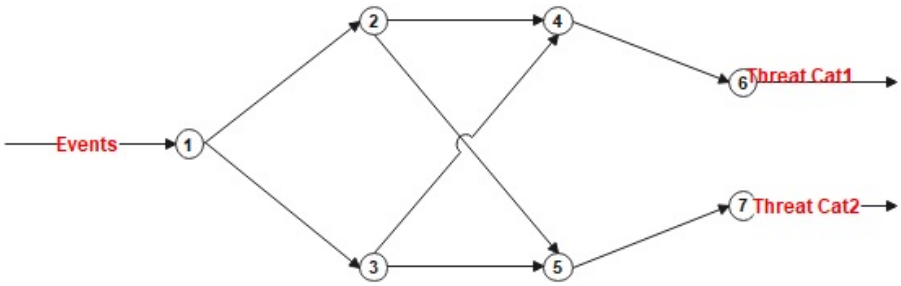

Figura 8.8: Categoria da Ameaça

A estratégia de agrupar as ameaças em categorias no RTRMM, reduz a quantidade de entradas a serem pesquisadas no módulo Controls DB. Para cada categoria existe ao menos 1 (um) mecanismo de segurança atrelada a mesma. Desta forma, quantidade de entradas fica limitada somente às categorias e não a todo e qualquer tipo de ameaça, reduzindo o tempo de resposta em selecionar o mecanismo de segurança a ser aplicado no sistema IoT.

8.2.5 Módulo Controls DB

O módulo Controls DB tem como objetivo armazenar e disponibilizar para o administrador do sistema qual ou quais mecanismos de segurança que poderão ser aplicados no sistema IoT, possibilitando que se minimize os riscos existentes. É talvez um dos módulos mais interessantes, pois o mesmo integra os resultados dos demais módulos do RTRMM e entrega uma solução para mitigar os riscos correlacionados ao ativo que está sofrendo um incidente de segurança.

O módulo Controls DB consiste em uma base de dados com a categoria de ameaça

e o mecanismos de segurança a ser aplicado. Todavia, ele também é responsável em correlacionar o risco (a ser tratado) com a categoria da ameaça, possibilitando a seleção do mecanismo de segurança que será aplicado.

Para que possa realizar as suas tarefas a base de dados do Controls DB é alimentada simultaneamente pelo módulo Threat Category, que informa qual categoria aquela ameaça pertence e o módulo Risk Management, que informa qual risco a ser tratado (envolve a os parâmetros da ameaça e o ativo a ser tratado), possibilitando que este possa realizar a equalização/processamento e armazenamento das informações.

Com o intuito de facilitar a aplicação do RTRMM, isto é, a sua implantação, tomou-se, inicialmente, a decisão de que os mecanismos de segurança a serem aplicados fossem todos definidos e armazenados na base de dados em tempo de projeto pelo administrador do sistema. A escolha destes mecanismos pode ser realizada com base em um conjunto de soluções oferecidas pelo mercado, pelo fornecedor do dispositivo IoT, ou mesmo pela aplicação dos controles de segurança existentes no Anexo A da ISO 27001 [ISO22b].

A estratégia de automatizar ao máximo a tomada de decisões na escolha da melhor solução de segurança a aplicar para reduzir os riscos reduz a carga de trabalho dos gestores de segurança da IoT e melhora o desempenho da gestão de riscos. Todavia, o RTRMM ainda delega a responsabilidade ao administrador do sistema a aplicação ou não do mecanismo de segurança. A delegação de responsabilidade de decisão de aplicação se deve ao fato do fator humano ainda ter a possibilidade de evitar quaisquer danos maiores ao sistema IoT, principalmente pelo conhecimento do seu perfil de funcionamento.

Dinâmica Controls DB

I - Risco X Categoria

Como citado, a composição da categoria é baseada nos parâmetros de detecção da ameaça pelo Threat Analyzer (número de portas, protocolo, endereço IP, ...). Em paralelo, o risco a ser tratado está diretamente relacionado a um ativo e a ameaça correlacionada ao risco.

Desta forma, o Controls DB tem como uma das suas tarefas realizar a correlação entre esses parâmetros. Esta relação é realizada via a ameaça, identificada via os parâmetros (porta,endereço IP, protocolo) informados pelo módulo Risk Management, ao informar qual risco será tratado, em conjunto com os mesmos parâmetros (porta, endereço IP e protocolo) que o Threat Category faz uso para determinar qual a categoria.

Com o batimento destes parâmetros, a categoria identificada, em conformidade com o risco a ser tratado, realiza-se a pesquisa no Controls DB e se determina qual mecanismo de segurança será aplicado no sistema IoT.

II - Categoria X Mecanismo de Segurança

Para realizar a ligação da categoria de uma ameaça às melhores medidas de segurança foi adotada a estratégia de processo estocástico, representada por um grafo acíclico,

conforme figura 8.9. Cada vértice do gráfico representa um elemento de segurança que a solução pretende implantar. Conforme citado, a base de dados já possui uma carga de mecanismos de segurança controles como: bloquear tráfego, desativar um dispositivo IoT, ou desativar um serviço, por exemplo.

A busca na base de dados por um mecanismo de segurança é a princípio sequencial, mas podendo posteriormente estruturar a base para que a pesquisa seja realizada de forma mais eficaz, reduzindo o tempo.

III - Estrutura do Grafo

Suponha momentos em tempos distintos t, representado por um vértice do gráfico. Cada um desses vértices corresponde a um componente de segurança que poderá fazer parte do mecanismo de segurança. Cada componente de segurança corresponde a uma característica da categoria da ameaça. Por exemplo, categoria de negação de serviço: pode possuir componentes relacionados ao volume de tráfego, largura de banda, que levará ao mecanismo de desativação de um dispositivo IoT.

$t \epsilon T = \{1, 2, 3, 4, 5, ...\}.$

$$X_t = \begin{cases} SecuritySolution, \text{se s e t em } V(G) \text{ existe um caminho} \\ No - SecuritySolution, \text{não existe um caminho} \end{cases}$$

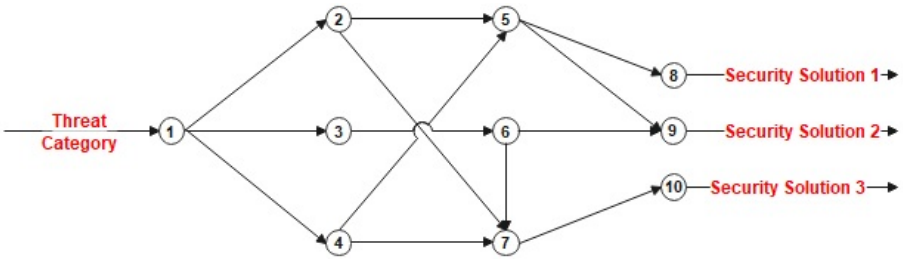

Figura 8.9: Seleção Mecanismos de Segurança

A entrada do grafo é a categoria de ameaça e, se houver um caminho para no final do grafo é porque foi realizada uma ligação entre a categoria da ameaça e uma solução de segurança. Caso exista mais de uma solução de segurança, decidir qual solução será aquela que oferece mais elementos de segurança para a solução, ou seja, o caminho que tem a maior contagem de vértices.

8.3 Considerações

O RTRMM é uma nova proposta de gestão de riscos em sistemas IoT. Durante a sua especificação foram estudadas e analisadas várias tecnologias. Algumas decisões de

projeto foram tomadas em função das funcionalidades que o RTRMM oferece. Talvez o modelo ainda necessite de mais desenvolvimento, como no cálculo de impacto.

Todavia, a sua estrutura lógica está madura a ponto de oferecer funcionalidades que não estão previstas em outros modelos de gestão de riscos para sistemas IoT. Técnicas de inteligência artificial foram utilizadas com o intuito de prover mais confiança e aproximação de uma realidade de riscos.

Aspectos como a utilização de modelo matemático para trabalhar questões de incerteza e prover uma visão ampla da análise e avaliação de riscos foi utilizada. Espera-se que este modelo venha prover ao menos uma pequena contribuição para o universo da segurança da informação, sendo um modelo de referência para outros que venham posteriormente.

9

Protótipo RTRMM

A capítulo tem como objetivo apresentar um protótipo com alguns módulos do RTRMM implementados, tendo como base o modelo lógico apresentado na figura 8.1, bem como alguns testes de desempenho realizados. Nem todos os módulo deste protótipo foram implementados e testados, mas aqueles realizados tiveram como objetivo validar as funcionalidades propostas, bem como a sua eficácia. As funcionalidades dos módulos implementados foram realizados com base em programação lógica e lógica probabilística, com programas escritos em Prolog e Problog [Rae07].

9.1 Implementação do Módulo Threat Analyzer

Esta seção tem como objetivo apresentar um exemplo de como o módulo Threat Analyzer pode ser implementado. Neste exemplo, ele foi implementado em Problog e fez uso do IoT-23, um conjunto de dados de tráfego de rede de dispositivos da Internet das Coisas (IoT), gerado pelo "Stratosphere Laboratory, AIC group, FEL, CTU University,

Czech Republic"[Lab20], que visa oferecer um grande conjunto de dados, infectados por malware ou benigno, de dispositivos IoT reais.

O código fez uso de 6 campos (parâmetros de identificação do tráfego IoT) da base de dados IoT-23, simulando entradas Crisp, como no processo de fuzzificação: protocolo; data do evento; endereço IP de origem e o de destino; e porta de origem e porta de destino.Inicialmente. Foram especificadas 3 anomalias, baseadas na estratégias de regras de fuzzificação, e nos parâmetros de malwares apresentados no IoT-23. Para cada uma das anomalias foram selecionados três (3) parâmetros de identificação do tráfego IoT para a sua composição: protocolo, endereço IP de destino e porta de destino (figura 9.1).

```
%loading the database
  :- use_module(library(db)).
  :- csv_load('out_lim1.csv', 'pacote').
%the first anomaly
  0.3::anomaly(A,B,C,D,E,F):- pacote(A,B,C,D,E,23).
  0.6::anomaly(A,B,C,D,E,F):- pacote(tcp,B,C,D,E,F).
  0.7::anomaly(A,B,C,D,E,F):- pacote(A,B,C,'65.127.233.163',E,F).
%the second anomaly
  0.5::anomal_1(A,B,C,D,E,F):- pacote(A,B,C,D,E,49560).
  0.7::anomaly_1(A,B,C,D,E,F):- pacote(tcp,B,C,D,E,F).
  0.6::anomaly_1(A,B,C,D,E,F):- pacote(A,B,C,'147.7.65.203',E,F).
%the third anomaly
  0.4::anomaly_2(A,B,C,D,E,F):- pacote(A,B,C,D,E,60862).
  0.6::anomaly_2(A,B,C,D,E,F):- pacote(udp,B,C,D,E,F).
  0.8::anomaly_2(A,B,C,D,E,F):- pacote(A,B,C,'51.148.125.188',E,F).
%probability of query
  query(anomaly(tcp,_,_,'65.127.233.163',_,23)).
  query(anomaly_1(tcp,_,_,'147.7.65.203',_,49560)).
  query(anomaly_2(udp,_,_,'51.148.125.188',_,60862)).
```

Figura 9.1: Codificação do Threat Analyzer

A estratégia de inferir o valor da probabilidade para cada parâmetro na composição de cada anomalia foi, inicialmente baseado na análise/avaliação dos eventos apresentados no IoT-23. A estratégia analisou a quantidade de ocorrências de cada parâmetro na base de dados IoT-23 na razão N/M. O valor N corresponde à quantidade de ocorrências de um parâmetro na base de dados IoT-23, e M a quantidade total de fluxos de dados na base IoT-23. Estes valores foram inseridos em um dos nós da rede markov, que possibilitou o valor da probabilidade para que um pacote seja ou não uma ameaça.

9.2 Risk Management: Implementação

A implementação do módulo Risk Management foi realizada em Problog, tendo como entrada uma base de dados com as anomalias e suas respectivas probabilidades detectadas

pelo Threat Analyzer. Esta base continha onze (11) entradas (anomalias e probabilidades), onde cinco(5) eram de um tipo de anomalia, mais cinco (5) de outro tipo e, por fim, uma única ocorrência do terceiro tipo de anomalia, para cada tipo de anomalia o sistema calculou a probabilidade destas serem uma ameaça ou não, conforme pode ser constatado na figura 9.2.

```
root@retrimm:/bigfatdisk/otavio/teste# problog te.pl
     anomaly(tcp,'20180509-163031','192.168.100.103','65.127.233.163',51524,23):        0.916
     anomaly(tcp,'20180509-163032','192.168.100.103','65.127.233.163',51524,23):        0.916
     anomaly(tcp,'20180509-163034','192.168.100.103','65.127.233.163',51524,23):        0.916
     anomaly(tcp,'20180509-163038','192.168.100.103','65.127.233.163',51524,23):        0.916
     anomaly(tcp,'20180509-163046','192.168.100.103','65.127.233.163',51524,23):        0.916
  anomaly_1(tcp,'20180509-163033','192.168.100.103','147.7.65.203',34243,49560):        0.88
  anomaly_1(tcp,'20180509-163034','192.168.100.103','147.7.65.203',34243,49560):        0.88
  anomaly_1(tcp,'20180509-163036','192.168.100.103','147.7.65.203',34243,49560):        0.88
  anomaly_1(tcp,'20180509-163040','192.168.100.103','147.7.65.203',34243,49560):        0.88
  anomaly_1(tcp,'20180509-163048','192.168.100.103','147.7.65.203',34243,49560):        0.88
anomaly_2(udp,'20180509-163034','192.168.100.103','51.148.125.188',43763,60862):        0.952
```

Figura 9.2: Resultado Anomalias Detectadas

Com base nessa escala, o sistema responsável pela análise de riscos criou uma base de dados em Prolog, composto pelo tipo da anomalia e sua probabilidade, além dos níveis de ameaça proposto pelo RTRMM (Likely, Almost, Sometimes, Unlikely).Criou-se uma regra em Prolog que compara o valor da probabilidade calculada para cada anomalia com os níveis de ameaça. Os seguintes resultados foram obtidos:

1. Anomaly - esta anomalia teve 0.916 como probabilidade, sendo considerada Likely, uma alta possibilidade de ser uma ameaça.

2. Anomaly_1 - esta anomalia teve 0.88 como probabilidade, sendo considerada Almost, existe a possibilidade de ser uma ameaça.

3. Anomaly_2 - esta anomalia teve 0.952 como probabilidade, sendo considerada Likely, uma alta possibilidade de ser uma ameaça.

Considerando que estas anomalias são uma ameaça, o próximo passo do Risk Management do RTRMM é avaliar quais dessas ameaças serão tratadas e o seu grau de prioridade. O RTRMM assumiu que para qualquer ameaça que possua o grau de relevância (probabilidade) $\mu S \geq 0.7$ deverá ser tratado. A razão dessa decisão tem como objetivo reduzir a taxa de falsos positivos/negativos, melhorando a probabilidade de tratamento de ameaças detectadas. A taxa inicial de falsos positivos/negativos, sem aplicação do valor do grau de relevância $\mu S \geq 0.7$ foi cerca de 20% ocorrências, considerada muito alta. Com a aplicação do valor de relevância essa a taxa teve uma redução em torno de 60%, excluindo as ameaças com pouca probabilidade de serem verdadeiras, melhorando a confiabilidade de validação de ameaças.

Quanto a avaliação do risco, tomou-se a estratégia de priorizar o tratamento das ameaças do maior valor ao menor valor. Esta estratégia é válida porque o valor do risco está diretamente ligado ao valor da probabilidade da ameaça. Logo, o risco de maior valor será tratado primeiro que os demais.

O RTRMM, com base na aplicação desenvolvida, adotou inicialmente que a ordenação das probabilidades calculadas pelo sistema determinará a prioridade de tratamento. Isto significa que a sequência de tratamento será: Anomaly_2; Anamoly; e Anomaly_1. Assim, ao se calcular o risco, ter-se-a condições de priorizar o risco mais alto, atendendo os ativos mais críticos para o sistema IoT.

9.2.1 Cálculo do Risco

O resultado do programa em Problog apresentou o número de ocorrências de uma ameaça em tráfegos IoT, além das probabilidades dessas ameaças ocorrerem, agrupadas em ocorrências. No exemplo executado, pode-se observar que todas as ameaças devem ser tratadas, pois todas possuem valores $\mu S \geq 0.7$.

Para se calcular o risco, tem que se conhecer, além da probabilidade da ameaça, o valor de impacto e a probabilidade do ativo estar vulnerável. Na implementação inicial do RTRMM, para se calcular o risco foi levado em consideração somente a probabilidade da ameaça e o valor do impacto. O valor do impacto inicial do ativo, está baseado na avaliação da importância do mesmo para o sistema IoT. Este valor foi estabelecido pelo administrador do sistema em tempo de projeto.

A aplicação para cálculo do risco foi desenvolvida em Prolog, no qual foi estabelecida uma base da dados com os valores de probabilidade e impacto. De posse desses valores, tem-se condições de calcular o risco. Porém, para efetivar o cálculo, foi realizada uma verificação se a probabilidade atende as condições previstas (ser alto ou crítica), bem com o valor do impacto.

Atendendo as condições, a função de cálculo de risco foi implementada de forma que o valor do risco é calculado, se somente se, a ameaça realmente for alta/crítica para o ativo do sistema IoT. Para exemplificar o cálculo do risco, criou-se uma base de dados para probabilidade e impacto. Os valores selecionados foram com o intuito de apresentar um resultado que venha a tratar os riscos. A base de dados poderia ter vários outros valores, mas como exemplo tomamos estes apresentados.

Base de dados com valores de probabilidade e impacto

```
prob(alto, 0.75).
prob(alto, 0.8).
impact(alto, 4).
impact(alto, 5).
```

Somente calcula o risco se satisfizer os valores

```
valor(Y, Z) :-
    prob(X, A),
    Y >= A,
    impact(W,B),
    Z >= B, !.
```

Cálculo de risco

```
risco(V, Y*Z):-
    risco(V1, Y), risco(V2,Z), total(V1,V2,V).
```

9.3 Threat Category

O módulo Threat Category tem como ideia de implmentação inicial se basear basear nas informações provenientes do módulo Threat Analyzer. Ele possui uma base de dados, inicalmente textual, que armazena cada categoria, e seus respectivos parâmetros que possibilitaram a sua criação.

O RTRMM adotou como estratégia de agrupar ameaças em categorias baseada em no número da porta e do protocolo. Desta forma, toda e qualquer categoria possui um identificador e seus respectivos parâmetros (número da porta e protocolo).

Como seria o passo a passo das atividades deste módulo:

1. Recebe os parâmetros da ameaça;
2. Identifica o número da porta e o protocolo;
3. Compara, na base de dados, se os parâmetros coletados existem;
4. Caso existam, retorna informando da existência; e
5. Caso não exista, atribui um identificador para a noca categoria e cria uma nova linha com o identificador e seus respectivos parâmetros.

9.4 Desempenho

O RTRMM tem como objetivo tratar os riscos em ambientes IoT de forma proativa, minimizando os impactos de segurança que esse ambiente possa vir a sofrer. Com o objetivo de melhor verificar o desempenho do protótipo desenvolvido, foi realizada uma análise de desempenho com esta finalidade. O protótipo foi testado em um ambiente sem rede e tráfego simulados. Usamos uma base de dados existente [Lab20] com o objetivo de validar a viabilidade do RTRMM ser implementado. A figura 9.3 apresenta duas diferentes situações que foram avaliadas seu desempenho:

- Situação 1 (imagem 1) – nesta imagem cada anomalia trabalhou com 3 parâmetros (protocolo, endereço de destino e porta de destino), e o sistema utilizou 3, 5 e 7 anomalias para análise de tempo.

  Pode ser observado que o sistema gastou pouco menos de 1 segundo para detectar 3 anomalias; em torno de 3 segundos para detectar 5 anomalias; por fim mais de 5 segundos para 7 anomalias.

O que se pode observar com estes valores é que o tempo gasto é relativamente proporcional ao aumento de 2 anomalias quando se trabalha com 3 parâmetros.

- Situação 2 (imagem 2) - nesta imagem cada anomalia possuía 5 parâmetros (protocolo, endereço de origem e destino e porta de origem e destino), e o sistema utilizou 3, 5 e 7 anomalias para análise de tempo.

  Pode ser observado que o sistema gastou 3 segundos para detectar 3 anomalias; 5 segundos para detectar 5 anomalias; por fim 7 segundos para 7 anomalias.

  O que se pode observar com estes valores é que o tempo gasto é retilíneo ao aumento de 2 anomalias quando se trabalha com 5 parâmetros.

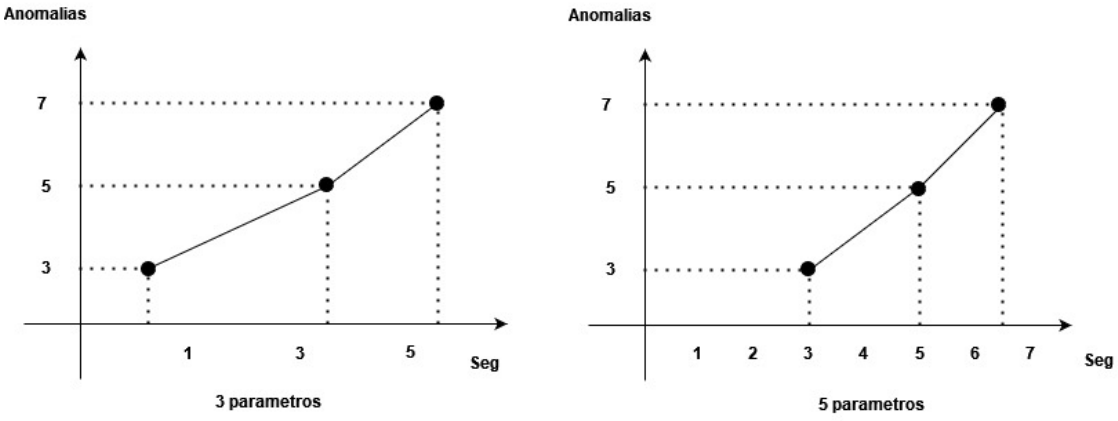

Figura 9.3: Run-time Comparação

Algumas considerações podem ser observadas quanto ao desempenho do protótipo desenvolvido:

- O protótipo apresentou um comportamento desempenho considerado adequado, tendo a sua taxa de crescimento de tempo praticamente proporcional ao crescimento do número de anomalias.
- O aumento de parâmetros a serem avaliados acarreta uma leve perda de desempenho, mas não impede a sua aplicação.
- Vale ressaltar que o protótipo desenvolvido apresentou requisitos válidos de que o RTRMM pode ser implementado e possivelmente aplicado.

9.5 Considerações

O objetivo deste protótipo era validar as principais funcionalidades do RTRMM, além de comprovar a validade de sua implementação. A sua implementação comprovou a disponibilidade de recursos para prover um gerenciamento proativo, uma das principais funcionalidades do RTRMM.

Todavia, a sua validação plena dependerá da implementação e aplicação das funcionalidades já validadas, em conjunto com as que ainda necessitam ser trabalhadas.

10

Comparação com o Estado da Arte

Os capítulos anteriores tiveram o objetivo de fundamentar os conceitos aplicados para o desenvolvimento deste trabalho, e a especificação, implementação e testes do modelo de gestão de riscos desenvolvido, o RTRMM.

Todavia, nenhuma descrição de melhorias e inovações em relação aos diversos modelos e metodologias estudados e analisados não foram apresentadas ou mesmo discutidas. Sendo assim, este capítulo tem como objetivo apresentar uma abordagem de comparativa entre o RTRMM e estes modelos e normas de gerenciamento de riscos de segurança que foram estudadas e analisadas durante este trabalho.

10.1 Proatividade e RTRMM

Como já citado gerenciar riscos de segurança, seja de forma reativa ou proativa, é fundamenta ou mesmo estratégico para os processos de negócio de uma organização. A

gerência reativa tem as suas vantagens, pois lida com as consequências dos eventos de segurança (incidentes) após ocorrerem. Em contrapartida, a gerência proativa de riscos de segurança tem como objetivo procurar evitar que incidentes venham acontecer, ou mesmo reduzir as consequências de incidentes, prevendo que os mesmos possam vir a ocorrer.

Logo, gerir riscos de segurança de forma proativa proporciona grandes benefícios aos sistemas, evitando incidentes antes que ocorram, possibilitando reduzir a probabilidade que as funcionalidades de sistemas se tornem inoperantes de forma plena. Esta técnica de gestão de riscos promove uma cultura de prevenção dando aos membros de uma equipe de segurança da informação uma maior responsabilidade no processo de gestão dos riscos.

Se for pensar no formato de sistemas IoT, no qual a comunicação entre dispositivos, infraestrutura de computação e clientes, a gestão proativa traz um benefício enorme para este universo. Se for pensar sob o prisma das propriedades de segurança (tripé de segurança), confidencialidade, integridade e disponibilidade, prover mecanismo de segurança com o intuito de prevenir e manter um monitoramento e controle do tráfego das informações armazenadas e trafegadas é com certeza sucesso para os sistemas IoT, pois irão garantir essas propriedades, dando mais confiança aos seus usuários.

O RTRMM, o modelo apresentado neste trabalho, segue as mesmas ideias do processo de gestão proativa de riscos de segurança da informação. Sua estrutura lógica possibilita prever possíveis ameaças, calcular os riscos de segurança e apresentar mecanismos de segurança, a fim de reduzir os riscos detectados, em tempo real. Esta seção vai apresentar um pouco do que vem a ser esta forma de gestão de riscos e como o RTRMM se adequa a essa realidade.

Devido a dinâmica do mundo digital, inclusive o universo IoT, gerenciar riscos, segundo [Kal06], tem que ser forma proativa e não reativa. Esta afirmação vem ao encontro de trabalhar de forma proativa possibilita mudar as probabilidades de sucesso de defesa contra ataques de hackers a favor da organização, pois conforme os riscos forem detectados estes poderão ser minimizados antes de causarem danos que venham a prejudicar de forma efetiva os processos de negócio de uma organização.

Desta forma, vale a pena ressaltar que o monitoramento de um ambiente computacional com a intenção de verificar novos riscos nunca poderá ser subestimado, pois a sua ausência por si só pode significar o fim da proatividade do processo de gestão de riscos. Isto porque em uma abordagem proativa a prevenção de incidentes, para manter os sistemas IoT operando busca sempre a identificação e a prevenção de ameaças antes que elas se tornem uma realidade. Logo, prever e antecipar problemas possibilita desenvolver estratégias, políticas e práticas que visa mitigar os riscos e promover um ambiente mais seguro.

10.1.1 Ser Proativo

Para que um sistema possua gerenciamento proativo de riscos, os seguintes componentes devem ser atendidos:

1. Entender as possíveis situações de riscos – para isso é necessário uma análise dos processos de negócio, além dos procedimentos e interações entre os dispositivos IoT no sistema. Realizar uma análise e avaliação de riscos com o objetivo de identificar os riscos mais críticos do sistema IoT. Conhecido os riscos críticos, existe a necessidade de aplicar, previamente, os mecanismos de segurança com o objetivo de prevenir incidentes antes que ocorram.

2. Implementar medidas de segurança preventivas e adequadas às necessidades do sistema – as medidas de segurança a serem aplicadas vão ao encontro dos riscos que foram detectados na fase de análise e avaliação dos riscos. Para isso é necessário a criação de procedimentos e políticas de segurança para estar em conformidade com a implantação e manutenção dos mecanismos de segurança.

3. Desenvolver procedimentos e políticas de segurança – o desenvolvimento destes componentes de segurança dão suporte aos mecanismos de segurança implantados. Para isso é necessário a colaboração de todos os envolvidos, gestores e administradores de processos de negócio, colaboradores, ... Fornecer treinamento adequado para a utilização dos mecanismos implantados.

4. Monitoramento e Controle – implantados os mecanismos de segurança, a necessidade de um monitoramento constante e avaliação da eficácia dos controles possibilita verificar se os mecanismos estão em conformidade com o planejado ou se necessitam de ajustes.

10.1.2 O RTRMM Proativo

O RTRMM é um modelo que trabalha de forma proativa em relação a gestão de riscos em sistemas IoT. As suas caraterísticas vão ao encontro das caraterísticas de um sistema de gestão de riscos proativo, pois:

1. Detecta as ameaças antes delas serem efetivadas, possibilitando uma análise e avaliação dos riscos antes que incidentes venham acontecer. O módulo Threat Analyzer é o responsável em realizar a detecção de ameaças.

2. O módulo Risk Management busca analisar e avaliar os riscos em tempo real, conhecendo de forma efetiva os riscos que o sistema IoT possui, possibilitando agir de forma proativa, nas possíveis tomadas de decisão em relação às medidas de segurança.

3. A possibilidade de aplicar as medidas de segurança em conformidade com os riscos calculados, em tempo real, é uma outra funcionalidade que provê ações proativa

de gestão de riscos, pois possibilita que os riscos possam ser reduzidos antes que as ameaças se concretizem ou mesmo venham causar danos maiores ao sistema IoT.

Todavia, existe ainda funcionalidades que ficam a mercê da equipe de gestão de segurança da informação, atividades que irão trabalhar em conjunto com a gestão proativa como: atividades de criação de procedimentos e políticas de segurança, bem como o monitoramento e controle dos mecanismos de segurança.

10.2 RTRMM e a Norma Internacional ISO 27005

A norma internacional de gestão de riscos ISO 27005 [ISO22b] é norma que serve como referência para o processo de gestão de riscos em qualquer tipo de ambiente computacional, inclusive sistemas IoT. A sua estrutura robusta e concreta, com funcionalidades de vão desde a definição do contexto do que será gerido até a forma pelo qual serão tratados os riscos calculados durante todo o processo.

A estrutura lógica do módulo Risk Management, do RTRMM, foi baseada na norma ISO 27005 [ISO22b]. A escolha pela ISO 27005 se baseou na forma como esta aborda o processo de gestão de riscos. Ela trabalha de forma separada e abrangente, levando e consideração os aspectos considerados mais críticos em um processo de gestão de riscos, como as tarefas de análise de impacto, análise e avaliação de riscos. Um outro fator considerado importante é que a norma apresenta uma estratégia que viabiliza os cálculos de riscos e sua respectiva avaliação. Todavia, nem todas as funcionalidades da norma ISO 27005 foram aplicadas, sendo utilizadas basicamente a estratégia dos componentes de cálculo de impacto, análise e avaliação de riscos. A escolhas por aplicar essas estratégias se deve a facilidade de adequar estas funcionalidades a um ambiente de tempo real e de gestão proativa.

Vale a pena ressaltar que, em paralelo, também foram analisadas outras normas como o COBIT [Ahm17], e a NIST 800-30 [Bla12]. Todas estas metodologias tratam a gestão de riscos com base no cálculo do impacto e avaliação de riscos como um todo, não tratando os processos em separado, provendo informações que possibilitem estabelecer uma interligação entre todas as atividades do processo de gestão de riscos. Esta estratégia dificulta a formalização de funcionalidades, separadas, viabilizando uma implementação clara e objetiva.

10.2.1 Estratégias

A estratégia da análise de impacto da ISO 27005 se baseia na identificação dos ativos a serem analisado e protegidos, o seu grau de importância e o impacto que ele irá determinar para o sistema. O RTRMM adota a mesma estratégia, identificando quais dispositivos IoT devem ser protegidos, estabelecido um conjunto de parâmetros para determinar o grau de importância do dispositivo, e por fim o impacto com base no grau de importância calculado.

Na análise de riscos, a ISO 27005 propõe a identificação das ameaças, a probabilidade de cada uma ter sucesso em explorar as vulnerabilidades do sistema e por fim calcular o risco, com base em uma análise qualitativa, dividindo em níveis em conformidade com o estabelecido com a Organização que irá realizar. Inclusive, nesta fase, é definida o nível de aceitação de riscos para uma organização. Isto significa em citar até que nível de riscos a organização irá aceitar e não tratar estes riscos.

10.2.2 Diferenciais do RTRMM

O RTRMM apesar da fazer uso de algumas estratégias da ISO 27005, apresenta alguns diferenciais. O RTRMM trabalha de forma qualitativa e quantitativa, sendo que as duas estratégias se entrelaçam na obtenção de valores. A abordagem quantitativa se deve ao fato que o RTRMM faz de uso probabilidades para calcular, por exemplo, se existe ou não uma ameaça em um tráfego IoT. Todavia, faz uso também de uma abordagem qualitativa, pois para cada de Intervalo probabilidades é apresentado um valor qualitativo (Really).

O processo de avaliação de riscos na ISO 27005 e no RTRMM seguem a mesma ideia, mas o RTRMM apresenta um método de classificação e determinação se o risco será ou não tratado diferente da norma. A ISO classifica pelo grau de risco e dá a organização a autoridade de tratar ou não o risco. O RTRMM avalia o risco em conformidade com alguns parâmetros, como custo operacional e a prioridade para as funcionalidades do sistema IoT. Este diferencial proporciona uma melhor avaliação, pois além da prioridade o RTRMM leva em consideração o quanto será custoso para o sistema e a organização a implementação do mecanismo de segurança.

O processo de tratamento de riscos no RTRMM trabalha com um conjunto de mecanismos de segurança que estão atrelados a um conjunto de ameaças. Em contrapartida, a ISO 27005 somente apresenta se o risco será tratado, ou delegado a terceiros, ou mesmo descartado. A ISO 27005 apresenta uma matriz de riscos, diferente do RTRMM, que com todas as informações referentes ao processo de análise e avaliação de riscos, possui um universo de maior de informações para a tomada de decisão.

10.3 RTRMM X Modelos de Gerenciamento de Riscos

Apresentado os diferenciais do RTRMM em relação a ISO 27005 e as razões deste ser proativo, esta seção tem como objetivo apresentar uma análise comparativa entre o RTRMM e alguns modelos de gerenciamento de riscos IoT que foram estudados durante a realização deste trabalho. Esta análise versará aspectos que diferenciam, de forma positiva, apresentando as razões de se especificar e implantar o RTRMM em qualquer sistema IoT.

10.3.1 Modelos de Gerenciamento de riscos para Saúde

O artigo [ZAHS19, SBH+19] apresenta um modelo de gestão de riscos baseado no COBIT5 [Ahm17], focado em gestão de riscos para o ambiente de saúde. Apesar de não se ter uma previsão de todos os riscos em um ambiente IoT de saúde, o modelo apresentado visa atender os objetivos de controle da informação e tecnologia. O autor do modelo optou por utilizar o COBIT5 pois o mesmo proporcionar uma visão holística do negócio, facilitando a gestão eficiente de TI de maneira que o setor possa contribuir para o crescimento de toda a organização. Logo, o COBIT5 é focado em características empresariais, e não estando limitado somente ao domínio de TI, se adaptando bem a situação que o artigo propõe.

O artigo cita que a principal razão da implantação de um modelo de gestão de riscos, em um sistema de saúde, é a falta de um mecanismo centralizado de segurança e de controle de riscos dos dispositivos IoT. Desta forma, com o modelo proposto pelo artigo, se espera reduzir os riscos, tendo em vista que as funcionalidades de sistema de saúde a ser protegido está diretamente associado a sistemas IoT. Segundo o artigo, esta implantação levará a uma melhor eficiência, desempenho, eficácia, inovação, qualidade de conexão e retrata uma nova ideia de solução, reduzindo os custos, o tempo e as tarefas operacionais.

O modelo de gestão de riscos apresentado é composto por 3 (três) partes: o Healthcare IoT Risk Management; alinhamento HPIA; e o COBIT5, além das fases de implementação como pode ser visualizado na figura 10.1.

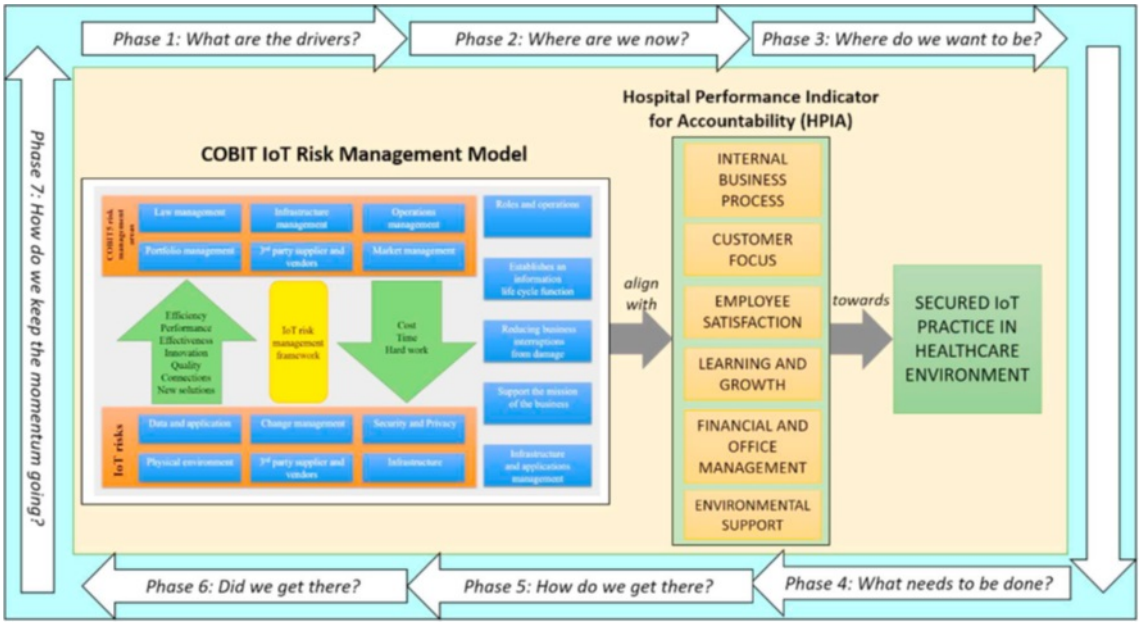

Figura 10.1: Modelo Proposto para Gestão de Riscos em Healthcare [ZAHS19]

A primeira parte do modelo, o COBIT IoT Risk Management, é composto com base na categoria dos riscos IoT (dados e aplicativos, gerenciamento de mudanças do usuário, segurança e privacidade, ambiente físico e infraestrutura e fornecedores), e a segunda parte nas áreas de gerenciamento do COBIT 5 (gerenciamento jurídico, gerenciamento

de infraestrutura, gerenciamento de operações, gerenciamento de portfólio, fornecedores e gerenciamento de mercado). Estas partes interagem entre si e proporcionam uma visão de quais riscos serão gerenciados e os setores atrelados a eles.

O modelo incorpora então as categorias HPIA que são Processos de negócios internos, com foco no cliente, satisfação de funcionários, aprendizagem e crescimento, gestão financeira e apoio ambiental. O modelo está alinhado ao KPI [1] de qualidade dos cuidados de saúde, e a HPIA [2] (Hospital Performance Indicator for Accountability) é uma regra obrigatória a ser seguida.

O modelo propõe então sete fases de implementação originadas no COBIT5 para orientar o processo de implementação do sistema IoT desde o seu início:

1. Quais são os objetivos? – são estabelecidos os objetivos que visam identificar e confirmar a necessidade de implementação do sistema IoT;

2. Qual é o escopo? – onde for necessário definir o escopo da implementação usando o mapeamento dos objetivos empresariais do COBIT5 para os objetivos relacionados à TI;

3. Onde se deseja alcançar? – significa que uma vez definida uma meta de melhoria, ela deve ser seguida por uma análise mais detalhada usando a orientação do COBIT5 para identificar lacunas e possíveis soluções;

4. O que precisa ser feito? – Refere-se a soluções práticas na definição de projetos apoiados em business case justificáveis;

5. Como alcançar o objetivo? – refere-se às soluções propostas que precisam de ser implementadas nas práticas quotidianas desta fase;

6. O objetivo foi alcançado? – refere-se à forma como é conduzido o funcionamento sustentável dos facilitadores novos ou melhorados;

7. Como realizar a manutenção? – Nesta fase, todo o sucesso da implementação da IoT é revisto e reforçada a necessidade de melhoria contínua.

Diferenciais do RTRMM

Apresentado o modelo, apesar de existir classes de ativos a serem gerenciados, não se foi observado como se pode ter uma visão do impacto de cada ativo para o sistema, caso este venha sofrer algum incidente. Não foi observado também qual a estratégia adotada para o cálculo do risco, entendendo que seja aplicada a estratégia do COBIT5. Por fim, não foi verificado como os riscos serão tratados a fim de prover mais segurança ao sistema IoT.

[1] https://resultadosdigitais.com.br/marketing/kpis/

[2] https://jknsarawak.moh.gov.my/hus/wp-content/uploads/2021/04/HPIA-SPECIFIC-INDICATORS-TECH-SPEC-VER-7.3.1-2021.pdf

Realizando uma comparação com o RTRMM, alguns aspectos são considerados relevantes.

1. O RTRMM trabalha de forma proativa, fundamental em um sistema de saúde. Fundamental porque possibiitaria a detecção dos riscos antes que algum paciente possa vir a ser afetado, por uma perda de integridade ou mesmo indisponibilidade de dados.

2. Ao trabalhar com módulos, em separado, no processo de gestão de riscos O RTRMM possibilita uma visão do impacto que cada dispositivo IoT poderá causar ao sistema IoT.

3. O processo de análise de riscos analisa cada risco apresentado e calcula um valor de cada um dos riscos para cada dispositivo IoT.

4. O processo de avaliação de riscos é seletivo, com base em componentes que afetam diretamente o sistema IoT.

5. As informações provenientes do processo de análise e avaliação de riscos viabiliza que a escolha de um mecanismo de segurança possa ser a mais adequada e então aplicada ao sistema, reduzindo a probabilidade de receber o mesmo tipo de ataque.

6. A forma de trabalho em tempo real do RTRMM dá ao processo de gestão de riscos a capacidade de reduzir de forma efetiva a chance de uma ameaça ser efetivada, isto é, evitar e/ou reduzir danos ao sistema IoT.

10.3.2 Gestão de Riscos em ambientes IoT via Software Proprietário

Um outro trabalho interessante que tem como objetivo gerenciar riscos em ambiente IoT é o artigo [MS19]. O artigo apresenta uma pesquisa que visa identificar e reduzir as vulnerabilidades em um sistema IoT via a utilização de um software inteligente proprietário (desenvolvido por um fornecedor) que enumera vulnerabilidades comuns em seu banco de dados e fornece a possível solução para mitigar as mesmas. O trabalho visa enfatizar vários riscos de segurança e privacidade, ataques, ameaças, vulnerabilidades, visão de IoT e pilha de protocolos, apoiando tecnologias, arquitetura e áreas de aplicações. Não foram apresentadas muitas características sobre a estrutura lógica do software, somente como o mesmo pode e foi aplicado na pesquisa realizada pelo artigo.

O software faz uso de uma base de dados baseada na lista CVE (Common Vulnerabilities and Exposures)[3], aplicado para identificar e reduzir as vulnerabilidades. O fornecedor do software identifica mais de 100 (cem) assinaturas de ataque, criando o seu próprio banco de dados com base no CVE (enumeração de vulnerabilidades comuns).

O software visa identificar uma assinatura e verificar se esta estrutura está compatível com o banco de dados. Caso exista o caminho para mitigar as vulnerabilidades no banco

[3]https://cve.mitre.org

de dados, o software atribui níveis de vulnerabilidade, de forma qualitativa: baixo, médio, alto e crítico. Com base nas informações do CVE, o software possui o detalhamento do tipo de vulnerabilidade e os detalhes da versão para decidir a relevância dos riscos de acordo com o tipo de aplicativo. Caso alguma vulnerabilidade detectada não corresponda ao item CVE, a equipe que dá suporte ao produto realiza a atualização.

Os resultados da pesquisa realizada indicam como a identificação de vulnerabilidades de segurança auxilia na priorização de decisões de negócios por meio da quantificação de vulnerabilidades. O software inteligente possibilita a mitigação de vulnerabilidades de segurança do sistema IoT, consultando o banco de dados de acordo com a identificação de vulnerabilidades e fazendo com que os desenvolvedores habilitem a quantificação e priorização de vulnerabilidades, fornecendo vários níveis a elas.

Diferenciais do RTRMM

Alguns aspectos podem ser considerados relevantes se comparar o RTRMM e a solução apresentada pelo artigo.

1. O RTRMM não trabalha diretamente com na lista CVE, isto é, não depende desta lista para mapear as ameaças e mecanismos de segurança a serem adotados. Todavia, não se deve descartar a hipótese de se adotar a lista de vulnerabilidades do CVE, por exemplo, pois a mesma já possui uma classificação quanto ao grau de criticidade, além de, na sua maioria, prover mecanismos de segurança a serem aplicados.

2. O RTRMM possui a sua estratégia própria de detectar, em tempo real, a probabilidade de existir uma ameaça ou não em tráfego IoT. Ele determina o tipo desta ameaça, classificando-a em categorias.

3. O RTRMM realiza o "link" entre a ameaça e o mecanismo de segurança a ser aplicado em tempo real. Esta funcionalidade agiliza o processo de tomada de decisão além de tornar o sistema independente de listas de vulnerabilidades. Reforçando o que já foi citado, isto não impede que o RTRMM faça uso de quaisquer listas de vulnerabilidades.

4. O RTRMM não está vinculado a um fornecedor, ou mesmo a uma linguagem de programação específica. Ele pode ser considerado como um novo modelo de referência para gestão de riscos, com funcionalidades inovadoras, e novas estratégias de gerir riscos em sistemas IoT.

10.4 Gestão de Riscos Descentralizada

Um outro modelo de gestão de riscos analisado é o proposto em [Che19], no qual o modelo está limitado a visão racionalizada dos participantes do sistema IoT. O modelo é baseado na construção, por cada usuário, de uma rede cognitiva esparsa de nós aos

quais se comunicam entre si. A partir desta rede cada usuário determina sua política de gerenciamento de segurança.

As tomadas de decisão racionais e a formação de sua rede cognitiva é interdependentes e, portanto, devem ser abordadas de uma forma holística. A proposta do modelo é estabelecida em uma estrutura "jogos-em-jogos" e propõe um conceito de solução com base no equilíbrio de Gestalt Nash (GNE) para caracterizar as decisões dos agentes, em cada rede, para assim quantificar o seu risco de percepção limitada à rede que está conectado.

Para isso, o modelo parte da premissa que os dispositivos nas redes IoT e suas interconexões podem ser modelados como nós e links, e a política de segurança de um dispositivo pode ter impacto no risco de segurança dos nós conectados a ele. Desta forma, a visão para propor o modelo é: os usuários possuem dispositivos diferentes, o gerenciamento de segurança no sistema IoT é de natureza descentralizada, proporcionando um processo de tomada de decisão descentralizado. Logo, este processo é modelado como um problema de jogo em que cada usuário aloca estrategicamente seus recursos para proteger os dispositivos.

Na visão do jogo, os riscos dos usuários são reduzidos quando seus vizinhos conectados possuem segurança de alto nível; os usuários não possuem conhecimento das políticas de segurança adotadas por todos os seus vizinhos conectados; tomando decisões de segurança somente dentro do seu universo. Desta forma, o modelo de jogo leva em consideração a racionalidade limitada dos jogadores.

Logo, o modelo de gestão proposto está baseado na decisão de segurança do usuário, como citado dentro do universo dos seus dispositivos IoT. A decisão usa o conceito de risco seguro onde o usuário percebe, observando um número de nós selecionado por ele, qual será a sua decisão de segurança, isto é, qual ou quais procedimentos/mecanismos de segurança serão aplicados.

O trabalho apresenta uma estrutura de gestão de riscos que é baseada em um vetor de cognição que representa a estrutura de observação de cada usuário de IoT. Este vetor de cognição, quando mais esparso, representa um usuário com capacidade de cognição mais fraca, pois ele possui menos condições de observar comportamentos de outros usuários no processo de decisão de sua estratégia. Isto significa que os dispositivos IoT não estão diretamente ligados ou alcançados de forma direta.

No modelo de gestão racional, os usuários precisam tomar decisões de gerenciamento de segurança, bem como projetar suas redes de cognição de maneira holística. Eles tomam essas decisões, definem um novo conceito de solução chamado equilíbrio de Gestalt Nash (GNE) para capturar a formação da rede cognitiva e o gerenciamento de segurança sob a racionalidade limitada simultaneamente. O objetivo do GNE é fornecer um método quantitativo para entender o risco de IoT massiva e fornecer políticas de gerenciamento de segurança tratáveis.

Junto ao modelo, foi projetado um algoritmo de base proximal para calcular o GNE que contém estratégia de gerenciamento de segurança e rede cognitiva dos agentes. O algoritmo descobre vários fenômenos, incluindo o surgimento do partidarismo, o preen-

chimento da desatenção e a atração dos poderosos. O resultado do algoritmo revela vários fenômenos típicos que combinam bem com as observações do mundo real.

O modelo apresentado possui uma estratégia bastante interessante, apesar de estar limitado ao universo do usuário, isto é, aos dispositivos IoT que o usuário possui acesso. Desta forma, a análise de segurança, isto é, determinar quais são os riscos existentes está limitado ao que o usuário visualiza ou tem acesso. Todavia, existe o equilíbrio neste ambiente IoT, porque pelo princípio da confiança acredita que os vizinhos também estão aplicando mecanismos de segurança de forma adequada. É um modelo com uma visão holística do sistema IoT, apesar de trabalhar de forma descentralizada, com base na decisão de cada usuário.

Diferenciais do RTRMM

Trabalhar com gerenciamento descentralizado é válido quando se tem universos independentes, e estão sob a gestão centralizada ("guarda-chuva") única de um gerente. A política de gerenciamento de riscos descentralizada proporciona que cada núcleo possua a sua própria política de gestão, o que pode causar problemas quanto determinar o que é um risco crítico ou não para cada um dos núcleos gerenciáveis.

Todavia, para que uma estratégia descentralizada de gestão de riscos possa ser efetiva, existe a necessidade que a política de gerenciamento de riscos seja única e aplicável a todos os núcleos. Desta forma, todos os critérios relacionados aos riscos serão os mesmos, deixando de levar em consideração que o vizinho é criterioso e que saberá como aplicar procedimentos e mecanismos de segurança adequados.

Sendo assim, algumas considerações comparativas com o RTRMM podem ser relatadas:

1. O RTRMM trabalha com o ambiente como um todo, isto é, ele está relacionado a um sistema IoT, onde todo o tráfego proveniente dos dispositivos IoT podem ser monitorados, analisados e avaliados.

2. O RTRMM trabalha com gerenciamento centralizado, o que facilita a especificação de parâmetros para análise e avaliação dos riscos existentes no sistema IoT.

3. O gerenciamento centralizado também facilita a tomada de decisões, independente, por exemplo, da sub-rede em que se encontra o dispositivo IoT.

4. As decisões de aplicação de procedimentos ou mesmos mecanismos de segurança independem do princípio de confiança entre usuários (gestores de riscos), mas diretamente relacionada as funcionalidades exercidas pelo RTRMM.

5. Vale a pena pensar quanto ao desempenho, mas se acredita que, apesar de centralizar toda a gestão, o processo se executará mais rápido. Esta afirmative se deve ao fato do processo de tomadas de decisão ser independe do conhecimento da estrutura alheia ou mesmo da dependência dessa estrutura.

10.4.1 Arquitetura de Gestão de Riscos com Machine Learning

Uma arquitetura analisada no desenvolvimento do RTRMM foi a descrita em [HHV17], no qual é proposta uma arquitetura de gerenciamento de risco da cadeia de suprimentos de segurança em sistemas IoT. A arquitetura tem como objetivo reduzir os riscos aplicando técnicas de aprendizado de máquina (ML – Machine Learning), monitoramento de hardware criptográfico e coordenação de sistema distribuído.

A arquitetura de gestão de riscos descrita se baseia em questões como:

- Reconhecer padrões com o objetivo de aprender e identificar o comportamento de dispositivos IoT.
- Ser capaz de responder a incidentes de segurança via um sistema de transmissão de dados.
- Trabalhar com criptografia em hardware; um operador de gestor de riscos; e uma verificação do operador em caso de eventos considerados questionáveis.

A arquitetura proposta emprega a arquitetura de redes neurais denominada Redes Adversárias Generativas (GAN)[4], pois a mesma emprega uma rede neural adicional para aproximação da função de perda. Ao contrário das redes neurais convencionais treinadas minimizando uma função de perda, a rede GAN emprega uma rede neural adicional para aproximação da função de perda. Na área de segurança da informação, a abordagem emprega uma arquitetura GAN para detectar invasões em redes de comunicação por usuários não autorizados. O objetivo é treinar inicialmente a rede neural para detectar intrusões de maneira supervisionada com base em dados rotulados e, com o tempo, a rede aprender a detectar nova invasão.

A arquitetura da rede GAN utilizada no modelo discutido, consiste em uma sub-rede discriminativa com três camadas ocultas de 20, 10 e 5 máquinas, e uma camada de saída de 1 máquina. A sub-rede discriminativa classifica os pacotes de dados em uma classe normal ou de ataque. A rede generativa possui uma camada de entrada de 200 máquinas, três camadas ocultas de 150, 100 e 60 unidades e uma camada de saída de 41 unidades. Uma taxa de abandono de 0,5 foi definida para as camadas ocultas para evitar overfitting. Os dados de entrada foram escalados entre 0 e 1. As redes fora treinadas com o método de otimização Adam, com taxa de aprendizado de 0,01 para a sub-rede discriminativa e taxa de aprendizado de 0,001 para a sub-rede generativa e para a rede GAN. A informação do gradiente para o treinamento dos modelos generativos e discriminativos foi calculada com retro-propagação.

A abordagem de detecção de anomalias não está ainda disponível, mas existe uma proposta de uma estratégia de correlação cruzada como parte da análise de segurança da rede. A estratégia utiliza uma máquina dos estados de saída esta atrelada a um dispositivo, correlacionada com a aprovação ou rejeição do operador. A saída do dispositivo do sistema não é transmitida entre os componentes conectados, e os estados anormais

[4]https://didatica.tech/introducao-a-gans-redes-adversarias-generativas/

percebidos possam ser monitorados e fundamentados pelo operador. O operador tem o controle final sobre a avaliação da anomalia.

Diferenciais do RTRMM

A abordagem apresentada por este artigo é muito interessante, trabalhando com uma técnica de aprendizagem muito apurada. A taxa de erro pode ser considerada mínima, segundo o artigo. Todavia, o RTRMM apresenta alguns diferenciais interessantes:

1. Ao contrário da arquitetura proposta, que utiliza Machine Learning, o RTRMM faz uso de lógica difusa (Fuzzy Logic). Com ML o aprendizado é realizado via um conjunto de dados rotulado que lhe permite aprender como um ser humano realiza uma tarefa. Mesmo fazendo usos de uma arquitetura de redes neurais GAN, não garante a plenitude quanto aos acertos.

2. A arquitetura de lógica difusa, usada no RTRMM, tem o seu início do aprendizado baseado em inferências, as quais tornam o processo inicialmente impreciso. Todavia, o aprendizado vai se solidificando toda vez que o sistema é realimentado com novos valores de probabilidade, possibilitando uma maior proximidade da realidade. Desta forma se pode dizer que o sistema é mais confiável, pois apresenta resultados mais próximos ao mundo real.

3. Não foi observado nesta arquitetura nenhuma funcionalidade que realize uma análise e avaliação dos riscos detectados e nem propor mecanismos de segurança para reduzir as chances de incidentes ocorrerem.

4. A proposta é uma arquitetura que visa usar ferramentas de reconhecimento para aprender e identificar o comportamento do dispositivo IoT sob falhas de hardware/software e ataques cibernéticos; ser capaz de responder a possíveis incidentes de segurança, via um sistema de dados; fazer uso de criptografia; e fazer uso de um operador nas tomadas de decisão (ainda não está plenamente definido como serão realizadas tais tarefas).

5. O RTRMM de forma global, executa as tarefas da arquitetura proposta e ainda oferece uma forma de gerir os riscos em sistemas IoT, em tempo real, propondo soluções de segurança com o objetivo de reduzir os riscos existentes.

6. Apesar que no RTRMM o processo desde a detecção e análise de uma ameaça, até a proposição de aplicar um mecanismo de segurança ser automatizada, ainda, por questões de validação, o uso do gestor de segurança para validar e implantar o mecanismo de segurança ainda é aplicado em algumas situações.

10.5 Considerações

Todos os modelos estudados e analisados possuem as suas características próprias para atender demandas específicas. São excelentes modelos que possuem funcionalidades

específicas em conformidade com os problemas apresentados. Porém, nenhum deles apresenta uma proposta que trabalhe com gestão de riscos em sistemas IoT de forma proativa, como o RTRMM.

A nossa proposta, trabalhar de forma proativa, é um fator que visa melhorar a forma de gerenciar sistemas IoT, apresentando funcionalidades proativas, possibilitando que incidentes possam ser minimizados ou mesmo até evitados antes de causarem danos ao sistema IoT. A arquitetura lógica do RTRMM é baseada em uma arquitetura de gestão de riscos, especificada por uma norma internacional (ISO 27005). Esta arquitetura possibilita uma gestão de riscos centralizada, que facilita o monitoramento e o controle dos dispositivos IoT, bem como o sistema IoT como um todo.

O RTRMM provê a detecção de ameaças em tempo real, no qual essa funcionalidade é executada por uma das arquiteturas analisadas. Todavia, a utilização da arquitetura de lógica difusa em conjunto a teoria de probabilidades tornou o RTRMM um sistema mais simples de ser implementado, e seu amadurecimento vai ao encontro ao aprendizado que a arquitetura proporciona. Esta característica possibilita que o RTRMM se aproxime da realidade, minimizando a possibilidade de resultados falsos.

A estratégia lógica de análise e avaliação de riscos aplicada no RTRMM segue a lógica da ISO 27005, mas a estratégia de implementação apresenta uma nova visão, baseada em cadeia de Markov e teoria dos jogos. O uso da cadeia de Markov apresenta uma evolução dos resultados no decorrer do tempo, com estados discretos e a propriedade de que a distribuição de probabilidade do próximo estado depende apenas do estado atual. O uso de teoria de jogos é uma forma de abordar a avaliação de riscos, porque se consegue levar em consideração ao menos 2 (dois) fatores relevantes para a seleção de riscos que serão ou não tratados.

Por fim, o RTRMM ainda apresenta mais 2 (duas) funcionalidades que os demais modelos. A categorização dos riscos e a escolha em tempo real dos mecanismos a serem aplicados. A categorização agiliza o processo de escolha, pois possibilita lincar a categoria do risco com o mecanismo a ser aplicado. Logo, o RTRMM é um modelo que apresenta além de novas funcionalidades, e atuara em tempo real, ele apresenta uma nova visão de gestão de riscos em IoT, podendo ser aplicado em qualquer tipo de ambiente IoT.

11

Conclusão

Este trabalho teve como objetivo apresentar uma nova proposta de gerenciar riscos em sistemas IoT, de forma proativa, em tempo real. Para alcançar tal objetivo foram estudados e analisados metodologias e modelos de gestão de riscos desenvolvidas e implementadas. As metodologias modelos de gestão riscos foram coletadas via normas internacionais e por artigos escritos por autores da área de pesquisa.

A partir deste estudo decidiu-se trabalhar com gestão de riscos de segurança em tempo real, possibilitando um maior dinamismo na análise e avaliação dos riscos em sistemas IoT. Com isso, foi criado e especificado um modelo de gestão de riscos capaz de conhecer as ameaças e a probabilidade delas acontecerem, analisar e avaliar os riscos, e tratá-los em tempo real. Para isso foi necessário criar uma arquitetura para detectar e analisar o tráfego IoT, com o objetivo de verificar se o mesmo possuía alguma ameaça. Para implementar esta arquitetura, foi estudada e analisadas tecnologias de programação lógica com o intuito de buscar a melhor solução.

A arquitetura de análise e avaliação de riscos baseou-se um uma tecnologia que possibilitava determinar qual a melhor forma de calcular o valor do risco e no processo de

avaliação se o risco será ou não tratado. Por fim, criou-se uma estrutura para classificar o risco com o objetivo de dar maior rapidez ao processo de seleção do mecanismo de segurança a ser aplicado, bem como determinar qual seria a melhor opção para assim ser aplicado e reduzir a probabilidade deste ocorrer.

Apesar do modelo prever várias funcionalidades, somente as funções dos Threat Analyzer e de Risk Management foram implementados, mostrando como o RTRMM poderá executar as suas funcionalidades de forma eficiente, atendendo aos seus objetivos. O módulo DB Controls, a estratégia de aprendizado no Threat Analyzer e a escolha da medida de segurança a ser aplicada não foram implementados. Atualmente o RTRMM visa somente desconectar o link com o dispositivo IoT como aplicação de mecanismo de segurança. A proposta futura é prover um módulo que selecione e aplique automaticamente, ou apresente ao administrador do sistema o mecanismo de segurança que mais se adéque ao problema.

Nem todos os módulos e funcionalidades foram completamente implementadas e testados, como o módulo DB Controls, a estratégia de aprendizado no Threat Analyzer e a escolha da medida de segurança a ser aplicada. Atualmente, o aprendizado está em fase de testes quanto a sua eficiência, e a aplicação de medidas de segurança, é utilizada a interrupção da comunicação com o dispositivo IoT que está sofrendo a ameaça/ataque. Contudo, todos estes elementos estão sendo trabalhados para se juntar às funcionalidades do RTRMM.

Por fim, o trabalho deu origem a 3 (três) artigos, publicados, em congressos distintos, em áreas de interesses distintas, provendo assim a sua viabilidade de ser implantado a fim possibilitar uma gerência efetiva de riscos em sistemas IoT. São estes os congressos:

1. Combining IoT Risk Management with Logic Programming - 2023 18th Iberian Conference on Information Systems and Technologies (CISTI) [LAP23a]

2. Um Modelo Declarativo para Gestão de Riscos em IoT - XXIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSEG2023) [LAP23b]

3. A Logic-Based Model to Reduce IoT Security Risks - 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) [LPA24]

Um dos grandes objetivos deste projeto, é, em um futuro próximo, aplicar o RTRMM em ambientes IoT Healthcare. A razão que leva a esta motivação se deve ao fato da necessidade de um monitoramento constante, além da garantia da confidencilidade e integridade dos dados que são trocados, manipulados e processados entre, por exemplo, dispositivos IoT (aplicados em pacientes) e sistemas de tratamento ou mesmo de monitoramento e aplicação de medicamentos remotos. O RTRMM pode ser um sistema, que possa evitar que danos, intencionais ou não, possam ser causados a pacientes remotos.

11.1 Trabalhos Futuros

A continuidade de desenvolvimento das funcionalidades do RTRMM deve ser mantida, principalmente a adequação / prototipação dessa funcionalidades do modelo ao ambiente IoT que for aplicado. Por esta visão, é que o RTRMM pode ser encarado como um modelo de referência para qualquer sistema IoT, pois o seu conjunto de funcionalidades oferece um leque de opções para reduzir, em tempo real, os riscos existentes em sistemas IoT. Para isso, implementar um protótipo que realmente reflita todas funcionalidades do RTRMM, fazendo uso de uma linguagem de programação difusa.

Por se relacionar ao tema pesquisado, de importância estratégica na área de segurança da informação, e haver um deficit de investigações, realizar uma pesquisa que relacione o RTRMM à sistemas de tratamento de incidentes em sistemas IoT, em tempo real. Atualmente, todo o processo existente é realizado de forma reativa, dificultando ações que venham em tempo real minimizar os danos que possam vir se causados aos sistemas IoT.

O modelo desenvolvido já oferece um conjunto de funcionalidades para a análise e avaliação de riscos em tempo real, bem como também para prover mecanismos de segurança. Realizar o "link"entre o RTRMM e um sistema de tratamento de incidentes de segurança da informação irá agilizar o processo de trabalho em um Centro de Prevenção e Tratamento de Incidentes e Respostas.

11.2 Limitações / Dificuldades

Apesar de gestão de segurança da informação ser um tema ser bastante desenvolvido, trabalhado e divulgado na comunidade de segurança da informação, aspectos relacionados a gestão de riscos em sistema IoT ainda carecem ainda de muita atenção.

A maioria dos modelos existentes e as metodologias estudadas não preveem soluções de gestão de riscos de segurança que trabalhem com a estratégia de tempo real. A não utilização da estratégia em tempo real, na maioria dos estudos realizados, dificultaram bastante uma melhor visão ou utilização desta técnica. Todavia, a utilização de gestão de riscos em tempo real em outras áreas, serviram como referências para uma melhor visualização e análise da aplicação desta estratégia.

Porém, uma melhor análise comprativa do modelo desenvolvido com outros modelos foi, em parte prejudicada, porque faltou modelos, com estratégias semelhantes, na área de gestão de segurança, dificultando uma análise mais precisa do trabalho. Por conseguinte, todos os modelos e metodologias estudados e analisados serviram como uma referência da necessidade de buscar soluções que agilizem o processo de gestão de riscos.

Bibliografia

[ABN09] ABNT. Abnt nbr iso 31000 - gestão de riscos — princípios e diretrizes. In *Normas ABNT*. ABNT, 2009.

[ACRZ16] M. Alberti, G. Cota, F. Riguzzi, and R. Zese. Probabilistic logical inference on the web. In *Proceedings of XVth International Conference of the Italian Association for Artificial Intelligence*, pages 351–363. Springer, 2016.

[Ahm17] H. Ahmed. Cobit 5 for risk—a powerful tool for risk management. In *COBIT*. ISACA, 2017.

[Aiy20] et.al. Aiyanyo, I. A systematic review of defensive and offensive cybersecurity with machine learning. In *Applied Sciences Jorunal*, pages 1–26. MDPI, 2020.

[AJ08] M. Andrade and M. A. Jacques. Estudo comparativo de controladores de mamdani e sugeno para controle de tráfego em interseções isoladas. In *Transportes XVI*, pages 24–31. Transportes, 2008.

[Alb01] et al Alberts, C. Octavesm method implementation guide version 2.0 - volume 1 introduction. In *Networked Systems Survivability Program, Carnegie Mellon*. Software Engineering Institute, 2001.

[Ans88] A. Q. Ansari. The basics of fuzzy logic: A tutorial review. In *Computereducation*, pages 1–9. Computer Education Group, 1988.

[ARC18] M. Ammara, G. Russellob, and B. Crispo. Internet of things: A survey on the security of iot frameworks. In *Journal of Information Security and Applications*, pages 8–27. Elsevier, 2018.

[Beh00] E. Behrends. Introduction to markov chains with special emphasis on rapid mixing. In *Advanced Lectures in Mathematics*. Spinger, 2000.

[Bis18] M. Bishop. *Computer Security Art and Sciense*. Addison Wesley, US, 2nd edition, 2018.

[Bla12] et al. Blank, R. Nist 800-30: Guide for conducting risk assessments. In *NIST*. U.S. Department of Commerce, 2012.

[BP02] S. Burnett and S. Paine. *Criprograia e Segurança - O Guia Oficial RSA*. Campus, Brasil, 1st edition, 2002.

[Cam15] et.al Campos, C. A teoria dos jogos e a mente brilhante de john nash. In *Revista de Filosofía y Ciencias, vol. 10*, pages 1–6. Filosofía y Ciencias, 2015.

[Cha91] E. Charniak. Bayesians networks without tears. In *AI Magazine Volume 12 Number 4*, pages 50–63. AI Magazine, 1991.

[Che19] J. Chen. Interdependent strategic security risk management with bounded rationality in the internet of things. In *IEEE Transactions on Information Forensics and Security, vol. 14*, pages 2958–2971. IEEE, 2019.

[CJ01] S. D. Connell and A. K. Jain. Template-based online character recognition. In *Pattern Recognition, Volume 34*, pages 1–14. Elsevier, 2001.

[Cox09] L.A. Cox. Game theory and risk analysis. In *Risk Analysis, Vol.29*, pages 1062–1068. Society for Risk Analysis, 2009.

[CR93] A. Colmerauer and P. Roussel. The birth of prolog. In *ACM SIGPLAN, Volume 28*, pages 37–52. ACM, 1993.

[DKM+15] A. Dries, A. Kimmig, W. Meert, J. Renkens, G. V. Broeck, J. Vlasselaer, and L. Raedt. Problog2: Probabilistic logic programming. In *ECML PKDD 2015*, pages 312–315. Springer Link, 2015.

[DM08] L.A. Dantas and J. P. Moura. Fundamentos matemáticos aplicados ao estudo de práticas experimentais de controle de processos. In *8ª ERMAC*. SBMAC, 2008.

[DNK97] Y. Dimopoulos, B.; Nebel, and J. Koehler. Encoding planning problems in non-monotonic logic programs. In *Proceedings of European Conference on Planning*, pages 169–181. Springer-Verlag, 1997.

[D 99] B. D'Ambrosio. Inference in bayesian networks. In *AI Magazine Volume 20 Number 2*, pages 21–36. AI Magazine, 1999.

[Ebr01] Rafee Ebrahim. Fuzzy logic programming. In *Fuzzy Sets and Systems, vol.117, number2*, pages 215–230, 2001.

[EIK09] T. Eiter, G. Ianni, and T. Krennwallner. Answer set programming: A primer. In *Reasoning Web. Semantic Technologies for Information*, pages 40–110. ResearchGate, 2009.

[Fil16] M.F. Filho. Risk model based on social network analysis. In *ISIE 2016*, pages 22–27. IEEE, 2016.

[Fra06] M. Frade. Lógica computacional - prolog. 2006.

[GL88] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proceedings of International Logic Programming Conference and Symposium*, pages 1070–1080. MIT Press, 1988.

[Glu08] J. C. Gluz. Introdução às lógicas probabilísticas. In *Unisinos*. Unisinos, 2008.

[GT80] M. M. Gupta and Y.. Tsukamoto. Fuzzy logic controllers??a perspective. In *Joint Automatic Control Conference, Vol.17*. InfoNa, 1980.

[Hal90] J. Halpern. An analysis of first-order logics of probability. In *Artificial Intelligence vol 46*, pages 311–350. Elsevier, 1990.

[Has19] et al. Hassija, V. A survey on iot security: application areas, security threats, and solution architectures. In *IEEE Access 7*, pages 82721–82743. IEEE, 2019.

[Hau02] K. Hausken. Probabilistic risk analysis and game theory. In *Risk Analysis, Vol.22*, pages 17–27. Society for Risk Analysis, 2002.

[Her02] H. Hermanns. Interactive markov chains and the quest quantified quality. In *International Journal of Approximate Reasoning*. Springer, 2002.

[HHV17] R. Hiromoto, M. Haney, and A. Vakanski. A secure architecture for iot with supply chain risk management. In *9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pages 23–27. IEEE, 2017.

[Hit16] G. Hitchcock. Remarkable similarities: A dialogue between boole and de morgan. In *Research in History and Philosophy of Mathematics*, pages 69–82. Springer Link, 2016.

[HMW+16] H. He, C. Maple, T. Watson, A. Tiwari, J. Mehnen, Y. Jin, and B. Gabrys. The security challenges in the iot enabled cyber-physical systems and opportunities for evolutionary computing and other computational intelligence. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1015–1021. IEEE, 2016.

[HS14] S. Holldobler and L. Schweizer. Answer set programming and clasp - a tutorial. In *YSIP, International Center for Computational Logic*, pages 77–95. Technische Universit¨at Dresden, 2014.

[Ian12] I. Iancu. A mamdani type fuzzy logic controller. In *Fuzzy Logic – Controls, Concepts, Theories and Applications*, pages 325–349. Intechopen, 2012.

[IC09] I. Iancu and M. Colhon. Mamdani flc with various implications. In *11th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 368–375. IEEE, 2009.

[ISO04] ISO/IEC. Enterprise risk management — integrated framework. In *Committee of Sponsoring Organizations of the Treadway Commission*. Committee of Sponsoring Organizations of the Treadway Commission, 2004.

[ISO17] ISO/IEC. Iso/iec 27002:2017 - information security, cybersecurity and privacy protection — information security controls. In *ISO/IEC*. ISO/IEC, 2017.

[ISO22a] ISO/IEC. Iso/iec 27001:2022 information security management systems. In *ISO/IEC*. ISO/IEC, 2022.

[ISO22b] ISO/IEC. Iso/iec 27005:2022 information security, cybersecurity and privacy protection — guidance on managing information security risks. In *ISO/IEC*. ISO/IEC, 2022.

[Jan07] J. Jantzen. Tutorial on fuzzy logic. In *Tech. report no 98-E 868*. Technical University of Denmark, Oersted-DTU, 2007.

[JIe20] P. Julián-Iranzo and et.al. The fuzzy logic programming language fasill: Design and implementation. pages 139–168, 2020.

[Joh22] D. Johnson. Fuzzy logic tutorial: What is, architecture, application, example. 2022.

[Kal06] M. Kaliprasad. Proactive risk management. In *Cost Engineering*, pages 26–35. ProQuest, 2006.

[KJ14] J. Kwon and M. E. Johnson. Proactive versus reactive security investments in the healthcare sector. In *MIS Quarterly Vol. 38*, pages 451–471. JSTOR, 2014.

[Kos92] B. Kosko. *Neural Networks and Fuzzy Systems*. Prentice-Hall., New Jersey, US, 1992.

[Kow74] R. Kowalski. Predicate logic as programming language. In *Information Processing 74*, pages 569–574. North-Holland Publishing Company, 1974.

[Lab20] Stratosphere Laboratory. Aposemat iot-23. 2020.

[LAP23a] L. O. Lento, S. Abreu, and P. Patinho. Combining iot risk management with logic programming. In *2023 18th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–5. IEEE Xplore, 2023.

[LAP23b] L. O. Lento, S. Abreu, and P. Patinho. Um modelo declarativo para gestão de riscos em iot. In *2023 - Anais do XXIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 570–575. SBC - Sociedade Brasileira de Computação, 2023.

[Lar80] P.M. Larsen. Industrial applications of fuzzy logic control. In *International Journal of Man-Machine Studies*, pages 3–10. Elsevier, 1980.

[Le88] Q. Liu and et.al. A survey on security threats and defensive techniques of machine learning: A data driven view. In *IEEE Access, vol.6*, pages 12103–12117. IEEE, 1988.

[Len12] L. O. Lento. *Gestão de Segurança*. UNISUL, Brasil, 1st edition, 2012.

[Len15] L. O. Lento. *Governança e Gestão da Segurança de Informação*. UNISUL, Brasil, 1st edition, 2015.

[Len18] L. O. Lento. *Segurança da Informação*. UNISUL, Brasil, 1st edition, 2018.

[Lif08] L. Lifschitz. What is answer set programming? In *UT Computer Science*. University of Texas, 2008.

[Llo74] E. H. Lloyd. What is, and what is not, a markov chain. In *Journal of H*, pages 1–28. Elsevier, 1974.

[Lo 17] M. Loève. *Probability Theory*. Dover Publications Inc., US, 2017.

[LPA24] L. O. Lento, P. Patinho, and S. Abreu. A logic-based model to reduce iot security risks. In *Proceedings of ICAART 2024, ISBN: 978-989-758-680-4*, pages 1–8. Science and Technology Publications, Lda., 2024.

[Lu17] Y. Lu. Industry 4.0:a survey on technologies, applications and open research issues. In *Journal of Industrial Information Integration*, pages 1–10. Elsevier, 2017.

[MA75] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. In *Int. J. Man-machine Studies*, pages 1–13. Elsevier, 1975.

[MLS13] B. Marcello, Y. Lierler, and P. Schüller. Prolog and asp inference under one roof. In *Lecture Notes in Computer Science, vol 8148*, pages 148–160. Springer, 2013.

[Moo85] R. Moore. Semantical considerations on nonmonotonic logic. In *Artificial Intelligence 25*, pages 75–94. Elsevier, 1985.

[MS19] V. Malik and S. Singh. Security risk management in iot environment. In *Journal of Discrete Mathematical Sciences and Cryptography vol. 22, no 4*, pages 697–709. Taru Publications, 2019.

[Mye99] R.B. Myerson. Nash equilibrium and the history of economic theory. In *Journal of Economic Literature*, pages 1067–1082. American Economic Association, 1999.

[NBBW06] C. J. Needham, J. R. Bradford, A. J. Bulpitt, and D. R. Westhead. Inference in bayesian networks. In *Nature Biotechnology Volume 24 Number 1*, pages 51–53. Springer Nature, 2006.

[Osm14] E. Osmanoglu. Identity and access management - business performance through connected intelligence. In *Syngress*. Elsevier, 2014.

[Pal97] L. Palazzo. *Introdução à Programação PROLOG*. Editora da Universidade Católica de Pelotas, Pelotas - Brasil, 1st edition, 1997.

[Pe19] B. Pourghebleh and al et. A comprehensive study on the trust management techniques in the internet of things. In *IEEE Internet of Things Journal, vol. 6*, pages 9326–9337. IEEE, 2019.

[Per08] J. Peregrin. What is the logic of inference? In *Stud Logica88*, pages 263–294. Spinger, 2008.

[Pim21] B. Pim. Uma breve introdução à teoria dos jogos. In *Revista Eletrônica Paulista de Matemática, vol. 1*, pages 69–80. C.Q.D, 2021.

[QFG12] Y. Qian, Y. Fang, and J. Gonzalez. Managing information security risks during new technology adoption. In *Computers and Security*, pages 859–869. Elsevier, 2012.

[Rae07] L. D. Raedt. A probabilistic prolog and its application in link discovery. In *IJCAI-07*, pages 2468–2472. IJCAI, 2007.

[Rai22] G. Raimondo. Nist ai 100-1: Artificial intelligence risk management framework (ai rmf 1.0). In *NIST*. U.S. Department of Commerce, 2022.

[REC15] K. Rose, S. Eldridge, and L. Chapin. The internet of things: an overview - understanding the issues and challenges of a more connected world. In *ISOC 2015*. The Internet Society, 2015.

[Rig17] F. Riguzzi. Causal inference incplint. In *International Journal of Approximate Reasoning*. Elsevier, 2017.

[Rig19] F. Riguzzi. *Foundations of Probabilistic Logic Programming: Languages, Semantics, Inference and Learning*. River Publishers, US, 1st edition, 2019.

[RN95] S. Russel and P. Norvig. *Artificial Intelligence A Modern Approach*. Prentice Hall, New Jersey - US, 1st edition, 1995.

[RPIKR18] S. Rizvi, J. Pfeffer III, A. Kurtz, and M. Rizvi. Securing the internet of things (iot): A security taxonomy for iot. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 163–168. IEEE, 2018.

[RRM+19] P. Radanliev, D. Roure, C. Maple, J. Nurse, R. Nicolescu, and U. Ani. Cyber risk in iot systems. In *UK EPSRC*. UK EPSRC, 2019.

[RS92] N. Raymond and V.S. Subrahmanian. Probabilistic logic programming. In *Information and Computation 101*, pages 150–201. Elsevier, 1992.

[RS18] F. Riguzzi and T. Swift. A survey of probabilistic logic programming. In *Declarative Logic Programming: Theory, Systems, and Applications*, pages 185–228. ACM DL, 2018.

[Ré07] A. Rényi. *Probability Theory*. Dover Publications Inc., US, 2007.

[S.20] Puig.; S. Prolog and answer set programming: Languages in logic programming. 2020.

[SAH+21] K. Shaukat, T. Alam, I. Hameed, W Khan, and N. Abbas. A review on security challenges in internet of things (iot). In *Proceedings of the 26 th International Conference on Automation and Computing*. IEEE, 2021.

[SAI10] Deris Stiawan, A. H. Abdullah, and M. Y. Idris. The trends of intrusion prevention system network'. In *2nd international Conference on Education Technology and Computer (ICETC)*, pages 217–221. IEEE Xplore, 2010.

[San07] B. Sanjaa. Fuzzy and probability. In *IEEE Xplore*, pages 141–143. IEEE, 2007.

[Sat95] T. Sato. A statistical learning method for logic programs with distribution semantics. In *12th International Conference on Logic Programming (ICLP 1995)*, pages 715–729. MIT Press., 1995.

[Sau05] S. Saunders. What is probability? In *Quo Vadis Quantum Mechanics*, pages 209–238. Spinger, 2005.

[SBH+19] F. Salih, N. Bakar, H. Hassan, F. Yahya, N. Kama, and J. Shah. Iot security risk management model for healthcare industry. In *Malaysian Journal of Computer Science*, pages 131–144. Malaysian Journal of Computer Science, 2019.

[Sen06] E. Seneta. Markov and the creation of markov chains. In *Markov Anniversary Meeting*, pages 1–25. University of Sydney, 2006.

[SGB+04] B. Sartini, G. Garbugio, H. Bortolossi, P. Santos, and L. Barreto. Uma introdução a teoria dos jogos. In *II Bienal da SBM*, pages 1–62. UFB, 2004.

[SI18] Izquierdo S. and Luis R. Izquierdo. Mamdani fuzzy systems for modelling and simulation: A critical assessment. In *Journal of Artificial Societies and Social Simulation*. JASSS, 2018.

[Spi08] M. Spivey. *An Introduction of Logic Programming through Prolog*. Prentice-Hall., New Jersey, US, 2008.

[SQO19] Sana S. Sabry, Noor A. Qarabash, and Hadeel S. Obaid. The road to the internet of things: a survey. In *9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON)*, pages 290–296. IEEE, 2019.

[SRJ18] Kazi M. Sadique, R. Rahmani, and P Johannesson. Towards security on internet of things: Applications and challenges in technology. In *Procedia Computer Science Volume 141*, pages 199–206. Elsevier, 2018.

[Sto02] et al. Stoneburner, G. Risk management guide for information technology systems. In *NIST Special Publication 800-30*. NIST, 2002.

[Tan80] R Tanscheit. Sistemas fuzzy. 1980.

[VA16] S. Vaidyanathan and A. T Azar. Takagi-sugeno fuzzy logic controller for liu-chen four-scroll chaotic system. In *International Journal of Intelligent Engineering Informatics, Vol. 4*, pages 135–150. InderScience, 2016.

[Wai02] M. Wainwright. Stochastic processes on graphs with cycles: geometric and variational approaches. pages 1–271, 2002.

[Wan92] L. X. Wang. Fuzzy systems are universal approximators. In *Proc. of IEEE Inter. Conf. on Fuzzy Systems*, pages 1163–1170. IEEE, 1992.

[Won94] S. Wong. Construction of a markov network from data for probabilistic inference. In *University of Regina, Canada*. University of Regina, 1994.

[Xe19] Y. Xu and et.al. Internet of things (iot) cybersecurity research:a review of current research topics. In *IEEE Internet of Things Journal*, pages 2013–2115. IEEE, 2019.

[Ye14] Z. Yan and al et. A survey on trust management for internet of things. In *Journal of Network and Computer Applications*, pages 120–134. Elsevier, 2014.

[Zad65] L.A. Zadeh. Fuzzy sets. In *Information and Control 8*, pages 338–353. Elsevier, 1965.

[Zad78] L.A. Zadeh. Fuzzy sets as a basis for a theory of possibility. In *Fuzzy Sets and Systems, Vol 1*, pages 3–28. Elsevier, 1978.

[Zad88] L.A. Zadeh. Fuzzy logic. In *IEEE Computer, vol. 21*. IEEE, 1988.

[ZAHS19] N. Zakaria, H.and Bakar, Azaliah A., Noor H. Hassan, and Yaacob S. Iot security risk management model for secured practice in healthcare environment. In *The Fifth Information Systems International Conference*, pages 1241–1248. Elsevier, 2019.

[ZBSA12] A. Zelenkauskaite, N. Bessis, S. Sotiriadis, and E Asimakopoulou. Interconnectedness of complex systems of internet of things through social network analysis for disaster management. In *IEEE Computer Society, Fourth International Conference on Intelligent Networking and Collaborative Systems*, pages 503–5083. IEEE, 2012.

[Zha14] et al Zhang, Z. Iot security: ongoing challenges and research opportunities. In *IEEEXplore*. IEEE, 2014.

UNIVERSIDADE DE ÉVORA
INSTITUTO DE INVESTIGAÇÃO
E FORMAÇÃO AVANÇADA

Contactos:
Universidade de Évora
Instituto de Investigação e Formação Avançada — IIFA
Palácio do Vimioso | Largo Marquês de Marialva, Apart. 94
7002 - 554 Évora | Portugal
Tel: (+351) 266 706 581
Fax: (+351) 266 744 677
email: iifa@uevora.pt