UNIVERSIDADE FEDERAL DE JUIZ DE FORA
FACULDADE DE ENGENHARIA & INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS GRADUAÇÃO EM MODELAGEM COMPUTACIONAL

Eduardo Santos de Oliveira Marques

Integração de Padrões de Ataque, Fraquezas e Modelagem de Ameaças para Suporte à Análise de Segurança no Domínio Automotivo

Juiz de Fora

2025

Eduardo Santos de Oliveira Marques

Integração de Padrões de Ataque, Fraquezas e Modelagem de Ameaças para Suporte à Análise de Segurança no Domínio Automotivo

Dissertação apresentada ao Programa de Pós Graduação em Modelagem Computacional da Faculdade de Engenharia & Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Modelagem Computacional Área de concentração: Modelagem Computacional

Orientador: Prof. Dr. Bernardo Martins Rocha

Coorientador: Prof. Dr. André Luiz de Oliveira

Juiz de Fora

2025

Ficha catalográfica elaborada através do programa de geração
automática da Biblioteca Universitária da UFJF,
com os dados fornecidos pelo(a) autor(a)

Santos de Oliveira Marques, Eduardo.
Integração de Padrões de Ataque, Fraquezas e Modelagem de Ameaças para Suporte à Análise de Segurança no Domínio Automotivo / Eduardo Santos de Oliveira Marques. -- 2025.
120 f. : il.

Orientador: Bernardo Martins Rocha
Coorientador: André Luiz de Oliveira
Dissertação (mestrado acadêmico) - Universidade Federal de Juiz de Fora, ICE/Engenharia. Programa de Pós-Graduação em Modelagem Computacional, 2025.

1. Segurança cibernética. 2. Modelo de ameaças. 3. Bases de segurança. 4. ISO/SAE 21434. 5. Sistemas automotivos autônomos. I. Martins Rocha, Bernardo, orient. II. Luiz de Oliveira, André, coorient. III. Título.

Eduardo Santos de Oliveira Marques

Integração de padrões de ataque, fraquezas e modelagem de ameaças para suporte à análise de segurança no domínio automotivo

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem Computacional da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Modelagem Computacional. Área de concentração: Modelagem Computacional.

Aprovada em 04 de julho de 2025.

BANCA EXAMINADORA

Prof. Dr. Bernardo Martins Rocha - Orientador
Universidade Federal de Juiz de Fora

Prof. Dr. André Luiz de Oliveira - Coorientador
Universidade Federal de Juiz de Fora

Prof. Dr. Alex Borges Vieira
Universidade Federal de Juiz de Fora

Prof.ª Dr.ª Kalinka Regina Lucas Jaquie Castelo Branco
Universidade de São Paulo

Juiz de Fora, 01/07/2025.

Documento assinado eletronicamente por **Bernardo Martins Rocha**, **Professor(a)**, em 04/07/2025, às 10:17, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Alex Borges Vieira**, **Professor(a)**, em 04/07/2025, às 10:18, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Kalinka Regina Lucas Jaquie Castelo Branco**, **Usuário Externo**, em 04/07/2025, às 17:32, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no Portal do SEI-Ufjf (www2.ufjf.br/SEI) através do ícone Conferência de Documentos, informando o código verificador **2477242** e o código CRC **04F6732D**.

Dedico este trabalho à minha família, que sempre apoiou e acreditou em mim; principalmente à minha mãe e avó, que, mesmo longe, nunca deixaram de cuidar de mim.

AGRADECIMENTOS

Agradeço a todas as pessoas que me apoiaram ao longo desta jornada acadêmica; professores, amigos e colegas que sempre estiveram dispostos a ajudar, e especialmente à minha família, por seu apoio incondicional em todos os momentos. Cada um de vocês foi um pilar fundamental para que eu pudesse continuar seguindo em frente.

Sou profundamente grato aos meus orientadores por acolherem meu sonho de trabalhar com segurança cibernética. Ao professor Bernardo, meu sincero agradecimento por ter me acompanhado com zelo ao longo desta pesquisa, oferecendo apoio e suporte sempre que precisei. Ao professor André, agradeço pela constante disponibilidade e dedicação ao projeto, sempre buscando o aprimoramento deste trabalho com atenção e compromisso. A ambos, meu muito obrigado por caminharem comigo nessa jornada, não apenas como orientadores, mas como verdadeiros mentores.

“Ninguém que é curioso é idiota. As pessoas que não fazem perguntas permanecem ignorantes para o resto de suas vidas.”
(Neil deGrasse Tyson).

RESUMO

A crescente interconectividade entre sistemas computacionais, dispositivos embarcados e redes de comunicação tem intensificado a preocupação com a segurança cibernética, especialmente em sistemas críticos, como os veículos autônomos. Nesse contexto, a norma ISO/SAE 21434 foi desenvolvida com o propósito de estabelecer diretrizes para a análise de ameaças e avaliação de riscos cibernéticos (*Threat Analysis and Risk Assessment* - TARA) ao longo do ciclo de vida dos sistemas automotivos. Para apoiar esse processo, técnicas como o modelo STRIDE (*Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege*) e a análise de árvores de ataque têm sido amplamente utilizadas para apoiar o modelo da ISO. De forma complementar, bases estruturadas como o CAPEC (*Common Attack Pattern Enumeration and Classification*) e CWE (*Common Weakness Enumeration*) oferecem catálogos robustos de padrões de ataque e fraquezas, com potencial para enriquecer as atividades de análise de ameaças e definição de requisitos de segurança. No entanto, a ausência de uma integração sistemática entre essas bases e os modelos normativos representa uma lacuna significativa na literatura, limitando sua aplicação em contextos práticos. Diante deste cenário, o trabalho apresentado propõe uma abordagem metodológica para apoiar a análise de riscos em sistemas automotivos autônomos, em conformidade com a norma ISO/SAE 21434. A proposta visa integrar de forma estruturada o modelo de ameaças STRIDE, o catálogo de ataques CAPEC e a base de fraquezas CWE, com o objetivo de apoiar as etapas de identificação, avaliação e tratamento de risco no processo TARA. Para isso, foi desenvolvido um modelo de dados capaz de representar os principais atributos e relacionamentos entre essas bases, bem como um banco de dados em MongoDB orientado a documentos, possibilitando consultas eficientes e contextualizadas. A viabilidade e o impacto dos ataques são inferidos com base nos atributos qualitativos do CAPEC, enquanto o tratamento de risco é expandido por meio das estratégias de mitigação extraídas da CWE, complementando aspectos operacionais não previstos na norma. A metodologia proposta foi avaliada por meio de um estudo de caso que simula ataques ao sistema de farol de um veículo por meio da interface Bluetooth, demonstrando sua aplicabilidade prática e capacidade de oferecer maior rastreabilidade, fundamentação técnica e consistência ao processo de análise de riscos cibernéticos.

Palavras-chave: Segurança cibernética; Modelo de ameaças; Bases de segurança; ISO/SAE 21434; Sistemas automotivos autônomos.

ABSTRACT

The growing interconnectivity among computing systems, embedded devices, and communication networks has intensified concerns regarding cybersecurity, particularly in critical domains such as autonomous vehicles. In this context, the ISO/SAE 21434 standard was developed to provide guidelines for Threat Analysis and Risk Assessment (TARA) throughout the automotive system lifecycle. To support this process, techniques such as the STRIDE model (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege) and attack tree analysis have been widely adopted to support the standard's methodology. Complementarily, structured databases such as CAPEC (Common Attack Pattern Enumeration and Classification) and CWE (Common Weakness Enumeration) offer comprehensive catalogs of attack patterns and weaknesses, with the potential to enrich threat identification and security requirement definition. However, the lack of a systematic integration between these databases and normative models remains a gap in the literature, hindering their practical application. To address this issue, this work proposes a methodological approach to support cybersecurity risk analysis in autonomous automotive systems, in accordance with ISO/SAE 21434. The proposal aims to structurally integrate the STRIDE threat model, the CAPEC attack pattern catalog, and the CWE weakness database, supporting the identification, evaluation, and treatment stages of the TARA process. For this purpose, a data model was developed to represent the main attributes and relationships among these bases, along with a document-oriented database implemented in MongoDB, enabling efficient and context-aware queries. Attack feasibility and impact rating are inferred from CAPEC's qualitative attributes, while the risk treatment is enhanced through mitigation strategies extracted from CWE, complementing operational aspects not explicitly addressed by the standard. The proposed methodology was evaluated through a case study simulating attacks on a vehicle's headlight system via Bluetooth, demonstrating its practical applicability and its ability to provide traceability, technical foundation, and consistency to the cybersecurity risk analysis process.

Keywords: Cybersecurity; Threat model; Security databases; ISO/SAE 21434; Autonomous automotive systems.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

ATT&CK	Adversarial Tactics, Techniques & Common Knowledge
CAL	Cybersecurity Assurance Level
CAN	Controller Area Network
CAPEC	Common Attack Pattern Enumeration and Classification
CIA	Confidentiality, Integrity and Availability
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
DoS	Denial of Service
DSL	Domain-Specific Language
DSRC	Dedicated Short-Range Communications
ECU	Electronic Control Unit
ENISA	European Network and Information Security Agency
FFRDC	Federally Funded Research and Development Centers
GPS	Global Positioning System
HEAVENS	HEAling Vulnerabilities to Enhance Software Security and Safety
IEEE	Institute of Electrical and Electronics Engineers
ICS	Industrial Control System
ISO	International Organization for Standardization
LIDAR	LIght Detection and Ranging
LOD	Linked Open Data
NASA	National Aeronautics and Space Administration
NIST	National Institute of Standards and Technology
NVD	National Vulnerability Database
RADAR	Radio Detection and Range Sensor
SAE	Society of Automotive Engineers
SAST	Static Application Security Testing
SOAR	Security Orchestration, Automation and Response
STIX	Structured Threat Information eXpression
STRIDE	Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege
TARA	Threat Analysis and Risk Assessment
UML	Unified Modeling Language
UNECE	United Nations Economic Commission for Europe
UNR	United Nations Regulation
USGCB	United States Government Configuration Baseline
V2I	Vehicle to Infrastructure
V2N	Vehicle to Network
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything

SUMÁRIO

1 INTRODUÇÃO

A segurança cibernética, também conhecida por cibersegurança, pode ser compreendida como o conjunto de práticas destinadas à proteção de sistemas computacionais, dispositivos móveis, redes e dados contra acessos não autorizados e ataques maliciosos [1]. Com o avanço exponencial da tecnologia e a crescente interconectividade dos sistemas, essa área tornou-se um pilar essencial não apenas para o funcionamento seguro da infraestrutura digital tradicional, mas também para sistemas mais complexos, como os sistemas ciberfísicos [2].

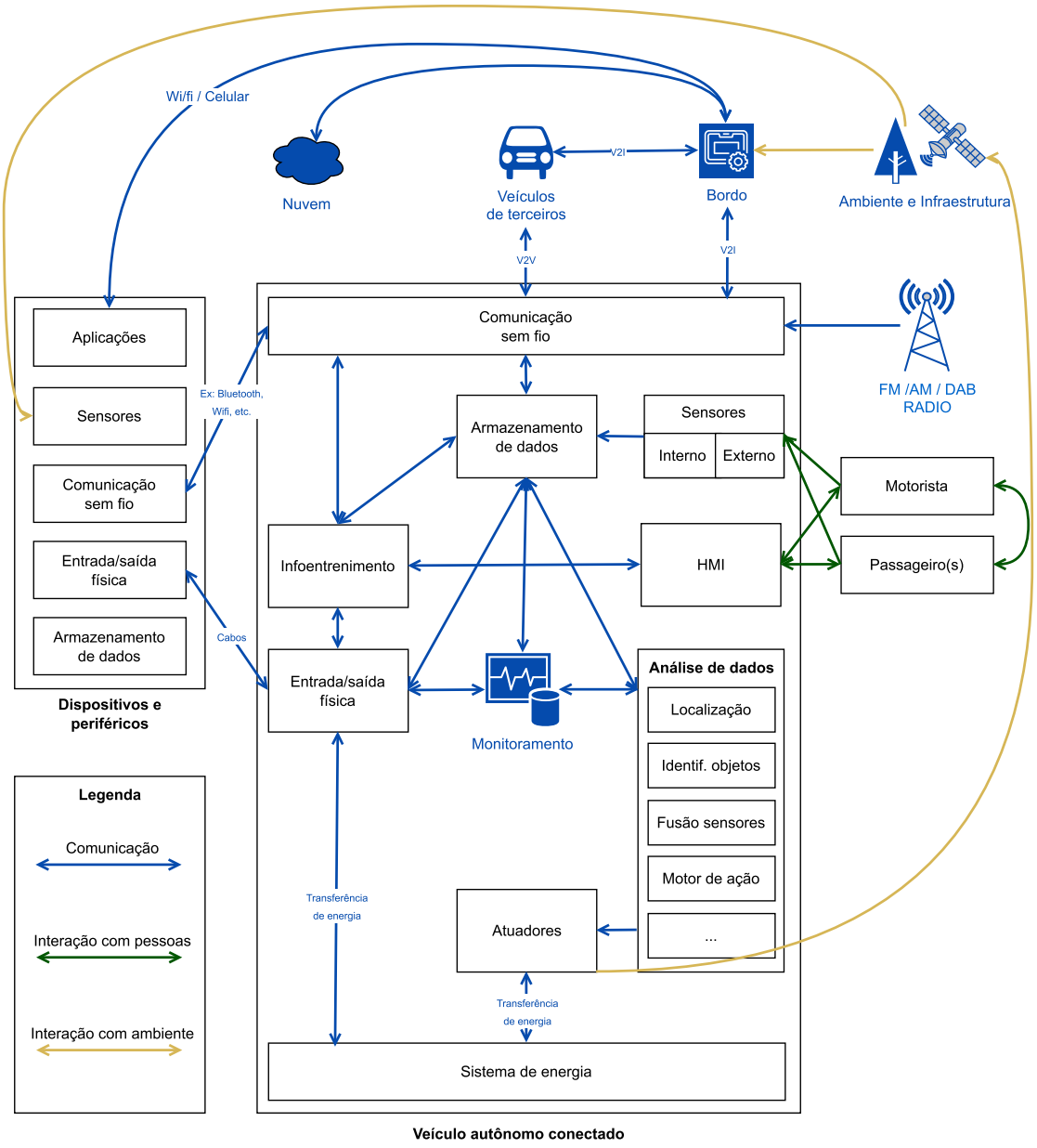
Esses sistemas, caracterizados pela integração entre componentes computacionais, de comunicação e físicos, têm se tornado cada vez mais presentes em infraestruturas críticas[i], desempenhando papel fundamental na transformação digital de setores estratégicos. O aumento da complexidade e da capacidade computacional desses sistemas impulsiona o desenvolvimento de mecanismos de autoadaptação, particularmente em contextos onde a tomada de decisão autônoma em tempo real é crucial. Nesse cenário, emergem os sistemas robóticos críticos para a segurança, como veículos autônomos, robôs logísticos e plataformas de transporte inteligente, cuja adoção tende a crescer de forma exponencial nas próximas décadas, exigindo validação e certificação contínuas em tempo de execução [3].

No contexto automotivo, essa transformação se manifesta através da incorporação de sensores, atuadores e interfaces de comunicação que possibilitam a coleta e o intercâmbio de dados entre veículos e a infraestrutura viária. Essa arquitetura distribuída, característica dos veículos autônomos e conectados, define um ecossistema altamente dinâmico e sensível, no qual a confiabilidade dos sistemas está diretamente relacionada à segurança dos ocupantes e do ambiente [4]. A Figura 1 apresenta a arquitetura de um carro autônomo, destacando suas conexões internas e interações com agentes externos. A figura ilustra os componentes que permitem a interligação entre sistemas internos do veículo e agentes externos, tais como sensores, módulos de análise e armazenamento, monitoramento e atuadores integrados ao sistema de energia. A comunicação sem fio permite a interação com a nuvem, veículos de terceiros e infraestrutura externa, enquanto a interface HMI (*Human Machine Interface*) conecta o sistema a motoristas e passageiros. Essa arquitetura sintetiza os fluxos de dados e controle que permitem o funcionamento do veículo, evidenciando sua complexidade operacional.

Essa evolução tecnológica dos veículos está intimamente associada à transformação em cidades inteligentes, um conceito que engloba dimensões como mobilidade, governança, sustentabilidade e infraestrutura digital integrada [6]. Nesse contexto, destaca-se a mobilidade inteligente, viabilizada pela adoção de veículos autônomos e conectados como

[i] Sistemas, instalações e ativos essenciais para o funcionamento da sociedade e economia [Fonte]

Figura 1 – Arquitetura de um carro autônomo



Fonte: Adaptado de Maple et al. [5]

instrumentos para aumentar a eficiência e a sustentabilidade da mobilidade urbana. No entanto, ao dependerem de comunicação constante com outros veículos e com elementos da infraestrutura urbana, esses sistemas ampliam significativamente a superfície de ataque cibernético, tornando-se alvos atrativos para agentes maliciosos [7].

Diante dessa nova realidade, o setor automotivo respondeu com iniciativas normativas, dentre as quais destaca-se a ISO/SAE 21434 [8], voltada especificamente à gestão da segurança cibernética ao longo de todo o ciclo de vida do veículo. Essa norma estrutura o processo de TARA (*Threat Analysis and Risk Assessment*), promovendo a identificação, avaliação e tratamento de riscos cibernéticos em sistemas automotivos. No entanto, por se

tratar de uma norma orientada a processos, ela não fornece especificações diretas para tecnologias, soluções ou métodos de remediação [9], o que resulta em lacunas práticas, especialmente nas etapas de identificação de ameaças e definição de contramedidas. Assim, modelos como o STRIDE (que categoriza ameaças em seis classes, sendo: *Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service* e *Elevation of Privilege*), e ferramentas como árvores de ataque (*attack tree analysis*) vêm sendo empregados para apoiar o processo TARA em nível sistêmico [10, 11].

Complementarmente, catálogos estruturados de segurança, tais como o CAPEC (*Common Attack Pattern Enumeration and Classification*) [12] e o CWE (*Common Weakness Enumeration*) [13] oferecem mecanismos valiosos para descrever, classificar e compreender as vulnerabilidades exploráveis e os caminhos de ataque utilizados por agentes maliciosos. O CAPEC contribui com a caracterização tática dos ataques, enquanto o CWE permite identificar fraquezas estruturais em diferentes fases do ciclo de vida do *software*. No entanto, a literatura ainda apresenta lacunas quanto à integração entre essas bases e os modelos propostos pelas normas, limitando a eficácia das análises em ambientes de alta complexidade, como o automotivo.

Diante desse cenário, este trabalho propõe uma metodologia que integra o modelo de ameaças STRIDE ao catálogo de ataques CAPEC e à base de fraquezas CWE, com o objetivo de apoiar a identificação de cenários de ameaça, caminhos de ataque e definição de riscos em conformidade com a norma ISO/SAE 21434. A proposta busca preencher a lacuna existente entre modelos de risco e bases de dados estruturadas, proporcionando um processo mais sistemático e fundamentado para a análise de ameaças em sistemas automotivos. Como principais contribuições, destacam-se: (i) o desenvolvimento de um modelo de dados capaz de representar os atributos e relações existentes entre o STRIDE, CAPEC e CWE; (ii) a construção de um banco de dados estruturado contendo mapeamentos entre os elementos das bases, viabilizando consultas e análises integradas; (iii) a definição de uma ontologia que estabelece as relações entre os elementos das bases e o processo TARA; e (iv) uma avaliação por meio de um estudo de caso que simula ataques via Bluetooth a sistemas de iluminação automotiva. Os resultados obtidos reforçam o potencial da abordagem proposta em oferecer maior abrangência e precisão na identificação de riscos cibernéticos.

1.1 TRABALHOS RELACIONADOS

A crescente demanda por soluções de cibersegurança voltadas ao setor automotivo tem motivado diversas iniciativas acadêmicas e industriais que visam estruturar metodologias de análise de risco alinhadas às particularidades dos veículos modernos. Com a evolução das tecnologias embarcadas e o aumento da conectividade veicular, diferentes propostas vêm sendo desenvolvidas com o intuito de aprimorar os processos de identificação, avaliação e mitigação de ameaças cibernéticas. Esta seção apresenta um panorama de

trabalhos relevantes que, de diferentes formas, contribuem para o amadurecimento das práticas de segurança automotiva, seja por meio da adaptação de modelos pré-existentes, do desenvolvimento de novas abordagens formais, ou da análise comparativa de metodologias aplicáveis ao contexto da norma ISO/SAE 21434.

O trabalho de Esmaeili e Esterabadi [14] apresenta uma análise comparativa de metodologias de ataque aplicadas à indústria automotiva. Este estudo foi conduzido em 2019, momento em que a norma ISO/SAE 21434 ainda não havia sido publicada. Portanto, os autores alinharam sua abordagem principalmente às diretrizes da SAE J3061 [15], então reconhecida como o primeiro esforço formal para estruturar práticas de cibersegurança no setor automotivo. O trabalho investiga arquiteturas veiculares, avaliando diferentes modelos e metodologias (como STRIDE, árvores de ataque e grafos de ataque) para indicar a mais adequada no setor automotivo, com apoio de entrevistas realizadas com engenheiros da indústria e especialistas acadêmicos. Como contribuição, os autores propõem uma abordagem híbrida entre árvores e grafos de ataque, através de um modelo de pontuação quantitativa para determinar a viabilidade de ataque baseando-se em parâmetros como o tempo, conhecimento técnico e acessibilidade. A dissertação antecipa conceitos da ISO/SAE 21434, como a análise contínua de riscos e a integração de ataques ao processo de avaliação de risco.

O trabalho de Dantas et al. [16], realizado em 2020, propõe uma abordagem de engenharia de segurança voltada à automação e manutenção incremental das atividades exigidas pela ISO/SAE 21434, com foco na certificação contínua de veículos conectados. O trabalho apresenta o conceito de avaliações de segurança rigorosas (*Rigorous Security Assessments*), utilizando modelos formais baseados em linguagem específica de domínio (*Domain-Specific Language - DSL*) para construção de argumentos de segurança completos e verificáveis; introduzindo também métodos incrementais para atualização dos artefatos de segurança frente a modificações no sistema ou no ambiente operacional (*Incremental Assessment Maintenance*). Além disso, os autores demonstram como diversas etapas do processo de análise de risco (como identificação de ameaças, caminhos de ataque, determinação de risco e sugestão de contramedidas) podem ser automatizadas por meio de uma ferramenta baseada em lógica.

Lautenbach et al. [17] apresentam o modelo HEAVENS 2.0 (*HEAling Vulnerabilities to Enhance Software Security and Safety*) como uma evolução do HEAVENS original, desenvolvido para fornecer uma abordagem sistemática na derivação de requisitos de segurança para sistemas elétricos e eletrônicos automotivos [18]. Embora a versão 1.0 realizasse a avaliação de riscos cibernéticos, ela carecia de alinhamento formal com as normas internacionais emergentes. Com a publicação da ISO/SAE 21434 e dos regulamentos UNR[ii] 155 e 156 da UNECE (*United Nations Economic Commission for Europe*),

[ii] Regulamentos que estabelecem requisitos de cibersegurança e software para veículos [Fonte]

tornou-se necessário atualizar o modelo para atender às exigências regulatórias e fortalecer sua aplicabilidade na indústria. O HEAVENS 2.0 reestruturou o fluxo metodológico para contemplar os conceitos e artefatos definidos pela ISO 21434 em 2021, incorporando 17 aprimoramentos sendo 12 voltados ao atendimento direto da norma e 5 para superar limitações práticas do modelo anterior. Entre as melhorias destacam-se: a inclusão da etapa de identificação de cenários de dano, a modelagem explícita de caminhos de ataque, e a reformulação da matriz de risco. O modelo também introduziu parâmetros detalhados para análise de viabilidade e impacto, propondo uma abordagem mais sistemática e consistente para a condução do processo TARA. Assim, o HEAVENS 2.0 diferencia-se como uma das primeiras metodologias adaptadas à ISO/SAE 21434, conciliando rigor normativo com a viabilidade de aplicação prática em contextos industriais.

Existem ainda diversos trabalhos na literatura que buscam aprimorar metodologias e ferramentas aplicadas à análise de riscos no contexto automotivo. Dentre eles, destaca-se o estudo de Saulaiman et al. [19], que investigaram o uso de grafos de ataque como ferramenta de apoio à modelagem de ameaças, evidenciando sua aplicabilidade na etapa de análise do processo TARA. Já Jakobs et al. [20] apresentaram uma heurística voltada ao tratamento de riscos em projetos de desenvolvimento automotivo, com o objetivo de apoiar a tomada de decisões em cenários que envolvem múltiplas alternativas técnicas. Por sua vez, Merola et al. [21] desenvolveram uma abordagem baseada em lógica *fuzzy* para lidar com incertezas associadas à quantificação de riscos, oferecendo maior flexibilidade na etapa de análise.

1.2 OBJETIVOS E METODOLOGIA

Este trabalho tem como objetivo propor um metamodelo através de um diagrama UML (*Unified Modeling Language*) integrado que auxilie o processo TARA da norma ISO/SAE 21434 por meio da utilização do modelo de ameaças STRIDE, do catálogo de ataques CAPEC e da base de fraquezas CWE. O intuito é enriquecer o processo de identificação, análise e tratamento de ameaças, superando as limitações observadas na ISO 21434, especialmente no que tange à etapa de tratamento de risco, que apresenta diretrizes diretas e pouco detalhadas. A metodologia adotada neste trabalho está dividida em cinco partes principais:

1. **Levantamento e análise das bases STRIDE, CAPEC e CWE:** Esta etapa compreende a coleta e organização dos dados de cada base, com ênfase nas relações possíveis entre seus elementos e sua aplicabilidade ao processo TARA;

2. **Modelagem das relações entre as bases:** Através da análise anterior, será desenvolvido um metamodelo que estabeleça conexões entre os elementos das bases,

relacionando as ameaças (STRIDE), caminhos de ataque (CAPEC) e fraquezas descritas (CWE);

3. **Construção de um banco de dados integrado:** As informações mapeadas serão armazenadas em uma estrutura de banco de dados NoSQL, facilitando consultas durante o processo de avaliação de riscos;

4. **Integração com o processo TARA:** O modelo será adaptado para apoiar cada etapa do processo TARA da ISO/SAE 21434, especialmente nas atividades de identificação de ameaças, avaliação de risco e definição de estratégias de mitigação;

5. **Avaliação do modelo:** Será realizado um estudo de caso baseado no sistema de controle de farol presente no Anexo H da ISO, com o objetivo de demonstrar a aplicabilidade prática do modelo proposto.

Com a abordagem proposta, busca-se não apenas tornar a análise de risco em sistemas automotivos mais precisa e fundamentada, mas também oferecer uma base estruturada e reutilizável que possa ser aplicada em outras iniciativas de segurança cibernética, indo além do setor automotivo. Para viabilizar a integração entre as bases e o processo TARA, foram definidos os seguintes objetivos específicos:

1. **Modelagem de Ameaças:** Utiliza-se o modelo STRIDE para identificar e classificar as ameaças relevantes ao sistema em análise, conforme as diretrizes da norma;

2. **Mapeamento de Caminhos de Ataque:** Para cada ameaça identificada, são associadas entradas do catálogo CAPEC, de forma a descrever os mecanismos de ataque plausíveis, sua viabilidade técnica e os impactos esperados;

3. **Tratamento de Risco:** A partir dos ataques CAPEC selecionados, identifica-se um conjunto de fraquezas CWE associadas, a fim de propor mitigações técnicas que possam ser aplicadas ao sistema, promovendo decisões mais assertivas na etapa de tratamento do risco.

Essas estratégias buscam garantir que a identificação de ameaças, os caminhos de ataque e as fraquezas exploráveis estejam devidamente conectados e contextualizados no fluxo de análise e tratamento de riscos. A aplicação sequencial das etapas permite não apenas maior rastreabilidade das decisões tomadas ao longo do processo, mas também a utilização de informações técnicas consolidadas para fundamentar a seleção de contramedidas eficazes. Este trabalho está organizado da seguinte forma: o Capítulo 2 apresenta os fundamentos teóricos necessários para compreender os principais conceitos relacionados à segurança cibernética no contexto automotivo, incluindo a norma ISO/SAE 21434, o modelo de ameaças STRIDE, a técnica de árvores de ataque, bem como as bases de dados

CAPEC e CWE. O Capítulo 3 descreve as principais contribuições do trabalho, com ênfase na modelagem conceitual, no desenvolvimento do banco de dados e na ontologia construída para integrar os elementos da norma às bases de conhecimento analisadas. No Capítulo 4, é apresentada a aplicação do processo TARA da ISO/SAE 21434 em um estudo de caso envolvendo um sistema de iluminação veicular. O Capítulo 5 detalha a aplicação prática do modelo proposto, com foco na análise da viabilidade de ataque, a avaliação de impacto e decisão de tratamento de risco. Por fim, o Capítulo 6 reúne as considerações finais e propõe direções para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 SEGURANÇA

A palavra segurança pode ser entendida como o conjunto de medidas destinadas à proteção contra riscos, perigos ou perdas que possam afetar pessoas, bens ou sistemas [22]. Trata-se de uma palavra amplamente utilizada tanto no cotidiano quanto em ambientes corporativos, assumindo diferentes significados a depender do contexto de sua aplicação [23]. Por ser um termo utilizado em diferentes áreas, existem estratégias de segurança específicas para cada setor, onde em cada situação, há um conjunto específico de medidas a serem tomadas. Entre os principais tipos de segurança amplamente reconhecidos, destacam-se a segurança nacional, a segurança da informação, a segurança no trânsito e a segurança privada [24].

2.1.1 *Safety* e *security*

Em inglês, a palavra segurança pode possuir diferentes significados, sendo os termos mais comuns representados por *security* e *safety*, que correspondem, respectivamente, à proteção contra ameaças intencionais e à segurança física diante de falhas ou acidentes. No contexto de sistemas embarcados[i], tais palavras podem ser entendidas como [25]:

- *Safety:* Um sistema que durante a sua operação normal não causa danos ao usuário ou qualquer outra pessoa. Quando um sistema apresenta potencial de causar danos físicos ou até mesmo a morte em casos de falha ou mau-funcionamento, eles são classificados como *safety-critical*.
- *Security:* Um sistema que possui a capacidade de manter os seus valores e funções disponíveis à um usuário autorizado, enquanto se protege de acessos não autorizados. O objetivo é evitar que alguém possa modificar o fluxo de execução, obter dados privados, corromper informações ou até mesmo indisponibilizar o sistema.

Em resumo, *safety* refere-se à prevenção de acidentes e perigos não intencionais, enquanto *security* concentra-se na proteção contra ameaças intencionais (como crimes e ataques) [26]. Em muitos casos, *security* é necessário para garantir *safety*, uma vez que vulnerabilidades podem resultar em consequências severas para seres humanos ou o meio ambiente [27]. No contexto de sistemas automotivos autônomos, essa relação torna-se ainda mais crítica, pois falhas na proteção cibernética podem comprometer diretamente a segurança dos ocupantes, podendo levar a acidentes graves ou fatais.

A identificação dos fatores afetados causados por um acidente (ou evento perigoso) é essencial para a caracterizar seus impactos, abrangendo aspectos como a saúde humana,

[i] Sistema eletrônico microprocessado dedicado ao dispositivo ou sistema que ele controla [Fonte]

bens materiais (propriedade) e o meio ambiente [28]. Por exemplo, Pelaric et al. [3] discute que, no caso de um acidente veicular, é possível que uma pessoa dentro do veículo sofra lesões, afetando o fator saúde. Além disso, colisões também podem ocasionar danos estruturais ao próprio automóvel, comprometendo o patrimônio e, portanto, atingindo o fator propriedade. Por fim, é possível que o acidente provoque danos a elementos naturais ou construções no entorno, gerando efeitos sobre o meio ambiente.

2.2 SEGURANÇA DA INFORMAÇÃO

De acordo com Whitman e Mattord [29], a segurança da informação é definida como a proteção da informação e de seus elementos críticos, incluindo os sistemas e dispositivos que utilizam, armazenam e transmitem esses dados. A segurança da informação não se limita à adoção de produtos ou tecnologias específicas, configurando-se como um processo contínuo [30].

Com a crescente digitalização, a segurança da informação passou por uma transformação conceitual significativa. Antes tratada como uma disciplina exclusivamente técnica, voltada à proteção de sistemas computacionais, ela passou a abranger também aspectos organizacionais, humanos e estratégicos. A evolução das tecnologias, aliada à complexidade dos sistemas e conexões em escala global, evidenciou que a proteção dos ativos informacionais requer políticas estruturadas, controles internos, conscientização dos usuários e processos institucionais bem definidos [31].

Nesse contexto, a segurança da informação consolidou-se como um componente essencial da governança corporativa e da resiliência organizacional, sendo tratada como um processo sistemático de identificação, avaliação e mitigação de riscos [32]. De modo geral, a proteção da informação é definida com base em um conjunto de propriedades fundamentais que a informação deve preservar, entre as quais se destacam-se a confidencialidade, a integridade e a disponibilidade, podendo incluir características adicionais [33].

2.2.1 CIA

O termo CIA significa uma tríade que representa o modelo de segurança da informação mais utilizado e aplicado no mundo [3]. Ele é usado para fornecer a organizações e sistemas o conhecimento necessário sobre como manter os dados seguros. Este modelo consiste nos seguintes aspectos [34]:

- Confidencialidade: prevenção de divulgação não autorizada ou uso de informações;
- Integridade: prevenção de modificação não autorizada de ativos de informação;
- Disponibilidade: garantia de acesso autorizado de informação quando obrigatório.

A tríade CIA é utilizada como base para identificar quais propriedades da informação são comprometidas em um ataque, permitindo compreender os objetivos e impactos das ações maliciosas. Ao analisar ameaças sob essa perspectiva, torna-se possível alinhar medidas de proteção às vulnerabilidades mais críticas de cada sistema.

2.2.2 Cibersegurança

A cibersegurança, como mencionada anteriormente, é a área responsável por proteger sistemas contra ameaças digitais, como ataques cibernéticos, *malware*[ii], *phishing*[iii], entre outras atividades maliciosas. O seu objetivo principal é garantir a confidencialidade, integridade e disponibilidade dos recursos digitais [35].

Apesar da cibersegurança e segurança da informação serem conceitos parecidos, eles apresentam algumas diferenças entre si. A segurança da informação trata da proteção de todos os dados sigilosos de uma empresa (sejam eles arquivados de forma digital ou física); enquanto a cibersegurança limita-se apenas ao meio digital [36].

O principal papel da cibersegurança é garantir que os sistemas e dados estejam protegidos contra ameaças digitais. Isso envolve a implementação de medidas técnicas, educação dos usuários, monitoramento de ameaças em tempo real e resposta a incidentes. Os profissionais de cibersegurança são responsáveis por projetar, implementar e manter essas medidas para proteger os ativos digitais [35].

2.2.3 Ataques cibernéticos

Ataques cibernéticos são ações maliciosas realizadas com o objetivo de comprometer propriedades de cibersegurança de sistemas computacionais, redes ou dados. Esses ataques podem ser conduzidos por indivíduos, grupos organizados ou até mesmo por Estados-nação; envolvendo técnicas que vão desde a exploração de vulnerabilidades técnicas até manipulações sociais através de engenharia social [37].

Segundo Stallings [38], os ataques cibernéticos podem ser classificados em dois grandes grupos: passivos e ativos. Os passivos têm como foco a obtenção de informações sem alteração do sistema-alvo, como na interceptação de dados em trânsito. Já os ataques ativos buscam modificar ou danificar os sistemas e seus dados, por meio de ações como injeção de código, negação de serviço ou elevação de privilégios.

O relatório da ENISA (*European Union Agency for Cybersecurity*, antigamente chamada de *European Network and Information Security Agency*) destaca que os ataques mais recorrentes incluem *ransomware*[iv], exploração de vulnerabilidades conhecidas, ataques de negação de serviço distribuído (*Distributed Denial of Service* - DDoS) e *phishing* [39].

[ii] Código ou programa de computador escrito intencionalmente para prejudicar sistemas [Fonte]
[iii] Técnica usada para enganar usuários usando fraude eletrônica para obter informações [Fonte]
[iv] Software de extorsão que bloqueia o computador, exigindo resgate para desbloqueá-lo [Fonte]

A crescente digitalização e a interconectividade dos sistemas críticos como os automotivos ampliam consideravelmente a superfície de ataque, exigindo abordagens mais robustas de defesa.

2.3 SISTEMAS AUTOMOTIVOS AUTÔNOMOS

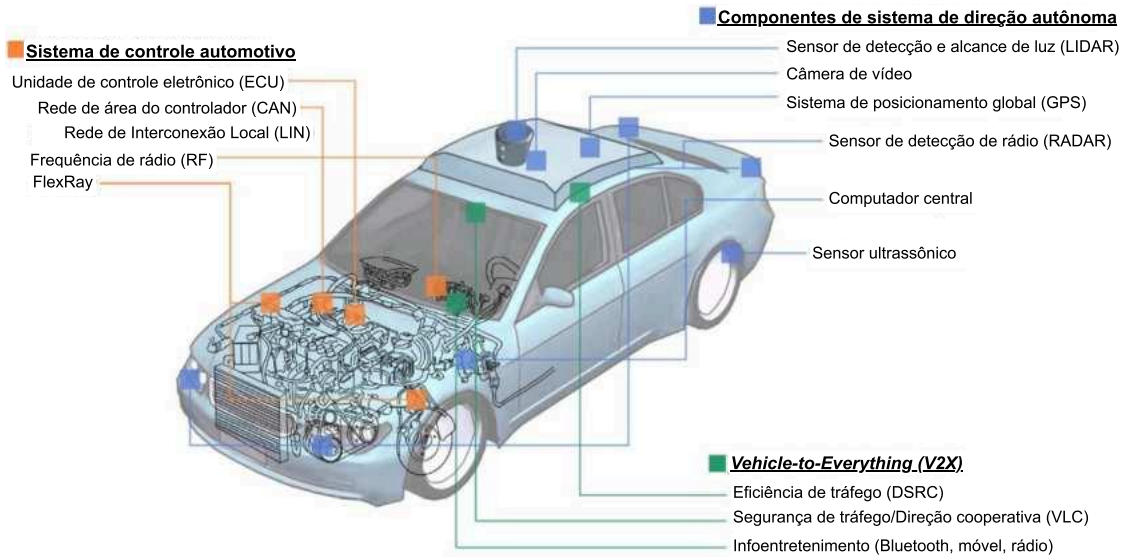
Veículos autônomos são sistemas ciberfísicos complexos, compostos por múltiplas camadas de *hardware* e *software* integradas que permitem sua operação independente de intervenção humana. Esses sistemas são capazes de perceber o ambiente, processar dados em tempo real e executar ações de controle (como aceleração, frenagem e direção) a partir da interpretação de sinais externos, do comportamento de outros veículos ou da infraestrutura rodoviária [7]. Esse ecossistema dinâmico configura os veículos autônomos como sistemas críticos, cuja confiabilidade está diretamente relacionada à segurança dos ocupantes e do ambiente ao redor [4].

A arquitetura de um veículo autônomo consiste em uma rede que conecta o dispositivo principal com os demais dispositivos, a Figura 2 representa os principais componentes de um carro autônomo. As unidades mais importantes são as unidades de controle eletrônico (*Electronic Control Units* - ECUs), responsáveis por controlar o estado da transmissão automática do motor e gerenciar os sensores dentro do veículo [40]. Veículos de pequeno e médio porte costumam ter cerca de 50 ECUs [41], modelos de luxo costumam ter mais 70 ECUs [42], e alguns de última geração possuem até 80 ECUs devido às novas funcionalidades [43]. Essas ECUs são interconectadas por barramentos de comunicação, incluindo a rede de área do controlador (*Controller Area Network* - CAN), um dos protocolos mais utilizados para troca de mensagens em tempo real entre os componentes críticos do veículo, garantindo a troca eficiente e em tempo real de mensagens entre os subsistemas críticos [40, 44, 45]. O protocolo CAN é um padrão ISO de comunicação de dados, sendo registrado como ISO 11.898 [46].

Para operar de forma autônoma, os veículos contam com um conjunto de sensores que fornecem percepção abrangente do ambiente. Entre os principais dispositivos, destacam-se o sensor de detecção e alcance de luz (*LIght Detection And Ranging* - LIDAR), sensor de detecção e alcance de rádio (*RAdio Detection And Range* - RADAR), câmeras, sistema de posicionamento global (*Global Positioning System* - GPS) e sensores ultrassônicos, que, integrados a um computador central, permitem a detecção precisa de obstáculos, sinais e outros veículos [47, 48]. Esses dados são continuamente processados para gerar comandos de aceleração, frenagem e direção de forma segura e adaptativa.

Além da percepção local, a comunicação com o ambiente externo denominada *Vehicle-to-Everything* (V2X) desempenha papel crucial na condução cooperativa e na segurança. As tecnologias V2V (*Vehicle-to-Vehicle*), V2I (*Vehicle-to-Infrastructure*) e V2N (*Vehicle-to-Network*) permitem o intercâmbio de informações entre veículos, semáforos,

Figura 2 – Principais componentes de um carro autônomo



Fonte: Adaptado de Kim et al. [40]

sensores de estrada e serviços em nuvem. Tais comunicações ocorrem por meio de protocolos como DSRC (Dedicated Short-Range Communications), C-V2X (*Cellular-V2X*) e redes móveis, viabilizando ações coordenadas, como frenagens automáticas em grupo ou reações a condições de tráfego adversas [40]. Por fim, os veículos também incluem interfaces de diagnóstico, como o OBD-II, que permitem o monitoramento de parâmetros internos e o acesso a dados operacionais via conectores acessíveis [16].

2.4 ISO/SAE 21434

A ISO/SAE 21434 *Road Vehicles - Cybersecurity Engineering* [8] é um padrão da indústria automotiva que define os requisitos de segurança cibernética para veículos rodoviários. Ele foi desenvolvido pela Organização Internacional de Padronização (*International Organization for Standardization* - ISO) e pela Sociedade de Engenheiros Automotivos Internacional (*Society of Automotive Engineers* - SAE International). Esta norma de segurança visa substituir a norma anterior J3061:2016 [15] com recomendações mais estruturadas, a fim de garantir a cibersegurança automotiva [49]. As principais características deste padrão são [9]:

(a) Definir um processo estruturado para garantir um design ciberseguro;

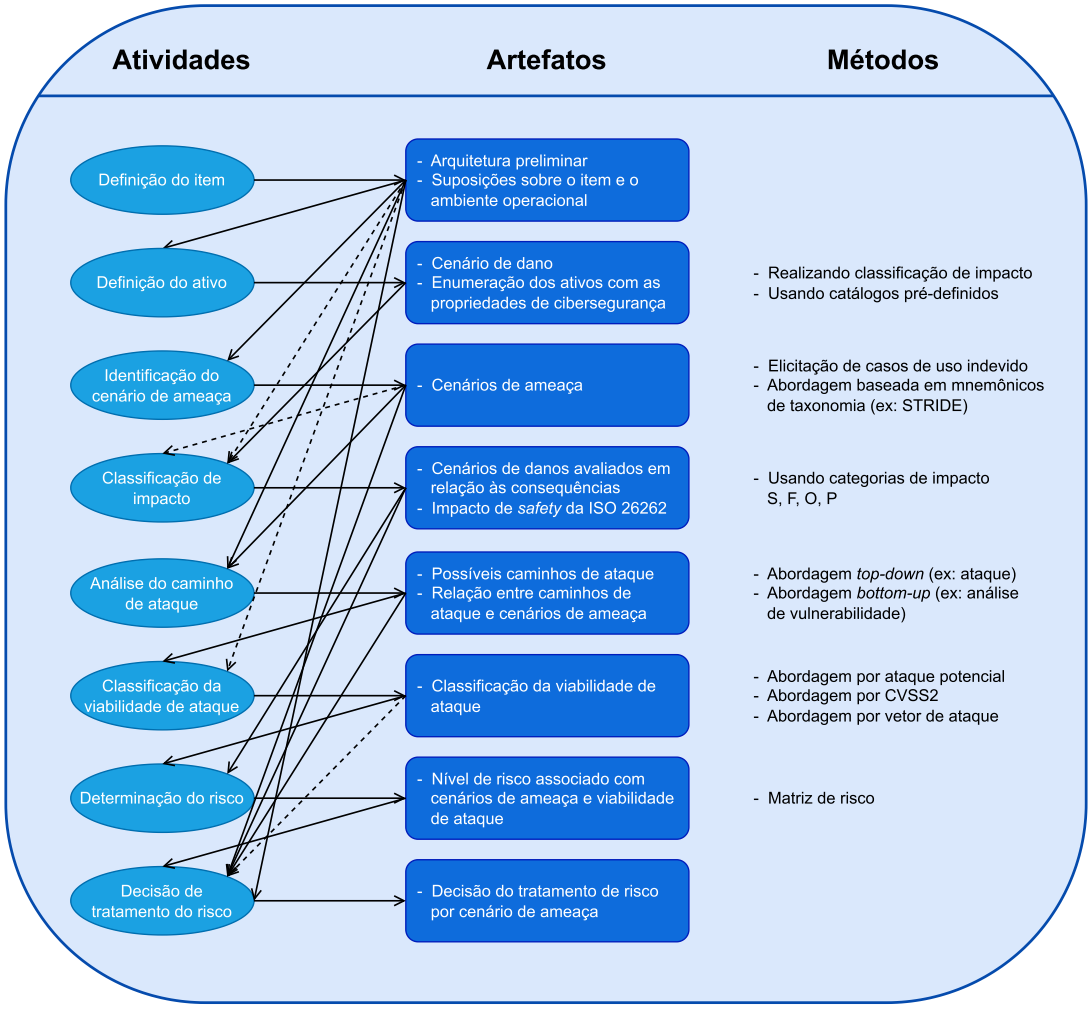
(b) Reduzir o potencial de um ataque bem-sucedido e a probabilidade de perdas;

(c) Fornecer meios claros para reagir a ameaças de segurança cibernética.

Seu principal objetivo é garantir a segurança digital durante todo o ciclo de vida do veículo, desde a fase conceitual até o descomissionamento, promovendo a identificação,

avaliação e mitigação sistemática de ameaças e riscos cibernéticos. Contudo, a norma não prescreve métodos técnicos específicos nem soluções prontas, tampouco define requisitos exclusivos para veículos autônomos ou infraestrutura rodoviária. Em vez disso, propõe um fluxo estruturado de análise que pode ser adaptado conforme o contexto do sistema e as ameaças identificadas [9]. A Figura 3 ilustra o modelo proposto pela ISO 21434 para a condução do processo TARA, destacando a relação entre atividades, artefatos gerados e métodos recomendados em cada etapa.

Figura 3 – Atividades, artefatos e métodos definidos na ISO 21434



Fonte: Adaptado de Dantas et al. [16]

O processo começa com a definição do item (sistema), que inclui a definição da arquitetura preliminar, bem como suposições relevantes sobre o item e seu ambiente operacional. Na ISO, um **item** representa um componente ou um conjunto de componentes que implementam uma função no nível do veículo. Após definir os limites do item e seu ambiente operacional, os ativos, suas propriedades de cibersegurança associadas e cenários de danos são identificados. Um **ativo** (*asset*) é um objeto que tem valor ou contribui para

um valor, podendo ser um sistema, componente, dados (informações) ou comunicação de dados. Uma **propriedade de cibersegurança** (*cybersecurity property*) é um atributo de um ativo cuja sua proteção é considerada essencial, como, por exemplo, confidencialidade, integridade e/ou disponibilidade. Um **cenário de dano** (*damage scenario*) é uma consequência adversa envolvendo um veículo ou sua função, afetando um usuário da estrada. Um **cenário de ameaça** (*threat scenario*) é uma causa potencial de violação das propriedades de cibersegurança de um ou mais ativos para realizar um cenário de dano.

A atividade de **identificação de ativos** envolve a etapa de **identificação de cenários de dano**, cujo objetivo é reconhecer possíveis violações às propriedades de cibersegurança associadas a cada ativo. Os cenários de dano levantados são, então, avaliados quanto às suas possíveis consequências adversas para os usuários da estrada, categorizando o impacto nas seguintes dimensões: segurança (S), financeira (F), operacional (O) e privacidade (P). O **impacto** refere-se ao grau de severidade do dano ou prejuízo físico decorrente da concretização de um cenário de dano. A **classificação de impacto** (*impact rating*) pode ser definida como grave, maior, moderada ou insignificante, conforme os critérios estabelecidos para avaliação, os quais são derivados da norma de segurança funcional ISO 26262 [50], aplicável a veículos rodoviários.

A **identificação do cenário de ameaça** descreve um conjunto de ações potenciais que podem levar a um ou mais cenários de dano, que especificam as consequências adversas (resultados) de um ataque, por exemplo, usando o método de análise de ameaças STRIDE. Um cenário de ameaça pode levar à ocorrência de vários cenários de dano. A ISO 21434 também prescreve a realização da análise de caminhos de ataque para identificar os possíveis ataques que podem realizar cada cenário de ameaça. Um **caminho de ataque** (*attack path*) é um conjunto de ações deliberadas para realizar um cenário de ameaça. A análise do caminho de ataque pode ser realizada usando abordagens *top-down*, partindo do cenário de ameaça para identificar possíveis caminhos, ou *bottom-up*, que partem das vulnerabilidades detectadas nos ativos do sistema para inferir possíveis ameaças e rotas de ataque. Posteriormente, cada caminho identificado deve receber uma classificação de viabilidade de ataque (alta, média, baixa ou muito baixa). A **classificação de viabilidade de ataque** (*attack feasibility rating*) pode ser determinada com base em critérios como tempo decorrido, conhecimento prévio, janela de oportunidade e recursos técnicos, conforme definidos pela ISO/IEC 18045 [51].

A ISO/SAE 21434 estabelece que o valor de risco de um cenário de ameaça deve ser obtido considerando os maiores valores atribuídos à viabilidade de ataque e ao impacto do cenário de dano associado, de forma a não subestimar o risco potencial da ameaça [8]. A combinação desses dois fatores viabilidade de ataque e impacto é utilizada como entrada para determinar uma matriz de risco, que define cinco níveis de criticidade (de 1 a 5), os quais orientam a classificação dos cenários de ameaça segundo sua gravidade. A partir dessa classificação, deve-se tomar uma decisão quanto ao tratamento do risco,

selecionando uma entre quatro opções: evitar o risco (eliminando sua causa), transferi-lo ou compartilhá-lo (via contratos ou seguros), aceitá-lo (quando considerado gerenciável sem medidas adicionais), ou reduzi-lo por meio da implementação de controles de segurança adicionais no item analisado. Para os cenários cujo risco é reduzido, a norma prevê a definição de **objetivos de cibersegurança** (*cybersecurity goals*), os quais consistem em requisitos de alto nível que orientam o desenvolvimento de medidas mitigatórias. Esses objetivos são formalizados por meio de níveis de garantia de cibersegurança (*Cybersecurity Assurance Levels* - CALs), que definem o rigor necessário nas atividades de validação. Por exemplo, ameaças classificadas com CAL 4 exigem técnicas avançadas de verificação, como testes funcionais, análise de vulnerabilidades, testes de penetração e *fuzzing*. Já nos casos de aceitação ou transferência do risco, devem ser definidas **reivindicações de cibersegurança** (*cybersecurity claims*), que justificam a aceitabilidade do risco ou apresentam suposições que precisam ser atendidas para que o risco seja considerado tolerável.

2.5 STRIDE

O STRIDE é uma metodologia desenvolvida pela Microsoft utilizada para identificar e categorizar ameaças cibernéticas em sistemas computacionais [52]. O termo STRIDE é derivado das letras iniciais de diferentes ameaças, sendo classificadas com base nos objetivos e propósitos de ataques. As ameaças são: falsificação (*Spoofing*), adulteração (*Tampering*), repúdio (*Repudiation*), divulgação de informações (*Information disclosure*), negação de serviço (*Denial of Service* - DoS) e elevação de privilégio (*Elevation of privilege*). Essas categorias correspondem diretamente às propriedades de cibersegurança que podem ser comprometidas por ataques, como autenticidade, integridade, confidencialidade, disponibilidade, não repúdio e autorização, conforme ilustrado na Figura 4 [53].

Figura 4 – Categorias de ameaças STRIDE

	Ameaça	Propriedade violada	Definição da ameaça
S	Falsificação de identidade	Autenticação	Fingir ser algo ou alguém diferente de você mesmo
T	Adulteração de dados	Integridade	Modificar algo no disco, rede, memória ou em outro lugar
R	Repúdio	Não repúdio	Afirmar que você não fez algo ou não foi responsável; podendo ser honesto ou falso
I	Divulgação de informações	Confidenciabilidade	Fornecer informações a alguém não autorizado a acessá-las
D	Negação de serviço	Disponibilidade	Esgotamento dos recursos necessários para prestar o serviço
E	Elevação de privilégio	Autorização	Permitir que alguém faça algo que não está autorizado a fazer

Fonte: Adaptado de Shevchenko et al. [53]

A metodologia STRIDE oferece uma abordagem estruturada para a identificação de riscos de cibersegurança, ao mapear possíveis cenários de ameaça a partir das interações

entre os elementos do sistema e suas potenciais vulnerabilidades [54]. Contudo, o modelo não apresenta detalhes técnicos ou métodos específicos de exploração [55], restringindo-se a uma categorização conceitual das ameaças. Assim, para garantir uma análise mais abrangente, considera-se a utilização de fontes de dados que descrevam comportamentos maliciosos, técnicas de exploração e vulnerabilidades conhecidas [56].

O modelo STRIDE permite mapear ameaças a partir da relação entre ativos e atributos de segurança, podendo ser aplicado tanto sob a perspectiva do atacante quanto em função das consequências esperadas do ataque [18]. Ao invés de focar em ataques específicos, o STRIDE orienta a análise para os impactos potenciais, o que contribui para uma organização mais estratégica das defesas. Ele também amplia o tradicional modelo CIA, incorporando atributos adicionais como autenticidade, não repúdio e autorização. Essa abordagem possibilita que as organizações priorizem ameaças com base na sua probabilidade de ocorrência e impacto potencial. Por exemplo, instituições de saúde podem usar o STRIDE para identificar riscos críticos à confidencialidade de dados sensíveis, e, a partir disso, adotar contramedidas mais eficazes [57].

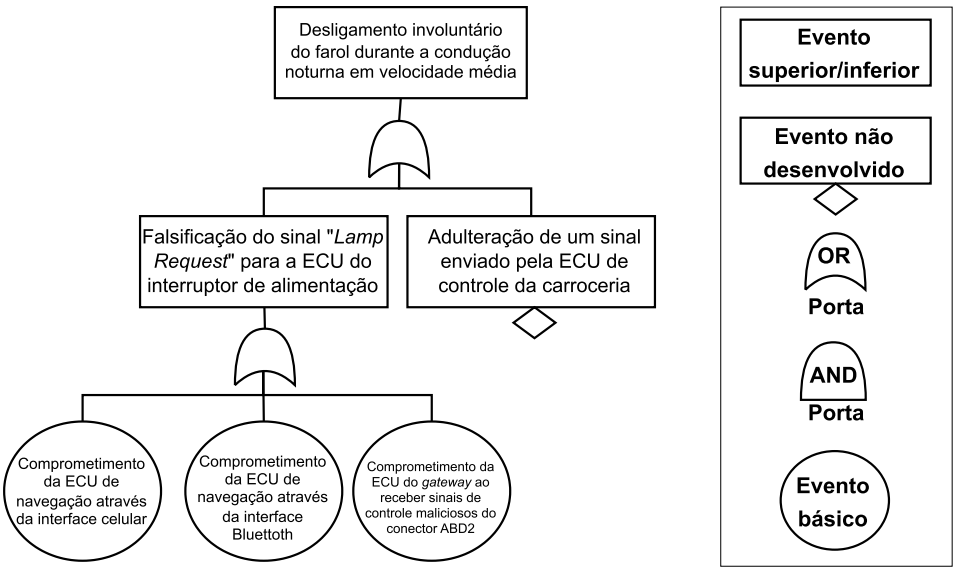
2.6 ÁRVORE DE ATAQUE

A metodologia de árvore de ataque é uma abordagem estruturada para modelar possíveis cenários de ataque contra um sistema, organizando-os de maneira hierárquica [10]. Árvores de ataque são técnicas de raciocínio *top-down* para análise de segurança, fornecendo diagramas que especificam como ataques de nível inferior (eventos básicos) podem ser combinados logicamente para causar violações das propriedades de cibersegurança dos ativos. Esses diagramas são representados por grafos acíclicos direcionados, nos quais os nós internos correspondem a condições intermediárias; e os nós-folha (elementos sem ramificações), a ações básicas do atacante. A estrutura lógica pode incluir diferentes operadores como AND e OR que definem as condições sob as quais um evento superior pode ocorrer. Um mesmo evento básico pode contribuir simultaneamente para múltiplos caminhos de ataque, sendo reutilizado em diferentes ramos da árvore, o que implica que subárvores podem ser compartilhadas entre múltiplos nós superiores.

A Figura 5 apresenta um exemplo de árvore de ataque aplicada a um sistema automotivo, ilustrando a decomposição de um evento indesejado em diferentes etapas de comprometimento. O evento superior descreve um possível cenário de dano, a partir do qual diferentes cenários de ataque são modelados. A árvore subdivide esse evento em causas intermediárias, como a falsificação e a adulteração de sinais. Esses eventos, por sua vez, derivam de ações básicas do atacante, como a exploração de comunicação via Bluetooth e de redes celulares para comprometer os componentes do veículo. A notação empregada inclui símbolos padronizados para representar eventos básicos, intermediários e não desenvolvidos, bem como conectores lógicos do tipo OR. Essa estrutura permite

visualizar a progressão lógica de um ataque, facilitando tanto a análise qualitativa de vulnerabilidades quanto o cálculo quantitativo da probabilidade de ocorrência por meio da atribuição de pesos probabilísticos aos eventos básicos.

Figura 5 – Exemplo de uma árvore de ataque



Fonte: Adaptado de Nascimento et al. [58]

2.7 BASES DE DADOS DE ATAQUES E FRAQUEZAS

De acordo com os princípios do *Linked Open Data* (LOD)[v], o conhecimento de segurança de engenharia e sistemas de controle industrial (*Industrial Control System* - ICS) está interligado com dados de fontes públicas [59]. Por exemplo, informações vinculadas sobre vulnerabilidades (como o *Common Vulnerabilities and Exposures* - CVE) e atividades de ameaças (por exemplo, alertas baseados em *Structured Threat Information eXpression* - *STIX*) podem ser obtidas no Banco de Dados Nacional de Vulnerabilidades dos Estados Unidos (National Vulnerability Database - NVD) e no ICS-CERT (*Industrial Control Systems Cyber Emergency Response Team*), respectivamente.

No entanto, o grande volume de informações disponíveis em fontes públicas impõe desafios significativos à priorização e à resposta de vulnerabilidades. Para mitigar esses riscos, é essencial que as informações sejam coletadas de maneira rápida e estruturada. Além disso, é necessário compreender não apenas a natureza das vulnerabilidades, mas também as técnicas de ataque associadas. A avaliação da gravidade e a prioridade de uma vulnerabilidade exige, portanto, o cruzamento de informações sobre fraquezas conhecidas e padrões de ataque recorrentes. Para coletar essas informações, repositórios públicos referentes a segurança cibernética podem ser usados [60].

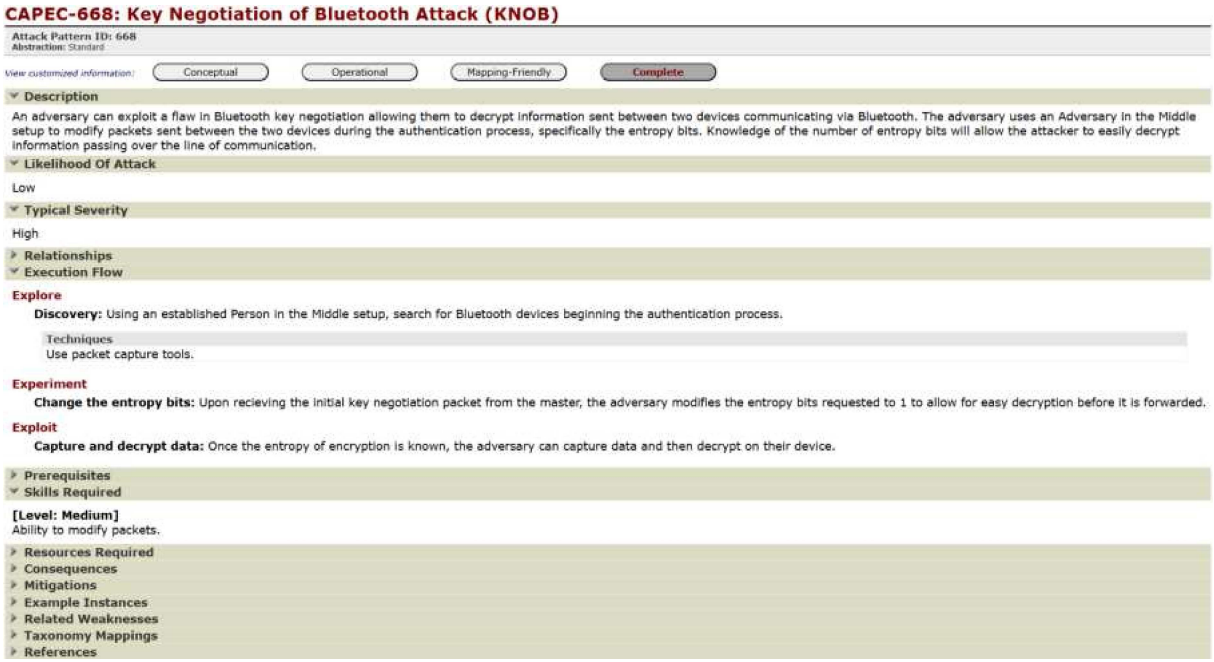
[v] Definição de todos os dados publicados na Web com um conjunto de melhores práticas [Fonte]

Repositórios públicos incluem o *Common Weakness Enumeration* (CWE) [13] e o *Common Attack Pattern Enumeration and Classification* (CAPEC) [12]; ambas as bases são mantidas pela MITRE, uma organização sem fins lucrativos que opera centros de pesquisa e desenvolvimento financiados pelo governo federal dos Estados Unidos (*Federally Funded Research and Development Centers* - FFRDC). O CWE lista as fraquezas comuns de *software* e *hardware*; e o CAPEC é um dicionário de identificadores comuns para padrões de ataque empregados por adversários para explorar fraquezas. Essas bases serão detalhadas a seguir.

2.7.1 CAPEC

O *Common Attack Pattern Enumeration and Classification* (CAPEC) [12] é um catálogo online de padrões de ataques, contendo mais de 500 registros. Um padrão de ataque é uma descrição dos atributos e abordagens comuns usados pelos adversários para explorar pontos fracos conhecidos nos sistemas cibernéticos [61]. Esses padrões descrevem os desafios enfrentados por atacantes e os métodos utilizados para superá-los. A estrutura dos padrões CAPEC é inspirada no conceito de padrões de design, amplamente utilizado em engenharia de software, porém, em vez de representar soluções construtivas, os padrões CAPEC descrevem estratégias destrutivas, derivadas da análise sistemática de incidentes reais e práticas observadas em cenários de ataque do mundo real, permitindo capturar estratégias típicas de exploração observadas em cenários concretos [62]. Um exemplo de registro CAPEC é apresentado na Figura 6.

Figura 6 – Exemplo de um registro CAPEC



Fonte: Common Attack Pattern Enumeration and Classification [12]

Cada padrão de ataque captura conhecimento sobre como partes específicas de

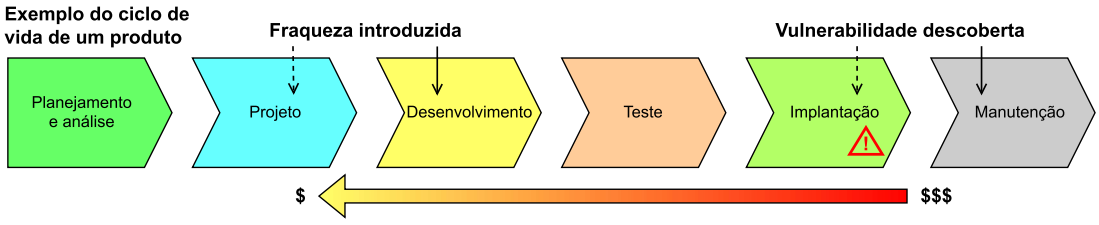
um ataque são projetadas e executadas, fornecendo orientação sobre maneiras de mitigar a eficiência do ataque. Eles são descritos por meio de atributos que podem ser usados para auxiliar na compreensão e mitigação das ameaças. Entre esses atributos, destacam-se a probabilidade de ataque, severidade, consequências, habilidades necessárias, fraquezas e padrões de ataque relacionados. Um catálogo de ataques foca no comportamento dos atacantes e nos métodos empregados para comprometer sistemas. Isso permite que profissionais de segurança compreendam melhor o ciclo de vida de um ataque e implementem estratégias para prevenção e resposta.

2.7.2 CWE

O *Common Weakness Enumeration* (CWE) [13] é um catálogo que descreve fraquezas estruturais comuns em *software*, *hardware* e *firmware*. Essas fraquezas (*weaknesses*) representam condições presentes em um sistema que, sob determinadas circunstâncias, podem ser exploradas, resultando em vulnerabilidades de segurança [63].

A CWE é estruturada como uma taxonomia hierárquica que facilita a análise, categorização e mitigação de causas técnicas de falhas. Ao contrário de vulnerabilidades específicas, as fraquezas descritas no CWE referem-se a problemas típicos e recorrentes que afetam a segurança e a confiabilidade de sistemas computacionais. Tais fraquezas são oriundas, em sua maioria, de erros cometidos nas fases iniciais do ciclo de vida do desenvolvimento, como a etapa de projeto ou codificação [64]. A Figura 7 ilustra a relação entre as etapas do ciclo de vida do produto, apresentando a relação entre o momento em que as fraquezas são introduzidas com os custos associados à sua correção. À medida que o sistema avança nas fases de desenvolvimento (passando por planejamento, projeto, desenvolvimento, testes e implantação), os custos para detectar e corrigir uma falha aumentam consideravelmente. Detectar uma fraqueza logo após sua introdução, ainda na fase de design, representa um custo muito menor do que identificá-la após a fase de operação ou manutenção. Esse cenário reforça a importância de mecanismos preventivos e de boas práticas de segurança desde as fases iniciais do desenvolvimento.

Figura 7 – Associação entre descoberta de fraquezas e impacto financeiro

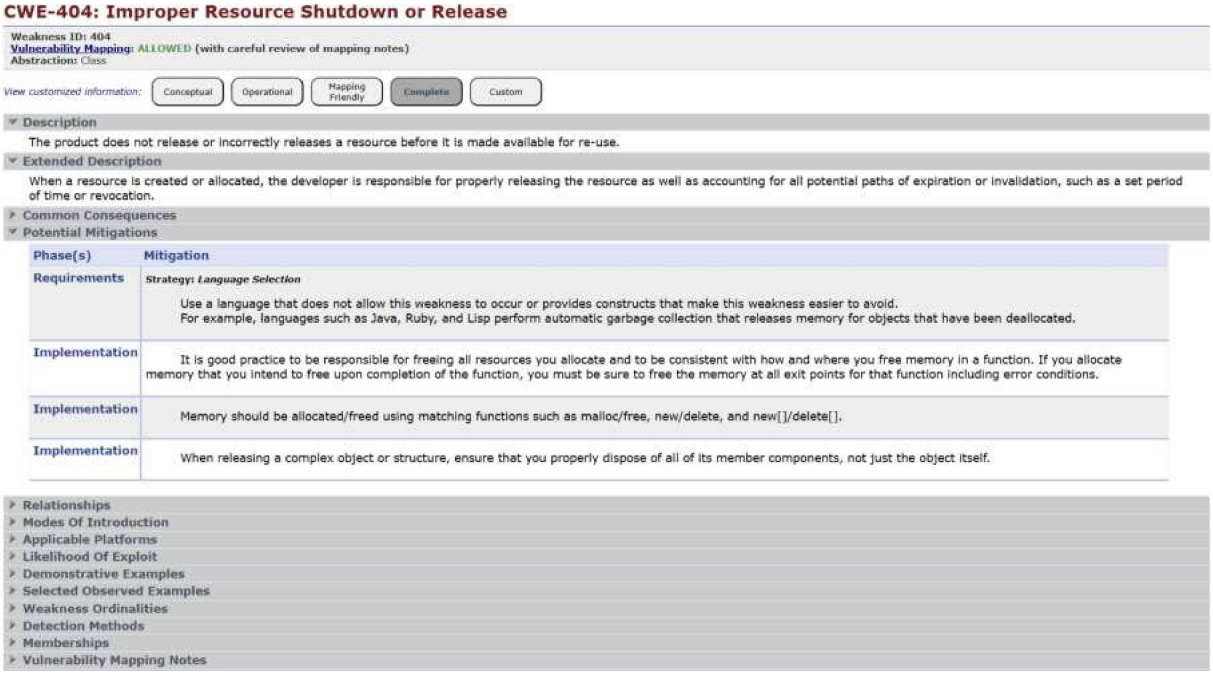


Fonte: Adaptado de MITRE [63]

Um exemplo de registro CWE é apresentado na Figura 8. Cada entrada no catálogo

do CWE define a natureza da fraqueza, suas possíveis causas, consequências, métodos de detecção e técnicas de mitigação. Além disso, são fornecidas informações adicionais como exemplos de código, mecanismos de ataque e sua relação com outras entradas do catálogo. Ao mapear essas fraquezas de forma sistemática, o CWE permite que profissionais da área de segurança compreendam melhor os fatores que levam à introdução de fraquezas em sistemas complexos.

Figura 8 – Exemplo de um registro CWE



Fonte: Common Weakness Enumeration [13]

3 DESENVOLVIMENTO DO MODELO PROPOSTO

Esta seção apresenta as principais contribuições deste trabalho, que envolvem: (i) a construção de um modelo de dados para representar, de forma estruturada, os relacionamentos entre o modelo de ameaças STRIDE, o catálogo de ataques CAPEC e a base de fraquezas CWE; (ii) o desenvolvimento de um banco de dados orientado a documentos, dedicado ao armazenamento e consulta das informações extraídas dessas bases; e (iii) a implementação de uma abordagem metodológica para integrar, de forma sistemática, os elementos das bases ao processo TARA em conformidade com a ISO/SAE 21434.

Cabe destacar que, devido ao tamanho e à complexidade dos diagramas apresentados, algumas figuras podem apresentar perda de nitidez ou limitar a visualização completa de seus elementos. Para melhor examinar os modelos, estruturas e relações propostas neste trabalho, recomenda-se o acesso direto ao projeto hospedado no Google Drive através da ferramenta Diagrams.net[i], disponível em: https://app.diagrams.net/

3.1 MODELO DE DADOS

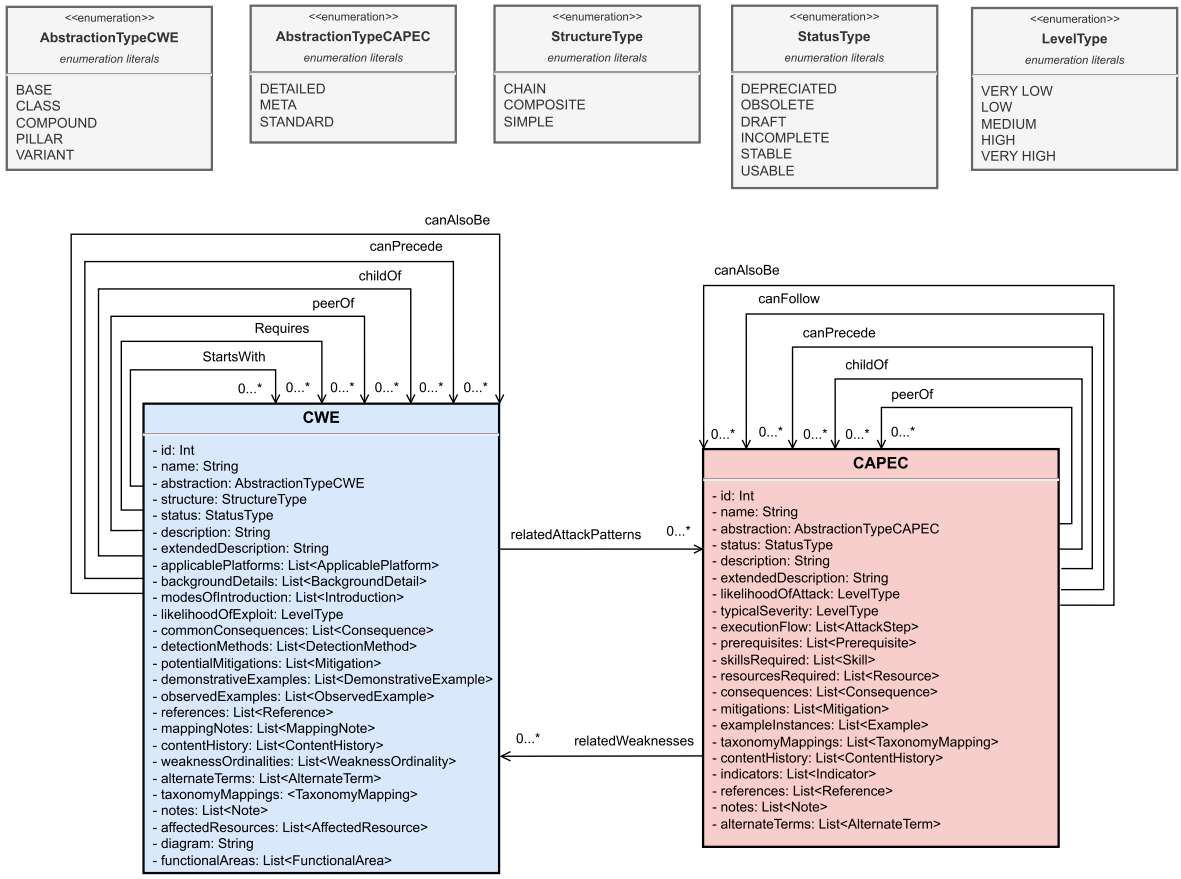
A Figura 9 apresenta o metamodelo conceitual proposto através de um diagrama UML desenvolvido para representar de forma estruturada os atributos das bases de dados CAPEC e CWE, bem como os relacionamentos existentes entre elas. Esse modelo foi desenvolvido a partir da análise completa dos dados oficiais disponibilizados pela MITRE, refletindo tanto os relacionamentos internos de cada base quanto os vínculos semânticos entre padrões de ataque e fraquezas estruturais.

Neste modelo, destacam-se os relacionamentos bidirecionais entre as bases, como `relatedWeaknesses` e `relatedAttackPatterns`, que representam conexões semânticas diretas entre fraquezas (CWE) e padrões de ataque (CAPEC). Tais vínculos são essenciais para permitir a navegação entre os ataques e suas respectivas fraquezas associadas, oferecendo suporte à análise integrada de ameaças e vulnerabilidades. Além disso, observa-se a presença de múltiplos autorelacionamentos em ambas as bases, que descrevem a forma como os elementos internos de cada catálogo se conectam entre si. Entre os relacionamentos comuns às duas bases, destacam-se os vínculos `canAlsoBe`, `canPrecede`, `childOf` e `peerOf`, os quais representam, respectivamente, equivalência conceitual, relação de precedência lógica, hierarquia de generalização e vínculos laterais entre instâncias correlacionadas.

Há ainda relacionamentos exclusivos de cada base: no caso do CWE, destacam-se os vínculos `Requires` e `StartsWith`, que modelam dependências funcionais e ordens de introdução de fraquezas; enquanto no CAPEC observa-se o relacionamento `canFollow`, útil para representar sequências viáveis de execução de ataques. Esses autorelacionamentos

[i] Para acessar o modelo online, é preciso ter o draw.io instalado no Google Drive [Tutorial]

Figura 9 – Estrutura do CAPEC e CWE



Fonte: Desenvolvido pelo autor (2025)

possibilitam a construção de estruturas hierárquicas e encadeadas, permitindo modelar desde grupos de fraquezas que ocorrem em conjunto até cadeias complexas de exploração por agentes maliciosos. No caso do CAPEC, essas relações são especialmente úteis para mapear cadeias de ataque (*attack chains*) e variações táticas, reforçando a aplicabilidade prática do modelo em contextos de análise de ameaças avançadas.

Outro ponto relevante do modelo diz respeito aos tipos de abstração presentes em cada base, representados pelos atributos `AbstractionTypeCWE` e `AbstractionTypeCAPEC`. Embora ambos indiquem o nível de generalização ou especificidade de uma entrada na base, suas categorizações seguem lógicas distintas. No caso do CWE, os tipos de abstração incluem valores como `Base`, `Class`, `Compound`, `Pillar` e `Variant`, que representam desde categorias genéricas de fraquezas (como a `Class`) até instâncias específicas de vulnerabilidades (como a `Variant`). Essa taxonomia permite estruturar a base em diferentes níveis de granularidade, favorecendo tanto análises abstratas quanto investigações mais direcionadas para casos específicos. Já no CAPEC, o atributo `AbstractionTypeCAPEC` adota valores como `Standard`, `Meta` e `Detailed`. Enquanto as entradas `Standard` representam padrões de ataque comuns e reutilizáveis, as `Detailed` fornecem descrições aprofundadas

de técnicas específicas; as entradas do tipo `Meta`, por ouro lado, agrupam múltiplos ataques sob um conceito comum, funcionando como categorias agregadoras. Essa estrutura visa apoiar diferentes níveis de análise, desde o planejamento estratégico até a simulação tática de ataques por agentes maliciosos.

O diagrama também inclui os tipos de dados enumerados utilizados por ambas as bases, como `StatusType` e `LevelType`, reforçando a importância da padronização e da consistência semântica nos campos. Em UML, um *enumeration* representa um tipo de dado definido por um conjunto fixo de valores possíveis, chamados de *enumeration literals*. Esses valores são utilizados para restringir e padronizar o conteúdo de atributos que assumem apenas determinados estados ou classificações pré-definidas por exemplo, os níveis de severidade ou status de uma entrada no banco de dados. A presença dessas classes no modelo conceitual contribui para tornar o esquema mais preciso, legível e fundamentado entre diferentes domínios de análise. Mais detalhes sobre os atributos específicos de cada base, bem como exemplos práticos de sua utilização, podem ser encontrados no Apêndice A.

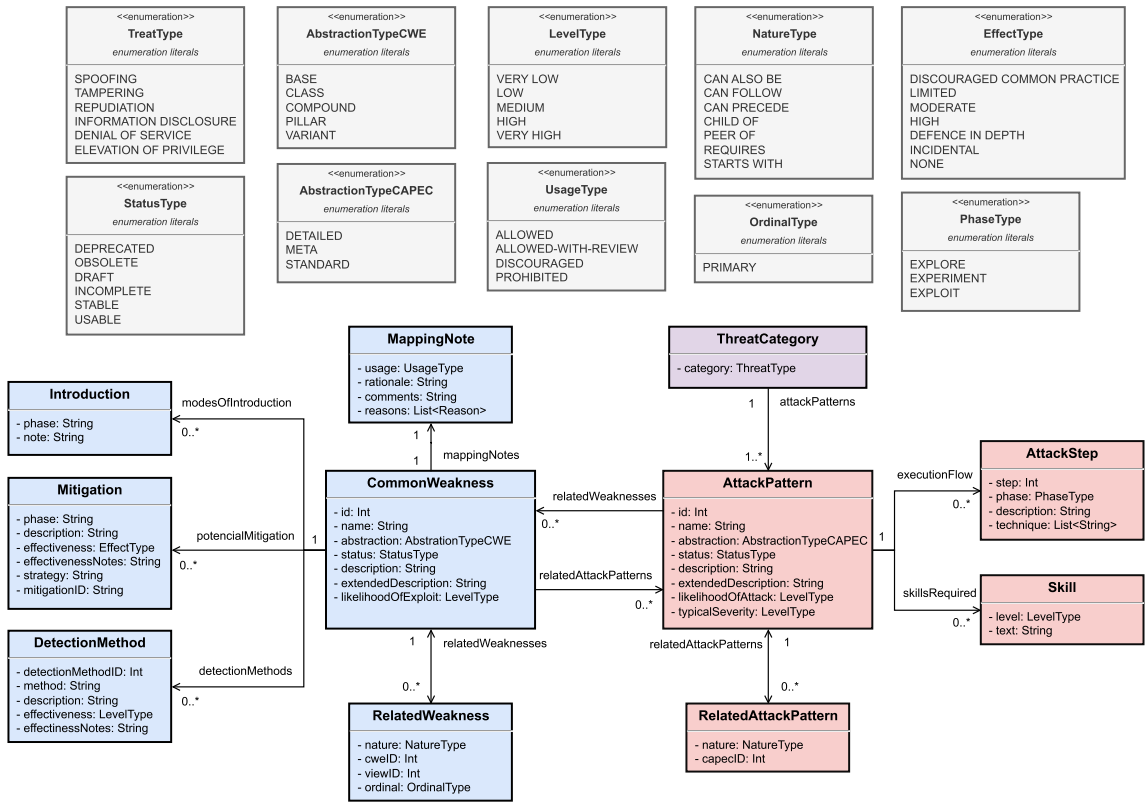
3.1.1 Modelo relacionando o STRIDE com as bases CAPEC e CWE

A partir da análise das informações fornecidas pelas bases mantidas pela MITRE, foi desenvolvido um modelo de dados com o objetivo de representar de forma estruturada os relacionamentos entre o modelo de ameaças STRIDE, e as bases de segurança CAPEC e CWE. A Figura 10 ilustra esse modelo usando UML, no qual também são destacados os principais atributos utilizados posteriormente para a determinação do valor de risco, conforme as diretrizes da ISO/SAE 21434. Esses atributos foram fundamentais para estabelecer a correspondência entre os elementos das bases e as etapas do processo TARA, permitindo a construção de uma estrutura integrada que oferece suporte técnico à identificação, avaliação e tratamento de riscos cibernéticos em sistemas automotivos.

No modelo acima proposto, o STRIDE está representado por meio da entidade `ThreatCategory`, que atua como ponto de partida para a identificação das ameaças. Cada categoria do STRIDE é associada a um conjunto de padrões de ataque CAPEC, os quais, por sua vez, possuem ligações com fraquezas estruturais descritas na CWE. Essa ligação direta entre STRIDE e CAPEC fornece a base conceitual para a construção de cenários e caminhos de ataque no processo de avaliação de risco; já a conexão entre CAPEC e CWE permite detalhar a exploração técnica de cada ameaça identificada, além de indicar fraquezas subjacentes e possíveis estratégias de mitigação.

A estrutura apresentada reflete não apenas os vínculos diretos entre as entidades centrais das três bases, mas também seus principais atributos internos utilizados para o construção do modelo. No caso do CAPEC, destacam-se os elementos como `executionFlow`, `skillsRequired`, `typicalSeverity` e `likelihoodOfAttack` que fornecem subsídios relevantes para a análise de viabilidade, impacto e caminhos

Figura 10 – Relacionamento entre STRIDE, CAPEC e CWE



Fonte: Desenvolvido pelo autor (2025)

de ataque durante o processo TARA. Já no CWE, os campos `modesOfIntroduction`, `detectionMethods` e `potentialMitigations` oferecem suporte direto à etapa de tratamento de risco, proposta pela ISO/SAE 21434.

No CAPEC, a entidade `executionFlow` representa a sequência típica de passos que um atacante segue para explorar uma vulnerabilidade, enquanto `skillsRequired` expressa o grau de especialização necessário para conduzir o ataque; os campos `typicalSeverity` e `likelihoodOfAttack` estimam qualitativamente o impacto potencial e a probabilidade de sucesso da ação maliciosa, respectivamente. No CWE, a entidade `modesOfIntroduction` indica as fases do ciclo de vida do *software* onde as falhas tendem a ser introduzidas; `detectionMethods` sugere técnicas de identificação; e `potentialMitigations` aponta medidas para neutralizar ou reduzir o risco. A entidade `MappingNote` complementa essas informações ao registrar justificativas, observações e restrições quanto à aplicabilidade do mapeamento das fraquezas. Detalhes completos de todos os atributos podem ser consultados no Apêndice A.

A partir do entendimento das associações semânticas entre os elementos das bases e a organização estrutural dos dados disponibilizados, é possível elaborar um modelo capaz de explorar as conexões entre essas bases com foco na análise de risco cibernético para

sistemas automotivos, em conformidade com os princípios da norma ISO/SAE 21434. O modelo proposto visa apoiar a identificação, avaliação e mitigação de riscos de segurança durante o ciclo de vida do sistema.

3.2 BANCO DE DADOS

Com o objetivo de viabilizar a integração entre diferentes fontes de informação e facilitar consultas estruturadas, este trabalho elaborou um banco de dados orientado a documentos, através do NoSQL, usando o *softwar*e MongoDB [65]. A escolha por uma base NoSQL justificou-se pela flexibilidade na manipulação de documentos complexos e pela capacidade de integrar, de forma eficiente, informações semiestruturadas oriundas de diferentes fontes. Essa característica foi particularmente importante para lidar com a estrutura rica e hierárquica dos catálogos CAPEC e CWE.

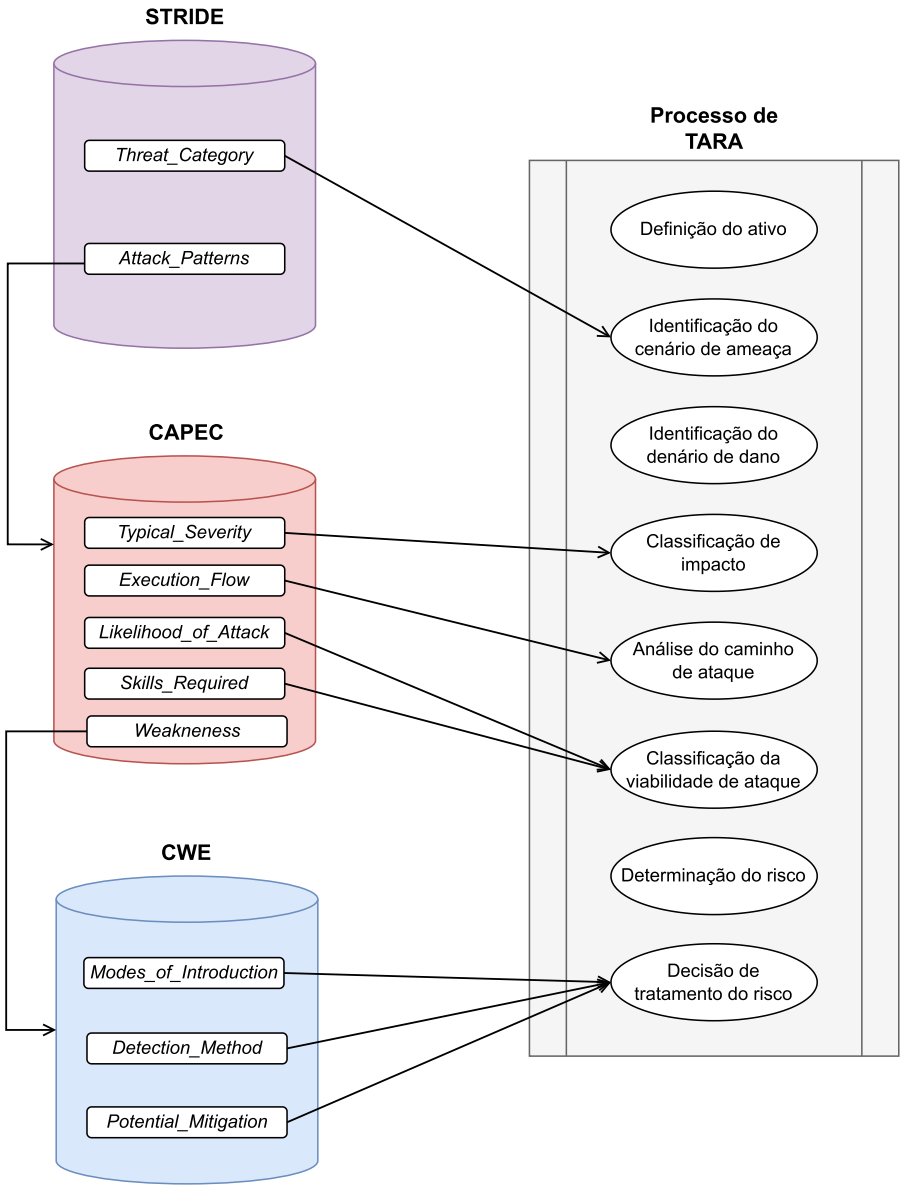
As informações dos catálogos CAPEC e CWE foram extraídas diretamente da base oficial mantida pela MITRE, em formato XML, garantindo o uso de versões atualizadas e confiáveis. Para a inserção dos dados no MongoDB, foi realizada a conversão dos arquivos para o formato JSON por meio de *scripts* desenvolvidos em Python. Esse processo envolveu o mapeamento de atributos, a padronização de campos e a organização dos relacionamentos entre entidades. Com o objetivo de promover a reprodutibilidade dos resultados e facilitar futuras extensões do trabalho, os códigos utilizados para essa etapa foram organizados e disponibilizados em um repositório público no GitHub, disponível em: https://github.com/

Além dos catálogos da MITRE, foi incorporado ao banco de dados um terceiro conjunto de informações: o mapeamento entre padrões de ataque CAPEC e categorias do modelo STRIDE. Esse mapeamento, disponível em formato JSON, foi obtido do projeto *CAPEC-STRIDE Mapping* [66], o qual associa sistematicamente padrões de ataque a categorias de ameaça. A integração entre esse modelo de ameaças com os dados estruturados permitiu uma análise mais alinhada e coerente com os requisitos da ISO/SAE 21434. A representação completa dessa associação, incluindo os diagramas para cada categoria de ameaça, podem ser encontrados no Anexo B.

A Figura 11 ilustra a correspondência do banco de dados com as fases do processo TARA definido na ISO 21434, destacando como as entidades contribuem para a construção do modelo. Cada elemento das bases foi mapeado para apoiar uma ou mais etapas do processo, desde a identificação da ameaça até a definição das estratégias de tratamento de risco. A estrutura relacional adotada baseia-se no modelo de dados definido previamente na Figura 10.

A modelagem do banco foi projetada para refletir as interações entre os atributos das bases e o processo da norma, estabelecendo vínculos claros entre os dados técnicos extraídos e atividades normativas, como identificação de ameaças, análise de caminhos de ataque, avaliação de impacto e definição de contramedidas. Cada conjunto de atributos

Figura 11 – Relacionamento entre o banco de dados e o processo TARA da ISO 21434



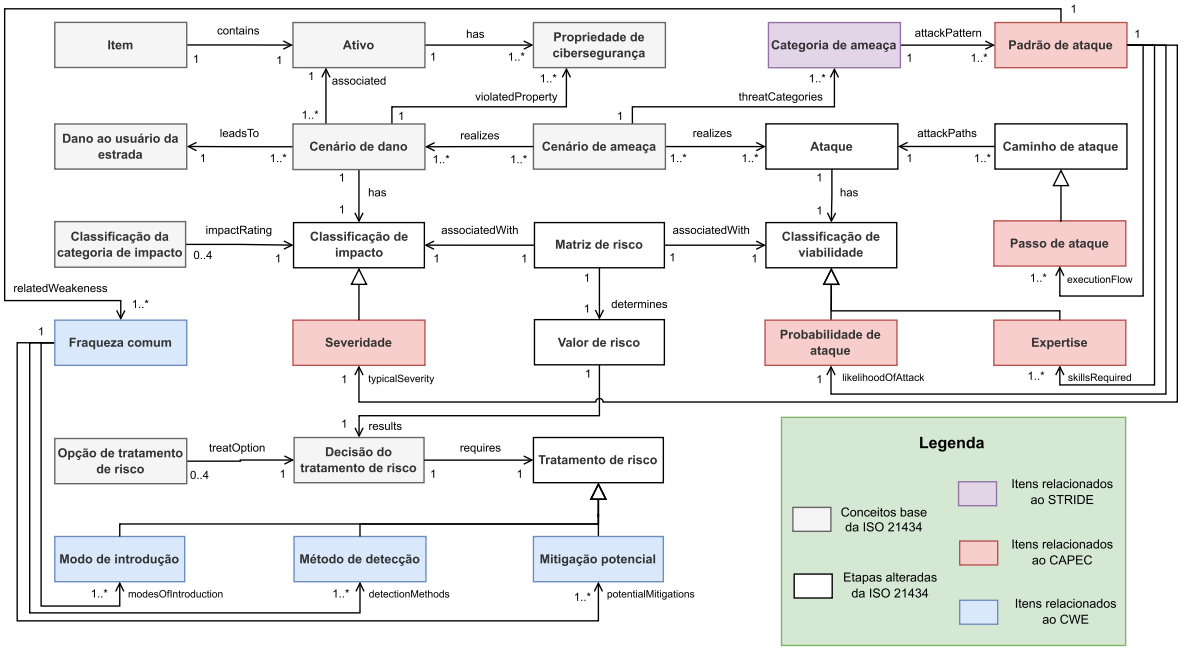
Fonte: Desenvolvido pelo autor (2025)

como categorias de ameaça (STRIDE), severidade e habilidades requeridas (CAPEC), ou modos de introdução e mitigações potenciais (CWE) foi mapeado a um ponto específico do fluxo TARA, promovendo rastreabilidade e coesão entre dados e o processo. Essa estrutura não apenas facilita a automação parcial da análise de risco, mas também permite uma navegação direta entre os elementos, viabilizando consultas avançadas com foco técnico e normativo. Mais detalhes sobre a aquisição e transformação dos dados, infraestrutura utilizada no banco, e escolha dos atributos podem ser encontrados no Apêndice B.

3.3 INTEGRAÇÃO COM A ISO 21434

A fim de organizar e estruturar a análise de ameaças e avaliação de riscos em sistemas automotivos autônomos, foi desenvolvida uma ontologia que estabelece relações entre os principais elementos de segurança definidos pela ISO/SAE 21434 e os catálogos STRIDE, CAPEC e CWE. Esse modelo, ilustrado na Figura 12, proporciona uma abordagem sistemática para a identificação de ameaças, avaliação de ataques e determinação do valor de risco, promovendo maior consistência e rastreabilidade ao processo TARA.

Figura 12 – Ontologia integrando a ISO TARA 21434 com STRIDE-CAPEC-CWE



Fonte: Desenvolvido pelo autor (2025)

O modelo preserva as etapas iniciais propostas pela ISO 21434, começando pela definição dos itens e ativos, seguido da identificação dos cenários de ameaça e de dano. A partir daí, introduz-se a noção de **ataques**, que se materializam a partir dos cenários de ameaça e se desdobram em **caminhos de ataque**. Cada caminho é composto por uma sequência de **passos de ataque**, modelados a partir do fluxo de execução descrito nos padrões CAPEC, permitindo representar com precisão a lógica de exploração das vulnerabilidades.

A ocorrência de cada ataque é analisada por meio da classificação de viabilidade, que considera dois atributos centrais: a **probabilidade de ocorrência do ataque** e a **habilidade necessária** (expertise). Essas entidades, oriundas do CAPEC, permitem estimar o esforço requerido para executar um ataque, servindo como apoio para a determinação da matriz de risco.

Além disso, o modelo trata da avaliação de impactos causados por ataques bem-

sucedidos. Os cenários de dano são associados aos possíveis efeitos sobre os ocupantes do veículo e terceiros, categorizados conforme a classificação de impacto. Essa classificação é determinada com base na **severidade típica** dos padrões de ataque relacionados, conforme indicado no CAPEC. Dessa forma, o modelo permite estimar o potencial prejuízo e integrá-lo diretamente à matriz de risco, resultando na atribuição do valor de risco através da determinação conjunta da classificação de viabilidade estimada anteriormente.

Por fim, o modelo realiza o tratamento de risco, etapa pouco detalhada nas diretrizes da ISO 21434. Essa etapa é caracterizada através do **modo de introdução**, **método de detecção** e **mitigações**, atributos presentes no CWE. Tais atributos oferecem suporte técnico direto à tomada de decisão no processo de tratamento de risco, fornecendo uma base empírica para justificar ações de engenharia de segurança. Ao incorporar essas informações, o modelo amplia a rastreabilidade entre as ameaças identificadas e as contramedidas adotadas, promovendo uma abordagem mais técnica, estruturada e transparente ao processo de mitigação, em conformidade com os princípios da ISO 21434.

Essa integração entre elementos normativos e bases de dados técnicas permite uma análise mais robusta e contextualizada, conectando a modelagem conceitual da ISO 21434 com fontes concretas de padrões de ataque, fraquezas e ameaças. O resultado é uma estrutura que favorece a aplicação prática do TARA em sistemas automotivos, com maior profundidade e precisão na análise dos riscos cibernéticos.

4 AVALIAÇÃO DO MODELO DA ISO

4.1 ESTUDO DE CASO

Para avaliar a viabilidade do modelo proposto, foram utilizados dois exemplos extraídos do Anexo H da ISO 21434, que abordam vulnerabilidades associadas à comunicação via Bluetooth em um sistema de controle de farol. Esses exemplos representam dois cenários distintos: ataques de falsificação de sinais e negação de serviço. O sistema de farol é considerado um sistema crítico, uma vez que falhas podem resultar em acidentes graves, especialmente durante a condução noturna. A escolha pela tecnologia Bluetooth como meio de acesso deve-se à sua função como um dos principais canais de comunicação entre os componentes do sistema, o que o torna especialmente relevante para a análise de ameaças em ambientes ciberfísicos. Nesse contexto, a presença de conectividade sem fio amplia significativamente a superfície de exposição a ataques, introduzindo novos caminhos para exploração de vulnerabilidades.

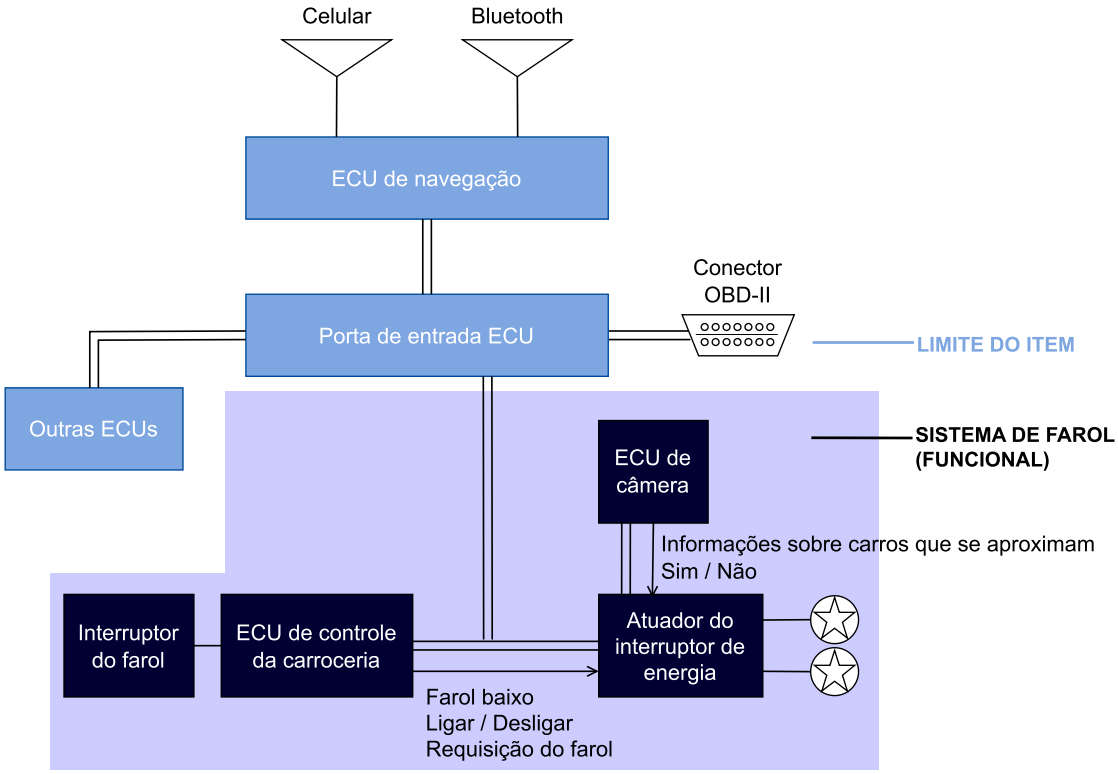
4.1.1 Sistema de farol

A seguir, é apresentada uma visão conceitual e simplificada de um sistema de farol de um veículo autônomo, onde sua arquitetura preliminar é representada na Figura 13. Essa representação destaca os limites do item, bem como seus principais componentes e sua interação com o sistema operacional, servindo como base para a análise das ameaças nos cenários selecionados.

Um sistema de farol é responsável por ligar/desligar o farol de acordo com a demanda do motorista. Esse sistema também possui dois modos de luz quando estiver ligado: farol alto e farol baixo. Se a luz estiver no modo de farol alto, o sistema muda automaticamente para o modo de farol baixo quando um veículo que se aproxima pela via oposta é detectado pela câmera. Ele retorna automaticamente para o modo de farol alto se mais nenhum outro veículo for detectado.

A ECU de controle da carroceria transmite mensagens ao atuador do interruptor de energia por meio do barramento CAN, enviando comandos para ativação e desativação dos faróis. Para alternar automaticamente entre os modos alto e baixo do farol, uma ECU de câmera é integrada ao atuador do interruptor de energia, possibilitando a detecção de veículos que se aproximam. O barramento CAN é conectado a uma porta de entrada ECU, responsável por regular o acesso de outras ECUs externas ao sistema de farol, como a ECU de navegação. A ECU de navegação dispõe de duas interfaces de comunicação: celular e Bluetooth. Além disso, a porta de entrada ECU está interligada a um conector OBD-II por meio de um barramento CAN distinto. O módulo de diagnóstico OBD-II coleta e monitora dados do veículo, incluindo velocidade. As interfaces Bluetooth, celular e OBD-II encontram-se fora do perímetro do item, podendo servir como possíveis vetores

Figura 13 – Arquitetura preliminar do sistema de farol



Fonte: Adaptado de Dantas et al. [16]

de ataque ao sistema de farol, conforme discutido a seguir.

4.1.2 Exemplo de aplicação do processo TARA

Após a definição do item e de seu ambiente operacional, aplica-se o processo TARA para a avaliação de risco. Para a apresentação deste trabalho, o método será realizado da seguinte forma: identificação dos cenários de dano; identificação dos cenários de ameaça; análise do caminho de ataque; classificação da viabilidade de ataque; classificação de impacto; determinação do valor de risco; e decisão do tratamento de risco.

O Quadro 1 apresenta a identificação de dois ativos cibernéticos envolvendo o funcionamento do sistema de iluminação veicular: a comunicação de dados responsável pela solicitação da lâmpada e a comunicação de dados com informações sobre veículos que se aproximam. Esses ativos são avaliados quanto às propriedades de cibersegurança que podem ser comprometidas sendo: confidencialidade (C), integridade (I) e disponibilidade (A). No primeiro caso, a violação da integridade de solicitação da lâmpada pode resultar no desligamento inesperado dos faróis em uma situação de condução noturna, o que pode levar a uma colisão frontal com objetos fixos, como árvores ou postes. Já para o segundo ativo, a indisponibilidade das informações impede a ativação do sistema que alimenta o acionamento do farol alto automático, fazendo com que o veículo mantenha o farol

baixo mesmo em condições onde o alto seria necessário, comprometendo a visibilidade do condutor e aumentando o risco de acidentes.

Quadro 1 – Lista de ativos e cenários de dano

Ativo	**Prop. Cib.**			**Cenário de dano**
	C	**I**	**A**	
Comunicação de dados (solicitação de lâmpada)	–	X	–	Colisão frontal com um objeto estreito e fixo (por exemplo, uma árvore) causada pelo desligamento não intencional do farol durante direção noturna em velocidade média
Comunicação de dados (informações de carros que se aproximam)	–	–	X	Mau funcionamento do farol alto automático causado pelo farol sempre permanecer no modo baixo durante a condução noturna

Fonte: Adaptado de ISO [8]

O Quadro 2 expande essa análise ao descrever os cenários de ameaça correspondentes, conectando os danos potenciais às suas possíveis causas. Para o primeiro caso, relacionado à colisão frontal provocada pelo desligamento inesperado dos faróis, a falsificação de sinais representa o cenário de ameaça identificado. Essa ameaça compromete a integridade da comunicação de dados enviada ao atuador do interruptor de energia, resultando na desativação não intencional dos faróis. Já no segundo caso, em que ocorre o mau funcionamento do sistema de farol alto automático, o cenário de ameaça é caracterizado por uma negação de serviço. Nesse contexto, a interrupção do recebimento de informações sobre veículos que se aproximam impede que a funcionalidade automática de ligar o farol no modo alto seja acionada adequadamente, afetando diretamente a visibilidade e a segurança da condução noturna.

Quadro 2 – Cenários de ameaça associados ao dano

Cenário de dano	**Cenário de ameaça**
Colisão frontal com [...]	Falsificação de um sinal leva à perda de integridade da comunicação de dados do sinal da requisição da lâmpada para o atuador do interruptor de energia, fazendo com que o farol desligue involuntariamente
Mau funcionamento [...]	Negação de serviço de informações sobre carros que se aproximam

Fonte: Adaptado de ISO [8]

Por fim, o Quadro 3 apresenta os caminhos de ataque associados a cada cenário de ameaça, revelando a sequência de passos necessários para realizar um ataque bem-sucedido, explorando interfaces Bluetooth como vetor de entrada. No cenário de falsificação, o atacante compromete inicialmente a ECU de navegação por meio da interface Bluetooth,

passando a emitir sinais maliciosos que são encaminhados ao interruptor do atuador, resultando na falsificação do comando de solicitação da lâmpada. Já no cenário de negação de serviço, o invasor utiliza um dongle OBD habilitado com Bluetooth para comprometer o *smartphone* do motorista injetando um volume excessivo de mensagens na rede veicular, sobrecarregando assim a ECU responsável pela entrada de dados e impedindo o funcionamento adequado da comunicação entre sensores e atuadores.

Quadro 3 – Caminhos de ataque para cenários de ameaça

Cenário de ameaça	**Caminho de ataque**
Falsificação	i. O invasor compromete a ECU de navegação a partir da interface Bluetooth ii. A ECU de navegação comprometida transmite sinais de controle maliciosos iii. A porta de entrada ECU encaminha sinais maliciosos para o atuador iv. Sinais maliciosos falsificam a solicitação da lâmpada (OFF)
DoS	i. O invasor conecta um dongle OBD habilitado para Bluetooth ao conector OBD quando o veículo está estacionado e destrancado ii. O invasor compromete o smartphone do motorista com interface Bluetooth iii. Envia mensagens via smartphone e dongle Bluetooth para a porta de entrada ECU iv. A porta de entrada ECU encaminha sinais maliciosos para o atuador v. O invasor inunda o barramento de comunicação com um grande número de mensagens

Fonte: Adaptado de ISO [8]

O Quadro 4 apresenta a classificação da viabilidade de ataque por duas abordagens: baseada em vetor de ataque (*attack vector-based approach*) para a falsificação de sinais, e em potencial de ataque (*attack potential-based approach*) para o ataque de DoS. A primeira abordagem considera o quão remoto (lógica ou fisicamente) um invasor pode estar para realizar um ataque quanto mais distante, maior a viabilidade atribuída. Já o potencial de ataque, conforme definido pela ISO/IEC 18045 [51], mede o esforço necessário para realizar um ataque, com base em cinco fatores: tempo, especialização, conhecimento do item, janela de oportunidade e equipamento. Neste trabalho, apresenta-se apenas o resultado final do mapeamento desses parâmetros. Para manter compatibilidade com os termos técnicos utilizados nas normas e catálogos internacionais, as classificações qualitativas foram mantidas no idioma original (inglês).

Quadro 4 – Classificação da viabilidade de ataque

Caminho de ataque	**Class. viab. de ataque**
Sinais maliciosos falsificam a solicitação do farol	Medium
O invasor inunda o barramento de comunicação	Low

Fonte: Adaptado de ISO [8]

O Quadro 5 apresenta a classificação de impacto nas seguintes categorias: segurança (*safety*), financeiro, operacional e privacidade. Os critérios para a categoria de *safety* seguem a ISO 26262-3:2018 [50]; e relacionados à privacidade, como dados pessoais (*Personally Identifiable Information* - PII), são definidos conforme a ISO/IEC 29100 [67]. Para este trabalho, apresenta-se apenas o resultado final das categorias e suas respectivas classificações.

Quadro 5 – Classificação de impacto

Cenário de dano	**Cat. impacto**	**Class. impacto**
Colisão frontal com [...]	S	Severe (S3)
Mal funcionamento [...]	O	Moderate

Fonte: Adaptado de ISO [8]

Por fim, a matriz de risco é definida a partir dos resultados de viabilidade de ataque (Quadro 4) e impacto (Quadro 5) obtidos, sendo apresentada no Quadro 6. A interseção entre essas dimensões resulta em um valor de risco que pode assumir valores de 1 a 5, sendo que valores mais altos indicam ameaças mais críticas.

Quadro 6 – Matriz de risco

		Class. viab. de ataque			
		Very low	Low	Medium	High
Class. impacto	Severe	2	3	4	5
	Major	1	2	3	4
	Moderate	1	2	2	3
	Negligible	1	1	1	1

Fonte: Adaptado de ISO [8]

O Quadro 7 consolida os resultados obtidos para a determinação do valor de risco ao sistema de farol veicular analisado, sintetizando as abordagens realizadas nas etapas anteriores. Para cada cenário de ameaça identificado, foram atribuídas classificações de viabilidade de ataque e impacto, com base nos possíveis cenários de dano e caminhos de ataque correspondentes. A partir da matriz de risco apresentada no Quadro 6, foi possível determinar os valores de risco: para o cenário de falsificação, a combinação de um impacto

classificado como *Severe* e uma viabilidade de ataque *Medium* resultou em um risco de nível 4, associado à categoria de *safety* (S); já o ataque de DoS, o impacto *Moderate* e viabilidade *Low* resultou em um risco de nível 2, relacionado à categoria operacional (O).

Quadro 7 – Determinação dos valores de risco

Cen. de ameaça	**Class. viab. de ataque**	**Class. impacto**	**Valor de risco**
Falsificação	Medium	Severe	S : 4
DoS	Low	Moderate	O : 2

Fonte: Adaptado de ISO [8]

O Quadro 8 sintetiza o processo de decisão para o tratamento de risco, conforme definido na Seção 15.9 da ISO/SAE 21434. Nele, são apresentadas as possíveis opções de tratamento aplicadas a diferentes cenários de ameaça, considerando seus respectivos valores de risco. As alternativas de resposta ao risco incluem: evitar, reduzir, compartilhar ou reter. A escolha das opções de decisão de tratamento de risco são aplicadas de acordo com o nível atribuído ao cenário avaliado, refletindo a estratégia adotada para assegurar a eliminação, transferência ou controle adequado das ameaças identificadas.

Quadro 8 – Decisão de tratamento de risco

Cen. de ameaça	**Valor de risco**	**Opção de tratamento de risco**
Falsificação	S : 4	Reduzir o risco
DoS	O : 2	Reduzir o risco

Fonte: Adaptado de ISO [8]

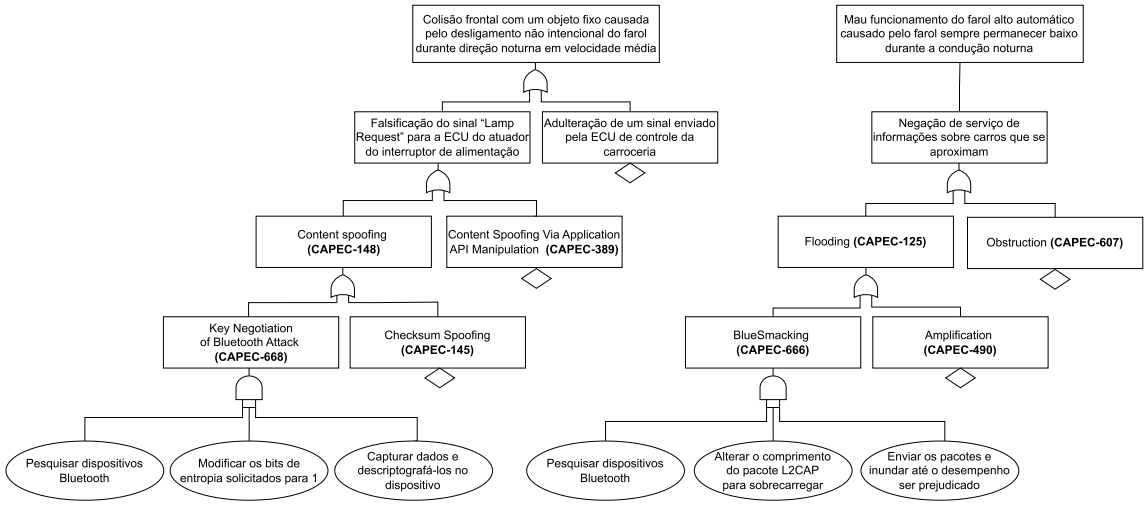
O exemplo apresentado demonstrou, de forma prática, a aplicação do processo TARA conforme definido na ISO/SAE 21434, abrangendo todas as suas etapas da identificação de ameaças até a definição das estratégias de tratamento de risco. Detalhes adicionais sobre os critérios utilizados na norma para a classificação de impacto, estimativa da viabilidade de ataque e decisão de tratamento de risco, podem ser encontrados no Anexo A.

5 AVALIAÇÃO DO MODELO PROPOSTO

Para avaliar a efetividade do modelo proposto, foi conduzida uma análise utilizando o exemplo do sistema de controle de farol apresentado anteriormente. O processo foi conduzido com base na estrutura de relacionamento definida entre as bases e a ISO, iniciando-se pela aplicação da metodologia STRIDE para categorização das ameaças. Em seguida, foram identificadas as conexões entre os caminhos de ataque definidos no exemplo analisado e os padrões CAPEC correspondentes, permitindo o mapeamento direto entre as categorias de ameaça e seus respectivos métodos de exploração.

Diante do contexto do sistema avaliado, foram identificadas ameaças que comprometem a integridade e a disponibilidade da comunicação sem fio via Bluetooth. Essas ameaças foram classificadas em duas categorias do STRIDE: falsificação e negação de serviço. A partir dessa categorização, foram mapeados padrões de ataque CAPEC que exploram o Bluetooth como meio de acesso. A Figura 14 apresenta a árvore de ataque construída com base nos caminhos de exploração identificados.

Figura 14 – Árvore de ataque definida através do catálogo CAPEC



Fonte: Desenvolvido pelo autor (2025)

A árvore de ataque é organizada de forma hierárquica, partindo dos cenários de dano observados no sistema até alcançar os ataques específicos e suas ações práticas de execução. O nó raiz da árvore representa os efeitos indesejados no veículo (cenários de dano), enquanto os nós intermediários descrevem os cenários de ameaça associados, conectando-se diretamente aos padrões de ataque CAPEC, sendo mapeados através do modelo STRIDE. Esses padrões são então decompostos em subnós que detalham técnicas específicas (CAPEC associados) e, por fim, ações técnicas que o atacante executaria na prática para viabilizar a exploração (os passos de ataque do registro, definidos no atributo `executionFlow`).

Considerando os cenários de dano e ameaça definidos previamente no Capítulo anterior (Seção **4.1.2**), foram selecionados os padrões de ataque mais relevantes com base no mapeamento realizado entre as bases. Entre os ataques que realizam a falsificação de sinais, destaca-se o CAPEC-668 (*Key Negotiation of Bluetooth Attack* - KNOB), que explora a negociação de chaves no protocolo Bluetooth, permitindo que um invasor enfraqueça a criptografia e se passe por um dispositivo legítimo. Já no caso de negação de serviço, foi identificado o CAPEC-666 (*BlueSmacking*), no qual a sobrecarga da pilha Bluetooth causa indisponibilidade do sistema. Esses padrões foram identificados por meio de consultas direcionadas à base CAPEC, utilizando termos-chave como "*Spoofing*", "*Flooding*" e "Bluetooth"; através da análise dos relacionamentos formais entre os padrões de ataque e as categorias de ameaça do modelo STRIDE.

A estrutura lógica da árvore permite visualizar como diferentes padrões CAPEC se encadeiam até resultar na materialização de um dano concreto. Além disso, o uso da CAPEC na construção da árvore fornece granularidade tática e técnica, enquanto a relação com o STRIDE assegura uma análise coerente em nível conceitual. A avaliação desses ataques foi realizada por meio da matriz de risco, onde foram analisados os atributos de viabilidade e impacto dos ataques conforme diretrizes da ISO 21434.

5.1 DEFINIÇÃO DA MATRIZ DE RISCO

Com base nos padrões CAPEC selecionados nas árvores de ataque, é possível calcular o valor de risco associado a cada ameaça identificada, em conformidade com as diretrizes da ISO/SAE 21434. Esse valor é definido pela combinação entre a viabilidade de ataque e a classificação de impacto, conforme o processo elaborado pela norma. Apesar da ISO não detalhar um método específico para atribuição desses parâmetros, este trabalho utiliza os atributos qualitativos disponíveis no CAPEC para estimar essas classificações de forma sistemática, viabilizando a construção da matriz de risco.

A classificação da viabilidade de ataque para o modelo proposto é determinada com base em dois atributos definidos na taxonomia CAPEC: `Likelihood_of_Attack` e `Skills_Required`. Ambos os atributos são categorizados em três níveis baixo (*low*), médio (*medium*) e alto (*high*) aos quais são atribuídos valores numéricos (1 para *low*, 3 para *medium* e 5 para *high*). A união desses dois parâmetros é utilizada para estimar a classificação da viabilidade de ataque, calculada por meio da soma dos valores atribuídos a cada atributo. Essa abordagem é baseada em uma adaptação da metodologia proposta pela norma ISO/IEC 18045 [51], conforme apresentado no Quadro 9.

Quadro 9 – Mapeamento da classificação da viabilidade de ataque

Class. viab. de ataque	**Valores**
Unlikely	0 - 2
Possible	3 - 6
Likely	> 6

Fonte: Desenvolvido pelo autor (2025)

O método para calcular a classificação da viabilidade de ataque pode ser representado através da seguinte fórmula:

$$V = L + S \quad ; \quad \text{onde } L, S \in \{1, 3, 5\} : \begin{cases} 1 & = & \text{low} \\ 3 & = & \text{medium} \\ 5 & = & \text{high} \end{cases}$$

$$\text{Classificação da viabilidade de ataque } (A): \quad \begin{cases} \text{Unlikely} & \text{se} & 0 \leq V \leq 2 \\ \text{Possible} & \text{se} & 3 \leq V \leq 6 \\ \text{Likely} & \text{se} & V > 6 \end{cases} \tag{5.1}$$

onde o L representa o valor atribuído ao `Likelihood_of_Attack` e S representa o valor atribuído ao `Skills_Required`. A pontuação final V é então mapeada para categorias de viabilidade, conforme os intervalos definidos no Quadro 9. Essa categorização permite avaliar o quão factível é realizar um ataque, com base na severidade e probabilidade de exploração.

Após a definição da viabilidade de ataque, a classificação de impacto é determinada com base no atributo `Typical_Severity`, fornecido na base CAPEC. Esse atributo é categorizado em cinco níveis: *Very Low*, *Low*, *Medium*, *High* e *Very High*, representando uma escala crescente de severidade associada à exploração bem-sucedida do ataque. Embora suas categorias sejam semelhantes aos atributos utilizados para determinar a viabilidade de ataque no modelo, optou-se por manter os valores e o idioma original a fim de preservar a consistência com a taxonomia original do CAPEC.

Por fim, o valor de risco é obtido por meio do cruzamento entre a viabilidade de ataque e a classificação de impacto, conforme estruturado na matriz de risco apresentada no Quadro 10. Esse procedimento segue o princípio estabelecido pela ISO/SAE 21434, onde a interseção entre essas duas dimensões gera um valor numérico, variando de 1 a 5. Assim, a matriz de risco é definida a partir dos atributos presentes no CAPEC.

Quadro 10 – Definição da matriz de risco para o modelo proposto

		Class. viab. de ataque (A) `Likelihood_of_Attack` + `Skills_Required`		
		Unlikely	Possible	Likely
Class. impacto (I) `Typical_Severity`	Very High	3	4	5
	High	2	3	4
	Medium	2	2	3
	Low	1	2	2
	Very Low	1	1	1

Fonte: Desenvolvido pelo autor (2025)

O valor de risco também pode ser obtido através da seguinte fórmula:

$$R = 1 + (I \times A) \quad ; \quad R_{final} = \begin{cases} \lfloor R \rfloor, & \text{se } R - \lfloor R \rfloor \leq 0{,}5 \\ \lceil R \rceil, & \text{se } R - \lfloor R \rfloor > 0{,}5 \end{cases}$$

$$I: \begin{cases} 0.0 = \text{very low} \\ 0.5 = \text{low} \\ 1.0 = \text{medium} \\ 1.5 = \text{high} \\ 2.0 = \text{very high} \end{cases} \quad ; \quad A: \begin{cases} 1.0 = \text{unlikely} \\ 1.5 = \text{possible} \\ 2.0 = \text{likely} \end{cases} \tag{5.2}$$

onde o I representa o valor atribuído ao `Typical_Severity` e A representa o valor obtido através da Equação (5.1). Para fins de classificação, o valor calculado é então arredondado de acordo com a seguinte regra: se a parte decimal de R for menor ou igual a 0.5, considera-se o valor inferior (função piso, denotada por $\lfloor R \rfloor$[i]); caso contrário, o valor é arredondado para cima (função teto, denotada por $\lceil R \rceil$[ii]) [68]. Esse resultado é mapeado para um valor final entre 1 e 5, conforme ilustrado no Quadro 10.

Com a metodologia de cálculo de risco estabelecida, selecionam-se os padrões de ataque definidos previamente na árvore de ataques, sendo eles o CAPEC-668 e CAPEC-666. Esses padrões foram então submetidos ao modelo de avaliação proposto, utilizando a matriz de risco para classificar o impacto (severidade) e a viabilidade (probabilidade e expertise) dos ataques. O Quadro 11 apresenta os atributos dos CAPECs selecionados.

[i] Converte um número real no maior número inteiro menor ou igual ao valor original [Fonte]
[ii] Converte um número real no menor número inteiro maior ou igual ao valor original [Fonte]

Quadro 11 – Atributos relevantes dos CAPECs selecionados

Padrão de ataque (CAPEC)	**Severidade (Typical_Severity)**	**Prob. de ataque (Likelihood_of_Attack)**	**Expertise (Skills_Required)**
668: KNOB	High	Low	Medium
666: BlueSmacking	Medium	Medium	Low

Fonte: Desenvolvido pelo autor (2025)

Considerando os valores extraídos do CAPEC, é possível definir o valor de risco. A classificação de impacto é obtida diretamente da severidade, já a classificação da viabilidade de ataque é obtida através da probabilidade e expertise do atacante, onde para ambos os cenários de ameaça, foram definidos como *Possible* através do Quadro 9. Por fim, o Quadro 12 apresenta a determinação do valor de risco associado.

Quadro 12 – Determinação dos valores de risco para o modelo proposto

Cen. ameaça	**Class. viab. de ataque**	**Class. impacto**	**Valor de risco**
Falsificação	Possible	High	3
DoS	Possible	Medium	2

Fonte: Desenvolvido pelo autor (2025)

Para verificar a consistência e a aplicabilidade do modelo proposto, foi realizada uma comparação com os resultados fornecidos no Anexo H da ISO/SAE 21434. O Quadro 13 resume essa comparação, destacando os valores de risco obtidos do processo TARA para cada cenário de ameaça.

Quadro 13 – Comparação dos resultados entre a ISO e o modelo proposto

	Exemplo ISO		**Modelo proposto**	
	Falsificação	DoS	Falsificação	DoS
Class. impacto	Severe	Moderate	High	Medium
Viab. ataque	Medium	Low	Possible	Possible
Valor de risco	4	2	3	2

Fonte: Desenvolvido pelo autor (2025)

Para o ataque de falsificação de sinais via Bluetooth (CAPEC-668), o modelo proposto atribuiu um valor de risco igual a 3, enquanto a avaliação do exemplo da norma indicou um valor de 4. Essa diferença pode ser atribuída à forma como a classificação de impacto é definida na abordagem proposta, que considera apenas um atributo para a determinação de sua severidade. Já no segundo cenário, referente ao ataque de negação de serviço (CAPEC-666), os resultados foram idênticos nos dois métodos, com ambos os modelos classificando o risco como 2. Esse alinhamento reforça a consistência da abordagem

proposta e sua capacidade de manter compatibilidade com modelos já consolidados, ao mesmo tempo em que fornece um maior nível de detalhamento técnico, devido à utilização de bases de dados referentes a ataques.

Esses resultados indicam que, embora possam ocorrer pequenas variações nos resultados, o modelo proposto foi capaz de refletir adequadamente a severidade e a viabilidade dos ataques, mantendo aderência aos princípios normativos. Tal proposta contribui para uma abordagem aplicada ao contexto do sistema avaliado.

5.2 DECISÃO DO TRATAMENTO DE RISCO

Embora a ISO/SAE 21434 forneça diretrizes robustas para a avaliação de riscos cibernéticos no contexto automotivo, sua abordagem em relação ao tratamento de risco ainda se apresenta de forma limitada. Conforme descrito no Anexo H da norma, são listadas apenas quatro opções de resposta ao risco: evitar, reduzir, transferir ou aceitar. A norma não fornece orientações específicas sobre como implementar essas respostas, tampouco detalha estratégias ou metodologias para a seleção e aplicação de contramedidas.

Diante dessa lacuna, este trabalho propõe uma extensão ao processo de tratamento de risco definido na ISO, integrando a base de dados CWE como um recurso complementar para apoiar decisões técnicas durante essa etapa. A utilização do CWE permite identificar informações detalhadas que incluem: a fase do ciclo de vida em que a fraqueza pode ser introduzida, possíveis abordagens de mitigação, boas práticas de codificação e ferramentas de detecção recomendadas. Dessa forma, o CWE se apresenta como uma ponte entre a identificação da ameaça e a aplicação concreta de medidas preventivas e corretivas.

A abordagem adotada neste trabalho parte da associação entre os padrões de ataque CAPEC identificados e seus respectivos CWEs correlacionados. Por meio do banco de dados desenvolvido, foi possível estabelecer essas conexões, resultando na identificação das fraquezas relevantes para cada ameaça. Para o cenário de falsificação de sinais, descrito anteriormente com base no padrão CAPEC-668 (KNOB), foi identificado o CWE-732 (*Incorrect Permission Assignment for Critical Resource*) como uma fraqueza associada. Já para o cenário de negação de serviço, baseado no CAPEC-666 (*BlueSmacking*), foi mapeado o CWE-404 (*Improper Resource Shutdown or Release*) como principal fraqueza estrutural.

A fraqueza CWE-732 é caracterizada por uma má configuração de permissões, permitindo que usuários não autorizados acessem ou modifiquem recursos sensíveis. O CWE-404 refere-se à falha em encerrar ou liberar corretamente recursos críticos, como arquivos, memória ou conexões de rede, o que pode resultar em vazamentos de recursos, condições de corrida, ou negação de serviço. A seguir, é apresentada uma síntese das mitigações, modos de introdução e métodos de detecção relacionados às fraquezas definidas, permitindo que a equipe de segurança tome decisões sobre quais estratégias de tratamento

são mais adequadas ao contexto técnico e organizacional.

O Quadro 14 apresenta as estratégias de tratamento de risco através do CWE-732, que trata de configurações permissivas de controle de acesso. Essa fragilidade está fortemente ligada a decisões de arquitetura, design e instalação, refletindo um caráter estrutural que pode ser introduzido ainda nas fases iniciais do ciclo de vida do sistema. Do ponto de vista de detecção, o CWE-732 pode ser identificado tanto por ferramentas automatizadas de análise estática/dinâmica quanto por métodos mais específicos, como o uso de *scanners* de configuração (SOAR - *Security Orchestration, Automation and Response*) ou o monitoramento de chamadas ao sistema em testes de *black box*[iii]. Seu tratamento, portanto, exige uma abordagem mais ampla e integrada, que envolve desde práticas de projeto (como segmentação por níveis de privilégio) até a configuração adequada em ambientes de produção e nuvem. Destaca-se a relevância de mitigações voltadas ao contexto de instalação e operação, como a aplicação de políticas de segurança por padrão e o uso de *hardening*[iv] com base em guias como o USGCB[v] (*United States Government Configuration Baseline*). A eficácia das estratégias varia, sendo mais alta quando aplicadas durante a instalação e mais limitada quando dependem de mecanismos de isolamento, como o *sandboxing*[vi].

[iii] Software em que o funcionamento interno ou a estrutura do código não são conhecidos [Fonte]
[iv] Processo de reforçar um sistema para torná-lo mais resistente a ataques cibernéticos [Fonte]
[v] Agência com objetivo de criar configurações de segurança para produtos de TI federais [Fonte]
[vi] Prática de segurança na qual utiliza-se um ambiente isolado para testes [Fonte]

Quadro 14 – Estratégias de tratamento de risco através do CWE-732

Aspecto	**Descrição e Estratégias**
Modos de Introd.	• Arquitetura e Design • Implementação (suposições incorretas sobre ambiente) • Instalação (permissões flexíveis por padrão) • Operação
Mét. de Detecção	• **Automatizados:** Análise estática e dinâmica para detecção de problemas de permissão no sistema • **Manuais:** Testes de penetração, análise estática/dinâmica manual com foco em regras de negócios • ***Black-box*:** Monitoramento de chamadas ao sistema com ferramentas como *strace* e *Process Monitor* • **SOAR (parcial):** *Scanners* de configuração, *scanners* de aplicações web e banco de dados
Mitigações	• **Design:** Dividir o sistema por níveis de privilégio com controle granular sobre permissões (eficácia: moderada) • ***Sandboxing*:** Executar em ambientes isolados como chroot, AppArmor ou SELinux (eficácia: limitada). • **Instalação:** Configurar permissões restritivas por padrão (eficácia: alta). • **Operação:** Seguir guias de configuração seguros como o USGCB. • **Nuvem:** Desabilitar acesso público a dados sensíveis (*buckets S3*, *blobs*, etc.)

Fonte: Desenvolvido pelo autor (2025)

Em contrapartida, o CWE-404 (Quadro 15) diz respeito à liberação imprópria de recursos, caracterizando-se como uma falha de implementação. Diferentemente do CWE-732, que pode ser mitigado com diretrizes arquiteturais e de configuração, o CWE-404 demanda disciplina na codificação e na escolha de tecnologias que reduzam a exposição à falha. No que se refere à detecção, o CWE-404 pode ser identificado por meio de uma combinação de testes dinâmicos (como *fuzzing*[vii] e testes de estresse), simulações manuais de falhas, e principalmente por ferramentas de análise estática (SAST - *Static Application Security Testing*), que possuem alta eficácia ao modelar fluxos de dados que não são encerrados de forma correta. Por exemplo, linguagens que gerenciam automaticamente a

[vii] Técnica de teste de *software* que envia entradas aleatórias para o sistema em questão [Fonte]

memória (como Java e Ruby) podem mitigar a vulnerabilidade ainda na etapa de requisitos. Além disso, boas práticas de codificação como garantir a liberação de todos os recursos em todos os caminhos de execução e utilizar funções de desalocação compatíveis são essenciais para reduzir a ocorrência desse tipo de erro.

Quadro 15 – Estratégias de tratamento de risco através do CWE-404

Aspecto	**Descrição e Estratégias**
Modos de Introd.	• Implementação
Mét. de Detecção	• **Automatizados:** Utilização de *fuzzing*, testes de robustez e injeção de falhas com múltiplas entradas e cenários de estresse. Identificar vazamentos ou falhas em condições de carga elevada (eficácia: moderada) • **Manuais:** Execução controlada em cenários de erro não triviais (ex.: falta de memória, falhas de rede, interrupções prematuras) e observação de comportamentos inesperados, como exceções ou falhas silenciosas • **SAST:** Construção de modelos de fluxo de dados e controle para localizar padrões inseguros que conectam pontos de entrada a funções que manipulam recursos do sistema sem encerramento apropriado (eficácia: alta)
Mitigações	• **Requisitos:** Utilizar linguagens de programação que oferecem gerenciamento automático de recursos, como Java, Ruby ou Lisp, que realizam coleta de lixo (*garbage collection*) para evitar liberação manual incorreta • **Implementação:** Garantir a liberação de todos os recursos alocados em todos os caminhos de execução, incluindo em condições de erro. Utilizar funções pareadas corretamente (por exemplo, `malloc`/`free`, `new`/`delete`, `new[]`/`delete[]`). Ao desalocar estruturas complexas, liberar adequadamente todos os seus componentes

Fonte: Desenvolvido pelo autor (2025)

Ao comparar os dois casos, observa-se que, enquanto o CWE-732 exige ações coordenadas entre diferentes fases e atores do desenvolvimento (arquitetos, administradores de sistema, engenheiros de DevSecOps[viii]), o CWE-404 concentra-se mais no comportamento individual do código e na responsabilidade do desenvolvedor na gestão adequada dos recursos alocados. Ambos, no entanto, evidenciam a importância de estratégias integradas de mitigação e detecção, combinando boas práticas de engenharia, automação de testes e ferramentas de análise de segurança. Além disso, a existência de modos de introdução

[viii] Faz testes de segurança em todas as etapas do processo de desenvolvimento de software [Fonte]

distintos para cada fraqueza destaca a necessidade de ações específicas por fase do ciclo de vida do sistema.

Com base nas evidências fornecidas pelo CWE, é possível mapear medidas de resposta ao risco com maior precisão, indo além da simples aceitação ou decisão tomada. Por exemplo, o conhecimento de que o CWE-732 pode ser introduzido na fase de instalação justifica a adoção de políticas de *secure-by-default* desde o início do ciclo de vida do *software*, além do uso de *scanners* automatizados e práticas de auditoria contínua para detecção e correção de configurações permissivas. A partir desses CWEs, é possível explorar recomendações técnicas para mitigar riscos, como controle de permissões, boas práticas de design seguro e uso de ferramentas de análise estática para detecção precoce dessas vulnerabilidades. Com isso, busca-se não apenas estimar o risco residual, mas também construir uma base técnica sólida para seu tratamento, promovendo maior alinhamento entre a teoria normativa e a prática da engenharia segura de *software* automotivo.

Essa abordagem contribui não apenas para a mitigação de riscos, mas também para a criação de uma cultura de segurança mais madura e tecnicamente embasada dentro do processo de desenvolvimento. Assim, ao invés de depender apenas de estratégias genéricas de mitigação, é possível adotar um plano de resposta mais assertivo, incorporando técnicas específicas como uso de ferramentas de análise de memória, estruturas de liberação automática de recursos, e monitoramento em tempo de execução. Essa abordagem contribui significativamente para a redução do risco associado à fraqueza e, consequentemente, para o aumento da resiliência cibernética do sistema.

6 CONCLUSÃO

Este trabalho apresentou uma abordagem baseada em um modelo conceitual que integra o modelo de ameaças STRIDE aos catálogos estruturados CAPEC e CWE, com o objetivo de apoiar a análise de riscos em sistemas automotivos, conforme as diretrizes estabelecidas pela norma ISO/SAE 21434. Por meio da construção de um banco de dados orientado a documentos, e da modelagem das relações entre ativos, ameaças, padrões de ataque e fraquezas, foi possível representar de forma sistemática os caminhos de exploração de vulnerabilidades. Essa estrutura viabilizou a construção de árvores de ataque, a classificação de ameaças e a definição do valor de risco com base na viabilidade de ataque e no impacto estimado, além de fornecer suporte técnico à etapa de tratamento de risco uma das fases menos discutidas pela norma.

A integração do catálogo CAPEC contribui diretamente para a etapa de identificação e análise de risco, oferecendo descrições detalhadas dos mecanismos de ataque, suas condições de exploração e consequências típicas. Essa incorporação possibilitou uma representação mais precisa e contextualizada dos cenários de ameaça, aumentando a robustez da análise ao alinhar padrões conhecidos de ataque com os ativos críticos definidos na arquitetura do sistema. Complementando esse processo, a inclusão da base CWE teve papel fundamental na etapa de tratamento de risco. Ao contrário da ISO 21434, que se limita em apenas apresentar opções de resposta (evitar, reduzir, compartilhar ou aceitar), a base CWE oferece um conjunto de atributos técnicos sobre cada fraqueza, incluindo modos de introdução, possíveis mitigações e métodos de detecção. Essas informações permitem definir estratégias de resposta robustas, possibilitando a associação direta entre um ataque identificado e as contramedidas aplicáveis ao longo do ciclo de vida do sistema.

A aplicação prática da metodologia foi avaliada por meio de um estudo de caso que envolveu o protocolo Bluetooth em sistemas de iluminação automotiva. Os resultados obtidos evidenciaram que a abordagem proposta é eficaz na identificação e categorização de riscos cibernéticos, apresentando potencial de aplicação em diferentes contextos e arquiteturas. A associação entre STRIDE, CAPEC e CWE mostrou-se vantajosa não apenas pela abrangência analítica, mas também pela rastreabilidade que oferece ao processo decisório. Ao integrar o conhecimento consolidado de bases públicas mantidas pela MITRE a uma estrutura alinhada às diretrizes da ISO/SAE 21434, este trabalho contribui para fortalecer a sistematização do processo de análise de risco, promovendo maior consistência, transparência e aplicabilidade na engenharia de segurança cibernética de sistemas embarcados.

Adicionalmente, destaca-se que, durante o desenvolvimento desta dissertação, foi elaborado um artigo científico, com o intuito de apresentar e discutir parte dos resultados obtidos. O artigo submetido concentrou-se nos fundamentos teóricos e na estruturação

conceitual do modelo proposto, com ênfase na construção de uma ontologia e na integração entre as bases CAPEC, CWE e CVE. No entanto, identificou-se que a aplicação prática da base CVE apresenta desafios consideráveis, especialmente pelo grande número de atualizações em sua base; e pela dificuldade de estabelecer relações diretas e estáveis entre suas entradas e os elementos com as demais bases analisadas. Assim, optou-se por restringir o escopo deste trabalho ao uso das bases CAPEC e CWE, que oferecem maior estabilidade e rastreabilidade para fins de modelagem relacional, deixando a exploração prática do CVE como uma possível extensão futura do trabalho. Informações complementares sobre o artigo, incluindo sua versão completa, podem ser encontradas no Apêndice C.

A proposta deste trabalho não se restringe apenas ao setor automotivo, uma vez que as bases de dados utilizadas são amplamente reconhecidas na área de segurança cibernética. Ao combinar modelagem de ameaças, análise de padrões de ataque e estruturação de fraquezas, a abordagem pode ser adaptada a diversos setores que operam com sistemas ciberfísicos críticos, como redes de energia inteligente, sistemas de saúde digital, manufatura 4.0 e controle de tráfego aéreo [69, 70]. Tais domínios compartilham desafios relacionados à complexidade sistêmica, conectividade intensa e à necessidade de respostas autônomas e seguras em tempo real. A estrutura da integração entre STRIDE, CAPEC e CWE, aliada à flexibilidade do banco de dados desenvolvido, permite generalizar a metodologia para apoiar o processo de análise e mitigação de riscos cibernéticos em diferentes contextos operacionais. Assim, este trabalho contribui não apenas para o avanço da segurança veicular, mas também para o fortalecimento de práticas de segurança cibernética em sistemas críticos de infraestrutura.

6.1 TRABALHOS FUTUROS

Como trabalhos futuros, pretende-se expandir a integração do modelo com as bases de dados ATT&CK (*Adversarial Tactics, Techniques & Common Knowledge*) [71] e CVE (*Common Vulnerabilities and Exposures*) [72]. A inclusão do ATT&CK irá permitir o enriquecimento da análise comportamental dos agentes maliciosos, mapeando as táticas, técnicas e procedimentos utilizados em ataques reais, contribuindo para o aprimoramento da análise comportamental e a validação de defesas no domínio automotivo. Por sua vez, a incorporação do CVE possibilitará a associação direta com vulnerabilidades reais e documentadas, trazendo uma perspectiva mais atualizada e orientada à aplicação prática da segurança cibernética. Essa evolução contribuirá para um *framework* mais completo e eficaz na gestão proativa de ameaças.

Outra linha para trabalhos futuros envolve a aplicação de técnicas de aprendizado de máquina e processamento de linguagem natural (PLN) com o objetivo de enriquecer e automatizar o mapeamento entre as bases de dados. Estudos como o de Kanakogi et al. [60], que utilizam modelos de PLN para identificar relações semânticas entre entradas

do CVE e CAPEC, evidenciam o potencial dessas técnicas na descoberta de conexões implícitas não formalizadas nas bases originais. Além disso, iniciativas como o projeto BRON [73], que constroem grafos de conhecimento cibernético a partir de fontes de dados públicas, evidenciam o valor da combinação entre dados extraídos e curadoria especializada. A integração dessas abordagens pode ampliar a cobertura e a precisão dos relacionamentos entre ameaças, ataques e fraquezas, contribuindo para um modelo mais adaptável e completo para apoiar a análise de riscos cibernéticos.

Além das direções anteriormente mencionadas, uma extensão natural deste trabalho consiste na automação parcial ou total da etapa de avaliação e tratamento de risco, integrando os dados consolidados no banco de dados desenvolvido com técnicas de aprendizado de máquina e PLN. A partir das relações estruturadas entre ameaças, fraquezas e padrões de ataque, seria possível treinar modelos capazes de inferir automaticamente o valor de risco de um cenário, com base em atributos como severidade, viabilidade, histórico de vulnerabilidades semelhantes (CVE) e impacto estimado. Complementarmente, esses modelos poderiam sugerir estratégias de tratamento mais adequadas como mitigação técnica, aceitação ou transferência com base em padrões recorrentes e decisões tomadas em casos semelhantes. Essa abordagem permitiria não apenas maior escalabilidade na análise de sistemas complexos, mas também promoveria um suporte inteligente à tomada de decisão, alinhado aos princípios da engenharia de segurança estabelecidos pela ISO/SAE 21434.

REFERÊNCIAS

1 TORRES, J. M.; COMESAÑA, C. I.; GARCÍA-NIETO, P. J. Review: Machine learning techniques applied to cybersecurity. *International Journal of Machine Learning and Cybernetics*, Springer, v. 10, n. 10, p. 2823–2836, 2019.

2 MARTINEZ, J. et al. Security debt: Characteristics, product life-cycle integration and items. In: IEEE. *2021 IEEE/ACM International Conference on Technical Debt (TechDebt)*. [S.l.], 2021. p. 1–5.

3 PEKARIC, I. et al. A systematic review on security and safety of self-adaptive systems. *Journal of Systems and Software*, Elsevier, p. 111716, 2023.

4 BIRO, M. et al. Software safety and security risk mitigation in cyber-physical systems. *IEEE Software*, IEEE, v. 35, n. 1, p. 24–29, 2017.

5 MAPLE, C. et al. A connected and autonomous vehicle reference architecture for attack surface analysis. *Applied Sciences*, MDPI, v. 9, n. 23, p. 5101, 2019.

6 KHATOUN, R.; ZEADALLY, S. Smart cities: concepts, architectures, research opportunities. *Communications of the ACM*, ACM New York, NY, USA, v. 59, n. 8, p. 46–57, 2016.

7 LIMA, A. et al. Towards safe and secure autonomous and cooperative vehicle ecosystems. In: *Proceedings of the 2nd ACM workshop on cyber-physical systems security and privacy*. [S.l.: s.n.], 2016. p. 59–70.

8 STANDARD, I. *ISO/SAE 21434-Road Vehicles–Cybersecurity engineering*. 2021.

9 MACHER, G. et al. ISO/SAE DIS 21434 automotive cybersecurity standard-in a nutshell. In: SPRINGER. *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops: DECSoS 2020, DepDevOps 2020, USDAI 2020, and WAISE 2020, Lisbon, Portugal, September 15, 2020, Proceedings 39*. [S.l.], 2020. p. 123–135.

10 SCHNEIER, B. Attack trees. *Dr. Dobbs Journal of Software Tools*, 1999.

11 SHOSTACK, A. *Threat Modeling: Designing for Security*. [S.l.]: Wiley, 2014.

12 MITRE Corporation. *Common Attack Pattern Enumeration and Classification*. 2024. https://capec.mitre.org/. Accessado em: 2024-10-31.

13 MITRE Corporation. *Common Weakness Enumeration*. 2024. https://cwe.mitre.org/. Accessado em: 2024-11-01.

14 ESMAEILI, S. R.; ESTERABADI, A. S. Attack analysis methodologies. 2019.

15 COMMITTEE, S. V. E. S. S. et al. SAE J3061-cybersecurity guidebook for cyber-physical automotive systems. *SAE-society of automotive engineers*, 2016.

16 DANTAS, Y.; NIGAM, V.; RUESS, H. Security Engineering for ISO 21434. *arXiv preprint arXiv:2012.15080*, 12 2020. Disponível em: <https://www.researchgate.net/publication/348078835_Security_Engineering_for_ISO_21434>.

17 LAUTENBACH, A.; ALMGREN, M.; OLOVSSON, T. Proposing HEAVENS 2.0–an automotive risk assessment model. In: *Proceedings of the 5th ACM Computer Science in Cars Symposium*. [S.l.: s.n.], 2021. p. 1–12.

18 LAUTENBACH, A.; ISLAM, M. Heavens–healing vulnerabilities to enhance software security and safety. *The HEAVENS Consortium (Borås SE)*, Vinnova, 2016.

19 SAULAIMAN, M. N.-E. et al. Use cases of attack graph in threat analysis and risk assessment for the automotive domain. In: IEEE. *2022 IEEE 1st International Conference on Cognitive Mobility (CogMob)*. [S.l.], 2022. p. 000085–000092.

20 JAKOBS, C. et al. Heuristic Risk Treatment for ISO/SAE 21434 Development Projects. In: IEEE. *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*. [S.l.], 2022. p. 653–662.

21 MEROLA, F.; BERNARDESCHI, C.; LAMI, G. A risk assessment framework based on fuzzy logic for automotive systems. *Safety*, Multidisciplinary Digital Publishing Institute, v. 10, n. 2, p. 41, 2024.

22 MICHAELIS. *Segurança*. 2024. Acesso em: 20 de setembro de 2024. Disponível em: <https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/seguran%C3%A7a/>.

23 ABDULKHALEQ, A. et al. Using STPA in compliance with ISO 26262 for developing a safe architecture for fully automated vehicles. *arXiv preprint arXiv:1703.03657*, 2017.

24 WIKIPEDIA. *Segurança*. 2024. Acesso em: 20 de setembro de 2024. Disponível em: <https://pt.wikipedia.org/wiki/Seguran%C3%A7a>.

25 SILVA, F. G. da. *Segurança além do óbvio: Distinguindo safety e security no contexto de sistemas embarcados*. 2023. Acesso em: 06 de fevereiro de 2024. Disponível em: <https://fernandoginez.medium.com/seguran%C3%A7a-al%C3%A9m-do-%C3%B3bvio-distinguindo-safety-e-security-no-contexto-de-sistemas-embarcados-845d720486a5>.

26 CAPS. *Qual a diferença entre seguranca safety e-seguranca security*. 2022. Acesso em: 16 de Junho de 2025. Disponível em: <https://caps-pro.com.br/info/qual-diferenca-entre-seguranca-safety-e-seguranca-security/>.

27 NETKACHOVA, K.; BLOOMFIELD, R. E. Security-informed safety. *Computer*, v. 49, n. 6, p. 98–102, 2016.

28 ALLOUCH, A. et al. Qualitative and quantitative risk analysis and safety assessment of unmanned aerial vehicles missions over the internet. *Ieee Access*, IEEE, v. 7, p. 53392–53410, 2019.

29 WHITMAN, M. E.; MATTORD, H. J. et al. *Principles of information security*. [S.l.]: Thomson Course Technology Boston, MA, 2009.

30 MITNICK, K. D.; SIMON, W. L. *The art of deception: Controlling the human element of security*. [S.l.]: John Wiley & Sons, 2003.

31 ROTHSCHILD, E. What is security? *Daedalus*, JSTOR, v. 124, n. 3, p. 53–98, 1995.

32 WOOD, C. C. Why information security is now multi-disciplinary, multi-departmental, and multi-organizational in nature. *Computer Fraud & Security*, Elsevier, v. 2004, n. 1, p. 16–17, 2004.

33 SOLMS, R. V.; NIEKERK, J. V. From information security to cyber security. *computers & security*, Elsevier, v. 38, p. 97–102, 2013.

34 OSCARSON, P. Information security fundamentals: graphical conceptualisations for understanding. In: SPRINGER. *Security Education and Critical Infrastructures: IFIP TC11/WG11. 8 Third Annual World Conference on Information Security Education (WISE3) June 26–28, 2003, Monterey, California, USA 3*. [S.l.], 2003. p. 95–107.

35 4-LINUX. *Diferença entre cibersegurança e segurança da informação*. 2024. Acesso em: 28 de dezembro de 2024. Disponível em: <https://4linux.com.br/o-que-e-cyberseguranca/>.

36 SIEMENS. *As diferenças entre Cibersegurança e Segurança da Informação*. 2024. Acesso em: 28 de dezembro de 2024. Disponível em: <https://www.siemens.com/br/pt/empresa/stories/tecnologia/ciberseguranca-e-seguranca-da-informacao.html>.

37 ANDRESS, J. *Cybersecurity: The beginner's guide*. [S.l.]: No Starch Press, 2019.

38 STALLINGS, W.; BROWN, L. *Computer Security: Principles and Practice*. 5. ed. [S.l.]: Pearson, 2023.

39 European Union Agency for Cybersecurity. *ENISA Threat Landscape 2022*. 2022. Accessed: 2025-05-30. Disponível em: <https://www.enisa.europa.eu/publications/enisa-threat-landscape-2022>.

40 KIM, K. et al. Cybersecurity for autonomous vehicles: Review of attacks and defense. *Computers & Security*, Elsevier, v. 103, p. 102150, 2021.

41 TETTAMANTI, T.; VARGA, I.; SZALAY, Z. Impacts of autonomous cars from a traffic engineering perspective. *Periodica Polytechnica Transportation Engineering*, v. 44, n. 4, p. 244–250, 2016.

42 KIM, S.-H. et al. A gateway system for an automotive system: Lin, can, and flexray. In: IEEE. *2008 6th IEEE International Conference on Industrial Informatics*. [S.l.], 2008. p. 967–972.

43 TAKEFUJI, Y. Connected vehicle security vulnerabilities [commentary]. *IEEE Technology and Society Magazine*, IEEE, v. 37, n. 1, p. 15–18, 2018.

44 MAKOWITZ, R.; TEMPLE, C. Flexray-a communication network for automotive control systems. In: IEEE. *2006 IEEE International Workshop on Factory Communication Systems*. [S.l.], 2006. p. 207–212.

45 ELECTRONICS, L. about. *How to Build a CAN Transceiver Circuit with an MCP2551 Chip*. 2018. Acesso em: 16 de fevereiro de 2024. Disponível em: <https://www.learningaboutelectronics.com/Articles/MCP2551-CAN-transceiver-circuit.php>.

46 STANDARDIZATION, I. I. O. for. *ISO 11898-1:2015 Road vehicles Controller area network (CAN)*. 2015. Acesso em: 16 de fevereiro de 2024. Disponível em: <https://www.iso.org/standard/63648.html>.

47 KONG, L. et al. Millimeter-wave wireless communications for iot-cloud supported autonomous vehicles: Overview, design, and challenges. *IEEE Communications Magazine*, IEEE, v. 55, n. 1, p. 62–68, 2017.

48 ECONOMIST, T. *How does a self-driving car work?* 2015. Acesso em: 15 de fevereiro de 2024. Disponível em: <https://www.economist.com/the-economist-explains/2015/05/12/how-does-a-self-driving-car-work>.

49 PONSARD, C.; RAMON, V.; DEPREZ, J.-C. Goal and threat modelling for driving automotive cybersecurity risk analysis conforming to iso/sae 21434. In: *SECRYPT*. [S.l.: s.n.], 2021. p. 833–838.

50 2018, I. .-. *Road vehiclesFunctional safetyPart 3: Concept phase*. 2018.

51 ISO, I. ISO/IEC 18045: Information technology–Security techniques–Methodology for IT security evaluation. *Geneva, Switzerland, August*, 2008.

52 SWIDERSKI, F.; SNYDER, W. *Threat modeling*. [S.l.]: Microsoft Press, 2004.

53 SHEVCHENKO, N. et al. Threat modeling: a summary of available methods. *Software Engineering Institute/ Carnegie Mellon University*, p. 1–24, 2018.

54 WARD, D.; WOODERSON, P. *Automotive Cybersecurity: An Introduction to ISO/SAE 21434*. [S.l.]: SAE International, 2021.

55 BODEAU, D. J.; MCCOLLUM, C. D.; FOX, D. B. Cyber threat modeling: Survey, assessment, and representative framework. *Mitre Corp, Mclean*, p. 2021–11, 2018.

56 TATAM, M. et al. A review of threat modelling approaches for APT-style attacks. *Heliyon*, Elsevier, v. 7, n. 1, 2021.

57 IRIUSRISK. *Threat Modeling Methodology: STRIDE*. 2025. Acesso em: 13 de Janeiro de 2025. Disponível em: <https://www.iriusrisk.com/resources-blog/threat-modeling-methodology-stride>.

58 NASCIMENTO, L. et al. Model-based security assurance cases for open and adaptive cyber-physical systems. In: SPRINGER. *International Conference on Advanced Information Networking and Applications*. [S.l.], 2025. p. 326–340.

59 ECKHART, M.; EKELHART, A.; WEIPPL, E. Automated security risk identification using automationml-based engineering data. *IEEE Transactions on Dependable and Secure Computing*, IEEE, v. 19, n. 3, p. 1655–1672, 2020.

60 KANAKOGI, K. et al. Tracing CVE vulnerability information to CAPEC attack patterns using natural language processing techniques. *Information*, MDPI, v. 12, n. 8, p. 298, 2021.

61 MARIOTTI, F. et al. An extension of the advise meta modeling framework and its application for an early-stage security analysis of a public transport supervision system. *Journal of Reliable Intelligent Environments*, Springer, v. 9, n. 3, p. 263–281, 2023.

62 MITRE. *About CAPEC: Objective*. 2019. Acesso em: 06 de setembro de 2024. Disponível em: <https://capec.mitre.org/about/index.html>.

63 MITRE. *About CWE*. 2024. Acesso em: 29 de novembro de 2024. Disponível em: <https://cwe.mitre.org/about/index.html>.

64 WU, Y.; BOJANOVA, I.; YESHA, Y. They know your weaknesses–do you?: Reintroducing common weakness enumeration. *CrossTalk*, v. 45, 2015.

65 INC., M. *MongoDB Documentation*. 2024. https://www.mongodb.com/docs/. Acessado em abril de 2025.

66 CRAWLEY, O. B. *CAPEC-STRIDE Mapping*. 2022. Acesso em: 16 de Janeiro de 2025. Disponível em: <https://ostering.com/blog/2022/03/07/capec-stride-mapping/>.

67 (ISO), I. O. for S. *ISO/IEC 29100: 2011; Information TechnologySecurity TechniquesPrivacy Framework*. [S.l.]: ISO Geneva, 2011.

68 Wikipedia contributors. *Floor and ceiling functions — Wikipedia, The Free Encyclopedia*. 2025. https://en.wikipedia.org/w/index.php?title=Floor_and_ceiling_functions&oldid=1286916472. [Online; accessed 21-May-2025].

69 LEE, E. A.; SESHIA, S. A. Cyber physical systems: Design challenges. *Handbook of Real-Time and Embedded Systems*, CRC Press, 2016.

70 GHAFOURI, A.; JAFARI, M. A.; RIGGINS, F. J. Cybersecurity in cyber-physical systems: A survey. *IEEE Access*, IEEE, v. 10, p. 98995–99015, 2022.

71 MITRE Corporation. *MITRE ATT&CK Framework*. 2024. https://attack.mitre.org/. Accessado em: 2024-11-01.

72 MITRE Corporation. *Common Vulnerabilities and Exposures*. 2024. https://cve.mitre.org/. Accessado em: 2024-11-01.

73 HEMBERG, E. et al. *Linking Threat Tactics, Techniques, and Patterns with Defensive Weaknesses, Vulnerabilities and Affected Platform Configurations for Cyber Hunting*. 2021.

74 MITRE. *CAPEC Glossary*. 2024. Accessed: 2024-11-01. Disponível em: <https://capec.mitre.org/about/glossary.html>.

75 MITRE. *CWE Glossary*. 2024. Accessed: 2024-11-01. Disponível em: <https://cwe.mitre.org/documents/glossary/>.

76 SANDBERG, C.; BOKESAND, A.; THORSSON, U. *HoliSec Deliverable D4. 1.1-Tailoring the HEAVENS risk assessment methodology for improved performance*. [S.l.], 2018.

77 COSTANTINO, G.; VINCENZI, M. D.; MATTEUCCI, I. In-depth exploration of ISO/SAE 21434 and its correlations with existing standards. *IEEE Communications Standards Magazine*, IEEE, v. 6, n. 1, p. 84–92, 2022.

APÊNDICE A – Modelos entidade-relacionamento

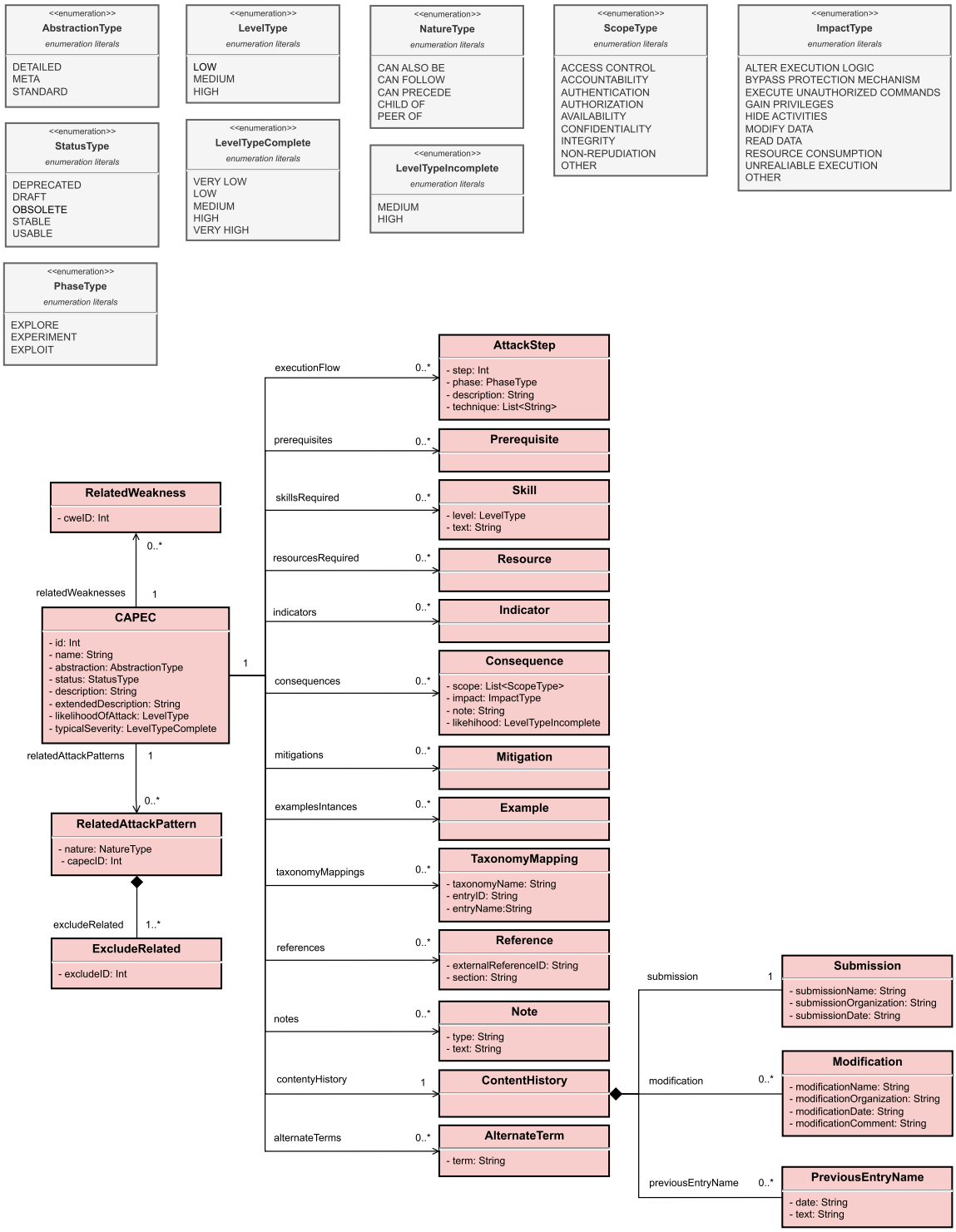
Com o objetivo de integrar padrões de ataque e fraquezas ao processo de análise TARA da ISO/SAE 21434, foram desenvolvidos metamodelos orientados a objetos através de um diagrama UML para o CAPEC e CWE. Os modelos foram construídos a partir da estrutura disponibilizada pela MITRE, visando organizar os dados de maneira sistemática, para facilitar consultas. As figuras a seguir apresentam a estrutura de cada catálogo, incluindo todas as entidades de sua estrutura. Além disso, também são descritos todos os atributos apresentados nos modelos, onde as entidades utilizadas para o processo TARA estão representadas em negrito.

O uso do DBSchema facilitou a visualização e validação das relações entre entidades, além de possibilitar futuras expansões do modelo para abranger outros conjuntos de dados (por exemplo, ATT&CK e CVE), conforme proposto nos trabalhos futuros deste estudo. Adicionalmente, *scripts* em Python foram utilizados para definir corretamente os identificadores únicos (*enumeration literals*).

Todos os diagramas e modelos apresentados neste trabalho foram desenvolvidos utilizando a ferramenta gráfica Diagrams.net, também conhecida como draw.io. Essa ferramenta permite a criação de diagramas diretamente na nuvem, com integração ao Google Drive, facilitando a organização e o acesso aos arquivos. Para melhor visualização dos modelos elaborados, recomenda-se o acesso ao diagrama completo por meio do seguinte link: https://drive.google.com/file/d/1pw8hJwKoOSxys0Wl3eFCCNljhdluJlOJ/view?usp=drive_link

CAPEC

Figura 15 – Metamodelo definindo as relações entre os atributos CAPEC



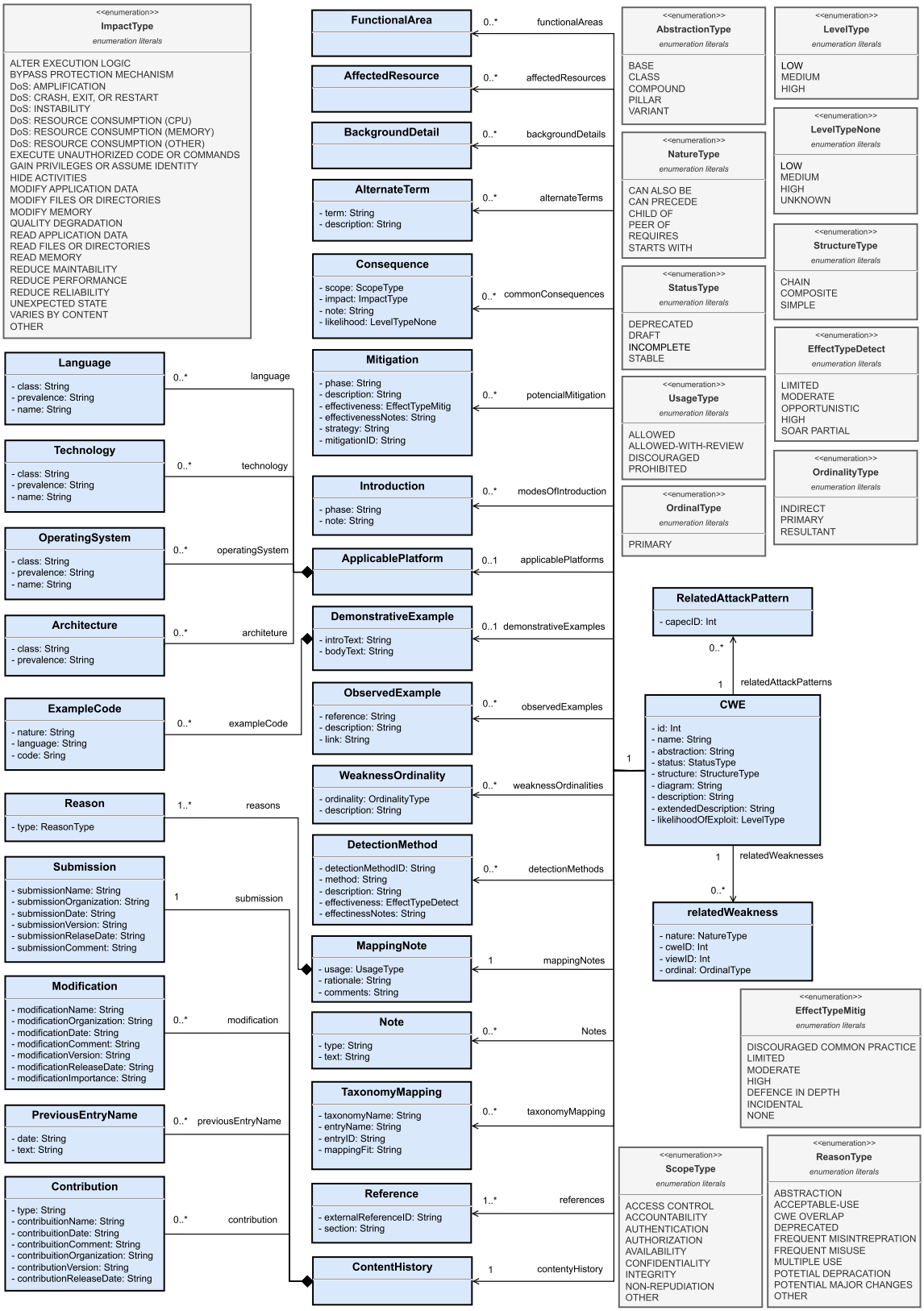
Fonte: Desenvolvido pelo autor (2025)

- **`id`**, **`name`** e **`description`**: Identificam de forma única o padrão de ataque e o descrevem de forma sucinta, sendo os atributos básicos para catalogação e indexação no banco CAPEC;
- `extendedDescription`: Fornece uma explicação mais abrangente sobre o funcionamento, motivação e cenário de aplicação do padrão de ataque, indo além da descrição básica;
- `status`: Indica o estado atual do padrão, sendo *Draft*, *Deprecated*, *Obsolete*, *Stable*, ou *Usable*; importante para compreender sua maturidade e confiabilidade;
- `abstraction`: Define o nível de generalização do padrão de ataque, podendo ser *Meta*, *Standard* ou *Detailed*, o que afeta a granularidade da análise;
- **`typicalSeverity`**: Representa uma estimativa do impacto que o ataque pode causar em um sistema típico. Esse atributo é expresso em níveis qualitativos (sendo *Very Low*, *Low*, *Medium*, *High* e *Very High*). Seus valores permitem determinar a severidade do impacto em métricas de análise de risco;
- **`likelihoodOfAttack`**: Estima a probabilidade de ocorrência do ataque em um ambiente real. Esse atributo também é categorizado em níveis qualitativos (sendo *Low*, *Medium* e *High*). Seus níveis permitem compor a definição da viabilidade de ataque em modelos de análise de risco;
- **`skillsRequired`**: Indica o nível de habilidade técnica que um atacante precisa possuir para executar o padrão de ataque com sucesso. Os níveis são classificados como *Low*, *Medium* ou *High*, refletindo a complexidade do ataque e sua acessibilidade para diferentes perfis de agentes;
- **`executionFlow`**: Descreve, de forma sequencial e estruturada os `AttackStep`, definidos como as etapas necessárias para que o ataque seja realizado. Isso inclui ações preliminares, condições necessárias para o sucesso do ataque, e os efeitos esperados sobre o sistema alvo (separados em *Explore*, *Experiment* e *Exploit*). Essa descrição permite a construção das árvores de ataque e o entendimento aprofundado sobre os vetores explorados pelo adversário;
- `prerequisites`: Define as condições ou estados do sistema que devem estar presentes para que o ataque seja possível. Esses pré-requisitos são fundamentais para avaliação da exposição de um sistema;
- `resourcesRequired`: Descreve os recursos materiais, computacionais ou de tempo necessários para realizar o ataque, como ferramentas específicas, acesso físico, ou tempo de preparação;

- `indicators`: Descreve sinais, evidências ou comportamentos que podem indicar que o padrão de ataque está sendo, ou foi, utilizado contra um sistema;
- `consequences`: Lista os possíveis impactos do ataque, como perda de integridade, disponibilidade ou confidencialidade, detalhando os efeitos adversos esperados após sua execução;
- `mitigations`: Apresenta estratégias de defesa, contramedidas ou práticas recomendadas que podem prevenir ou minimizar o sucesso do ataque descrito;
- `exampleInstances`: Fornece exemplos concretos ou históricos de incidentes em que o padrão de ataque foi identificado, permitindo ilustrar seu uso no mundo real (também são realizadas referências a CVEs);
- `notes`: Campo livre utilizado para observações adicionais que não se encaixam nos outros atributos formais, podendo incluir comentários da equipe mantenedora do CAPEC;
- `alternateTerms`: Apresenta termos alternativos usados para descrever o mesmo ataque, facilitando a busca por sinônimos e regionalismos técnicos;
- **`relatedAttackPatterns`**: Lista de outros padrões de ataque relacionados que compartilham técnicas, vetores ou objetivos semelhantes, auxiliando na identificação de cadeias ou famílias de ataques;
- **`relatedWeaknesses`**: Conecta o padrão de ataque a fraquezas conhecidas descritas na base da CWE, possibilitando o rastreamento técnico das falhas que podem ser exploradas;
- `taxonomyMappings`: Associa o ataque a outras taxonomias de segurança (como ATT&CK ou OWASP), permitindo interoperabilidade entre modelos e *frameworks* de análise de ameaças;
- `references`: Lista de fontes bibliográficas ou documentos utilizados na construção do padrão de ataque, que servem como material de apoio ou evidência técnica;
- `contentHistory`: Histórico de modificações do padrão, incluindo datas de criação, atualizações e revisões, importante para controle de versionamento.

CWE

Figura 16 – Metamodelo definindo as relações entre os atributos CWE



Fonte: Desenvolvido pelo autor (2025)

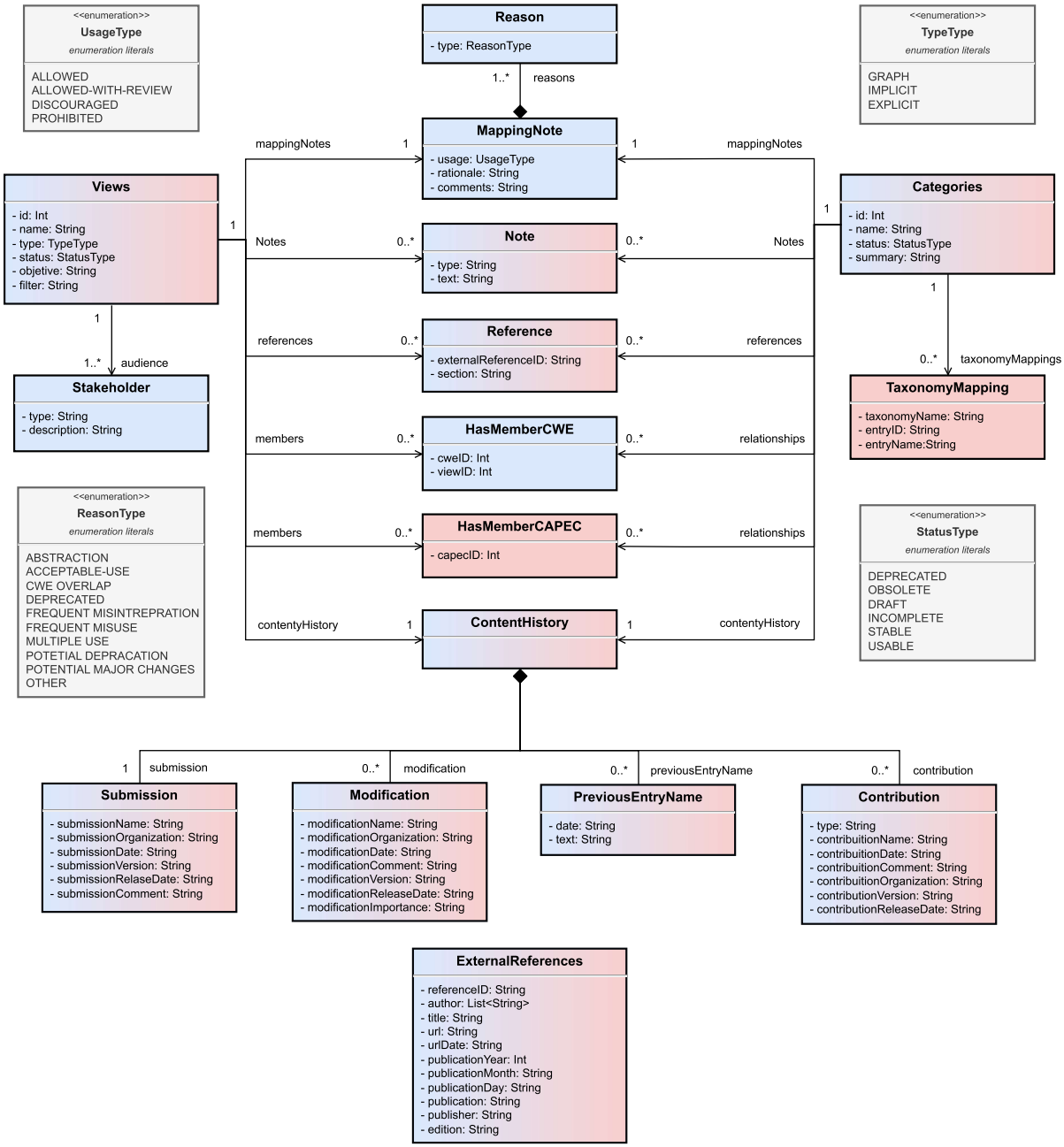
- **`name`**, **`ID`** e **`description`**: Identificam de forma única a fraqueza, fornecendo uma descrição resumida de seu funcionamento e impacto;
- `backgroundDetails`: Fornece informações históricas, motivações e observações sobre a fraqueza, contextualizando seu surgimento e relevância;
- `status`: Indica o estado atual da definição da fraqueza, como *Draft*, *Deprecated*, *Stable*, ou *Incomplete*, refletindo seu nível de maturidade;
- `abstraction`: Define o nível de generalização da fraqueza, podendo ser *Base*, *Class*, *Compound*, *Pillar* ou *Variant*, o que influencia a abrangência da análise;
- `structure`: Descreve a estrutura lógica da fraqueza, indicando se ela é composta (*Composite*), independente (*Simple*), ou parte de uma coleção maior de falhas (*Chain*);
- `weaknessOrdinalities`: indica relações hierárquicas ou de prioridade entre as fraquezas, sendo *Indirect*, *Primary* ou *Resultant*;
- **`mappingNotes`**: Fornece observações sobre como a fraqueza foi mapeada em relação a outras taxonomias ou padrões de segurança. Sua utilização é classificada como *Allowed*, *Allowed-with-review*, *Discouraged* ou *Prohibited*;
- `likelihoodOfExploit`: Estima a probabilidade de que a fraqueza seja explorada em um ambiente real. Utiliza categorização qualitativa, sendo *Low*, *Medium* e *High*;
- `applicablePlatforms`: Descreve os ambientes (`Language`), sistemas operacionais (`OperatingSystem`), arquiteturas (`Architecture`) ou contextos (`Technology`) nos quais a fraqueza pode ser explorada;
- `affectedResources`: Descreve os recursos ou ativos que são afetados diretamente pela fraqueza, como dados, processos ou *hardware*;
- `functionalAreas`: Categoriza a fraqueza dentro de áreas funcionais específicas, como criptografia, autenticação, ou gerenciamento de sessão, facilitando análises direcionadas;
- `modesOfIntroduction`: Explica como a fraqueza é introduzida no sistema e/ou produto, incluindo fases de desenvolvimento, operação ou manutenção;
- `commonConsequences`: Lista os impactos típicos associados à exploração da fraqueza, como vazamento de informações, negação de serviço ou execução de código não autorizado;
- **`detectionMethods`**: Descreve técnicas e ferramentas que podem ser empregadas para identificar a presença da fraqueza no sistema alvo;

- `potentialMitigations`: Apresenta práticas recomendadas, padrões de codificação segura e estratégias de defesa que podem ser adotadas para reduzir ou eliminar a fraqueza;
- `demonstrativeExamples`: Fornece exemplos ilustrativos mostrando como a fraqueza pode ocorrer em implementações reais ou fictícias;
- `observedExamples`: Apresenta casos reais documentados em que a fraqueza foi identificada e explorada, fortalecendo a compreensão prática do problema (são realizadas referências diretas a CVEs);
- `notes`: Espaço livre para informações adicionais relevantes sobre a fraqueza que não se enquadram nas categorias formais;
- `alternateTerms`: Apresenta termos alternativos ou sinônimos utilizados para descrever a fraqueza, facilitando a busca e o entendimento técnico;
- `diagram`: Representa visualmente a estrutura ou fluxo associado à fraqueza, podendo incluir diagramas de classes, fluxogramas ou representações arquiteturais;
- `relatedWeaknesses`: Lista de outras fraquezas associadas que compartilham características estruturais, funcionais ou contextuais semelhantes, auxiliando na análise de fraquezas combinadas;
- `relatedAttackPatterns`: Conecta a fraqueza a padrões de ataque conhecidos descritos na base CAPEC que têm como alvo essa fraqueza, permitindo avaliação de ameaças correlatas;
- `taxonomyMappings`: Associa a fraqueza a outras taxonomias ou *frameworks* de segurança (como OWASP pu CERT), fortalecendo a interoperabilidade entre bases de dados;
- `references`: Lista fontes externas que oferecem suporte técnico adicional sobre a fraqueza, servindo como material de apoio;
- `contentHistory`: Registra o histórico de alterações da fraqueza, incluindo a data de criação, atualizações e revisões.

Entidades em comum entre CAPEC e CWE

Além das entidades centrais específicas de cada base, os arquivos disponibilizados pela MITRE também incluem estruturas auxiliares, sendo elas as entidades de categoria (`Category`), visualização (`Views`) e referências externas (`ExternalReferences`). Essas estruturas desempenham papel fundamental na organização, categorização e rastreabilidade dos elementos, promovendo maior consistência semântica e interoperabilidade entre os dados. Embora sejam definidas separadamente em cada base, muitas dessas entidades possuem estruturas análogas entre os catálogos. Para ilustrar essas interseções, a Figura 17 apresenta o metamodelo conceitual que representa essas entidades auxiliares e seus relacionamentos com as demais estruturas das bases. As entidades compartilhadas entre as bases CAPEC e CWE são destacadas por meio de gradientes que combinam suas cores características (vermelho e azul), facilitando a visualização das interseções entre as duas bases.

Figura 17 – Metamodelo representando as estruturas auxiliares das bases



Fonte: Desenvolvido pelo autor (2025)

Além das entidades auxiliares apresentadas, o modelo também contempla estruturas comuns previamente discutidas, como `MappingNotes`, `Notes`, `References`, `ContentHistory` e `TaxonomyMappings`. Essas entidades são reutilizadas tanto no CAPEC quanto no CWE para fornecer contexto descritivo, histórico e referencial às bases. A seguir, são detalhadas as entidades auxiliares introduzidas nesta seção, conforme definidas nos glossários oficiais da MITRE [74, 75]:

- `Categories`: Estrutura que organiza os padrões de ataque e fraquezas em agrupamentos temáticos, com base em objetivos ou comportamentos comuns. Essa categorização facilita a navegação, permitindo análises mais amplas por área de interesse ou domínio funcional;
- `Views`: Coleções estruturadas de entradas que refletem diferentes perspectivas analíticas, como abordagens por linguagem de programação, arquitetura de sistema ou domínio específico. As `Views` também definem o público-alvo e os filtros aplicáveis, o que favorece a utilização especializada das informações por partes interessadas distintas;
- `ExternalReferences`: Referências cruzadas para fontes externas, como publicações acadêmicas, normas técnicas, diretrizes de segurança (ex: NIST, OWASP) e CVEs específicos. Essa entidade fortalece a rastreabilidade das entradas, fornecendo evidências e vínculos adicionais para validação ou exploração aprofundada;

A modelagem dessas entidades auxiliares contribui significativamente para a construção de um banco de dados mais completo, coeso e preparado para integração com outras fontes e ferramentas. Essas estruturas, embora não tenham sido diretamente exploradas no modelo principal deste trabalho, representam oportunidades de futuras extensões voltadas à interoperabilidade, análise cruzada de dados, versionamento de conteúdo e integração com taxonomias complementares.

APÊNDICE B – Banco de dados

A seguir, é apresentado o processo de construção do banco de dados utilizado neste trabalho, contendo informações extraídas diretamente das bases CAPEC, CWE e do mapeamento CAPEC-STRIDE. O objetivo foi consolidar essas informações em uma estrutura unificada, em uma base NoSQL, visando apoiar a análise de ameaças e o tratamento de riscos cibernéticos em conformidade com a ISO/SAE 21434.

Aquisição e Processamento dos Dados

Os dados utilizados neste trabalho foram extraídos de fontes oficiais mantidas pela MITRE e por colaboradores da comunidade. Os catálogos CAPEC e CWE foram obtidos em formato XML, enquanto o mapeamento STRIDE-CAPEC foi disponibilizado em JSON. Para viabilizar o uso dessas informações em um banco de dados orientado a documentos usando o MongoDB, foi necessário realizar a conversão e a reestruturação dos catálogos para o formato JSON. Esse processo foi conduzido por meio de *scripts* desenvolvidos em Python, que realizaram as seguintes tarefas: (i) extração dos dados brutos, (ii) transformação de estruturas hierárquicas XML em objetos JSON, (iii) padronização de nomenclaturas e atributos, e (iv) organização dos relacionamentos entre entidades de forma compatível com a modelagem proposta. A inserção dos dados no MongoDB foi realizada com base nesse conjunto de documentos transformados, permitindo consultas flexíveis e análises estruturadas. Abaixo são apresentadas as informações referentes aos dados utilizados:

- **CAPEC**
  - **Data de aquisição:** 31/10/2024
  - **Arquivo:** `capec_v3.9.xml`
  - **Fonte:** https://capec.mitre.org/data/downloads.html
- **CWE**
  - **Data de aquisição:** 01/11/2024
  - **Arquivo:** `cwec_v4.15.xml`
  - **Fonte:** https://cwe.mitre.org/data/downloads.html
- **STRIDE-CAPEC Mapping**
  - **Data de aquisição:** 30/10/2024
  - **Arquivo:** `capec-stride-mapping.json`
  - **Fonte:** https://ostering.com/blog/2022/03/07/capec-stride-mapping/

Todo o processo desde a aquisição até a inserção no banco está documentado em um repositório público no GitHub[i], que também disponibiliza *scripts* auxiliares para a inspeção dos atributos e identificação de valores únicos, utilizados para a construção dos *enumeration literals* presentes nos metamodelos definidos no Apêndice A.

Infraestrutura utilizada

A construção e operação do banco de dados presente neste trabalho foi realizada através da plataforma MongoDB Atlas, um serviço de banco de dados NoSQL em nuvem que oferece escalabilidade, redundância e segurança nativas. O MongoDB Atlas permite a criação de *clusters* com acesso controlado, garantindo maior confiabilidade e proteção dos dados manipulados durante o desenvolvimento.

A opção por uma solução em nuvem justificou-se tanto pela facilidade de implantação quanto pela possibilidade de acessar o banco de dados remotamente, a partir de diferentes dispositivos e ambientes de teste. O ambiente foi configurado com autenticação por usuário e IPs restritos, seguindo boas práticas de segurança para ambientes em desenvolvimento. O processo de ingestão e modelagem dos dados foi realizado por meio de *scripts* Python que se conectam diretamente com o cluster MongoDB hospedado no Atlas, utilizando a biblioteca `pymongo`.

Para inspeção, depuração e análise manual dos documentos armazenados, foi utilizado o MongoDB Compass, uma interface gráfica oficial da MongoDB Inc. Essa ferramenta permitiu realizar consultas, visualizar coleções e atributos, aplicar filtros e verificar a consistência dos dados durante o processo de validação do banco. Essa infraestrutura permitiu a criação de uma base robusta, flexível e facilmente expansível, integrando dados complexos oriundos de diferentes fontes (sendo o CAPEC, CWE e STRIDE) em um único ambiente de análise, compatível com os objetivos metodológicos do trabalho.

Devido a restrições de segurança da própria plataforma, não foi possível disponibilizar um banco de dados acessível ao público. No entanto, o repositório desenvolvido no GitHub contém todos os passos e arquivos necessários para que outros pesquisadores possam replicar o processo de criação do banco de dados. O repositório fornece os dados brutos convertidos, os *scripts* de importação e exemplos de consultas, possibilitando a reconstrução do banco de dados em instâncias locais ou privadas do MongoDB Atlas.

Atributos não utilizados

Durante a construção do banco de dados e do modelo proposto, diversos atributos disponibilizados pelas bases CAPEC e CWE não foram diretamente incorporados na

[i] Link para o repositório: https://github.com/EduProgram/STRIDE-CAPEC-CWE_DB

estrutura final. No entanto, tais campos apresentam potencial para enriquecer etapas específicas do processo de análise de risco, incluindo a identificação de ameaças, a avaliação de impacto e a definição de contramedidas. Nesse sentido, esses atributos representam oportunidades para trabalhos futuros, seja no refinamento do modelo proposto, ou em aplicações complementares, como o uso de técnicas de aprendizado de máquina.

O Quadro 16 apresenta uma consolidação dos principais atributos não utilizados, organizados segundo sua base de origem, frequência de ocorrência e aplicabilidade potencial no contexto do modelo proposto. Para fins analíticos, esses atributos foram agrupados em três categorias: (i) atributos com potencial de serem integrados em versões futuras do modelo; (ii) atributos com baixa ocorrência ou cobertura nas bases da MITRE; e (iii) atributos com relevância para aplicações que envolvem mineração de dados e aprendizado de máquina.

Quadro 16 – Atributos que podem ser utilizados futuramente no banco de dados

Atributo	**Base**	**Nž**	**Possível Aplicação**
`Consequence.Scope`	CAPEC CWE	369 917	Identificação do cenário de ameaça
`Consequence.Impact`	CAPEC CWE	368 917	Identificação do cenário de dano
`Taxonomy.Mappings`	CAPEC CWE	223 638	Integração com outras bases
`Prerequisites`	CAPEC	490	Análise do caminho de ataque
`Mitigations`	CAPEC	400	Decisão do tratamento de riscos
`Resources_Required`	CAPEC	272	Análise do caminho de ataque
`Applicable_Platforms`	CWE	726	Análise do caminho de ataque Decisão do tratamento do risco
`Likelihood_Of_Exploit`	CWE	185	Class. da viabilidade de ataque Decisão do tratamento de risco
`Consequence.Likelihood`	CAPEC CWE	2 59	Classificação de impacto (poucos resultados)
`Indicators`	CAPEC	56	Decisão do tratamento de risco (poucos resultados)
`Affected_Resources`	CWE	51	Identificação do cenário de dano (poucos resultados)
`Functional_Areas`	CWE	32	Identificação do cenário de dano (poucos resultados)
`Example_Instances`	CAPEC	269	Decisão do tratamento de riscos (uso de *Machine Learning*)
`Demonstrative_Examples`	CWE	576	Decisão do tratamento de riscos (uso de *Machine Learning*)
`Observed_Examples`	CWE	549	Decisão do tratamento de riscos (uso de *Machine Learning*)

Fonte: Desenvolvido pelo autor (2025)

Tais atributos podem ser utilizados para representar pré-condições de ataque, plataformas vulneráveis, escopos de impacto e até mesmo complementar a análise de probabilidade em cenários de risco. Sua inclusão poderá ampliar a profundidade e a precisão das avaliações. A seguir, é apresentada uma breve descrição sobre a aplicabilidade dos atributos listados para o modelo proposto:

- `Consequence.Scope`: Auxilia na **identificação do cenário de ameaça** associado ao possível dano causado por ataques e/ou fraquezas;
- `Consequence.Impact`: Complementa a **identificação do cenário de dano** ao caracterizar os domínios e tipos de danos causados por ataques e/ou fraquezas;
- `Taxonomy.Mappings`: Facilita a interoperabilidade com outras taxonomias de segurança e fontes de danos (como ATT&CK, NIST ou OWASP Top 10), possibilitando a extensão do modelo;
- `Prerequisites` e `Resources_Required`: Contribuem para a fundamentação da **análise do caminho de ataque**, estimando a viabilidade técnica, complexidade e ordem de execução dos ataques encadeados;
- `Mitigations` (CAPEC): Oferece diretrizes de contramedidas voltadas ao padrão de ataque, podendo ser utilizadas na etapa de **decisão do tratamento de risco**, atuando de forma complementar às estratégias de mitigação descritas no CWE;
- `Applicable_Platforms`: Permite filtrar ameaças por tipo de sistema ou arquitetura, podendo auxiliar na etapa de **análise do caminho de ataque**, ou na **decisão do tratamento de risco**;
- `Likelihood_Of_Exploit`: Representa uma estimativa qualitativa da probabilidade de uma fraqueza ser explorada, onde seu valor pode ser usado para a **classificação da viabilidade de ataque** ou ainda para metrificar a etapa de **decisão do tratamento de risco**;
- `Affected_Resources` e `Functional_Areas`: Apoiam a etapa de **identificação do cenário de dano**, apresentando os recursos impactados em decorrência da exploração da fraqueza e sua categorização funcional;
- `Indicators`: Aponta possíveis sinais de exploração ativa ou anomalias no sistema, podendo ser útil na etapa de **decisão do tratamento de risco**;
- `Example_Instances`, `Demonstrative_Examples` e `Observed_Examples`: Exemplos reais de como o ataque/fraqueza pode ser implementado e/ou explorado. São fontes ricas para aplicações em aprendizado de máquina, tanto supervisionado quanto não

supervisionado, definindo assim estratégias para a **decisão do tratamento de risco**;

Apesar de não fazerem parte da estrutura do modelo atual, esses campos permanecem armazenados e documentados no banco de dados, oferecendo suporte a futuras extensões. Embora alguns atributos, tais como o `Consequence.Likelihood` e `Affected_Resources` pudessem ser potencialmente relevantes para o processo de classificação de impacto e identificação do cenário de dano, foram desconsiderados devido à baixa frequência de preenchimento nos dados extraídos das bases originais. Por fim, também são apresentados atributos que fornecem exemplos reais de instâncias ou casos demonstrativos que podem ser explorados em contextos de aprendizado de máquina e mineração de dados. Esses dados podem ser utilizados para compor conjuntos de treinamento, validar algoritmos de detecção de vulnerabilidades ou ainda construir modelos preditivos voltados à recomendação de contramedidas ou à classificação automática de novos caminhos de ataque com base em padrões históricos.

APÊNDICE C – Artigo científico desenvolvido

Este apêndice apresenta o artigo científico desenvolvido durante a execução da dissertação de mestrado, intitulado "*Integration of Attack Patterns, Weaknesses and Vulnerability Databases to Support Security Analysis in Automotive Systems*". Embora compartilhe os mesmos fundamentos conceituais presentes na dissertação como o alinhamento à norma ISO/SAE 21434 e o uso de bases de dados de segurança o artigo adota uma abordagem complementar, centrada na formalização ontológica da norma e na construção de um metamodelo conceitual em UML que também integra a base CVE. O foco do trabalho está na representação semântica dos elementos envolvidos no processo de análise de riscos, propondo um mapeamento estruturado entre os conceitos das bases de dados e as etapas do processo TARA, além de discutir o uso do CVE como suporte empírico para identificação de vulnerabilidades e avaliação de impacto. Dentre os principais diferenciais do artigo, destacam-se:

- A especificação de uma ontologia formal da norma ISO/SAE 21434, construída a partir da análise da Fase de Conceito (Cláusula 9) contemplando entidades como itens, ativos, propriedades de segurança, cenários de dano, cenários de ameaça e caminhos de ataque. A ontologia incorpora, ainda, a abordagem baseada em potencial de ataque definida na ISO/IEC 18045, utilizando parâmetros como tempo, expertise, conhecimento do item, equipamentos e janela de oportunidade este último inspirado na modelagem proposta por Sandberg et al. [76] para estimar a viabilidade de ataques de forma estruturada e alinhada às diretrizes normativas;
- A elaboração de um metamodelo conceitual em UML que integra de forma empírica os catálogos CAPEC, CWE e CVE, com o objetivo de simplificar e estruturar as relações semânticas entre essas bases. A integração empírica da base CVE possibilita ao modelo explorar como os identificadores CVE podem apoiar o refinamento da análise de risco ao prover exemplos concretos de vulnerabilidades observadas;
- A inclusão de uma tabela que relaciona as principais atividades do processo TARA da ISO/SAE 21434 com os elementos das bases de dados de segurança, permitindo visualizar como essas fontes externas podem apoiar diretamente a identificação de cenários de dano, ameaça, impacto, viabilidade de ataque, e tomada de decisão em relação ao risco;

Assim, embora ambos os trabalhos compartilhem objetivos complementares, a dissertação e o artigo diferenciam-se em escopo e profundidade: o artigo enfatiza aspectos formais e conceituais da integração entre as bases, enquanto a dissertação foca na aplicação prática e na viabilidade do modelo em cenários automotivos concretos. A seguir, apresenta-se o artigo por completo.

Integrating Attack Patterns, Weaknesses, and Vulnerabilities Databases to support Security Analysis in the Automotive Domain

Abstract. *The higher interconnectivity of contemporary cyber-physical systems in the automotive, aerospace, smart cities and other critical domains demand justification and demonstration they are protected against security threats. In the automotive domain, the ISO 21434 cybersecurity standard defines a detailed workflow for Threat Analysis and Risk Assessment (TARA). Public cybersecurity databases such as NIST's Common Weakness Enumerations (CWE), Common Vulnerabilities and Exposures (CVE), and Common Attack Pattern Enumeration and Classification (CAPEC) can support engineers with insights on potential vulnerabilities, threats, and attack patterns during threat analysis. Those databases have both implicit and explicit references to information from each other. The integration of information from security databases may support engineers during cybersecurity analysis. This paper introduces a conceptual model that explicitly describes CAPEC, CWE, and CVE security database entities, their attributes, and relationships. We also describe how those databases can support ISO 21434 cybersecurity analysis activities. We illustrate concrete examples of CAPEC, CWE, and CVE entities and their relationships by performing security analysis of an automotive headlamp system example from the ISO 21434 standard.*

1. Introduction

The ubiquitous presence of technological systems in domains such as transportation, with autonomous railway, aviation, and automotive systems equipped with sensors and mechanisms that control the vehicle's behaviour, making them essential for the operation and evolution of modern societies [Retouniotis et al., 2025]. Since these systems interact with physical processes and their environment, they are becoming cyber-physical. They integrate physical processes and computer systems, equipped with sensors for observing the environment, and actuators to modify physical processes.

In the automotive domain, new vehicle business models are emerging based on adopting Information and Communication Technologies (ICT), such as V2X (to vehicle, infrastructure) communication, and Artificial Intelligence. Currently, there are tens of millions of connected vehicles, enabling the provision of relevant services such as online streaming, and Advanced Driver Assistant Systems (ADAS) with the potential for achieving a high level of autonomy shortly [Dantas et al., 2020].

The adoption of ICT in automotive systems raises vehicle cybersecurity concerns. The communication channel could be a vulnerability that could be exploited by attack vectors to compromise the vehicle system or data confidentiality, integrity, availability, or other cyber-security properties of interest [Biro et al., 2017]. Attackers would necessarily have to be physically close to carry out attacks against vehicles earlier, which is no longer needed with connected cars [Dantas et al., 2020]. For instance, the infamous Jeep attack [Wired, 2015] demonstrated that without appropriate

countermeasures, attackers can take control of any vehicle function remotely, such as turning the engine off, steering, and wheel braking.

The automotive industry has increasingly become a target of cyberattacks focused on compromising the ICT infrastructure of car manufacturers, aiming to steal data or demand ransom through ransomware, posing significant threats to both companies and their customers [Huq, 2024]. The first decade of automotive cybersecurity was marked by an increase in the number of cyber incidents and attacks against OEMs (Original Equipment Manufacturers) and the ecosystem, continuously introducing new attack vectors (e.g., ECUs, APIs, infotainment, telematics, and cloud systems, remote keyless) and methods [Upstream, 2024]. The annual Upstream Global Automotive Cybersecurity has tracked 1763 automotive-related incidents since 2010, with 295 only in 2023. For instance, a vulnerability was identified in Toyota[1] and Lexus models, which allowed thieves to exploit the CAN bus system. By sending messages through the CAN bus, attackers could trick the vehicle into believing a valid key was nearby, unlocking the doors and disabling the immobilizer [Huq, 2024]. The number of automotive-related CVEs (Common Vulnerability and Exposures) has increased from 24 in 2019 to 725 in 2023 [Upstream, 2024]. The 378 new automotive-related CVEs identified in 2023 represent 52% of the total CVEs. Moreover, in 2023, 95% of the attacks in automotive systems were remote, with 64% of them executed by black hat actors (i.e., malicious external agents).

The higher connectivity of contemporary cyber-physical systems in the automotive and other critical domains demands justification and demonstration that they are protected against cybersecurity threats caused by attacks performed by malicious external agents aimed at compromising system or data confidentiality, integrity, availability, or other security properties of interest. In the automotive domain, the ISO 21434 cybersecurity standard defines a detailed workflow for Threat Analysis and Risk Assessment (TARA), comprising item definition, identification of damage and threat scenarios, and attack path analysis as part of the certification process[2].

Publicly available cybersecurity databases such as MITRE Common Weakness Enumerations (CWE), Common Vulnerabilities and Exposures (CVE), and Common Attack Pattern Enumeration and Classification (CAPEC) can support engineers with insights on potential vulnerabilities, threats, and attack patterns during threat analysis. Those databases have both implicit and explicit references to information from each other. The integration of information from security databases may support engineers during cybersecurity analysis. Related works [Hemberg et al., 2024, Yuan et al., 2021] use graphs to describe complex relationships between concepts embedded into security databases, which are difficult to understand. Moreover, these studies do not describe how CVE, CWE, and CAPEC databases can support engineers during ISO 21434 cybersecurity analysis process. In this paper, we introduce a conceptual model, specified in the Unified Modelling Language (UML), to explicitly describe CAPEC, CWE, and CVE security database entities, their attributes, and relationships. We also describe how

[1] https://www.leighday.co.uk/news/news/2024-news/toyota-and-lexus-owners-could-be-owed-compensation-as-some-keyless-vehicles-found-to-be-vulnerable-to-a-device-which-allows-them-to-be-stolen-in-minutes/

[2] Process of assuring that a product or process has certain properties, which are recorded in a certificate.

those databases can support ISO 21434 cybersecurity analysis activities. We illustrate concrete examples of CVE, CWE, and CAPEC entities and their relationships by performing a security analysis of an automotive headlamp system from the ISO 21434.

This paper is organized as follows. Section 2 presents background concepts on the ISO 21434 terminology and cybersecurity assurance process, STRIDE [Hernan et al., 2006] and attack tree analysis [Schneier, 1999] techniques, CVE, CWE, and CAPEC cybersecurity databases needed for better understanding the contributions of this paper. Section 3 introduces the ISO 21434 ontology, and a conceptual model built upon on it that explicitly describes the relationships between CAPEC, CVE, and CWE security database entities. Section 4 illustrates examples of relationships between CAPEC, CVE, and CWE in an automotive headlamp system. Section 5 presents conclusion and sketches future work.

2. Background

2.1. The ISO 21434 Standard and Cybersecurity Security Analysis Techniques

The ISO 21434 [ISO/SAE, 2021] is an automotive cybersecurity standard that defines the concept of Cybersecurity Assurance Levels, and a detailed workflow for Threat Analysis and Risk Assessment (TARA) without prescribing a specific methodology (see Fig. 1). The process starts with the ***item*** (*system*) ***definition***, which includes the definition of the **item boundary** and **function**, the **preliminary architecture** as well as its **operational environment**, and relevant assumptions about the item and its environment in an initial data-flow diagram (DFD).). The DFD includes the most important entities and data-flow of the item and it is the input for **asset**, **damage scenario**, and **threat scenario identification** activities of the process.

In the ISO 21434, an **item** stands for a component or a set of components that implement a function at the vehicle level. The **operational environment** of an **item** or a **component** stands for their **operational** use, in a vehicle function, in production, and/or in service. After defining the item boundaries and its operational environment, the assets, their associated cybersecurity properties, and damage scenarios are identified. An **asset** is an object that has value, or contributes to a value. An **asset** could be a **system**, a **component**, **data**, or **data communication**. An asset has one or more **cybersecurity properties** whose compromise (by an attack) can lead to one or more damage scenarios. A **cybersecurity property** is an attribute of an asset that can worth to protect, e.g., confidentiality, integrity, availability, authenticity, authorization, or non-repudiation. The **assets** worth protecting and their **cybersecurity properties** are identified during the asset identification. Then, the potential violations of asset's cybersecurity properties (**damage scenarios**) are identified. A **damage scenario** is an adverse consequence involving a vehicle or vehicle function, and affecting a road user.

Each identified **damage scenario** is later assessed against its potential adverse consequences for road users in **impact categories** according to their **safety**, **financial**, **operational**, and **privacy** (S, F, O, P) impact. The **impact** is the degree of magnitude of damage or physical harm from a damage scenario. Additional impact categories can be considered to determine the **impact rating** of a **damage scenario**. In this case, the rationale and explanation of these categories can be shared in the supply chain in

conformance with the ISO 21434 Clause 7. The **impact rating** of a damage scenario can be classified as **severe**, **major**, **moderate**, or **negligible** according to the assigned safety, financial, operation, and privacy impact rating criteria, derived from the ISO 26262 [ISO, 2018] functional safety standard for road vehicles. During **the threat scenario identification**, engineers describe a set of **potential causes** (threats, e.g., known vulnerabilities) that may lead to one or more **damage scenarios**, which specify the adverse consequences (results) of an **attack**. A **threat scenario** is a potential **cause** of violation of **cybersecurity properties** of one or more assets in order to realize a damage scenario. The STRIDE (**S**poofing, **T**ampering, **R**epudiation, **I**nformation disclosure, **D**enial of service, **E**levation of privilege) method [Hernan et al., 2006] supports engineers to enumerate cybersecurity threats on the assets based on the DFD. A single threat scenario may lead to the occurrence of the multiple damage scenarios. . For instance, spoofing of CAN messages for the braking ECU of a vehicle leads to the loss of integrity of the CAN messages, and to the loss of integrity of the braking function. Multiple threat scenarios can also be the cause of the same damage scenario. ISO 21434 also prescribes performing the attack path analysis to identify the possible attacks that might realize each identified threat scenario. An **attack path** (**attack**) is a set of deliberate **actions** to realize a threat scenario. Attack path analysis can be performed using top-down approaches, e.g., attack trees and attack graphs, that deduces attack paths by analysing the ways a threat scenario could be realized, or bottom-up approaches that build attack paths from the identified vulnerabilities.

Attack trees [Schneier, 1999] are similar to fault trees used in safety modelling, facilitating the communication between safety and security engineers. Since damage scenarios and threat scenarios have been identified, we'll create an **attack tree** for each **damage scenario**, which is the **root node** of the tree. **Threat scenarios** leading to those **damage scenarios** are **children** of the root node. Depending on the specificity of the **threat scenario**, it can be split into **sub-nodes**, or new nodes can be added on a path to an external interface which may be the entry point for one or more **attacks** that realize the threat scenario. The total set of identified **attack paths** are the set of unique **paths** from the **leaf** to the **root node** for all **attack trees**. Later, each identified **attack path** should receive an **attack feasibility rating** (high, medium, low, or very low). The attack

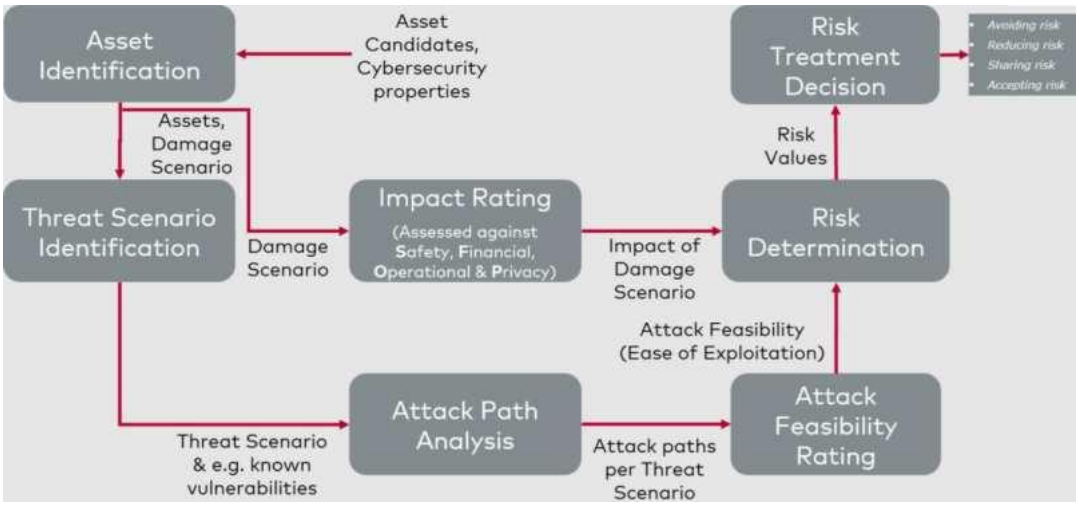


Fig. 1. ISO 21434 cybersecurity assessment process [ISO, 2021].

feasibility rating can be determined based on the **elapsed time**, **specialist expertise**, **knowledge** of the **item** or component, **window of opportunity**, and **equipment** core factors from the attack potential-based approach. Later, the **risk value** for each identified **threat scenario** is determined. A single **threat scenario** can have multiple associated **damage scenarios** and **attack paths**, which implies that there are multiple associated impact ratings and feasibility ratings respectively. ISO 21434 explicitly recognizes the problem of multiple attack feasibility ratings to calculate the risk value for a given **threat scenario**, and it recommends to consider the **highest value** assigned to any associated **attack path**, but the same is not acknowledge for determining the impact of damage scenarios. The standard recommends considering the highest value of both associated **damage scenario impact** rating and **attack path feasibility** rating to determining the **risk value** of a **threat scenario**, to do not underestimate it [ISO/SAE, 2021]. The **risk value** of a **threat scenario** is determined based on a risk matrix, as illustrated in Fig. 2, which defines five risk levels prescribed by the ISO 21434 cybersecurity standard for classifying the **risk** of a **threat scenario** based on its **impact** and **feasibility ratings**.

Based on the **risk values** assigned to **threat scenarios**, **risk treatment decision** has to be taken to define if the **risk** should be **avoided** (by removing the source of the risk), **shared** or **transferred** (achieved through contracts), **accepted** (manageable without measures), or **reduced** through additional **security controls** in the item. The standard also defines an approach to define the **cybersecurity goals/claims** for each identified **threat scenario**, based on its **risk value**, in the form of a **Cybersecurity Assurance Level** (**CAL**). **Cybersecurity goals** are high-level **cybersecurity requirements**. The ISO 21434 defines **cybersecurity goals** in the form of a **CAL**, which specifies the target level of process rigor for the **security validation process**. For instance, to mitigate the risk posed by a **threat scenario** with a higher **CAL 4**, we need to perform **functional, fuzzy**, and **penetration testing**, and **vulnerability analysis** in the asset. **Cybersecurity claims** for **accepted** or **transferred** risk are statements about why a risk is **acceptable**, or assumptions that must be fulfilled for the risk to be acceptable.

		Impact rating			
		Negl.	Mod.	Maj.	Sev.
Attack feasibility rating	Very Low	1	1	2	3
	Low	1	2	3	4
	Medium	2	3	4	5
	High	2	4	5	5

Fig. 2. ISO 21434 risk matrix.

2.2 Attack Trees

Attack Tree (AT) [Schneier, 1999] is a top-down reasoning technique for cybersecurity attack path analysis. AT is a direct acyclic graph, with sub-trees that can have multiple parent gates, used to describe how lower level attacks (basic events) propagate through the system leading to the occurrence of threat scenarios that cause a higher-level damage scenario (top-event) that violates a cybersecurity property of the

asset (system, hardware/software/ electronic component, data, or data communication). Attack tree is recommended by the ISO 21434 cybersecurity standard to support attack path analysis. Attack tree analysis was proposed in later ‘90 as the security counterpart of Fault Tree Analysis (FTA) [NASA, 2002] technique. An attack tree is described using a visual notation, beginning with a damage scenario (the top event) associated with a particular asset, and progressively exploits possible combinations of causes (threat scenarios) until reaching attacks that realize threat scenarios leading to the top event. Like a fault tree, an attack tree describes the logical relationships between damage scenarios, threat scenarios, and attacks using AND, OR, NOT gates.

Attack trees support both qualitative (logical), including Minimal Cut Sets (MCSs) calculation to indicate the combinations of basic events leading to the occurrence of the top event, and quantitative (probabilistic) analysis. Quantitative analysis computes dependability metrics such as system availability, attack probability, feasibility, and costs. For example, by attaching probabilities to basic events (component faults or attacks), it is possible to calculate the likelihood of a system level failure or an attack. Fig. 4 illustrates an example of attack tree for the damage scenario related to the lamp request data communication asset of the headlamp system used in the ISO 21434 cybersecurity standard. The *unintended turning-off of headlamp system during night driving at medium speed* related to the lamp request violates *integrity* with a **severe** (**S3**) safety impact, leading to front collision with a narrow stationary object is the top event. There are two potential causes for the identified damage scenario. An ***spoofing*** of the lamp request signal to the power switch ECU, or ***tampering*** with a signal sent by the body control ECU potentially lead to the headlamp to turning-off unintentionally. An ***spoofing*** of the lamp request signal can happen when an attacker compromisses the navigation ECU via cellular (**atack path 1**) or Bluetooth (**attack path 2**) data comunication interfaces, or by compromissing the gateway ECU (**attack path 3**) which receives malicious control signals from OBD-2 connector.

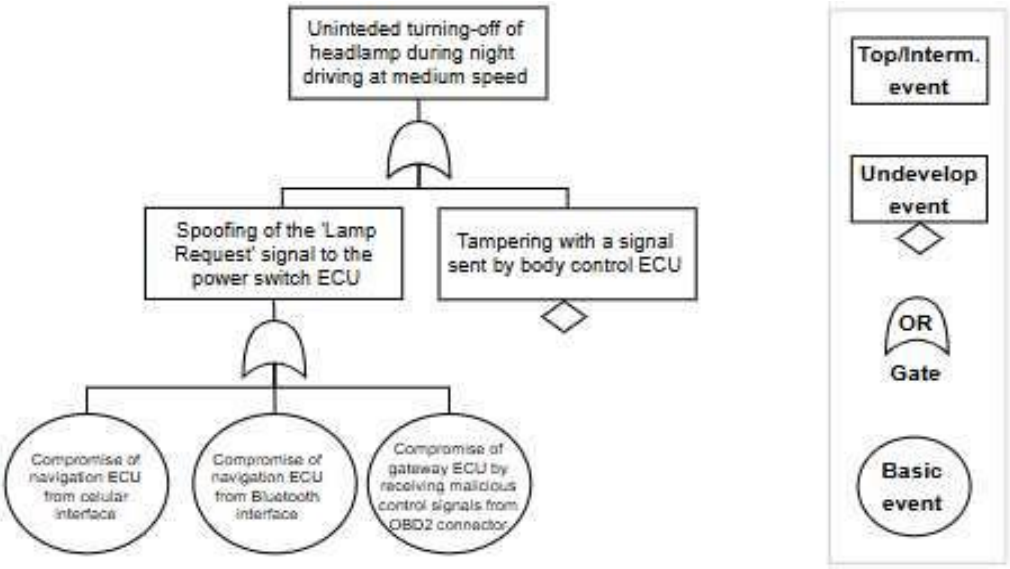


Fig. 3. Attack tree for a headlamp damage scenario.

2.3 Security Databases

This section provides an overview of Common Vulnerabilities and Exposures (CVE) [MITRE, 2025], Common Weakness Enumeration (CWE) [MITRE$_a$, 2025], and Common Attack Pattern Enumeration and Classification (CAPEC) [MITRE$_b$, 2025] security databases, their internal and external relationships highlighted in Fig 3.

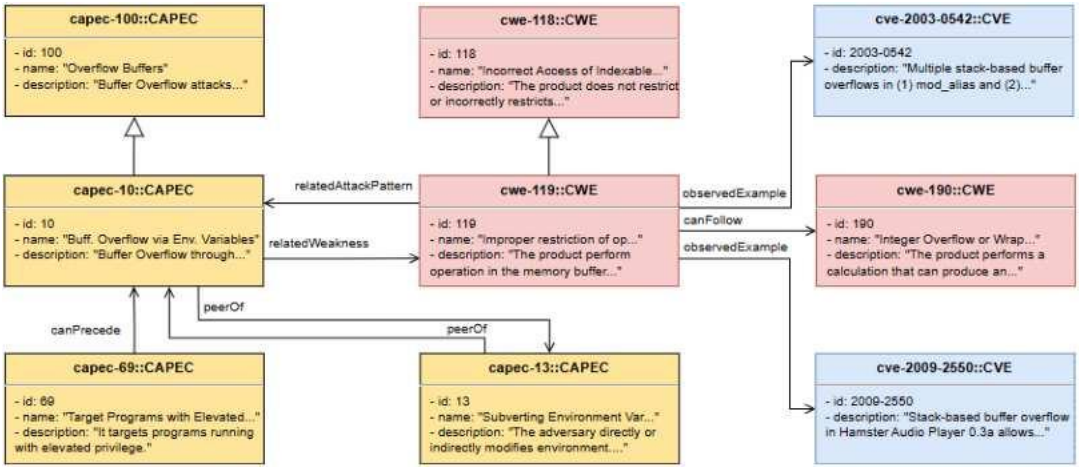


Fig. 4. Relationships between vulnerabilities, weaknesses, and attack patterns in a UML object diagram.

2.3.1 Common Vulnerability and Exposures

CVE is a catalogue of publicly known cybersecurity vulnerabilities. A vulnerability stands for a weakness in the computational logic found in software or hardware components that, when exploited by attacks, result in the violation of confidentiality, integrity, availability, or other cybersecurity property of interest [NIST, 2025]. Vulnerability is an instance of one more weaknesses in a product that can be exploited, leading to a negative impact to confidentiality, integrity, or availability, or as a set of conditions or behaviours that allows the violation of an explicit or implicit security policy. Each CVE entry, i.e., vulnerability, contains a unique identifier, a description, and at least one reference for publicly known cybersecurity vulnerabilities. Additional entry information may include fix information, severity scores, and impact ratings according to the Common Vulnerability Scoring System (CVSS), and links to exploit and advisory information [MITRE$_c$, 2025]. CVE-2003-0542 and CVE-2009-2550 vulnerabilities are observed examples of CWE-119 weakness (see Fig. 3).

2.3.2 Common Weakness Enumeration

The CWE is a community-developed source of software and hardware weakness types. Weaknesses are flaws, faults, bugs, or other errors in software, firmware, hardware, or service design, architecture, code, or implementation that if left unaddressed could result in system, network, or hardware being vulnerable to attack [MITRE$_b$, 2025]. A CWE entry has a relationship with a CVE entry. The relationship implies that a Vulnerability is an example of the (type of) Weakness. The CVE entry contains fields with severity score in CVSS, and Known Affected Hardware or Software Configurations. These Affected Product Configurations document specific software or hardware releases that are affected. Affected Product Configurations specifically are of interest since with an inventory, security operators and analysts can search for them to be aware of specific targets in their systems. CWE-118, CWE-119, and CWE-190 are examples of

weaknesses (see Fig. 3). The CWE-119 weakness is a subtype of CWE-118, which can be followed by the CWE-190 weakness. CWE-119 is a weakness that can be exploited by the CAPEC-10 related attack pattern.

2.3.3 Common Attack Pattern Enumeration Classification

The CAPEC list enumerates and classifies Attack Patterns to support security analysts in identifying and understanding attacks. Attack patterns connect attacks to CWE entries, i.e. Weaknesses, which can be exploited by potential attacks. Fig. 3 illustrates examples CAPEC attack patterns, their internal and external relationships with CWEs. The CAPEC-010 *Buffer overflow via environment variables* exploits the CWE-119. CAPEC-010 is a subtype of CAPEC-100 attack pattern, and it is the peer of CAPEC-013 *Subverting environment variables*. The CAPEC-069 *Target programs with elevated privilege* can precede the CAPEC-010 attack pattern. Expressing complex internal and external relationships between vulnerabilities, weaknesses, and attack patterns stored into different security data sources using UML object diagrams may improve readability and understanding compared to graphs [Hemberg et al., 2024, Yuan et al., 2021].

3. Integrating Vulnerability, Weakness, and Attack Pattern Databases

Here, we present the contributions of this study. We introduce the ontology for the ISO 213434 cybersecurity standard (Subsection 3.1). We present a conceptual model, specified in a UML Class Diagram, which integrates CVE, CWE, and CAPEC security databases (Subsection 3.2). The proposed model was built upon the analysis of the entities of those security databases, their internal and external relationships. This model aims to simplify the specification of complex relationships between entities from different security databases. The proposed model may support security analysts during cybersecurity analysis of automotive systems. It may also guide the development of security analysis tools to support threat analysis and risk assessment, and attack path analysis. Subsection 3.3 highlights the relationships between the ISO 21434 threat analysis and risk assessment process activities, and security databases.

3.1 ISO 21434 Ontology

Based on the analysis of the ISO 21434 Clause 9 - Concept Phase, which describes the Threat Analysis and Risk Assessment (TARA) workflow, we specified an ontology with the concepts embedded in the standard and their relationships illustrated in Fig. 4. An i**tem** stands for a **component** or a set of **components** that implement a **function** at the vehicle level. The operational environment of an **item** or a **component** refers to its operational use in a vehicle function, in production, and/or in service. An item may contain zero or more assets. An **asset** stands for an object that has value or contributes to a value. An asset could be a **system**, a **component**, **data**, or **data communication**. An asset has one or more cybersecurity properties whose compromise (by an **attack path**) can lead to one or more **damage scenarios**. A **cybersecurity property** is an attribute of an **asset** that can be worth to protect, e.g., confidentiality, integrity, availability. A **damage scenario** is an adverse consequence involving a vehicle or vehicle function, and affecting a road user. A **damage scenario** refers to the violation of a **cybersecurity property** of an **asset**. The identified **damage scenarios** are later assessed against their potential adverse consequences for road users in **impact categories** according to their **safety**, **financial**, **operational**, and/or **privacy** (S, F, O, P) impact. The **impact** is the degree of magnitude of the damage or physical harm from a

damage scenario. Additional impact categories can be considered to determine the impact rating of a damage scenario. In this case, the rationale and explanation of these categories can be shared in the supply chain in conformance with the ISO 21434 Clause 7. The **impact rating** of a **damage scenario** can be classified as severe, major, moderate, or negligible according to the assigned safety, financial, operation, and privacy impact rating criteria, derived from the ISO 26262 [**Erro! Fonte de referência não encontrada.**] functional safety standard for road vehicles. The occurrence of a damage scenario leads to a **damage** to the **road user**. A **threat scenario** is a potential cause of violation of **cybersecurity properties** of one or more **assets** that realizes a **damage scenario**. A **threat scenario** may lead to the occurrence of the multiple **damage scenarios**. For instance, **spoofing** of CAN messages for the braking ECU of a vehicle (**threat scenario**) leads to the **loss** of **integrity** (**violated cybersecurity property**) of the **CAN messages** (**asset**), and to the **loss** of **integrity** of the **braking function**.

The identification of **threat scenarios** aims to describe a set of potential actions that may lead to one or more damage scenarios, which specify the adverse consequences (results) of an attack. Techniques such as the STRIDE (**S**poofing, **T**ampering, **R**epudiation, **I**nformation disclosure, **D**enial of service, **E**levation of privilege) method can be used to support engineers to enumerate **cybersecurity threats** on the **asset** based on a DFD. Each **STRIDE** threat category is associated with the violation of a particular **cybersecurity property**. A **spoofing** threat where attackers pretend to be legitimate entities violates **authenticity**. **Tampering** threats, which attackers modify data in transit or in a data store, violates **integrity**. A **repudiation** threat where attackers perform actions that cannot be traced back to them violates **non-repudiation**. **Information disclosure** threats related to attackers that get access to data violates **confidentiality/privacy**. **Denial of service** happens when attackers interrupt a system legitimate operation, violating **availability**. **Elevation of privilege** threats correspond to attackers performing unauthorized actions, violating **authorisation**.

An **attack path** (**attack**) is a set of **deliberate actions** to realize a **threat scenario**. An **attack path** may include zero or more **attack steps**, which need to be performed for its realization. **Attack steps** are a set of actions needed to perform an **attack path**. Attack path analysis can be performed using **top-down** approaches, e.g., **attack trees** and **attack graphs**, that deduces **attack paths** by analysing the ways a **threat scenario** could be realized, or **bottom-up** approaches that build **attack paths** from the identified **vulnerabilities**. According to the ISO 21434, **vulnerability** is a **weakness** (of the **asset**) that can be exploited as part of an **attack path**. **Weakness** stands for a **fault** or **characteristic** (from the asset) that can lead to undesirable behaviour. An attack tree is a specific type of deductive failure model (e.g., Fault Tree Analysis model) that describes the **root causes** of a violation of a **cybersecurity property** (**damage scenario**) of an **asset** (e.g., system, component, data, or data communication). Attack trees are similar to fault trees used in safety modelling, facilitating the communication between safety and cybersecurity engineers. Once **damage scenarios** and **threat scenarios** have been identified, an **attack tree** is created for each damage scenario, which is the root node of the tree. **Threat scenarios** leading to those **damage scenarios** are children of the root node. Depending on the specificity

of the threat scenario, it can be split into sub-nodes, or new nodes can be added on a path to an external interface which may be the entry point for one or more **attack paths** that realize the threat scenario. An attack tree describes a set of **attack paths** (**attacks**) leading to a **damage scenario** (i.e., the top-event). Each **path** of the tree contains the potential **threat scenario** that can lead to a **damage scenario**, associated **attack path** and related **attack steps** needed for the realization of the given threat scenario. **Attack paths** and/or **attack steps** are basic events (i.e., events that cannot be decomposed) of the tree**.** Logical relationships between **attack paths**, **attack steps**, and **threats scenarios** leading to the occurrence of a **damage scenario** are described using logical AND, OR, NOT gates.

The total set of identified **attack paths** are the set of **unique paths** from the **leaf** to the **root node** for all attack trees. An attack path can be decomposed into other attack paths or **attack steps**. An **attack step** is an action needed to perform an **attack path**. Later, each identified **attack path** (**attack**) receives a **feasibility rating** (high, medium, low, or very low). The attack feasibility rating can be determined based on attack potential, Common Vulnerability Scoring System, or Attack Vector, or other approaches. In this paper, we considered the attack potential based approach, defined in the ISO/IEC 18045 [ISO/IEC, 2008] security standard, which determines the **attack feasibility rating** based on **elapsed time**, **specialist expertise**, **knowledge** of the **item** or **component**, **window of opportunity**, and **equipment** parameters.

Each **attack potential parameter** can receive one of a set of possible **values**. **Expertise** is the required knowledge to perform an attack, which can be: **Layman**, no particular expertise is required, **Proficient**, general security and domain knowledge is required, **Expert**, security and domain knowledge required, or **Multiple Experts**, where expert security and domain knowledge are required for several domains. **Knowledge of the item** stands for the availability of information and the size of the community with access to that knowledge. It can assume one of the following values: **Public**, the necessary information to perform an attack is publicly available, **Restricted**, the information is shared among partners under non-disclosure agreements, **Sensitive**, the information is shared among specific teams, or **Critical**, when the information is restricted to a few individuals.

Equipment parameter refers to the required equipment to identify or exploit hardware or software vulnerabilities. The required equipment could be: **Standard**, readily available to the attacker, **Specialized**, the equipment is not readily available to the attacker, but it could be acquired without enough effort, **Bespoke**, the equipment is not readily available to the public, and it may need to be produced, or **Multiple Bespoke**, which **multiple types** of **bespoke** equipment are required for a successful attack. **Elapsed time** is the required time to an attacker performing an attack. **Window of opportunity** represents the **access means** (**physical** or **remote**) available to the attacker, and the **time** window an attacker has to perform a successful attack. The **access means** can be: **Unlimited** physical or network access, **Large**, high physical and/or remote availability with some time constraints, **Medium**, availability with severe time constraints, i.e., limited physical and/or remote access to the asset, e.g., physical access to the vehicle interior or exterior without using any specific tool, or **Small**, low availability. The **attack potential parameter values** are assigned in the form of

numerical values ranging from 0 to 3. The weighted **attack potential parameter's** sum is the **feasibility rating** to be assigned to a given **attack path**. We assume that each **attack potential parameter** is of equal importance, and has the same weight in the **feasibility rating** calculation. The resulting sum will always be between 0 and 1.

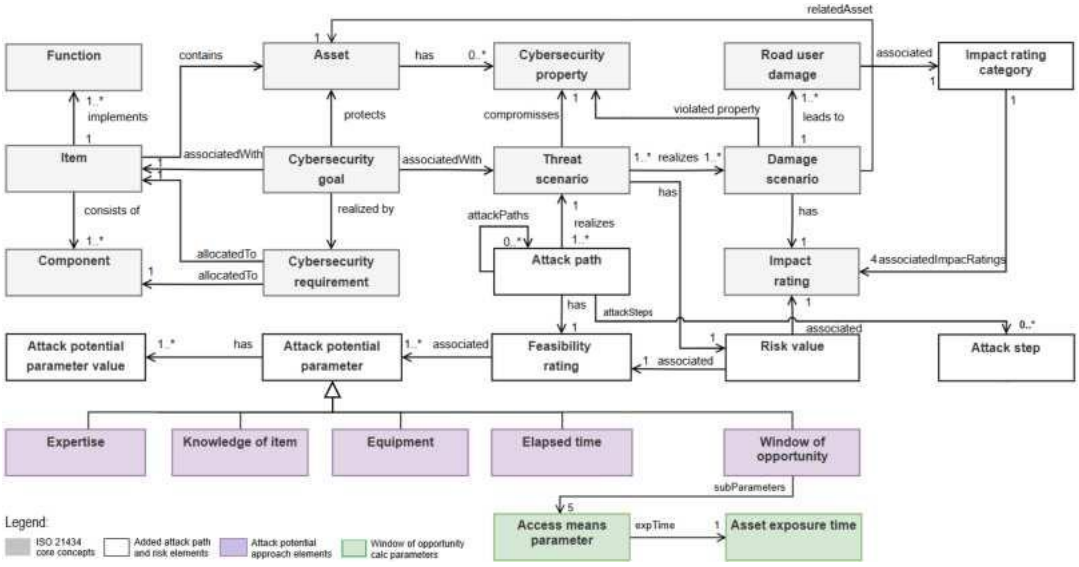


Fig. 4. The ISO 21434 cybersecurity standard ontology.

After assigning the **impact** of each identified **damage scenario** and calculating the **feasibility** of each identified **attack path**, it is needed to classify the risk posed by each identified **threat scenario**. **Risk** is a measure of the probability that the system will cause an **accident**. Each **threat scenario** should have a **risk value**, defined based on the highest **impact** assigned to an associated **damage scenario**, and the **highest feasibility rating value** assigned to an associated **attack path** leading to the threat scenario. For example, consider a threat scenario ***t1*** leading to the occurrence of two different damage scenarios (**d1** and **d2**) with **severe** and **moderate** safety impact, respectively, and two associated **attack paths** (***a1*** and ***a2***) with low and high feasibility rating. To determine **risk value** of ***t1*** threat scenario, we consider the **severe** impact assigned to ***d1*** damage scenario, and the **high** feasibility rating assigned to ***a2*** attack path. The risk value of a threat scenario is determined based on a risk matrix (see Fig. 2) comprising impact and attack feasibility rating. The ISO 21434 cybersecurity standard specifies five **risk levels** that can be assigned to a **threat scenario** according to its **impact** and **attack feasibility rating**. A risk matrix as illustrated in Fig. 2 can support the assignment of a risk level to a threat scenario. The matrix is almost symmetric except for a lower risk value for negligible impact and high attack feasibility rating. The definition of what constitutes an acceptable risk (i.e., **As Low As Reasonably Practicable** - **ALARP**) is dependent upon the project, but we can consider risk values between 1 and 2 as acceptable levels in most cases [Lautenbach et al., 2021].

Once risk values were assigned to each identified threat scenario, **cybersecurity goals/claims** and **cybersecurity requirements** are allocated to the **assets** to **avoid**, **share**, **transfer**, **accept**, or **reduce** security risks. **A cybersecurity goal** is a high-level

cybersecurity requirement associated with one or more **threat scenarios**. A cybersecurity claim is a statement about a risk, which can include a justification for retaining, sharing, or transferring the risk. A c**ybersecurity goal** can be stated in the form of a **Cybersecurity Assurance Level (CAL)**, which specifies the target level of process rigor for the **security validation process**.

3.2 Cybersecurity Metamodel

Here, we introduce the proposed cybersecurity conceptual (meta) model built upon the analysis of CVE, CWE, and CAPEC security database elements and their implicit and explicit relationships (see Fig. 5). The metamodel aims to guide engineers during threat analysis and risk assessment activities and for developing tooling support.

A CAPEC attack pattern is a common approach and attributes related to the exploitation of a weakness in a software, firmware, hardware, or service component [MITREb, 2025]. The specification of an attack pattern contains a unique identifier, a name that describes its purpose, a textual description, the level of the detail (standard or detailed) of the description (abstraction), the typical severity of the effects of the attack pattern, its likelihood of occurrence, and optionally an extended description as illustrated in Fig. 5.

An attack pattern may also include the specification of zero or multiple **execution flows**, i.e., sequence of **actions** needed to execute an **attack**. The specification of an execution flow comprises **experiment**, **explore**, and **exploit** elemen-

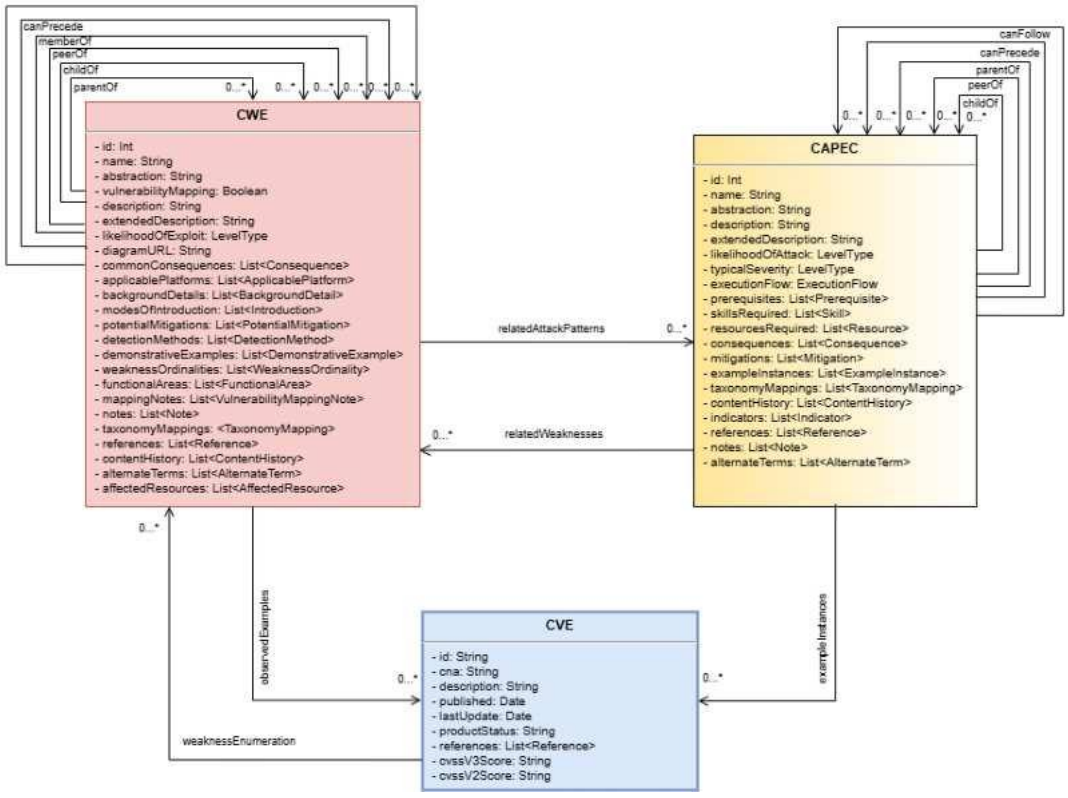


Fig. 5. CAPEC, CWE, and CVE integration metamodel.

ts. An **exploit** stands for an input or action aimed at taking advantage of a weakness (or multiple weaknesses) to achieve a negative technical impact on asset's cybersecurity properties. The existence (even if only theoretical) of an exploit is what makes a weakness a vulnerability [MITREb, 2025]. An attacker sending packets to the arget device until performance being degraded is an example of **exploit** action for the CAPEC-666: BlueSmacking attack pattern.

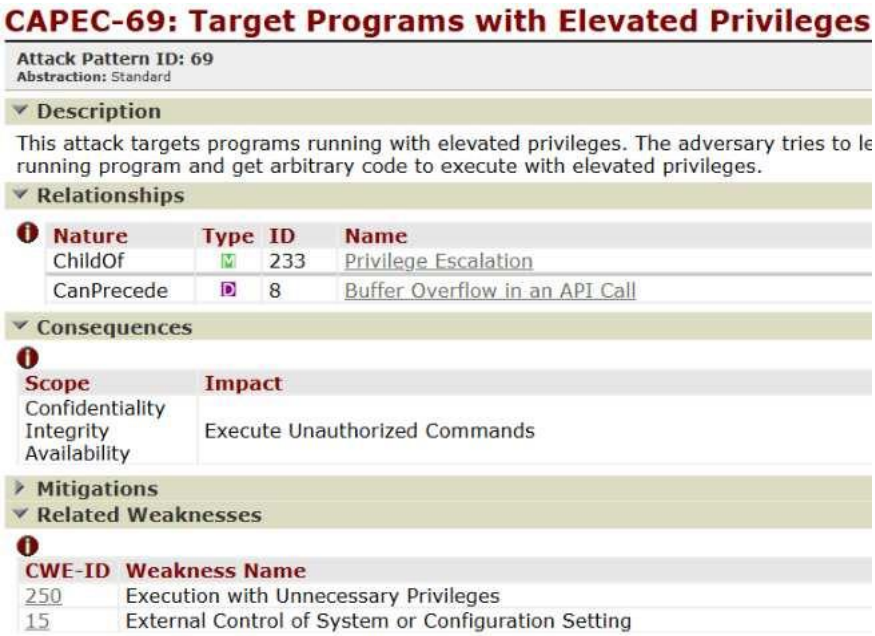
An **explore** stands for one or more actions/strategies aiming to identify vulnerabilities in the asset (software, system, communication, data). For instance, CAPEC-666 explore comprises scanning for Bluetooth enabled devices using BlueZ along with an antenna, where an attacker searchers for devices with Bluetooth on. The description of an **explore** can also include **techniques**, e.g., taking notes of the MAC address of the device intended to be attacked. An **experiment** describes potential strategies that can be performed by an adversary to execute an attack. Considering CAPEC-66 attack pattern, changing L2CAP packet length to create packets that will overwhelm a Bluetooth enabled device is an example of experiment that materializes an attack pattern.

A CAPEC attack pattern entry may also contain the specification of **prerequisites** (conditions), **skills**, and **resources** required to perform an attack as well as its **consequences**, **mitigation strategies**, and e**xamples**. In addition, it may also contain the specification of **mappings** to other security databases, e.g., OWASP and ATT&CK[3], **content history** with submission and modification dates, **indicators** concerning the synthoms of an attack is happening, **references** to documents used to describe an attack pattern, **notes**, and **alternative terms** used to refer to an attack pattern. An attack pattern may also contain references to **related attack patterns**, and **related weaknesses**. A CAPEC attack pattern can be the parent or child of another attack pattern. It can also be the peer of other attack patterns. An attack pattern can also precede or follow the execution of other attack patterns.

Fig. 6 illustrates an example of CAPEC entry. The CAPEC-69 Target Programs with Elevated Privileges attack pattern targets applications running with elevated privileges. The occurrence of such an attack pattern leads to the execution of unauthorized commands, compromising system confidentiality, integrity, and/o availability. CAPEC-69 is a child of CAPEC-233 Privilege Escalation attack pattern, and it can precede the execution of CAPEC-8 Buffer Overflow in an API call attack pattern. A CAPEC-69 attack pattern can happen in the presence of the following weaknesses: CWE-250 Execution with Unnecessary Privileges, or CWE-15 External Control of System or Configuration Setting.

CAPEC database organizes attack patterns into categories. For instance, CAPEC-125 Flooding and CAPEC-130 Excessive Allocation are attack pattern categories related to the STRIDE Denial of Service security threat (see Fig. 7). CAPEC-489 SSL flood and 666 Blue Smacking are subtypes of Flooding attack pattern category. There exist relationships between other CAPEC attack pattern and STRIDE model threat categories as illustrated in Table 1. CAPEC 151 and 148 attack pattern categories are related to spoofing security threat. DSN Rebuilding is an example of spoofing attack

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



CAPEC-69: Target Programs with Elevated Privileges

Attack Pattern ID: 69
Abstraction: Standard

Description

This attack targets programs running with elevated privileges. The adversary tries to le running program and get arbitrary code to execute with elevated privileges.

Relationships

Nature	Type	ID	Name
ChildOf	V	233	Privilege Escalation
CanPrecede	D	8	Buffer Overflow in an API Call

Consequences

Scope	Impact
Confidentiality Integrity Availability	Execute Unauthorized Commands

Mitigations

Related Weaknesses

CWE-ID	Weakness Name
250	Execution with Unnecessary Privileges
15	External Control of System or Configuration Setting

Fig. 6. CAPEC-69 attack pattern entry [MITREb, 2025].

pattern related to CAPEC-194 Fake the Source of Data subcategory from 194 Identity Spoofing attack pattern category.

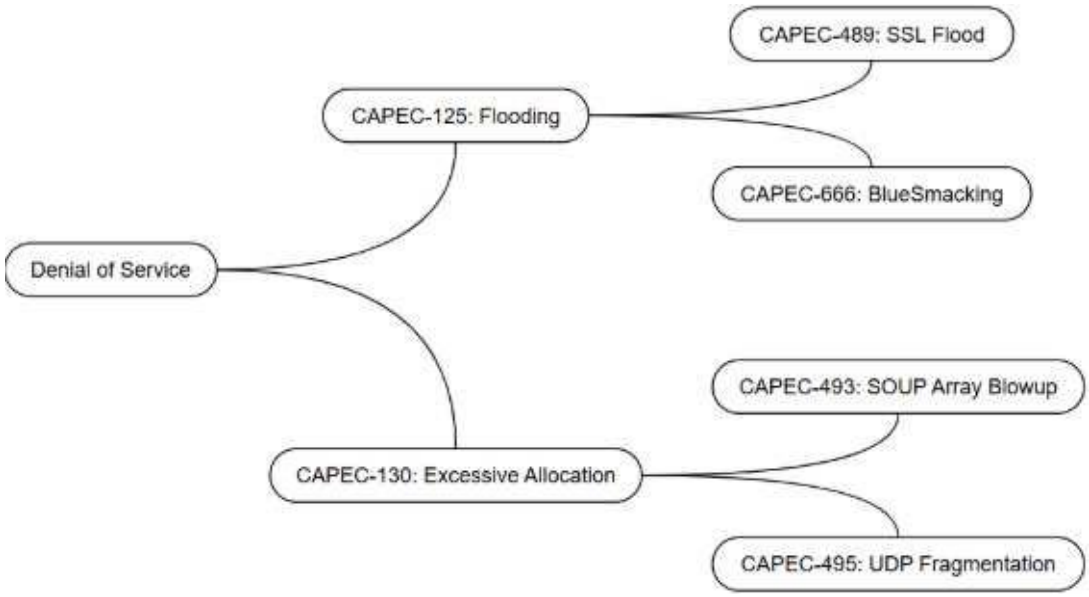


Fig. 7. Example of STRIDE, CAPEC attack patterns and categories.

Table 1. Relationships between STRIDE threat categories and CAPEC security database.

STRIDE Threat Category	CAPEC Category/Subcategory	CAPEC Attack Pattern
Spoofing	151: Identity Spoofing → 194: Fake the Source of Data	275: DNS Rebuilding
	148: Content Spoofing	145: Checksum Spoofing
Tampering	123: Buffer Manipulation	540: Over Read Buffers
	123: Buffer Manipulation → 100: Overflow Buffers	256: SOAP Array Overflow
Repudiation	268: Audit Log Manipulation	81 Web Logs Tampering
Information disclosure	212: Functionality Misuse	111: JSON Hijacking
Denial of service	125: Flooding	489: SSL Flood
		666: Blue Smacking
	130: Excessive Allocation	493: SOUP Array Blowup
		495: UDP Fragmentation
Elevation of Privilege	549: Local Code Execution → 542: Targeted Malware	552: Install Rootkit

The specification of a CWE entry contains a unique **identifier**, a **name** that describes the weakness, a textual **description**, the **level of the detail** (standard or detailed) of the description (**abstraction**), a Boolean variable indicating whether or not **vulnerability mapping** is allowed, its **likelihood** of exploit (low, medium, or high), and optionally an **extended description** as highlighted in Fig. 5. A CWE entry may also include a diagram/picture (i.e., **diagramURL**) as part of its description, its **common consequences** in terms of their impact and violation of cybersecurity properties if exploited, the **applicable platforms** (e.g., targeting technologies, languages), **background details** (needed to understand the weakness) and **modes of introduction** (i.e., implementation, installation, architectural design, and operation phases), potential **mitigation** strategies and **detection methods** (e.g., manual analysis, architecture and design review), as well as **demonstrative examples**. A CWE entry may also include selected **observed examples,** i.e., references to **Common Vulnerabilities and Exposures** entries, which support users to understand a variety of ways in which a given weakness can be introduced. The selected **observed examples** element is a list of all CVEs related to a CWE entry. For instance, CVE-2020-3812, *mail program runs as root but does not drop its privileges before attempting to access a file,* is a concrete example of CWE-250 Execution with Unnecessary Privileges.

A CWE entry may contain the description **functional areas** in terms of threatened cybersecurity properties, **vulnerability mapping notes** concerning CWE usage, reason, rationale, and/or comments, specific **notes** about relationships (association, overlapping) with other CWEs, mappings to weaknesses listed in other databases (**taxonomy mappings**), **references** cited in its description, **content history** with submission and modification dates, **alternative terms** used to refer to a weakness, and the **affected resources**. A CWE entry may also contain references to r**elated attack patterns**. For instance, the CWE-250 Execution with Unnecessary Privileges contains a reference to the CAPEC-69 attack pattern. A CWE can be the parent, child, peer, or member of other CWEs. A CWE can also precede or follow other CWEs.

Finally, a CVE entry contains a unique identifier, comprising CVE-YYYY-NNNN, a Cve Numbering Authority (CNA), publication and last update dates, a textual description, the product status, references to external documents, and its associated

Common Vulnerability Scoring System (CVSS) v2 and v3 scores. The product status comprises its name, vendor, and versions. CVE-2020-3812 is an example of CVE entry reported by Debian GNU/Linux CNA, related to qmail-verify used in netqmail 1.06 which is prone to information disclosure vulnerability. It may allow a local attacker to verify files and directories anywhere in the filesystem. A CVE entry can refer to related **weaknesses**. For instance, CVE-2009-2550, related to the stack-based buffer overflow in Hamster Audio Player 3.0 refers to CWE-787 Out-of-bounds write weakness.

3.3 The ISO 21434 Assurance Process and Cybersecurity Databases

The relationships between ISO 21434 cybersecurity concepts, security database elements, and security analysis techniques are highlighted in Table 3. CVE catalogue can support the identification of damage scenarios for the assets in conformance with ISO 21434 Part 9.4. Since CVE entries are concrete examples of CWE weaknesses. The impact rating of a damage scenario can be estimated with the support of CVE entries CVSS score, and ISO 21434 safety, financial, operational, and privacy impact categories. STRIDE model, CWE weaknesses catalogue, and the consequence attribute of CAPEC attack patterns can guide the identification of threat scenarios. Attack trees can be used together with CWE and CAPEC attack patterns catalogues to support attack path analysis to identify the root causes of each threat scenario.

CAPEC attack pattern execution flow can be used to detail the steps of an attack. The feasibility of each identified attack path can be determined using Attack Potential-based approach, Attack Vector-based approach, or Common Vulnerability Scoring System (CVSS) method. The likelihood attribute of CAPEC attack patterns can be used to determine the feasibility of attack paths. Skill and Asset Exposure Time attributes of a CAPEC attack pattern entry, and CWE Likelihood of exploit correspond to Expertise, Equipment, and Window of Opportunity attack feasibility parameters from Attack Potential-based approach, respectively. Based on CVSS information from CVE entries and CAPEC attack pattern feasibility parameters, risk determination can be performed to classify the risk posed by each identified threat scenario. A risk matrix combining the impact of damage scenarios and feasibility of attack paths associated with a threat scenario can be used to classify its risk in terms of a cybersecurity assurance level, which defines the level of process rigor.

4. Evaluation

Here, we describe how cybersecurity database entries can support attack path analysis for a damage scenario concerning data communication related to the lamp request (asset) of an automotive headlamp system from the ISO 21434 standard. The headlamp system turns on/off the headlamp in accordance with the switch by demand of the driver. If the headlamp is in high-beam mode, the system switches the headlamp automatically to the low-beam mode when an oncoming vehicle is detected by the camera. It also returns the headlamp automatically to the high-bean mode if the oncoming vehicle is no longer detected.

Table 2. ISO 21434, security databases, and techniques.

ISO 21434 Concept	**Security Database Element**	**Security analysis methods**
Damage scenario	CVE Entries	ISO 21434 Part 9.4 TARA
Impact rating	CVE CVSS score	ISO 21434 safety, financial, operational, and privacy impact categories.
Threat scenario	CAPEC Consequence, CWE Description and Common Consequences	STRIDE
Attack path	CAPEC Attack Pattern	Attack Tree Analysis
Attack step	CAPEC Execution Flow	
Feasibility rating	CAPEC Likelihood of Attack	Attack Potential-based approach, Attack Vector-based approach, or CVSS
Expertise	CAPEC Skill	Attack Potential based approach
Equipment	CAPEC Asset Exposure Time	
Window of Opportunity	CWE Likelihood of exploit	

We considered the following damage scenario: *Uunintended turning-off of headlamp system during night driving at medium speed* related to lamp request, violating *integrity*, with a **severe** (**S3**) safety impact, leading to front collision with a narrow stationary object like a tree. We identified the potential threat scenarios leading to the damage scenario: ***Spoofing*** of the lamp request signal to the power switch ECU, or ***tampering***

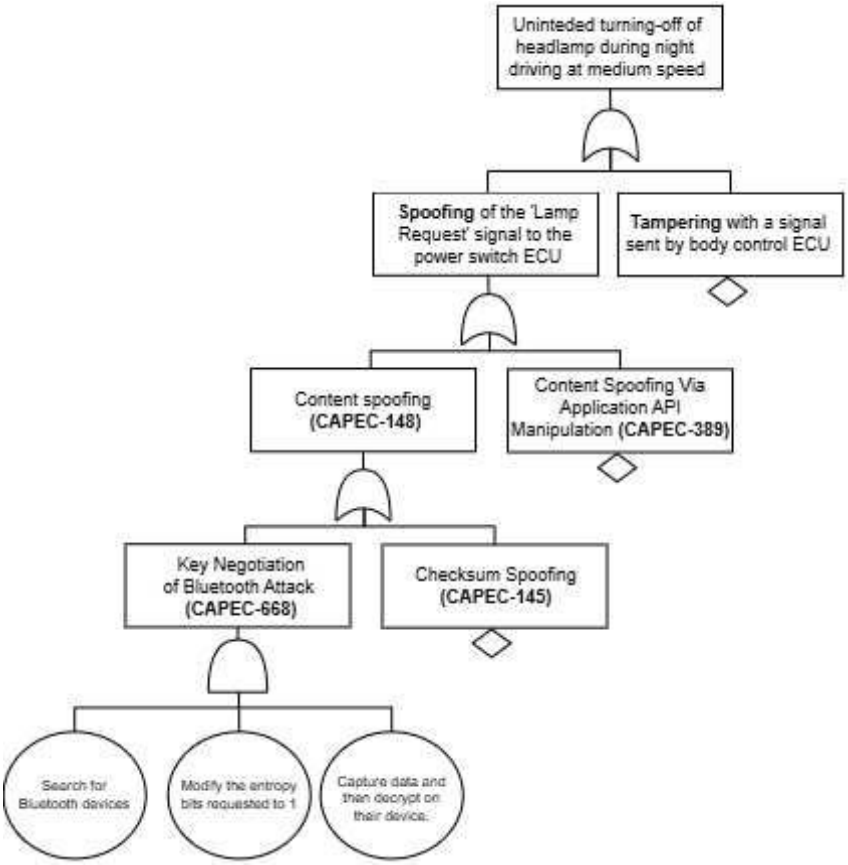


Fig. 8. Attack tree for a headlamp damage scenario.

with a signal sent by the body control ECU potentially causes the headlamp to turn-off unintentionally. An ***spoofing*** of the lamp request signal (threat scenario) can happen when an attacker compromisses the navigation ECU via cellular or Bluetooth data comunication interfaces, or by compromissing the gateway ECU which receives malicious control signals from OBD2 connector illustrated in the attack tree model from Fig. 8, specified in the Fault Tree Analysis visual notation. The spoofing of the lamp request signal can be realized by a CAPEC-668 Key negotiation of Bluetooth, CAPEC-145 Checksum spoofing, or by a content spoofing via application API manipulation (CAPEC-389). CAPEC-668 attack path is detailed into three attack steps: search for Bluetooth devices, modify the entropy bits requested to 1, and capture data and then decription on their device.

5. Conclusion

In this study, we introduced a conceptual metamodel that integrates CAPEC, CWE, and CVE security databases, built upon the ISO 213434 ontology. We also described the relationships between ISO 21434 cybersecurity concepts, cybersecurity database elements and attributes, and security analysis methods. The main contributions include the development of a formal ontology to represent core ISO 21434 entities, a metamodel capturing the semantic relationships among security databases, and a systematic mapping between TARA activities and database elements. The metamodel may support engineers during security analysis, and the development of tooling support for attack path analysis. Despite these benefits, the current security databases present weak interconnection between CVE entries and their corresponding CAPEC and CWE representations. In both CAPEC and CWE, CVEs are typically referenced only as examples, lacking formal, bidirectional links that capture contextual relationships. One potential direction for future work is to incrementally enrich these relationships through machine-assisted inference or expert curation, enabling more precise mappings between vulnerabilities, weaknesses, and attack techniques. Moreover, the integration of complementary knowledge bases such as MITRE ATT&CK, NVD metadata, or vendor-specific repositories may increase the coverage threat modelling. We also intend to implement tool support for the proposed metamodel and evaluate it through real-world case studies in the automotive sector.

References

Biro, M., Mashkoor, A., Sametinger, J., Seker, R.: Software safety and security risk mitigation in cyber-physical systems. IEEE Software. 35 (1), 24–29, 2017.

Dantas, Y. G., Nigam, V., Ruess, H. Security Engineering for ISO 21434. Fortiss GmbH White Paper, Munich, Germany, 2020.

Hemberg, E., Turner, M. J., Rutar, N., and O'reilly, U. M. 2024. Enhancements to Threat, Vulnerability, and Mitigation Knowledge for Cyber Analytics, Hunting, and Simulations. **Digital Threats** 5, 1, Article 8, 2024.

Hernan S., Lambert S., Ostwald T., Shostack A. Threat Modeling - Uncover Security Design Flaws Using the STRIDE Approach, MSDN Mag. 2006.

Huq, N. Automotive Cyber Security - Emerging Risks and New Case Study Insights. *ATZ Electron Worldw* **19**, 14–19 (2024). **DOI**: https://doi.org/10.1007/s38314-024-1890-0.

ISO/SAE 21434: Road Vehicles – Cybersecurity Engineering, ISO/TC 22/SC 32, 2021.

ISO 26262. Road Vehicles— Functional Safety – Part 1: Vocabulary, ISO/TC 22/SC 32, 2018.

ISO/IEC. ISO/IEC 18045:2008 – Information technology – Security techniques – Methodology for IT security evaluation, 2008.

MITRE. Common Vulnerabilities and Exposure. Online: https://cve.mitre.org/.

MITREa. Common Weakness Enumeration. Online: https://cwe.mitre.org/.

MITREb. Common Attack Pattern Enumeration and Classification. Online: https://capec.mitre.org/.

MITRE. Cve and nvd relationship. Online: https://cve.mitre.org/about/cve_and_nvd_relationship.html.

NASA. “Fault Tree Analysis Handbook for Aerospace Applications”. WA, USA, 2002.

Retouniotis, A., Papadopoulos, Y., Sorokos, I. Andromeda: A model-connected framework for safety assessment and assurance, Journal of Systems and Software, v. 220, 2025, 112256.

Schneier, B. Modeling security threats. Dr. Dobb’s J. 24 (12), 1999.

Upstream. Global Automotive Cybersecurity Report. Upstream Security Ltd., 2024. Online:https://info.upstream.auto/hubfs/Security_Report/Security_Report_2024/Upstream_2024_Global_Automotive_Cybersecurity_Report.pdf.

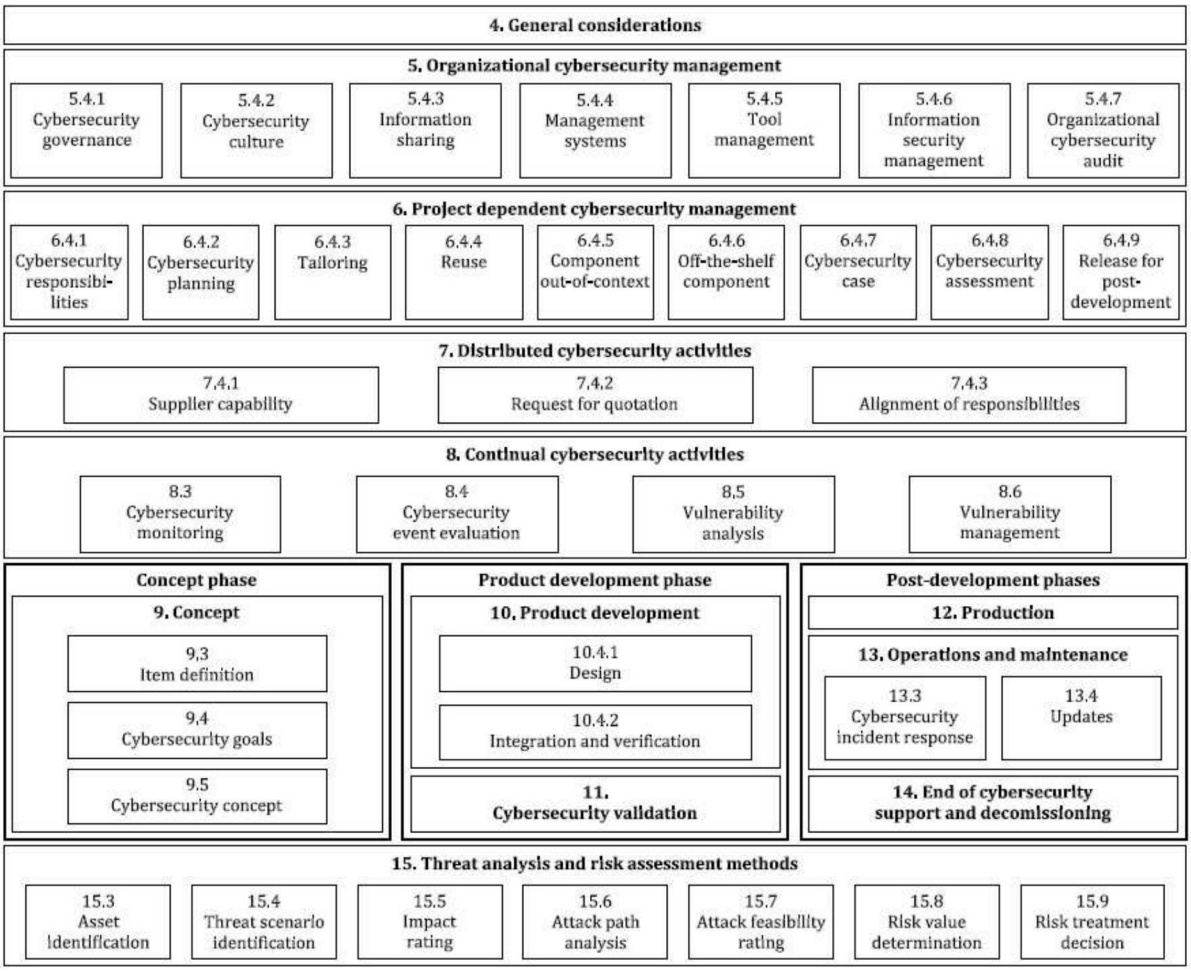
Wired. Hackers remotely kill a Jeep on the highway with me in it, 2015. Online: https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/.

Yuan, L., Bai, Y., Xing, Z., Chen, S., Li, X., and Deng, Z. "Predicting Entity Relations across Different Security Databases by Using Graph Attention Network," 2021 IEEE 45th COMPSAC, Madrid, Spain, 2021, pp. 834-843.

ANEXO A – ISO/SAE 21434

O documento ISO/SAE 21434 é composto por 15 seções e 8 anexos. As duas primeiras seções definem o escopo e as referências normativas. A Seção 3 contém uma série de definições para criar um glossário comum para a segurança cibernética automotiva, sendo uma das principais inovações esperadas da ISO/SAE 21434 [77]. As seções subsequentes são chamadas de "cláusulas", onde os requisitos (RQ), recomendações (RC) e produtos de trabalho (WP) são definidos. A Cláusula 4 descreve o ecossistema do veículo, o gerenciamento de segurança cibernética organizacional e o ciclo de vida automotivo. A Cláusula 5 inclui descrições sobre a estratégia, política e objetivos de segurança cibernética organizacional. A Cláusula 6 abrange a gestão da segurança cibernética e suas atividades no nível de projeto. A Cláusula 7 apresenta os requisitos para atribuição de responsabilidades para atividades de segurança cibernética entre cliente e fornecedor. A Cláusula 8 fornece informações de avaliações de risco contínuas, definindo o gerenciamento de vulnerabilidades de sistemas E/E (elétricos e eletrônicos) até o fim do suporte de segurança cibernética. A Cláusula 9 determina os riscos, metas e requisitos de segurança cibernética para um item. A Cláusula 10 define as especificações de segurança cibernética, além de implementar e verificar seus requisitos. A Cláusula 11 realiza a validação de segurança cibernética de um item no nível do veículo. A Cláusula 12 aborda os aspectos relacionados à segurança cibernética de fabricação e montagem de um item ou componente. A Cláusula 13 apresenta atividades relacionadas à resposta a incidentes de segurança cibernética e atualizações de um item ou componente. A Cláusula 14 inclui considerações de segurança cibernética para o fim do suporte e descomissionamento de um item ou componente. Por fim, a Cláusula 15 identifica métodos modulares para realizar a análise de ameaças e avaliação de riscos (TARA) para determinar a extensão do risco de segurança cibernética, para que o tratamento de risco seja realizado.

Figura 18 – Visão geral da estrutura da ISO/SAE 21434



Fonte: ISO/SAE 21434-Road Vehicles – Cybersecurity engineering [8]

Como já mencionado, a ISO/SAE 21434 destina-se à aplicação em veículos rodoviários, se concentrando em definir critérios mínimos para engenharia de segurança cibernética automotiva. No padrão, não são fornecidas especificações para tecnologias de segurança cibernética, soluções ou métodos de remediação. Também não são fornecidos requisitos exclusivos para veículos autônomos ou infraestrutura rodoviária. Por este motivo, é incentivada uma abordagem orientada ao risco para priorizar ações e definir de forma sistemática medidas de segurança cibernética [9].

Classificação de Impacto

O anexo F da ISO apresenta diretrizes para a avaliação da classificação de impacto. Este anexo fornece exemplos de critérios envolvendo cenários de dano considerando quatro domínios principais: *safety*, financeiro, operacional e privacidade. Embora os exemplos fornecidos pela norma não considerem aspectos como a escalabilidade do dano (extensão do impacto a múltiplos usuários da estrada em um mesmo cenário de dano), tais fatores podem ser incorporados conforme o contexto da organização, conforme apropriado. Os

quadros a seguir apresentam os critérios de classificação para cada uma das categorias de impacto mencionadas.

Quadro 17 – Critérios de classificação de impacto de *safety*

Class. impacto	**Critérios para classificação de impacto de *safety***
Severe	S3: Lesões com risco de vida (incertas de sobrevivência), lesões fatais
Major	S2: Lesões graves e com risco de vida (provável sobrevivência)
Moderate	S1: Lesões leves e moderadas
Negligible	S0: Sem lesão[a]
[a] A classificação para S0 pode ser baseada na ISO 26262-3:2018, Tabela B.1.	

Fonte: Adaptado de ISO [8]

A classificação de impacto em *safety* (Quadro 17) é baseada nos níveis de gravidade das lesões provocadas por um cenário de dano, conforme estabelecido na ISO 26262-3:2018 [50]. É possível, ainda, considerar variáveis como controlabilidade e exposição, desde que uma justificativa técnica seja fornecida. Essa abordagem visa garantir que o impacto à integridade física dos usuários da estrada seja tratado com a devida prioridade na análise de risco.

Quadro 18 – Critérios de classificação de impacto financeiro

Class. impacto	**Critérios para classificação de impacto financeiro**
Severe	Os danos financeiros levam a consequências catastróficas que o usuário afetado pode não superar
Major	Os danos financeiros levam a consequências substanciais que o usuário da estrada afetado poderá superar
Moderate	Os danos financeiros levam a consequências inconvenientes que o usuário da estrada afetado poderá superar com recursos limitados
Negligible	Os danos financeiros não levam a nenhum efeito, consequências insignificantes ou são irrelevantes para o usuário da estrada

Fonte: Adaptado de ISO [8]

No caso do impacto financeiro (Quadro 18), os critérios são definidos com base na magnitude das perdas econômicas sofridas pelo usuário afetado. Os níveis variam desde prejuízos irrelevantes até consequências catastróficas com potenciais danos irreversíveis.

Quadro 19 – Critérios de classificação de impacto operacional

Class. impacto	**Critérios para classificação de impacto operacional**
Severe	O dano operacional leva à perda ou comprometimento de uma função principal da função EXEMPLO 1: Veículo não funcionando ou mostrando comportamento inesperado das funções principais, como direção autônoma para um local não intencional
Major	O dano operacional leva à perda ou comprometimento de uma função importante do veículo EXEMPLO 2: Aborrecimento significativo do motorista
Moderate	O dano operacional leva à degradação parcial de uma função de veículo EXEMPLO 3: Satisfação do usuário afetada negativamente
Negligible	O dano operacional leva ao não comprometimento ou comprometimento não perceptível de uma função do veículo

Fonte: Adaptado de ISO [8]

A classificação de impacto operacional (Quadro 19) foca nas consequências sobre as funcionalidades do veículo. Desde comprometimentos severos que afetam funções principais como falhas de navegação autônoma até degradações parciais que afetam apenas a satisfação do usuário. Os critérios operacionais podem ou não ter consequências para o *safety*.

Quadro 20 – Critérios de classificação de impacto de privacidade

Class. impacto	**Critérios de classificação de impacto de privacidade**
Severe	O dano à privacidade leva a um impacto significativo ou até irreversível para o usuário da estrada As informações sobre o usuário da estrada são altamente sensíveis e fáceis de vincular a um PII principal
Major	O dano à privacidade leva a um sério impacto no usuário da estrada. As informações sobre o usuário da estrada são: a) altamente sensível e difícil de vincular a um PII principal; b) sensível e fácil de vincular a um PII principal
Moderate	Os danos à privacidade levam a consequências inconvenientes para o usuário da estrada. As informações sobre o usuário da estrada são: a) sensível, mas difícil de vincular a um PII principal; ou b) não sensível, mas fácil de vincular a um PII principal
Negligible	O dano à privacidade leva a nenhum efeito, ou consequências insignificantes, ou é irrelevante para o usuário da estrada As informações sobre o usuário da estrada não são sensíveis e difíceis de vincular a um PII principal

Fonte: Adaptado de ISO [8]

Já o impacto à privacidade (Quadro 20) considera o grau de sensibilidade das informações comprometidas, bem como a facilidade com que podem ser associadas a dados de identificação pessoal (PII). As informações de PII e o PII principal podem ser definidos de acordo com a norma ISO/IEC 29100 [67].

Classificação da Viabilidade de Ataque

O anexo G da norma fornece diretrizes para realizar a estimativa da classificação da viabilidade de ataque através de três abordagens: potencial de ataque, CVSS e vetor de ataque. Embora o método baseado no CVSS seja citado como uma alternativa válida, ele não foi abordado neste anexo devido ao exemplo apresentado no Anexo H não utilizar essa abordagem como critério de avaliação. Além das abordagens, considerações se um ataque pode ser escalonado (facilidade de replicação em múltiplas instâncias ou alvos) podem ser incluídas na classificação de viabilidade do ataque.

O potencial de ataque, conforme definido na norma ISO/IEC 18045 [51], representa uma métrica do esforço necessário para comprometer um item ou componente. Essa medida é expressa com base nos recursos e conhecimentos exigidos do invasor, considerando cinco parâmetros principais: tempo decorrido (*elapsed time*), experiência especializada (*specialist expertise*), conhecimento do item ou componente (*knowledge of the item or component*),

janela de oportunidade (*window of opportunity*) e equipamento (*equipment*). Os quadros a seguir apresentam os parâmetros para definir o potencial de ataque.

Quadro 21 – Tempo decorrido

Tempo decorrido
$\leq$ 1 dia
$\leq$ 1 semana
$\leq$ 1 mês
$\leq$ 6 meses
$>$ 6 meses

Fonte: Adaptado de ISO [8]

O parâmetro tempo decorrido (Quadro 21) representa a quantidade de tempo necessário para identificar uma vulnerabilidade, desenvolver um *exploit* e aplicá-lo com sucesso. Essa estimativa deve considerar o estado atual do conhecimento técnico disponível no momento da classificação, refletindo o grau de esforço temporal exigido para a execução do ataque.

Quadro 22 – Experiência especializada

Experiência especializada
Layman: Sem conhecimento algum em comparação com especialistas ou pessoas proficientes, sem especialização específica EXEMPLO 1: Pessoa comum usando descrições passo-a-passo de um ataque que está disponível publicamente
Proficient: Algum conhecimento, familiaridade com o comportamento de segurança do produto ou tipo de sistema EXEMPLO 2: Proprietário experiente, técnico comum que conhece ataques simples e populares, como ajuste do odômetro, instalação de peças falsificadas
Expert: Familiaridade com os algoritmos, protocolos, *hardware*, estruturas, comportamento de segurança, princípios e conceitos de segurança empregados, técnicas e ferramentas para a definição de novos ataques, criptografia, ataques clássicos para o tipo de produto, métodos de ataque, entre outros, implementados no produto ou tipo de sistema EXEMPLO 3: Técnico ou engenheiro experiente
Multiple experts: Diferentes campos de especialização (nível de especialista) são necessários para etapas distintas de um ataque EXEMPLO 4: Vários engenheiros altamente experientes que têm experiência em diferentes áreas e que são necessários em um nível de especialista para etapas distintas de um ataque

Fonte: Adaptado de ISO [8]

O parâmetro experiência especializada (Quadro 22) está associado ao nível de conhecimento técnico exigido pelo atacante. Esse critério abrange desde indivíduos sem formação técnica até equipes compostas por especialistas altamente qualificados em diferentes domínios. A classificação busca refletir a complexidade das habilidades envolvidas na execução das diversas etapas do ataque.

Quadro 23 – Conhecimento do item ou componente

Conhecimento do item ou componente
Public information: Informações públicas sobre o item ou componente, obtidos na internet EXEMPLO 1: Informações e documentos publicados na página inicial do produto ou em um fórum da Internet
Restricted information: Informações restritas sobre o item ou componente, conhecimento controlado na organização EXEMPLO 2: Documentação interna compartilhada entre fabricante e fornecedor, requisitos e especificações de design, dados sob contrato de não divulgação
Confidential information: Informações confidenciais sobre o item ou componente, conhecimento compartilhado entre equipes específicas EXEMPLO 3: Informações relacionadas a imobilizadores de carro, código-fonte de software
Strictly confidential information: Informações estritamente confidenciais sobre o item ou componente, onde apenas alguns indivíduos possuem conhecimento EXEMPLO 4: Calibrações específicas do cliente, mapas de memória documentados internamente pelo fabricante e/ou fornecedor

Fonte: Adaptado de ISO [8]

O parâmetro conhecimento do item ou componente (Quadro 23) avalia o grau de familiaridade do atacante com o sistema-alvo. Essa familiaridade pode variar entre informações disponíveis publicamente até conhecimentos confidenciais altamente restritos. Quanto mais específicas e protegidas forem as informações necessárias, maior será o esforço exigido para realizar o ataque.

Quadro 24 – Janela de oportunidade

Janela de oportunidade
Unlimited: Alta disponibilidade via rede pública/não confiável sem limitação (ou seja, o ativo é sempre acessível). Acesso remoto sem presença física ou limitação de tempo, bem como acesso físico ilimitado ao item ou componente EXEMPLO 1: Ataque remoto (por exemplo, interfaces de V2X ou celular) sem pré-condições, acesso físico ilimitado pelo proprietário para ajuste de chip
Easy: Alta disponibilidade e tempo de acesso limitado. Acesso remoto sem presença física ao item ou componente EXEMPLO 2: Tempo de emparelhamento de Bluetooth, atualização remota de software, ataque remoto que requer o veículo parado
Moderate: Baixa disponibilidade do item ou componente. Acesso físico e/ou lógico limitado. Acesso físico ao interior ou exterior do veículo sem usar nenhuma ferramenta especial EXEMPLO 3: O invasor entra em um carro desbloqueado e obteve acesso à interface física exposta, como por exemplo, acesso físico via porta de diagnóstico de bordo
Difficult: Disponibilidade muito baixa do item ou componente. Nível impraticável de acesso ao item ou componente para executar o ataque EXEMPLO 4: Decapando um circuito integrado para extrair informações, quebrando uma chave criptográfica por força bruta mais rápida que a chave é girada

Fonte: Adaptado de ISO [8]

O parâmetro janela de oportunidade (Quadro 24) considera o contexto necessário para que o atacante execute o ataque com sucesso, incluindo a duração, o tipo de acesso (lógico ou físico) e as restrições impostas pelo ambiente do sistema. Essa janela pode variar desde acessos remotos ilimitados até situações que requerem presença física prolongada ou invasiva.

Quadro 25 – Equipamento

Equipamento
Standard: O equipamento está prontamente disponível para o atacante. Este equipamento pode fazer parte do próprio produto (por exemplo, um depurador em um sistema operacional) ou pode ser facilmente obtido (por exemplo, fontes da internet, analisador de protocolo ou *scripts* de ataque simples) EXEMPLO 1: Laptop, adaptador CAN, dongle de diagnóstico a bordo, ferramentas comuns (chave de fenda, ferro de solda, alicate)
Specialized: O equipamento não está prontamente disponível para o atacante, mas pode ser adquirido sem esforço indevido. Isso pode incluir a compra de quantidades moderadas de equipamento (por exemplo, ferramentas de análise de energia, o uso de centenas de PCs vinculados na internet se enquadrariam nessa categoria) ou o desenvolvimento de *scripts* ou programas de ataque mais extensos. Se bancos de teste claramente diferentes que consistem em equipamentos especializados forem necessários para etapas distintas de um ataque, isso seria classificado como *Bespoke* EXEMPLO 2: Dispositivo de depuração de *hardware* especializado, dispositivos de comunicação em veículos (*hardware* no equipamento de teste de loop, osciloscópio de alta qualidade, gerador de sinal), produtos químicos especiais
Bespoke: O equipamento é produzido especialmente produzido para o propósito (por exemplo, software muito sofisticado), não estando prontamente disponível ao público (por exemplo, mercado negro), ou o equipamento é tão especializado que sua distribuição é controlada, possivelmente até restrita. Como alternativa, o equipamento é muito caro EXEMPLO 3: Ferramentas restritas ao fabricante, microscópio eletrônico
Multiple bespoke: Diferentes tipos de equipamentos sob medida são necessários para etapas distintas de um ataque

Fonte: Adaptado de ISO [8]

O parâmetro equipamento (Quadro 25) diz respeito às ferramentas necessárias para descobrir ou explorar a vulnerabilidade. A classificação considera desde equipamentos amplamente acessíveis no mercado até soluções altamente especializadas ou customizadas, cuja obtenção pode envolver altos custos ou acesso restrito.

Com base nesses cinco parâmetros, é possível atribuir valores numéricos, conforme definido na ISO/IEC 18045 [51]. A soma desses valores compõe o potencial de ataque total de um cenário. O Quadro 26 apresenta a escala de pontuação atribuída a cada nível dos parâmetros.

Quadro 26 – Agregação do potencial de ataque

Tempo dec.		**Exp. especial.**		**Conhec. do i./c.**		**Jan. de oport.**		**Equipamento**	
Enum.	**V.**	**Enum.**	**V.**	**Enum.**	**V.**	**Enum.**	**V.**	**Enum.**	**V.**
$\leq$ 1 day	0	Layman	0	Public	0	Unlimited	0	Standard	0
$\leq$ 1 week	1	Proficient	3	Restricted	3	Easy	1	Specialized	4
$\leq$ 1 month	4	Expert	6	Confident.	7	Moderate	4	Bespoke	7
$\leq$ 6 mont.	17	Mult. exp.	8	Stric. conf.	11	Diff./none	10	Mult. bes.	9
$>$ 6 mont.	19								

Fonte: Adaptado de ISO [8]

Por fim, o Quadro 27 estabelece os intervalos numéricos para a classificação da viabilidade de ataque: quanto menor o valor, maior a viabilidade do ataque. Tais valores foram inspirados na ISO/IEC 18045 [51].

Quadro 27 – Mapeamento de potencial de ataque

Class. da viab. de ataque	Valores
High	0 - 9
	10 - 13
Medium	14 - 19
Low	20 - 24
Very low	$\geq$ 25

Fonte: Adaptado de ISO [8]

Como complemento às abordagens para determinar a classificação da viabilidade de ataque, a ISO 21434 também propõe a utilização da baseada em vetor de ataque. Essa perspectiva considera o contexto pelo qual a exploração de um caminho de ataque pode ocorrer, atribuindo maior viabilidade a ataques que possam ser realizados sem a necessidade de acesso físico ao sistema-alvo. Quanto maior a acessibilidade remota (lógica e fisicamente), maior o número de agentes de ameaça com capacidade de exploração, aumentando, portanto, a viabilidade do ataque. O Quadro 28 apresenta os níveis de classificação de viabilidade com base nesse critério.

Quadro 28 – Abordagem baseada em vetores de ataque

Class. da viab. de ataque	**Critério**
High	Network: O caminho de ataque potencial está ligado à pilha de rede sem nenhuma limitação EXEMPLO 1: Conexão de rede celular, tornando a ECU diretamente conectada e acessível na internet
Medium	Adjacent: O caminho de ataque potencial está ligado à pilha de rede; no entanto, a conexão é limitada fisicamente ou logicamente EXEMPLO 2: Interface Bluetooth, conexão de rede privada virtual
Low	Local: O caminho de ataque potencial não está ligado à pilha de rede e os agentes de ameaças exigem acesso direto ao item para realizar o caminho de ataque EXEMPLO 3: Dispositivo de armazenamento em massa de barramento serial universal, cartão de memória
Very low	Physical: Agentes de ameaças exigem acesso físico para realizar o caminho do ataque

Fonte: Adaptado de ISO [8]

Decisão do Tratamento de Risco

A Seção 15.9 trata do processo de decisão para o tratamento de riscos, o qual exige um conjunto mínimo de informações prévias para sua realização. De acordo com a norma, as entradas obrigatórias para essa etapa incluem a definição do item em análise, os cenários de ameaça identificados e os respectivos valores de risco associados a cada cenário. Além dessas informações, a norma recomenda considerar dados adicionais para apoiar a tomada de decisão, tais como:

- Especificações de cibersegurança;
- Decisões de tratamento de risco anteriores (do mesmo item ou de itens similares);
- Classificações de impacto e suas categorias associadas;
- Caminhos de ataque;
- Classificações da viabilidade de ataque.

A norma apresenta quatro opções principais para o tratamento de cada cenário de ameaça, a serem selecionadas com base nos valores de risco:

a) **Evitar o risco:** Implica em eliminar as fontes de risco ou interromper atividades que o geram. Por exemplo, evitar o risco removendo as fontes de risco, decidindo não iniciar ou continuar com a atividade que dá origem ao risco;

b) **Reduzir o risco:** Consiste na aplicação de medidas técnicas ou organizacionais para diminuir o impacto ou a probabilidade de exploração do risco;

c) **Compartilhar o risco:** Envolve a distribuição da responsabilidade pelo risco, como por meio de contratos, terceirizações ou seguros;

d) **Reter o risco:** Refere-se à aceitação consciente do risco residual, desde que devidamente justificado e monitorado ao longo do ciclo de vida.

A norma ressalta que, nos casos em que o risco é retido ou compartilhado, as justificativas devem ser documentadas na forma de reivindicações de cibersegurança (*cybersecurity claims*) e incorporadas aos processos contínuos de monitoramento e gestão de vulnerabilidades, conforme estabelecido na Cláusula 8.

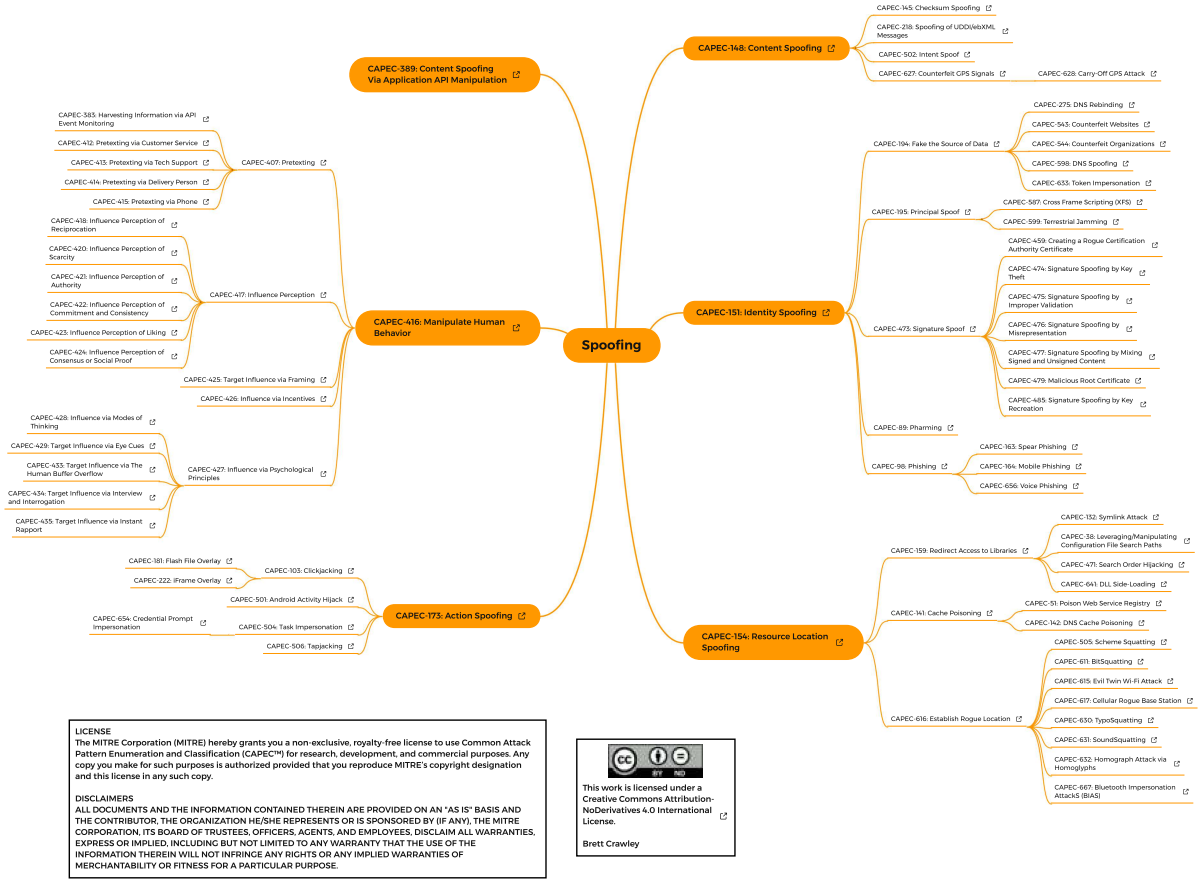
Além da Seção 15.9, que apresenta as opções formais para o tratamento de risco, a Seção 9.4.2 da ISO fornece um conjunto complementar de requisitos que apoia e detalha esse processo de maneira integrada ao longo do ciclo de vida do item automotivo. Essa seção reforça que, ao se optar pela redução do risco como decisão de tratamento para um determinado cenário de ameaça, é obrigatória a definição de um ou mais objetivos de cibersegurança (*cybersecurity goals*) ou seja, metas de cibersegurança voltadas à proteção dos ativos frente às ameaças identificadas. Além disso, a norma ressalta que, em certos casos, evitar o risco pode envolver a remoção da fonte de risco, o que pode demandar modificações no próprio item, sendo tais alterações regidas pelas diretrizes de gerenciamento de mudanças estabelecidas na Seção 5.4.4 da norma.

ANEXO B – Mapeamento entre STRIDE e CAPEC

Este anexo apresenta o mapeamento entre as categorias de ameaça do modelo STRIDE e os padrões de ataque do catálogo CAPEC. Tal associação foi desenvolvida por Brett Crawley através do projeto *CAPEC-STRIDE Mapping* [66], ao qual todos os créditos são atribuídos a ele. Para uma melhor visualização das associações, foram utilizados mapas mentais representando os padrões de ataque vinculados a cada categoria STRIDE. Devido à complexidade e ao tamanho das estruturas, recomenda-se o acesso ao site com os diagramas completos por meio do seguinte link: https://ostering.com/blog/2022/03/07/capec-stride-mapping/

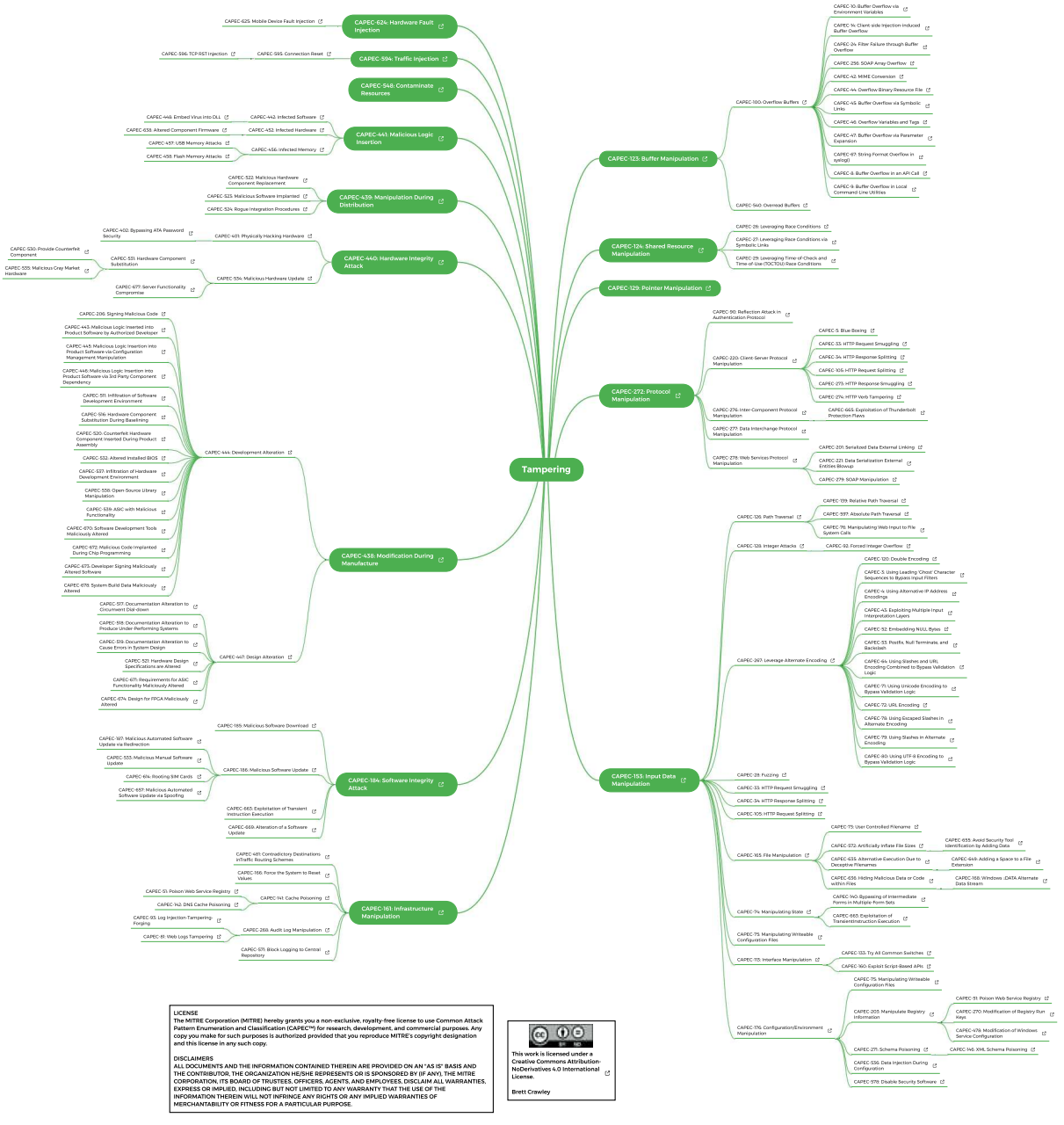
O *framework* do projeto foi adotado neste trabalho como base para correlacionar, de forma sistemática, cada categoria de ameaça STRIDE com os respectivos ataques descritos na base CAPEC. A incorporação desse mapeamento ao banco de dados contribuiu significativamente para a etapa de identificação de ameaças no processo TARA, promovendo a integração com os dados estruturados da MITRE, viabilizando consultas mais precisas e alinhadas aos requisitos da norma ISO/SAE 21434.

Figura 19 – STRIDE e CAPEC: Falsificação



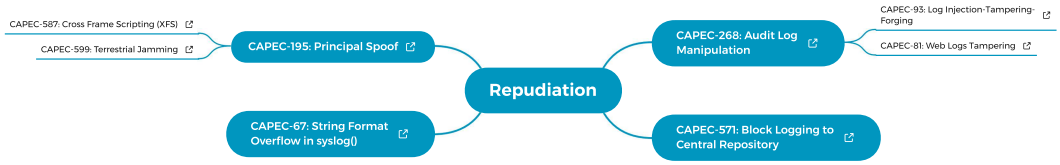
Fonte: CAPEC-STRIDE Mapping - Brett Crawley (OSTERING) [66]

Figura 20 – STRIDE e CAPEC: Adulteração



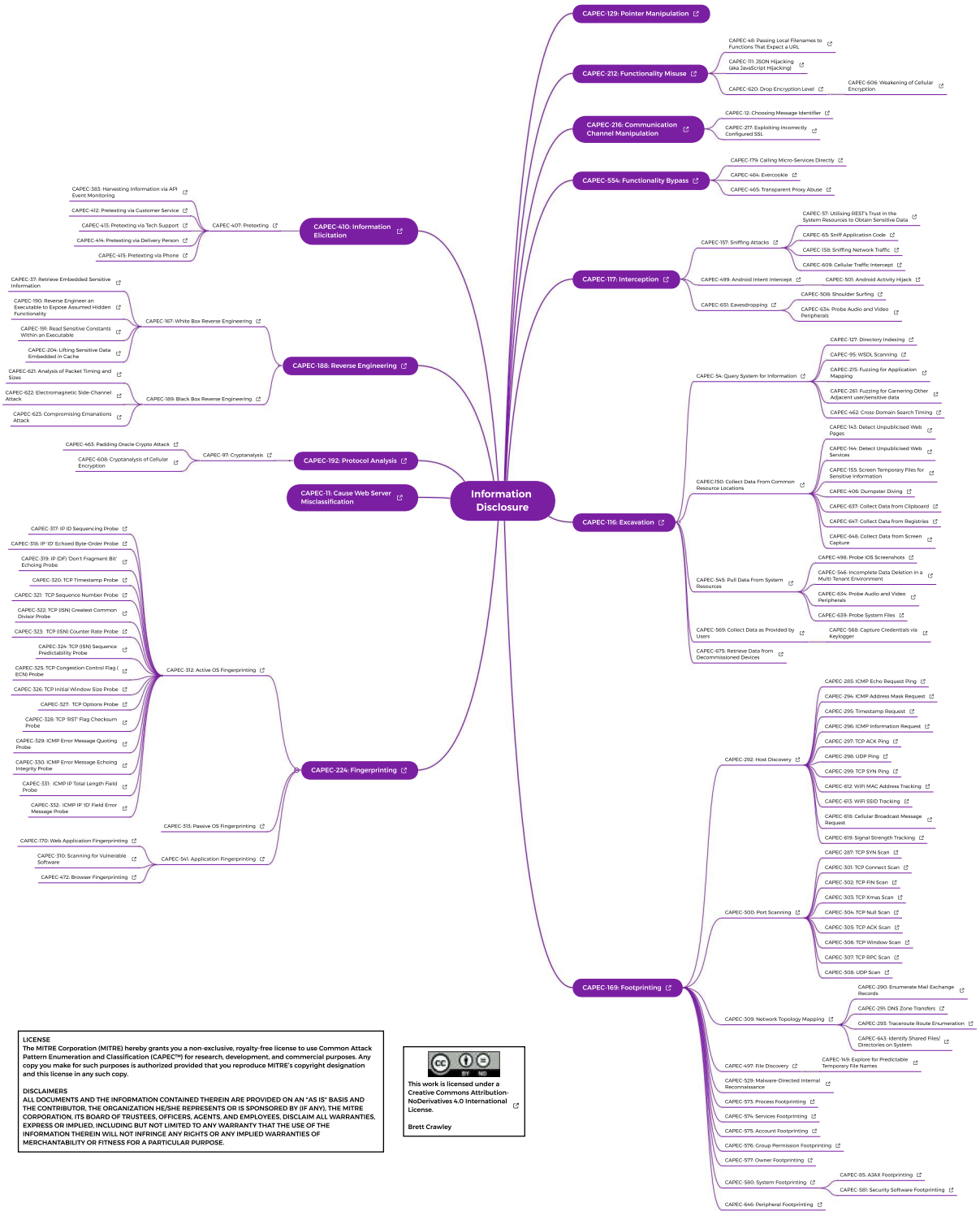
Fonte: CAPEC-STRIDE Mapping - Brett Crawley (OSTERING) [66]

Figura 21 – STRIDE e CAPEC: Repúdio



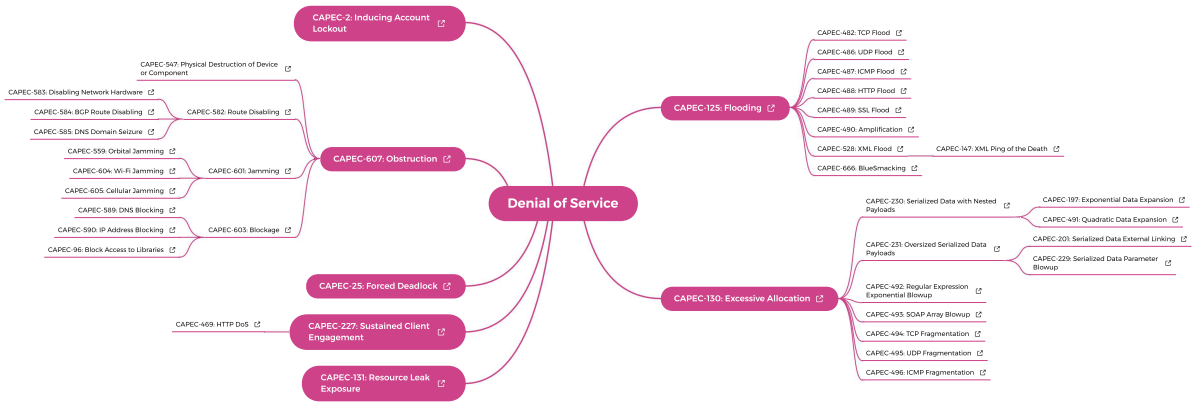
Fonte: CAPEC-STRIDE Mapping - Brett Crawley (OSTERING) [66]

Figura 22 – STRIDE e CAPEC: Divulgação de Informação



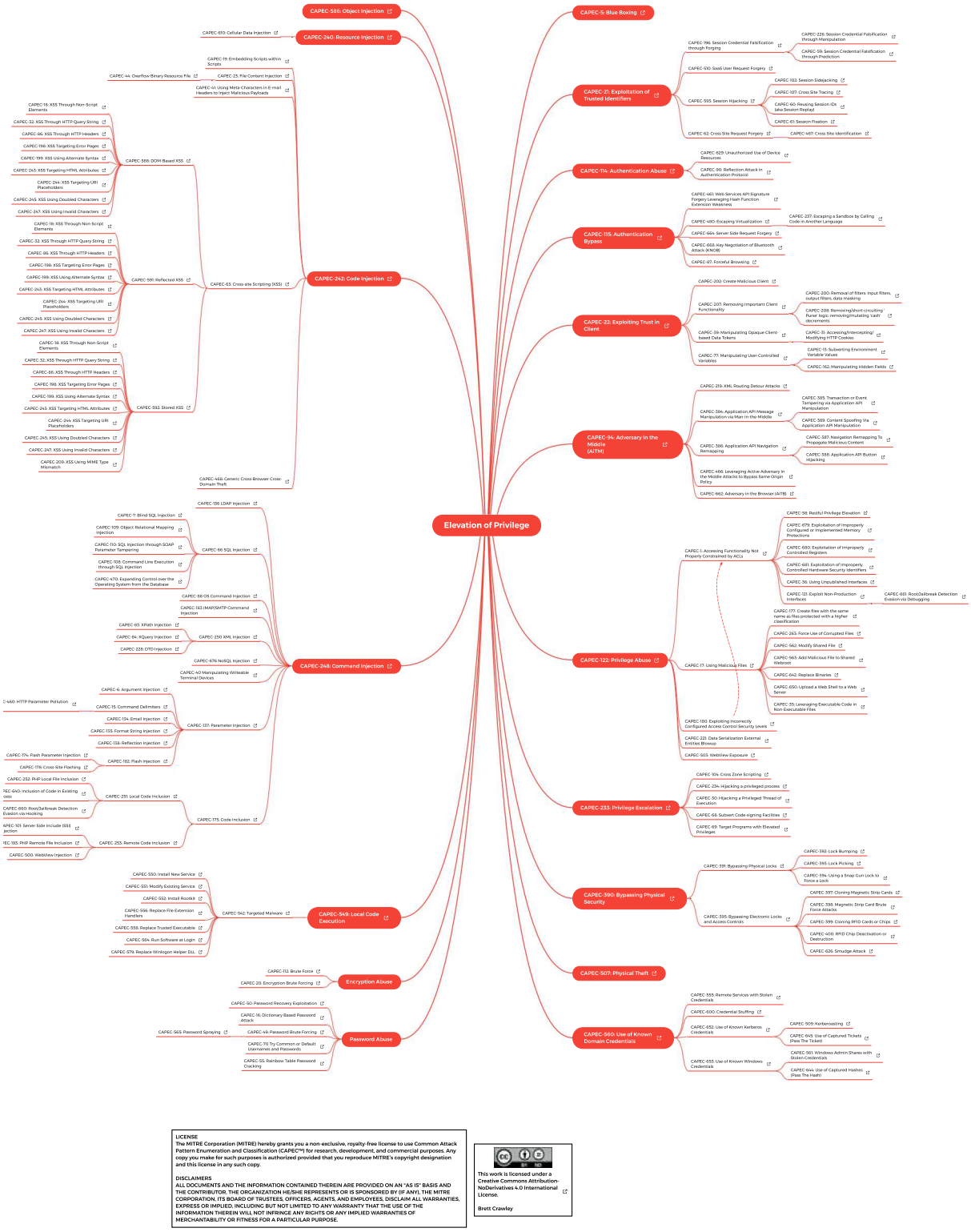
Fonte: CAPEC-STRIDE Mapping - Brett Crawley (OSTERING) [66]

Figura 23 – STRIDE e CAPEC: Negação de Serviço



Fonte: CAPEC-STRIDE Mapping - Brett Crawley (OSTERING) [66]

Figura 24 – STRIDE e CAPEC: Elevação de Privilégio



Fonte: CAPEC-STRIDE Mapping - Brett Crawley (OSTERING) [66]