Codificação Segura em Sistemas Embarcados: Uma Exploração de Vulnerabilidades

Kemily Teixeira
Acadêmica em Análise e Desenvolvimento de Sistemas;
Faculdade Senac Joinville
kemily.rosa@alunos.sc.senac.br

Vitor Gustavo de Oliveira
Acadêmico em Análise e Desenvolvimento de Sistemas;
Faculdade Senac Joinville
vitor.oliveira@alunos.sc.senac.br

Cesar Augusto Becerra Oquendo
Acadêmico em Análise e Desenvolvimento de Sistemas;
Faculdade Senac Joinville
cesar.oquendo@alunos.sc.senac.br

Erich Wanderley Formiga
Acadêmico em Análise e Desenvolvimento de Sistemas;
Faculdade Senac Joinville
erich.formiga@alunos.sc.senac.br

Gabriel Caixeta Silva
Mestre em Computação Aplicada
Docente na Faculdade Senac em Joinville
gabriel.silva@prof.sc.senac.br

Resumo

E X E M P L O / ORIENTAÇÃO Aqui precisamos apresentar a ideia do trabalho, com informações como objetivos, métodos, justificativa, resultados e conclusões. Tudo dentro de apenas um parágrafo. Nesse tópico não serão incluídos tabelas e gráficos. Esses elementos podem ser utilizados, posteriormente, na sequência do resumo expandido. O resumo que é escrito dentro do resumo expandido deve ter, no máximo, 250 palavras.

Primeira, Segunda, Terceira, Quarta

1 Introdução

Compreender as complexidades e desafios da segurança de software em sistemas embarcados é um objetivo crucial diante da crescente interdependência desses sistemas em nosso cotidiano (SELL, 2019). A questão central que norteia esta pesquisa reside na identificação dos fatores que contribuem para a existência das principais categorias de vulnerabilidades de codificação nesse contexto específico. Ademais, busca-se compreender como práticas de codificação segura podem ser implementadas de forma eficaz para mitigar essas vulnerabilidades e fortalecer a segurança dos sistemas embarcados.

Este estudo se propõe a analisar e responder a diversas questões interligadas, abrangendo desde a identificação das principais categorias de vulnerabilidades e técnicas de exploração presentes nos sistemas embarcados até a investigação das causas subjacentes a essas

vulnerabilidades de codificação. Além disso, busca-se compreender as técnicas e ferramentas frequentemente utilizadas para reforçar a segurança de software nesse contexto específico.

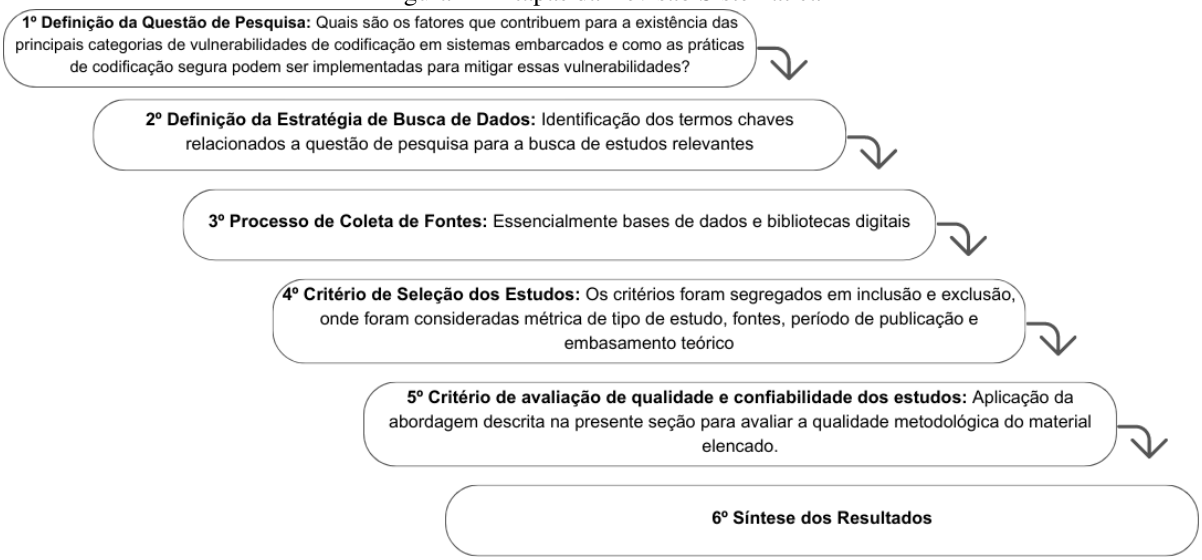
Adicionalmente, será explorado o impacto das regulamentações e padrões de segurança na condução de uma codificação mais segura em sistemas embarcados, avaliando como esses parâmetros influenciam as práticas adotadas no desenvolvimento desses sistemas.

Ao analisar as lacunas atuais na pesquisa sobre codificação segura em sistemas embarcados, esta revisão sistemática visa oferecer uma compreensão mais abrangente do campo, identificando áreas de maior necessidade de investigação e potenciais direções para futuros estudos. A resposta a estas questões secundárias se apoia em uma revisão cuidadosa, que pode proporcionar um panorama abrangente e embasado sobre a segurança de software na codificação de sistemas embarcados.

2 Estratégia de Pesquisa

Este estudo é classificado como uma revisão sistemática, cujo, segundo os autores (MOHER et al., 2017), o objetivo principal é sintetizar a melhor evidência disponível para responder a uma pergunta específica de pesquisa. Ademais, os autores enfatizam a importância de etapas bem definidas para que estes dados sejam adequadamente expostos com a finalidade de responder claramente a questão levantada no objetivo da pesquisa. As etapas definidas para a elaboração deste estudo podem ser observadas na Figura 1.

Figura 1 – Etapas da Revisão Sistemática



Fonte: Adaptação de Moher e Shamseer (2017).

Na abertura desta revisão sistemática, descrita na primeira etapa, acontece a definição da questão de pesquisa, onde o objetivo principal é exposto para abrir e aprofundar o diálogo entre os estudos relacionados ao tema. A questão de pesquisa foi apresentada na seção de Intro-

dução, onde foi explorado o objetivo principal e houve a segregação em questões securandárias que serão ponderadas nas seções de Resultados e Discussões.

A segunda etapa define a estratégia de busca de dados para elencar os estudos examinados nesta revisão. Quando superada a questão de pesquisa, é possível estabelecer os termos chaves e frases relacionadas que conduzem a fase de busca de dados. A presente revisão foi embasada na seguinte expressão, e sua versão no idioma inglês, para realizar a busca nas bases de dados:

- (segurança) AND ("sistemas embarcados"OR "sistema embarcado") AND (vulnerabilidades OR vulnerabilidade)

Estabelecida a estratégia de pesquisa, é necessário que o processo de busca de dados esteja alinhado com as métricas estabelecidas nesta estratégia, esta é a terceira etapa descrita na Figura 1. Para tal fim, por melhor adequação com esta estratégia, as fontes para a presente pesquisa foram selecionadas a partir da base de dados eletrônica indexada e biblioteca digital, IEEE[1]. Ademais, foram realizadas buscas em motores de busca eletrônica, como Scopus[2] e Google Acadêmico[3].

A diversificada gama de conteúdo oferecido pelas fontes de pesquisa descritas na terceira etapa, estabelecem a necessidade de uma filtragem ou seleção destes materiais. Isto posto, a quarta etapa pormenoriza os critérios para esta seleção, que foram segregados em critérios de inclusão e exclusão, conforme descritos na Tabela 1.

Figura 2 – Critérios de Inclusão e Exclusão

Critérios de Inclusão	Critérios de Exclusão
• Estudos empíricos e estudos teóricos • Período de publicação entre 2018 e 2023 • Idioma Português e Inglês • Embasamento científico	• Estudos de caso e estudo transversal • Período de publicação anterior a 2018 • Idiomas não especificados no critério de inclusão • Estudos sem citações e/ou referências

Elaborada pelos autores (2023).

Os estudos elencados para esta revisão, seguem essencialmente critérios de estudos empíricos, com aplicação prática, e estudos teóricos de levantamento de hipóteses consistentes e relevantes para o tema. Contrariamente, não serão considerados estudos de caso e estudos transversais, para que a revisão seja dotada de imparcialidade e coerência no contexto geral na análise dos fatos.

[1] Institute of Electrical and Electronics Engineers: associação global que promove as ciências da engenharia elétrica, eletrônica, computação e afins através de publicações e conferências

[2] Banco de dados bibliográfico e de resumos de artigos científicos que abrange diversas áreas

[3] Mecanismo de pesquisa de artigos acadêmicos e afins

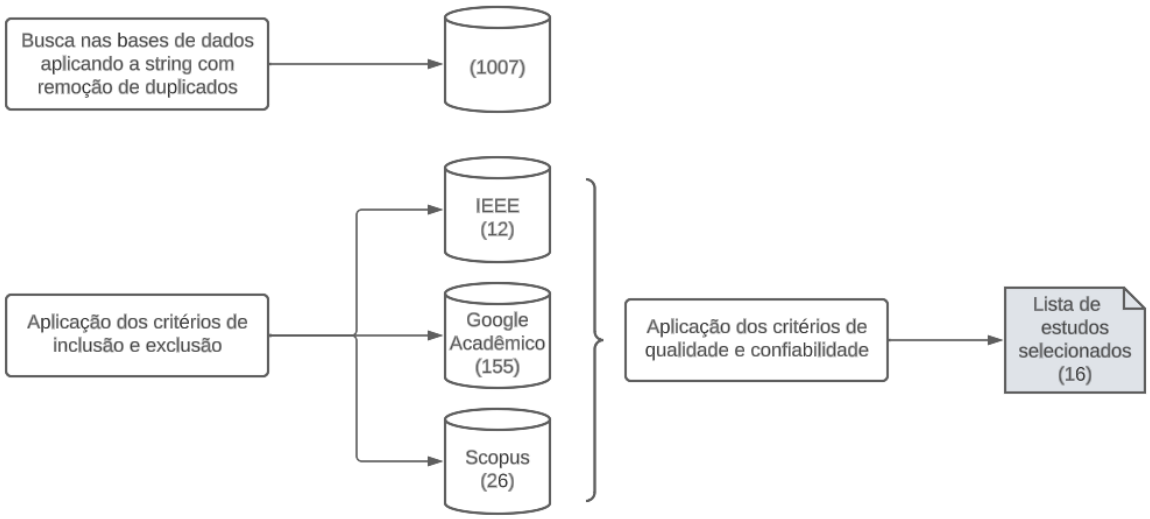
Um período de cinco anos foi definido como janela para seleção dos estudos, de maneira a sempre priorizar estudos recentes e com validade temporal em relação ao tema abordado. Ainda, os estudos nos idiomas português e inglês foram definidos para melhor entendimento por parte dos autores desta revisão. Por fim, estudos com falta de relevância científica comprovada e sem citações serão excluídos da seleção inicial da pesquisa.

A quinta etapa desta revisão sistemática conta com o critério de avaliação metodológica de qualidade e confiabilidade dos estudos selecionados. Para este fim, esta pesquisa utilizou da abordagem proposta por (DYBA; DINGSOYR; HANSSEN, 2007), que avalia três diretrizes de qualidade do estudo: credibilidade, rigor e relevância. Com base nesta abordagem, foram aplicadas as pontuações para definir se a publicação satisfaz plenamente ou se não atende aos critérios de qualidade descritos a seguir:

1. Credibilidade: as descobertas são bem apresentadas e significativas?
2. Rigor: uma abordagem completa e apropriada foi aplicada aos principais métodos de pesquisa no estudo?
3. Relevância: quão úteis são as descobertas para o indústria de software e a comunidade de pesquisa?
4. Relevância: a pesquisa está relacionada a algum domínio dos métodos aplicados à qualidade de software a partir dos requisitos?

Para a presente revisão sistemática, foram considerados como fonte de qualidade e alta confiabilidade os estudos que atendem os três critérios descritos acima, não somente, também foram excluídos os estudos que não atenderam o critério descrito no item *4*. Assim, as etapas que compõem a seleção dos estudos foram definidas como detalha a Figura *3*, que será detalhada nos itens abaixo.

Figura 3 – Seleção dos Estudos



Elaborada pelos autores (2023).

3 Resultados e Discussões

Na seção anterior, foi explicitado o processo de seleção dos 16 estudos incluídos nesta revisão sistemática. Uma grande parcela desses estudos (62.5%) concentra-se na avaliação e análise das vulnerabilidades em sistemas embarcados. Além disso, abordam-se tópicos específicos como a segurança em sistemas embarcados automotivos (6.75%) e a importância da atualização de Firmware em sistemas embarcados (12.5%). A seleção também contemplou estudos que exploram o desenvolvimento, monitoramento e as contramedidas relacionadas às vulnerabilidades em sistemas embarcados (12.5%).

Outrossim, com base nos estudos selecionados, foi possível identificar os fatores que crontribuem para a existência das principais vulnerabilidades e quais as práticas conhecidas na literatura para mitigá-las. Estes pontos serão explorados neste capítulo com o objetivo de responder a questão principal de pesquisa.

- Quais são os fatores que contribuem para a existência das principais categorias de vulnerabilidades de codificação em sistemas embarcados e como as práticas de codificação segura podem ser implementadas para mitigar essas vulnerabilidades?

Isto posto, cinco subquestões serão expostas a seguir com a finalidade de fracionar a questão principal e encontrar as principais categorias de vulnerabilidade de codificação, as técnicas utilizadas para explorar estas vulnerabilidades e suas causas, e também práticas que devem ser adotadas para garantir a segurança na codificação. Por fim, espera-se entender o impacto de regulamentações e padrões de segurança na codificação e possíveis lacunas no entendimento da prática de segurança para sistemas embarcados.

3.1 Quais são as principais categorias de vulnerabilidades e técnicas de exploração em sistemas embarcados?

Os sistemas embarcados enfrentam uma diversidade de ameaças que comprometem sua segurança. Além dos ataques que visam esgotar os recursos de energia, existe a preocupação significativa com invasões físicas, o que pode permitir que os invasores acessem diretamente o sistema (ALOSEEL et al., 2020).

Esses desafios não se limitam apenas à exaustão de recursos, mas também envolvem ameaças à integridade do sistema, potencialmente afetando o barramento principal e podem, inclusive, danificar sensores ou periféricos. Dentro desse panorama desafiador, é crucial priorizar a segurança global desses sistemas, o que abrange pilares como confidencialidade, integridade e disponibilidade.

Este tópico visa reunir as principais categorias de ameaças identificadas na literatura, proporcionando uma compreensão abrangente do panorama de vulnerabilidades nos sistemas embarcados. Essa abordagem pretende estabelecer uma base sólida nestas categorias que demandam atenção especial e fornecer um contexto amplo para o entendimento das ameaças e, posteriormente, para a análise das estratégias de exploração empregadas pelos atacantes.

3.1.1 Buffer Overflow

O *Buffer Overflow* é um tipo específico de ataque de *malware*[4] mencionado no contexto dos sistemas embarcados, mais especificamente quando trata-se do comprometimento do pilar da disponibilidade, direcionados a capacidades de memória limitadas e recursos computacionais (ALOSEEL et al., 2020). Estes ataques podem afetar, não somente os dados armazenados, mas inclusive as chaves criptográficas essenciais para a segurança dos sistemas embarcados.

3.1.2 Code Injection

Esta categoria é comumente utilizada por invasores para inserir e executar códigos maliciosos em um sistema dos mais variados tipos (AHMET; SARIKAYA; ALTINBAŞ, 2019), e também aplicam-se a sistemas embarcados, que pode ser especialmente preocupante, dado que em grande parte das vezes lidam com restrições de recursos e ambientes operacionais complexos.

No âmbito dos sistemas embarcados, a Injeção de Código pode manifestar-se com a manipulação de pacotes de redes, permissão de escrita, leitura e captura de arquivos ou inserção de arquivos sem proprietário, permissão de arquivos de senhas ou inclusive ações maliciosas em dispositivos conectados (SANTOS; SABLÓN; IANO, 2018).

[4] Software intencionalmente feito para causar danos a um computador, servidor, cliente, ou a uma rede de computadores.

O *Code Injection* pode, inclusive, ser utilizado para explorar a vulnerabilidade de *Buffer Overflow*, citado anteriormente neste capítulo. Estes ataques, quando combinados, são capazes, por exemplo, de inserir um código que sobrescreva áreas críticas da memória, o que leva a execução de comandos indesejados e pode comprometer a integridade do sistema.

3.1.3 Erros de Criptografia

Os erros de criptografia em sistemas embarcados representam um ponto crítico de vulnerabilidade pois estes sistemas frequentemente lidam com informações sensíveis e operam em ambientes onde os recursos são limitados, o que pode impactar a implementação efetiva de protocolos de segurança.

Esta vulnerabilidade é proveniente, principalmente, de implementações inadequadas de criptografia (FRANÇA, 2018), protocolos de segurança frágeis falta de atualizações e correções de segurança, limitações de recursos, falhas na gestão de chaves, entre outros.

Devido ao grande volume de exposição que um sistema pode ser submetido, ser resiliente a ataques torna-se cada vez mais necessário para a indústria de sistemas eletrônicos. As vulnerabilidades citadas nos tópicos anteriores, assim como inúmeras outras, podem ter a finalidade de encontrar uma chave de criptografia e obter acesso a dados sensíveis, portanto faz-se necessário a criação de uma zona segura (SANT'ANA, 2019)(CHAI et al., 2019).

3.1.4 Desreferenciação de Ponteiros Nulos

Como o nome define, a desreferenciação de ponteiros nulos ocorre quando um sistema tenta acessar ou manipular dados por um ponteiro que está definido como nulo, que não aponta para um local váldo na memória, esta problema pode promover falhas mais graves e causar desde erros de execução até vulnerabilidades de segurança (YUN et al., 2022).

Em sistemas embarcados, onde os recursos são limitados e a gestão de memória pode ser mais delicada, esta vulnerabilidade pode ser especialmente problemática, visto que o sistema fica suscetível a um comportamento inesperado(ALOSEEL et al., 2020). Este comportamento inesperado por sua vez pode ocasionar falhas no funcionamento do dispositivo, interrupções no serviço e inclusive a possibilidade de ataques de exploração por parte de invasores que podem manipular esse comportamento inesperado para ganhar controle sobre o sistema.

3.2 Quais são as causas subjacentes das vulnerabilidades de codificação em sistemas embarcados?

Referenciar autores selecionados afim de responder a questão com base nos conhecimentos existentes.

3.3 Quais são as técnicas e ferramentas frequentemente usadas para fortalecer a segurança de software em sistemas embarcados?

Nos ambientes cada vez mais conectados e dependentes de sistemas embarcados, a segurança de software tornou-se uma prioridade crucial. Diante da complexidade das ameaças cibernéticas, é fundamental adotar estratégias e ferramentas robustas para fortalecer a segurança nesses sistemas. Nesse contexto, técnicas e ferramentas dedicadas à identificação e correção de vulnerabilidades tornam-se essenciais (SCHNEIDER et al., 2019).

3.3.1 Técnicas de Análise de Segurança em Sistemas Embarcados

Discorrer melhor sobre as técnicas —— > Análise Estática de Código, Análise Dinâmica de Vulnerabilidades, Atualizações e Patches, Auditoria e Monitoramento Contínuo, entre outras.

3.3.2 Ferramentas de Segurança de Software em Sistemas Embarcados

SAFECode é um compilador para segurança de memória construído com base na estrutura do LLVM e nos drivers de compilação do Clang (BISHOP, 2002). A função do SAFECode é identificar trechos de código vulneráveis a erros de memória durante a compilação e inserir código adicional para verificá-los em tempo de execução. Esses erros incluem estouro de vetores, uso de variáveis após a liberação de memória e referência a ponteiros incoerentes. Com instrumentação adicional para rastrear informações de depuração, um compilador de segurança de memória pode ser usado para localizar e diagnosticar erros de segurança de memória em programas, sendo essa funcionalidade semelhante ao que o Valgrind faz.

AddressSanitizer assim como o SAFECode, a ferramenta AddressSanitizer atua na segurança de acesso à memória, inserindo código durante a compilação para verificação em tempo de execução (YUN et al., 2022). A ferramenta é capaz de identificar estouros tanto no heap quanto na pilha, além de estouros globais. Além disso, a ferramenta garante a segurança contra posições de memória apontadas por ponteiros cujo conteúdo referenciado já foi liberado (*dangling pointer*).

3.4 Quais práticas de codificação segura são recomendadas para mitigar as vulnerabilidades em sistemas embarcados?

Referenciar autores selecionados afim de responder a questão com base nos conhecimentos existentes.

3.5 Como as regulamentações e padrões de segurança influenciam a codificação segura em sistemas embarcados?

Referenciar autores selecionados afim de responder a questão com base nos conhecimentos existentes.

3.6 Quais são as lacunas atuais na pesquisa sobre codificação segura em sistemas embarcados?

Referenciar autores selecionados afim de responder a questão com base nos conhecimentos existentes.

4 Conclusões

Ao longo deste estudo, foram analisados os fatores que contribuem para as vulnerabilidades de codificação em sistemas embarcados e exploradas, na teoria, as técnicas de codificação segura como meio de mitigar essas vulnerabilidades. Foi possível identificar as principais categorias de vulnerabilidades, compreender suas causas subjacentes e examinar as técnicas e ferramentas frequentemente utilizadas para reforçar a segurança desses sistemas.

Ao responder à questão principal que norteou esta pesquisa, enfatizou-se a importância de implementar práticas de codificação segura, tais como validação cuidadosa de entradas, utilização de técnicas de proteção de memória e atualizações regulares de segurança. Medidas que são cruciais para mitigar as vulnerabilidades presentes nos sistemas embarcados, afim de assegurar de maneira efetiva os pilares integridade, confidencialidade e disponibilidade das informações.

Como estudo futuro, recomenda-se a realização de uma avaliação prática dessas técinas de segurança em ambientes de sistemas embarcados reais. Um experimento que simule tentativas de ataques baseadas nas técnicas de exploração citadas ao longo deste estudo seria uma abordagem valiosa. Este estudo permitiria não apenas testar a eficácia das práticas de codificação segura, mas também forneceria dados significativos sobre a resiliência desses sistemas diante de potenciais ameaças.

Outrossim, uma sugestão de aplicação direta dessas práticas seria a criação de diretrizes de segurança específicas para desenvolvedores e fabricantes de sistemas embarcados. Essas diretrizes poderiam abranger desde a implementação de técnicas de criptografia robusta até a criação de processos sólidos de atualização de software.

Em suma, a pesquisa destacou a importância de abordar proativamente as vulnerabilidades de codificação em sistemas embarcados por meio de práticas de segurança robustas e da compreensão das técnicas de exploração utilizadas por potenciais invasores. Estas conclusões fornecem uma base sólida para a melhoria contínua da segurança de software nesse contexto cada vez mais vital em nossas vidas cotidianas.

Referências

AHMET, E.; SARIKAYA, M.; ALTINBAŞ, M. The security vulnerabilities on internet-enabled embedded systems. **Erzincan University Journal of Science and Technology**, Erzincan Binali Yildirim University, v. 12, n. 3, p. 1468–1484, 2019.

ALOSEEL, A. et al. Analytical review of cybersecurity for embedded systems. **IEEE Access**, IEEE, v. 9, p. 961–982, 2020.

BISHOP, M. **Computer Security: Art and Science**. [S.l.]: Addison-Wesley, 2002.

CHAI, H. et al. A short review of security-aware techniques in real-time embedded systems. **Journal of Circuits, Systems and Computers**, World Scientific, v. 28, n. 02, p. 1930002, 2019.

DYBA, T.; DINGSOYR, T.; HANSSEN, G. K. Applying systematic reviews to diverse study types: An experience report. In: IEEE. **First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)**. [S.l.], 2007. p. 225–234.

FRANÇA, F. F. d. Considerações sobre a segurança da informação em sistemas embarcados automotivos. Universidade Tecnológica Federal do Paraná, 2018.

MOHER, D. et al. Preferred reporting items for systematic review and meta-analysis protocols (prisma-p) 2015 statement. **Systematic reviews**, BioMed Central, v. 4, n. 1, p. 1–9, 2017.

SANTOS, T. C. dos; SABLÓN, V. I. B.; IANO, Y. Segurança da informação para aplicações interativas no sistema brasileiro de televisão digital. 2018.

SANT'ANA, A. C. **Vulnerabilidades de segurança e contramedidas em plataformas MP-SoCS**. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul, 2019.

SCHNEIDER, D. et al. **Safety and security coengineering in embedded systems**. [S.l.]: Hindawi, 2019.

SELL, P. F. Atualização e carga de firmware em sistemas embarcados de forma segura e confiável. 2019.

YUN, J. et al. Fuzzing of embedded systems: A survey. **ACM Computing Surveys**, ACM New York, NY, v. 55, n. 7, p. 1–33, 2022.