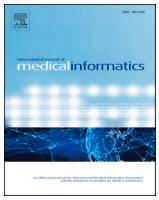
International Journal of Medical Informatics 194 (2025) 105699

Contents lists available at ScienceDirect

International Journal of Medical Informatics

journal homepage: www.elsevier.com/locate/ijmedinf



A method and validation for auditing e-Health applications based on reusable software security requirements specifications

Carlos M. Mejía-Granda, José L. Fernández-Alemán, Juan M. Carrillo de Gea, José A. García-Berná *

Department of Informatics and Systems, Faculty of Computer Science, University of Murcia, Murcia 30100, Spain

ARTICLE INFO

Keywords:
e-Health
Requirements engineering
Medical software
Audit methodology
Software security
Software requirements specification

ABSTRACT

Objective: This article deals with the complex process of obtaining security requirements for e-Health applications. It introduces a tailored audit and validation methodology particularly designed for e-Health applications. Additionally, it presents a comprehensive security catalog derived from primary sources such as law, guides, standards, best practices, and a systematic literature review. This catalog is characterized by its continuous improvement, clarity, completeness, consistency, verifiability, modifiability, and traceability.

Methods: The authors reviewed electronic health security literature and gathered primary sources of law, guides, standards, and best practices. They organized the catalog according to the ISO/IEC/IEEE 29148:2018 standard and proposed a methodology to ensure its reusability. Moreover, the authors proposed SEC-AM as an audit method. The applicability of the catalog was validated through the audit method, which was conducted on a prominent medical application, OpenEMR.

Results: The proposed method and validation for auditing e-Health Applications through the catalog provided a comprehensive framework for developing or evaluating new applications. Through the audit of OpenEMR, several security vulnerabilities were identified, such as DDOs, XSS, JSONi, and CMDi, resulting in a "Secure" classification of OpenEMR with a compliance rate of 66.97%.

Conclusion: The study demonstrates the proposed catalog's feasibility and effectiveness in enhancing health software security. The authors suggest continuous improvement by incorporating new regulations, knowledge from additional sources, and addressing emerging zero-day vulnerabilities. This approach is crucial for providing practical, safe, and quality medical care amidst increasing cyber threats in the healthcare industry.

Abbreviations: AICPA, American Institute of Certified Public Accountants; Apps, Applications; ASD STIG, Application Security and Development Security Technical Implementation Guide; AWS, Amazon Web Services; CDSS, Clinical Decision Support Systems; CHS, Consumer health solutions; CIA, Confidentiality, Integrity, and Availability; CMS, Centers for Medicare and Medicaid Services; CMDi, CoMmanD injection; DAST, Dynamic Application Security Testing; DDOs, Distributed Denial-of-service; DevSecOps, Development, Security, and Operations; DoS, Denial of Service; e-Health, electronic health; EHR, Electronic Health Record; EMR, Electronic Medical Records; EMR-S, Electronic Medical Record Storage; HIE, Health Information Exchange; HIPAA, Health Insurance Portability and Accountability Act; HIS, Hospital Information Systems; IAST, Interactive Application Security Testing; ICFR, Internal Controls over Financial Reporting; ISO, International Organization for Standardization; ISSO, Information System Security Officer; IT, Information technology; JSONi, JavaScript Object Notation injection; NFR, Non-Functional Requirement; NIST, National Institute of Standards and Technology; NVD, National Vulnerability Database; OECD, Organization for Economic Co-operation and Development; ONC, Office of the National Coordinator for Health Information Technology; OpenEMR, Open-source Electronic Medical Record; OWASP, Open Web Application Security Project; PKI, public key infrastructure; PRISMA, Preferred Reporting Items for Systematic Reviews and Meta-Analysis; RBAC, Role-based access control; RPM, Remote patient monitoring; SA, System Administration; SAML, Security Assertion Markup Language; SAST, Static Application Security Testing; SEC-AM, SECurity Audit Method; SEC-CAT, SECurity CATalog; SIREN, SImple Reuse of software requiremeNts; SLR, Systematic Literature Review; SOC, Systems and Organization Controls; SOAP, Simple Object Access Protocol; SRE, security requirements engineering; SRS, Software Requirements Specification; XSLT, Extensible Stylesheet Language Transformations; XSS, Cross-Site Scripting.

* Corresponding author.
E-mail addresses: carlosmichael.mejiag@um.es (C.M. Mejía-Granda), aleman@um.es (J.L. Fernández-Alemán), jmcdg1@um.es (J.M. Carrillo de Gea), josealberto.garcia1@um.es (J.A. García-Berná).

https://doi.org/10.1016/j.ijmedinf.2024.105699
Received 29 March 2024; Received in revised form 29 October 2024; Accepted 8 November 2024
Available online 10 November 2024

1. Introduction

The increasing adoption of software applications in the healthcare sector has significantly transformed the way healthcare services are delivered [1], facilitating online consultations, medical record management, and real-time monitoring of vital signs [2,3]. These digital platforms, known as e-Health applications, offer a wide range of functionalities that improve the delivery of medical services and provide advanced tools for disease diagnosis, health promotion, and EMR constitution [4,5].

Privacy and software security are closely interconnected. However, privacy refers to safeguarding personal data processed by the system, while security refers to policies and procedures to protect assets critical to the system [6,7]. Several works address security in a focused and separate manner [8,9,10]. A security requirement can be characterized as a non-functional requirement (NFR) [11] that translates a higher-level rule, regulation, or organizational directive into the specific system needs [12,13,14].

Sensitive information handled in e-Health applications requires high levels of privacy and information security. As an illustrative example, ransomware attacks on healthcare delivery organizations have been on the rise, with incidents more than doubling between 2016 and 2021, exposing the personal health information of millions of patients [15]. The cumulative cost of cybercrime in the world between 2020 and 2025 is expected to be $125 billion and, highlighting the severe privacy and security risks associated with e-Health apps [16], which, despite their advantages, can lead to significant social, legal, and financial repercussions [17], including data breaches that result in civil penalties of up to $25,000 per individual per violation, criminal penalties of up to $250,000, and ten years in jail under HIPAA regulations [18,19], as well as losses of reputation and brand harm. These breaches have substantial financial and care costs, putting patients' lives at risk due to decisions made with inaccurate information [20,21,22].

Security in healthcare software faces significant challenges due to errors in design and implementation. Research by McGraw [23] and Smith [24] indicates that the majority of cyberattacks are due to a lack of attention to information security in companies and vulnerabilities in the implementation of security requirements at the application level [25,26]. Other factors include the introduction of malicious code [27] and deficiencies in software analysis, design, and development [28,29,30]. Unpredictable behaviors in e-Health systems can cause critical availability problems [31,32] that put patients' lives at risk [33,34].

There are various approaches, mechanisms, and technical initiatives that are not necessarily linked to security requirements for validating and improving the security of medical software [35], including encryption [36,37,38,39], secure data transfer protocols [40,41,42,43], and defense mechanisms against physical and software vulnerabilities [44,45,46,47,48]. Additionally, Blockchain-based frameworks [49,50,51] and smart contracts [52,53,54] have been developed to ensure the integrity and availability of medical records in e-Health systems [55,56,57]. The above proposals focus on the "how". However, inquiring into the "what" at a higher level of abstraction is often decisive for increasing the security of medical software.

H. S. Gardiyawasam Pussewalage et al. [7] presented seven general privacy approaches based on HIPAA, suggestions for implementation mechanisms such as cryptography and RBAC, and the need to consider a holistic approach to e-Health security. However, this proposal could be complemented by an audit or validation method to technically test these seven proposed mechanisms.

Rezaeibagha F et al. [58] systematically reviewed the literature on EHR systems' technical security perspectives, identifying 13 essential characteristics that were taken from the standards of ISO/IEC 27,002 and ISO/IEC 27,001 [59,60]. The need to implement appropriate security and privacy policies and architectures to improve health software is concluded since, after evaluating selected ISO standards concerning the flexibility of EHR systems, gaps are revealed by focusing mainly on the perspective of technical mechanisms. Hence, the need to delve into an approved and trustworthy set of security characteristics for e-Health applications remains.

One of the biggest challenges for implementing controls in SRE is extracting requirements from any source text of security requirements [61]. These texts are often ambiguous, frequently updated, and contain numerous cross-references and complicated exceptions, making them difficult to translate into concrete, verifiable software requirements [60,62,63,64]. Despite the methodologies and tools developed to improve the extraction of textual requirements, such as the proposals of A. K. Massey et al. [22] that highlight the validation of software requirements and designs against relevant regulations, a significant gap remains between regulations and a clear proposal or definition of a single catalog of requirements and their practical implementation and evaluation in the software has not been achieved [58,65,66].

Harmonizing models through their combination and integration has become a powerful tool for organizations, offering benefits such as minimizing time and effort costs [67,68]. This document addresses the critical need to improve enterprise information security by focusing on application-level security vulnerabilities and software analysis, design, and development deficiencies. It highlights the necessity of an audit or validation method to technically test security approaches. The document proposes a harmonized criterion for security features in e-Health applications, including a single catalog of security requirements. This catalog focuses on security through a centralized repository, satisfying the CIA security triad, and is traceable, applicable, reusable, and subject to continuous improvement. It also includes a practical implementation and evaluation proposal for e-Health software against this catalog.

The article introduces a general software requirements specification (SRS) catalog for e-Health applications. The catalog sets technical and objective criteria for compliance and harmonization with laws, guidelines, standards, academic articles, and best practices, organized according to the ISO/IEC/IEEE 29148:2018 standard [69]. It uses the SIREN methodology to ensure reuse and auditability [70]. The target audience for this SRS includes those involved in designing, developing, and auditing secure e-Health applications and requirements engineers. Additionally, it will assist senior management in defining the terms of reference and the minimum characteristics required for e-Health products.

In summary, the main contributions of this article are:

1) A SLR on e-Health software and device security requirements.
2) A reusable, traceable, applicable and updateable catalog of security control requirements for e-Health software according to the ISO/IEC/IEEE 29148:2018 standard [69].
3) An audit method for evaluating security requirements for health e-Health software.
4) The validation of the proposed approach by generating a security requirements audit report for e-Health software.

2. Methods

The method used in this scientific work is composed of three stages: (i) An SLR, (ii) the construction of a catalog of security requirements, and (iii) the development of an audit method aimed at verifying the effectiveness of our catalog of requirements. The subsequent sections of the document show the described process in detail.

2.1. Research review

A SLR was performed to get security recommendations for e-Health software that follows the phases described below:

2.1.1. Selecting data sources and the search string

SCOPUS was selected to identify studies on security requirements

published between 1991 and 2023. It represents the most comprehensive source of high-quality, peer-reviewed publications for emerging research fields [71]. It is the largest and most widely used bibliometric database for citation data, offering broad interdisciplinary coverage, including biomedical sciences [72]. Several bibliographic studies have used SCOPUS as a primary source in their works [73,74]. Regarding our work, the literature search was conducted on July 9, 2023, and was limited to "Title of the article, Abstract, Keywords". Therefore, any work published after this date will not be considered for inclusion in the analysis. The PICO search string [75] is made up of the following components and is described in Table 1 as follows:

1. The population or field of study: Patient or medic or clinic or care or health.
2. Software as the intervention.
3. Security as a factor of comparison.
4. As the outcomes: requirement or catalog or criteria or guide or specification.

To reduce the possibility that the keywords used do not focus on the literature related to security requirements in e-Health, an exhaustive investigation of the most recurrent keywords was carried out, which led to the creation of a co-occurrence map that illustrates the interrelation of the keywords associated with the application of security in the field of e-Health. This map was generated using VOSviewer software (version 1.6.20) [76]. Keywords were extracted from the last 20,000 of 63,436 articles related to "security and health". The analysis resulted in a collection of 1,000 words or phrases that appeared at least fifty-five times that included synonyms, abbreviations, repeated terms (such as "insurance", "security", "hospital", "health", "electronic medical record ", "EMR") and terms not related to the central topic (such as "article", "review", "model", "survey", "literature review"). The authors standardized the nomenclature, removed abbreviations with similar meanings, and eliminated irrelevant terms. The new refined collection consisted of 65 keywords related to safety and health. The most frequent terms and their interconnections are shown in Fig. 1. These keywords guided researchers to construct wildcard nouns included into four separate areas to generate the PICO string.

Table 1
PICO-compliant SLR search string for e-Health software security recommendations.

Scope	String
Population	**((**"patient" OR "medic*" OR "clinic*" OR "care" OR " health*"**)** AND
Intervention	**(**"software"**)** AND
Control	**(**"secur*"**)** AND
Outcomes	**(**"requirement*" OR "catalog*" OR "criteria*" OR "guide*" OR "specification*"**))** AND
Additional filters	**(**LIMIT-TO (SUBJAREA,"COMP") OR LIMIT-TO (SUBJAREA,"MEDI") OR LIMIT-TO (SUBJAREA,"ENGI") OR LIMIT-TO (SUBJAREA,"HEAL")**)** AND **(**LIMIT-TO (LANGUAGE, "English")**)** AND **(**LIMIT-TO (EXACTKEYWORD, "Computer Security") OR LIMIT-TO (EXACTKEYWORD, "Security Of Data") OR LIMIT-TO (EXACTKEYWORD, "Security") OR LIMIT-TO (EXACTKEYWORD, "Network Security") OR LIMIT-TO (EXACTKEYWORD, "Confidentiality") OR LIMIT-TO (EXACTKEYWORD, "Privacy") OR LIMIT-TO (EXACTKEYWORD, "Security Systems") OR LIMIT-TO (EXACTKEYWORD, "Data Privacy") OR LIMIT-TO (EXACTKEYWORD, "Security Requirements") OR LIMIT-TO (EXACTKEYWORD, "Cryptography") OR LIMIT-TO (EXACTKEYWORD, "Access Control") OR LIMIT-TO (EXACTKEYWORD, "Authentication") OR LIMIT-TO (EXACTKEYWORD, "Safety Engineering") OR LIMIT-TO (EXACTKEYWORD, "Specifications") OR LIMIT-TO (EXACTKEYWORD, "Security And Privacy") OR LIMIT-TO (EXACTKEYWORD, "Safety") OR LIMIT-TO (EXACTKEYWORD, "Information Security") OR LIMIT-TO (EXACTKEYWORD, "Cybersecurity")**)**

As detailed in Table 1, the researchers established the following filters in the query entered into the SCOPUS database:

1. Only articles published in the English language will be considered.
2. As the subject area is limited to:
   - Computer Science
   - Medicine
   - Engineering
   - Health Professions
3. Those established in the "Additional filters" section and the subcriterion "EXACTKEYWORD" were considered keywords.

As a result of the initial search, 886 scientific papers[1] were obtained and subject to inclusion and exclusion criteria to maintain rigor and consistency in study selection.

2.1.2. Inclusion and exclusion criteria

Research work must meet specific criteria to be included:

- **IC:** The work furnishes insights into security guidelines, standards, or requirements for e-Health software and devices.

Furthermore, for a research work to be excluded, it must satisfy any of the conditions mentioned below:

- **EC1:** The work does not provide a suggestion, list, or catalog of security requirements for software development.
- **EC2:** The work cannot be accessed.
- **EC3:** The work does not provide requirements that differ from existing standards, norms, or best practices.

Six scientific articles were not considered because they repeated their title (duplicates), and after applying the EC1 and EC2 criteria, another 877 were excluded.

To apply EC1 during the full text review stage R4, a thorough analysis of each paper was carried out, carefully assessing its content in relation to security guidelines, standards and regulations. In the comprehensive review of the 69 articles, many highlighted the importance of security in software development and explored security concepts in a general way. However, they did not provide practical guidance, structured catalogue, lists or recommendations of security requirements. Rather than providing specific recommendations, these papers presented a predominantly theoretical approach, without translating into guidelines applicable to healthcare software development.

This analysis was based on the baseline SRS created by using canonical sources from ISO, HIPAA, NIST, DISA STIG, OWASP, Canada Health Infoway CWE/SANS. By comparing each paper with the baseline SRS, it was possible to determine whether it provided additional and useful information, with the aim of complementing, enriching and reinforcing previously established guidelines, standards, and regulations. The use of the baseline SRS as a reference was key to ensuring consistency and depth in the evaluation of each paper, thus strengthening the e-health security framework.

To apply EC3 in the exhaustive review of the remaining 44 articles during the full text review stage R4, requirements based on guidelines, recommendations, best practices, standards and regulations addressed in the baseline SRS were identified. As these items did not add value to this already holistic and robust catalogue in the baseline SRS, they were discarded.

Finally, three scientific works were selected after using the PRISMA directive [77,78]. Fig. 2 shows the diagrammatic representation of PRISMA's proposed flowchart for determining scientific articles on

[1] https://bit.ly/4ds2xw3.

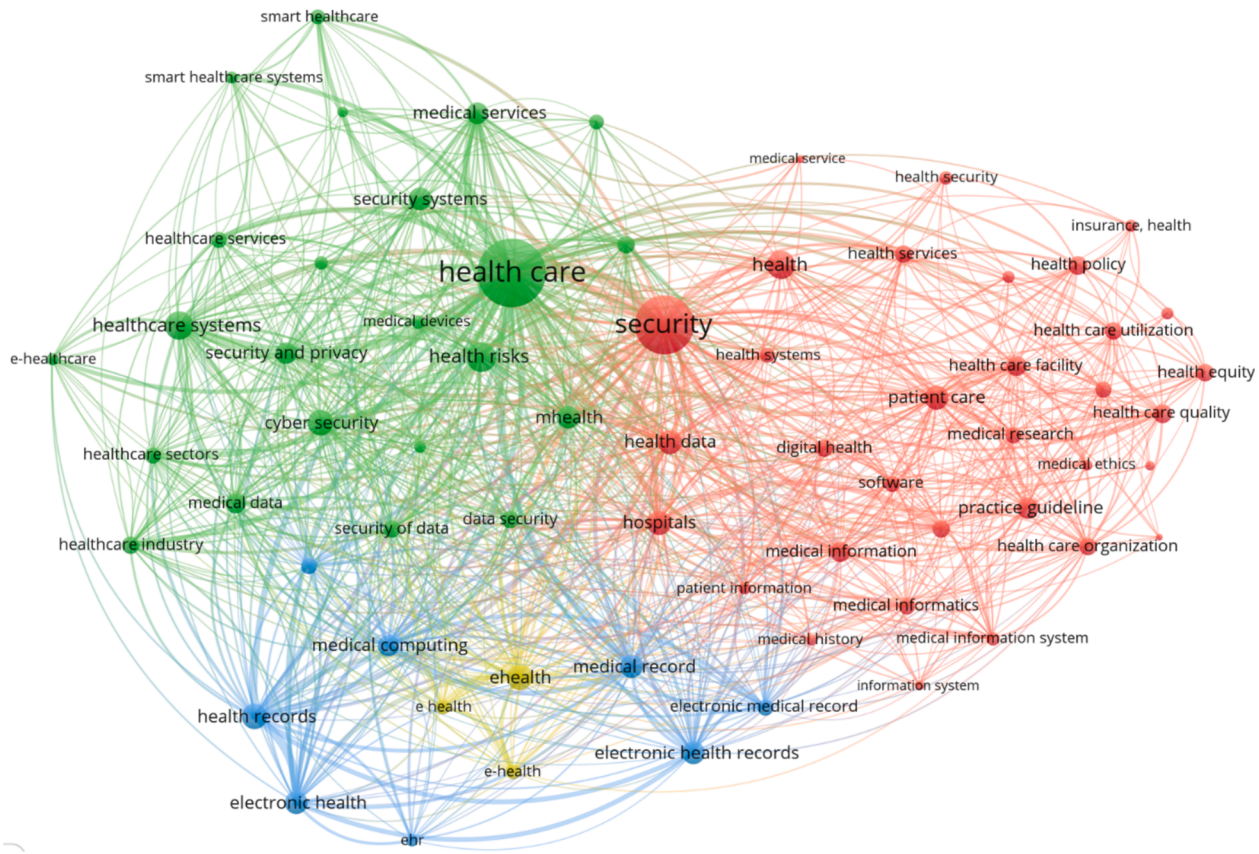


Fig. 1. Keyword co-occurrence map for security in health. Source: elaborated by the authors using VOSviewer software.

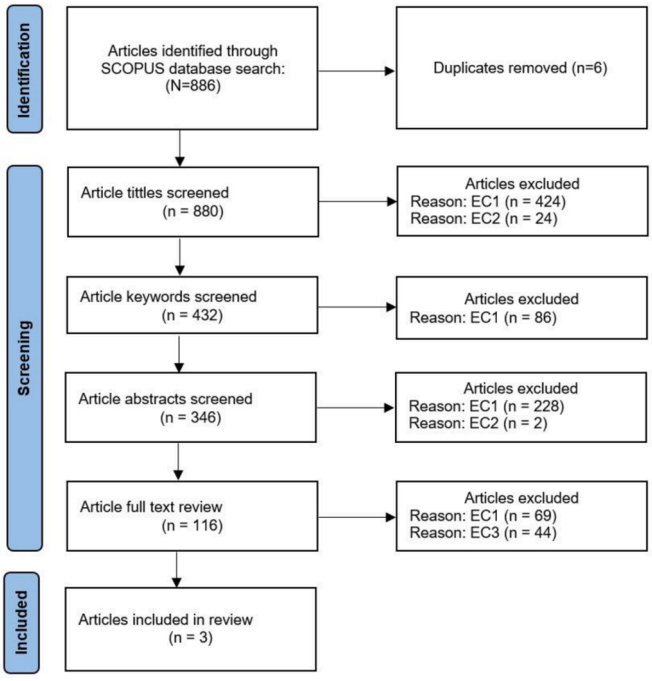


Fig. 2. PRISMA diagram of the SLR of security recommendations.

security recommendations for e-Health applications and devices.

Many papers were excluded because they did not meet these specific criteria or were duplicates. Our goal was to maintain rigor and consistency in our study selection, ensuring that only the most pertinent studies were included in our final analysis. Nevertheless, to guarantee inclusion of numerous pertinent primary studies, considering the keywords in the search string detailed in Table 1, we have added the terms "issue" and "challenge" in the search string, recovering 3,272 publications between 1991 and 2023. These were exported in.csv format from SCOPUS, of which 23 duplicates and 926 articles in the original search chain were excluded, resulting in 2,323 new articles.

A sensitivity analysis of the publications obtained from the Scopus database was performed to evaluate the accuracy of the results concerning security in e-Health applications. This evaluation was done by randomly selecting a subset of publications from the complete list and reviewing their relationship to the field of study. The number of documents selected for validation was determined using Cochran's sample size formula (1) [79]:

$$(1)\ n = \frac{NZ^2p(1-p)}{(N-1)e^2+Z^2p(1-p)}$$

Where:

- **n** is the number of documents randomly selected for validation,
- **N** is the total number of publications obtained from the Scopus database (2,323),
- **Z** is the deviation from the accepted mean value for the confidence level (1.96 for a 95 % confidence level).
- **e** is the margin of error (0.05),
- **p** is the proportion of invalid expected results (0.08, randomly selected and expected to be low).

A sample of 108 publications was required for validation, and of these, none presented security requirements with a formal structure or format, resulting in a proportion of 0 % of invalid results (8 % expected).

2.2. Security requirements catalog

The SIREN methodology guides this catalog to ensure that requirements are reusable, traceable, applicable, and auditable. It incorporates templates and a repository of reusable requirements organized according to international recommendations [70]. SIREN uses a spiral process model inspired by Kotonya and Sommerville [80] that covers the extraction, analysis, negotiation, documentation, and validation phases, adhering to standards like IEEE 830–1998 [81] and ISO/IEC/IEEE 29148:2018 [69].

Previous works have employed SIREN to create reusable horizontal catalogs of sustainability, usability, and internationalization requirements [5,82,83] and vertical catalogs about CARE tools and teleoperated systems [84,85]. The framework complies with the eight key precepts for an effective requirements process [86]:

- K1. Organization of the reusable requirements: Structure and organize the requirements to facilitate their reuse and reduce search time.
- K2. Search engine for reusable requirements: Facilitates an intuitive search to meet the needs of engineers.
- K3. Requirement selection and reuse with different granularity levels: Allows specific requirements to be reused.
- K4. Requirement attributes reuse: Defines a common set of attributes based on the IEEE standard, allowing changes and the addition of new attributes.
- K5. Traceability relationships reuse: Connect entities within the same artifacts, managing variability through parent and child requirements.
- K6. Parameterized requirements management: Allows requirements values to be parameterized according to the specific application.
- K7. Repository improvement: Supports the creation of new categories and requirements.
- K8. CARE support to reuse: Provides data necessary for entry into any chosen application, with support from tools.

Additionally, for the articles obtained from Section "2.1 Research Review", the requirements extraction was also based on various laws, guidelines, and standards[2] according to the specialized knowledge in software security that the authors have. OECD Privacy Principles, GDPR, privacy legislation in the United Kingdom and Canada, and other resources oriented toward privacy will not be considered since this work focuses on security. A summary of the requirements for each source document is presented in Table 2 below.

The catalog creation process involves four stages: step one begins by identifying the appropriate sources, literature research, and standards. Next, the requirements for security are extracted from the sources. The catalog SEC-CAT is generated based on the extracted criteria. A fourth step is followed to maintain and develop the catalog. Fig. 3 illustrates the catalog creation process.

SEC-CAT is supported in the ISO/IEC/IEEE 29148:2018 structure [69] and presents the requirements classified in the OWASP Top 10 Web Application Security Risks: 2021 scheme [96]. Table 3 provides an overview of the catalog's structure.

A requirement may have more than one source since various documents referred to and studied have been considered. To avoid overlapping concepts in the requirements contained in the SEC-CAT, the authors performed a harmonization of the source documents by applying a process adapted from the framework proposed by C. Pardo et al [98]. This process includes two main techniques: homogenization and comparison.

Homogenization Technique: The objective of homogenization is to prepare the documents for later comparative analysis [99]. The steps followed were:

[2] https://bit.ly/3SB0CNI.

Table 2
Requirements extracted from document sources.

Document source	Requirements Associated
D. L. Hamilton [87]	7
H. J. Baur, U. Engelmann, F. Saurbier, A. Schröter, U. Baur, and H. P. Meinzer [88]	7
A. Strielkina, O. Illiashenko, M. Zhydenko, and D. Uzun [89]	1
ISO 27002:2022 [60]	9
ISO 27799:2016 [90]	18
HIPAA [91]	1
HIPAA Security rules [92]	16
National Institutes of Standards and Technology (NIST) SP 800–53 rev. 5 [93]	29
Canada Health Infoway: Privacy & Security Requirements and Considerations for Digital Health Solutions [94].	23
DISA STIG Application Security and Development Security Technical Implementation Guide: 2022 [95]	108
Open Web Application Security Project (OWASP) Top 10 Web Application Security Risks:2021 [96]	34
CWE/SANS TOP 25 Most Dangerous Software Errors: 2021 [97]	19

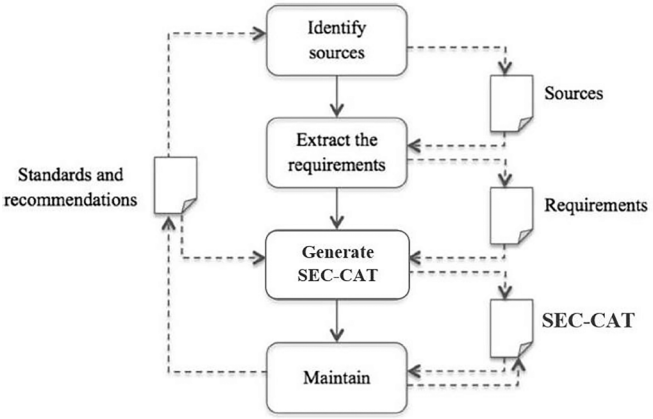


Fig. 3. Security Catalog Creation Process.

1. **They acquired knowledge:** Study of the relevant norms, standards, and procedures, and analysis of their structure and terminology to define selection criteria.
2. **Identification of Information:** Identification of the information from the documents to be combined and organized was identified.
3. **Correspondence Process:** Homogenization through an objective pairing of the information and the elements of the model process about each of the aspects of the OWASP Top 10 structure.

Comparison technique: It allows the identification of differences and similarities between the regulatory documents [100]. The process followed was:

1. **Expert Analysis:** Two experts in secure software development analyzed each requirement obtained from the documents.
2. **Identification of Differences and Similarities:** A comparison was carried out to identify how the requirements of the analyzed documents align or do not align with the OWASP Top 10 structure proposed in the SRS, discarding requirements that are essentially the same.
3. **Level of Coverage:** The experts determined the level of coverage of the sources.
4. **Review of Results**: At the end of each comparison iteration, the experts reviewed the results again to verify their reliability and that of the comparison process.

Table 3
Table of contents (IEEE 29148:2018).

Introduction
SRS overview
Purpose
Scope
Product perspective
Product functions
User characteristics
Limitations
Limitations
Assumptions and dependencies
Apportioning of requirements
Requirements
Specified requirements
External Interfaces
Functions
Usability requirements
Performance Requirements
Logical database requirements
Design constraints
Standards compliance
Software system attributes
Reliability
Availability
Security
Broken Access Control
Cryptographic Failures
Injection
Insecure Design
Security Misconfiguration
Vulnerable and Outdated Components
Identification and Authentication Failures
Software and Data Integrity Failures
Security Logging and Monitoring Failures
Server-Side Request Forgery (SSRF)
Verification
Supporting Information
References

In our security requirements catalog, according to IEEE 830–1998 [81], each requirement includes attributes such as Unique ID (PUID), description, source, rationale, verification method, validation criteria, priority, justification, etc., enabling traceability relationships to be defined and parameterizable to adapt to different applications, ensuring reuse. A catalog of security requirements elicited from the main norms, standards, guides and procedures allows for communication at the managerial and operational level. Stakeholders do not need technical knowledge to understand the system functionality. Only PUID, description, and rationale are mandatory attributes. A requirement may have more than one source since various documents referred to and studied have been taken into account.

The authors present 142 security requirements distributed in 10 subcategories according to the OWASP top ten 2021 [101], detailed in Section "2.9.3. Security" of the "Software Requirements Specification for Security on e-Health applications"[3] document. Table 4 presents an example of the requirements catalog with its base sources, extracted from the requirements document mentioned above: subsection 2.9.3.1 Broken Access Control, page 16.

2.3. Audit method

Software security audit and risk management methods have been proposed for generating certifications to increase user confidence, considering the impact of IT on financial statements [102]. One of the most used is the SOC (Systems and Organization Controls), developed by the AICPA (American Institute of Certified Public Accountants), which includes: SOC 1, which evaluates the ICFR (Internal Control over Financial Reporting) to ensure consistent processing and reliable transactions and data manipulation [102]; SOC 2, which provides information and assurance about security controls, availability, processing integrity, confidentiality and privacy in a service organization [103]; and SOC 3, similar to SOC 2 but less detailed and general purpose, allowing its free distribution [104]. Some tools have been proposed to support SOC: Vanta,[4] Drata,[5] Tugboat Logic,[6] Strike Graph,[7] AuditBoard[8] and Secureframe.[9] Compared to these tools, our catalog-based audit method offers greater flexibility to adapt the audit method framework, with a finer control to select, refine, extend, and narrow down the security requirements for which compliance will be verified. Organizations can easily adopt our lightweight audit method, adaptable to any process, and develop their own catalogs by using any evolving source of security requirements.

This section outlines an audit methodology designed to assess the security of e-Health software, **characterized** by objectivity, usability, agility, systematicity, and repeatability. The SEC-AM is an adaptation of methodologies proposed in similar audit methods [105,106,107], tailored specifically for security considerations in requirements engineering. This proposed audit method builds upon the CISA standard framework, aiming to enhance "Domain 1: Information Systems Auditing Process" by refining the planning and execution phases of the audit. The objective and scope of the security audit are comprehensively detailed in section "2.3.3 Phases and Activities of SEC-AM" of the article, particularly within the subsections "a. Context Analysis" and "b. SEC-CAT* Generation." A unique contribution of SEC-AM is the incorporation of a specialized team that analyzes the relevant information security conditions of the software product under evaluation, as depicted in Fig. 4.

2.3.1. Artifacts of SEC-AM

As a consequence of the interaction of the different actors involved in the Security Audit Process, the following artifacts are generated:

- **SEC-CAT:** The catalog of security requirements outlined in the above section is the base element of the audit method since it collects knowledge about standards, guides, regulations, and research inherent in security.
- **SEC-CAT*:** Generally, it is an adaptation of SEC-CAT to the application where the audit method will be applied, containing a subset of requirements but retaining all the properties specified within each requirement.
- **Checklist:** This is a list of elements that will be verified against the e-Health system in the study. It is generated based on the SEC-CAT* artifact and can take on the form most appropriate to the audited software. It may consist of a web-based questionnaire or any other electronic or paper format. In our case, the Checklist is in paper format.
- **Audit report:** This is the result of the audit process and may comprise preliminary, in-depth reports and a final report that includes the security deficiencies, areas to improve in the audited application, the audit team's recommendations, and a summary of the primary vulnerabilities.

2.3.2. Roles of SEC-AM

The audit technique is carried out by the players or roles listed below:

[3] https://bit.ly/4fMJhLV.

[4] https://www.vanta.com/demo.
[5] https://drata.com/.
[6] https://tugboatlogic.com/.
[7] https://www.strikegraph.com/.
[8] https://www.auditboard.com/.
[9] https://secureframe.com/.

Table 4
Example of a requirement. The source field has two items.

PUID: [SEC-CAT-BAC-003]
Requirement description: The application must implement security measures to prevent the elevation of privileges. Prevent situations where a user can act without being authenticated or an administrator can act as a regular user after logging in. **Source:** **A01:2021 – Broken Access Control:** Avoid elevation of privilege. Acting as a user without being logged in or acting as an admin when logged in as a user [3]. **CWE-306: Missing Authentication for Critical Function:** The product does not perform any authentication for functionality that requires a provable user identity or consumes a significant amount of resources [14].

- **Client:** Any individual or business needing an audit to assess the security characteristics of an e-Health system.
- **Engineering Team (EN):** The group responsible for developing, installing, setting up, or managing the e-Health system.
- **Security Team (SE):** The team is tasked with analyzing and formulating security policies and configurations, assessing vulnerabilities, conducting penetration tests, and providing detailed insights into the vulnerabilities within the e-Health system.
- **Audit Team:** The person or group in charge of completing the audit should ideally have previous expertise in auditing e-Health systems; however, this is not mandatory.
- **Requirement Engineering Team (RE):** Group or people who elaborated and maintained the SEC-CAT and SEC-CAT*. Individuals with backgrounds as requirements engineers or analysts are preferred.

2.3.3. *Phases and activities of SEC-AM*

The following are the tasks included in the security audit process:

a. **Context analysis:** The EN Team thoroughly examines the operational settings and conditions of the targeted e-Health system for the audit. It involves thoroughly assessing its context, encompassing background information, environment, purpose, scope, user profiles, and other pertinent factors. Subsequently, leveraging the functional context provided by the EN Team, the SE Team formulates security policies, defines configurations, outlines the execution environment,

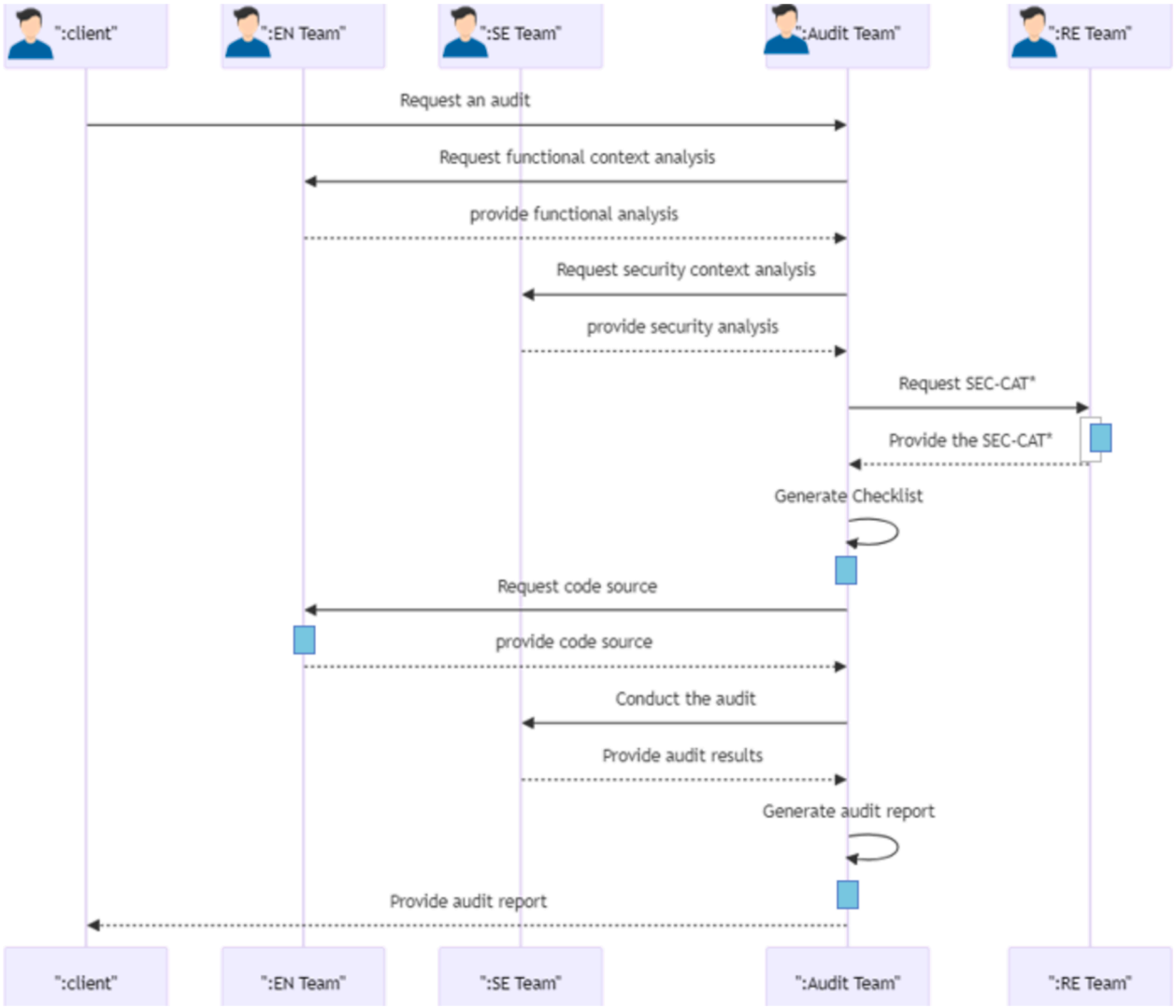


Fig. 4. Security Audit Method (SEC-AM).

identifies risks, and specifies the desirable behavior for the application under audit.

b. **SEC-CAT* generation:** The RE equipment, considering the context analysis described in the previous step, defines a new adapted catalog from the requirements specified in the SEC-CAT.
c. **Generate Checklist:** The audit team establishes the Checklist for conducting the audit, using the SEC-CAT* as a basis.
d. **Execute the audit:** The Security Team (SE) experts thoroughly examine the application under audit, utilizing a thorough checklist.

Nowadays, for the analysis and audit of vulnerabilities, there are some approaches:

- **SAST** (Static Application Security Testing) to detect problems in the source code before its implementation. [108]
- **DAST** (Dynamic Application Security Testing) to evaluate security while the application is being executed in a real environment. [109]
- **IAST** (Interactive Application Security Testing) evaluates the application while running and analyzes the source code simultaneously. [110]
- **Manual testing** is used to assess specific requirements compliance with the application. [111]

A combination of three approaches has been selected because of the comprehensive battery of tests to detect bad practices and possible vulnerabilities in applications that combine SAST (HP Fortify SCA V4.40) [112], DAST (with Acunetix v 23.7.230728157) [113] and a manual assessment for specific security requirements [114].

e. **Generate audit report:** This phase encompasses the prospective establishment of interim audit reports, succeeded by the subsequent compilation of the conclusive audit report. During this stage, the audit team assembles comprehensive outcomes from the audit technique, integrating thorough information and suggestions into the final report.

3. Results

To validate the SEC-CAT presented in this research, a comprehensive audit was conducted on an actual open-source e-Health application, leveraging access to the source code for a thorough examination. The selected application was OpenEMR v7.0 [115], acknowledged as the premier open-source platform for electronic medical records and medical practice management, and it holds ONC 2015 Cures Update certification. OpenEMR boasts widespread global usage, with over 100,000 medical professionals in more than 100 countries providing healthcare services to a patient population exceeding 90 million people globally [116,117,118]. The insights derived from this audit aim to benefit the application's present and future users.

The audited software is available for Docker, AWS Cloud, Windows, and Linux Systems [119]. OpenEMR was analyzed employing the SEC-AM. The complete process is detailed as follows.

3.1. Context analysis

The initial stage of the SEC-AM method involves examining the application, users, software attributes, potential deployment settings, functionality, and components. The audited application has the following functionality:

- Patient Demographics
- Patient Scheduling
- EMR
- e-Prescribing
- Medical Billing
- CMS Reporting
- Lab Integration
- Clinical Decision Rules
- Advanced Security
- Multilanguage Support

This comprehensive evaluation was performed by Oct. 19, 2023, extracting pertinent information from the official website [120]. Further details regarding OpenEMR can be accessed in Table 5.

3.2. Generate SEC-CAT*

Starting from the context analysis, the RE Equipment generated the SEC-CAT*[10] adapted from the SEC-CAT[11] proposed in our research. Since OpenEMR is not released to a specific user group, the default values have been taken to those requirements containing configurable parameters.

3.3. Generate Checklist

In this phase, a Checklist will be produced from the SEC-CAT*, which was utilized by the audit team to assess the OpenEMR application. In the specific instance of this audit, the Checklist document,[12] consisted of 142 items, and each check element will possess the subsequent potential response alternatives:

- "Yes" if the condition is consistently met.
- "No" if the condition is not consistently met.
- "N/a" when the condition is not applicable.

Partial responses were disregarded due to their lack of contribution to the security requirements. Non-compliance is inferred from the violation of a single-premise argument. The security coverage of the application was then assessed in terms of the total percentage score obtained, which was subsequently discretized into a 5-interval scale using an unsupervised global discretization method [121]. This approach is a modified version of equal width interval binning [122], where the lower and upper bins are smaller than the others, aiming to differentiate between extreme scores and categorize each assessment into:

- Very high coverage
- High coverage
- Moderate coverage
- Low coverage
- Very low coverage

The score boundaries for every group are specified in Table 6.

Table 5
Information on OpenEMR v7.0 from its official website.

Name:	*OpenEMR*
Developer:	Open-emr.org
Category	EHR and medical practice management solution
Supported platforms	Windows, Linux, Mac OS X
Version	7.0.0
Last update	11/12/2023
Compatible idioms	More than thirty
Source	Open source
License	GNU General Public License
Main language	PHP + MySQL

[10] https://bit.ly/3SBypWV.
[11] https://bit.ly/4fMJhLV.
[12] https://bit.ly/3SHjwm3.

3.4. Audit execution and audit report

The outcomes following the meticulous application of the SEC-AM detailed process to the OpenEMR application are depicted in Fig. 5. Additionally, the audit team conducted a thorough analysis, intricately documenting comprehensive details of the code files affected by each vulnerability uncovered during the DAST and SAST scans. This exhaustive examination involved scrutinizing all recorded results in the Checklist, ultimately leading to the development of the final report.[13]

4. Discussion

Upon reviewing all checklist items, the security coverage percentage of the OpenEMR application is calculated to be 66.97 %. Of the 142 items assessed, 88 were deemed compliant ("YES"), while 35 did not meet the specified criteria. Furthermore, 19 items were marked as "N/A". Alternatively, if we focus solely on the 123 items applicable to the application (either "YES" or "NO"), the security level of OpenEMR stands at 71.54 %. Consequently, the application is classified by the percentage obtained in the "High coverage" category. The most common vulnerabilities in OpenEMR were DDOs, XSS, JSONi, and CMDi. Table 7 presents the details of the vulnerabilities detected according to the method used.

Effective access control is crucial for minimizing the risk of security breaches, enhancing operational efficiency, and reducing the potential for users to initiate DOs or DDOs attacks that could overwhelm server resources [123]. However, OpenEMR has been found to have a deficiency in access control, specifically in the lack of implementation or enforcement of dual access control measures before deleting medical and user records, leading to a "No" assessment. Failure to adhere to this requirement exposes the system to the possibility that a single user could carry out critical or sensitive actions without additional supervision, thereby increasing the risk of privilege abuse and malicious activities. This deficiency also directly threatens the integrity and confidentiality of stored information, allowing a single user to access sensitive data without the necessary oversight.

Three identified issues, marked as "No", are linked to deficiencies in data entry validation measures, posing potential vulnerabilities to XSS, XSLT, JSON, and CMDi. Previous research anticipates a rising trend in security breaches within medical care devices and software [124]. According to Static Application Security Testing (SAST), OpenEMR exhibits susceptibility to 20 XSS-related vulnerabilities in its PHP files, primarily within the administrator and medical history reviewer modules. Recent studies highlight that code or command injection is a severe threat capable of disrupting program execution, potentially leading to data loss, corruption, accountability issues, or denial of access [125].

User notification before granting access is not complied with by OpenEMR, so one check is assessed as "No". Consequently, users do not know the property of the system, the monitoring, recording, and auditing of use, the penalties for unauthorized use, and the acceptance of these conditions. In systems dealing with confidential and critical information, focused attention becomes crucial by requiring users to accept the terms and conditions before accessing the application in adherence to the applicable legal measures [126].

Table 6
Security coverage score groups.

Security Coverage Group	Interval score
Very high coverage	(87.5 % – 100 %]
High coverage	(62.5 % – 87.5 %]
Moderate coverage	(37.5 % – 62.5 %]
Low coverage	(12.5 % – 37.5 %]
Very low coverage	[0 % – 12.5 %]

[13] https://bit.ly/4d78UVZ.

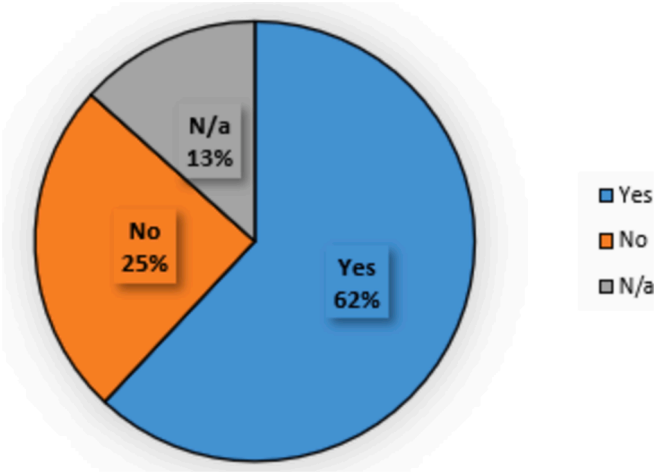


Fig. 5. Checklist responses.

Four items on the Checklist are flagged as "No" due to inherent vulnerabilities arising from an insecure design, encompassing issues such as the lack of isolation between user and management functionalities, the consolidation of all functions within a general-purpose server, the inclusion of authentication data in code files, and potential manipulation of routes. This non-compliance escalates the risk of unauthorized access, undermining the certainty that system management functions are exclusively executed by authorized personnel and deviating from established data protection practices. Approximately 74 % of security breaches are linked to human factors like errors, privilege misuse, stolen credentials, or social engineering [127].

Four additional items in the Checklist are marked as "No" and are linked to security misconfigurations. These include (i) lack of security configurations, (ii) insufficient anti-DDOs security settings, (iii) disclosure of technical details to users in case of errors, and (iv) no secure flag configuration in cookies. In contemporary cybersecurity, misconfiguration is one of the most formidable risks [128].

Table 7
Vulnerabilities found.

Vulnerability	Detection method	Flaws number
Authentication and password problems	Manual	8
Log weaknesses		7
Account management		1
No notification before granting access		1
Separation of system and user functionality		1
Application housed on a general-purpose server		1
No notifications or alerts for product updates		1
XSS	SAST	20
Path manipulation		7
JSON injection		7
Misconfiguration		4
DDOs		3
Password Management		2
Command Injection		2
Header manipulation		1
XSLT		1
Variable override		1
Vulnerable packages dependencies	DAST	21
Misconfiguration		13
Display of sensitive information		6
Outdated libraries		5
Weak cryptography		2
Hardcoded passwords		1

Three Checklist requirements concerning Vulnerable and Outdated Components have been identified as "No". OpenEMR lacks notifications or updates regarding product updates; instead, it contains outdated libraries, some of which may even lack support. Insufficient allocation of time and resources by organizations for the timely updating of software and ensuring system security has resulted in critical vulnerabilities [31]. Outdated systems become fertile ground for attackers to inject malware into EMR systems, potentially leading to the execution of malicious code and compromising the system functionality, causing disruptions in patient care and medical operations. Unauthorized access by attackers further jeopardizes patient data confidentiality, integrity, and availability. Beyond security breaches, attackers seek healthcare data for financial gain, exploiting insurance credentials and swiftly accumulating profits through fraudulent activities [31]. The heightened risk of data breaches exposes sensitive patient information to unauthorized parties and carries significant legal, financial, and reputational implications for healthcare providers. The presence of vulnerable or outdated components also threatens regulatory compliance, such as with HIPAA in the United States, resulting in legal consequences and penalties. Additionally, manipulating or altering medical records due to known vulnerabilities in outdated components directly threatens patient safety, emphasizing the urgency for healthcare organizations to implement robust security practices; it includes regular software updates, patching, comprehensive vulnerability assessments, and adherence to industry best practices in healthcare cybersecurity to proactively uphold the integrity of EMR systems and safeguard sensitive information.

Seven requirements associated with identification and Authentication Failures are checked as "No". It was due to (i) a lack of safe login procedures that protect against login attempts by brute force, (ii) missing reauthentication before executing critical actions, (iii) flaws in password and account management, and (iv) lack in restrict the duration of connections to application services. These seven identified flaws significantly contribute to the vulnerability of EMR systems, creating an environment conducive to brute-force attacks. In such attacks, malicious actors persistently try to gain unauthorized access, compromising the system integrity. The absence of reauthentication requirements before executing critical actions, such as modifying medical records or accessing sensitive information, further exacerbates these security risks. Weak or easily exploitable passwords become potential targets for attackers, undermining the confidentiality and security of patient information within OpenEMR. Extended connection durations present a vast opportunity for attackers to exploit vulnerabilities, potentially leading to unauthorized access, data manipulation, or service disruptions. It is imperative to counter these security consequences in EMR systems by adopting robust login procedures, enforcing reauthentication measures for critical actions, enhancing password and account management practices, and implementing appropriate session duration restrictions. A captcha mechanism can mitigate force attacks [129,130].

Five checklist items associated with software and data integrity were marked as "No". The audit team cannot prove that OpenEMR libraries and components are exclusively obtained from official sources, thereby lacking secure mechanisms for the software supply chain. Likewise, it has not been demonstrated that the coding reviewing process through continuous integration channels/deployment to prevent the inclusion of malicious code or configurations is satisfied. The absence of assurance in these aspects poses tangible risks, as acquiring software from unofficial repositories exposes it to potential malware, leading to infections in both the central system and clients' devices [131]. Malware can spread across the entire hospital network, resulting in unavailability, disruptions, and data loss [132]. To address these critical vulnerabilities within EMR systems, enforcing stringent policies mandating the exclusive sourcing of libraries and components from reputable, official channels is imperative. Furthermore, implementing a robust and continuous code review process through integration channels becomes pivotal in averting the inclusion of malicious code, thereby safeguarding the security, integrity, and reliable operation of the EMR system.

Another security flaw in OpenEMR is the lack of robust security logging and monitoring. Security records lack specific information, automatic backup of audited records is not conducted, and audit records are not retained for a designated period. Additionally, there is no configurable mechanism to transfer audit records to a different system for further analysis, audit records when a concurrent session is not produced, and there is no option to alert the ISSO and the SA in case of a failure in audit processing. When different actions and application configurations are not audited, identifying attempted attacks will be challenging, and there will be no audit trail for forensic investigation of actions taken after the fact [95].

One check was marked as "N/a" because OpenEMR is not only an API that offers methods: PUT, POST, DELETE, or GET linked to a flexible access flow. So, the requirement is not applicable. Another check was evaluated as "N/a" because it relates to message encryption when the SessionIndex is linked to privacy data. However, OpenEMR does not link SessionIndex to privacy data. Twelve more checks are classified as " N/a " because they are related to elements and ways of communication using SAML assertions in SOAP messages. However, audited software does not implement SOAP messages as described in the requirements. Four identification and authentication failure checks qualified as "N/a" because audited OpenEMR does not implement PKI-based authentication. Finally, one requirement is "N/a" related to non-signed or non-encrypted serialized data not sent to non-reliable customers without integrity verification or digital signature. The audit team has reviewed the software documentation, and there is no information about this feature; consequently, it does not apply.

In OWASP Top 10 [96] and the DISA ASD STIG [95], there are several recommendations to solve the safety gaps and identified requirements promptly in the SV-222624R864409_RULE of DISA ASD STIG, the importance of automated Fuzzing tests with manual corroboration is based.

4.1. Lessons learnt

The features met in the proposed framework foster applicability, auditability, reusability and traceability:

1. **Applicability:** Meets industry standards, ensuring relevance for various e-Health applications. The SIREN methodology supports this by providing structured and standardized requirements.
2. **Auditability:** Uses a comprehensive audit methodology that combines DAST, SAST and manual assessments, with the SEC-CAT tool facilitating compliance assessment and detailed audit reports.
3. **Reusability**: Includes a repository of reusable requirements, structured for easy adaptation to different applications, supported by the SIREN methodology and allowing parameterization.
4. **Traceability:** Requirements are organized in catalogs according to ISO/IEC/IEEE 29148:2018 and OWASP Top 10 Web Application Security Risks: 2021, with detailed attributes that allow tracking throughout their lifecycle.

Through the audit of security requirements in e-Health applications, we have identified several valuable lessons that can benefit both researchers and professionals in the field of computer security and health applications. Below are the most notable lessons from our research:

1. **Use of Checklists for Combined Assessments:** The proposed audit methodology, which includes a checklist, made it possible to evaluate the quality of the application under test through a combination of DAST (Dynamic Application Security Testing), SAST (Static Application Security Testing) approaches and manual evaluations. This combined approach provides a comprehensive and detailed view of application security, improving audit accuracy.
2. **SRS Enrichment for Health Applications:** The security requirements audit revealed that the SRS (Software Requirements

Specification) product within the framework of this research can significantly clarify and enrich the landscape of desired security requirements for health-related applications. This will help developers and auditors to better understand the expectations and standards necessary to ensure the security of these applications.

3. **Facilitation of Compliance with SEC-CAT:** The SEC-CAT tool facilitated the compliance assessment of existing or newly developed applications, providing a clear understanding of the situational status of the software product through the audit report. This tool is crucial to ensure that applications meet established security requirements.
4. **Involvement of Security Experts:** In the proposed framework (SEC-AM), the participation of computer security experts is essential to ensure proper audit scope and effective identification of vulnerabilities. The experience and knowledge of a specialized security team are vital to effectively identify and mitigate potential vulnerabilities.
5. **Expansion of Recommendations with SEC-CAT:** SEC-CAT can enrich and expand established recommendations and controls, serving as a technical complement to the requirements for test cases in the security of health applications, especially those aimed at meaningful use of data. This tool provides an additional layer of rigor and detail in security assessment.
6. **Complement to IT Governance:** Security analysis work from the operational level complements and strengthens IT governance, accurate decision-making and cyber-attack prevention by corporate leaders, senior managers, IT cybersecurity professionals and experts, and specialists in cybersecurity interested in strategic issues. Integrating operational analytics improves cyber resilience and incident response capabilities.
7. **Importance of Effective Leadership:** Effective leadership is crucial to ensuring corporate attention to information security, fostering a culture of cyber resilience and developing a comprehensive approach to security. This includes involvement and support of senior management in cybersecurity strategies to improve governance, integration and transformational support in digital business environments [133].
8. **Improvements in Process Automation:** Tooling support can improve the audit process thus saving time and reducing error-prone activities. As a result, the effectiveness and efficiency of the process are enhanced. Since we currently lack a specific support tool for the method, our research group is making efforts to adapt another NFR audit tool focused on usability to the security domain [134].

Finally, the lessons learned from this research improve our understanding of security in e-Health applications and provide valuable guidance for future research and practice.

4.2. Threats to validity

Two kinds of threats to validity are discussed below:

Construct validity: Threats to validity in a mapping study concern identifying primary studies [135,136]. A sensitivity analysis of the publications retrieved from the Scopus database was conducted to assess the accuracy of the results of the search. Among 108 publications, none provided security requirements with a formal structure or format, thus mitigating this validity threat.

Internal validity: Internal validity focuses on the process of extracting and analyzing data [135,136]. To avoid overlapping concepts in the requirements contained in the SEC-CAT, the authors have carried out harmonization of the base documents using a process adapted from the framework proposed by C. Pardo et al. [98].

5. Conclusions and future work

This research reviews the literature related to security requirements for e-Health from 1991 to July 9, 2023. It served as a starting point to provide a comprehensive catalog of safety requirements for e-Health software with rationale and validation criteria that are continually improvable, unambiguous, complete, consistent, verifiable, modifiable, and traceable over time.

Security requirements engineering in e-Health must adapt to current challenges, incorporating solid practices and advanced technologies to protect patient data and ensure system reliability. Requirements should be based on existing regulations, guides, standards, and best practices to improve regulatory compliance and software quality. This approach increases productivity, security, and quality throughout the software development.

Similarly, this work presents an SEC-AM audit method to evaluate safety in e-Health applications through a requirement catalog. The catalog and the audit method were validated in OpenEMR, one of the most popular applications for managing electronic medical records. The proposed audit method measures the security of an application following the criteria of the security catalog adapted to the application (SEC-CAT*). The security classification obtained after applying the SEC-AM places the software in the “secure” category. Considering that Checklist is based on closed response options: “Yes”, “No”, or “N/a”, SEC-AM was a simple process and did not present any difficulty for the auditor’s team.

The SEC-AM, designed as an iterative method, exhibits complexity and execution time correlating with the number of requirements to be evaluated. In our specific scenario, SEC-CAT comprises 142 requirements that necessitate assessment through a comprehensive approach involving SAST, DAST, and manual verification.

Therefore, organizations interested in the security of e-Health applications and the certification of EHR systems can leverage SEC-CAT to evaluate, audit, or enhance the safety measures for testing software products better before introducing them to the market.

The authors of this scientific work have reported the vulnerabilities detected in the OpenEMR v7.0.0 software to the OpenEMR community, ensuring that the issues are known and validated and that risk mitigation practices are implemented to enhance the security of this vital healthcare application; the vulnerabilities were submitted via email, as specified on the official software portal.

As part of future work, the research team could develop a methodology for manual verification and exploitation of possible vulnerabilities in e-Health software for training purposes. A guide and recommendations with practical examples to help software developers avoid and solve safety bugs can be proposed.

To complete the audit method outlined in this research, an open-source computer tool supported by artificial intelligence could be developed to automate the assessment of organizational security controls, encompassing policy controls, versioning, infrastructure, and deployment. This automation will draw upon periodically published vulnerabilities from the NVD and a customized catalog of security requirements. On the other hand, a more fine-grained measure of a system’s vulnerability should be provided. Obviously, a higher percentage of vulnerability coverage means greater security. However, a simple percentage does not take account of specific potential risks of the system for each uncovered vulnerability. The precise percentage required to ensure “reasonable security” against real-world vulnerabilities cannot currently be defined with certainty. When evaluating an application’s security, it is essential to adopt a comprehensive approach that addresses perceived security, privacy policy transparency, and the security of service delivery [120]. A holistic evaluation encompassing these aspects can help determine if an application has “reasonable coverage,” ensuring user trust and regulatory compliance [137]. This challenge will be the subject of future work.

As future work, the use of managerial-level security coverage metrics obtained through the application of the SEC-AM audit method can be considered. Each vulnerability linked to a requirement that is not met may be reported. This non-compliance would be reflected in a health information system security dashboard. Integrated into the DevSecOps

approach, this dashboard would provide early and real-time feedback of the evaluated systems, facilitating informed decision making and the establishment of terms of reference for the development and acquisition of IT solutions. This integration would improve the efficiency and effectiveness of the audit process, ensuring that e-Health systems meet high security standards on an ongoing basis and promoting a culture of constant improvement and regulatory compliance.

Moreover, as a new contribution to strengthening the study and definition of security requirements in mobile health applications, we propose developing a security requirements catalog for m-health applications. This catalog can be implemented and evaluated through a continuous audit method aligned with continuous integration methodologies like DevSecOps.

As a future effort, a comprehensive catalog of privacy requirements for e-Health applications, aligned with norms, standards, guidelines, and academic works, could be developed to address the identified gaps. This catalog will improve privacy protection in these applications by providing precise, actionable requirements that integrate with frameworks such as OECD Privacy Principles, GDPR, privacy legislation in the United Kingdom and Canada, ISO/IEC 27701, NIST Privacy Framework, and other resources oriented toward privacy. Thus, e-Health applications will meet regulatory standards, build user trust, and effectively protect personal health information.

6. Summary table

What was already known on the topic

- E-Health applications have advantages but face security breaches, leading to financial, social, and legal risks and even endangering patients' lives.
- Cases of information security incidents are increasing, exposing the personal health information of millions of patients per year.
- Improving e-Health security by addressing software application vulnerabilities and deficiencies is critical.

What this study added to our knowledge

- A security catalog (SEC-CAT) with reusable, traceable, applicable, and auditable requirements developed by e-Health experts facilitates the developing and auditing of more secure, efficient, interoperable, and reliable e-Health applications.
- A flexible, technical and objective criteria to measure the percentage of compliance in implementing and developing e-Health applications by an audit method.
- A layer of abstraction that enhances synergy between technical security and IT governance, strengthening decision-making and cyber-attack prevention for corporate leaders, managers, IT security professionals, technology experts, cybersecurity specialists, and academics.

Term	Definition
e-Health	*"(…) e-Health is the cost-effective and secure use of information and communication technologies (ICT) in support of health and health-related fields. It encompasses multiple interventions, including telehealth, telemedicine, mobile health (mHealth), electronic medical or health records (eMR/eHR), big data, wearables, and even artificial intelligence. The role of e-Health has been recognized as pivotal in attaining overarching health priorities such as universal health coverage (UHC) and the Sustainable Development Goals (SDGs). (…)"* [138]
ISO 29148:2018	*"(…) This document specifies the required processes implemented in the engineering activities that result in requirements for systems and software products (including services) throughout the life cycle; provides guidelines for applying the requirements and requirements related processes described in ISO/IEC/IEEE 15,288 and ISO/IEC/IEEE 12207; specifies the required information items produced through the implementation of the requirements processes; specifies the required contents of the required information items; provides guidelines for the format of the required and related information items. (…)"* [69]
ISO 27002:2022	This document, intended for organizations of all types and sizes, is a reference for determining and implementing controls for information security risk treatment in an ISMS based on ISO/IEC 27001. It also guides implementing commonly accepted information security controls and developing industry and organization-specific information security management guidelines tailored to their specific risk environments, with additional controls determined through risk assessment as needed. [60]
ISO 27799:2016	This International Standard offers healthcare organizations and custodians of personal health information guidance on protecting its confidentiality, integrity, and availability, extending the general guidance of ISO/IEC 27002:2013 to address the unique information security management needs of the health sector [90]
HIPAA	The Health Insurance Portability and Accountability Act of 1996 (HIPAA) is a federal law establishing national standards to protect sensitive patient health information from unauthorized disclosure. The US Department of Health and Human Services (HHS) issued the HIPAA Privacy Rule to implement HIPAA's requirements. This rule applies to all HIPAA-covered entities and certain federal agencies, mandating the confidentiality, integrity, and availability of EPHI based on their functions and use of such information. [91,139]
HIPAA Security rules	The HIPAA Security Rule mandates the protection of electronically protected health information (EPHI) by requiring all HIPAA-covered entities, including some federal agencies, to safeguard the confidentiality, integrity, and availability of EPHI. This rule ensures that EPHI created, received, maintained, or transmitted by these entities is protected against reasonably anticipated threats, hazards, and unauthorized uses or disclosures. [92,140]
GDPR	The GDPR (General Data Protection Regulation) is an EU regulation that mandates stringent data protection and privacy measures for individuals within the EU and EEA, ensuring secure handling and control of personal data while protecting fundamental rights and allowing free movement of data within the Union without restrictions due to privacy concerns. [141]
NIST SP 800–53 rev. 5	Establishes the security and privacy controls for information systems and organizations, which may be used voluntarily by nongovernmental organizations and is not subject to copyright in the United States. This

(continued on next column)

(continued on next page)

(*continued*)

Term	Definition
	document is developed by the National Institute of Standards and Technology under the Federal Information Security Modernization Act (FISMA). [93]
Privacy & Security Requirements and Considerations for Digital Health Solutions	It addresses the privacy and security challenges of new digital health solutions, such as remote patient monitoring and consumer health solutions, and emerging technologies, such as cloud computing and mobile devices [94]
DISA ASD STIG: 2022	This Security Technical Implementation Guide, based on NIST 800–53 and related documents, is designed to enhance the security of Department of Defense (DOD) information systems. [95]
OWASP Top 10 Web Application Security Risks:2021	The OWASP Top 10 is a widely recognized standard awareness document that outlines the most critical security risks to web applications, providing essential guidance for developers and web application security.[93,142]
CWE/SANS TOP 25: 2021	The CWE™ Top 25 list identifies the most dangerous software weaknesses, which are easy to exploit and can lead to severe vulnerabilities, serving as a critical resource for software professionals to mitigate risks. [97,143]

CRediT authorship contribution statement

Carlos M. Mejía-Granda: Writing – original draft, Methodology, Investigation, Conceptualization, Data curation, Formal analysis, Software, Visualization, Writing – review & editing. **José L. Fernández-Alemán:** Supervision, Resources, Project administration, Conceptualization, Funding acquisition, Investigation, Methodology, Validation, Writing – original draft, Writing – review & editing. **Juan M. Carrillo de Gea:** Writing – review & editing, Investigation, Methodology, Supervision, Validation, Writing – original draft. **José A. García-Berná:** Visualization, Data curation, Investigation, Methodology, Supervision, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study is a component of the OASSIS-UMU (PID2021-122554OB-C32) project and the Network of Excellence in Software Quality and Sustainability (RED2022-134656-T), all of which are funded by the Spanish Ministry of Science, Innovation, and Universities. Additionally, the European Regional Development Fund established these initiatives (ERDF).

References

[1] P. Singh, E-Health Application for E-Blood Analysis, E-Diagnosis, and Digital Diet Guidance, *Adv. Exp. Med. Biol.* vol. 1194 (2020) 343–350, https://doi.org/10.1007/978-3-030-32622-7_32.

[2] S. Roy, U. Roy, D. Sinha, R.K. Pal, Imbalanced ensemble learning in determining Parkinson's disease using Keystroke dynamics, *Expert. Syst. Appl.* 217 (2023), https://doi.org/10.1016/j.eswa.2023.119522.

[3] H. Van Uytsel, et al., Effect of the e-health supported INTER-ACT lifestyle intervention on postpartum weight retention and body composition, and associations with lifestyle behavior: A randomized controlled trial, *Prev. Med (baltim)* 164 (2022), https://doi.org/10.1016/j.ypmed.2022.107321.

[4] C. Antunes, C. Coutinho, Employment of Artificial Intelligence Mechanisms for e-Health Systems in Order to Obtain Vital Signs Improving the Processes of Online Consultations and Diagnosis, in: 2022 International Symposium on Sensing and Instrumentation in 5G and IoT Era (ISSI), 2022, pp. 109–114, https://doi.org/10.1109/ISSI55442.2022.9963223.

[5] S. Ouhbi, J.L. Fernández-Alemán, J.M. Carrillo-de-Gea, A. Toval, A. Idri, E-health internationalization requirements for audit purposes, *Comput Methods Programs Biomed* 144 (2017) 49–60, https://doi.org/10.1016/j.cmpb.2017.03.014.

[6] J. D. Young and A. I. Anton, A Method for Identifying Software Requirements Based on Policy Commitments, in: 2010 18th IEEE International Requirements Engineering Conference, 2010, pp. 47–56. https://doi.org/10.1109/RE.2010.17.

[7] H. S. Gardiyawasam Pussewalage and V. A. Oleshchuk, Privacy preserving mechanisms for enforcing security and privacy requirements in E-health solutions, Int. J. Inf. Manage., vol. 36, no. 6, Part B, pp. 1161–1173, 2016, https://doi.org/10.1016/j.ijinfomgt.2016.07.006.

[8] J.L. Fernández-Alemán, A.B.S. García, G. García-Mateos, A. Toval, Technical solutions for mitigating security threats caused by health professionals in clinical settings, in: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2015, pp. 1389–1392, https://doi.org/10.1109/EMBC.2015.7318628.

[9] J.L. Fernández-Alemán, A. Sánchez-Henarejos, A. Toval, A.B. Sánchez-García, I. Hernández-Hernández, L. Fernandez-Luque, Analysis of health professional security behaviors in a real clinical setting: An empirical study, Int. J. Med. Inform. 84 (6) (2015) 454–467, https://doi.org/10.1016/j.ijmedinf.2015.01.010.

[10] B. Zapata, J. Fernández-Alemán, A. Toval, Security in Cloud Computing: a Mapping Study, Comput. Sci. Inf. Syst. 12 (Jan. 2015) 161–184, https://doi.org/10.2298/CSIS140205086C.

[11] D. Mairiza, D. Zowghi, N. Nurmuliani, An Investigation into the Notion of Non-Functional Requirements, in Proceedings of the 2010 ACM Symposium on Applied Computing, in SAC '10, Association for Computing Machinery, New York, NY, USA, 2010, pp. 311–317.

[12] T. M. K. Kumar, A Road Map to the Software Engineering Security, in: Proceedings of the 2009 Second International Conference on Computer and Electrical Engineering - Volume 02, in ICCEE '09. USA: IEEE Computer Society, 2010, pp. 306–310. https://doi.org/10.1109/ICCEE.2009.62.

[13] N. R. Jennings, Agent-Oriented Software Engineering, in: Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World: MultiAgent System Engineering, in MAAMAW '99. Berlin, Heidelberg: Springer-Verlag, 1999, pp. 1–7.

[14] G. Marquez, H. Astudillo, C. Taramasco, Security in Telehealth Systems from a Software Engineering Viewpoint: A Systematic Mapping Study, IEEE Access 8 (2020) 10933–10950, https://doi.org/10.1109/ACCESS.2020.2964988.

[15] H.T. Neprash, et al., Trends in Ransomware Attacks on US Hospitals, Clinics, and Other Health Care Delivery Organizations, 2016-2021, JAMA Health Forum 3 (12) (2022) E224873, https://doi.org/10.1001/jamahealthforum.2022.4873.

[16] S.P. Keehan, et al., National health expenditure projections, 2019–28: Expected rebound in prices drives rising spending growth, Health Aff. 39 (4) (2020) 704–714, https://doi.org/10.1377/HLTHAFF.2020.00094.

[17] B. Aljedaani, A. Ahmad, M. Zahedi, M.A. Babar, End-users' knowledge and perception about security of clinical mobile health apps: A case study with two Saudi Arabian mHealth providers, *J. Syst. Softw.* 195 (2023), https://doi.org/10.1016/j.jss.2022.111519.

[18] J.C. Maxwell, A.I. Antón, in: The Production Rule Framework: Developing a Canonical Set of Software Requirements for Compliance with Law, Association for Computing Machinery, New York, NY, USA, 2010, pp. 629–636, https://doi.org/10.1145/1882992.1883092.

[19] T. Breaux, A. Antón, Analyzing Regulatory Rules for Privacy and Security Requirements, IEEE Trans. Softw. Eng. 34 (1) (2008) 5–20, https://doi.org/10.1109/TSE.2007.70746.

[20] F.T. Chimuco, J.B.F. Sequeiros, C.G. Lopes, T.M.C. Simões, M.M. Freire, P.R. M. Inácio, Secure cloud-based mobile apps: attack taxonomy, requirements, mechanisms, tests and automation, Int. J. Inf. Secur. 22 (4) (2023) 833–867, https://doi.org/10.1007/s10207-023-00669-z.

[21] J.D. Young, Commitment analysis to operationalize software requirements from privacy policies, Requir. Eng. 16 (1) (2011) 33–46, https://doi.org/10.1007/s00766-010-0108-6.

[22] A.K. Massey, P.N. Otto, L.J. Hayward, A.I. Antón, Evaluating existing security and privacy requirements for legal compliance, Requir. Eng. 15 (1) (2010) 119–137, https://doi.org/10.1007/s00766-009-0089-5.

[23] G. McGraw, Software Security: Building Security In, Addison-Wesley Professional, 2006.

[24] B. Smith, et al., Challenges for Protecting the Privacy of Health Information: Required Certification Can Leave Common Vulnerabilities Undetected, in: Proceedings of the Second Annual Workshop on Security and Privacy in Medical and Home-Care Systems, in SPIMACS '10, Association for Computing Machinery, New York, NY, USA, 2010, pp. 1–12.

[25] E.R. Aruna, A.R.M. Reddy, K.V.N. Sunitha, in: elicitation and Analysis of Security Requirements and Patterns for IoT Based Health Monitor, in: Advances in Cybernetics, Cognition, and Machine Learning for Communication Technologies, Springer Singapore, Singapore, 2020, pp. 49–56, https://doi.org/10.1007/978-981-15-3125-5_6.

[26] Vulnerabilities | OWASP Foundation. [Online]. Available: https://owasp.org/www-community/vulnerabilities/.

[27] M. Alqaradaghi, M. Z. I. Nazir, and T. Kozsik, Design and Implement an Accurate Automated Static Analysis Checker to Detect Insecure Use of SecurityManager, Computers, vol. 12, no. 12, 2023, https://doi.org/10.3390/computers12120247.

[28] A. Agrawal, et al., Evaluating the Security Impact of Healthcare Web Applications Through Fuzzy Based Hybrid Approach of Multi-Criteria Decision-Making Analysis, IEEE Access 8 (2020) 135770–135783, https://doi.org/10.1109/ACCESS.2020.3010729.
[29] N. Kshetri, J. Voas, Ransomware as a Business (RaaB), IT Prof. 24 (2) (2022) 83–87, https://doi.org/10.1109/MITP.2022.3157208.
[30] FinCEN Analysis Reveals Ransomware Reporting in BSA Filings Increased Significantly During the Second Half of 2021 | FinCEN.gov. [Online]. Available: https://www.fincen.gov/news/news-releases/fincen-analysis-reveals-ransomware-reporting-bsa-filings-increased-significantly.
[31] L. Coventry, D. Branley, Cybersecurity in healthcare: A narrative review of trends, threats and ways forward, Maturitas 113 (Apr. 2018) 48–52, https://doi.org/10.1016/j.maturitas.2018.04.008.
[32] S.B. Weber, S. Stein, M. Pilgermann, T. Schrader, Attack Detection for Medical Cyber-Physical Systems-A Systematic Literature Review, IEEE Access 11 (2023) 41796–41815, https://doi.org/10.1109/ACCESS.2023.3270225.
[33] P. J. Escamilla Ambrosio et al., Securing mHealth applications using IoTsecM security modelling, Computacion y Sistemas, vol. 23, no. 4, pp. 1139–1158, 2019, https://doi.org/10.13053/CyS-23-4-3093.
[34] B. Aljedaani, A. Ahmad, M. Zahedi, M.A. Babar, End-users' knowledge and perception about security of clinical mobile health apps: A case study with two Saudi Arabian mHealth providers, J. Syst. Softw. 195 (2023) 111519, https://doi.org/10.1016/j.jss.2022.111519.
[35] R. Jáuregui-Velarde, D. H. Celis, C. Y. Arias, and L. Andrade-Arenas, A critical review of the state of computer security in the health sector, Bulletin of Electrical Engineering and Informatics, vol. 12, no. 6, pp. 3805–3816, 2023, https://doi.org/10.11591/eei.v12i6.5394.
[36] R.U. Rasool, H.F. Ahmad, W. Rafique, A. Qayyum, J. Qadir, Security and privacy of internet of medical things: A contemporary review in the age of surveillance, botnets, and adversarial ML, J. Netw. Comput. Appl. 201 (2022), https://doi.org/10.1016/j.jnca.2022.103332.
[37] D. Noori, H. Shakeri, M. Niazi Torshiz, An elliptic curve cryptosystem-based secure RFID mutual authentication for Internet of things in healthcare environment, *EURASIP J. Wirel. Commun. Netw.* 1 (2022) 2022, https://doi.org/10.1186/s13638-022-02146-y.
[38] M.-D. Cano, A. Cañavate-Sanchez, K. Sha, Preserving Data Privacy in the Internet of Medical Things Using Dual Signature ECDSA, Sec. Commun. Netw. 2020 (Jan. 2020), https://doi.org/10.1155/2020/4960964.
[39] A.M. Norouzzadeh Gil Molk, M.R. Aref, R. Ramazani Khorshiddoust, Leveled Design of Cryptography Algorithms Using Cybernetic Methods for Using in Telemedicine Applications, Comput. Intell. Neurosci. (2021), https://doi.org/10.1155/2021/3583275.
[40] R. K. N.V., M. K. P., Application of SDN for secure communication in IoT environment, Comput. Commun. 151 (2020) 60–65, https://doi.org/10.1016/j.comcom.2019.12.046.
[41] T. Tervoort, M.T. De Oliveira, W. Pieters, P. Van Gelder, S.D. Olabarriaga, H. Marquering, Solutions for Mitigating Cybersecurity Risks Caused by Legacy Software in Medical Devices: A Scoping Review, IEEE Access 8 (2020) 84352–84361, https://doi.org/10.1109/ACCESS.2020.2984376.
[42] A.J. Ghazali, W. Al-Nuaimy, A. Al-Ataby, M.A. Al-Taee, Building IPv6 based tunneling mechanisms for VoIP security, in: 2016 13th International Multi-Conference on Systems, Signals & Devices (SSD), 2016, pp. 171–176, https://doi.org/10.1109/SSD.2016.7473737.
[43] W. Iqbal, H. Abbas, M. Daneshmand, B. Rauf, Y.A. Bangash, An In-Depth Analysis of IoT Security Requirements, Challenges, and Their Countermeasures via Software-Defined Security, IEEE Internet Things J. 7 (10) (2020) 10250–10276, https://doi.org/10.1109/JIOT.2020.2997651.
[44] R. Rawat, V. Mahor, B. Garg, M. Chouhan, K. Pachlasiya, S. Telang, in: chapter Fifteen - Modeling of Cyber Threat Analysis and Vulnerability in IoT-Based Healthcare Systems during COVID, in Lessons from COVID-19, Academic Press, 2022, pp. 405–425, https://doi.org/10.1016/B978-0-323-99878-9.00016-9.
[45] S. S. Ambarkar and N. Shekokar, Toward Smart and Secure IoT Based Healthcare System, in Internet of Things, Smart Computing and Technology: A Roadmap Ahead, N. Dey, Parikshit. N. Mahalle, P. M. Shafi, V. V Kimabahune, and A. E. Hassanien, Eds., Cham: Springer International Publishing, 2020, pp. 283–303. https://doi.org/10.1007/978-3-030-39047-1_13.
[46] S. Kaddoura, R. A. Haraty, K. Al Kontar, and O. Alfandi, A Parallelized Database Damage Assessment Approach after Cyberattack for Healthcare Systems, Future Internet, vol. 13, no. 4, 2021, https://doi.org/10.3390/fi13040090.
[47] A. Sharma, H. Babbar, A.K. Vats, Detection of Attacks in Smart Healthcare deploying Machine Learning Algorithms*, in: 2023 4th International Conference for Emerging Technology (INCET), 2023, pp. 1–6, https://doi.org/10.1109/INCET57972.2023.10170367.
[48] M. Habiba, M.R. Islam, S.M. Muyeen, A.B.M.S. Ali, Edge intelligence for network intrusion prevention in IoT ecosystem, Comput. Electr. Eng. 108 (2023) 108727, https://doi.org/10.1016/j.compeleceng.2023.108727.
[49] I. Singh and S.-W. Lee, SRE_BBC: A Self-Adaptive Security Enabled Requirements Engineering Approach for SLA Smart Contracts in Blockchain-Based Cloud Systems, Sensors (Basel), vol. 22, no. 10, 2022, https://doi.org/10.3390/s22103903.
[50] D. Lee, M. Song, MEXchange: A Privacy-Preserving Blockchain-Based Framework for Health Information Exchange Using Ring Signature and Stealth Address, IEEE Access 9 (2021) 158122–158139, https://doi.org/10.1109/ACCESS.2021.3130552.
[51] M. Hijji, G. Alam, A Multivocal Literature Review on Growing Social Engineering Based Cyber-Attacks/Threats during the COVID-19 Pandemic: Challenges and Prospective Solutions, IEEE Access 9 (2021) 7152–7169, https://doi.org/10.1109/ACCESS.2020.3048839.
[52] E. A. P. Rincón and L. G. Moreno-Sandoval, Design of an architecture contributing to the protection and privacy of the data associated with the electronic health record, Information (Switzerland), vol. 12, no. 8, 2021, https://doi.org/10.3390/info12080313.
[53] A. Sengupta and H. Subramanian, User Control of Personal mHealth Data Using a Mobile Blockchain App: Design Science Perspective, JMIR Mhealth Uhealth, vol. 10, no. 1, 2022, https://doi.org/10.2196/32104.
[54] A. Farouk, A. Alahmadi, S. Ghose, A. Mashatan, Blockchain platform for industrial healthcare: Vision and future opportunities, *Comput. Commun.* 154 (2020) 223–235, https://doi.org/10.1016/j.comcom.2020.02.058.
[55] H. Subramanian and S. Subramanian, Improving Diagnosis through Digital Pathology: Proof-of-Concept Implementation Using Smart Contracts and Decentralized File Storage, J Med Internet Res, vol. 24, no. 3, 2022, https://doi.org/10.2196/34207.
[56] K. Miyachi and T. K. Mackey, hOCBS: A privacy-preserving blockchain framework for healthcare data leveraging an on-chain and off-chain system design, Inf. Process Manag., vol. 58, no. 3, 2021, https://doi.org/10.1016/j.ipm.2021.102535.
[57] Z. Nie, Y. Long, S. Zhang, and Y. Lu, A controllable privacy data transmission mechanism for Internet of things system based on blockchain, Int. J. Distrib. Sens. Netw., vol. 18, no. 3, 2022, https://doi.org/10.1177/15501329221088450.
[58] F. Rezaeibagha, K.T. Win, W. Susilo, A Systematic Literature Review on Security and Privacy of Electronic Health Record Systems: Technical Perspectives, *Health Inform. Manage. J.* 44 (3) (Oct. 2015) 23–38, https://doi.org/10.1177/183335831504400304.
[59] ISO/IEC 27001 Standard – Information Security Management Systems, (2022). https://www.iso.org/standard/27001 (accessed March 9, 2024).
[60] ISO/IEC 27002 Standard – Information security, cybersecurity and privacy protection — Information security controls, (2022). https://www.iso.org/standard/75652.html (accessed March 9, 2024).
[61] O. Olukoya, Assessing frameworks for eliciting privacy & security requirements from laws and regulations, Comput. Secur. 117 (2022), https://doi.org/10.1016/j.cose.2022.102697.
[62] C. Ilioudis, G. Pangalos, A Framework for an Institutional High Level Security Policy for the Processing of Medical Data and their Transmission through the Internet, J. Med. Internet Res. 3 (2) (2001) e14.
[63] P.N. Otto and A.I. Anton, Addressing Legal Requirements in Requirements Engineering, in: 15th IEEE International Requirements Engineering Conference (RE 2007), 2007, pp. 5–14. https://doi.org/10.1109/RE.2007.65.
[64] T.D. Breaux, D.L. Baumer, J. Doyle, E.H. Spafford, and M.A. Vouk, Legal Requirements Acquisition for the Specification of Legally Compliant Information Systems, (2009). https://repository.lib.ncsu.edu/handle/1840.16/3376 (accessed March 9, 2024).
[65] F.A. Al-Zahrani, Evaluating the Usable-Security of Healthcare Software Through Unified Technique of Fuzzy Logic, ANP and TOPSIS, *IEEE Access* 8 (2020) 109905–109916, https://doi.org/10.1109/ACCESS.2020.3001996.
[66] P. Llorens-Vernet and J. Miró, Standards for mobile health-related apps: Systematic review and development of a guide, JMIR Mhealth Uhealth, vol. 8, no. 3, 2020, https://doi.org/10.2196/13057.
[67] C. Pardo, F. Pino, F. García, F. R. Romero, M. Piattini, and M. T. Baldassarre, HProcessTOOL: A Support Tool in the Harmonization of Multiple Reference Models, in: Computational Science and Its Applications - ICCSA 2011, B. Murgante, O. Gervasi, A. Iglesias, D. Taniar, and B. O. Apduhan, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 370–382.
[68] C. Pardo, F. Pino, F. Garcia, M. Piattini, M. Baldassarre, A Process for Driving the Harmonization of Models, ACM Int. Conf. Proceeding Ser (Jun. 2010), https://doi.org/10.1145/1961258.1961271.
[69] IEEE/ISO/IEC 29148 Standard – Systems and software engineering – Life cycle processes – Requirements engineering, (2018). https://standards.ieee.org/ieee/29148/6937/ (accessed March 9, 2024).
[70] A. Toval, J. Nicolás Ros, B. Moros Valle, and F. Garcia, Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach, Requir. Eng., vol. 6, pp. 205–219, Jan. 2002, https://doi.org/10.1007/PL00010360.
[71] P. Mongeon, A. Paul-Hus, The journal coverage of Web of Science and Scopus: a comparative analysis, Scientometrics 106 (1) (2016) 213–228, https://doi.org/10.1007/s11192-015-1765-5.
[72] M.M. Pellegrini, F. Ciampi, G. Marzi, B. Orlando, The relationship between knowledge management and leadership: mapping the field and providing future research avenues, J. Knowl. Manag. 24 (6) (2020) 1445–1492, https://doi.org/10.1108/JKM-01-2020-0034.
[73] J.A. García-Berná, et al., Green IT and sustainable technology development: Bibliometric overview, Sustain. Dev. 27 (4) (Jul. 2019) 613–636, https://doi.org/10.1002/SD.1927.
[74] J. Yuen, Comparison of Impact Factor, Eigenfactor Metrics, and SCImago Journal Rank Indicator and h-index for Neurosurgical and Spinal Surgical Journals, World Neurosurg. 119 (2018) e328–e337, https://doi.org/10.1016/j.wneu.2018.07.144.
[75] P.W. Stone, Popping the (PICO) question in research and evidence-based practice, Appl. Nurs. Res. 15 (3) (2002) 197–198, https://doi.org/10.1053/apnr.2002.34181.
[76] N.J. van Eck, L. Waltman, Software survey: VOSviewer, a computer program for bibliometric mapping, Scientometrics 84 (2) (2010) 523–538, https://doi.org/10.1007/s11192-009-0146-3.

[77] D. Moher et al., Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement, PLoS Med, vol. 6, no. 7, 2009, https://doi.org/10.1371/JOURNAL.PMED.1000097.
[78] M.J. Page, et al., The PRISMA 2020 statement: an updated guideline for reporting systematic reviews, BMJ 372 (Mar. 2021) n71, https://doi.org/10.1136/bmj.n71.
[79] W. G. Cochran, Sampling techniques, 3rd ed. in: Wiley series in probability and mathematical statistics. New York [etc: John Wiley & Sons, 1977.
[80] G. Kotonya, I. Sommerville, *Requirements Engineering: Processes and Techniques*, 1st ed., Wiley Publishing, 1998.
[81] IEEE 830 Standard – Recommended Practice for Software Requirements Specifications, (1998). https://standards.ieee.org/ieee/830/1222/ (accessed March 9, 2024).
[82] S. Ouhbi, J.L. Fernández-Alemán, J.R. Pozo, M. El Bajta, A. Toval, A. Idri, Compliance of Blood Donation Apps with Mobile OS Usability Guidelines, J. Med. Syst. 39 (6) (2015) 63, https://doi.org/10.1007/s10916-015-0243-1.
[83] J.A. García-Berná, et al., Energy efficiency in software: A case study on sustainability in personal health records, J. Clean. Prod. 282 (2021) 124262, https://doi.org/10.1016/j.jclepro.2020.124262.
[84] J.M. Carrillo de Gea, J. Nicolás Ros, J. Fernández-Alemán, A. Toval, Automated support for reuse-based requirements engineering in global software engineering, J. Softw. Evol. Process 29 (2017) May, https://doi.org/10.1002/smr.1873.
[85] J. Nicolás, J. Lasheras, A. Toval, F.J. Ortiz, B. Álvarez, An integrated domain analysis approach for teleoperated systems, Requir. Eng. 14 (1) (2009) 27–46, https://doi.org/10.1007/s00766-008-0072-6.
[86] A. Toval, B. Moros Valle, J. Nicolás Ros, J. Lasheras, Eight key issues for an effective reuse-based requirements process, Comput. Syst. Sci. Eng. 23 (2008) 373–385.
[87] D.L. Hamilton, "Identification and evaluation of the security requirements in medical applications," in, Proceedings Fifth Annual IEEE Symposium on Computer-Based Medical Systems 1992 (1992) 129–137, https://doi.org/10.1109/CBMS.1992.244954.
[88] H.J. Baur, U. Engelmann, F. Saurbier, A. Schröter, U. Baur, H.P. Meinzer, How to deal with security issues in teleradiology, Comput. Methods Programs Biomed. 53 (1) (1997) 1–8, https://doi.org/10.1016/S0169-2607(96)01798-1.
[89] A. Strielkina, O. Illiashenko, M. Zhydenko, D. Uzun, Cybersecurity of healthcare IoT-based systems: Regulation and case-oriented assessment, in: 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), 2018, pp. 67–73, https://doi.org/10.1109/DESSERT.2018.8409101.
[90] ISO 27799 Standard — Health informatics — Information security management in health using ISO/IEC 27002, (2016). https://www.iso.org/standard/62777.html (accessed March 9, 2024).
[91] N. Archives and R. A. O. of the Federal Register, Public Law 104 - 191 - Health Insurance Portability and Accountability Act of 1996, (1996). https://www.govinfo.gov/app/details/PLAW-104publ191 (accessed March 9, 2024).
[92] eCFR :: 45 CFR Part 164 Subpart C – Security Standards for the Protection of Electronic Protected Health Information. Accessed: Sep. 27, 2023. [Online]. Available: https://www.ecfr.gov/current/title-45/subtitle-A/subchapter-C/part-164/subpart-C.
[93] J. T. Force, Security and Privacy Controls for Information Systems and Organizations, Jul. 2020, https://doi.org/10.6028/NIST.SP.800-53R5.
[94] Canada Health Infoway, Privacy and Security Requirements and Considerations for Digital Health Solutions, (2014). https://www.infoway-inforoute.ca/en/component/edocman/resources/technical-documents/architecture/2154-privacy-and-security-requirements-and-considerations-for-digital-health-solutions (accessed March 9, 2024).
[95] Unified Compliance Framework, Application Security and Development Security Technical Implementation Guide, (2024). https://www.stigviewer.com/stig/application_security_and_development/ (accessed March 9, 2024).
[96] OWASP Foundation, OWASP Top 10, (2021). https://owasp.org/Top10/en/ (accessed March 9, 2024).
[97] SANS Institute, Top 25 Software Errors, (2021). https://www.sans.org/top25-software-errors/ (accessed March 9, 2024).
[98] C. Pardo, F. Pino, F. Garcia, M. Piattini, and J. Rosado, Armonizando ISO/IEC 20000 e ISO/IEC 27001 para integrar la gestión de servicios y la seguridad de la información. 2010.
[99] C. Pardo, F. Pino, F. Garcia, and M. Piattini, Homogenization of Models to Support Multi-model Processes in Improvement Environments., vol. 1. 2009.
[100] F.J. Pino, M.T. Baldassarre, M. Piattini, G. Visaggio, Harmonizing maturity levels from CMMI-DEV and ISO/IEC 15504, J. Softw. Maint. Evol. Res. Pract. 22 (4) (Jun. 2010) 279–296, https://doi.org/10.1002/spip.437.
[101] OWASP Foundation, OWASP Top Ten, (2024). https://owasp.org/www-project-top-ten/ (accessed March 9, 2024).
[102] AICPA & CIMA, T. A. I. of CPAs, Information for service organization management in a SOC 1® engagement, (2023). https://www.aicpa-cima.com/resources/download/information-for-service-organization-management-in-a-soc-1-engagement (accessed March 9, 2024).
[103] AICPA & CIMA, T. A. I. of CPAs, SOC 2® - SOC for Service Organizations: Trust Services Criteria, (2023). https://www.aicpa-cima.com/topic/audit-assurance/audit-and-assurance-greater-than-soc-2 (accessed March 9, 2024).
[104] AICPA & CIMA, T. A. I. of CPAs, SOC 3® - SOC for Service Organizations: Trust Services Criteria for General Use Report, (2023). https://www.aicpa-cima.com/topic/audit-assurance/audit-and-assurance-greater-than-soc-3 (accessed March 9, 2024).
[105] M.A. Martínez, J. Lasheras, E. Fernández-Medina, A. Toval, M. Piattini, A Personal Data Audit Method through Requirements Engineering, Comput. Stand. Interfaces 32 (4) (2010) 166–178, https://doi.org/10.1016/j.csi.2010.01.001.
[106] M.A. Aguilar, A. Toval, M. Campos, Requirements Engineering to Audit Privacy Issues in, Medical and Health Software, 2008.
[107] B. Cruz Zapata, J.L. Fernández-Alemán, A. Toval, A. Idri, Reusable Software Usability Specifications for mHealth Applications, J. Med. Syst. 42 (3) (2018) 45, https://doi.org/10.1007/s10916-018-0902-0.
[108] P. Krishnan, C. Cifuentes, L. Li, T.F. Bissyande, J. Klein, Why Is Static Application Security Testing Hard to Learn? IEEE Secur. Priv. 21 (5) (2023) 68–72, https://doi.org/10.1109/MSEC.2023.3287206.
[109] C. Cifuentes, F. Gauthier, B. Hassanshahi, P. Krishnan, D. McCall, The role of program analysis in security vulnerability detection: Then and now, Comput. Secur. 135 (2023), https://doi.org/10.1016/j.cose.2023.103463.
[110] F. M. Tudela, J. R. B. Higuera, J. B. Higuera, J. A. S. Montalvo, and M. I. Argyros, On Combining Static, Dynamic and Interactive Analysis Security Testing Tools to Improve OWASP Top Ten Security Vulnerability Detection in Web Applications, Appl. Sci.-Basel, vol. 10, no. 24, 2020, https://doi.org/10.3390/app10249119.
[111] V. Casola, A. De Benedictis, C. Mazzocca, V. Orbinato, Secure software development and testing: A model-based methodology, Comput. Secur. 137 (2024) 103639, https://doi.org/10.1016/j.cose.2023.103639.
[112] CyberRes, Static Code Analyzer - Static Code Analysis Security, (2024). https://www.microfocus.com/en-us/cyberres/application-security/static-code-analyzer (accessed March 9, 2024).
[113] Acunetix, Vulnerability Scanner - Web Application Security, (2024). https://www.acunetix.com/vulnerability-scanner/ (accessed March 9, 2024).
[114] C. M. Mejía-Granda, Desarrollo de servicios web REST 'inseguros' para auto-aprendizaje en la explotación de vulnerabilidades, (2018). https://reunir.unir.net/handle/123456789/7435 (accessed March 28, 2023).
[115] OpenEMR, (2020). https://www.open-emr.org/ (accessed June 6, 2020).
[116] Y. Wang, P. Tran, and J. Wojtusiak, From Wearable Device to OpenEMR: 5G Edge Centered Telemedicine and Decision Support System, in: International Conference on Health Informatics, (2022). https://api.semanticscholar.org/CorpusID:247114293 (accessed March 28, 2023).
[117] Y. He, E. Zamani, I. Yevseyeva, and C. Luo, Artificial Intelligence–Based Ethical Hacking for Health Information Systems: Simulation Study, J. Med. Internet Res., vol. 25, p. e41748, 2023, https://doi.org/10.2196/41748.
[118] M.M. Moncy, M. Pilli, M. Somasundaram, S. Purkayastha, and C.R. Fulton, Evaluation of accessibility of open-source EHRs for visually impaired users, in: AMIA Annual Symposium Proceedings, vol. 2023, pp. 1165–1174, (2024). https://www.ncbi.nlm.nih.gov/pubmed/38222344 (accessed March 9, 2024).
[119] OpenEMR, OpenEMR Downloads - OpenEMR Project Wiki, (2024). https://www.open-emr.org/wiki/index.php/OpenEMR_Downloads (accessed March 9, 2024).
[120] M. Siavvas, D. Kehagias, D. Tzovaras, E. Gelenbe, A hierarchical model for quantifying software security based on static analysis alerts and software metrics, Softw. Qual. J. 29 (2) (2021) 431–507, https://doi.org/10.1007/s11219-021-09555-0.
[121] J. Dougherty, R. Kohavi, and M. Sahami, Supervised and Unsupervised Discretization of Continuous Features, in: International Conference on Machine Learning, (1995). https://api.semanticscholar.org/CorpusID:2527609 (accessed March 9, 2024).
[122] J.M. Carrillo De Gea, J. Nicolás, J.L. Fernández Alemán, A. Toval, C. Ebert, A. Vizcaíno, Requirements engineering tools: Capabilities, survey and assessment, Inf. Softw. Technol. 54 (10) (2012) 1142–1157, https://doi.org/10.1016/j.infsof.2012.04.005.
[123] J. Ning, X. Huang, W. Susilo, K. Liang, X. Liu, Y. Zhang, Dual Access Control for Cloud-Based Data Storage and Sharing, IEEE Trans. Dependable Secure Comput. 19 (2) (2022) 1036–1048, https://doi.org/10.1109/TDSC.2020.3011525.
[124] C.M. Mejía-Granda, J.L. Fernández-Alemán, J.M. Carrillo-de-Gea, J.A. García-Berná, Security vulnerabilities in healthcare: an analysis of medical devices and software, Med. Biol. Eng. Comput. 62 (1) (2024) 257–273, https://doi.org/10.1007/s11517-023-02912-0.
[125] D. Chauhan, C. Singh, D. Kudande, Y.-C. Hu, Cyber Security for IoT-Enabled Industry 4.0, IGI Global (2022) 89–124, https://doi.org/10.4018/978-1-6684-6444-1.ch006.
[126] M.Y.P.M. Yusof, C.H. Teo, C.J. Ng, Electronic informed consent criteria for research ethics review: a scoping review, BMC Med. Ethics 23 (1) (2022) 117, https://doi.org/10.1186/s12910-022-00849-x.
[127] S. Mikuletič, S. Vrhovec, B. Skela-Savič, and B. Žvanut, Security and privacy oriented information security culture (ISC): Explaining unauthorized access to healthcare data by nursing employees, Comput. Secur., vol. 136, p. 103489, 2024, https://doi.org/10.1016/j.cose.2023.103489.
[128] D.G. Arce, Cybersecurity and platform competition in the cloud, Computers & Security, 93, 101774, (2020). https://doi.org/10.1016/j.cose.2020.101774.
[129] A. A. Süzen, UNI-CAPTCHA: A Novel Robust and Dynamic User-Non-Interaction CAPTCHA Model Based on Hybrid biLSTM+Softmax, Journal of Information Security and Applications, vol. 63, p. 103036, 2021, https://doi.org/10.1016/j.jisa.2021.103036.
[130] M. Guerar, A. Merlo, M. Migliardi, F. Palmieri, Invisible CAPPCHA: A usable mechanism to distinguish between malware and humans on the mobile IoT, Comput. Secur. 78 (2018) 255–266, https://doi.org/10.1016/j.cose.2018.06.007.
[131] M. Azeem, D. Khan, S. Iftikhar, S. Bawazeer, and M. Alzahrani, Analyzing and comparing the effectiveness of malware detection: A study of machine learning approaches, Heliyon, vol. 10, no. 1, p. e23574, 2024, https://doi.org/10.1016/j.heliyon.2023.e23574.
[132] K. Habib, W. Leister, Threats identification for the smart Internet of Things in eHealth and adaptive security countermeasures, in: 2015 7th International

Conference on New Technologies, Mobility and Security (NTMS), 2015, pp. 1–5, https://doi.org/10.1109/NTMS.2015.7266525.
[133] J. Loonam, J. Zwiegelaar, V. Kumar, C. Booth, Cyber-Resiliency for Digital Enterprises: A Strategic Leadership Perspective, IEEE Trans. Eng. Manag. 69 (6) (2022) 3757–3770, https://doi.org/10.1109/TEM.2020.2996175.
[134] R. Sobrino-Duque, J. M. Carrillo-de-Gea, J. J. López-Jiménez, J. Nicolás Ros, and J. L. Fernández-Alemán, Usevalia: Managing Inspection-Based Usability Audits, Int. J. Hum. Comput. Interact., vol. 40, no. 3, pp. 719–743, Feb. 2024, https://doi.org/10.1080/10447318.2022.2121879.
[135] A. Ampatzoglou, S. Charalampidou, I. Stamelos, Research state of the art on GoF design patterns: A mapping study, J. Syst. Softw. 86 (7) (2013) 1945–1964, https://doi.org/10.1016/j.jss.2013.03.063.
[136] F. Elberzhager, J. Münch, V.T.N. Nha, A systematic mapping study on the combination of static and dynamic quality assurance techniques, Inf. Softw. Technol. 54 (1) (2012) 1–15, https://doi.org/10.1016/j.infsof.2011.06.003.
[137] M. Hatamian, Engineering Privacy in Smartphone Apps: A Technical Guideline Catalog for App Developers, IEEE Access. 8 (2020) 35429–35445, https://doi.org/10.1109/ACCESS.2020.2974911.
[138] World Health Organization, Guiding optimal development and use of digital health towards improved health outcomes, (2024). https://www.who.int/westernpacific/activities/guiding-optimal-development-and-use-of-digital-health-towards-improved-health-outcomes (accessed March 9, 2024).
[139] U.S. Centers for Disease Control and Prevention, Health Insurance Portability and Accountability Act of 1996 (HIPAA) | Public Health Law | CDC, (1996). https://www.cdc.gov/phlp/php/resources/health-insurance-portability-and-accountability-act-of-1996-hipaa.html (accessed March 9, 2024).
[140] J. A. Marron, HIPAA Security Rule | NIST, 2022, 10.6028/NIST.SP.800-66R2.IPD.
[141] I. T. G. P. TEAM, EU General Data Protection Regulation (GDPR) – An implementation and compliance guide, fourth edition. IT Governance Publishing, 2020. 10.2307/j.ctv17f12pc.
[142] AICPA & CIMA, T. A. I. of CPAs, Learn about SOC for Cybersecurity – Resources, (2018). https://www.aicpa-cima.com/resources/download/learn-about-soc-for-cybersecurity (accessed March 9, 2024).
[143] Common Weakness Enumeration, CWE Top 25 Most Dangerous Software Weaknesses, (2022). https://cwe.mitre.org/top25/archive/2022/2022_cwe_top25.html (accessed March 9, 2024).

Carlos M. Mejía-Granda is a Ph.D. candidate in Informatics Engineering at the Faculty of Computer Science, University of Murcia, Murcia, ES-30100, Spain. Contact him at carlosmichael.mejiag@um.es

José L. Fernández Alemán is an associate professor with the Software Engineering Research Group, Faculty of Computer Science, University of Murcia, Murcia, ES-30100, Spain. Contact him at aleman@um.es

Juan M. Carrillo de Gea is an associate professor with the Software Engineering Research Group, Faculty of Computer Science, University of Murcia, Murcia, ES-30100, Spain. Contact him at jmcdg1@um.es.

José A. García-Berná is a researcher with the Software Engineering Research Group, Faculty of Computer Science, University of Murcia, Murcia, ES-30100, Spain. Contact him at josealberto.garcia1@um.es