NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY
"POLITEHNICA" BUCHAREST

DOCTORAL SCHOOL OF AUTOMATIC CONTROL AND COMPUTERS
COMPUTER SCIENCE DEPARTMENT

Ph.D. Summary Thesis

Security of e-Health Systems

Author:
Cristian Contasel

Thesis supervisor:
Prof. Dr. Ing. Răzvan Rughiniș

BUCHAREST

2024

Table of Contents

Abstract

Nowadays, the utilization of e-Health software systems is becoming increasingly prevalent in our daily lives. Whether we are patients, physicians, or collaborators with a medical institution, we utilize these systems. These systems enhance medical safety, minimize human error, and reduce costs by automating a variety of flows and procedures. In order to safeguard against security threats, e-Health systems necessitate robust cybersecurity due to their rapid adoption. These sorts of threats have the potential to compromise patient care, disrupt healthcare, and compromise data privacy.

The current thesis investigates software-based solutions for improving the cybersecurity resilience of e-Health systems. The thesis investigates and suggests innovative solutions and methodologies for effectively mitigating the security threats that impact e-Health cloud services.

The thesis is organized into eight chapters, including an introductory chapter, a state-of-the-art chapter, five contributions chapters, and a conclusion chapter.

The initial chapter of the contributions focuses on the enhancement of architecture to mitigate security threats. This is accomplished by proposing a new architectural model that is based on microservices, which enables the isolation and mitigation of threats without compromising the systems functionality.

The second chapter of the contributions addresses the enhancement of security for e-Health systems in terms of authentication mechanisms by leveraging the capabilities of smartphones. It introduces an improved authentication mechanism that is based on OAuth 2.0.

The third chapter of contributions is dedicated to the improvement of the security of communication between portable medical devices and e-Health ecosystem services. In order to achieve this, a new module called the Personal Medical Hub is proposed.

The fourth chapter of the contributions is dedicated to the enhancement and investigation of communication between cloud services and personal/portable medical devices. The objective of this investigation is to optimize energy consumption and guarantee the efficient exchange of data between medical devices and smartphones.

The final contribution chapter introduces a novel security framework that employs the device attestation API to protect cloud services, with the goal of reducing the risks associated with: Action Tampering, Flooding Attacks, Injection and Payload Attacks, and JSON hijacking.

Software solutions are the foundation of all chapters, which are intended to verify the proposed methodologies and frameworks. The thesis serves as a foundation for future research, particularly in the field of securing e-Health ecosystems through the deployment of cloud features and smartphones.

Keywords: e-Health systems, e-Health security, OAuth 2.0, Attestation API, cross-platform frameworks, e-Health microservices, REST services, SafetyNet, Xamarin, patient monitoring, authentication, cloud computing.

1. Introduction

1.1 Motivation

e-Health systems have become omnipresent, a part of our daily lives, regardless of whether we are ordinary citizens or medical professionals. Taking into consideration the fact that we rely on them not only to monitor the state of health, but also to treat diseases or keep under control chronic uncurable illnesses, we need to also empower them with the ability to face challenges related to security and to be able to guarantee the availability of their services.

Given the continuous advancement of technology, the increasing computing power, the growing precision and sophistication of robots, and the ongoing development of artificial intelligence (AI) it is expected that e-Health systems will progressively improve and their services will become accessible for the general public.

However, it is important to acknowledge that the development of e-Health systems capable of independently treating patients without the need for medical professionals is still an unreachable scenario. The medical system, being inherently conservative, and the technology, not yet advanced enough, do not currently support such a scenario. Although the robots and professional devices used in clinics are moving towards this goal, there is an alarming rise in vulnerabilities in the security of the software systems they rely on. These vulnerabilities are primarily driven by the excessive consumption of hardware resources and the high cost of the necessary hardware.

This thesis examines different security measures for e-Health systems, focusing on the main services and their integrated components like portable devices (e.g., insulin pumps). It aims to address security challenges faced by medical professionals and consumers by providing practical solutions tested in real-world scenarios.

1.2 Objectives

This thesis aims to investigate solutions for enhancing the security of e-Health systems. It focuses on addressing security issues identified at the core service level, communication level, and end consumer level. The proposed solutions for improving the security of e-Health systems have undergone testing and validation in real-world scenarios.

The primary questions that the thesis seeks to address are:

- How can we take advantage of smartphones to enhance the cybersecurity of e-Health systems?
- How can we improve cybersecurity and resilience of e-Health systems by adopting a microservices-based architecture?
- How can we securely integrate portable medical devices into an e-Health system without compromising system cybersecurity?

1.3 Thesis Structure

The present thesis consists of 8 chapters as follows: an introductory chapter, a state of the art chapter, 5 chapters of contributions, and a conclusion chapter that outlines the contributions of this thesis and presents future work.

The initial chapter, known as the introduction, seeks to explain and describe the motivation behind this thesis. Additionally, it outlines the primary research questions that the thesis seeks to address.

The second chapter discusses the state of the art in e-health systems security. The security solutions for the e-health system are divided into the components that are being targeted. Therefore, the chapter addresses the following topics: the security and availability of the e-Health core system, the security of medical devices, the security of medical REST client applications, and the global security of the e-Health ecosystem.

The third chapter corresponds to the guarantee of a high level of availability of e-Health services in order to enable the fundamental elements of the system to cope with traffic fluctuations. The principal architectural model for the core of the e-Health system is identified in the first subchapter, which also presents the impact of COVID-19 disease on the network traffic of the e-Health ecosystem. The proposed framework is defined in the second subchapter for preventing cybersecurity incidents by using the transition to microservices. Additionally, a deployment framework for e-Health systems is specified.

The fourth chapter pertains to the improvement of security for e-Health systems (in terms of authnetification mecanisms) by utilizing the features of smartphones. Initially, a brief introduction is provided to introduce the threats that impact cloud services for e-Health. This was followed by a classification and discussion of the existing authentication mechanism in terms of its security level and user-friendliness. Given this, an assessment of the security vulnerabilities and threats associated with OAuth 2.0 was conducted. The primary objective of this chapter is to introduce a novel authentication framework that is built on OAuth 2.0. This framework utilizes user biometric and geolocation data to fully satisfy the security requirements of e-Health systems.

The fifth chapter of contributions concentrates on the enhancement of the security of communication between portable medical devices and e-Health ecosystem services. In order to accomplish this, an examination of the security issues that impact mobile medical devices is conducted, and a novel module known as the Personal Medical Hub is presented. The Personal Medical Hub is designed to protect medical devices from security threats by utilizing a variety of models and flows to eliminate vulnerabilities.

Chapter 6 is dedicated to the investigation and improvement of communication between personal/portable medical devices and cloud services. The purpose of this investigation is to optimize energy consumption and ensure the efficient exchange of data by analyzing the communication technologies between medical devices and smartphones. Concurrently, an assessment is made of the influence of cross-platform application development frameworks and methodologies on communication performance. The aim of the analysis is to identify a solution that allows the software product to be executed on multiple mobile operating systems with a single codebase, which decreases security risks while simultaneously maintaining the performance of the e-Health ecosystem's components and flows.

Chapter 7 introduces a novel security framework that utilizes the device attestation API to safeguard cloud services. Initially, a concise overview of the vulnerabilities in cloud e-Health systems is provided, followed by an explanation of the operation of device attestation services. The chapter's enhancements apply to session management, with the objective of mitigating various security threats.

Chapter 8, which is the final chapter, provides an overview of the thesis. It presents the primary conclusions and findings. All of the original contributions, as well as the future research work in terms of e-Health ecosystems, are presented in this chapter. Additionally, the publications list is presented here.

2. State of the art

The importance of ensuring the security and accessibility of e-Health services has become crucial in our current society. Obtaining a robust level of security and ensuring the accessibility of e-Health services can be a complex task, conditioned by the operational context and the features offered by the systems. In the attempt to make the functionalities reachable to a large number of users, the security level and system performance may be rapidly degraded by exposing the services in a public, unsupervised environment. Furthermore, the existence of traffic fluctuations can lead to errors in the functionalities, causing certain features to become unavailable and others features to operate incorrectly. Researchers have proposed multiple approaches over time to address challenges and minimize the occurrence of errors in e-Health systems.

The initial focus of this study will be on a set of articles that discuss the security of cloud software services designed for use in public environments. This includes the transition from a traditional operating environment to a cloud environment. Following this, there will be a review based on a selection of articles that aim to secure the devices integrated into e-Health systems. This section will cover both device security and the communication between the device and the e-Health core services. Lastly, the final section provides an analysis of the security enhancements for client-type applications. These applications, which can be mobile, web, or desktop applications, have the ability to utilize cloud services.

2.1 Core system security and availability

The issue of ensuring the security and availability of web core applications, whether they use a monolithic or decoupled architecture, is extensively discussed in numerous articles of the specialized technical literature.

The core system of an e-Health type system refers to the collection of software modules that provide users and client applications with the endpoints and functionalities of the e-Health system.

To initiate a discussion on web core application security, the following hypotheses are defined upfront:

- The web application is harmless and is hosted on a reliable and secure infrastructure. The infrastructure consists of the operating system, web server, and interpreter.
- The attacker has the abilities to manipulate the content or sequence of web requests submitted to the web application, but does not have the ability to directly compromise the infrastructure or the application code.

The primary vulnerabilities and methods to attack the web applications, as identified in the literature and presented in [1] consist of the following:

- Input validity;
- SQL injection;
- Cross-Site Scripting;
- State integrity;

- Logic correctness;

To mitigate these attacks, many different countermeasures have been developed over time. The following section will present the countermeasures specifically designed for each of these categories.

In order to analyze the approaches discussed in literature regarding these attacks, we will distinguish between two main priorities that need to be addressed: the input validity property and the logic correctness property.

2.1.1 Input validity

To analyze security measures against vulnerabilities related to "input validity" as defined in the technical literature, the same methodology classifying system will be used. Consequently, the literature provides the following strategies for addressing vulnerabilities through the application of the "security by constructions" concept.

William Robertson and Giovanni Vigna [2] suggest a web application framework that is built on a robust typing system. This framework is designed to prevent XSS and SQL injection by statically enforcing a separation between the structure and content of web documents and database queries generated by a web application. The framework utilizes type-specific sanitization routines that precisely recognize and sanitize various types of user input.

Stephen Thomas et al. [3] suggest a novel approach that involves applying a prepared statement replacement algorithm to mitigate vulnerabilities caused by SQL injections. During the evaluations, the proposed solution successfully eliminated 94% of the vulnerabilities that were studied. The primary benefit of this solution is that it does not need to be integrated into the runtime environment and only needs to be executed once.

In addition, the literature presents various methods for addressing vulnerabilities through the application of "security by validation" principles.

Table 1 summarizes each methodology examined in the thesis.

Table 1

Summary of techniques for input validity

Method	Security by construction	Security by verification	Security by protection
Robertson method	yes	no	no
Stephen method	yes	no	no
SQL DOM	yes	no	no
CANDID	yes	no	no
WebSSARI	no	yes	yes
FlowSpec	no	yes	no
Livshits method	no	yes	no

Nguyen-tuong method	no	yes	no
Saner	no	yes	no
Noncespaces	no	no	yes
ScriptGard	no	no	yes
AMNESIA	no	no	yes
Kruegel method	no	no	yes

2.1.2 Logic correctness

The correctness of logic is dependent on the application due to the fact that the vulnerabilities generated by application logic are closely tied to the application itself, as each application is distinct. In order to mitigate this category of vulnerabilities, the security policies can be explicitly defined during the application development process or implemented at a later time through the use of inferred policies.

To address vulnerabilities through the principle of "security by construction," the literature offers several suggestions.

The technical literature provides a variety of solutions with varying performance impacts for implementing the "security by protection" concept.

Table 2 summarizes each methodology examined in the thesis.

Table 2

Summary of techniques for input validity

Method	Security by construction	Security by verification	Security by protection
SIF	yes	no	no
SELinks	yes	no	no
UrFlow	yes	no	no
MiMoSA	no	yes	no
RoleCast	no	yes	no
NoTamper	no	yes	no
BLOCK	no	no	yes
CLAMP	no	no	yes

2.2 Medical devices security

According to the World Health Organization (WHO), a medical device is a device, apparatus, or embedded system that is utilized for monitoring, treating, and diagnosing illnesses of patients [23].

Medical device security includes a range of tools and policies that are specifically designed to thwart unauthorized access by attackers, with the ultimate goal of preventing them from gaining control over the devices or compromising the data they produce.

The security features of medical devices are categorized based on their functionality and characteristics [24], which include software-based, hardware-based, and software-hardware-based features.

The primary focus of the subsequent section will be on the vulnerabilities and countermeasures that are directed at the software component of medical devices. The primary forms of attacks that specifically target medical devices, as defined by Maria Papaioannou et al [25], include:

- Eavesdropping attacks
- Spoofing attacks
- Traffic analysis attacks
- Masquerading attacks
- Physical attacks
- Malware attacks
- Man-in-the-middle attacks
- Denial-of-service attacks
- Battery drainage attacks
- Impersonation attacks
- Message fabrication/modification/replay attacks

In order to mitigate these vulnerabilities, the solution presented in the literature will be presented in the subsequent summaries.

In their study, Minchul Kim et al. [34] present a method that uses encryption on portable devices as an approach to safeguard against eavesdropping. The solution relies on an XOR operation that uses a key generated by the timer embedded in the microcontroller of the device. The encryption is performed using hardware-based XOR gates. To prevent an unauthorized usage of the key, they suggest implementing a regeneration mechanism linked to a device hardware event.

In their study, Yan et al. [36] present the PHY-IDS tool, which is designed to identify spoofing attacks specifically aimed at wearable devices. PHY-IDS utilize statistical learning techniques to analyze and identify unusual signal behavior and data. The solution relies on the power level of a received frame, which is measured at the receiver antenna.

Ibbad Hafeez et al. [38] suggest that IoT-KEEPER can be used to detect and prevent traffic attacks. To accomplish this, IoT-KEEPER conducts traffic analysis. It employs fuzzy C-means clustering and fuzzy interpolation to identify malicious network traffic. In order to safeguard against this vulnerability, IOT-KEEPER implements network access restrictions depending on the identification of malicious traffic. IoT-KEEPER employs a traffic classification scheme that does not require side-channel information, like device-type data, to detect malicious activity on devices. Instead, it uses unlabeled network traffic metadata for feature extraction.

Mohan Sai et al. [41] developed a machine learning-based lightweight denial-of-service attacks detection technique. Their approach involves employing a support vector machine to differentiate between attack traffic and normal traffic A classification algorithm that is based on a light data model is employed to identify the anomalies. A correlation-based feature selection algorithm is implemented to achieve the data model's lightness. However, the solution is only capable of detecting denial-of-service attacks, but it is unable to provide protection for devices against them.

Shanshan Tu et al. [42] propose a novel Q-learning algorithm for the precise detection of impersonation attacks. The algorithm is viable in both static and dynamic environments. The objective of the static environment solution is to implement a zero-sum game between the attacker and the receiver by implementing a patch for physical layer security. The solution for a dynamic environment considers the channel state information, including the average time, false alarm rate, miss detection rate (MDR), and average error rate (AER).

Jorge Maestre Vidal et al. [44] propose a novel masquerader detection strategy that is capable of detecting masquerade attacks even when the attacker imitates the behavior of legitimate users. To achieve this, the strategy suggests three stages: the analysis, verification, and mimicry recognition stages. In the analysis stage, the alignment algorithms identify discrepancies from typical user behavior to identify a masquerade attack. The discrepancies detected are subjected to a validation scheme that is based on the U-test during the verification stage. The strategy involves comparing historical data with current user behavior to identify inconsistencies that indicate an attack in order to perform masquerade recognition.

Grant A. Jacoby et al [46] have introduced the concept of utilizing the battery status of a device as an indicator of potential physical tampering or intrusion in order to identify and mitigate battery drainage attacks. He recommends that the battery status be monitored using sensors or dedicated circuitry that can detect changes, such as sudden drops in voltage or current, when the battery is tampered with, in order to detect the attack. The system initiates lockdown procedures and triggers an alarm when an anomaly in the battery status is detected. The primary benefit is that it does not necessitate substantial additional resources or device modifications, as it capitalizes on the battery and its monitoring circuits.

Ensieh Modiri Dovom et al [47] suggest a method for malware mitigation that utilizes a Fuzzy Pattern Tree (FPT). An FPT is a hierarchical structure that employs fuzzy logic to represent patterns in data. The FPT is employed to identify and classify malware on the device in order to detect it. The device is where the detection process takes place. The method extracts and clusters certain fuzzy features based on the behavior of various devices and software states (network traffic patterns, resource usage, method calls). The system is capable of identifying patterns of normal and malicious behavior through the use of FTP, thereby enabling the early mitigation of potential threats.

2.3 Medical REST client applications security

Medical REST client applications represent the entire range of medical web and mobile applications that use RESTful APIs for accessing or alter data and resources on a remote medical system.

In order to discuss the security of REST applications, it is necessary to separately address the security of web applications and mobile applications. This is due to the fact that the security threats and solutions are distinct due to the particular design and ecosystem of the applications.

2.3.1 Security of medical REST web application client

The term "client side" in web development denotes all aspects of a web application that are displayed or are executed on the client device. This includes the visible components of the user interface, such as text, images, and other elements that are contained by the UI, as well as any operation that an application performs inside the user's browser. The main security challenges.

The primary security challenges arise from the actions performed by certain applications within the browser. These actions are frequently executed with the support of JavaScript, either directly or through frameworks that are based on this technology. The specialized literature provides the subsequent examples of securing JavaScript applications that run on the client side.

The main security measures proposed by literature refer to the use of the concept of isolating the execution of JavaScript code in safe areas and the analysis of these interactions using various tools to determine a potentially harmful or incorrect execution of the code.

2.3.2 Security of medical REST mobile application client

Mobile attacks have been classified into four categories: application-based attacks, web-based attacks, network-based attacks, and physical-based attacks, as documented in the studies [54]. The integrity and confidentiality of mobile data and applications are impacted by these attacks.

The large part of literature offer solution for securing medical REST mobile application client based on defined policies end try to enforce them via different solutions. Another method of ensuring security is to monitor the device's activities in order to identify potential anomalies or malicious applications. This is done to prevent interaction with the medical application and to isolate it in a safe environment.

2.4 Security approaches for e-Health ecosystems

The subsequent scientific literature review is focused on various types of e-Health ecosystems and investigates different approaches of securing them. This section concludes the state-of-the-art chapter by illustrating potential future research areas and developments in the field of security and availability for e-Health systems.

In the solutions proposed by literature, a lot of the approaches that are capable of securing e-Health ecosystems are built on an analysis conducted using NLP or AI techniques to identify vulnerabilities, which cannot be mitigated directly. The confidentiality and integrity of the data are the primary objectives of the other techniques, which include the methods that are based on blockchain or cryptography, without considering the system's remaining vulnerabilities.

3. Integration of microservices in medical frameworks

An innovative architectural model for e-Health systems is presented in this chapter, with the objective of enhancing cyber resilience and ensuring high availability in the face of fluctuating traffic loads. It investigates the correlations between typical cybersecurity incidents in the field of e-Health and architectural defects, as well as the frequent design patterns in currently operational systems. It presents a testing approach that is based on the research, identifies the weaknesses, and suggests viable solutions. This chapter introduces a comprehensive support strategy for the transition from traditional monolithic architectures to microservices. This modification leverages the vertical and horizontal scalability of cloud computing to optimize resource utilization and guarantee system reliability. Additionally, it provides deployment strategies for the new microservices, with a focus on cybersecurity and operational resilience in e-Health environments.

3.1 Introduction

One of the global impacts of the coronavirus was in the healthcare industry. COVID-19 caused overcrowding in the hospitals, which makes it impossible for patients and doctors to meet in person for a consultation. e-Health systems, in addition to the roles for which they were developed, also took on a new role as a mediator in order to respond to this situation. Additionally, new features were either developed or used more frequently in order to meet this demand:

- Contact tracing;
- Telehealth (online consultation with a doctor);
- Automated diagnosis;
- Forecasting of material resource requirements;
- Individual medical record about the COVID-19 illness.

The most common COVID-19-related attacks on e-Health systems are ZOOM bombing, COVID-19 phishing attacks, malware, and network availability [71].

This chapter addresses network availability issues in e-Health systems. To accomplish this, the first step is to extract a common architectural model based on a medical unit study. This study identifies the most commonly used e-Health software systems, as well as the key features required. The study included 45 hospitals and medical clinics in Bucharest, in the public and private sectors.

This chapter addresses network availability issues in e-Health systems. To accomplish this, the first step is to extract a common architectural model based on a medical unit study. This study identifies the most commonly used e-Health software systems, as well as the key features required. The study included 45 hospitals and medical clinics in Bucharest, in the public and private sectors. Based on the architecture of e-Health software systems, 17 out of 45 entities use only local software systems, while 28 use web applications. The most common architecture followed the model-view-controller monolith pattern.

3.1.1 COVID-19 and network traffic

To determine the traffic variation for e-Health systems, a traffic analysis was conducted using CO APCD public data from April 2018 to March 2024. The obtained results are shown in Figure 3 and Figure 4.

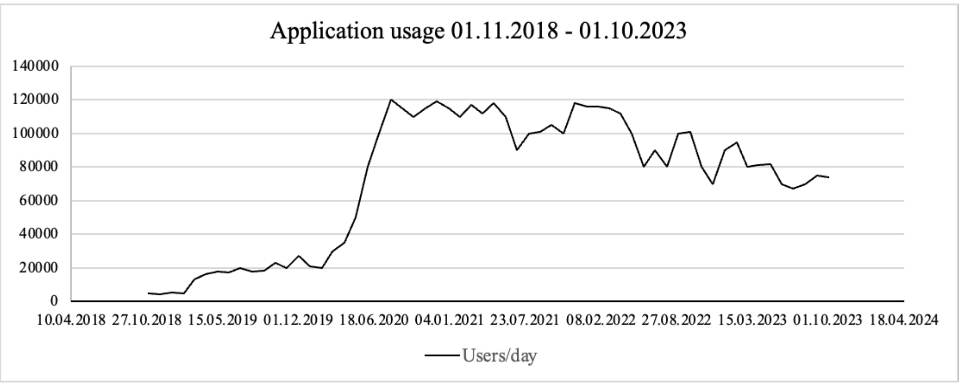

Fig. 3. Application usage between 01.11.2018 to 01.10.2023 based on number of user/day

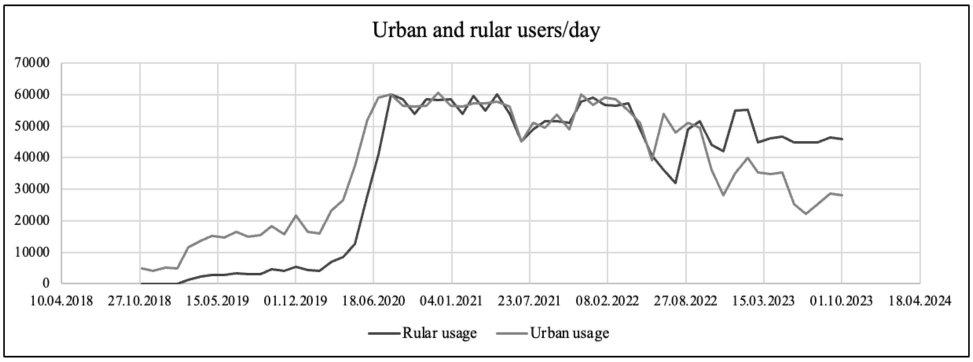

Fig. 4. Application usage between 01.11.2018 to 01.10.2023 splitted area type.

The current challenge in medical software systems is how to mitigate cybersecurity threats and risks, and also how to handle the performance issues caused by the significantly increased workload as a consequence of their rapid adoption.

3.1.2 e-Health architecture pattern

To create a common overview of the analyzed e-Health software systems, the research identified the core and optional modules that can be combined to create a system that includes all of the previously described features.

The system is comprised of seven interconnected modules that are safeguarded from the external environment by a firewall. The main components of the architecture are:

- the web application;
- the AI result interpreter;
- the laboratory system;
- streaming server;
- the SQL server ;
- the external systems API that manage the portable devices hub and the SMS function.

3.2 Proposed framework

In order to effectively manage load differences and maintain a lower infrastructure cost, the primary characteristic of the new architecture is its vertical and horizontal scalability.

The proposed system architecture is based on microservices, which allow the original monolithic application to be split into multiple independent services capable of performing work independently. This independence enables the booth scaling system to be implemented. The e-Health ecosystem will be able to initiate new workers for each service in accordance with the system load. Additionally, the e-Health ecosystem will be able to enhance the computational power of current workers.

The splitting of the microservices has been done based on the functionality of the system in order to be able to provide the features to users independently of each other, so that if a set of microservices no longer works properly, the system can manage the rest of the features independently.

The transition step involves the addition of cloud services one by one, and the new API gateway module ensures seamless connectivity with these services. In order to optimize performance, a caching system was implemented.

The microservices architecture that results is composed of 16 services, which are categorized into Layer 1 and Layer 2 levels.

Layer 1 services are mapped to various system features in order to provide system functionality. They have a caching and optimization of the request mechanism in place, and they are also capable of storing data in the SQL module.

The Layer 2 services are the ones that provide support for the level 1 services and are capable of integrating with various subsystems that are not scalable, such as external providers or outdated applications (e.g., laboratory systems).

3.2.1 Microservices e-Health systems deployment

The main options for deploying microservices in Microsoft Azure are Cloud Services and Azure Web Apps. To determine which solution was used, the effective cost for 24 hours was used.

The estimated cost of Azure Cloud Service deployment is $231.82, while Azure App Service deployment is estimated at $151.2. In conclusion, Azure Cloud Service is the most cost-effective option for deployment. However, the decision to deploy the microservices into one of these solutions should also consider the computational power required to handle the same number of requests.

3.3 Metrics for performance evaluation

To identify the main issues in the system, the following types of performance tests were carried out: stress testing, endurance testing, and spike testing.

In e-Health software, the main concerns about the number of users come from the patient role. Because of this issue, the designed scenarios should be based on the main action that patients can perform.

Apache JMeter was used to perform automated testing. In order to be able to perform a large number of threads, multiple JMeter instances were needed to run in parallel. The instances of JMeter was hosted on Azure by using Azure Cloud Service.

The upper limit for JMeter determined during the test was 1000 threads per instance; after this limit, JMeter's performance was degraded.

3.4 Benchmark for performance

In order to see the performance difference from the original architecture to microservices architecture, the same set of tests was performed. The test was conducted on the Azure Cloud Service and also on Azure App Service deployment scenarios.

3.4.1 Original architecture performance

The system performance was severely damaged after 170.000 users. To guarantee the e-Health system's availability and prevent system failure, a series of actions can be implemented at a reduced cost in accordance with the results of the testing scenario.

Given that the main concerns about the number of users are generated by the patient role, it is possible to restrict the access for that category of users based on a queuing system, in order to limit the active users to a maximum number of 150.000 or lower, depending on how many doctors should be accommodated within the system.

3.4.2 Microservices e-Health architecture performance

As the original architecture was able to handle the tests for 170,000 users, now the test scenarios begin at 200.000 and are run in Azure Cloud to provide adequate power.

According to the stress tests, the system's performance has been substantially improved. However, the addition of new machines to the system, which is caused by horizontal scaling, results in some failed requests.

The proposed architecture has an improved rate of 98.7%, according to the tests performed with JMeter and presented in thesis.

3.5 Conclusions

In order to mitigate the risk that impacts network availability and guarantee the high availability of e-Health software systems, this chapter suggests the implementation of cloud computing and microservices as a solution. The main goal of the transition from monolithic architecture to microservices architecture in e-Health software systems is to establish a new layer between legacy software and the new expectations and behaviors of users. Additionally, it intends to provide support and scaling to accommodate a high volume of concurrent requests.

To make that performance decoupling possible, this chapter proposes a transition to microservices that allow vertical and horizontal scaling. In order to minimize hardware overhead, the proposal is to utilize cloud solutions to host all components of the e-Health system.

Because microservices are separate, updates and bug fixes can be applied to individual services without crashing the system. This lets security patches and new features be released quickly and securely, keeping the system secure. In addition, microservices architecture separates services, so a security breach in one doesn't affect others. Containment reduces data breaches.

Each microservice can have customized security measures. This enables more precise security tailored to each service's data or transactions.

3.5.1 Contributions

- A novel microservice e-Health system was developed to effectively manage fluctuations in traffic and reduce costs by incorporating both vertical and horizontal scaling methods.
- An entire cloud infrastructure was setup using the Azure cloud solution in order to evaluate the proposed architectural solution.
- A testing methodology was proposed to evaluate the performance of various e-Health systems in relation to network availability and to determine the limitations of the systems.
- A series of measures were developed and evaluated to guarantee the availability of e-Health software systems in the event that they are unable to handle variations in network traffic.
- A migration strategy was created to ensure the availability of e-Health software systems at a reduced cost. This strategy employs both vertical and horizontal scaling mechanisms to specifically address the points of system failure.
- A set of deployment strategies for microservice e-Health systems based on Azure cloud solutions was created, taking into account the cost of the services.
- A general software model was developed by analyzing the available products on the market and focusing on the current system's capabilities and constraints.
- A classification model for the adoption of e-Health systems was proposed and validated using actual data collected during the COVID-19 pandemic.

4. Increasing the security of framework by using smartphones

In the medical field, web applications security systems often use the authentication strategy and credentials to assess the identity of the user. Based on the credentials, the system is able to claim the identity of the user. Also, the authenticity of identity is claimed based on authentication strategy. This chapter analyzes and compares different strategies used to enforce the web applications security systems by using location and biometric features of the smartphones to provide two-way authentications.

4.1 Introduction

Authentication is a problem that is often revized due to the continuous increase in computing power. As a consequence, password requirements increase year by year, and the complexity of passwords involves significant effort for the end user, yet insecure passwords still exist.

This chapter compares different authentication strategies used by web applications to enforce the authentication process by using smartphone features as location and biometric sensors.

4.2 Existing authentication mechanisms categories

This section presents an overview of the most frequent authentication methods for web applications by using the classification scheme proposed by Renaud et al. [85] and completed by Yampolski in [86].

The classification scheme shows that there are four categories of authentication systems based on the location of the user or based on what the user knows.

4.2.1 Password based authentication

The password authentication methods can be categorized into two main categories [88] as follows:

- Text based passwords;
- Picture passwords;

The distinction between text-based passwords and picture passwords is presented in Table 9.

Table 9

Comparison of password categories

Criteria	**Password authentication**	
	Text based passwords	**Picture passwords**
Security	Low	High
Availability	Always	Always
Usability	Easy	Easy
Cost	Reduced	High

4.2.2 Location based authentication

Location-based authentication systems utilize geolocation data to validate the user's identity. There are two distinct categories of authentication methods: one that requires the specification of all geolocation data during the setup, such as latitude, longitude, and position accuracy; the other category is based on user behavior and relies on artificial intelligence (AI) techniques.

The comparison between static and dynamic location authentication is outlined in Table 10.

Table 10

Comparison of location authentication system

Criteria	**Location authentication**	
	Static location	**Dynamic location**
Security	High	Low
Availability	Always	Sometimes
Usability	Easy	Easy
Cost	Reduced	High

4.2.3 Biometric based authentication

The biometric authentication compares the current physical characteristics with the stored samples in order to find a match between them. If it is a match, the user is authenticated.

The main differences of the biometric characteristics are presented in Table 11. The comparison is based on the research conducted by Parvathi Ambalakat [93].

Table 11

Comparison of biometrics categories

Criteria	**Biometric characteristic**			
	Fingerprint	**Face**	**Iris**	**Retina**
Performance	High	Low	High	High
Acceptance	Medium	High	Low	Low

Criteria	**Biometric characteristic**			
	Fingerprint	**Face**	**Iris**	**Retina**
Circumvention	High	High	Low	Low
Collectabilty	Medium	High	Medium	Low
Distinctivity	High	Low	High	High

4.3 Vulnerability of OAuth 2.0

OAuth 2.0 is centered around bearer tokens. As a result, the integration of this mechanism is straightforward; however, bearer tokens lack any internal security mechanisms. Nowadays, this is the standard protocol for industry.

4.3.1 Bearer token vulnerabilities

One of the first set of vulnerabilities is caused by the fact that users who use this type of authorization do not have to prove their identity.

The second set of vulnerabilities consists of the following token vulnerability threats:

- Token redirection;
- Token reusing;
- Token forgery;
- Token decryption.

The second set of vulnerabilities can be mitigated or eliminated by implementing a minimal set of measures regarding token generation and transport.

4.4 Proposed security improvement for OAuth 2.0

A novel enhanced authentication system is proposed to enhance the security of the OAuth 2.0 protocol for cloud applications. This system utilizes biometric data and device location to enhance the cybersesilience of the applications.

The proposed architecture is based on a previous solution that employs Android, an operating system that is already safeguarded by SafetyNet services.

The authorization component differs depending on the source of the request as follow:

- The initial flow corresponds to requests that are directly initiated from an Android client application;
- The second flow corresponds to requests that are initiated from other devices or web applications.

4.4.1 Android biometric usability

In the Android ecosystem, the security of biometric data is a top priority. To prevent data leakage and compromise, smartphones implement a Trusted Execution Environment (TEE) secure area. In order to utilize the TEE, a collection of software components known as Trusty needs to be used.

4.4.2 Android location usability

Android provides two methods for requesting the device's location: Utilizing the Google Play Services or utilizing the LocationListener

In order to establish a more secure location representation, the acquired latitude and longitude will be encrypted with an bijective function that takes into acknowledging the session GUID (f(latitude, longitude, guid) -> locationGUID).

The result of this function will be decrypted inside the client application to retrieve the specific location of the user (latitude and longitude). The final result will be compared with the registration step location input, so as to correspond with a tolerance defined to the entire system.

4.5 Usage statistics

The tests were performed with the following parameters: 1000 threads, 500 seconds ramp-up time, 5 loops, 50 seconds maximum active time and 5000 seconds wait time.

The results show that the response time of the modified authentication protocol is almost doubled. This emphasizes the difference between the overhead of communication of the Android device and that of the authentication component. For the initial request, the time is longer due to the fact that the device must obtain the current location, which results in an increased overhead. Consequently, the response time is almost four times slower.

4.6 Conclusions

In this chapter, the primary features that can be employed in an authentication system were examined, as well as the manner in which these features are combined to create a more secure protocol that is based on OAuth 2.0.

Even if the authentication mechanism is more secure now, the system can still be affected by attacks. For example, the smartphone can be patched with Magisk, in order to patch the SELinux.

4.6.1 Contributions

- A novel and enhanced OAuth 2.0 authentication model was developed. This model includes a stronger security mechanism that relies on the use of biometric data and the location of smartphones to verify the identity of the user.
- A categorization model of the current authentication mechanisms was proposed to present and categorize different authentication models.

- An upgraded registration process was developed, which included the concept of trusted geographical zones.
- A performance and security analysis model was developed and used to properly evaluate performance changes and security enhancements for various authentication models.

5. Increasing the security of communication for personal/portable medical devices

This chapter introduces Personal Medical Hub, an efficient security solution designed specifically for Bluetooth enabled medical devices. The main objective of this solution is to improve data privacy and provide strong protection against various security threats such as man-in-the-middle attacks, security breaches, and backdoors.

5.1 Introduction

Mobile medical devices can collect data on a patient's health status and provide treatment procedures, which makes them useful for both prevention and treatment purposes.

Because of the different purpose of these devices, some of them can only collect data from patients, data that needs to be downloaded at a doctor's office or needs to be sent in real time to the e-Health systems.

Bluetooth is the most frequently used communication technology for real-time connected medical devices to exchange information. Bluetooth is often used for transmitting patient data to a mobile phone for local analysis or for transferring data to remote medical systems.

A modified communication architecture based on a Personal Medical Hub is introduced in this chapter in order to improve security of Bluetooth medical devices.

The Personal Medical Hub is a powerful security solution that improves data privacy and provides protection against security threats such as man-in-the-middle attacks, security breaches, and backdoors.

5.2 General system architecture

The presented system is made up of four subsystems: the medical device, the patient 's smartphone, the cloud system, and the e-Health system.

According to the security level classification for medical devices developed by Johannes Sametinger et al. [104], portable medical devices are classified as having a medium to very high risk.

One of the most common security breaches is associated with man-in-the-middle attacks. Attacks that involve a man in the middle constitute nearly 35% of all exploitation activity [105].

5.3 Proposed security improvement architecture

The Personal Device Hub represents a Bluetooth hub interposed between all the communication channels of the medical device. The hub is equipped with Bluetooth and Wi-Fi interfaces for communication. The main goal of the Personal Device Hub is to manage and mitigate denial of service and man-in-the-middle attacks.

To achieve this objective, the Personal Device Hub employs three distinct approaches:

- Mitigate the vulnerability by upgrading the device to the latest firmware provided by the device vendor.
- Analyze network traffic to detect unusual behavior and apply filters or block non-whitelisted sources.
- Encrypt traffic between device and cloud to prevent certificate spoofing.

5.3.1 Handle vulnerabilities and attacks

The key components of the Personal Device Hub are the traffic analyzer and the device security manager, which are derived from the primary two functions of the device (network traffic control and communication encryption for the cloud infrastructure; patching vulnerable devices that still have vendor support).

Depending on the status of a vulnerability, the system can choose from three scenarios:

- Filtering the attacks;
- Fetching, notifying and applying the patch;
- Dealing with an unsecure device.

5.4 Security evaluation

To evaluate the Personal Medical Hub's efficiency, a legacy insulin pump with Bluetooth connectivity was used.

The most problematic are those that necessitate filter rejection, and the rate of fixation is dependent on the quality of the filters. In the event of denial of service attacks, the traffic analyzer provides the most effective protection.

In future development, it is necessary to strengthen the Personal Medical Hub's ability to address authentication issues and XSRF (Cross-Site Request Forgery) vulnerabilities.

5.5 Contributions

- A novel solution was designed to improve communication between portable medical devices and e-Health cloud services. The solution, called Personal Medical Hub, enhances the security of communication between portable medical devices and e-Health cloud services and reduces the risks of man-in-the-middle attacks.
- A security strategy was developed to address various attacks that are based on vulnerabilities. The strategy has the ability to determine between three scenarios: filtering the attacks, addressing an unsecure device, and determining the vulnerability path of the device.
- A security model evaluation for medical device communication was developed, and it was used to assess how effectively Personal Medical Hubs worked with different medical devices.

6. Increase the performance of communication between personal/portable medical devices and cloud services

This chapter will examine the impact of performance on a variety of short-distance communication techniques that are supported by cross-platform frameworks, as well as the overarching impact of performance generated by cross-platform solutions, using the comparative test method.

6.1 Introduction

The primary communication technologies that allow mobile devices to exchange data can be divided into two categories: long-range and short-range technologies.

The main exponents of short-range technologies are Bluetooth, Wi-Fi, NFC, and infrared technology.

Long-range technologies are represented by GSM systems and satellite technologies such as GPS.

This chapter will examine how mobile device communications technologies perform over short distances and how cross-platform frameworks affect this performance.

6.2 Performance of communications technologies for short distances between mobile devices

The main communication technologies for short distances communication according to Toshiya Tamura et al. [111] are: Wi-Fi, Bluetooth and NFC.

6.2.1 Bluetooth communication

Bluetooth operates on the licensed ISM (Industrial, Scientific, Medical) band of 2.4 GHz. There are 79 communication channels, and each packet will be transmitted only once. Each channel has a 1MHz bandwidth.

The communication protocol has a master-slave structure and is package-based [113]. Each master can communicate with up to seven slave devices. All network devices use a single master clock. The role of master is determined by mutual agreement (a master may become a slave at some point). At some point in time, only the master and a slave can communicate, with the master determining which device to address.

6.2.2 Wi-Fi Direct communication

Wi-Fi Direct (previously called Peer to Peer Wi-Fi) is a standard that enables devices to connect without the need for an access point (AP). It enables communication at the speed of a Wi-Fi network.

The device that functions as an AP is decided upon through negotiations; consequently, both devices must execute the client and AP roles (which are logical roles).In the same manner as Wi-Fi Protected, this access point is protected by a PIN.

6.2.3 NFC communication

The NFC communication technology enables communication between two devices that are situated at a maximum distance of 10 cm. Every device has the ability to operate in three distinct modes: card emulation, read/write, and P2P (peer-to-peer) [116].

The standard permits transfer rates ranging from 106 Kbps to 424 Kbps and operates at 13.56 MHz (frequency that does not necessitate a license) [117]. It is founded on the principle of magnetic induction. There are two operational modes: active and passive.

6.2.4 Comparison of technologies

All characteristics of technologies for short range communication are presented in Table 15.

Table 15

Comparison of technologies for short-range communication

Characteristic	Bluetooth	Wi-Fi Direct	NFC
Maximum coverage area	50 m	45 m	20 cm
Frequency	2.4 GHz	2.4 GHz	13.56 MHz
Transfer rate	1 Mbps	54 Mbps	424 Kbps
Network type	WPAN	WPAN	P2P
Configuration	Needs adjustments	Needs adjustments	Nearby
Connection time	6s	6s	0.1s
Standard	IEEE 802.15.1	-	ISO 13157

6.2.5 Technological Validation of Technologies

In order to technically verify the technologies provided above, a test was conducted to exchange information in plain text and binary format.

Functional validation was conducted using an application that was specifically designed for this purpose and is compatible with the Android ecosystem. The tests were conducted independently in separate activities. All technologies necessitated the existence of an asynchronous task for reception and a distinct thread for transmission. All of the aforementioned technologies underwent successful functional validation.

6.3 Cross-platform application communication performance

The performance of hybrid cross-platform and cross-platform compiled applications will be the primary focus of the analysis, as mobile web applications are restricted to dynamic updating and their fundamental interaction with operating system services [120] is insufficient to satisfy the requirements of mobile medical applications [121].

6.3.1 Development using cross-platform compiling techniques

The primary exponent of cross-platform development solutions based on code compilation is Microsoft MAUI (Xamarin), which is corresponding to.NET-Android /.NET-iOS . By employing C# as the primary programming language, the solution facilitates the usage of native interferences and offers direct access to the native API. Each platform (apart from.NET MAUI, which offers a common UI possibility) needs to have its own project created specifically for it, and portable libraries (PCLs) are used to share code between them.

6.3.2 Development using web hybrid techniques

Scripts are the foundation of applications, which can be executed by the diverse routines of their web browsers [124]. A collection of APIs for manipulating down-level components (hardware features, resource management, etc.) that are visible from JavaScript is also available to the developer [125].

The hybrid applications exhibit a significant performance gap when contrasted with native applications, primarily as a result of the distinct nature of the resources.

6.3.3 Comparative power consumtion analysis

Table 16 illustrates the results of the evaluation for the Bluetooth transmission.

Table 16

Power consumption for Bluetooth-based communication

Application type	Consumption short distance (mW)	Consumption long distance (mW)
Native	524,8	536,8
Compiled cross	527,1	539,1
Hybrid	648,3	685,7

Table 17 illustrates the results of the evaluation for the NFC transmission.

Table 17

Power consumption for NFC-based communication

Application type	Consumption (mW)
Native	42
Compiled cross	51,2
Hybrid	67,9

6.3.4 Comparative analysis in terms of computational performance

The native application achieved the highest data transmission speed, followed by the cross-compiled application.This suggests that ACW and MCW have a lower overhead than JavaScript callbacks.

6.4 Cross-platform application user-perspective performance

To evaluate the impact of cross-platform frameworks on mobile medical applications performance, a real-time analysis was conducted. This analysis was conducted by utilizing the following four primary criteria:

- Execution time;
- Startup time;
- CPU usage;
- Memory usage.

It is noticeable that Flutter is more efficient than .Net in terms of rendering and navigation time. This discrepancy is produced by the Flutter render engine Skia, which is utilized for rendering graphics.

In relation to the startup time, the tests indicate that both cross-platform frameworks exhibit variability in performance, with more noticeable fluctuations reported for Flutter.

Related to CPU usage Flutter consumes higher CPU resources compared to .NET, resulting in increased battery usage and device heating. .NET applications shows reduced CPU usage and demonstrate superior efficiency in managing system resources.

In terms of memory usage, .NET consumes less memory and generates more stable memory usage compared to Flutter.

If native applications are not a viable alternative due to their increased financial cost and development time, compiled cross-platform applications are the preferred solution in the e-Health ecosystems for resource consumption efficiency. This is supported by the performance analysis, which is based on execution time, startup time, CPU usage, and memory usage.

6.5 Conclusions

Comparative tests have shown that the cross-platform compilation frameworks achieved the highest performance, with a 10.6% decrease in performance compared to native applications. In reverse, web-based techniques result in a 34.4% performance depreciation.

In terms of the short-range communication technologies that can be employed to exchange data between devices and based on the hardware availability of portable medical devices, Bluetooth is a viable solution, particularly due to the small data sets that must be shared and its adequate coverage area.

6.5.1 Contributions

- A comparison and evaluation of the various technologies for short distances were conducted to determine their technological limitations and communication performance.
- An evaluation and comparison of the various cross-platform frameworks (Flutter and .Net) were conducted to assess the performance overhead from the user's perspective.
- A comparative model analysis was proposed to compare the computational performance, power consumption, and ease of development of native, compiled cross-platform, and hybrid cross-platform applications.

- A comparative model analysis was proposed to determine the overhead of a cross-platform solution in the context of the performance of communication technologies for short distances.

7. Cloud based medical mobile application security enforcement using device attestation API

7.1 Introduction

This chapter examines the primary security issues and their resolution through the utilization of attestation services. These services confirm the authenticity of the client application as well as the application-running device.

7.2 Vulnerability of rest services for mobile usage

The built-in security features provided by the operating system reduce the frequency of security issues present in applications. Unfortunately, these features are incapable of safeguarding the system from attacks that originate at lower stack levels.

To reduce this kind of security issues, Android provides the feature of device attestation called SafetyNet [141]. SafetyNet is able to check the smartphone integrity, can provide information about device root status, bootloader unlock and application integrity. Implementing these types of validations reduces the possibility of an attack and prevents attackers from using dynamic analysis frameworks.

7.3 Attestation API for Android

Android SafetyNet [141] represents a security solution provided via Google Play Services that is able to protect the application for some extra security issues, like: fake users, device tampering, malware applications, URL tampering.

SafetyNet solution is composed of four components: Attestation API, Safe Browsing API, reCAPTCHA API and Verify Apps API. The roles of each component are described in Table 23 [141].

7.4 Using attestation API to secure cloud services

The session management component is an integral part of nearly all web services. The main properties that need to be considered for the session management component are encryption, high availability, and security. The encryption component is needed to prevent the man-in-the-middle attacks. To avoid the modification or theft of session cookies, the SSL/TLS needs to be used.

Even if on the server side the integrity and security checks are in our control, unfortunately, the client application runs in an uncontrolled environment. Starting from this premise, the goal is to ensure the integrity and security control of the client application. SafetyNet Attestation enables more secure and trusted connections between the server and the client side.

The main improvement that the attestation API can make is to remove the client's access to session management components if the communication was already disregarded by the attestation mechanism. Continuous validation is essential, not only when the session starts but also during each session refresh or expiration.

7.5 Security evaluation

Through the qualitative analysis of the protection mechanism, after 1 year of operation, the mechanism managed to protect the system from a number of 3871 attacks, at an active number of application users of 52.3 K.

By analyzing the threat models for cloud computing using the STRIDE (spoofing, tampering, repudiation, information disclosure, denial of service, elevation of privilege) model [145], the device attestation integration gives a more secure cloud environment.

Unfortunately, like all security mechanisms, there are a few methods to by-pass SafetyNet attestation API. For example Magisk, that represent a modern way to unlock the bootloader, and to patch SELinux policy.

7.6 Contributions

- A new and innovative concept of a secure session management system was developed. This system incorporates detailed information regarding the integrity of the client, including both the client application and the client device.
- A security analysis methodology was developed to identify security vulnerabilities in REST services for cloud applications, such as Flooding Attacks, Action Tampering, Injection and Payload attacks, and JSON hijacking.
- A security analysis methodology was proposed to assess the effectiveness of different tools in addressing various device integrity issues.
- A classification model was developed for security issues affecting mobile applications. The objective is to present the main security issues.

8. Conclusions

8.1 Contributions

This thesis provides validated solutions and prototype in order to ensure a high availability of e-Health services and to provide a secure way of integration for legacy software but also for new and legacy devices, in order to generate a full and secure experience for all users that use the system, regardless of their role. By implementing these solutions in production applications, this thesis contributes to the security of e-Health software in different contexts.

8.2 Detailed list

- **Migration of e-Health systems from monolithic to microservices architecture model**
  - A novel microservice e-Health system was developed to effectively manage fluctuations in traffic and reduce costs by incorporating both vertical and horizontal scaling methods.
  - An entire cloud infrastructure was setup using the Azure cloud solution in order to evaluate the proposed architectural solution.
  - A testing methodology was proposed to evaluate the performance of various e-Health systems in relation to network availability and to determine the limitations of the systems.
  - A series of measures were developed and evaluated to guarantee the availability of e-Health software systems in the event that they are unable to handle variations in network traffic.
  - A migration strategy was created to ensure the availability of e-Health software systems at a reduced cost. This strategy employs both vertical and horizontal scaling mechanisms to specifically address the points of system failure.
  - A set of deployment strategies for microservice e-Health systems based on Azure cloud solutions was created, taking into account the cost of the services.
  - A general software model was developed by analyzing the available products on the market and focusing on the current system's capabilities and constraints.
  - A classification model for the adoption of e-Health systems was proposed and validated using actual data collected during the COVID-19 pandemic.
- **e-Health cloud-based mobile application security enforcement using device attestation API**
  - A new and innovative concept of a secure session management system was developed. This system incorporates detailed information regarding the integrity of the client, including both the client application and the client device.
  - A security analysis methodology was developed to identify security vulnerabilities in REST services for cloud applications, such as Flooding Attacks, Action Tampering, Injection and Payload attacks, and JSON hijacking.

- A security analysis methodology was proposed to assess the effectiveness of different tools in addressing various device integrity issues.
- A classification model was developed for security issues affecting mobile applications. The objective is to present the main security issues.

- **Increasing the security of e-Health applications by using smartphone localization and biometric data**
  - A novel and enhanced OAuth 2.0 authentication model was developed. This model includes a stronger security mechanism that relies on the use of biometric data and the location of smartphones to verify the identity of the user.
  - A categorization model of the current authentication mechanisms was proposed to present and categorize different authentication models.
  - An upgraded registration process was developed, which included the concept of trusted geographical zones.
  - A performance and security analysis model was developed and used to properly evaluate performance changes and security enhancements for various authentication models.
- **Increasing communication security for Bluetooth Medical Devices in e-Health systems**
  - A novel solution was designed to improve communication between portable medical devices and e-Health cloud services. The solution, called Personal Medical Hub, enhances the security of communication between portable medical devices and e-Health cloud services and reduces the risks of man-in-the-middle attacks.
  - A security strategy was developed to address various attacks that are based on vulnerabilities. The strategy has the ability to determine between three scenarios: filtering the attacks, addressing an unsecure device, and determining the vulnerability path of the device.
  - A security model evaluation for medical device communication was developed, and it was used to assess how effectively Personal Medical Hubs worked with different medical devices.

- **Impact of cross-platform development frameworks on the performance of mobile communications for short distances**
  - A comparison and evaluation of the various technologies for short distances were conducted to determine their technological limitations and communication performance.
  - An evaluation and comparison of the various cross-platform frameworks (Flutter and .Net) were conducted to assess the performance overhead from the user's perspective.
  - A comparative model analysis was proposed to compare the computational performance, power consumption, and ease of development of native, compiled cross-platform, and hybrid cross-platform applications.
  - A comparative model analysis was proposed to determine the overhead of a cross-platform solution in the context of the performance of communication technologies for short distances.

8.3 Future work

From the perspective of the solutions introduced in the present thesis, the primary areas for additional research and development in the security of e-Health ecosystems are as follows:

- The integration of smartwatches to enhance security by enhancing the perception of users' activities and concerns.
- Incorporating artificial intelligence and robotics to safeguard medical processes from human error.
- An additional area of research that aims to enhance the security of e-Health systems is the identification of vulnerabilities and the protection of artificial intelligence algorithms that are available for laboratory medicine.
- The secure and accurate mapping of large data sets for e-Health is another area of research.

8.4 List of original publications

Journals (accepted to publication)**:**

- **Cristian Contaşel**, Razvan Rughinis, Dumitru-Cristian Tranca and Dinu Țurcanu, "Enhancing e-Health cybersecurity and resilience: shifting from monolithic to microservices architecture", National University Of Science And Technology "POLITEHNICA" Bucharest Scientific Bulletin. Series C: Electrical Engineering and Computer Science. 2024.

Conferences:

- Robert-Mihai Ciurea and **Cristian Contaşel** "Impact of cross platform mobile frameworks on end user performance. Flutter vs .NET 6. " XGEN International Conference on Science Communications, Journal OPACJ, No. 3, 2024
- **Cristian Contaşel**, Dumitru-Cristian Tranca and Alexandru-Viorel Palacean, "Increasing the security of web applications by using smartphones." 2022 21st RoEduNet Conference: Networking in Education and Research (RoEduNet). IEEE, 2022
- **Cristian Contaşel**, Dumitru-Cristian Tranca, Alexandru-Viorel Palacean and Daniel Rosner, "Increasing communication security for Bluetooth Medical Devices in eHealth systems." *2022 21st RoEduNet Conference: Networking in Education and Research (RoEduNet). IEEE, 2022.*
- **Cristian Contaşel**, Dumitru Cristian Tranca and Alexandru-Viorel Palacean, "Cloud based mobile application security enforcement using device attestation API." *2021 20th RoEduNet Conference: Networking in Education and Research (RoEduNet). IEEE, 2021.*
- Alexandru-Viorel Palacean, Dumitru-Cristian Trancă, **Cristian Contaşel**, Răzvan Tătăroiu and Cristian Duţescu, "IoT Enabled Optimized Architectures for GPS Anti-

Theft Tracking Devices." 2021 20th RoEduNet Conference: Networking in Education and Research (RoEduNet). IEEE, 2021.

- Alina Irina Pîrvan, George Cristian Pătru, Dumitru Cristian Trancă, **Cristian Contaşel** and Daniel Rosner, "Infrastructure independent rail quality diagnosis and monitoring system." 2019 18th RoEduNet Conference: Networking in Education and Research (RoEduNet). IEEE, 2019.
- **Cristian Contaşel**, Razvan Rughinis, Daniel Rosner, and Dumitru-Cristian Tranca, "Impact of Cross-Platform Development Frameworks on the Performance of Mobile Communications for Short Distances." The International Scientific Conference eLearning and Software for Education. Vol. 3. " Carol I" National Defence University, 2018.

References

[1] Hoffman, Andrew. " Web application security. ". " O'Reilly Media, Inc.", 2024.

[2] Robertson, William K., and Giovanni Vigna. "Static Enforcement of Web Application Integrity Through Strong Typing." USENIX Security Symposium. Vol. 9. 2009.

[3] Thomas, Stephen, Laurie Williams, and Tao Xie. "On automated prepared statement generation to remove SQL injection vulnerabilities." Information and Software technology 51.3 (2009): 589-598.

[4] McClure, Russell A., and Ingolf H. Krüger. "SQL DOM: compile time checking of dynamic SQL statements." Proceedings of the 27th international conference on Software engineering. 2005.

[5] Bisht, Prithvi, Parthasarathy Madhusudan, and V. N. Venkatakrishnan. "CANDID: Dynamic candidate evaluations for automatic prevention of SQL injection attacks." ACM Transactions on Information and System Security (TISSEC) 13.2 (2010): 1-39.

[6] Huang, Yao-Wen, et al. "Securing web application code by static analysis and runtime protection." Proceedings of the 13th international conference on World Wide Web. 2004.

[7] Smits, Jeff, Guido Wachsmuth, and Eelco Visser. "Flowspec: A declarative specification language for intra-procedural flow-sensitive data-flow analysis." Journal of Computer Languages 57 (2020): 100924.

[8] Livshits, V. Benjamin, and Monica S. Lam. "Finding Security Vulnerabilities in Java Applications with Static Analysis." USENIX security symposium. Vol. 14. 2005.

[9] Nguyen-Tuong, Anh, et al. "Automatically hardening web applications using precise tainting." IFIP International Information Security Conference. Boston, MA: Springer US, 2005.

[10] Balzarotti, Davide, et al. "Saner: Composing static and dynamic analysis to validate sanitization in web applications." 2008 IEEE Symposium on Security and Privacy (sp 2008). IEEE, 2008.

[11] Van Gundy, Matthew, and Hao Chen. "Noncespaces: Using Randomization to Enforce Information Flow Tracking and Thwart Cross-Site Scripting Attacks." NDSS. 2009.

[12] Saxena, Prateek, David Molnar, and Benjamin Livshits. "SCRIPTGARD: automatic context-sensitive sanitization for large-scale legacy web applications." Proceedings of the 18th ACM conference on Computer and communications security. 2011.

[13] Halfond, William GJ, and Alessandro Orso. "AMNESIA: analysis and monitoring for neutralizing SQL-injection attacks." Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering. 2005.

[14] Kruegel, Christopher, Giovanni Vigna, and William Robertson. "A multi-model approach to the detection of web-based attacks." Computer Networks 48.5 (2005): 717-738.

[15] Chong, Stephen, Krishnaprasad Vikram, and Andrew C. Myers. "SIF: Enforcing Confidentiality and Integrity in Web Applications." USENIX Security Symposium. 2007.

[16] Corcoran, Brian J., Nikhil Swamy, and Michael Hicks. "Cross-tier, label-based security enforcement for web applications." Proceedings of the 2009 ACM SIGMOD International Conference on Management of data. 2009.

[17] Chlipala, Adam. "Static Checking of {Dynamically-Varying} Security Policies in {Database-Backed} Applications." 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 10). 2010.

[18] Li, Xiaowei, and Yuan Xue. "Block: a black-box approach for detection of state violation attacks towards web applications." Proceedings of the 27th Annual Computer Security Applications Conference. 2011.

[19] Parno, Bryan, et al. "CLAMP: Practical prevention of large-scale data leaks." 2009 30th IEEE Symposium on Security and Privacy. IEEE, 2009.

[20] Son, Sooel, Kathryn S. McKinley, and Vitaly Shmatikov. "Rolecast: finding missing security checks when you do not know what checks are." Proceedings of the 2011 ACM international conference on Object oriented programming systems languages and applications. 2011.

[21] Balzarotti, Davide, et al. "Multi-module vulnerability analysis of web-based applications." Proceedings of the 14th ACM conference on Computer and communications security. 2007.

[22] Bisht, Prithvi, et al. "Notamper: automatic blackbox detection of parameter tampering opportunities in web applications." Proceedings of the 17th ACM conference on Computer and communications security. 2010.

[23] World Health Organization. "Health technology assessment of medical devices." (2011).

[24] Almohri, Hussain, et al. "On threat modeling and mitigation of medical cyber-physical systems." 2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE). IEEE, 2017.

[25] Papaioannou, Maria, et al. "A survey on security threats and countermeasures in internet of medical things (IoMT)." Transactions on Emerging Telecommunications Technologies 33.6 (2022): e4049.

[26] Nguyen, Tri-Hai, and Myungsik Yoo. "A hybrid prevention method for eavesdropping attack by link spoofing in software-defined Internet of Things controllers." International Journal of Distributed Sensor Networks 13.11 (2017): 1550147717739157.

[27] Van Der Merwe, J. Rossouw, et al. "Classification of spoofing attack types." 2018 European Navigation Conference (ENC). IEEE, 2018.

[28] Basyoni, Lamiaa, et al. "Traffic analysis attacks on Tor: A survey." 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT). IEEE, 2020.

[29] Abbas, Sohail, et al. "Masquerading attacks detection in mobile ad hoc networks." IEEE Access 6 (2018): 55013-55025.

[30] Pandey, Abhishek Kumar, et al. "Trends in malware attacks: Identification and mitigation strategies." Critical Concepts, Standards, and Techniques in Cyber Forensics. IGI Global, 2020. 47-60.

[31] Mahjabin, Tasnuva, et al. "A survey of distributed denial-of-service attack, prevention, and mitigation techniques." International Journal of Distributed Sensor Networks 13.12 (2017): 1550147717741463.

[32] Franklin, Joshua M., et al. Security analysis of first responder mobile and wearable devices. US Department of Commerce, National Institute of Standards and Technology, 2020.

[33] Grassi, Paul A., Michael E. Garcia, and James L. Fenton. "Digital Identity Guidelines." NIST special publication 800 (2017): 63-3.

[34] Kim, Minchul, and Taeweon Suh. "Eavesdropping vulnerability and countermeasure in infrared communication for IoT devices." Sensors 21.24 (2021): 8207.

[35] Lei, Hongjiang, et al. "Safeguarding UAV IoT communication systems against randomly located eavesdroppers." IEEE Internet of Things Journal 7.2 (2019): 1230-1244.

[36] Yan, Wenqing, et al. "PHY-IDS: A physical-layer spoofing attack detection system for wearable devices." Proceedings of the 6th ACM Workshop on Wearable Systems and Applications. 2020.

[37] Shoukry, Yasser, et al. "Pycra: Physical challenge-response authentication for active sensors under spoofing attacks." Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. 2015.

[38] Hafeez, Ibbad, et al. "IoT-KEEPER: Detecting malicious IoT network activity using online traffic analysis at the edge." IEEE Transactions on Network and Service Management 17.1 (2020): 45-59.

[39] Ahmed, M. Meraj, et al. "Defense against on-chip trojans enabling traffic analysis attacks." 2020 Asian Hardware Oriented Security and Trust Symposium (AsianHOST). IEEE, 2020.

[40] Paudel, Ramesh, Timothy Muncy, and William Eberle. "Detecting dos attack in smart home iot devices using a graph-based approach." 2019 IEEE international conference on big data (big data). IEEE, 2019.

[41] Sai, Kuthada Mohan, et al. "Lightweight Intrusion Detection System In IoT Networks Using Raspberry pi 3b+." SysCom. 2021.

[42] Tu, Shanshan, et al. "Security in fog computing: A novel technique to tackle an impersonation attack." IEEE Access 6 (2018): 74993-75001.

[43] Lee, Seo Jin, et al. "IMPACT: Impersonation attack detection via edge computing using deep autoencoder and feature abstraction." IEEE Access 8 (2020): 65520-65529.

[44] Vidal, Jorge Maestre, Ana Lucila Sandoval Orozco, and Luis Javier García Villalba. "Online masquerade detection resistant to mimicry." Expert Systems with Applications 61 (2016): 162-180.

[45] Jo, Hyo Jin, et al. "Mauth-can: Masquerade-attack-proof authentication for in-vehicle networks." IEEE transactions on vehicular technology 69.2 (2019): 2204-2218.

[46] Jacoby, Grant A., Randy Marchany, and NathanielJ Davis. "Battery-based intrusion detection a first line of defense." Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004.. IEEE, 2004.

[47] Dovom, Ensieh Modiri, et al. "Fuzzy pattern tree for edge malware detection and categorization in IoT." Journal of Systems Architecture 97 (2019): 1-7.

[48] HaddadPajouh, Hamed, et al. "A deep recurrent neural network based approach for internet of things malware threat hunting." Future Generation Computer Systems 85 (2018): 88-96.

[49] Fielding, Roy Thomas. Architectural styles and the design of network-based software architectures. University of California, Irvine, 2000.

[50] Goltzsche, David, et al. "Trustjs: Trusted client-side execution of javascript." Proceedings of the 10th European Workshop on Systems Security. 2017.

[51] Van Acker, Steven, and Andrei Sabelfeld. "Javascript sandboxing: Isolating and restricting client-side javascript." Foundations of Security Analysis and Design VIII: FOSAD 2014/2015/2016 Tutorial Lectures 15 (2016): 32-86.

[52] De Groef, Willem. "Client-and Server-Side Security Technologies for JavaScript Web Applications." eng. PhD thesis. University of Leuven (2016).

[53] Stats, StatCounter Global. "Mobile operating system market share worldwide." Dostopno prek https://gs. statcounter. com/os-market-share/mobile/worldwide (2024).

[54] Nachenberg, Carey. "A window into mobile device security–Examining the security approaches employed in Apple’s iOS and Google’s Android." Symantec Security Response (2011).

[55] Bwalya, Michael, and Christopher Chembe. "A Security Framework for Mobile Application Systems: Case of Android Applications." Zambia ICT Journal 3.2 (2019): 31-43.

[56] Popa, Daniela, et al. "A security framework for mobile cloud applications." 2013 11th RoEduNet International Conference. IEEE, 2013.

[57] Lima, António, et al. "A security monitoring framework for mobile devices." Electronics 9.8 (2020): 1197.

[58] Nyambo, Devotha, Zaipuna Yonah, and Charles Tarimo. "Framework for developing secure converged web and mobile applications." International Journal of Computing and Digital Systems 9.2 (2020): 167-177.

[59] Hussain, Muzammil, et al. "A security framework for mHealth apps on Android platform." Computers & Security 75 (2018): 191-217.

[60] Phung, Phu H., et al. "Hybridguard: A principal-based permission and fine-grained policy enforcement framework for web-based mobile applications." 2017 IEEE Security and Privacy Workshops (SPW). IEEE, 2017.

[61] Krupp, Brian, Nigamanth Sridhar, and Wenbing Zhao. "SPE: security and privacy enhancement framework for mobile devices." IEEE Transactions on Dependable and Secure Computing 14.4 (2015): 433-446.

[62] Savola, Reijo M., and Markus Sihvonen. "Metrics driven security management framework for e-health digital ecosystem focusing on chronic diseases." Proceedings of the International Conference on Management of Emergent Digital EcoSystems. 2012.

[63] Kubendiran, Mohan, Satyapal Singh, and Arun Kumar Sangaiah. "Enhanced security framework for e-health systems using blockchain." Journal of Information Processing Systems 15.2 (2019): 239-250.

[64] Sfar, Arbia Riahi, et al. "Privacy preservation using game theory in e-health application." Journal of information security and applications 66 (2022): 103158.

[65] Ksibi, Sondes, Faouzi Jaidi, and Adel Bouhoula. "A Comprehensive Quantified Approach for Security Risk Management in e-Health Systems." ICETE (2). 2020.

[66] Modi, Kirit J., and Nirali Kapadia. "Securing healthcare information over cloud using hybrid approach." Progress in Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2017, Volume 2. Springer Singapore, 2019.

[67] Zhou, Jun, et al. "PSMPA: Patient self-controllable and multi-level privacy-preserving cooperative authentication in distributedm-healthcare cloud computing system." IEEE transactions on parallel and distributed systems 26.6 (2014): 1693-1703.

[68] Silvestri, Stefano, et al. "Cyber threat assessment and management for securing healthcare ecosystems using natural language processing." International Journal of Information Security 23.1 (2024): 31-50.

[69] Kalis, Brian, Matt Collier, and Richard Fu. "10 promising AI applications in health care." Harvard business review (2018): 2-5.

[70] Business Research Company, “Patient Access /Front-end RCM Solutions Global Market Report 2024,” 2024.

[71] Weil, Tim, and San Murugesan. "IT risk and resilience—Cybersecurity response to COVID-19." IT professional 22.3 (2020): 4-10.

[72] Zaragoza, Pascal, et al. "Leveraging the layered architecture for microservice recovery." 2022 IEEE 19th International Conference on Software Architecture (ICSA). IEEE, 2022.

[73] Blinowski, Grzegorz, Anna Ojdowska, and Adam Przybyłek. "Monolithic vs. microservice architecture: A performance and scalability evaluation." IEEE Access 10 (2022): 20357-20374.

[74] Chouhan, Utkarsh, Vaibhav Tiwari, and Hradesh Kumar. "Comparing Microservices and Monolithic Applications in a DevOps Context." 2023 3rd Asian Conference on Innovation in Technology (ASIANCON). IEEE, 2023.

[75] Dickstein, Michael J., Kate Ho, and Nathaniel Mark. "Market segmentation and competition in health insurance." Journal of Political Economy 132.1 (2024): 96-148.

[76] Mulvaney-Day, Norah, et al. "Trends in use of telehealth for behavioral health care during the COVID-19 pandemic: considerations for payers and employers." American Journal of Health Promotion 36.7 (2022): 1237-1241.

[77] Millnert, Victor, and Johan Eker. "HoloScale: Horizontal and vertical scaling of cloud resources." 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC). IEEE, 2020.

[78] Gupta, Bulbul, Pooja Mittal, and Tabish Mufti. "A review on amazon web service (aws), microsoft azure & google cloud platform (gcp) services." Proceedings of the 2nd International Conference on ICT for Digital, Smart, and Sustainable Development, ICIDSSD 2020, 27-28 February 2020, Jamia Hamdard, New Delhi, India. 2021.

[79] Breneman, James E., Chittaranjan Sahay, and Elmer E. Lewis. “Introduction to reliability engineering. “ John Wiley & Sons, 2022..

[80] Pargaonkar, Shravan. "A comprehensive review of performance testing methodologies and best practices: software quality engineering." International Journal of Science and Research (IJSR) 12.8 (2023): 2008-2014.

[81] Czuper, Michal. “Applying automated performance testing with Apache Jmeter”. MS thesis. 2022.

[82] “Internet Security Threat Report” , United Stated of America, Volume 24, February 2019

[83] Bonneau, Joseph, et al. "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes." 2012 IEEE symposium on security and privacy. IEEE, 2012.

[84] Mayer, Peter, et al. "Supporting Decision Makers in Choosing Suitable Authentication Schemes." HAISA. 2016.

[85] Renaud, Karen. "Quantifying the quality of web authentication mechanisms a usability perspective." Journal of Web Engineering (2004): 095-123.

[86] Yampolskiy, Roman V. "User authentication via behavior based passwords." 2007 IEEE Long Island Systems, Applications and Technology Conference. IEEE, 2007.

[87] Aravindhan, K., and R. R. Karthiga. "One time password: A survey." International Journal of Emerging Trends in Engineering and Development 1.3 (2013): 613-623..

[88] Joshi, Abhilash M., and Balachandra Muniyal. "Authentication Using Text and Graphical Password." 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2018.

[89] Bhand, Amol, et al. "Enhancement of password authentication system using graphical images." 2015 International Conference on Information Processing (ICIP). IEEE, 2015.

[90] Chen, Chien-Ming, Xiaojie Zhang, and Tsu-Yang Wu. "A secure condition-based location authentication protocol for mobile devices." 2016 Third International Conference on Computing Measurement Control and Sensor Network (CMCSN). IEEE, 2016.

[91] Zhang, Feng, Aron Kondoro, and Sead Muftic. "Location-based authentication and authorization using smart phones." 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications. IEEE, 2012.

[92] Bhattacharyya, Debnath, et al. "Biometric authentication: A review." International Journal of u-and e-Service, Science and Technology 2.3 (2009): 13-28.

[93] Ambalakat, Parvathi. "Security of biometric authentication systems." 21st Computer Science Seminar. Vol. 1. 2005.

[94] Singh, Jaimandeep, and Naveen Kumar Chaudhary. "OAuth 2.0: Architectural design augmentation for mitigation of common security vulnerabilities." Journal of Information Security and Applications 65 (2022): 103091.

[95] Sievierinov, Oleksii, and Oleh Kholosha. "Securing Bearer token in OAuth2. 0." COMPUTER AND INFORMATION SYSTEMS AND TECHNOLOGIES (2021).

[96] Contașel, Cristian, Dumitru-Cristian Trancă, and Alexandru-Viorel Pălăcean. "Cloud based mobile application security enforcement using device attestation API." 2021 20th RoEduNet Conference: Networking in Education and Research (RoEduNet). IEEE, 2021.

[97] Paquette, Allie, Frank Painter, and Jennifer Leigh Jackson. "Management and risk assessment of wireless medical devices in the hospital." Biomedical Instrumentation & Technology 45.3 (2011): 243-248.

[98] Omboni, Stefano, Luca Campolo, and Edoardo Panzeri. "Telehealth in chronic disease management and the role of the Internet-of-Medical-Things: the Tholomeus® experience." Expert Review of Medical Devices 17.7 (2020): 659-670.

[99] Meisner, Marta. "Financial consequences of cyber attacks leading to data breaches in healthcare sector." Copernican Journal of Finance & Accounting 6.3 (2017): 63-73.

[100] Williams, Patricia AH, and Andrew J. Woodward. "Cybersecurity vulnerabilities in medical devices: a complex environment and multifaceted problem." Medical Devices: Evidence and Research (2015): 305-316.

[101] Tervoort, Tom, et al. "Solutions for mitigating cybersecurity risks caused by legacy software in medical devices: a scoping review." IEEE access 8 (2020): 84352-84361.

[102] Hassija, Vikas, et al. "Security issues in implantable medical devices: Fact or fiction?." Sustainable Cities and Society 66 (2021): 102552.

[103] Liu, Long, et al. "Use-related risk analysis for medical devices based on improved FMEA." Work 41.Supplement 1 (2012): 5860-5865.

[104] Sametinger, Johannes, et al. "Security challenges for medical devices." Communications of the ACM 58.4 (2015): 74-82.

[105] “X-Force Threat Intelligence Index 2022 “, IBM Security, 2022

[106] European Parliament and Council of the European Union, “Regulation (EU) 2017/745 of the European Parliament and of the Council of 5 April 2017 on medical devices,” Official Journal of the European Union, vol. 60, no. April 2014, pp. 1–175, 2017

[107] Arora, Rachna, Anshu Parashar, and Cloud Computing Is Transforming. "Secure user data in cloud computing using encryption algorithms." International journal of engineering research and applications 3.4 (2013): 1922-1926.

[108] Joh, HyunChul, and Yashwant K. Malaiya. "Defining and assessing quantitative security risk measures using vulnerability lifecycle and cvss metrics." The 2011 international conference on security and management (sam). 2011.

[109] Dikkers, Frederik G., et al. "Live surgery broadcast: who is benefiting?." European Archives of Oto-Rhino-Laryngology 273 (2016): 1331-1333.

[110] Crescente, Mary Louise, and Doris Lee. "Critical issues of m-learning: design models, adoption processes, and future trends." Journal of the Chinese institute of industrial engineers 28.2 (2011): 111-123.

[111] Dalmasso, Isabelle, et al. "Survey, comparison and evaluation of cross platform mobile application development tools." 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC). IEEE, 2013.

[112] Tamura, Toshiya, and Isao Masuda. "Device connectivity technologies using short-distance wireless communications." Fujitsu Sci. Tech. J 49.2 (2013): 213-219.

[113] Bisdikian, Chatschik. "An overview of the Bluetooth wireless technology." IEEE Communications magazine 39.12 (2001): 86-94.

[114] Sobya, D. "Embedded multiple source real time monitoring and control by bluetooth support with master slave architecture and algorithms." (2018).

[115] Lee, Jae Hyeck, Myong-Soon Park, and Sayed Chhattan Shah. "Wi-Fi direct based mobile ad hoc network." 2017 2nd International Conference on Computer and Communication Systems (ICCCS). IEEE, 2017.

[116] Belghazi, Zakariae, et al. "Secure WiFi-direct using key exchange for IoT device-to-device communications in a smart environment." Future Internet 11.12 (2019): 251.

[117] Roland, Michael, Josef Langer, and Josef Scharinger. "Relay attacks on secure element-enabled mobile devices: virtual pickpocketing revisited." Information Security and

Privacy Research: 27th IFIP TC 11 Information Security and Privacy Conference, SEC 2012, Heraklion, Crete, Greece, June 4-6, 2012. Proceedings 27. Springer Berlin Heidelberg, 2012.

[118] Lathiya, Poonam, and Jing Wang. "Near-field communications (NFC) for wireless power transfer (WPT): An overview." Wireless Power Transfer–Recent Development, Applications and New Perspectives (2021): 95-122.

[119] Preethi, K., Anjali Sinha, and April Nandini. "Contactless communication through near field communication." International Journal of Advanced Research in Computer Science and Software Engineering 2.4 (2012): 158-163.

[120] Kostakos, Vassilis, and Eamonn O'Neill. "NFC on mobile phones: issues, lessons and future research." Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07). IEEE, 2007.

[121] Jobe, William. "Native Apps vs. Mobile Web Apps." International Journal of Interactive Mobile Technologies 7.4 (2013).

[122] Kulkarni, Prajakta, and Yusuf Öztürk. "Requirements and design spaces of mobile medical care." ACM SIGMOBILE Mobile Computing and Communications Review 11.3 (2007): 12-30.

[123] Mark Reynolds, "Xamarin Mobile Application Development for Android", Pakt Publishing, UK, 2014

[124] Dickson, Jared. "Xamarin mobile development." (2013).

[125] Andrew Lunny, "PhoneGap Beginner's Guide", Pakt Publishing, UK, 2011

[126] Shrivasi, Avinash, and Anandkumar Pardeshi. "Implementation of cross-platform mobile application using phone-gap framework." International Journal of Computer Science and Engineering (IJCSE) 3 (2014): 23-30.

[127] Wargo, John M. PhoneGap essentials: Building cross-platform mobile apps. Addison-Wesley, 2012.

[128] Gomez, Carles, Joaquim Oller, and Josep Paradells. "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology." sensors 12.9 (2012): 11734-11753.

[129] Moron, María José, et al. "Overhead and Segmentation Mismatch Effect on Bluetooth WPAN Performance." Wireless personal communications 50 (2009): 161-180.

[130] Andonoska, Anita, and Kire Jakimoski. "Performance Evaluation of Mobile Applications." Proceedings of the Third International Conference on Autonomic and Autonomous Systems (ICAS'07), Struga, Macedonia. 2018.

[131] Datta, Diya, and Kajanan Sangaralingam. "Do app launch times impact their subsequent commercial success?." International Journal of Big Data Intelligence 3.4 (2016): 279-287.

[132] Dorfer, Thomas, Lukas Demetz, and Stefan Huber. "Impact of mobile cross-platform development on CPU, memory and battery of mobile devices when using common mobile app features." Procedia Computer Science 175 (2020): 189-196.

[133] Gil, Celio, et al. "A conceptual exploration for the safe development of mobile devices software based on OWASP." Int. J. Appl. Eng. Res 13.18 (2018): 13603-13609.

[134] Mulligan, Gavin, and Denis Gračanin. "A comparison of SOAP and REST implementations of a service based interaction independence middleware framework." Proceedings of the 2009 Winter Simulation Conference (WSC). IEEE, 2009.

[135] Neumann, Andy, Nuno Laranjeiro, and Jorge Bernardino. "An analysis of public REST web service APIs." IEEE Transactions on Services Computing 14.4 (2018): 957-970.

[136] Díaz-Rojas, Josué Alejandro, et al. "Web api security vulnerabilities and mitigation mechanisms: A systematic mapping study." 2021 9th International Conference in Software Engineering Research and Innovation (CONISOFT). IEEE, 2021.

[137] Zhao, Fengyu, Xin Peng, and Wenyun Zhao. "Multi-tier security feature modeling for service-oriented application integration." 2009 Eighth IEEE/ACIS International Conference on Computer and Information Science. IEEE, 2009.

[138] Masood, Adnan, and Jim Java. "Static analysis for web service security-Tools & techniques for a secure development life cycle." 2015 IEEE International Symposium on Technologies for Homeland Security (HST). IEEE, 2015.

[139] Zhang, Yi. "User Identity Hiding Method of Android." Research Anthology on Securing Mobile Technologies and Applications. IGI Global, 2021. 413-425.

[140] Garg, Shivi, and Niyati Baliyan. "Android security assessment: A review, taxonomy and research gap study." Computers & Security 100 (2021): 102087.

[141] Mulliner, Collin. "Inside Android's SafetyNet Attestation: Attack and Defense." 34th Chaos Communication Congress, 2017

[142] Kim, Taehun, et al. "Breaking ad-hoc runtime integrity protection mechanisms in android financial apps." Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. 2017.

[143] Nguyen-Vu, Long, et al. "Android rooting: An arms race between evasion and detection." Security and Communication Networks 2017.1 (2017): 4121765.

[144] Furfaro, Angelo, et al. "Modelling and simulation of a defense strategy to face indirect DDoS flooding attacks." Internet and Distributed Computing Systems: 7th International Conference, IDCS 2014, Calabria, Italy, September 22-24, 2014. Proceedings 7. Springer International Publishing, 2014.

[145] Xin, Tong, and Ban Xiaofang. "Online banking security analysis based on STRIDE threat model." International Journal of Security and Its Applications 8.2 (2014): 271-282.

[146] Florea, Iulia Maria, Gabriel Ghinita, and Razvan Rughinis. "Sharing of network flow data across organizations using searchable encryption." 2021 23rd International Conference on Control Systems and Computer Science (CSCS). IEEE, 2021.

[147] Vochescu, Alexandru, Ioana Culic, and Alexandru Radovici. "Multi-layer security framework for IoT devices." 2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet). IEEE, 2020.

[148] Pepito, Joseph Andrew, et al. "Intelligent humanoid robots expressing artificial humanlike empathy in nursing situations." Nursing Philosophy 21.4 (2020): e12318.

[149] Liu, Tangyou, et al. "A Review on the Form and Complexity of Human–Robot Interaction in the Evolution of Autonomous Surgery." Advanced Intelligent Systems (2024): 2400197.

[150] Reed, J. Craig, and Nicolas Dunaway. "Cyberbiosecurity Implications for the Laboratory of the Future." Frontiers in bioengineering and biotechnology 7 (2019): 182.

[151] Adeghe, Ehizogie Paul, Chioma Anthonia Okolo, and Olumuyiwa Tolulope Ojeyinka. "The role of big data in healthcare: A review of implications for patient outcomes and treatment personalization." World Journal of Biology Pharmacy and Health Sciences 17.3 (2024): 198-204.