jamk

Security Analysis of Web Application for Industrial Internet of Things

Cybersecurity Principles and Guidelines

Srikar Bhava Narayana

Master’s thesis
May 2022
Cyber Security
Master's Degree Programme in Information Technology, Cyber Security

jamk | Jyväskylän ammattikorkeakoulu
University of Applied Sciences

jamk

Description

Narayana, Srikar Bhava

Security Analysis of Web Application for Industrial Internet of Things

Jyväskylä: JAMK University of Applied Sciences, May 2022, 84 Pages

Cyber Security. Master's Degree Programme in Information Technology. Master's thesis.

Permission for web publication: Yes

Language of publication: English

Abstract

With the technological advancement in Internet of Things (IoT), modern manufacturing and service industries are massively shifting towards connected devices and creating a connected ecosystem by deploying third-party Industrial Internet of Things (IIoT) related web applications and connectivity technologies. Due to rising global cyber-attacks on web applications, there is a need for security auditing of IIoT web applications against cyber threats by third-party companies before deploying them onto the industrial network.

The research objective of the thesis was to provide a security auditing and testing guide and principles for the IIoT web applications to perform an internal audit and verify the security of web applications against cyber-attacks before and after installing them onto an industrial network. The thesis used a constructive research approach to create SSDLC and ISACA unified innovative security auditing framework and principles for IIoT web applications by examining and analyzing academic publications, security standards, expert community recommendations documents, and best practices. The research evaluated developed constructed solutions by conducting interviews with application experts using solutions.

The results showed that implementing the security principles checklist and the security analysis framework, including GDPR policies, authentication, and secure data transmission to audit, helped detect security issues beforehand at every SSDLC phase and deploy IIoT web applications resistant to major cyberattacks after deploying to industrial networks. Utilizing security principles and implementing the proposed security auditing framework construct solves the problems stated by the research questions. The proposed construct provides the desired security auditing and testing guidelines to develop and deploy secured IIoT web applications in industrial network systems.

Keywords/tags (subjects)

Industrial IoT, IIoT, web applications, Security Analysis, Auditing ,ISACA,SSDLC

Miscellaneous (Confidential information)

Contents

Figures

Tables

Terms and abbreviations

ASVS	Application Security Verification Standard
BSIMM	Building Security In Maturity Model
CA	Certificate Authority
CIS	Center for Internet Security
DPIA	Data Protection Impact Assessment
GDPR	General Data Protection Regulations
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IIoT	Industrial Internet of Things
IoT	Internet of Things
ISACA	Information Systems Audit and Control Association
ISO	International Standards Organization
IEC	International Electrotechnical Commission
OWASP	Open Web Application Security Project
PCI DSS	Payment Card Industry Data Security Standard
SDLC	Software Development Life Cycle
SSDLC	Secure Software Development Life Cycle
WASC	Web Application Security Consortium

1 Introduction

Industrial Internet of Things (IIoT) is the term that refers to the usage of specific Internet of Things (IoT) technologies in combination with specified smart devices in cyber-physical systems in a manufacturing and industrial environment. Internet of Things (IoT) is a network that connects uniquely identifiable "things" to the internet, which has sensing, actuation, and potential programmability capabilities and collects information about the "things" from anywhere, anytime, including its state change (Chebudie et al., 2014). In IIoT, things refer to devices and equipment used by manufacturing and services industries to monitor and control devices. Presently, modern technology companies are designing and developing IoT-based connectivity devices and web applications for manufacturing and service industries. A recent trend shows manufacturing industries are the most enthusiastic IIoT product consumers. The telecommunications, construction, mining, military, and health care sectors are deploying and utilizing third-party IIoT-related web applications and connectivity technologies to remotely connect, monitor, and manage devices and equipment.

The web applications installed in industrial equipment monitoring network systems are generally third-party products that interact with both the IIoT devices and internal network servers. As a result, third-party applications threaten industrial network security by exposing information stored by several industrial networks and internal resources through web applications to external threats and vulnerabilities. So, companies require web applications security against threats and vulnerabilities without breaching and exploiting organization network security by adhering to security guidelines. Thus, this created the importance of researching the security of IIoT web applications against cyberattacks.

The research thesis is for small-sized consulting and development services to companies that design IoT embedded devices and systems. Currently, the host company is creating and developing IIoT web applications and related IIoT equipment products for various industrial customers. The company requires the security of web applications against threats and vulnerabilities by not breaching and exploiting the enterprise network security by adhering to security guidelines after deploying to the IIoT company’s internal systems. This research thesis provides guidelines and innovative solutions to the host company to design, develop and deploy secure IIoT web applications.

This thesis investigates the vulnerabilities of IIoT web applications against cyber-attacks and documents IIoT applications security auditing and testing guidelines. The auditing guidelines include security standards, processes, principles, and validation methods to be accepted and used when designing, developing, deploying, and monitoring third-party IIoT web applications. This research examines an IoT-related web application that exploits an organization's network security vulnerabilities and describes the standard security principles for auditing and testing preinstallation web applications to ensure network security in cloud computing and remote monitoring control systems. It also includes recommendations to protect web applications for secure data transfer in cloud computing systems and validate application data is compliant with General Data Protection Regulations (GDPR).

2 Research purpose, objectives, questions and methodology

2.1 Research objective and purpose

The Master's thesis researches and analyzes academic publications, industry standards, and best practices to design and implement secure IIoT-based web applications for industrial manufacturing and service industries vulnerable to cyberattacks. This research will help IIoT application developers achieve the desired level of transparency and security awareness for developed applications and enable IoT application providers to get adequate information about the threats and vulnerabilities of their applications. The thesis also provides a security auditing and testing guide for the IIoT web applications to perform an internal audit and verify the security of web apps against cyber-attacks before and after installing them onto an industrial network. The conclusive purpose of the thesis is to report the security process and guidelines to design, develop, validate, and deploy secure web applications in the testing and implementation phase and later during the update process.

The objectives for this thesis focused on researching and preparing a security framework to audit and test the security of IIoT web applications that:

- Research and document the innovative framework to audit and test a secure IIoT web application in all software development life cycle (SDLC) phases as a third-party product in the organization's industrial monitoring network using security standards, processes, regulations, and checklists.
- Ensure the web application is not breaching any security standards in the installed system of the organization's network.
- Protect personnel data and check that collected data is compliant with General Data Protection Regulations (GDPR) depending on regional and global regulations. For example, EU regulations and Finnish regulations are applicable if operated in Finland.
- Recommend secure data transmission techniques and solutions for cloud data and web application servers.

2.2 Research methodology

The thesis study utilizes the "constructive research approach" to steer the research by gathering qualitative data through literature review and security standards document analysis. The "con-

structive research approach" is decided to build an innovative security auditing and testing framework for any web application, accomplishing all software phases, including development, testing, and installation, to solve the web applications' practical security threats and vulnerabilities.

According to Lukka (2003), "The constructive research approach is a research procedure for producing innovative construction intended to solve problems faced in the real world, and that means making a contribution to the theory of the discipline in which it is applied" (p. 83). According to Kasanen (1993), there are six phases in the constructive research approach: "1) Find a practically appropriate problem that also has research prospects to be solved. 2) acquire a comprehensive understanding of the research topic. 3) Create an innovative solution concept or framework.4) Show how the practical solution works.5) Examine the scope of applicability of the solution. 6) Demonstrate the theoretical connections and the contribution of the solution concept to research "(p. 246). The constructive research approach listed phases is practically applied to this research thesis to develop and examine solution.

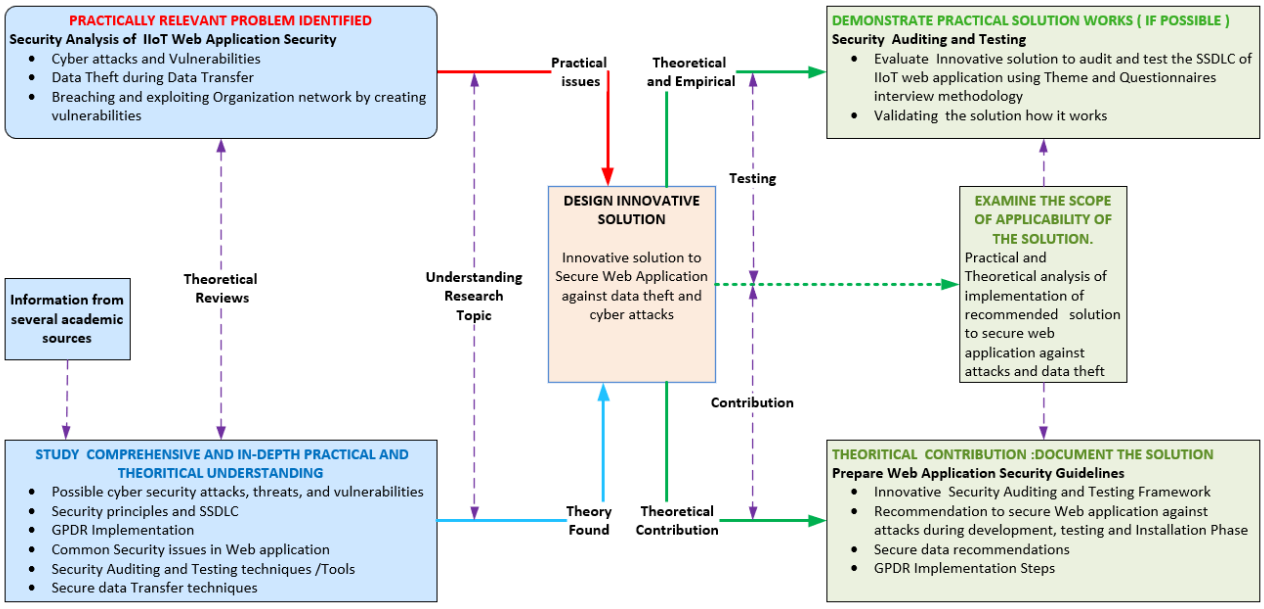

Figure 1. Elements of a constructive research approach used in thesis work.

Above, Figure 1. illustrates the application of the constructive research methodology process for thesis work. The current plan is to implement a six-phased constructive research approach to develop a solution to audit and test the security of IIoT web applications that collect, monitor, and transmit data in secure mode from IIoT devices against cyber-attacks and data thefts.

The first phase identifies the practical research topic by comprehending and reviewing the web application security problems and requirements of IIoT web application and device developing enterprises. Next, in the second phase, web application-related security attacks, security standards, principles, and legislation are studied using literature review methodology in comprehensive and in-depth ways to understand the research topic. The outcome of the second phase is a practical and theoretical understanding of the research topic.

Next, in the third phase, an innovative solution is constructed using earlier found practical issues and theoretical acquired knowledge. Subsequently, conducting the software expert Interviews using theme and questionnaires method with the host company, the innovative solution is evaluated using y to demonstrate testing in solving the research problem in the fourth phase. Next, in the fifth phase, the innovative constructed solution validation results in the fourth phase are examined to analyze the applicability of the solution concept and recommend future research suggestions (Lukka,2003). Finally, the theoretical connections and the research contribution of the developed solution concept are documented in a report to complete the solution analysis (Lukka,2003).

2.3 Ethicality of the thesis work

According to Ethical Recommendations for Thesis Writing at Universities of Applied Science (2020), it is essential to Encourage responsible research, to avoid deceit in research, and, on its own, to increase the quality of the thesis throughout the thesis process (p. 4). The thesis work is adherent and follows all recommendations, ethical principles, and practices recommended by ARENE (the Rectors' Conference of Finnish Universities of Applied Sciences) in the document Ethical Recommendations for Thesis Writing at Universities of Applied Science (2020) and the Finnish Advisory Board on Research Integrity (TENK) in the document Responsible Conduct of Research and Procedures for Handling Allegations of Misconduct in Finland (2012).

The research study is not managing or processing any personal information, and if the research comprises the processing of personal data, it has regulated the General Data Protection Regulation and the Data Protection Act. The method used for data acquisition, research, and evaluation fits scientific criteria and is ethically sustainable (Responsible Conduct of Research and Procedures for Handling Allegations of Misconduct in Finland, 2012). The thesis has given due consideration to the work and accomplishments of other researchers by respecting their work, appropriately citing

their publications, and giving the credit and weight they deserve in conducting the researcher's research and publishing its findings by not copying their work and always citing sources and reference listing of their work (Responsible Conduct of Research and Procedures for Handling Allegations of Misconduct in Finland, 2012).

Acquired Research permits from the thesis's target organization and thesis supervisor through the signing of a Masters' Thesis agreement and a Non-Disclosure Agreement. The thesis work is committed to ethical rules and practices in the target organization. The thesis work is also committed to thesis targeted organization that Information about the organization and its products specification is kept confidential. Research thesis work utilizing openly available data only. The research report published to Thesis repository document only non-confidential findings and publish findings only. Thesis work is committed to ethical rules and practices in the research target organization. The research utilizes officially known security standards, threats, and vulnerabilities, which mitigates the possibility of criminals exploiting the findings and contents. The report does not disclose the names, network infrastructure architecture, and tools of companies and organizations' the research is targeted.

2.4 Research questions

The research study is an extensive literature and document review to define the framework by answering the following main research question.

How to do auditing and testing of industrial Internet of Things (IIoT) web application security and ensure that user data is compliant with the General Data Protection Regulations?

The primary research question is sub-divided into the following four sub-questions:

- How to validate and ensure the security of IIoT web application system is not breached?
- How to validate and ensure the security of an organization's network is not breached and not exploited by IIoT web applications?
- How to secure data upload and download to and from a cloud data server?
- How to implement and ensure that user data in IIoT web applications comply with the EU's and Finland's General Data Protection Regulations?

2.5 Research tasks

The thesis explores the research question by completing following tasks:

- The thesis collects information by reviewing the online accessible literature and documents using search terms and terminologies.
- Classifying the information collected related to web applications for IIoT devices into the below categories:
  - Secure Software development life cycle for IIoT web applications
  - Web application security challenges, cybersecurity attacks, threats, and vulnerabilities
  - Web application security principles and standards such as International Standards Organization (ISO)/IEC 27002, and other security standards are to be considered during web application development, testing, and validation.
  - Data encryption and certificates for secure data transfers of IIoT web applications
  - Finland, and the EU GDPR Implementation
  - Auditing and testing of web application security
- Construct a security auditing and testing framework for web applications.
- Prepare a security guide to evaluate the security of web applications before installing them on the organization's network.

- Evaluate the innovative solution developed in this research using constructive methodology by collecting feedback from experts from the thesis host company who will enforce and integrate this solution in the secure software development life cycle of the new IIoT web applications to conduct a security analysis using Interviews and questionnaire methodology.

3 Research related theoritical concepts

3.1 Industrial Internet of Things (IIoT)

The IIoT provides a gateway into the company's operations and assets by integrating machine sensors, middleware, software, and backend cloud compute and storage systems (Gilchrist, 2016). For the IIoT, GE (General Electric) coined the term "Industrial Internet." Cisco called it the Internet of Everything, while others called it Internet 4.0 (Gilchrist, 2016).

According to Yu and Gu (2019,1), the "IIoT connects and intelligently manages industrial systems through sensing devices and actuators equipped with ubiquitous networking and computing capabilities, is a critical component of future industrial systems"(Yu & Gu ,2019). According to the IIoT Consortium, the development of new global industrial internet networks is manifesting itself in industries such as the power industry, healthcare, manufacturing, mining, transport, and logistics, as well as in industrial applications related to those industries (van Lier, 2017).

The European Commission has also acknowledged that IIoT development allows for physical and digital world integration (van Lier, 2017). According to the European Commission, this new integration potentially creates a new intelligent environment capable of sensing, analyzing, adapting, and making our lives easier, securer, more efficient, and more user-friendly. IoT inaugurates a new era of global connectivity and intelligence when parts, goods, services, platforms, and other things connect and work together through a communication network for digital processing (van Lier, 2017).

3.2 Industrial Internet of Things web application

Currently, web-based applications are the most commonly used communication and information exchange. In general, a web application is a software program that performs specified functions and tasks over the Internet using web technologies, communication protocols, and browsers. In the context of IIoT, a web application is a server or controlling program providing web interfaces for users and administrators (Lee & Park, 2017). The web application is the user-interfacing com

ponent of an IoT system that runs on an operating system, an Android smartphone, or tablet to connect, monitor, control, and collect data from the connected device. In the case of an industrial network environment, the web application often runs as a web application server, also supporting web application clients to access IoT devices remotely.

3.3 Industrial Internet of Things web application system architecture

IIoT system architecture: the basic design in a complex system consists of a web server, web application content that runs on the web server, and a backend database storage or cloud storage that the application accesses to store and retrieve data in a secure mode (Cross, 2007). Below Figure 2, depicts the IIoT system architecture and component interactions, and Table 1 describes the IIoT system components, which include a web application.

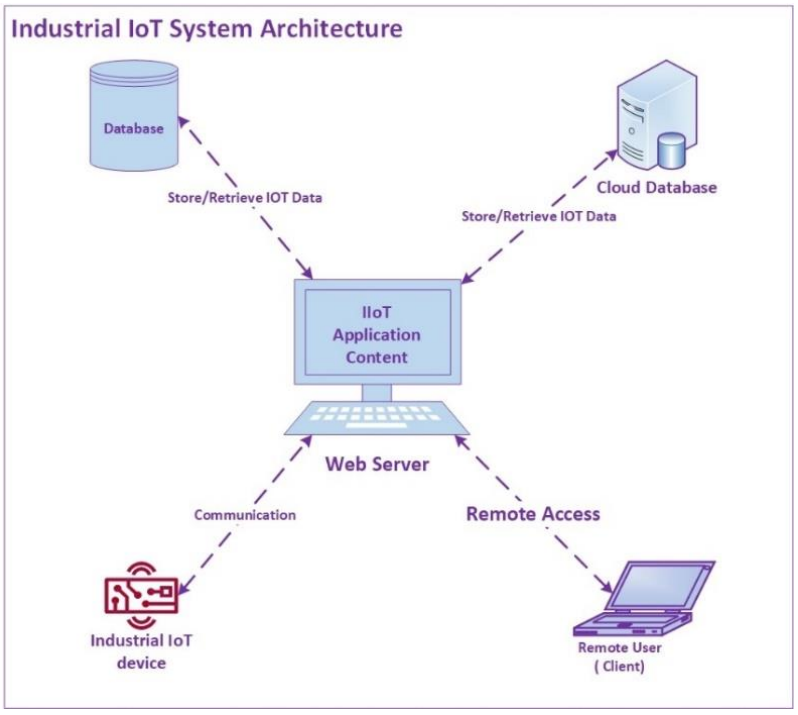

Figure 2 . IIoT web application system architecture

Table 1.Components of IIoT System Architecture

IIOT SYSTEM COMPONENT	DESCRIPTION
Web Server	Web Server is an IoT device monitoring and administrating service that runs application server on the organization's system that serves up web content (Cross, 2007). These services are accessed using the Hypertext Transfer Protocol Secure (HTTPS) protocol on port 443, while some servers may utilize non-standard ports. In an IoT system, a web server interacts with the web client application and IoT devices.
Application Content	The Application Content is interactive software that processes web requests and performs various activities using parameters provided by the web browser and IoT devices (Cross, 2007). The application content runs on the webserver
Database	The database is an organized collection of structured information or data stored in an electronic format on a system. The web serves to access the database servers to retrieve or store data. Commonly, the data is stored in a database server in a completely different system in the same or different network other than the webserver is running. Nowadays, the webserver stores database in the remote cloud. Usually, web servers store data collected from IoT devices.
Remote User (Client)	Client applications run in browsers or as desktop applications or mobile applications. Remote users monitor IoT devices by accessing the application content running on a web server using a client application.
IIOT device	IIOT devices provides as an interface between industrial equipment's or employees and web server. Web servers connect, control, and monitor industrial equipment's using IoT devices.

3.4 Modern IoT web applications software architecture

Earlier, web applications used server-side frameworks that rendered HTML, JS, and CSS pages to the client, and later clients requested new page updates over Hypertext Transfer Protocol (HTTP) when required (Hoffman, 2020). Current web applications in the market are designed and built using advanced tools in a brief time frame. Presently, server web applications use Apache, ASP.Net, and other platforms, and client applications use HTML+, JavaScript, and XML.

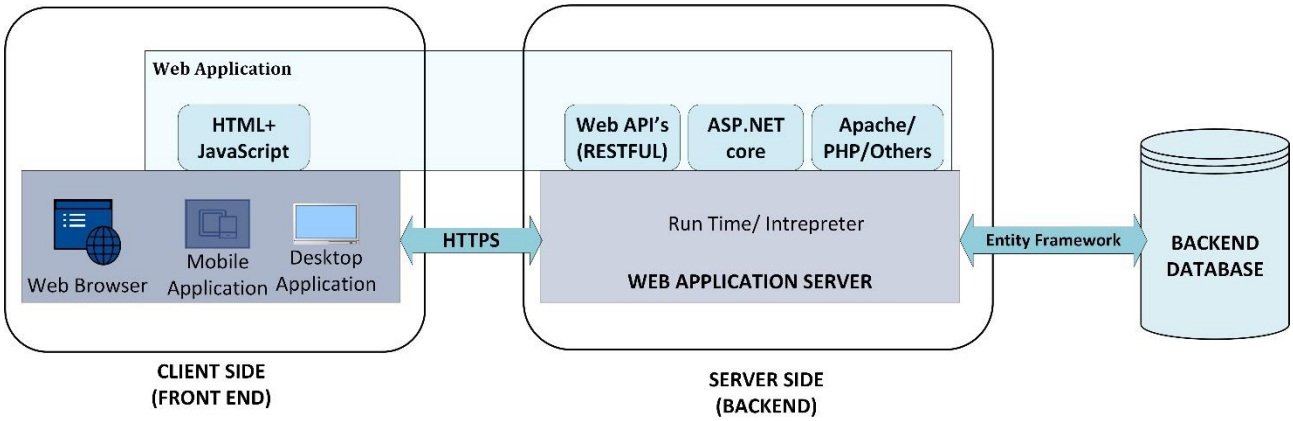

Figure 3. Three Tier Architecture of Modem web Application Software

Modern Web Applications consist of three layers, and Figure 3 presents the architecture of the web application. The first layer is the user-side and consists of a browser or desktop/mobile application that displays the IoT device data. The second layer is server-side, which generates the web content of IoT device data. It contains restful web API's and various generation tools, such as ASP.Net, Apache, and PHP. The third layer is the backend databases, where the data is stored.

3.5 IIoT web application server hosting enterprises network design

Traditional SCADA (Supervisory Control and Data Acquisition) systems were standard in the industry, providing real-time data and remote administration of manufacturing equipment such as PLCs or field devices (Rubio et al., 2018). They are now increasingly integrated with other services, allowing for the sharing of data and the adoption of new enterprise applications, which was a result of the standardization of the software and hardware used in control systems and enhanced by the widespread use of Ethernet or TCP/IP, as well as wireless technologies such as IEEE 802.11 a/c/g or Bluetooth (Rubio et al., 2018). Currently, following this fourth Industrial Revolution, or Industry

4.0, started by the deployment of networking and communication technologies, mainly IoT or cloud computing, to current control and automation systems. The generic network architecture of industrial 4.0 for control and monitoring automation systems is illustrated in Figure 4

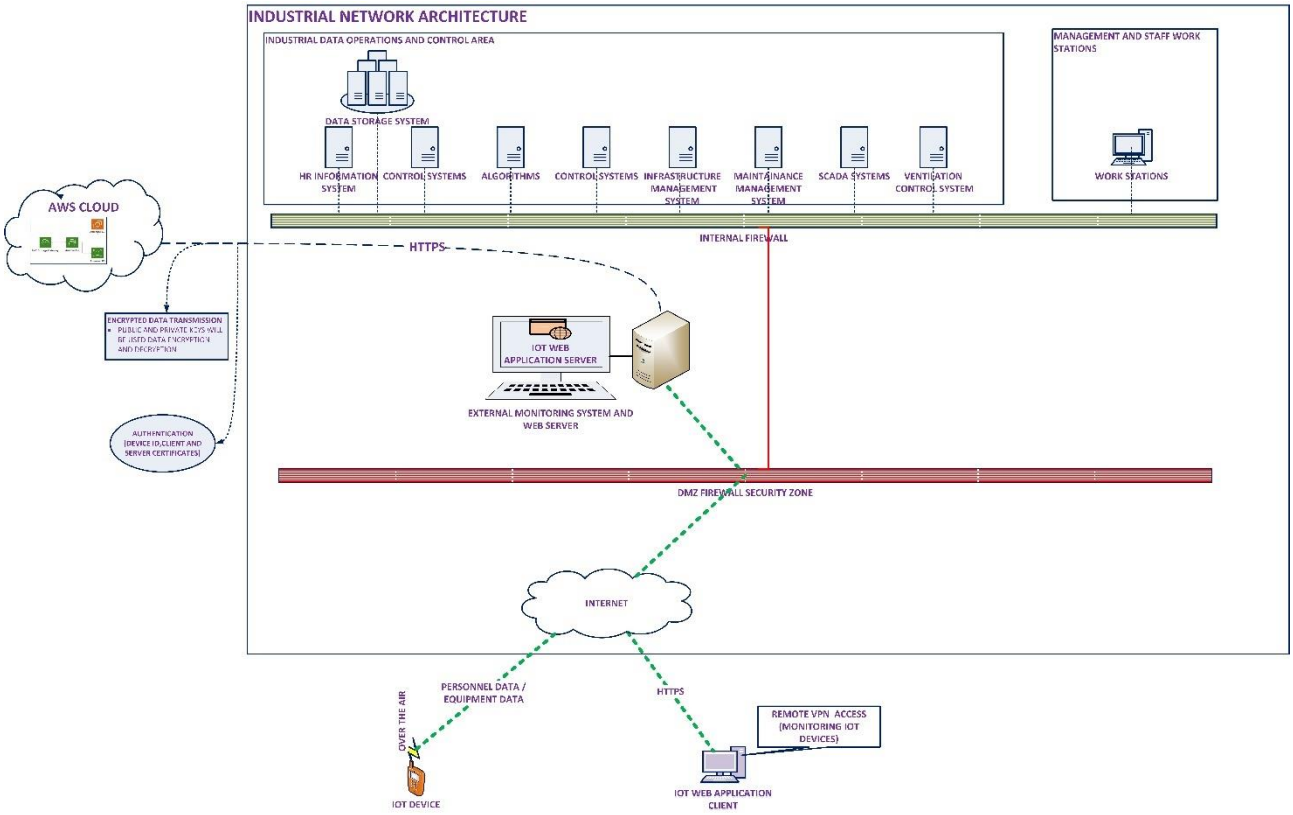

Figure 4. Network Architecture for industrial monitoring and operations

The web application of the product is installed and used as a third-party application deployed in organization network is presented in Figure 4. The IIoT web application is executed directly on the organization's application server, connected to the internal and extranet networks. The primary functionality of the IoT web application server is to collect user data from IoT devices on the monitoring and controller networks of the organization. The data gathered is processed on an internal server, uploaded, and downloaded to and from a cloud data server on the organization's network.

3.6 Industrial Internet of Things security objectives

Security is fundamentally the protection of assets of industrial organizations, which includes network devices, IoT devices, business information, and technical data. Cybersecurity is a critical concern for industrial networks due to IIoT devices, related web applications, and other systems connected to organizational networks. IIoT security objectives must include the pillars of information assurance (IA), in addition to physical security and resilience. Data security and integrity, control robustness, and the safety of physical assets and people are all critical in IIoT devices, networks, and applications (Bhattacharjee, 2018).

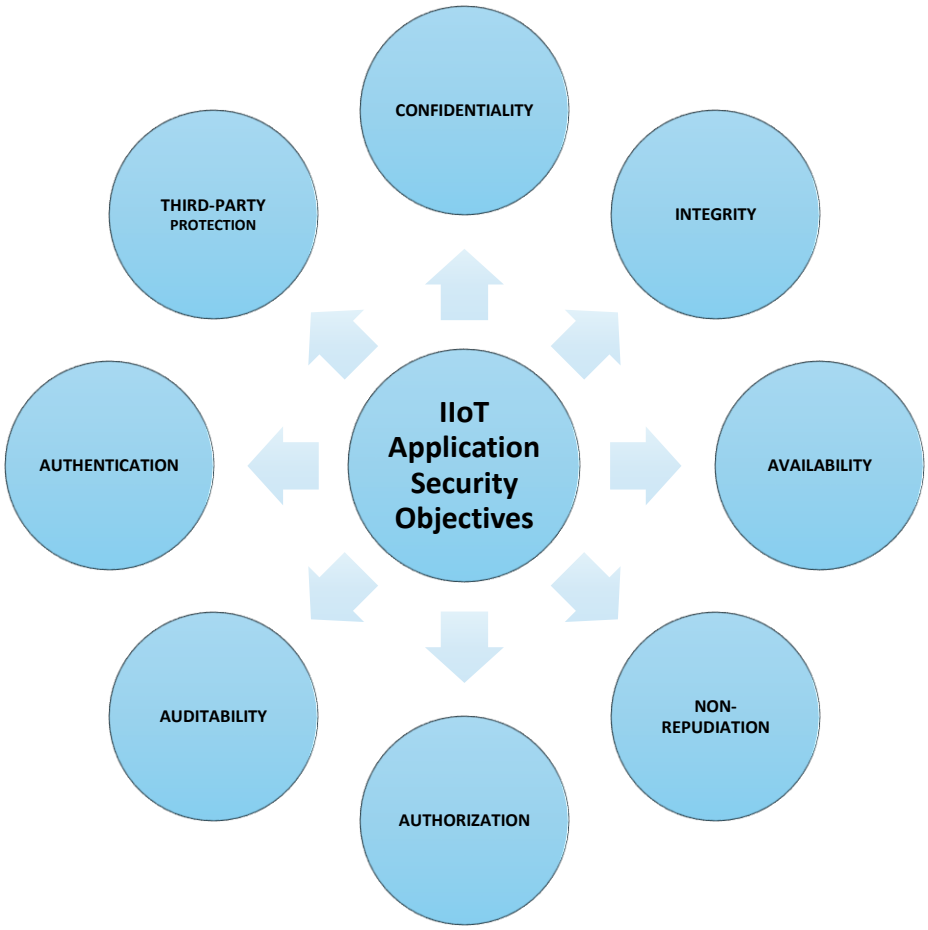

Figure 5. Security objectives of IIoT web applications

In the IIoT security context, the pillars of information assurance and the primary security objectives of industries that need to be defended and protected against threats and attacks as presented in Figure 5 are confidentiality, integrity, availability, non-repudiation, authorization, auditability, authentication, and third-party protection (Bhattacharjee, 2018; Rubio et al., 2018;

van Lier, 2017). The goal of security requirements of the IIoT web application and the security control mechanism implemented in the IIoT web application is required to ensure the security objectives. The following sub-section summarizes the security objectives in the context of IIoT web applications.

3.6.1 Confidentiality

The confidentiality objective refers to preventing IIoT web applications or organizations' sensitive data or information from being accessed by or disclosed to unauthorized individuals or systems (Chopra & Chaudhary, 2020; Dzung et al., 2005). Confidential data such as passwords, personnel information, and sensitive information might be stolen from the unsecured communication channel, leading to confidentiality compromise (Panchal et al., 2018). IIoT web applications that connect remote machines and sensor devices in industrial environments gather extensive, non-uniform, and critical data that needs to be processed in real-time (Yu & Guo, 2019). A Data breach in the IIoT web application system can have a disastrous impact on the industry or its critical infrastructure (Yu & Guo, 2019). Thus, data confidentiality safeguard is one of the most intense security challenges and requirements in IIoT applications and systems.

3.6.2 Integrity

The Integrity objective refers to preventing undetected modification of IIoT web application or organization information or data by unauthorized individuals or systems (Dzung et al., 2005). In IIoT applications and systems, this objective is to ensure the information such as IIoT control commands, technical sensitive or critical data values is not modified. This objective includes protection against application data tampering on the internet through message injection, replay, or delay. Interruption of integrity may result in security concerns that can harm equipment or individuals (Dzung et al., 2005). Thus, application data integrity is critical to preventing the takeover of IIoT devices through the modification of transferred or application data.

3.6.3 Availability

The availability objective refers to guaranteeing that unauthorized individuals or systems cannot restrict access or use to authorized users (Dzung et al., 2005). An attack may prevent a legitimate

user or component of the system from accessing specific services (Panchal et al., 2018). In IIoT applications and systems context, it is to assure the communication systems between industrial systems, servers, and other web applications, and outside IIoT equipment is not denied. Availability breaches, also known as denial-of-service (DoS), result in financial losses and detrimental effects on safety, as operators may lose their capacity to monitor and manage the process (Dzung et al., 2005).

3.6.4 Non-repudiation

The non-repudiation objective refers to being able to furnish indisputable evidence to a third party who originated a specific activity in the system, even if the initiating actor does not cooperate (Dzung et al., 2005). It ensures that an individual or system cannot later deny after initiating the activity (Bhattacharjee, 2018). The objective applies to establishing responsibility and liability (Dzung et al., 2005). In the IIoT applications and systems context, this is a critical security requirement concerning regulations.

3.6.5 Authorization

The authorization objective is to prevent access to the IIoT web applications and systems by individuals or systems who do not have permission to access them user (Dzung et al., 2005). It is also known as access control and refers to the methodology to differentiate legitimate and illegitimate application and system users for all other IIoT security objectives user (Dzung et al., 2005). In the IIoT web applications and systems context, it refers to limiting the permission to users to issue commands and control of IIoT equipment using the web application user (Dzung et al., 2005).

3.6.6 Auditability

The Auditability objective refers to rebuilding the complete Industrial system and application conducted from chronological records of all relevant operations performed on systems (Dzung et al., 2005). In the IIoT system and applications context, this objective is significant to detect and find the cause of system malfunctions after the attack and establish the scope of the malfunction or the effects of a security event (Dzung et al., 2005). It helps to gather and analyze diagnostic reports of the systems.

3.6.7 Authentication

Authentication is the process of determining a system user's legitimate identity and mapping that identity to a system-internal regulation that identifies the legitimate user (Dzung et al., 2005). In simple words, it refers to the validation of IIoT applications and system user identity. The majority of other security objectives, notably objective authorization, utilize an authentication-based distinction between authorized and illegitimate users (Dzung et al., 2005).

3.6.8 Third-party protection

The Third-party protection's objective is to prevent harming the third party through an information system and web applications (Dzung et al., 2005). In the network security context of an IIoT organization, the threat agent exploits information system and web application software vulnerabilities to successfully attack the organization's systems, data, and external third parties without jeopardizing physical security or causing damage to devices and industrial equipment (Dzung et al., 2005). The attackers may use distributed DoS (DDoS) or worm attacks. As a result, the market prestige of the IIoT company is hurt, which impacts the organization's future business.

3.7 Secure software development life cycle (SSDL) of web application

Earlier, the traditional software development process was the classic waterfall model, a monolithic releases framework in which development proceeds sequentially through a series of phases: requirement, design, implementation, verification, and maintenance phases (Huang et al., 2010). Later, developed various iterative and incremental SDLC approaches to enable more frequent delivery and adaptation of software and reduce the rigidity of the software development cycle. The SDLC is a process commonly used by developers. Formerly, security was not a priority throughout the software development life cycle, but modern web applications cannot wait for security subsequently. The only feasible approach is to develop secure software by including security at every phase of the software development life cycle, from the requirement to release, which creates a Secure Software Development Life Cycle (SSDLC).

In practice, several secure SDLC frameworks are available that give both descriptive and prescriptive guidance. In industry practically, prescriptive guidance defines how the secure SDLC should

function, whereas descriptive guidance details how to use in industrial use cases (OWASP Web Security Testing Guide Version 4.2, 2020). A prescriptive framework lists possible security controls and policies integrated within the SDLC. Descriptive guidance assists in driving the decision-making process by demonstrating what has worked effectively for other companies. Security standards Building Security In Maturity Model (BSIMM) included in descriptive secure SDLCs, and the prescriptive secure SDLCs include OWASP's Open Software Assurance Maturity Model (Open SAMM) and ISO/IEC 27034 (OWASP Web Security Testing Guide Version 4.2, 2020)

According to Hoffman (2020), the Secure Software Development Life Cycle (SSDL) is a "common framework that allows software engineers and security engineers to work together in order to write more secure code." (Hoffman, 2020). SSDLC is one of the significant factors, which comprises software requirement analysis, design, development, testing, deployment, and maintenance. The best practice for any manufacturing and service industry companies to mitigate the threats and risk of vulnerabilities exposed in web applications started from the requirement analysis phase to the regression testing phase. Mitigation techniques primarily depends on a secure software development life cycle (SSDL), and others include secure coding best practices, secure application architecture, regression testing frameworks (Hoffman,2020).

According to the OWASP Web Security Testing Guide Version 4.2 (2020), one of the most practical methods to avoid security vulnerabilities in IIoT web applications is to enhance the Software Development Life Cycle (SDLC) by incorporating security into all of its phases. Prevention is the ideal technique of managing vulnerabilities, and it may be embedded into a web application fundamentally (Lepofsky, 2014). SSDLC is a development framework that is imposed on software artifacts (OWASP Web Security Testing Guide Version 4.2, 2020). The process of adding security into the development process is summarized in Figure 6 , which depicts a typical secure software development life cycle and the estimated increased expense of fixing security vulnerabilities in SSDLC (OWASP Web Security Testing Guide Version 4.2, 2020). IIoT web applications developing organizations should conduct auditing of the complete SDLC to verify development process includes security in every phase, as shown in Figure 6. Every cycle of the SDLC has associated security concerns that need to be incorporated directly into the tools and procedures utilized at that step.

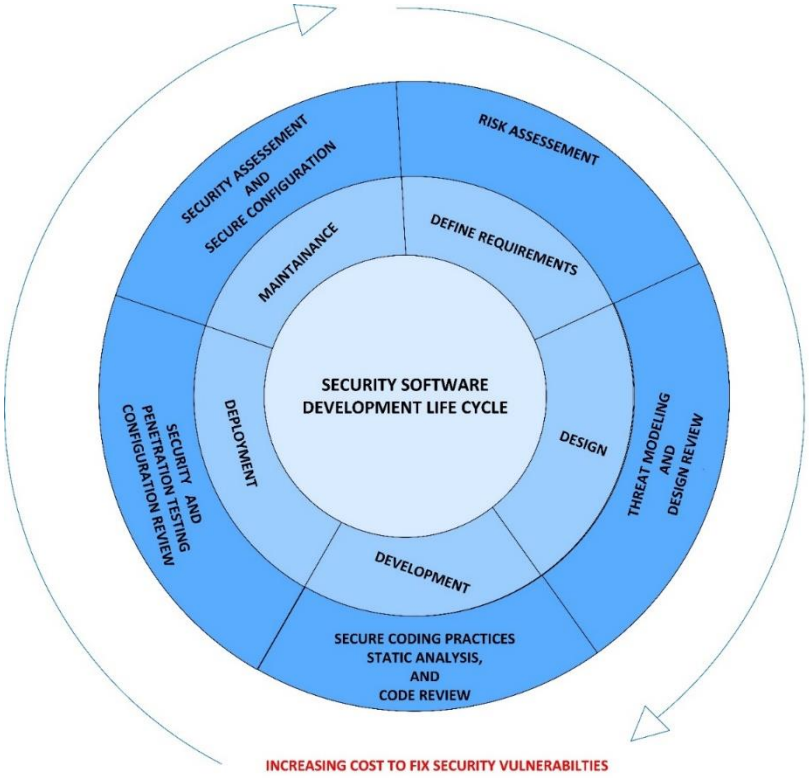

Figure 6. Secure Software Development Life Cycle

Secure Software Development Life Cycle consists of five distinct phases as shown in Figure 6: Define requirements, Design, Development, Deployment, and Maintenance (OWASP Web Security Testing Guide Version 4.2, 2020). The following sub-section describes the security in the software development life cycle in the context of IIoT web applications.

3.7.1 Define requirements phase

Defining requirements in SSDLC is the initiation phase of building security into the software development life cycle. The initiation phase comprises two critical activities contributing to the SDLC process's overall security: business and security requirements specification and feasibility study (Lepofsky, 2014). Generally, industrial business requirements specifications of IIoT web application functionalities are identified and studied in adequate detail and clarity so that the application design phase can proceed efficiently to the next stage (Lepofsky, 2014). Along with functional business requirements, it is vital to identify all security requirements from the perspective of the manufacturing and service industries. Security risk analysis and assessment are incorporated and play a significant role in the requirements definition process.

Before a web application's security requirements are defined, the adequate SDLC methodology is clarified and documented., which means establishing appropriate policies, standards, and documentation based on the application's company requirements. For example, - Java coding standards, cryptography standards, tools, and other programming languages used in developing web applications. Now security requirements are defined by conducting risk analysis and assessment web application working in the security perspective. Risk analysis and assessment need to consider security mechanisms in perspective applications are web application security objectives presented in Figure 5 along with session management, transport security, legislative and standard compliance (OWASP Web Security Testing Guide Version 4.2, 2020).

3.7.2 Design phase

In the design phase of the IIoT web application, companies create secure software development programs from requirements. The design phase entails converting business requirements to architecture and developing the functionality of the IoT web applications (Lepofsky, 2014). The design phase activities flow illustrated in Figure 7 shows Initially, enterprises that build IIoT web applications document the design and architecture in the design phase. The documentation comprises models, textural documents, and artifacts, alongside enforcing security levels defined in the earlier requirements phase (OWASP Web Security Testing Guide Version 4.2, 2020).Designing a secure web application is not feasible unless the possible threats are known. So, Security weaknesses and threats are discovered in this design phase to reduce bug fixing expenses later by changing the architecture and design of the application. After finalizing the design and architecture, build Unified Modeling Language (UML) describing the models' application functionality (OWASP Web Security Testing Guide Version 4.2, 2020). Now, the software development program conducts a more comprehensive security analysis than the preliminary risk assessment conducted in the previous phase. As an outcome in this phase, exercised threat modeling activity (Vinod et al., 2008).

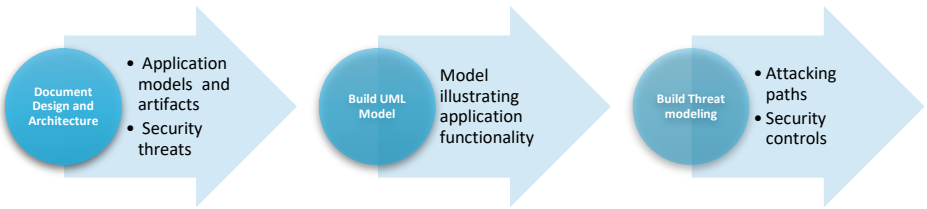

Figure 7. Security in SDLC Design phase tasks flow

Threat modeling activity determines potential threats related to the web application to consider required security controls only. Additionally, threat modeling supports detecting threats posed by possible vulnerabilities and the security controls functional to manage related risk (Lepofsky, 2014). In this, every threat's different attacking paths and corresponding security controls are determined to prepare web application, specific threat model. Analyze the design and architecture to determine it mitigates all threats, and the company approves them.

3.7.3 Development phase

The development phase is the implementation of the earlier design phase. In this phase, web application, developers must follow secure programming techniques to develop secured and robust solutions resistant to hacker attacks. For example: - if the application is using SQL, the developer should use dynamic queries to avoid SQL injection. Daily code testing is a necessary part of software development activity, where it identifies and resolves coding issues by including security scanning of the code in the daily build process (Vinod et al., 2008).

In this phase, the developer uses automated scanning and static analysis tools to detect security weaknesses in daily builds (Vinod et al., 2008). Additionally, the security team conducts code walkthroughs with application developers to comprehend a high-level understanding of the application's data and control flow, layout, and structure (OWASP Web Security Testing Guide Version 4.2, 2020). Later, the security team and code review team also conduct code reviews to examine the actual web application code for security flaws (OWASP Web Security Testing Guide Version 4.2, 2020).

3.7.4 Deployment phase

The deployment phase comprises security testing, penetration testing, and configuration review activities in addition to application functionality testing. This phase starts assuming the earlier stages of requirements testing, design analysis, and reviewing code detected almost all the web application security and functionality issues (OWASP Web Security Testing Guide Version 4.2, 2020). The deployment phase helps uncover any additional security vulnerabilities that may have occurred attributed to negligence or modifications made after the threat modeling phase (Vinod

et al., 2008). In web applications, the webserver is configured securely. The ports needed only for the application's operation should be opened on the firewall. (Vinod et al., 2008).

Penetration testing after web application deployment in an industrial network environment provides additional security verification reports to ensure security vulnerabilities are exposed as threats to exploit. This testing also reviews the web application server's deployment network environment and application configuration aspects to avoid vulnerabilities exposed due to application default settings (OWASP Web Security Testing Guide Version 4.2, 2020). Also, conducted security testing of web applications to check if the application satisfies organization security requirements defined during the initial phases and new threats to the not considered during threat modeling or weakness during testing (Vinod et al., 2008). An application will evolve in the future by adding, modifying, or removing new features in response to user feedback or reported bugs. Additionally, every modification to the application code needs regular testing to avoid exposing new security vulnerabilities Need to conduct frequent security testing and penetration testing on the web application.

Penetration testing after web application deployment in an industrial network environment provides additional security verification reports to ensure security vulnerabilities are exposed as threats to exploit. This testing also reviews the web application server's deployment network environment and application configuration aspects to avoid vulnerabilities exposed due to application default settings (OWASP Web Security Testing Guide Version 4.2, 2020). Also, conducted security testing of web applications to check if the application satisfies organization security requirements defined during the initial phases and new threats to the application that were not considered during threat modelling.

3.7.5 Maintenance phase

In the maintenance phase, organizations define a methodology to manage operational aspects of the IIoT web application and its deployed organization network environment. The maintenance phase conducts regular security checks on IIoT applications and deployed infrastructure to confirm that no new security risks are exposed and that the current security level is stable (OWASP Web Security Testing Guide Version 4.2, 2020). It is critical to look out for network security, hosts, and application configurations on the application hosting servers in this phase.

In this phase, an application will evolve in the future by adding, modifying, or removing new features in response to user feedback or reported bugs. Additionally, every modification to the application code needs regular testing to avoid exposing new security vulnerabilities (Vinod et al., 2008). During maintenance, after every modification of the web application is accepted, verified, and deployed, it is critical to confirm that no security level is compromised (OWASP Web Security Testing Guide Version 4.2, 2020). This maintenance phase consists of additional security assessments and security configuration management activities. This stage processes also include backup and recovery, change control, and incident response (Vinod et al., 2008). Need to conduct regular security testing and penetration testing on the web application in this phase.

3.8 IIoT web application security standards

Web security standards refer to secure coding standards, essential practices, and controls that an organization follows during the development, enhancement, and deployment of websites and web applications. This section lists some of the most commonly accepted regulations and standards from authoritative sources of information. IIoT organizations could consider operational industrial security standards and regulations at the strategic level, while expert organizations provide tactical advice (Lepofsky, 2014). Expert community organizations' recommendations are advantageous because they are more specific and reflect actual threats, while regulations and standards establish obligations at a higher level.

The purpose of web security standards controls is to prevent insecure code introduction and deploy security controls to achieve the objectives of IIoT web application security presented in section 3.6. In Table 2 lists the industrial standards that need to be considered in the SSDLC of the IIoT web applications.

Table 2. Industrial standards for IIoT web application

Industrial Standard	DESCRIPTION
ISO/IEC 27001	ISO/IEC 27001 defines the standard requirements specifications for information security management systems by ISO, an international organization for standardization. It provides standards security requirements for all phases of SSDLC of IIoT web applications (Vinod et al., 2008).
ISO/IEC 27002	ISO/IEC 27001 defines the standard code of practices or implementation guidelines for the information security management system. It provides standards security implementation of requirements specified in ISO27001 for all phases of SSDLC of IIoT web applications (Vinod et al., 2008).
ISO/IEC 27034	ISO/IEC 27034 defines standard technical guidelines to support organizations in integrating security into the techniques used for managing their applications. It provides technical guidelines for all phases of SSDLC for application systems.
Payment Card Industry Data Security Standard (PCI DSS)	PCI Security Standards Council (PCI SSC), a global organization, develops and promotes data security standards and resources for secure payments globally. IIoT web applications storing, processing, and transferring payment card data must implement PCI DSS to secure payments.

The recommendation documents of expert organizations can provide detailed technical guidance to IIoT web application development organizations (Helmiawan et al., 2020) using industrial standards. For example, OWASP. Table 3 lists expert organizations' advising recommendations to be considered in the SSDLC of the IIoT web application.

Table 3. Expert Organisations recommendations for IIoT web application

Expert organization	DESCRIPTION
OWASP Application Security Verification Standard (ASVS)	OWASP ASVS defines the functional and non-functional requirements and security controls that are necessary for all phases of SSDLC of modern web applications and web services (OWASP Application Security Verification Standard 4.0.3, 2021)
OWASP Top 10	OWASP Top 10 presents most critical security risks to web applications
OWASP Web Security Testing Guide	OWASP Web Security Testing Guide is a detailed guideline to assess the security of web applications and web services. It offers a foundation of best practices for web application penetration testers and companies worldwide (OWASP Web Security Testing Guide Version 4.2, 2020).
Center for Internet Security Critical Security Controls (CIS Controls)	The CIS Critical Security Controls (CIS Controls) is a prioritized group of countermeasures designed to protect systems and networks from the most common cyber-attacks. (CIS Critical Security Controls Version 8, 2021).CIS Control 16: Application Software Security defines controls to manage any application software to prevent, find, and address security vulnerabilities before affecting the organization (CIS Critical Security Controls Version 8, 2021).
Penetration Testing Execution Standard (PTES)	PTES is a security standard that was created to enable the provision of security services based on penetration testing analysis. It presents a framework for software intrusion testing with an integrated approach as a goal (Correa et al., 2021).

3.9 Web application threats, vulnerabilities, and attacks

The word "threat" is used in ISO 27001 to refer to the process of identifying and assessing scenarios that are unexpected or undesirable that, if they occur, might cause harm to the web application and organization (Chopra & Chaudhary, 2020).A web application threat is any malicious or harmful activity of threat actors that targets one or multiple components potential exploits of the

web application system, such as the webserver and user-side interfaces such as desktop, mobile, or browser-based applications (Sadqi & Maleh, 2020). The term threat actor refers to the adversary who initiates or inflicts the web application exploit.

A “vulnerability” is a weakness in system or application that makes it susceptible to threats. A web application vulnerability is a mistake creating a weakness in application software. WASC Threat Classification (2010) defined vulnerability in terms of the Common Weakness Enumeration (CWETM), a community-maintained database of software vulnerabilities. "An occurrence of a vulnerability (or numerous vulnerabilities) in software that may be exploited by a party to enable the program to manipulate or access unintended data, disrupt proper execution, or conduct improper activities not explicitly allowed to the person exploiting the vulnerability." (WASC Threat Classification ,2010.).

According to WASC Threat Classification (2010), a weakness as vulnerability definition is “A type of error in software that, in good conditions, could contribute to the introduction of vulnerabilities within that software. This term applies to errors regardless of whether they occur in implementation, design, or other phases of the Software Development Life Cycle” (WASC Threat Classification,2010).Threats and attacks to the web application are consequences caused by the vulnerability. Analysis of Vulnerabilities is significant to prevent threats and attacks on the web application. Web Application vulnerabilities affects the web server, application servers and web applications.

Web applications have continued to dominate cyberattacks in recent years and are continually increasing. Globally, we find several lists of web application threats and vulnerabilities. Threats and vulnerabilities in web applications depend on the software language and operating systems used to develop the application. In design and development, web applications need to gather a list of threats which makes a good starting point (Mueller, 2015). Though threats such as viruses and malware on web application client systems have generally been among the most severe, there are innumerable threats that may be harmful, sometimes even catastrophic, to an industrial organization and its capacity to do business (Steinke et al., 2011). With the proliferation of online threats, web application security has become one of the most critical aspects of an industrial company.

Internationally, several organizations document and publish web application threats and vulnerabilities to analyze and validate the impact of different threats. The resources and tools are available also help identify vulnerabilities that are often exploited by attackers. In this research, web threats, vulnerability, and security resources by the Open Web Application Security Project (OWASP) and the Web Application Security Consortium (WASC) (Steinke et al., 2011) are taken into consideration to design and develop solutions. The following subsections discuss a list of the top web threats as defined by the two web security resources mentioned earlier. The Common Weakness Enumeration (CWE) Top 25 Most Dangerous Software Errors is also taken into consideration to design and develop solutions.

3.9.1 Open Web Application Security Project (OWASP)

The Open Web Application Security Project (OWASP) is a nonprofit foundation that operates to enhance the security of web-based software applications. OWASP is one of most reputed authorized in web application security. OWASP advises addressing application security as a challenge, including people, processes, and technology, since the most successful approaches to application security need improvements in these areas (OWASP Top 10, 2017). Since 2003 approximately every three years, OWASP publishes a report called the OWASP Top 10.

The report documents a ranking of the ten top security vulnerabilities, threats, and risks to make people aware of application security by highlighting specific vulnerabilities and critical threats encountered during the development of web applications (Sadqi & Maleh, 2020). Currently, OWASP classification is the de facto industry standard in the web applications security context. The OWASP website supplies every risk with information regarding the likelihood and impact of the threat additionally provides resources to mitigate each threat. Below Table 4, lists the top ten threats and risks for year 2021.

Table 4. Top ten critical web application security threats and vulnerabilties (OWASP Top 10, 2021)

ID AND THREAT	DESCRIPTION
A01:2021 Broken Access Control	In web applications, Restrictions to only authorized users are allowed to use is not enforced systematically exposes this threat (OWASP Top 10, 2017). Failures commonly result in unauthorized information disclosure, alteration of data, complete data loss, or the performance of a business function outside the user's capabilities (OWASP Top 10, 2021).
A02:2021 Cryptographic Failures	Generally, IIoT Applications and API do not secure sensitive data in transit or stored. For example: - Health records, personnel information, entrepreneurial secrets, and any data which falls under privacy laws and regulations need extra security using encryption. Previously OWASP addressed this as Sensitive Data Exposure (OWASP Top 10, 2021).
A03:2021 Injection	SQL injection, NoSQL injection, OS injection, and LDAP injection are examples of injection threats, which occur when untrusted data or unauthorized data formats are transferred to an interpreter as part of a command or query (OWASP Top 10, 2021).
A04:2021 Insecure Design	Insecure design is a broad term that encompasses a variety of flaws and is defined as missing or ineffective control. It cannot be resolved by a flawless implementation of required security controls that were not developed to secure specified threats (OWASP Top 10, 2021).
A05:2021 Security Misconfiguration	The most commonly seen threat is Security misconfiguration, the resultant of Insecure defaults, incomplete or ad hoc configurations, open cloud storage, misconfigured protocol (http) headers, and logs containing sensitive data (OWASP Top 10, 2017).
A06:2021 Vulnerable and Outdated Components	Web applications software components such as API, framework, libraries, and modules execute with the same permissions. Suppose components vulnerabilities are exploited cyberattacks result in significant information loss or server hacking. Applications, APIs, and its

	components with known weaknesses degrade security with several attacks (OWASP Top 10, 2017)
A07:2021 Identification and Authentication Failures	Validation of the user identity, authentication, and session management is crucial to secure against authentication-related threats and attacks (OWASP Top 10, 2021). If session verification is enforced erroneously allows attacks to steal passwords, session keys, and other application users' identities (OWASP Top 10, 2017).
A08:2021 Software and Data Integrity Failures	It is a Threat related to application code, libraries, modules, and environment Failures to secure against integrity violations. An insecure Continuous Integration or Continuous Development of application modules can add malicious code, unauthorized permissions or access, and errors (OWASP Top 10, 2021).
A09:2021 Security Logging and Monitoring Failures	Security logging and monitoring failures, along with a lack of or inefficient integration with incident response, enable attackers to extend their assault, retain persistence, pivot to other systems, and hack, steal or delete data. In most cases, security breaches are undetected or detected after several days (OWASP Top 10, 2017).
A10:2021 Server-Side Request Forgery	SSRF Threats occur when the web application is accessing remote URLs or resources without verifying. It allows attackers to drive the application by sending a framed request to an unauthorized destination. Modern web applications are accessing remote server URLs more, which has resulted in increasing SSRF (OWASP Top 10, 2021).

3.9.2 Web Application Security Consortium threat classification

The Web Application Security Consortium (WASC) is a non-profit organization that consists of a group of international experts, industry practitioners, and organization representatives dedicated to developing open-source and generally accepted best-practice security standards for the World Wide Web (WASC Threat Classification, 2010). The WASC developed the WASC Threat Classification resource to identify and categorize web threats and vulnerabilities (WASC Threat Classification, 2010). The WASC maintains a Web Hacking Database Incidents (WHID), which collects statistical data of web application security incidents (Steinke et al., 2011). WASC classifies 49 distinct

web-related threats and divides them into three different methodologies or perspectives (WASC Threat Classification, 2010): Enumeration View, Development Phase view, and Cross-reference View (Sadqi & Maleh, 2020).

Table 5. WASC Enumeration view:List of Web application threats and vulnerabilties

ID and ATTACK	DESCRIPTION
WASC-42 Abuse of Functionality	Abuse of Functionality is a threat that exploits a web application's features and functionality to target applications. It is the abuse of application functionality intended to execute the unwanted outcome (WASC Threat Classification, 2010).
WASC-11 Brute force	A brute force attack is a technique for determining an unknown value by processing a wide range of potential values through an automated procedure (WASC Threat Classification, 2010).
WASC-7 Buffer Overflow	Buffer Overflow is a threat that occurs when more data is written to a block of memory or buffer than the buffer can store (WASC Threat Classification, 2010).
WASC-18 Credential/ Session Prediction	It is a technique for hijacking or spoofing a web application user. In this technique, the attacker uses the threat to determine or estimate the unique value to identify a given session or user uniquely (WASC Threat Classification, 2010).
WASC-12 Content Spoofing	It is threat that enables an attacker to inject a malicious payload into a web application and then disguise it as legitimate content of application (WASC Threat Classification, 2010).
WASC-8 Cross-Site Scripting	Cross-site Scripting (XSS) is a threat used by the attacker that includes the attacker's code replicated into the user's browser instance or web client application (WASC Threat Classification, 2010).
WASC-9 Cross-Site Request Forgery	A cross-site request forgery attack is one that includes coercing a client into sending an HTTP request to a target location without their knowledge or consent in order to conduct an action on their behalf (WASC Threat Classification, 2010).

WASC-10 **Denial of Service**	A denial of service (DoS) attack is a threat that aims to prohibit a website from providing regular user activity. DoS attacks, commonly used at the network layer, may also be used at the application layer. These harmful assaults may be successful by depleting a system's vital resources, exploiting a vulnerability, or abusing functionality (WASC Threat Classification, 2010).
WASC-37 **Session Fixation**	Session Fixation is a form of attack that forcibly modifies the client's session ID. Attackers can use several techniques to "fix" the session ID value, depending on the functionality of the destination website (WASC Threat Classification, 2010).
WASC-28 **Null Byte Injection**	Null Byte Injection is a method for circumventing integrity checking filters in web application by injecting URL-encoded null byte characters (i.e., percent 00, or 0x00 in hex) into user-supplied data (WASC Threat Classification, 2010).
WASC-36 **SSI Injection**	SSI Injection (Server-side Include) is a server-side exploit method that enables an attacker to inject code into a web application, which the webserver will then execute locally. SSI Injection takes advantage of a web application's inability to properly sanitize user-supplied data before its inclusion in a server-side interpreted HTML file (WASC Threat Classification, 2010).
WASC-19 **SQL Injection**	SQL Injection is a kind of threat used by attacked to take advantage of programs that generate SQL statements from user-supplied data. When an attacker succeeds, the attacker can modify the logic of SQL queries run against the database (WASC Threat Classification, 2010).
WASC-39 **XPath Injection**	XPath Injection is a kind of Threat used by the attacker that targets web applications that generate XPath (XML Path Language) queries from user-supplied data to query or explore XML documents (WASC Threat Classification, 2010).
WASC-23 **XML Injection**	XML Injection is a method of altering or undermining the logic of an XML application or service (WASC Threat Classification, 2010).
WASC-46 **XQuery Injection**	XQuery Injection is a subset of the traditional SQL injection attack that targets the XML XQuery Language (WASC Threat Classification, 2010).

Enumeration view lists the threats, attacks and vulnerabilities that can breach a web application, its data, and its users. Development phase view lists the vulnerabilities probably introduced during the application development life cycle: the design, implementation, and deployment phases (*WASC Threat Classification*, 2010). The Cross Reference view procedure maps WASC threats and vulnerabilities with other web security resources. The benefit of this categorization is that it creates a comprehensive list of all web-based attacks and vulnerabilities (Sadqi & Maleh, 2020). For the Industrial organization network, considered the Enumeration view list to examine potential threats related to IIoT web application and described above in Table 5.

In software development life WASC Threat classification: Development view is considered the to examine potential vulnerabilities related to IIoT web application and introduced development phase is presented in Table 6. In Software Development Lifecycle there are three phases: Design, Implementation and Deployment phase. In Table 6,the Design Phase addresses vulnerabilities that are likely to arise as a result of a lack of countermeasures specified in the software design and requirements, or a poorly or inadequately described design and requirement. The Implementation phase addresses the vulnerabilities added due to flawed implementation and deployment phase address the vulnerabilities added due to application or server misconfiguration and insufficient deployment (WASC Threat Classification, 2010).

Table 6. WASC development view list of vulnerabilties (*WASC Threat Classification*,2010).

Vulnerability	Design	Implementation	Deployment
Abuse of Functionality	X		
Application Misconfiguration		X	X
Brute Force	X	X	
Buffer Overflow		X	
Content Spoofing		X	
Credential/Session Prediction		X	
Cross-Site Scripting		X	
Cross-Site Request Forgery	X	X	
Denial of Service	X	X	

Directory Indexing			X
Improper Filesystem Permissions		X	X
Improper Input Handling		X	
Improper Output Handling		X	
Information Leakage	X	X	X
Insecure Indexing		X	X
Insufficient Anti-automation	X	X	
Insufficient Authentication	X	X	
Insufficient Authorization	X	X	
Insufficient Password Recovery	X	X	
Insufficient Process Validation	X	X	
Insufficient Session Expiration	X	X	X
Insufficient Transport Layer Protection	X	X	X
Null Byte Injection		X	
Remote File Inclusion (RFI)		X	X
Routing Detour			X
Server Misconfiguration			X
Session Fixation		X	X
SQL Injection		X	
URL Redirector Abuse	X	X	
XPath Injection		X	
XML Injection		X	
XQuery Injection		X	

3.10 IIoT web application attacker path to produce security risk and Impact

Concerning security, the IIoT's primary challenges include data protection, privacy protection, authentication, and control of access to servers and information management systems, all of which are affected by the scarcity of computing resources and their intrinsic autonomy (Rubio et al., 2018). In the IIoT context, threats, and attacks impact application content, including enterprise information, physical domains, and IoT devices. Additionally, unauthorized access and modification

of IoT platforms, software, and firmware are also a potential threat (Bhattacharjee, 2018). IIoT equipment and its control systems are also exposed and exploited to physical reliability and safety threats.

Generally, the Application security risk is a threat posed by attackers who use various unlawful methods to impact the company and inflict damage on organizations. Currently, IIoT Web applications are attacked by attackers using different paths to damage the business of manufacturing and service industries (OWASP Top 10, 2017). Below Figure 8, illustrated the process path attacker exploit the vulnerabilities in the IIoT web application to perform an attack to cause impact and damage to the business of manufacturing and service industries.

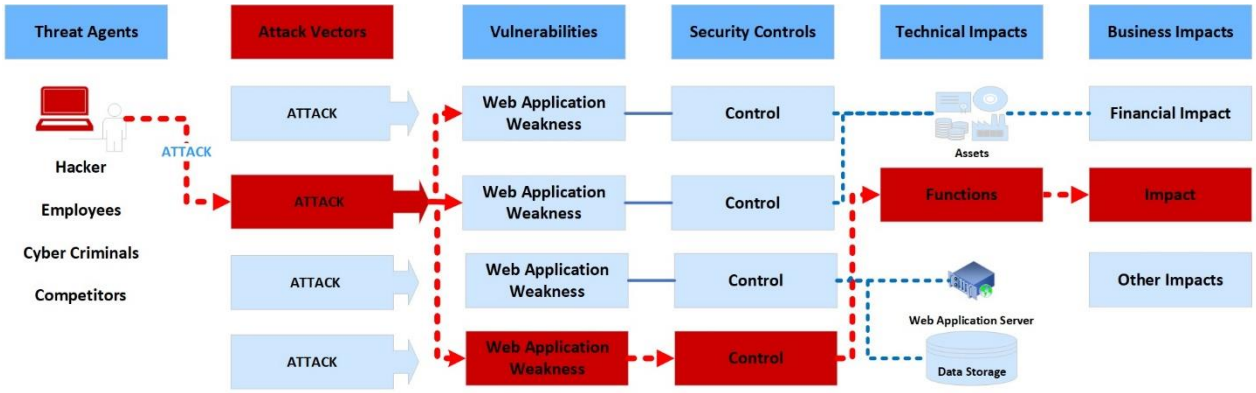

Figure 8. Web application attacker process path

According to OWASP Top 10 (2017), the aspects determining actor selection path as shown in Figure 8: Threat agents, attack vectors, application security vulnerabilities, security controls, technical impacts, and business impact. Threat Agents refer to any individuals capable of posing a threat to the system, such as hackers, competitors, employees, cybercriminals, and others. Attack vectors refer to the threat agent's potential to access the organization's assets, IIoT application system, or resources (Sadqi & Maleh, 2020). Security vulnerabilities are faults or issues within IIoT web applications. Security control refers to the defenses or fixes implemented to secure or patch vulnerabilities. The technical impact refers to the outcome of the initiated attack on web applications, organizational assets, web application servers, data, and web application functionalities. Business impact indicates the potential damage to IIoT web application, its company, and application deployed enterprise.

3.11 Web applications security analysis methodologies (Testing)

Security of web applications system and deployed network infrastructure is critical to IIoT company internet existence's success. Web application consists of three layers Client interface, web server application, and Backend Database, as shown in Figure 3. Several IIoT web application security testing tools are available to support developers in reducing application vulnerabilities during the design and development phases of SSDLC (Kapodistria et al., 2011). While various tools called penetration tools or web vulnerabilities scanners are also available to help web application security testers identify the weakness during the deployment and maintenance of web applications in use in IIoT companies' networks (Kapodistria et al., 2011).

In the context of SSDLC of the web application, there are now standard security testing and step by step analysis methodologies to build techniques to measure web application security issues and provide a comprehensive view of web application security (Correa et al., 2021). According to OWASP projects and Correa et al. (2021), the standard security analysis methodologies to assess web application security, including OWASP Top 10 Threats and WASC Threat Classification views, are the OWASP Application Security Verification Standard (ASVS), OWASP Web Security Testing Guide, and Penetration Testing Execution Standard (PTES). The following subsections briefly explain the application of security analysis methodologies.

3.11.1 OWASP Application Security Verification Standard (ASVS)

According to OWASP Application Security Verification Standard 4.0.3 (2021), The ASVS is a community-driven project to build a framework of security requirements and controls for designing, implementing, and testing contemporary web applications and web services (p. 9). The objective of ASVS has two primary purposes, to assist enterprises in developing and maintaining secure web applications; and facilitate the coordination of security service providers, security tool providers, and customers' expectations and offers (OWASP Application Security Verification Standard 4.0.3, 2021).

OWASP Application Security Verification Standard 4.0.3 (2021) defines three security verification levels: ”1) ASVS Level 1 is for applications with low assurance levels and is comprehensively penetration testable .2) ASVS Level 2 is for applications that need to protect sensitive data and is the

most recommended level for most applications. 3) ASVS Level 3 is for the most critical applications that perform high-value transactions, require the highest level of trust, or contain sensitive medical data. (p. 11).In this research thesis ASVS Level 2 is used for creating an innovative solution. ASVS Level 2 is for applications that process health data, sensitive industrial and business data. Application Security Verifications Standard to valid Level 2 Applications is presented in below Table 7

Table 7. OWASP Application Security Verification Standard 4.0 for Level 2 Applications

Application Phases	Security Review/Testing Methodology
Building	Security Architecture and Review
	Secure Coding
	Standards and Checklist
Building, Configuration, Deployment, Assurance and Verification	Secure and Peer Code review
	DevSecOps (Development, security, and operations)
	Unit and Integration Tests
Assurance and Verification	Hybrid Penetration Tests
	Static application security testing (SAST):- Technology to analyze source code, byte, binaries to identify security vulnerabilities offers (OWASP Application Security Verification Standard 4.0.3,2021).

3.11.2 OWASP web security testing guide

The OWASP Web Security Testing Guide is the standard framework used by industrial experts to assess the security of web applications and web services. OWASP Web Security Testing Guide Version 4.2 (2020) defines its aim as "to help people understand what, why, when, where, and how to assess web applications" (p.11). The testing guide describes The broad testing framework and the strategies needed to enforce the framework in practice. The OWASP Web Security Testing methodology defines two reference frameworks: the OWASP testing framework and the Web Application Security Testing framework (OWASP Web Security Testing Guide Version 4.2, 2020). The first reference framework is composed of processes and tasks relevant to the five phases of the SSDLC,

as explained in Section 3.7. These testing framework activities that should happen is listed in Table 8.

Table 8. OWASP Web Security Testing Standard 4.2 Testing Framework.

SSDLC Phases	Security Review/Testing Methodology		
Before development begins	Review SDLC process	Policy Review	Standards Review
During definition and design	Requirements Review	Design and Architecture Review	
	Create and Review Threat Models	Create and Review UML Models	
During development	Code Review	Code Walk Throughs	Unit and System Tests
During deployment	Penetration Testing	Configuration Management Review	
	Acceptance tests	Unit and System Tests	
During maintenance and operations	Chance verification	Operational Management Review	
	Health checks	Regression Tests	

The second OWASP reference framework is web application security testing consists of techniques to detect different security problems of web applications (Correa et al., 2021). The testing framework is a methodology to verify and validate the application security controls to evaluate the system or network security (OWASP Web Security Testing Guide Version 4.2, 2020). The framework for web application security testing splits testing into passive and active (OWASP Web Security Testing Guide Version 4.2, 2020). In passive testing, the tester tries to comprehend the application's logic and explore it from the user's perspective. Next, in Active testing, a tester uses the below 12 categories of the batch of tests.

1. “Information Gathering”
2. “Configuration and Deployment Management Testing”
3. “Identity Management Testing”
4. “Authentication Testing”
5. “Authorization Testing”
6. “Session Management Testing”
7. “Input Validation Testing”

8. “Error Handling”
9. “Cryptography”
10. “Business Logic Testing”
11. “Client-side Testing”
12. “API Testing”

3.11.3 Penetration Testing Execution Standard (PTES)

PTES is a standardized set of methods used to conduct penetration testing analysis by security experts to address the complete need for software intrusion testing (Correa et al., 2021). The PTES Technical Guidelines provide practical advice on testing methodologies and suggest security testing tools (OWASP Web Security Testing Guide Version 4.2, 2020). PTES standard framework defines penetration testing into seven phases listed in below:

1. “Pre-engagement Interactions”
2. “Intelligence Gathering”
3. “Threat Modeling”
4. “Vulnerability Analysis”
5. “Exploitation”
6. “Post Exploitation”
7. “Reporting”

3.12 Web application security auditing

An audit is a systematic evaluation of compliance with a well-defined standard by an independent expert (Pompon, 2016). Information Systems Audit and Control Association (ISACA) has standardized the auditing process in ISACA’s Information Systems Auditing: Tools and Techniques—Creating Audit Programs (2016) document .

According to ISACA’s Information Systems Auditing: Tools and Techniques—Creating Audit Programs (2016),the information systems auditor gathers evidence, assesses the weaknesses and strengths of internal controls based on data from audit tests, and compiles an audit report that shows flaws objectively and recommends how to fix them, all in the course of the audit processes. ISACA’s auditing process is divided into three distinct phases: planning, fieldwork, and reporting. Web application Security Auditing framework will be designed based on ISACA Audit Process Steps by Phase presented in Figure 9.

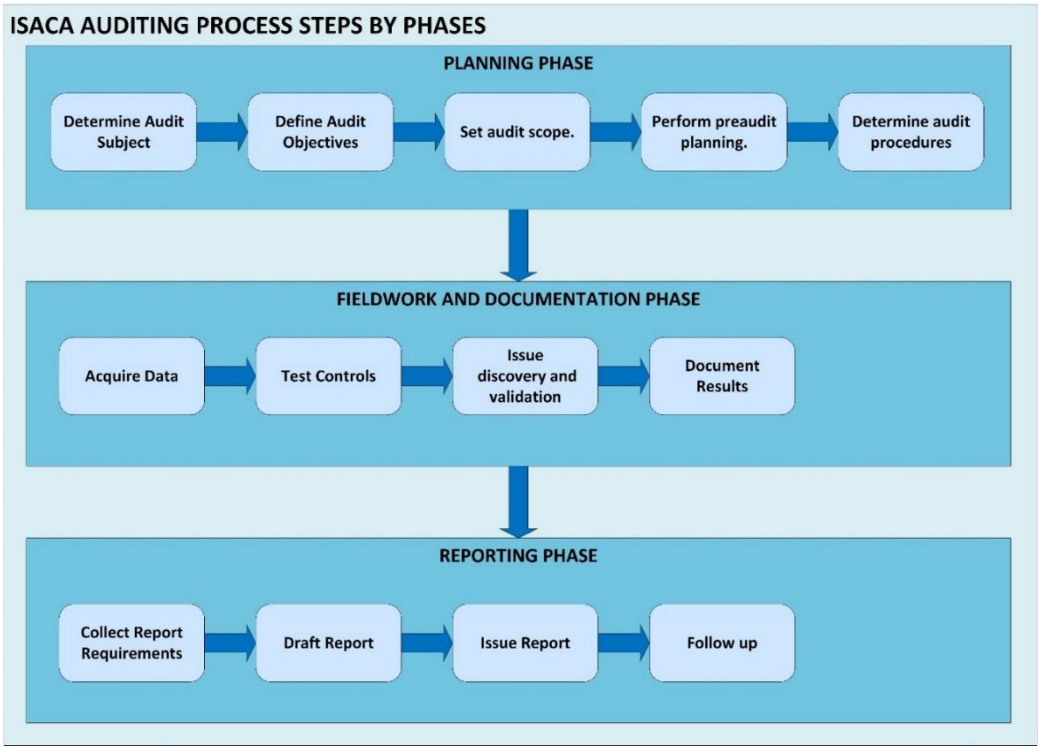

Figure 9. ISACA Three Phased Auditing Process

3.13 General Data Protection Regulation (GDPR)

3.13.1 Personal data

Presently, Industrial employees' and consumers personnel data is stored in data and cloud servers in the IIoT web applications system based on the application's requirements and use cases. According to the European Union (EU) Regulation (EU) 2016/679 (2016) personnel data is defined as "any information relating to an identified or identifiable natural person (hereinafter: data subject); an identifiable person is one who can be identified, directly or indirectly, in particular by reference to one or more factors specific to his physical, physiological, mental, economic, cultural, or social identity (Skendzic et al. 2018.)". Currently, any global IIoT web applications organization storing and using EU citizens' personnel data in Finland must process following the EU's GDPR Regulation (EU) 2016/679 and the Finnish Data Protection Act (1050/2018) to protect personnel data (Tsochev, 2020; Data Protection Act (1050/2018),2018).

3.13.2 GDPR data Controller, data Processor and data Subjects

According to GDPR, the data controller is the IIoT web application organization that decides what is accomplished with personnel data, why, and how it is done (Colesky et al., 2019). When other organizations process the personnel data on behalf of the data controller, GDPR refers to it as a data processor (Colesky et al., 2019). The individual about whom the data controller processes data is called the data subject (Colesky et al., 2019). All the users of IIoT devices and applications personnel data are data subjects.

3.13.3 GDPR purpose and focus areas for software applications

The GDPR's primary goal is to safeguard personal data through organizational, administrative, and technological measures and protect the fundamental rights of natural persons related to application processing personal data.

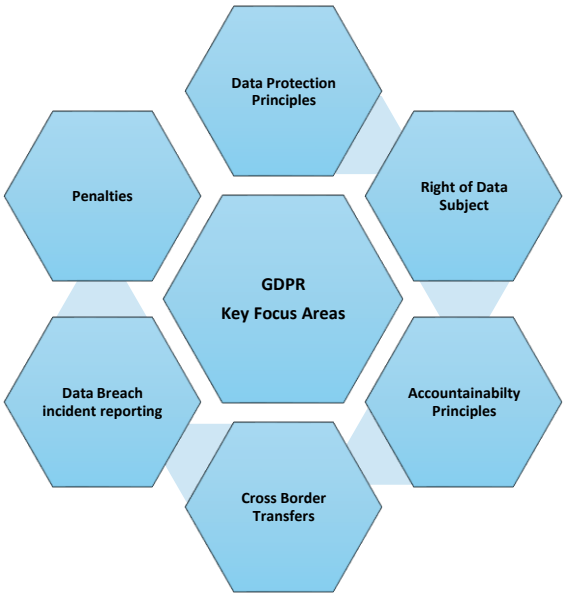

Figure 10. Key GDPR focus areas for IIoT Application companies

According to GDPR, IIoT web application organizations are primarily responsible and accountable for the storage, security, and erasure of personal data, and data processors are responsible depending on an agreement with the data controller. The regulation defines the purpose, permission to store, erase and protect. The GDPR sets no explicit criteria for software design, but the regulation prescribes particular data processing principles and obligations that an organization needs to

follow (Colesky et al., 2019). IIoT web applications collect and process personnel data. Organizations must focus on implementing GDPR as stated in Finland's Data Protection Act (1050/2018) and EU Regulation (EU) 2016/679. The Figure 10 illustrates the key areas of GDPR application on which IIoT developing companies must focus. According to Regulation (EU) 2016/679 (2016), the data protection and processing principles need to be implemented by organizations designing and developing secure IIoT web applications is listed in Table 9.

Table 9. EU Regulation (EU) 2016/679 personal data protection principles for application

PRINCIPLES	DESCRIPTION
Lawfulness of processing	IIoT organizations must have legal reasons to collect and process personal data. Data Collection and processing should not affect the data subject (Regulation (EU) 2016/679, 2016).
Consent' of the data subject	IIoT organizations must collect agreement to the processing of personal data from data subjects (Regulation (EU) 2016/679, 2016).
Purpose Limitation	An IIoT organization is not allowed to apply new data subject processing if it is incompatible with the initial purpose (Regulation (EU) 2016/679, 2016).
Data Limitation	IIoT Organizations should collect only the personal data necessary to accomplish the declared purpose of their processing (Regulation (EU) 2016/679, 2016).
Accuracy	The IIoT organization is accountable for updating, correcting, and ensuring personal data is valid (Regulation (EU) 2016/679, 2016).
Storage Limitation	IIoT organizations are not permitted to store the data subject's data for a longer duration than is necessary for the reason it is collected (Goncalves & Correia, 2019).
Integrity and confidentiality	IIoT organizations are responsible for safeguarding personal data from threats such as destruction, accidental loss, or illegal data processing (Goncalves & Correia, 2019).

3.13.4 Data Protection Impact Assessment (DPIA)

The DPIA is one of the GDPR's particular processes. Multiple organizations compile to DPIAs, and in many cases, an organization may find the procedure beneficial even if the regulation does not mandate it. DPIAs assess the risks to personal data caused by processing operations (IT Governance Privacy Team, 2020). To ensure security and avoid processing in violation of GDPR, the organization as controller or processor should assess the risks associated with the processing and use risk-mitigation measures. Several European authorities have published documentation on DPIAs. The approaches are equivalent, with few minor variations according to local legislation and historical data protection practices. Additionally, the GDPR specifies what must be in a DPIA at a minimum level.

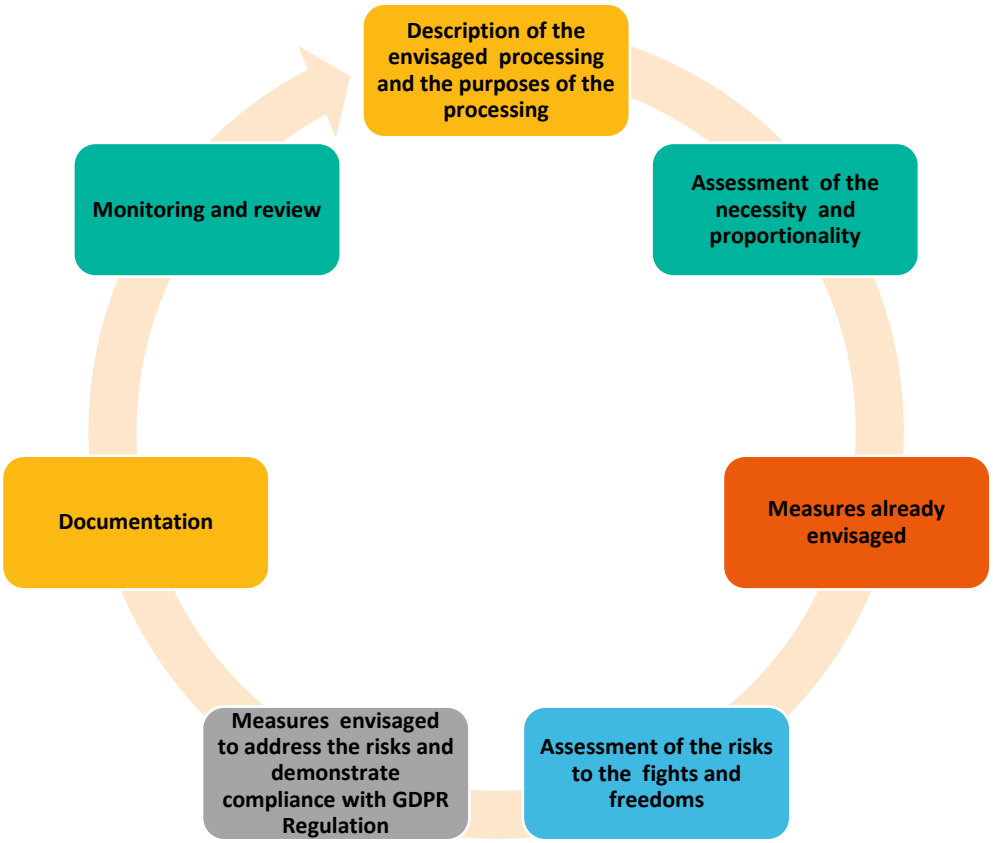

Figure 11.DPIA iterative process GDPR setting minimum features

According to Guidelines on Personal Data Breach Notification under Regulation 2016/679 (wp250rev.01) (2017), the generic iterative process to conduct DPIA (GDPR setting minimum features of a DPIA) is shown in Figure 11. All the necessary standards set out in the GDPR give a

wide, general framework for establishing and carrying out a DPIA procedure. The actual execution of a DPIA will rely on the requirements laid out in the GDPR which may be complemented by more comprehensive practical instructions. The DPIA implementation is hence scalable (Guidelines on Personal Data Breach Notification under Regulation 2016/679 (wp250rev.01), 2017) This implies that even a small data controller may develop and implement a DPIA that is suited for their processing activities. the DPIA under the GDPR is a mechanism for managing threats to the rights of the data subjects, and thereby adopts their viewpoint, as is the case in particular domains

3.14 Secure data transmission and source authentication

Securing network connections and data transfers is crucial for IIoT web applications to accomplish their security objectives of integrity, confidentiality, and authentication. To achieve security objectives related to secure network connection and data transmission, studying theoretical concepts secure transfer protocols, user authentication and data encryption is required.

3.14.1 Secure network connection and data transfer protocols

In modern client and server architecture-based web applications, web server application enables HTTP protocol over Transport Layer Security/Secure Socket Layer (TLS/SSL) called HTTPS protocol to secure network connection and data transfer. Figure 4 shows remote IoT devices and a client application running on browsers, mobiles, or desktops connecting to the server and then transferring data in encrypted format using HTTPS protocol upload and download data.

The Web server communicating to the client web application using the HTTPS protocol listens on port 443. Figure 12 presents user authentication and secure data transfer between web application client and server using HTTPS protocol. Both HTTPS and TLS/SSL depend on a client-server X.509 digital certificate for authentication (Cross, 2007).The web application uses HTTPS protocol for secure data transmission over the internet and in an encrypted format. As the data is encrypted, it protects data against eavesdroppers, and man-in-the-middle attacks, else might result in unauthorized users gaining access to the data (Cross, 2007). HTTPS protocol is also used between web server as client and cloud database as server.

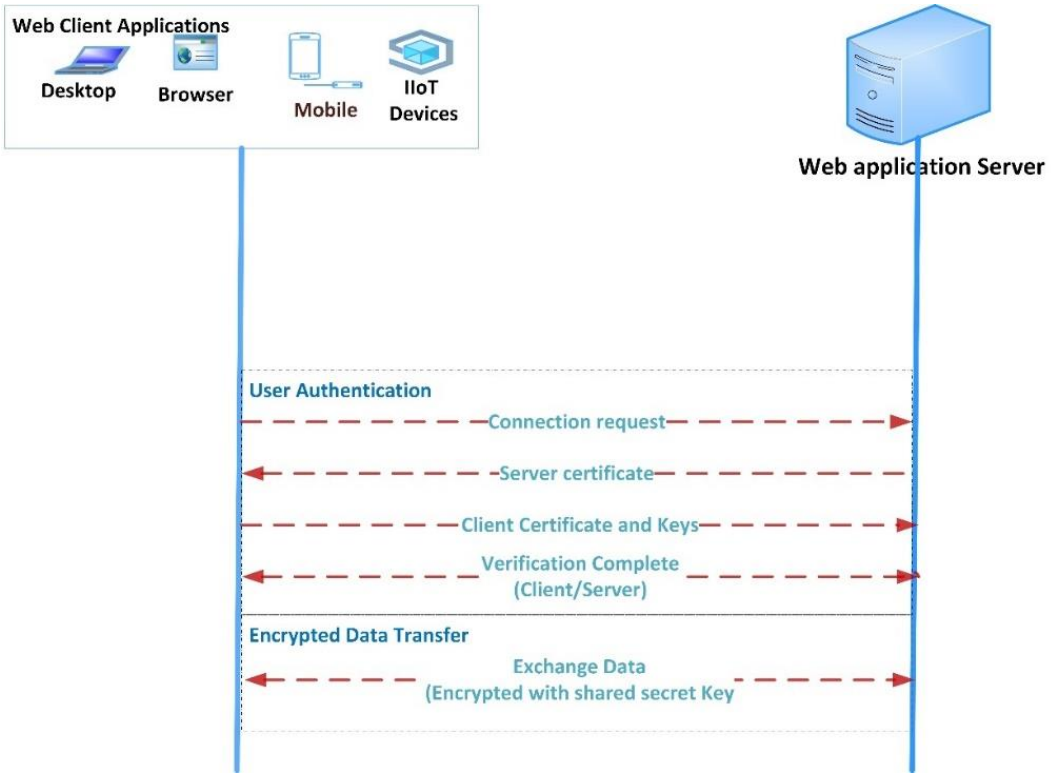

Figure 12. Client-Server Authentication and Data Transfer using HTTPS protocol

3.14.2 Client authentication

Authentication serves as the first line of defense against identity theft or fraud by allowing only authorized users to access web application services (Olanrewaju et al., 2021). Authentication is a term that refers to the process of verifying a user's identity and is part of HTTPS. If the authentication system is insecure, it may be hacked, resulting in significant negative consequences for the firm (Olanrewaju et al., 2021). Authentication decreases the likelihood that a man-in-the-middle attack will succeed in obtaining unauthorized access to data (Mueller, 2015).

Earlier web applications protected web servers with effective authentication mechanisms that enabled user-friendly operations. These methods included PINs or passwords (Olanrewaju et al., 2021) and are composed of two steps: identification and confirmation. Web applications are also utilizing multi-factor authentication. For the client application, multiple authentications use username and password to log in first, and after confirmation in the second stage, the user uses a mobile authenticator-generated random number (McDonald, 2020). Client application registers mobile authenticator during first-time signup to a web server (McDonald, 2020).

Nowadays, web applications clients are using encryption keys and digital certificates, adding a layer of authentication as credentials to identify. Modern Web applications use X.509 digital certificate-based authentication, in which trusted authorities issue server CA (Certificate Authority) certificates and client device certificates to validate client application devices and web servers (Olanrewaju et al., 2021). Server CA certificates installed on the web application server and device certificates on client application devices. User Authentication shown in Figure 12 illustrates the authentication process, where certificates and client application information are exchanged between server and clients over HTTP/TLS protocol and validated to finish authentication.

3.14.3 Data encryption

Data encryption is hardware or software-based algorithm in which the system encrypts data using public and private keys to secure it from unauthorized and unintended users (Chavan & Tamane, 2020). There are various of set of cryptographic algorithms called a cipher suite used for encrypting data and a few popular to name RSA(Rivest–Shamir–Adleman), AES (Advanced Encryption Standard), DES (Data Encryption Standard), ECC (Elliptic-curve cryptography), SHA-256 (Secure Hash Algorithm-256), and others.

Modern web applications generate public and private encryption keys from trusted authorities. Using generated keys, certificate signing requests (CSRs) that contain the public key and domain and upload the request to the certificate authority to obtain server and client digital certificates. Figure 12 illustrates encrypted data transfer between client and server. During data transmission over HTTP/TLS network connection, the encrypted data using keys is used to transfer data securely.

4 Implementation and evaluation

4.1 Practical problem and research activities

Securing and protecting its IIoT web application deployed in industrial production and service networks against cyberattacks and network breaches is the practical challenge faced by the host company and is addressed by this research by providing an innovative solution. Currently, the host company does not have any process and framework defined to evaluate the security of the designed and developed IIoT web application, which impacts customer's business if web application issues affect its industrial customer's systems and network security. The research methodology uses the constructive research approach described in chapter 2.2 and steps illustrated in Figure 1.

The research host company has already defined the starting point of the constructive research method, which is identifying and selecting research issues. The host company needs an internal security auditing and testing framework to evaluate the security of the IIoT web application against cyber-attacks and vulnerabilities. Additionally, the hosting company requires a solution for developing and verifying web applications that comply with EU and Finnish GDPR regulations, along with mechanisms for securing data transmission between web application clients, servers, and data storage devices or the cloud. The following selection of choosing a research problem is to formulate the research question documented in chapter 2.4: which complexity-based strategies, techniques, and frameworks (potential constructs) use and support collaborative security standards processes.

The second research phase of the constructive research approach builds on the pre-understanding necessary for the targeted constructs of innovative solutions. The pre-study based on the literature from several academic sources, security standards, frameworks, and expert recommendations provide the research with a thorough understanding of the research problem and its context accomplished in chapter 3. The knowledge of academic researchers, experts, and security standards influenced my interpretation of the IIoT web application security and related contexts and concepts. In this phase, performed a systematic literature review to learn about the existing standard solutions, auditing, and testing frameworks for the stated research question.

The systematic literature review data included academic research papers, state regulations, books written by industrial security experts, security standards, and experts' recommended guidelines published between 2005 and 2021, including OWASP, WASC, IEEE, ISO standards, International Journal of Automation and Computing, Finland, and EU data regulations, ISACA's information systems auditing guidelines. This literature review examined web application security-related problems and existing solutions by evaluating IIoT web application system architecture, SSDLC, security standards, security threats, vulnerabilities, attack paths, standard security auditing, and testing procedures. Subsequently, in the innovative design construct phase, the research study retrieved information from this systematic literature review to conduct auditing and testing on the IIoT web application utilized to construct an auditing framework and guidelines comprehending the existing solutions.

4.2 Designing the IIoT web applications security analysis constructs

In this design construct phase, the constructs are the outcome-based of the pre-understanding study of web application security-related concepts and the target company's problem-specific needs. This chapter presents created collaborative process and framework to be conducted a security analysis of IIoT web applications before deploying to the customer's industrial network by the application developing company.

Based on the research problem requirements of the host company web application related to the lack of a suitable process and framework for collaboration-based security analysis of IIoT web application, there is a need for including the security standards and expert recommendations in web application security auditing, testing, identifying, and mitigating the threats and vulnerabilities. The solution to this research problem is to address how to integrate the existing solutions, standards, and recommendations by defining the security auditing and testing process and framework. Two auditing and testing constructs recommended making progress correspond to the above needs. The constructs include:

1. IIoT Web applications security design and auditing principles checklist
2. Integrated security analysis framework for IIoT Web Applications (includes GDPR and secure data transfer auditing guidelines)

4.3 IIoT web applications security design and validation principles checklist

The first innovative solution (Construct 1) is IIoT Web applications security design and auditing principles checklist that need to be implemented and audited by the host company at various phases of the web application's SDLC. This chapter lists the defined security principles that need to be implemented and internally audited by the IIoT web application developing organization. Secure web application development principles, Secure web application maintenance principles, client-side web application protection principles, server-side web application protection principles, server-side network infrastructure protection principles, server-side architecture protection principles, server-side data backup protection principles, and server-side monitoring and incident reporting are classifications of the defined security principles for designing and validating IIoT web applications.

Table 10. Secure web application development and maintanance principles

Secure web application development principles checklist	
1	**Identify and create a threat model for the IIoT web application based on the requirements:-** Developing a secure web application is impractical without identifying potential security threats. After identifying potential threats Implementing security is the principal approach (Tanaskovic & Zivkovic, 2011).
2	**Implement and conduct a technical code review using standards before integrating into the repository:-** Using secure programming techniques is important for developers to make sure their solutions are safe and resistant to hacker attacks (Tanaskovic & Zivkovic, 2011).
4	**Conduct security scans on the source code of web applications using static code analysis tools** (OWASP Web Security Testing Guide Version 4.2, 2020).
3	**Validate client application manages sensitive data**:- Client application manages data both in local user system or device memory and on the network (Tanaskovic & Zivkovic, 2011).
4	**Proper Exception handling :-** Ensure application handles exception and recover from errors to avoid sensitive and technical data leakages (Tanaskovic & Zivkovic, 2011).
5	**Modularize the application into separate modules:-**Reduces the potential damage in case the attacker downloads the application by preventing an attacker from access to the network or other resources is restricted (Tanaskovic & Zivkovic, 2011).

6	**Remove or Disable unused software modules and codes in web application:-** This reduces the attack
Secure web application maintenance principles checklist	
1	**Patch and Update process :-** Within a few hours after identifying a new vulnerability, the application development organization releases a solution and ensures it protects the server through patch and update processes (Tanaskovic & Zivkovic, 2011).

Table 11. Client-side application protection principles

Client-side web application protection principles checklist	
1	**Encrypted data transfer:-** Use TLS version 1.2/1.3 based HTTPS protocol and digital certificates to protect client application users from network attacks and secure data transfer .This ensures all connections and data transfer between Client application users and web application server are encrypted(Vinod et al., 2008).
2	**Input Validation :-** check if the input received by client application is valid and secure (OWASP Top 10, 2021).
3	**Enforce encryption uses HTTP Strict Transport Security (HSTS) directives:-** Use HSTS directives on the application server to encryption server connection and to protect client application users from SSL stripping attacks and middleman attacks:-Through the use of a specific response header, the HSTS feature enables a web application to warn the client that it should never create an unencrypted HTTP connection to the specified domain servers (OWASP Top 10, 2021).
4	**Implement global standard security risks protection mechanisms in client applications:-** Protect client use against WASC Threat Classification, OWASP top 2017 and 2021 vulnerabilities by implementing the protection mechanisms. According to OWASP top (2021), Safe generation of HTML, using JavaScript safely, implementing an effective Content Security Policy to protect against XSS (Cross-Site Scripting) and other vulnerabilities the client application users such as browsers and mobile applications. For example:- Never use untrusted data without verifying in JavaScript and minimum or generate HTML safely to protect clients against XSS vulnerabilities (OWASP Top 10, 2021).
5	**Use Client certificates and Universal 2nd Factor Authentication (U2F)** :-This protects application users from phishing attacks. Suppose Client users' credentials are stolen, then protect

	against such attacks with U2F tokens and client certificates protect client application users against the attacker having server application credentials(Cross, 2007).
6	**Never include sensitive information in a URL:-**A URL cannot be encrypted or secret and it is saved in the client application browser or mobile application history. Never send sensitive data in the URL(Cross, 2007).

Table 12. Server-side application protection principles

Server-side web application protection principles checklist	
1	**Input validation:-** validating the input transmitted by client application users to protect against several vulnerabilities (OWASP Top 10, 2021).
2	**Implement application Authorization mechanisms:-** Implement the OAuth authorization protocol supporting Multi-Factor Authentication (MFA) to verify the client application user's authenticity(Cross, 2007).
3.	**Implement strict access controls** :-To prevent unauthorized access to server data or functionality.
4	**Use security-tested software libraries and tools** :-To prevent open-source vulnerabilities.
5	**Servers make safe queries:-**Servers should make safe database queries to avoid SQL injection vulnerabilities (WASC Threat Classification, 2010).
6	**Implement global standard security risks protection mechanisms in server applications:-** Protect server applications against WASC Threat Classification, OWASP Top 2017 and 2021 vulnerabilities by implementing expert-recommended protection mechanisms.
7	**Never use Internal resources for untrusted and unverified contents:-**Server applications should not use network filesystems and storage systems for untrusted and unverified content uploads.
8	**Prevent dynamic code execution on the server** :-To avoid remote code execution vulnerabilities. Protects server application and network resources from vulnerabilities like code injection, remote code execution, malware, and bots(OWASP Top 10, 2021).

Table 13. Server-side data backup protection principles

Server-side network Database and Backup protection principles checklist	
1	**Create regular backups:-**IIoT web application database and server configuration backups are created periodically.
2	**Encrypt the backup data :-** To secure application data.
3.	**Validate working of Backup data :-**Validate server configuration and application data backs are working properly.

Table 14. Server-side architecture protection principles

Server-side architecture protection principles checklist	
1	**Create server-internal software interfaces and components**: To authorize and access the network file system, databases, and network resources, This protects exposed network resources from being accessed by attackers who could get full application databases and network resources.
2	**Encrypt and authenticate all connections and communications:** Never send unencrypted data between an IIoT web application client and a server service over internal or external networks, databases, or cloud data services (Vinod et al., 2008).
3.	**Use secret management tools and services for centralized security control:** Client and server Credentials, certificate keys, and other valuable information about the web application should be kept in a safe place on the web application server(Tanaskovic & Zivkovic, 2011).
4	**Implement Multi-level defense mechanism:** Defense is implemented on multiple layers of the IIoT web application software component. Suppose if one layer of protection is the compromised application is still protected (Tanaskovic & Zivkovic, 2011).

Table 15. Server-side network infrastructure protection principles

Server-side network infrastructure protection principles	
1	**Firewalls should be used to protect against network vulnerabilities:** Install a web application firewall and configure firewall policies to protect against network vulnerabilities.
2	**Use containers and dockers to deploy server-side web applications:** This prevents and protects the network filesystems, resources, and systems in the event of network breaches against cyber-attacks.
3	**Configure control and access policies:** For dockers and containers running applications, configure access and control to only require resources.
4	**Restrict Server services and service accounts with minimum privileges:** Execution of the process performed using accounts with minimal privileges and access rights so that if the attacker is successful, protection and executes part of the code, reducing its opportunities and thus potential damage (Tanaskovic & Zivkovic, 2011).
5	**Control and limit network data transfer using firewalls:** Restrict data leaving a network in transit to an external location called data egress from a server using a firewall setting.
6	**Hardening the server environment:** Disable unused services, protocols, functionalities, and server ports: on web application servers. This hardens the system against attacks.

Table 16. Server-side monitoring and incident response

Server-side monitoring and incident reporting checklist	
1	**Implement a mechanism for collecting, analyzing, and alerting on IIoT web application logs:** The IIoT web application server is interfaced to the deployed organization's Security Information and Event Monitoring (SIEM) or similar services to collect, analyze, and trigger events in the event of any security breaches or cyber-attacks.
2	**Collect security logs, security events, web application firewall logs, and web server events during runtime.**
3.	**Protect logging credentials and application-sensitive information.**
4	**Define the incident reporting and response strategy** (Vinod et al., 2008)

4.4 Unified security analysis framework for IIoT web applications

The second innovative solution (Design Construct 2) is the Integrated security analysis framework for IIoT Web Applications, illustrated in Figure 13 . The solution integrates the SSDLC phase-based standard security auditing and testing process and the ISACA three-phase auditing process. The present research sets out to develop an empirically testable security analysis framework coupled with security standards, security expert standard recommendation, top security risks and vulnerabilities, legislative regulations, and suitable auditing methodologies which, together, allow IIoT web application developing host company to obtain a clearer understanding of the security analysis of web applications at every phase of SSDLC applying ISACA auditing process.

The primary goal of this security analysis framework is to help IIoT web application developing companies conduct internal security auditing and testing of applications starting from defining requirements to the deployment and maintenance of the application on its industrial production and service-oriented customer network infrastructure. The security analysis framework for IIoT web application consists of two primary components: SSDLC phases and ISACA auditing process steps integrated. This research developed a generic and iterative framework that integrates auditing scopes, auditing methodology, and auditing reports for every SSDLC phase. This framework consists of eight core elements: five SSDLC and three ISACA auditing processes components. The input for this framework is the SSDLC activities output.

Requirement phase auditing

The first input starts with gathering needs of Industrial web application security needs in addition to application functionality, and subsequently, application security requirement is defined. Next, in the planning phase of ISACA, the auditing scopes and procedures are planned, which include security policies, coding standards, cryptography standards, security standards, and IIoT web application objectives. Next, using a planned auditing methodology, the security risk assessment and risk analysis of defined IIoT web application security requirements are conducted in the ISACA fieldwork phase. In the third ISACA reporting phase, the outcome of fieldwork is approved and derived security requirement report.

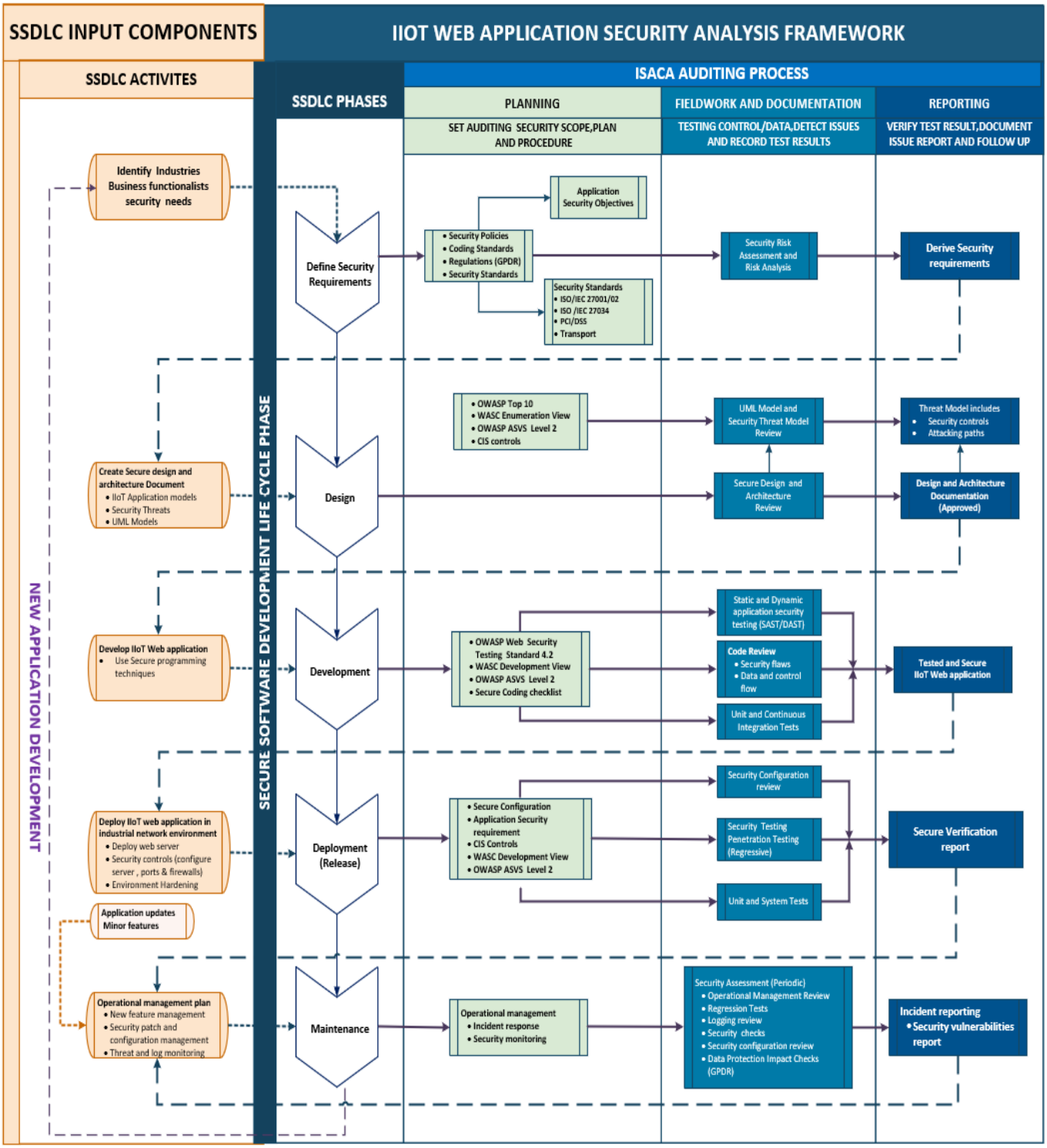

Figure 13. Unified IIoT web application security analysis framework

Design phase auditing

During the design phase, architecture, and design draft documents, including the UML model, threat model, and application model, are created using earlier approved security requirements for IIoT web applications. The architecture and design draft documents are input from the SSDLC design to the testing phase of ISACA. In this design phase, the ISACA audit sets scope, plans, and auditing procedures using standards expert recommendations such as OWASP top 10 risk, WASC enumeration view, OWASP ASVS level 2, and CIS controls. The next phase includes conducting audits of the application architecture and design using a specified audit technique. These design reviews also comprise threat and UML reviews. Next, the outcome of fieldwork in the ISACA reporting phase is approved architecture and design documents which include authorized thread models. The threat model documents the threat actors ,possible threat scenarios and attacking paths based on industrial IoT application use cases.

Development phase auditing

Next is the development phase, where the application developer develops IIoT web applications using predefined secure coding techniques, programming languages, and tools using approved architecture and design documents. The IIoT web application source code is the input from the SSDLC development phase to the ISACA auditing planning phase. In the ISACA audit plan phase, the scope, plans, and auditing procedures during this phase, based on recommendations from the security expert community, such as the OWASP web application testing standard, the WASC development view, the OWASP ASVS level 2 standard, and secure coding checklists.

Subsequently, in the next ISACA fieldwork phase, Static and Dynamic application security testing, unit testing, integration testing, and security expert code reviews are conducted using the planning auditing procedure in the planning phase. The outcome of this fieldwork phase is a list of all vulnerabilities identified in the source and secure coding issues. The security issues detected are reported back to the application developer. Subsequently, the developer addresses all security flaws in the code. The auditing fieldwork activities repeat until most of the security flaws get resolved. The outcome of ISACA reporting is a tested and secure IIoT web application and internal security test report which approves the deployment of a web application for the official release to industrial networks for usage.

Deployment phase auditing

Next is the deployment phase, where the application development company deploys the officially released IIoT web applications as part of SSDLC activity which consists of setting up web server application hardware and software and configuring and implementing security controls on the web application server. The primary activities of implementing security controls consist of enabling required ports, configuring firewall policies, and installing digital security certificates and keys. Industrial network environment hardening is also done by configuring web application-related resources and security configurations.

The deployment of IIoT web application on the industrial network is the input of the input from the SSDLC deployment phase to the ISACA auditing planning phase. As part of the deployment phase of ISACA audits, application and industrial security requirements, security configurations, and recommendations from security experts are used to set the scope, plan the audit, and set the auditing procedures. These include the OWASP web application testing standard, OWASP ASVS Level 2, and CIS controls.

Subsequently, in the next ISACA fieldwork phase, regressive security and penetration testing, unit testing, system testing, and security configuration reviews are conducted using the planning auditing procedure in the planning phase. The outcome of this fieldwork phase is a list of all vulnerabilities and security flaws identified in the web application and infrastructure. The sources issues detected are reported back to the application developer. The security flaws identified during security testing in the infrastructure are resolved by updating the software on the firewall or other network resources. Subsequently, the developer addresses all security flaws in the code. The auditing fieldwork activities repeat until most of the security flaws get resolved. The outcome of ISACA reporting is a security verification report which approves the operation and official launch of IIoT web application by industries.

Maintenance phase auditing

Next is the maintenance phase, where the application development company creates an operational management plan in this SSDLC phase activity. The operational management plan includes implementing a security and feature patch management mechanism, configuration management

mechanism, log monitoring mechanism, and incident reporting mechanism. The maintenance of the IIoT web application on the industrial network is the input of the input from the SSDLC maintenance phase to the ISACA auditing planning phase. As part of the maintenance phase of ISACA audits, standard incident response and security monitoring reviews methodologies are used to set the scope, plan the audit, and set the auditing procedures.

Subsequently, in the next ISACA fieldwork phase, periodic security assessments are conducted using the planning auditing procedure the planning phase. The security assessment consists of regular operational management Review, security regression tests, application and is related logs review, security checks, and security configuration review. The outcome of this fieldwork phase is a list of all vulnerabilities and security flaws identified in the web application and infrastructure. The result of fieldwork is to create an incident reporting document and a response strategy that lists security vulnerabilities reports. The incident strategy raises issue tickets for the IIoT web application developed company to provide security patches. Later, application development organizations release security patches to resolve identified web application risks.

4.4.1 Auditing guidelines for GDPR implementation

The SSDLC phases and ISACA's three-phase auditing process in the unified security analysis framework for IIoT web application presented in Figure 13 include GDPR regulations as input during the defining requirement phase and implemented during the design and development phase. Implementing and auditing the EU Regulation (EU) 2016/679 and Finland's GDPR principles listed in Table 9 and working in GDPR's key focus areas presented in Figure 10 is critical for IIoT web application companies to protect the personnel data of IIoT web application device users.

The innovative security analysis framework also includes auditing processes during deployment and operation phases by running regular data protection impacts checks. GDPR implementation and auditing guidelines support the new unified IIoT web application's security analysis framework compliant with EU and Finnish GDPR regulations. Table 17 lists the GDPR implementation and auditing policies to be implemented and utilized to conduct audits during the requirement, design, and development phases of SSDLC.

Table 17. Auditing policies for GDPR implementation

No	POLICIES
1	**Define, collect, and process only the bare minimum of personnel data; otherwise, avoid :-** First avoid or collect bare minimum personnel data .Erase the data if not required by data controller (Skendzic et al., 2018).
2	**The application collects personnel data that is "adequate, relevant, and limited to what is required in relation to the purposes for which it is processed"** (IT Governance Privacy Team, 2020).
3	**Using consent forms, data controllers get data subjects' permission and lawful legal rights to process personal data for one or more specific purposes:-** Application companies should get permission in visible terms and agreement form or consent form from the data subject for all purposes, including third party processing .(IT Governance Privacy Team, 2020).
4	**Data subjects' personnel data is collected for specified, explicit, and legitimate purposes only** (IT Governance Privacy Team, 2020).
5	**Protect and secure personnel data in IIoT application code, device, and tools** (Colesky et al., 2019)
6	**Transparency in informing data subjects clearly and openly about the intended use of personnel data.** (IT Governance Privacy Team, 2020).
7	**All personnel data should be transferred in encrypted and secure mode using TLS1.2/1.3 based HTTPS** (Vinod et al., 2008)**.**
8	**All personnel data should be encrypted and stored in a secure EU/Finland location** (Regulation (EU) 2016/679, 2016)**.**
9	**Data subjects are given transparency and erasure permission for their data** (IT Governance Privacy Team, 2020).
10	**Alert data breaches to data subjects** (IT Governance Privacy Team, 2020)
11	**Inform data subject about application logs** (IT Governance Privacy Team, 2020)
12	**The data subject should be able to access his stored personnel data through IIoT web application**.

Figure 14, show the basic GDPR auditing process flow. The IIoT web application development company uses GDPR implementation auditing policies listed in Table 17 to audit security risk assessment and analysis of defined requirements and derive GDPR requirements in the design and development phases. Next, the design phase uses documented GDPR requirements to create a web application architecture document and identify security and regulation threats. In the development phase, the company audits the developed web application using the GDPR implementation and auditing policies to mitigate any issues related to personnel data breaches or leakage violating GDPR.

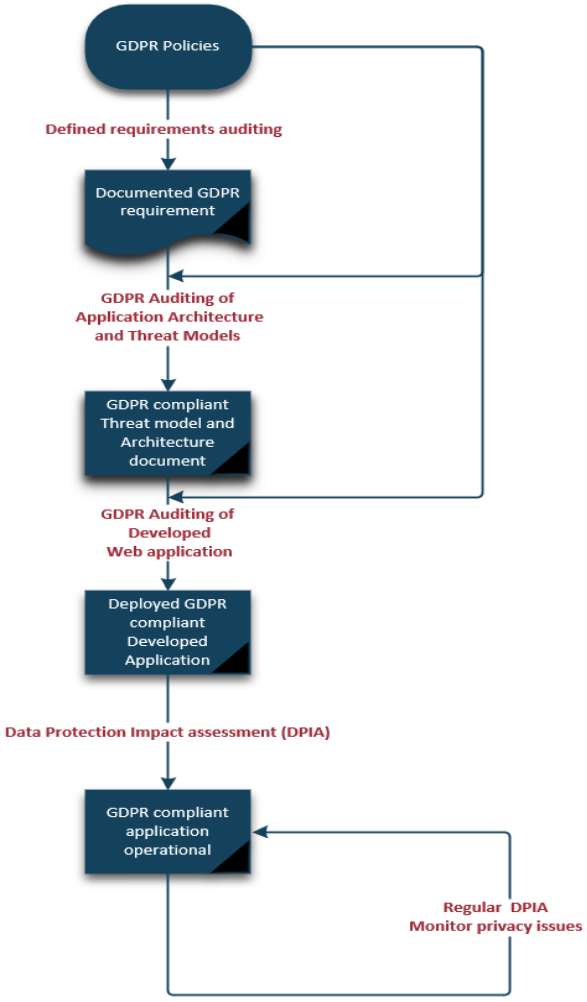

Figure 14. GDPR auditing process flow

In the deployment phase, the GDPR is verified by auditing web applications using a data protection impact assessment (DPIA) method to help the company identify and minimize privacy risks related

to GDPR. Also, in the maintenance phase of the IIoT web application, the DPIA is executed periodically to monitor and mitigate GDPR privacy risks.

4.4.2 Auditing guidelines for authentication and secure data transfer implementation

The SSDLC phases and ISACA's three-phase auditing process in the unified security analysis framework for IIoT web application presented in Figure 13 include secure data transfer implementation and auditing guidelines. Critical guidelines to audit data transfer security in specified Table 11 and Table 12. In Table 11, the crucial principles related to secure data transfer to be implemented and audited by IIoT web application development companies are encrypted data transfer, enforced encryption using HTTP Strict Transport Security (HSTS) directives, input validation and use of client certificates, and universal 2nd-factor authentication (U2F). In Table 12, the crucial principles related to secure data transfer to be implemented and audited by IIoT web application development companies are Input validation, strict access controls, and application authorization mechanisms implementation. Additionally, IIoT application developing organizations need to decide on the digital certificates and cryptographic key generation mechanism and their formats.

Public key-based digital certificates are the principal method for enabling authentication capability in the IIOT devices and client applications. Figure 4 shows certificates and cryptographic keys installed in IIoT client and server web applications, IoT client devices, and cloud and database storage servers for secure device authentication and data transmission. Figure 12 shows the device / application authentication and secured data transmission mechanisms commonly implemented across organizations. Generally, IIoT organizations use X.509 digital certificates for device authentication and TLS-based HTTPS protocols for secure data transmission. Table lists the data and source authentication, and secure data transmission implementation and auditing principles to be implemented and utilized to conduct audits during the requirement, design, and development phases of SSDLC.

Table 18. Auditing policies for autentication and secure data transmission implementation

	POLICIES
1	**Use SSL/TLS to send sensitive data from pages/forms/anywhere to server or third party sites:-** SSL ensures secretiveness by using encryption ciphers and non-repudiation through the usage of digital certificates signed by a recognized certifying authority and the use of public keys. (Vinod et al., 2008)
2	**Use asymmetric key algorithms for data integrity and source authentication and for key establishment:-** Generally, Digital Signature Algorithms use Asymmetric keys algorithms. Security standards have approved the Elliptic Curve Digital Signature Algorithm (ECDSA) and RSA algorithm to generate and verify digital signatures.
3	**Use algorithms security strength of key lengths greater then 112 Bits** (Vinod et al., 2008)
4	**Use symmetric key algorithms with longer key lengths :-** This key is used for encrypting and decrypting data. The longer the key, the higher the possibilities of protecting against brute-force attempts and other cracking attempts. AES is most well-known and safe algorithms .It is relatively secure, with the length of the symmetric key more than 112 bits (Vinod et al., 2008).
5	**Use Hashing algorithms with greater key length:-** Hashing is a method for converting an original string or text to a fixed-length text or string that represents the original. Use SHA-256 and SHA-512 are hashing algorithms that are more secure and robust and avoid MD5 and SHA1 (Vinod et al., 2008).
6	**Generated cryptographic keys must be safeguarded and securely stored:-** To secure encrypted data protect crytographic keys used during transmission. So Keys Generate random cryptographic keys that are server-specific.Independently of the application, store and manage the keys (Vinod et al., 2008).
7	**Define a proper and secure process for generating and installing digital certificates and cryptographic keys :-** IIoT web application companies must follow security standards such as NIST and ISO27001 defined key generation processes according to IIoT web application and device features and algorithms to secure the boarding of new devices (Vinod et al., 2008)**.**

8	**Define appropriate cryptoperiods and the process for renewing digital certificates and cryptographic keys:-** IIoT web application organizations must implement Security standards such as NIST and ISO27001 defined processes to renew digital certificates and algorithm renewals (Vinod et al., 2008; Chopra & Chaudhary, 2020).

4.5 Evaluation of designed constructs

The evaluation and demonstration of the practical solution of the constructive research steps' aims to find, assess, and evaluate both innovatively designed solutions to perform security analysis of the software life cycle of IIoT web applications. The feedback provided during this phase is crucial to the study results and could change the constructs. The research has two possibilities to examine. The first option was to evaluate the solutions based on a demonstration using the security analysis technique in the software development life cycle of the IIoT web application. Due to it being time-consuming and challenging enough to define a realistic implementation conclusion and objectives did not use this option to evaluate. The second option was to interview software application experts, so this research selected software experts from the host company who are potential users of these innovative solutions to evaluate both constructs.

In this evaluation phase, conducted the host company's expert online interviews and also collected feedback using a questionnaire interview method to evaluate and demonstrate the audit procedure of both the IIoT Web applications security design and auditing principles checklist and the Integrated Security Analysis Framework for IIoT Web Applications (including GDPR and secure data transfer auditing guidelines), constructive solutions presented in sections 4.3 and 4.4. Research collected feedback for evaluation through questions and interviews with two software architects and experts from the research host company, both of whom had over 20 years of industrial experience. One expert is in charge of integrating a security analysis solution to audit and test the security of IIoT web applications at each stage of the software development life cycle. In this research used the theme Interview and questionnaire Interview methodologies to evaluate the constructs.

Initially, started the evaluation method by drafting questions (see Appendix 1: Evaluation of IIoT Web Application Security Analysis Questionnaires). Subsequently, in this process shared the thesis

document and the evaluation questionnaires with two software specialists and received feedback answers in the e-mail to evaluation questions. Later, to review the questionnaire answers and collect more feedback scheduled an online interview with both interviewees. During the interview introduced and familiarized the experts with the innovative constructs. The gathered results of the interview are summarized and presented in the next chapter. The objective of both interviews was to check the applicability of the suggested construct and collect feedback on the results. The total time spent on both interviews was around approximately an hour.

5 RESULTS

5.1 Theme and questionnaire interview results

The questionnaire interview results showed that for the first question related to the validation and accuracy of the research question and problem description, both interviewees from the host company answered that the company is new and are not having much previous experience in web application development. The research is helping the host company establish the proper development phases and advises them on how to protect against common security threats. Both Interviewees agreed problem description was correct and the research question was accurate.

The next question was related to opinions about the security concept of web applications presented in the thesis document, and interviewees said that the basic concepts of the security objectives and process development lifecycle provided them good understanding. Both interviewees said that web application security concepts and mechanisms were not previously familiar with, and the thesis theory part made them understand clearly. According to both interviewees, the theory part of the research clarified their basic concepts of the security objectives and processes required for the software development lifecycle. Both the interviewees were able to connect the theoretical concepts and understand the information flow.

Next, the questions were related to the feedback on both constructed solutions, where collected the feedback on the demonstrated constructs in two parts: one through answers to questions and the other through an online interview. Interviewees stated in response to the questionnaire that the first solution, the security principles checklist for auditing web applications, secures all systems and infrastructures and provides a reference to audit but observed that there is a gap in that use cases such as securely integrating new IoT devices or users into the system are not covered. They had an open question-related checklist not covering principles related to updating the software updatable of all system modules with the latest security patches and features. Next, about the second construct solution, the interviewees stated that the Unified SSDL and ISAC framework picture clearly describes separate phases and corresponding functionalities, and all the components are once in place, the process of using them is efficient. In practice, the development happens more in

parallel, at least when the company does not yet have anything ready for the development flow, and it causes complexity.

Next, during the online interview explained all the theoretical concepts used to design the constructive solution in detail and then demonstrated to both interviewees how the constructed solution works, and the background related to solutions. In the presentation, a one-page unified security analysis framework covering all the auditing processes and security standards of all SSDLC phases was explained clearly, including the auditing flow.

After the presentation, reviewed the questionnaire feedback answers from both interviewees. Responded to inquiries about why the security auditing checklist does not contain principles related to application updates, clarified security checklist outlines the security principles to audit. Subsequently, during interview discussions, accepted in practice, development occurs more in parallel, at least when the company does not yet have anything ready for the development flow, which adds complexity; however, this solution only applies to security auditing and testing; software development can occur in parallel. In an idealistic situation, web application development must adhere to the SSDLC procedure, and repeatedly companies have these execution issues.

Interviewees gave feedback that using NIST is not considered in his research which answered that NIST is extensive and web applications developers in the EU and Finland generally use ISO standards and OWASP expert advice. Interviewees also commented that the security analysis framework is not showing the failure of the field testing in the flow s for answered by telling them that it is understood field testing failure is a checkpoint and application development cannot go future without success.

Finally, interviewees feedback concluded stating that they did not see any critical gaps in the two constructed solutions. Security analysis framework solution clarifies what are the practices needed in every the SSDLC phases . Both interviewees told that the most common web application vulnerabilities are well documented in thesis and helped them understand the security risks and mitigate those and also stated that GDPR policies provide them a good overview of the company's responsibilities under GDPR, handling sensitive data, and also provides clear principles used as requirements for the system implementation.

The interviewee suggested a couple of improvements, including minor software feature updates, new application version development flow, and each phase's failure flow needed to add to the unified SSDL and ISACA framework. Finally, Interviewees validated the innovative solutions by giving feedback that the solutions helped them consider needed security practicalities during the application and system implementation phases. Both interviewees accepted that the demonstrated solution would help them consider needed security practicalities during the application development and system implementation phases.

Based on the Interviewees' improvements modified in Figure 13 Unified security analysis framework for IIoT web application showing the new minor feature and application updates during SSDLC. Also, added flow demonstrating new application version development should start from the requirement phase by collecting feedback and data from currently deployed applications for operations.

5.2 Research results

This section presents how the construct's solutions meet the standards for this thesis and answer the research questions for this thesis. The IIoT Web applications security design and auditing principles checklist and the Integrated Security Analysis Framework for IIoT Web Applications (including GDPR and secure data transfer auditing guidelines) are constructive solutions presented in sections 4.3 and 4.4 to answer the primary question and its subdivided questions also.

The auditing and testing of industrial Internet of Things (IIoT) web application security and ensuring that user data is compliant with the General Data Protection Regulations conducted using the security analysis framework for IIoT web applications presented in Figure 13 . Additional, supporting the security analysis framework by the security auditing principles defined in tables in IIoT web applications security design and validation principles checklist, by auditing guidelines for GDPR, source authentication, and secure data transmission defined in Table 17 and Table 18, respectively.

According to the evaluation results, the constructed solutions assist IIoT web application development companies in the auditing, defining requirements, designing, and developing application

phases of the SSDLC to mitigate and reduce the security issues in applications. The security analysis framework solution covers all the security-related concepts that need to be taken care of during the requirement, design, and development phases and in the deployment and maintenance phases of applications in the industrial network. The first constructed security principles check list helps to list all the security related issues to validate and ensures the security of IIoT web application system and the security of an organization's network is not breached and not exploited by IIoT web applications. Evaluation conducted using expert interview showed implementing both security principles checklist and security analysis framework including GDPR polices , authentication and secure data transmission to audit helps to detect secure issue beforehand and develop IIoT web application resistant to major cyber-attacks after deploying to industrial networks. The first constructed security principles checklist helped collect all the security-related issues to validate and ensure the security of the IIoT web application system is not breached and of organization's network is not breached and not exploited by IIoT web applications. An evaluation conducted using expert interviews showed that implementing the security principles checklist and the security analysis framework, including GDPR policies, authentication, and secure data transmission to audit, helped detect security issues beforehand and develop IIoT web applications resistant to major cyberattacks after deploying to industrial networks

The first constructed solution, the security principles checklist, includes the principles defined in ISO and OWASP security standards. The security principles checklist also considers policies implemented to protect all assets and infrastructure, such as client and server application systems, server-side architecture, databases, network infrastructure, incident monitoring, and reporting systems, involved in deploying and operating IIoT web applications in an industrial environment. The security principles also consider theoretical concepts studied during research.

The second constructed solution, a unified security analysis framework for IIoT web applications, integrates all security concepts studied during the initial research phase of the thesis. Evaluation results showed that the one-page framework gave a complete idea of how to conduct an audit and test a web application. It showed them all security-related theoretical concepts, which include security standards such as ISO27001 and ISO27002, expert community advice reports such as OWASP web application testing and CIS controls, security testing tools, security, and penetration testing methodologies, threats and vulnerabilities lists such as OWASP Top 10 vulnerabilities,

WASC threats, and GDPR. Additionally, Evaluation results showed that the framework presented in which phase of SSDLC and ISACA auditing steps which best part of this solution. To support the security analysis framework the auditing polices defined for GDPR , source authentication and secure data transmission helped to achieve all IIoT web application security objectives. Supporting the security analysis framework, the auditing policies defined for GDPR, source authentication, and secure data transmission helped achieve all IIoT web application security objectives. The GDPR theory concepts present the six principles and policies defined in the GDPR auditing guidelines and include corresponding EU principles. The solution specified source authentication and secure data transmission policies that match the security concepts in the theoretical section.

The proposed construct and research provide policies and checklists that include most of the security standards mechanisms to protect industrial network systems and web applications from cyber-attacks and security breaches that meet the research requirements established for this thesis. Using this approach, it is possible to conclude that the primary research questions specified at the start of the thesis process are complete. Based on the interview answers and the author's personal experience, the construct solution provides the required auditing testing mechanism and solutions for IIoT web application that starts from defining requirements to deploying and maintaining the web application operations in industrial networks. The interviews provided input on the thesis subject's validity and universal applicability. The security analysis framework was modified in response to interview comments to include minor application feature changes and new application development processes.

6 CONCLUSION

The thesis started with the host company's need to audit and test procedures to protect their developing IIoT web application against cyber-attacks and security breaches in industrial networks. During the study of the theoretical concepts, the research phase observed that earlier published research and standards did not include a framework to cover audit and testing methodology during the SSDLC phase. The audit and testing methodology were independent of the SSDLC phases. Furthermore, The assets and infrastructure utilized by IIoT web applications operating in industrial networks are not part solutions, including security auditing principles illustrated in research papers and books offered. Most of the security concepts and solutions are available for a web application developer need to refer to multiple sources and documents to develop secured web applications. This revelation reaffirmed the importance of this study, and further future research on this issue is necessary.

The author found challenging to find an integrated solution to audit and test IIoT web applications covering both the application developer perspective (in this case, the host company) and the application user perspective (in this research), the manufacturing and servicing industries. The study of several theoretical concepts, such as security objectives, expert-advice community documents, security standard documentation, SSDLC, and ISACA auditing methodology, helped construct innovative solutions using all theoretical concepts studied during the research. Based on the author's acquired knowledge and experience created two constructive solutions: the IIoT Web applications security design and auditing principles checklist and the Integrated Security Analysis Framework for IIoT Web Applications (including GDPR and secure data transfer auditing guidelines).

The constructive research approach was a viable choice for this thesis, as it provides proper steps to proceed with this research and reduces the hurdles. The constructive approach steps provided scientific and logical guidance to every phase of this research fitted correctly. Following the constructive research approach, the method was helpful in constructing the innovative solution by understanding theoretical concepts and problems in depth.

During the research process and implementation of the study, constructed two innovative solutions using the ISACA three-step methodology applied to each phase of SSSLC to conduct security

audits and tests. The implementation phase integrated all required security standards, OWASP community experts, WASC threats, top security threats, auditing, and testing methods, testing tools, results in documentation, secure data transmission, source authentication methodology, and GDPR principles collected during theoretical concepts.

After developing two constructive solutions, evaluated the solution by demonstrating it to two experts from the host company through a theme and questionnaire interview methodology. According to both interviewees' feedback, the two constructed solutions assist the IIoT application development to assess necessary security considerations throughout the application and system implementation stages During evaluation, the interviewee suggested a couple of improvements, including minor software feature updates, new application version development flow, and each phase's failure flow needed to add to the unified SSDL and ISACA framework. Implemented minor software feature updates and new application version development flow modifying the security analysis framework. . Both interviewers agreed that the solution demonstrated would help them consider critical security considerations during the application development and deployment stages.

Due to limited resources and time restrictions, complete testing, and broad deployment of the proposed solutions in industrial environments seem impossible during the thesis project. It would have strengthened the reliability of the results and provided a clearer picture of the recommended construct's suitability for security analysis framework in industries. Additional practical research needs to be conducted with a greater implementation scope and a longer observation time.

However, in the future, the practical demonstration of the two constructed solutions in actual application development is significant for further verifying both solutions. Due to time restrictions and the difficulties of articulating the objective of the practical presentation of both options did not implement this assessment strategy. As a future objective, the two solutions' practical implementation would help evaluate, and the host company would implement them. Subsequently, in the future, updating the framework with auditing control flow when a failure occurs would improve these solutions. Using and validating the security principles, GDPR, authentication, and secure transmission guidelines would help with the maturity of the constructed solutions.

While conducting and assessing the interviews, the author observed that readily his own experiences and perspectives influences the interviewees' thoughts and responses. A minimal number of interviewers was conducted due to the thesis's timeline. Regardless of issues in findings, the interviews gave a broader context for the suggested design and thesis outcomes. To conclude , the required and possible constructs solutions developed in this thesis solves the problems states in the primary research question and its sub-divided research questions of the thesis.

References

Article 29 Data Protection Working Party, European Parliament, Council of the European Union, Guidelines on Personal data breach notification under Regulation 2016/679 (wp250rev.01) (2017). http://ec.europa.eu/newsroom/article29/itemdetail.cfm?item_id=612052

Bhattacharjee, S. (2018). *Practical Industrial Internet of Things Security: A practitioner's guide to securing connected industries.* Packt Publishing.

Chebudie, A. B., Minerva, R., & Rotondi, D. (2014). *Towards a definition of the Internet of Things (IoT)*. IEEE Internet of Things. https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf

Chopra, A., & Chaudhary, M. (2020). *Implementing an Information Security Management System: Security Management Based on ISO 27001 Guidelines*. Apress. https://doi.org/10.1007/978-1-4842-5413-4

Correa, R. A., Higuera, J. R. B., Higuera, J. B., Montalvo, J. A. S., Rubio, M. S., & Magreñán, Á. A. (2021). Hybrid Security Assessment Methodology for Web Applications. *CMES-Computer Modeling in Engineering & Sciences*, *126*(1), 89–124. https://doi.org/10.32604/cmes.2021.010700

CIS Critical Security Controls Version 8. (2021). Center for Internet Security. https://www.cisecurity.org/controls/v8/

Chavan, S., & Tamane, S. (2020). Enhancement in Cloud Security for Web Application Attacks. *2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC)*, 91–95. https://doi.org/10.1109/ICSIDEMPC49020.2020.9299629

Colesky, M., Demetzou, K., Fritsch, L., & Herold, S. (2019). Helping Software Architects Familiarize with the General Data Protection Regulation. *2019 IEEE International Conference on Software Architecture Companion (ICSA-C)*, 226–229. https://doi.org/10.1109/ICSA-C.2019.00046

Cross, M. (Ed.). (2007). *Web application vulnerabilities: Detect, exploit, prevent*. Syngress Pub.

Data Protection Act (1050/2018*)*, Ministry of Justice, Finland (2018). https://www.finlex.fi/en/laki/kaannokset/2018/en20181050.pdf

Dzung, D., Naedele, M., Von Hoff, T. P., & Crevatin, M. (2005). Security for Industrial Communication Systems. *Proceedings of the IEEE*, *93*(6), 1152–1177. https://doi.org/10.1109/JPROC.2005.849714

Ethical Recommendations for Thesis Writing at Universities of Applied Science. (2020). The Rectors' Conference of Finnish Universities of Applied Sciences (ARENE). https://www.arene.fi/wp-content/uploads/Raportit/2020/ETHICAL%20RECOMMENDATIONS%20FOR%20THESIS%20WRITING%20AT%20UNIVERSITIES%20OF%20APPLIED%20SCIENCES_2020.pdf?_t=1578480382

IT Governance Privacy Team. (2020). *EU General Data Protection Regulation (GDPR) : an implementation and compliance guide*. IT Governance Publishing Ltd.

Goncalves, A., & Correia, A. (2019). Determination of Compatibility between Activities according EU General Data Protection Regulation: An Initial Study. *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, 1–7. https://doi.org/10.23919/CISTI.2019.8760970

Gilchrist, A. (2016). *Industry 4.0*. Apress. https://doi.org/10.1007/978-1-4842-2047-4

Helmiawan, M. A., Firmansyah, E., Fadil, I., Sofivan, Y., Mahardika, F., & Guntara, A. (2020). Analysis of Web Security Using Open Web Application Security Project 10. *2020 8th International Conference on Cyber and IT Service Management (CITSM)*, 1–5. https://doi.org/10.1109/CITSM50537.2020.9268856

Hoffman, A. (2020). *Web Application Security. Exploitation and Countermeasures for Modern Web Application.* (1st ed.). O'Reilly Media.

Huang, W., Li, R., Maple, C., Yang, H.-J., Foskett, D., & Cleaver, V. (2010). A novel lifecycle model for Web-based application development in small and medium enterprises. *International Journal of Automation and Computing*, *7*(3), 389–398. https://doi.org/10.1007/s11633-010-0519-3

Information Systems Auditing: Tools and Techniques—Creating Audit Programs. (2016). Information Systems Audit and Control Association, Inc. (ISACA). https://community.mis.temple.edu/mis5201sec001sp2017/files/2017/03/IS-Auditing-Tools-and-Techniques-Creating-Audit-Programs.pdf

Kasanen, E., Lukka, K., & Siitonen, A. (1993). The Constructive Approach in Management Accounting Research. *Journal of Management Accounting Research*, *5*, 242–264.

Kapodistria, H., Mitropoulos, S., & Douligeris, C. (2011). An advanced web attack detection and prevention tool. *Information Management & Computer Security*, *19*(5), 280–299. https://doi.org/10.1108/09685221111188584

Lee, H. Y., & Park, Y. S. (2017). Common Requirements for Web Application Vulnerability Scanners for the Internet of Things. *2017 International Conference on Software Security and Assurance (ICSSA)*, 111–111. https://doi.org/10.1109/ICSSA.2017.31

Lepofsky, R. (2014). *The manager's guide to web application security: A concise guide to the weaker side of the web* (1st ed.). APress.

Lukka, K. (2003). The Constructive Research Approach. In O.-P. Hilmola & L. Ojala (Eds.), *Case study research in logistics* (pp. 83–101). Turku School of Economics and Business Administration, Turku. https://janet.finna.fi/Record/jamk.991011464806251

McDonald, M. (2020). *Web security for developers*. No Starch Press, Inc.

Mueller, J. P. (2015). *Security for web developers*. O'Reilly Media.

Olanrewaju, R. F., Khan, B. U. I., Morshidi, M. A., Anwar, F., & Kiah, M. L. B. M. (2021). A Frictionless and Secure User Authentication in Web-Based Premium Applications. *IEEE Access*, *9*, 129240–129255. https://doi.org/10.1109/ACCESS.2021.3110310

OWASP Application Security Verification Standard 4.0.3. (2021). OWASP Foundation. https://owasp.org/www-project-application-security-verification-standard/

OWASP Top 10. (2017). The Ten Most Critical Web Application Security Risks,OWASP. https://owasp.org/www-project-top-ten/2017/

OWASP Top 10. (2021). The Ten Most Critical Web Application Security Risks,OWASP. https://owasp.org/Top10/

OWASP Web Security Testing Guide Version 4.2. (2020). OWASP Foundation. https://owasp.org/www-project-web-security-testing-guide/v42/

Panchal, A. C., Khadse, V. M., & Mahalle, P. N. (2018). Security Issues in IIoT: A Comprehensive Survey of Attacks on IIoT and Its Countermeasures. *2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN)*, 124–130. https://doi.org/10.1109/GCWCN.2018.8668630

Pompon, R. (2016). *IT Security Risk Control Management*. Apress. https://doi.org/10.1007/978-1-4842-2140-2

Responsible conduct of research and procedures for handling allegations of misconduct in Finland. (2012). Finnish Advisory Board on Research Integrity (TENK). https://www.tenk.fi/sites/tenk.fi/files/HTK_ohje_2012.pdf

Regulation (EU) 2016/679, European Parliament, Council of the European Union, The protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (2016). http://data.europa.eu/eli/reg/2016/679/oj

Sadqi, Y., & Maleh, Y. (2020). A systematic review and taxonomy of web applications threats. *Information Security Journal: A Global Perspective*, 1–27. https://doi.org/10.1080/19393555.2020.1853855

Skendzic, A., Kovacic, B., & Tijan, E. (2018). General data protection regulation—Protection of personal data in an organisation. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1370–1375. https://doi.org/10.23919/MIPRO.2018.8400247

Steinke, G., Tundrea, E., & Kelly, K. (2011). Towards an Understanding of Web Application Security Threats and Incidents. *Journal of Information Privacy and Security*, *7*(4), 54–69. https://doi.org/10.1080/15536548.2011.10855923

Tsochev, G. (2020). Some Security Problems and Aspects of the Industrial Internet of Things. *2020 International Conference on Information Technologies (InfoTech)*, 1–5. https://doi.org/10.1109/InfoTech49733.2020.9211078

Tanaskovic, T. M., & Zivkovic, M. Z. (2011). Security principles for web applications. *2011 19thTelecommunications Forum (TELFOR) Proceedings of Papers*, 1507–1510. https://doi.org/10.1109/TELFOR.2011.6143843

van Lier, B. (2017). The industrial internet of things and cyber security: An ecological and systemic perspective on security in digital industrial ecosystems. *2017 21st International Conference on System Theory, Control and Computing (ICSTCC)*, 641–647. https://doi.org/10.1109/ICSTCC.2017.8107108

Vinod, V., Anoop, M., & Firosh, U. (2008). *Application security in the ISO27001 environment*. IT Governance Publishing.

WASC Threat Classification (Version 2.00). (2010). Web Application Security Consortium. http://projects.webappsec.org/f/WASC-TC-v2_0.pdf

Yu, X., & Guo, H. (2019). A Survey on IIoT Security. *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, 1–5. https://doi.org/10.1109/VTS-APWCS.2019.8851679

Appendices

Appendix 1. Evaluation of IIoT Web Application Security Analysis Questionaires

Evaluation of IIoT Web Application Security Analysis Framework and Checklist

Questions	
Question #1:	Is the research question correct in identifying the web application-related problem that the company is facing?
Notes:	Enter notes
Question #2:	Did you clearly understand the theoretical concepts and industrial infrastructures presented in this thesis report? How practical did you find this literature to address the problem? Any recommendations for improvements or feedback?
Notes:	Enter notes
Question #3:	Are you able to connect the theoretical concepts and their information flow?
Notes:	Enter notes
Question #4:	First solution:-The IoT web application security design and validation principles checklist cover all assets of IIoT web applications, starting from common security protection principles, client-side and server-side applications, server-side data backup, network infrastructure, monitoring, and incident reporting, as well as GPDR and secure transfer principles, to protect against web attacks. Do you find it covers all security requirements or any gaps?
Notes:	Enter notes
Question #5:	The second solution integrates the SSDLC phase-based standard security auditing and testing process on the Y-axis and the ISACA three-phase auditing process on X-Axis and additional development activities in one page. Do you clearly understand all components and their functional flow? Do you find implementing actual product development easier or more complex? Do you find it feasible to use this one page? Please give your feedback and suggestions
Notes:	Enter notes

Question #6: Do the security design and validation principles checklist to protect web attacks and SDLC and ISAC auditing steps unified framework designed for a company's IIoT web application's security auditing covers the requirements adequately?

Notes: Enter notes

Question #7: Have the solutions contributed towards improving web application satisfactorily compared to the objectives?

Notes: Enter notes

Question #8: Are there any gaps in your understanding the designed constructive solution before and after reading the research thesis? Please give your feedback

Notes: Enter notes

Question #9: Do you have clear understanding of how auditing framework and security principles is trying to mitigate the security risks and vulnerabilities before deploying to customer network? Is there something to be improved?

Notes: Enter notes

Question #10: Do you find GPDR and Secure Data solutions supporting unified IIoT web application framework feasible?

Notes: Enter notes

Question #11: Finally, Do you find it beneficial to implement both constructive solutions for internal auditing to help with the development, deployment, and maintenance of a secure IIoT web application?

Notes: Enter notes

Please provide additional Feedback

Enter Additional Notes.