ISSN: 2182-2069 / E-ISSN: 2182-2077

Security Mechanism in MAMATA Healthcare System Using Rule based Algorithm for Maternal Hospitals and Pathology Laboratories

Pradeepkumar C Dhage[1*], Rajesh A. Thakker[2], and Krishna K. Warhade[3]

[1*]Research Scholar, Gujarat Technological University (GTU), Ahmedabad, India.
pradeepdhage7@gmail.com, https://orcid.org/0000-0003-2318-6592

[2]Director, R & D Cell, GTU, Ahmedabad, India. rathakker2008@gmail.com,
https://orcid.org/0000-0001-9625-7588

[3]Professor, Department of Electrical & Electronics Engineering, Dr. Vishwanath Karad MIT World Peace University, Pune, India. krishna.warhade@mitwpu.edu.in,
https://orcid.org/0000-0002-0282-9766

Received: July 13, 2024; Revised: August 20, 2024; Accepted: September 24, 2024; Published: November 30, 2024

Abstract

Recently, innovation in the digital health industry has grown steadily, trending with medical cyber-physical systems' (MCPS) integrated solutions, regulations, real-time analytics, and software frameworks. Amid these advancements, the paramount concern is the security of medical information, particularly in maternity hospitals where data breaches must be rigorously avoided. While some researchers have explored security methods, they have encountered limitations and identified existing risks. In this research article, the authors propose a novel security mechanism/method called MAMATA Agent/BOT (i.e. Medical-data-Access-for-Maternal-Treatment-in-rural-Area) to discover and mitigate trending vulnerabilities, drawing reference from the OWASP (Open Web Application Security Project) Top 10 Report 2017/2020. The author's approach leverages a rule-based algorithm, enhancing the robustness of the security mechanism. This MCPS system and security mechanism are implemented within PHP and Laravel environments, and comparative analysis is carried out against a few existing security methods commonly employed in general healthcare (like SOA/SOD Framework, Andrew Austin test, Marcelo test, etc..). Results demonstrate a quantified improvement, e.g., 60-70% improvement in detectable vulnerabilities compared to prior approaches. Additionally, the authors extend their previous research by depicting different turns of events and applying a more stable, secured structure tailored to medical applications. Quantitative data presented with comparative analysis in tabular format validates the effectiveness of the proposed methods by detecting and, or fixing 7 to 9 major vulnerabilities (out of the top 10) identified by security assessment tools like 'Burp Suite Proxy' and 'Nikto'.

Keywords: Security Model, Intrusion Detection and Prevention, CPS (Cyber Physical System), Healthcare, CVSS Score, OWASP, Maternal Medical data, Rule-based architecture.

Journal of Internet Services and Information Security (JISIS), volume: 14, number: 4 (November), pp. 292-311.
DOI: *10.58346/JISIS.2024.I4.018*

*Corresponding author: Research Scholar, Gujarat Technological University (GTU), Ahmedabad, India.

1 Introduction

All Medical Cyber-Physical Systems (MCPS) refers to integrating information and communication technologies in the healthcare sector. The different modes of cyber-physical systems (Alhumud et al., 2016; Lydia et al., 2023) lead to the best directional extension for healthcare researchers. MCPS enhances medical treatment quality and efficiency by integrating medical devices, facilities, and patients facilitated by Cloud and Big Data (Zhang et al., 2015). Significant Research on Industrial Cyber-Physical Systems (ICPS) uses a physically unclonable function (PUF) for security and validates it using a real-or-random model (Shamshad et al., 2021) ensuring more stable security. Similarly, several patient monitoring systems were developed in a few of the studies (Clarke et al., 2017; Malasinghe et al., 2019), making them more adapted to the medical data usage. The advancements in MCPS make it life-critical, along with its challenges (Lee et al., 2011). Software plays an increasingly important role in the MCPS-based ecosystem. Digital healthcare systems may concurrently monitor numerous facets of the patient's physiology, i.e., they rely on software-defined functionality (Hu et al., 2015) to treat patients; this, in turn, poses several data assurance and security challenges. MCPS is vulnerable to cyber threats, including hacking, malware, and data breaches, which can lead to the loss of sensitive medical information, system disruption, and patient safety issues. The adopting Electronic Health Record (EHR) applications (Haque et al., 2014) raises concerns about unauthorized access, manipulation, and threats to healthcare infrastructure in various nations (Odilov et al., 2024).

Maternal women's health records, genetic information, and labor details are among the most sensitive types of data that are acquired throughout the treatment stages, making the security of electronic health (e-health) systems critical (Sofiene et al., 2023). Such data becomes essential for offering proper medical care, but given its distinctly personalized nature, it needs to be protected even more. Due to the exposure to personal healthcare information, data breaches on e-health platforms can have adverse effects on maternal women, including prejudice, anxiety, and potential trauma to their future health or the health of the infant (Kodric et al., 2021). Poor e-health security encounters the potential to erode women's trust in medical professionals, deterring them from acquiring critical prenatal and postnatal treatment, which could have a detrimental effect on maternal women. Since healthcare professionals have a moral and legal obligation to protect sensitive patient data and win women's assurance of care, reliable e-health security systems remain crucial to providing maternity treatment. Hence, it is essential to consider these security challenges as research gaps to ensure the secure and reliable operation of MCPS in the healthcare sector (Neelima et al., 2024).

1.1. CPS Security Challenges

- Lack of focus: Early vulnerability identifications by MCPS professionals have made it challenging to create a secured healthcare system, as identified vulnerabilities are reasonably more expensive to fix compared to earlier discoveries.
- The known HIPAA guidelines (Health Insurance Portability and Accountability Act) can be used to assess the application of open-source EHR for security weaknesses, but they are serve as only indicative.
- Contextual vulnerability analysis on eHealth systems can be done through manual/automated penetration testing or static analysis, with manual testing detecting design flaws and automated techniques being more effective, making software vulnerabilities a challenge.

To address this, a few security experts used their knowledge to attempt attacks on an application (Shar & Tan, 2012) in an exploratory and opportunistic way in a process known as penetration testing. However, the entire development team, and security professionals must build security into a product. Programming analysts and designers can collaborate proactively with security experts to develop a security test plan (i.e., black box). Hence, there is a solid need to design and develop a unique software-based security framework that addresses MCPS challenges. By ensuring secure transmission, storage, and use, these frameworks should protect medical systems and stop unauthorized use of real-time eHealth systems (Garkoti et al., 2014). Security incidents can still occur during development phases or due to platform defects, system complexity, inadequate evaluation of program inputs, or information security threats. To mitigate these risks, security testing methodologies (Salas et al., 2015) can be employed to evaluate the robustness of web services in eHealth applications. Thereby requires to simulate several forms of attacks for instance, XSS, injections, attacks (brute-force), etc. Attacks can result in the insertion of malicious content, i.e., malware, into the resulting SOAP message/document (Kakavand et al., 2019). In a nutshell, there is a strong need for a structured (rule-based) security mechanism that can help to cater to all the above challenges and gaps.

1.2. Objectives of the Research Work

In this article, the research aims to improve security in the previous work of 'MAMATA Healthcare Cyber-Physical System' (i.e., Medical data Access for Maternal Treatment in rural Areas) (Patent no. 2022-04698 South Africa). Maternal healthcare units and pathology laboratories often lack organization and protection for health records. MAMATA CPS is a new MCPS system that uses Lara-Admin, a lightweight frontend framework (designed on the Laravel/Php platform), to provide seamless access to maternal health records (Veera Boopathy et al., 2024). The system aims to improve security by introducing a rule-based mechanism for early detection and permanent vulnerability fixes. This will be achieved by studying the vulnerability attack response behaviour of six other systems and applying manual Top 10 OWASP attacks on MAMATA CPS. We intend to define the implementation of a systematic and reliable security plan. The plan of the security test aims to prevent attackers from invoking unintended system functionalities.

The organization of the paper is as follows. A literature review of security methods, OWASP concept, and CVSS score is discussed in 'Section 2'. In 'Section 3', we have discussed the solution with the proposed algorithm (i.e., MAMATA Manager/BOT) along with its architecture and behaviour. The test results obtained have been discussed in 'Section 4', and its comparative analysis with reported literature is discussed in 'Section 5'. Finally, the conclusive remarks are provided in 'Section 6'. In the following section, we will go through a quick literature review of the work done by several researchers around the world and the introduction of new security terminology.

2 Literature Review

In this section, a literature review is discussed in three parts: (1) Concept of OWASP, (2) Introduction of CVSS score, and (3) A Brief review of previous attempts at security methods.

2.1. OWASP and Top 10 Vulnerabilities, Risks

OWASP (Open Web Application Security Project) is a global community group dedicated to improving security applications. They provide independent, practical, and freely available information for web

application security. Predominantly, the majority of ethical hackers and security experts adhere to OWASP guidelines. The group believes in open, accessible, and shareable knowledge through its platform. It may be considered like the (World Health Organization (WHO) of web/server security environment. This group has a large size of active members and chapters. OWASP is widely regarded as trustworthy, and developers have grown to rely on it for crucial guidance on online application security. OWASP offers a range of resources, including the "OWASP TOP 10," which is a regularly updated report (OWASP) (comes every four years) that outlines the ten (10) most critical vulnerabilities/ threats to web application security currently present worldwide.

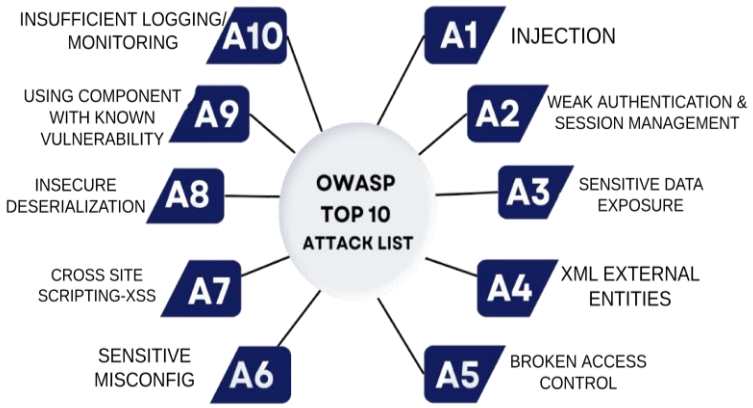

Figure 1: OWASP Top 10 Security Risks to Application

The 'Top 10' refers to an "awareness document"; the Top 10 list provides a baseline for researchers/developers to prioritize security efforts and allocate resources. It is advised that all firms follow/include the 'Top 10 report' during their practices to avoid security challenges, as shown in Figure 1. The top ten types of severe attacks (vulnerability) categories are recorded worldwide (Figure 1, A1 to A10 refers to attacks). The creation of safe applications must include testing for OWASP-identified vulnerabilities.

2.2. CVSS Vulnerability Score

CVSS (Common Vulnerability Scoring System) is a widely used standard for assessing the severity of security vulnerabilities. CVSS assigns severity scores to vulnerabilities (FIRST, 2019), and the score is calculated based on its three major factors, i.e., Base, Temporal, and Environmental; the 'Base' describes the nature of vulnerability (i.e., vulnerability's inherent characteristics that remain constant throughout time and across user environments), 'Temporal group shows potential impact (i.e., susceptibility traits that vary with time), and in contrast, 'Environmental' group addresses the ease of exploiting the vulnerability (i.e., attributes of a vulnerability that are one of a kind to a client's current circumstance). CVSS scores range from 0 to 10, with higher scores indicating more severe vulnerabilities (score band; 8-10: High, 7-5: Medium, and <= 4: Low). The scoring band (of 0-10) could be adjusted by the Temporal and Environmental measurements. Most certified security experts (commonly known as certified ethical hackers) use CVSS as a reference to measure vulnerability.

By assigning a CVSS score to a vulnerability, security experts can prioritize their efforts, allocate resources, and make informed decisions about which vulnerabilities to address first. Additionally, CVSS scores can be used as input to security automation tools, such as intrusion detection systems and vulnerability management systems, to help identify and mitigate security risks in real time. The

importance of CVSS scores in the security analysis of eHealth applications lies in their ability to provide a consistent and objective measure of the danger posed by a particular vulnerability.

2.3. Brief on Security Methods

In the aftermath of covid-19 year, a scheme proposed in (Mahmood et al., 2022) enhances security by preventing physical and de-synchronization attacks, thereby reducing computation and communication costs by over 37% and 40%, respectively. This method effectively secures wearable health monitoring systems. In other research works, eHealth-based software applications' security mechanisms have demonstrated significant progress in implementing the 'Standardized SOA' (Gazzarata et al., 2014) for Clinical Data Interchange (in Cardiac Tele monitoring) and the Security Oriented Design (SOD) Framework (Weider et al., 2014). Some researchers have also attempted security testing methods, but only one vulnerability case (XML Injection) was considered and produced reasonable results. Later, in (Farhadi et al., 2018), researchers presented the static analysis of HIPAA Security requirements in EHR applications with practical issues. Few authors have compared multiple techniques for discovering vulnerabilities (Austin & Williams, 2011) after realizing that one method is not enough. In (Smith & Williams, 2011), the authors aimed to systematize security test planning using functional requirements phrases and proposed five case studies of security attacks. In another research, the authentication mechanism (Shamshad et al., 2022) is commonly regarded as the most effective way for the Telecare-based health service to legitimate patient data and the server. To enhance the security of eHealth/MCPS systems, the authors of this paper tried to focus on two points:

- Addressing flaws in present security certification standards.
- Developing better weakness recognition in large-scale frameworks like EHR, MCPS, and eHealth.

To answer the above first point, studies were conducted with the CCHIT (Certification Commission for Healthcare Information Technology) security criteria (Ragland et al., 2013) to identify weaknesses and improve security. These studies showed that common code-level security risk vulnerabilities were not considered in early eHealth/MCPC development but were later added to the OWASP TOP 10 report in the year 2013. Another study framework (Pai et al., 2021) improves the workflow of health services and treats security criteria as an entrance criterion into the EHR security standardization process, which is crucial for a secure EHR.

The above studies and different result analyses showed that neither single technique discovered all 'OWASP TOP 10' risks in one run, nor could any systems find every type of vulnerability as suggested in the OWASP TOP 10 report 2017/20 (OWASP). In addition, it has been observed that none of the systems have worked for maternal women data and have not analyzed 'TOP 10' security threats. According to the above literature review, systematic manual penetration testing uncovered most security design defects. However, penetration testing (automated) was shown to be majorly effective in finding execution bugs and deliberate manual infiltration testing to find plan flaws. Based on the above studies, we saw and believed that there is a need for the development of a secure healthcare mechanism. This enhancement can be performed by enhancing the security algorithm implementation, which can be done using a combination of Intrusion detection system tools or BOT that monitors vulnerabilities and prevents the system from becoming high-risk. In the following section, we have discussed the proposed secure MCPS system along with the defined architecture, implementation steps, and testing.

3 Method (Secured MAMATA CPS)

MAMATA CPS is an extension of the MCPS system that attempts to position as an eHealth ecosystem between Maternal women, Hospitals (necessarily maternal clinics) and pathology labs to store, maintain, and retrieve maternal records as per standard formats. During the development of the MAMATA Security Mechanism, we have considered and referenced the OWASP-2017/20 report for vulnerability and risk analysis. Figure 1. shows the most high risk vulnerabilities that may affect the security plan. It is also known as one of the top 10 highly risked vulnerabilities as per the OWASP report. It is suggested to all software developers and healthcare researchers that while developing a securing mechanism/algorithm, one must give priority and develop solutions against (the mentioned) Top 10 security risks.

While developing the MAMATA Healthcare CPS framework, we have focused on the solutions to resolve the TOP 10 security risks (i.e., vulnerabilities) proposed by OWASP. This is of utmost importance, because, without security assurance of maternal records, hospitals/pathology labs and maternal women shall not trust and, or accept the MAMATA Healthcare CPS system. The proposed research work is introduced in the current section, wherein a security method/algorithm has been implemented in the MAMATA Healthcare Server. The MAMATA server was already demonstrated and protected (Patented in South Africa and copyrighted in India and Canada) by the authors (Patent no. 2022-04698 South Africa). Here, we extend the same work and propose a new algorithm (like an Intrusion Detection System) that helps to detect/monitor the outside attacks/vulnerabilities experienced by the MAMATA CPS Server. This novel algorithm-based framework/mechanism is named "MAMATA BOT".

3.1. MAMATA BOT

MAMATA BOT's motive is based on the philosophy and objective of intrusion detection systems (IDS) (Khraisat et al., 2019). IDS concept was originally meant to identify vulnerability exploits against a single application. It simply identifies threats; due to its out-of-band network infrastructure presence, it is not part of the true real-time communication path between sender and listener. The IDS monitors traffic and informs an administrator of its findings, as well as taking automatic action to prevent a suspected exploit from seizing control of the system. IDS is intended for use in a variety of situations. Hence, it has different security solutions deployments, e.g. host-based, web-based, signature, anomaly, or hybrid-based. Wherein looking towards the scope of the MAMATA CPS System (conceptual, as shown in Figure 2), authors have implemented a novel algorithm limited to web-based. The MAMATA CPS (in Figure 2) has created the MAMATA NMDBS server (i.e. National Maternal Database Server), which aims to address maternal care issues by connecting three key components: maternal hospitals, pathology labs, and maternal women.

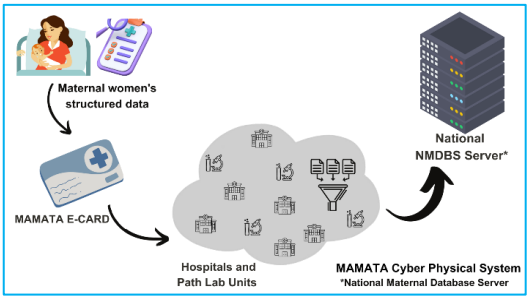

Figure 2: MAMATA CPS Concept Implementation in the Health Sector

The server collects data from these sources and updates the respective maternal women's accounts, which have unique credentials for access by the women themselves, hospitals, and labs.

3.1.1. Architecture and Configurations

During each request hit of end users, MAMATA BOT periodically reads the log file (powered by the proposed AEBAS identifier, i.e. Aadhar Enabled Biometric System) created by the MAMATA CPS NMDBS (National Maternal Database) Server as shown in Figure 3. The security mechanism reads requests from three user nodes: maternal women, hospitals, and pathology labs. After validating the incoming requests, it forwards them to the dedicated database that corresponds to the nature of the user query. MAMATA NMDBS Server is active on a cloud-based droplet. It has three different database clusters (i.e. Women DB, Hospital DB, and Path Lab DB) to maintain easy and quick database query executions.

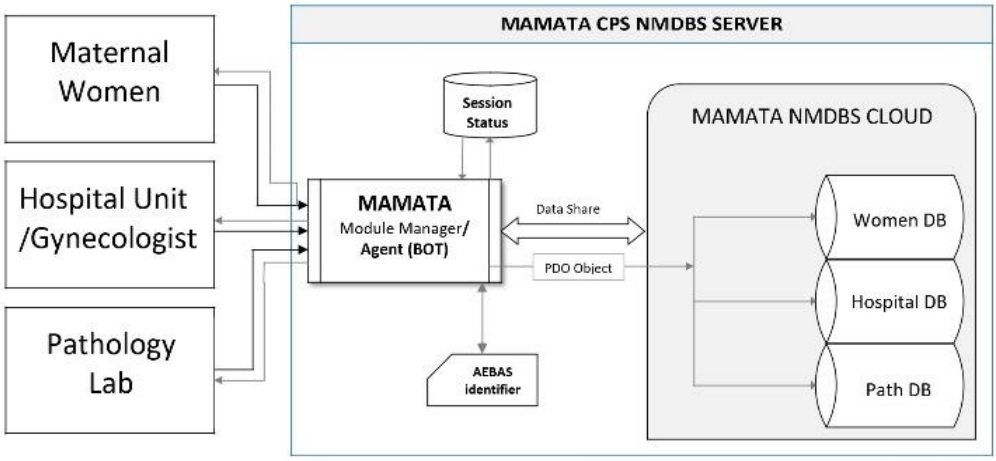

Figure 3: The Designed and Developed MAMATA BOT CPS NMBDS Server for Maternal Health Applications

The server used by the NMDBS application is Apache. The operating system used to run this server is Ubuntu OS (Linux). The log file is generated at the location "/var/log/apache2/access.log," named to be the location of the AEBAS identifier. However, like every log file, this may also face a few shortcomings, as mentioned below.

- The access log may contain the 'body of the request'.
- The access log may not contain 'headers'.

So, to overcome these shortcomings, we have introduced the concept of an open-source firewall that facilitates an array of hypertext transfer protocol (Chen & Cheng, 2016) requests with capabilities of response filtering. It may be referred to as a security model named 'Mod- Security'. This is installed and configured on the Apache server. This ModSec can be either used as a firewall or an audit tool. We used/treated it as an audit tool to collect all the web application request-related data.

The "modsecurity.conf" file is located in the "/etc/modsecurity/" directory. Then, the audit engine is enabled, and log body requests are merged. This configuration has been done to keep track of every transaction and overcome the common issues faced by web applications. The snippet of the same can be read in the appendix.

Log parsing and Control: The logs of an eHealth application contain valuable information about user activity and system events, which can be used to identify security threats and vulnerabilities. By parsing and controlling the logs, security personnel can monitor for suspicious activity and act to prevent security incidents before they occur. We attempted to apply the same at the Apache server, wherein the user can access the server through the required successful credentials, and access is controlled with following parsing that has been included in the architecture itself (discussed further in the algorithm). Here, MAMATA BOT has been designed with the following log-parsing format.

- Section A- *Address/Details (e.g., Timestamp, source & destination IP and port no)*
- Section B- *Request Header*
- Section C- *Request body (if any)*

The MAMATA BOT parses this log file and checks for any attempts made to attack the web application. It checks for the URL, the method, and other headers for injection attacks, and mainly, it checks for various attacks sent via the request. Now, MAMATA BOT is developed to perform vulnerability checks against OWASP's Top 10 list as mentioned earlier.

3.2. MAMATA BOT Functions and Algorithm

Here at the outset, MAMATA BOT functions (Figure 4) act like security module manager that listens to the user's request (Maternal woman, Hospital units, and Path labs) and check the session status from (i.e. Log-in /Off), if the session flag is positive, the next task is to verify the authenticity of the request using the Log file present inside the AEBAS identifier (for demonstration purposes, AEBAS is demonstrated with test data/log only).

Furthermore, the Module Manager (i.e. BOT) uses a novel algorithm (discussed separately) and performs the check against any suspicious activity/threats as defined in OWASP Top 10. This algorithm is a type of intrusion detection and prevention i.e., used to detect and prevent unauthorized access or malicious activity on the application system. Generally, it includes include network-based detection, signature-based detection, and/or behavioral-based detection, However, the authors proposed a rule-based algorithm, as discussed below.

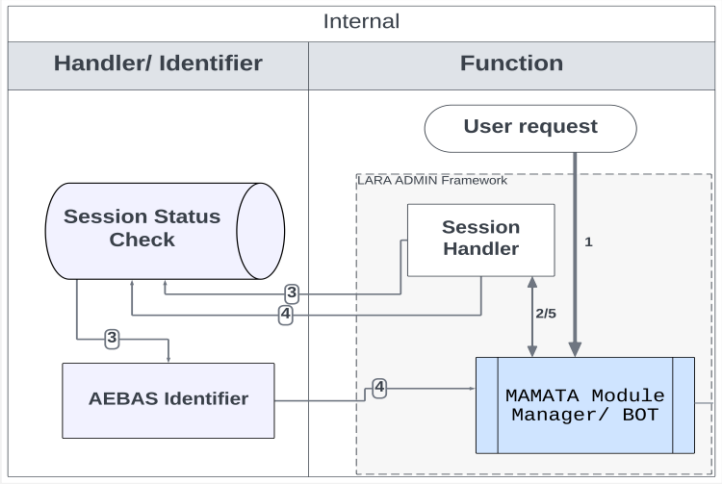

Figure 4: MAMATA Security Module/BOT Functional Operations

- **Rule-based Algorithm**

Well-defined rule checks are incorporated in MAMATA BOT as shown in Figure 5. The algorithm behavior of Manager/BOT, initially, acts to parse logs from the "modsec_audit.log" file and performs checks to detect any attack (if it has happened). Once the log file is parsed, it checks for tampering of the requests viz; URL, headers, and the request body. Further, it checks for the response body for sensitive information leakage (if any). The below steps provide a more mathematical/scientific representation of the 'Rule-based algorithm' for security analysis of the MAMATA CPS healthcare system as depicted in Figure 5. The steps are as below:

Step 1: Input Data Representation (log monitor).

Step 2: Rule-based Analysis.

Step 3: Pattern Matching (Section A, B, & C validation i.e. execute the rules of vulnerability).

Step 4: Alert Generation against failure of condition.

Step 5: Response Mechanism.

Step 6: Feedback Loop (follow steps 2-5 for every request received).

The algorithm introduces some formulaic elements to describe its functioning as briefed below:

Step 1- Input Data Representation (log monitor)

This Input log file i.e. 'modsec_audit.log' contains information about requests made to a server. The input data representation equation $\boldsymbol{X} = \{x_1, x_2, x_3 \ldots x_n\}$ defines a vector where each element represents a specific attribute or feature of the healthcare system's data. For example, x_1 could represent the number of failed login attempts, while x_2 could represent the frequency of data access requests.

Step 2- Rule-based Analysis

This step examines each request in the log file for tampering (if any) and checks; if it meets the rule format, request tampering is analyzed using rule:

$$r_j : \textbf{IF } (\text{condition}_j) \textbf{ THEN } (\text{condition}_j)$$

Consists of a condition that, when met, triggers a specific action. For instance, a rule could be: **IF** the number of failed login attempts exceeds 5, or the request format tampers **THEN** block the user account.

Step 3- Pattern Matching

As discussed earlier the algorithm has a specific rule pattern represented in sections A, B, and C to validate the nature of the input request. The pattern matching function evaluates.

$$\mathbf{f(X, R)} = \sum_{\mathbf{j=1}}^{\mathbf{m}} \mathbf{Match}\left(\mathbf{x}, \mathbf{r_j}\right)$$

How well the input data aligns with the rules. For example, if a rule states that IF data access requests exceed a certain threshold, ***THEN*** generate an alert, the matching function assesses the extent to which this condition is met. In equation; 'R' denotes the section rules *(r_j)* represent sections A, B, and C as shown in Figure 5, wherein section A holds the source IP address and timestamp pattern, section B saves the URL along with any other request headers, and in section C, the algorithm stores request body.

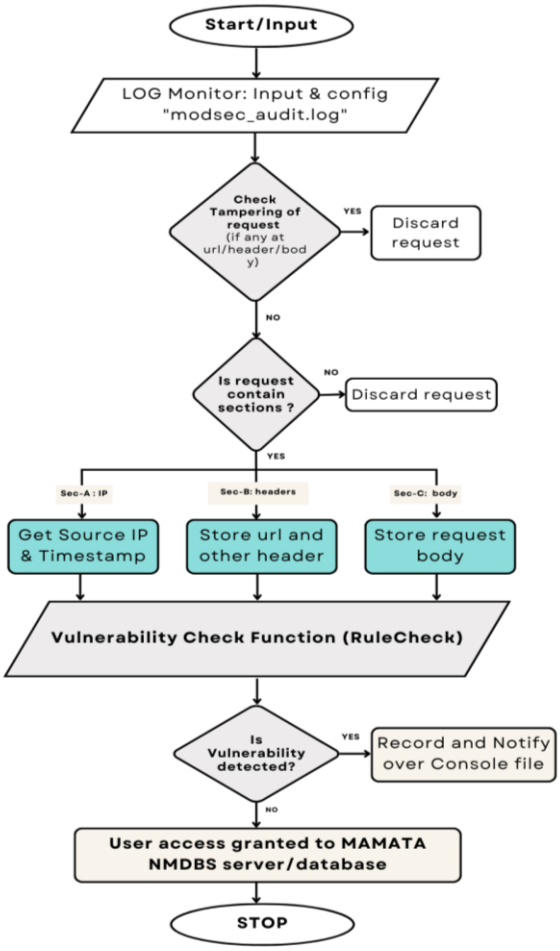

Figure 5: MAMATA Module Manager/BOT Behaviour, (Based on Rule-based Algorithm)

Step 4- Alert Generation

The alert generation equation:

$$\mathbf{A}\big(\mathbf{f}(\mathbf{X},\mathbf{R})\big) = \begin{cases} \text{Alert}, & \text{if } f(X,R) > \text{Threshold} \\ \text{No Alert}, & \text{otherwise} \end{cases}$$

Determines whether an alert should be raised based on the matching results. For instance, if the sum of matches exceeds a predefined threshold, an alert is triggered to indicate a potential security issue. In other words, if the pattern summation differences for sections A, B, and C match and exceed the threshold limit, the alert is triggered.

Step 5- Response Mechanism

The response mechanism function:

$$\mathbf{M}(\mathbf{A}) = \begin{cases} \text{Mitigation strategy}, & \text{if Alert is triggered} \\ \text{No action}, & \text{if No Alert} \end{cases}$$

defines the action taken in response to alerts. For example, if an alert is generated due to a suspicious pattern in data access requests, a mitigation strategy like blocking the source IP address could be implemented.

Step 6- Feedback Loop

The feedback loop equation:

$$\textbf{New Rule}_{j} = (\textbf{Old Rule}_{j} + \alpha \times \textbf{Feedback})$$

Updates rules based on feedback received from previous analyses and responses. This iterative process allows the algorithm to adapt and improve over time based on the outcomes of security assessments.

- Example scenario (for failed login attempt):
  - Vulnerability: Consider a vulnerability where an unauthorized user repeatedly attempts to access patient's sensitive data within the healthcare system.
  - Application of Equations:
    - Input Data: $\boldsymbol{X} = \{Failed\ Login\ Attempts, Data\ Access\ Requests\}$
    - Rule: IF Failed Login Attempts > 5 THEN Block User Account, if not THEN check pattern.
    - Pattern Matching: Evaluate how well the input data aligns with rule conditions.
    - Alert Generation: Trigger an alert if the sum of matches exceeds a predefined threshold.
    - Response Mechanism: Implement a mitigation strategy such as blocking the unauthorized user's access.
    - Feedback Loop: Update rules based on the effectiveness of the response to prevent similar vulnerabilities in the future.

Normally, an algorithm is used to check for known general vulnerabilities in the requests and notify the admin if it finds any, which is a common task in web application security. However, in the proposed case, a rule-based algorithm uses a set of predefined rules to identify suspicious activity for the eHealth application. These rules are based on known OWASP-recommended threats and attack patterns, unlike the general known security threats. We observed that this exercise gives unique benefits for system performance including simplicity, speed, flexibility, low-false positive,e and the ability to run over low computational resources, as discussed below,

- Simplicity: The MAMATA BOT algorithm is relatively simple and easy to understand, implement, and maintain and essential for securing maternal data in eHealth applications.
- Speed: It uses OWASP threat definitions and rule formats to process huge volumes of data efficiently, making it well-suited for real-time intrusion detection and prevention.
- Flexibility: Considering the constantly changing security landscape, the proposed eHealth application built on the 'Laravel framework', the rule-based algorithm's ability to swiftly adapt to new threats is essential.
- Low-false positives: The algorithm, based on a rule check, minimizes false positives.
- Low-computational resources Because rule-based patterns use less computing power, the suggested technique is better suited for systems with constrained resources. This is crucial in eHealth applications, especially in rural areas where hospitals and path labs may have limited resources.

Authors have observed that technical security threats as defined by OWASP have also been discovered. For these purposes, a pen test was performed i.e., security attacks on the MAMATA BOT

server were employed, and the results were enhanced. The same has been compared with SOA (service-oriented architecture), SOD (Service Oriented Design) frameworks, and some of the other eHealth systems (Soceanu et al., 2013). For pen-testing, several tools such as 'Burp Suite Proxy', 'Nikto' along with Browser, have been utilized. Burp Suite Proxy is a graded industry-standard tool that is appropriate for testing web-based applications because it uses a web penetration framework on the Java platform. It is capable of crawling web-based applications automatically. 'Nikto' is an open-source web server scanner that runs extensive tests against web servers for a variety of objects, including over 7000 potentially harmful files/programs. Nikto is not intended to be a stealthy tool. It will run a web server test as quickly as feasible. These tools cover many features including SSL, HTTP Proxy support, IDS encoding approach, simple updates, authentications, and Apache server enumeration. Both tools are compatible with Windows, OS X, and Linux.

To summarize, the proposed algorithm aims to cater to security risks and will only be accepted or piloted if it passes a security assessment (penetration) test conducted by a certified security expert. Thus, to prove the competency of this system, the authors presented the respective test results with security risk scores in the next section 4.

4 Test Results

As mentioned earlier, 'OWASP TOP 10' standard vulnerabilities (OWASP) have been taken as a reference for security simulation and design process. A 'Pen-Test' was conducted by two (2) different security experts at regular intervals. Table 1 shows the status of the system with a normal Pen-Test i.e. before the application of the security algorithm and observed status. The common issues (shown in Table 1) have been resolved after the incorporation of MAMATA BOT.

Table 1: Pen-test Results without MAMATA BOT

No	Vulnerabilities (OWASP)	MAMATA CPS
1	Injection	No
2	Broken authentication	Found
3	Sensitive data exposure	Found
4	XML external entities (XXE)	No or N/A
5	Broken access control	Found
6	Security misconfiguration	Found
7	Cross-site scripting XSS	No
8	Insecure deserialization	No
9	Using components with known vulnerabilities	Found
10	Insufficient logging & monitoring	N/A

Note: Found=Vulnerability found, NO= Vulnerability Not Found, N/A= Not applicable

Table 2: 1st Pen-Test, Pre-& Post MAMATA BOT Usage

No	Vulnerability names (OWASP)	Before	After
1	Injection	No	No
2	Broken authentication	Yes	Yes
3	Sensitive data exposure	Yes	No
4	XML external entities (XXE)	No	No
5	Broken access controls	Yes	No
6	Security misconfigurations	Yes	No
7	Cross-site scripting XSS	No	No
8	Insecure deserializations	No	No
9	Using components with known vulnerabilities	Yes	No
10	Insufficient logging & monitoring	N/A	N/A

Note: Yes=Vulnerability found, NO= Not Found, N/A= Not applicable.

The Pen-Test was performed by the first (1st) security expert after the incorporation of MAMATA BOT and a comparison was carried out as shown below in Table 2. It has been observed that the MAMATA CPS system is comparatively secure and stable for most of OWASP TOP 10 vulnerabilities, except broken authentication. Subsequently authors conducted a second test with the help of another security expert (ethical hacker). This time Pen-Test was carried out with intense attack time, and supported by CVSS (Common Vulnerability Scoring System) score generation. The results of the second Pen-Test along with comments are shown in Table 3. Here note that action is performed on found vulnerabilities in the second pen test, as per the comments/observations in the test report.

Table 3: Result of Second (2^{nd}) Pen-Test

No	Standard Vulnerabilities	MAMATA CPS	Comments/Observations
1	Injection	No	Not found (i.e., secure).
2	Broken authentication	Found	Minor Authentication leak
3	Sensitive data exposure	Found	-Application reveals Apache server version & operating system. -Attackers can use this information for further attack strategy.
4	XML external entities (XXE)	No or N/A	The application does not use XML to send/receive requests and responses.
5	Broken access control	No	Not found (i.e., secure).
6	Security misconfiguration	Found	The application is reproducing with stock tracked. An Attacker may use this to read code, as shown.
7	Cross-site scripting XSS	No	Not found (i.e., secure).
8	Insecure deserialization	No	Not found (i.e., secure).
9	Using components with known vulnerabilities	Found	The application uses Apache 2.4.29 with some known vulnerabilities.
10	Insufficient logging & monitoring	N/A	Not found (i.e., secure).

After the pen testing, the severity of vulnerabilities was measured using the CVSS score matrix (as discussed in the literature survey). This scoring system has been used to prioritize responses and resources according to the threat observed. Scores are calculated based on a formula that depends on several metrics of vulnerabilities that approximate the ease of risk exploit and the impact of risk exploit. Scores range from 0 to 10, with 10 being the most severe. CVSS scores reported against vulnerabilities are shown in Table 3a.

Table 3a: CVSS Score Reported Against Found Vulnerability in the Second Pen-Test

No.	**Found Vulnerabilities**	**CVSS Score/ Rating**	**CVSS Risk**
1	Sensitive data exposure Full data disclosure	5.4 2.9	Medium
2	Security misconfiguration	2.8	Low
3	Using components with known vulnerability	2.7	Low

Furthermore, based on the suggestions and observations reported in the second pen test, relevant actions have been taken in the MAMATA BOT and/or directly in MAMATA CPS system or System code, configurations wherever applicable. Table 4 shows the comparative summary by both security

experts and suitable actions taken by us to correct the findings. It is to be noted that test simulations were carried out by security experts under different environments (i.e. browser with Windows, Linux), using different test tools (viz Burp Suite Proxy', 'Nikto') and at different times of network traffic.

The pen test was run for four (4) days at regular intervals & the following recommendations have been drawn out for the application under assessment issues.

- Upgrade to the latest version of the Apache web server.
- Disable the debug option in Laravel.
- Do not disclose the server name and version number in the response.
- A review should be conducted post-implementation of recommended countermeasures to ensure that the vulnerabilities are fixed or closed.

Subsequent actions against the suggestions have been taken and updated in Table 4.

Table 4: Summary - Comparative of Vulnerability Findings with Suggestions and Actions

No.	Standard Vulnerabilities	By Expert-1	By Expert-2	Suggested Actions taken
1	Injection	No	No	-
2	Broken authentication	Yes	Yes	Best practice of Captcha /OTP planned.
3	Sensitive data exposure	No	No	Cloud setting configured.
4	XML external entities (XXE)	No	No- N/A	XML is not used hence not applicable.
5	Broken access control	No	No	N/A
6	Security misconfigurations	No	Yes	Tracing done.
7	Cross-site scripting XSS	No	No	-
8	Insecure deserialization	No	No	-
9	Using components with known vulnerabilities	No	Yes	The old Apache version shall be changed.
10	Insufficient logging & monitoring	No	No	

5 Observations and Comparative Analysis

The effectiveness of the suggested technique is demonstrated by a comparison of the acquired results with those from previous studies. As a consequence, Table 5 (i.e. comparative analysis) illustrates how the results of the suggested "MAMATA BOT" were compared with the published literature in terms of vulnerability parameters. In comparisons, it is observed that the work proposed has an upper edge over the other research work.

In Table 5, it is seen that in the proposed work no multi vulnerabilities/threats or scope of attack were found. If threats were spotted (if any) using the previously disclosed unique intrusion detection technique, they were fixed and/or prioritized. On the flip side of comparisons in (Gazzarata et al., 2014), the work attempted for standardized SOA wherein the system does not cater to all vulnerabilities, hence, limited to cardiac data format only, and there is no proof of known vulnerabilities. In (Weider et al., 2014), the system framework is designed with authenticity and encryption of input data only (i.e., Injection of OWASP); hence, provision for remaining parameters viz CSRF and Broken Access Control are missing. The static risk analysis for EHR applications has covered some threat detections, but a solution needs to be mentioned (Odilov et al., 2024). Similarly, in (Salas et al., 2015; Austin & Williams,

2011; Smith & Williams, 2011), other threats like 'XML external entities (XXE)', 'Broken authentication', 'Using components with known vulnerabilities' & 'Sensitive data exposure', are being detected. However, the clear plan for fixing such threats as per OWASP has not been touched.

The proposed method, due to the application of a novel security mechanism, no such vulnerability is detected, which can be seen in Table 5. As discussed earlier, the proposed security mechanism is based on A rule-based Intrusion Detection System (IDS) is a type of IDS that uses a set of predefined rules or signatures (formatted as Section A, B, C) to identify potential security threats. The mechanism works by comparing incoming requests or system activity against a database of known security threats as per OWASP Top 10. If the request format gets mismatched or is different than the rules; the system records such activity as an abnormality and notify in the log trace. This reveals that the proposed work is comparatively fullproof and effective compared to the six references of different works mentioned in Table 5.

Table 5: Obtained Result in Cumulative Comparisons (with other systems/articles) for Pen Test-1

No	Standard Vulnerability name	MAMATA Healthcare CPS (Proposed)	SOA Cardiac (Gazzarata et al., 2014)	SOD Framework (Weider et al., 2014)	Marcelo Test (Salas et al., 2015)	Maryam Analysis (Odilov et al., 2024)	Andrew Austin Tech (Austin & Williams, 2011)	Ben Smith Test (Smith & Williams, 2011)
1	Injection	✓	✓	✓	✓	✓	✓	✓
2	Broken authentication	*	*	*	*	*	*	*
3	Sensitive data exposure	✓	*	*	*	*	*	*
4	XML external entities (XXE)	✓	*	*	*	*	*	*
5	Broken access control	✓	*	*	*	*	*	*
6	Security misconfigurations	✓	*	*	✓	*	*	*
7	Cross-site scripting XSS	✓	*	*	✓	*	*	*
8	Insecure deserialization	✓	*	*	*	*	*	*
9	Using components with known vulnerabilities	✓	*	*	✓	*	*	*
10	Insufficient logging & monitoring	✓	*	*	*	*	*	*

Note: ✓ = Secured/Vulnerability fixed (if detected); * = Vulnerability detected and/or could not be fixed /did not touch.

It is noted while comparing that; the suggested MAMATA BOT model has detected and fixed majorly 7 (directly) and 2 (indirectly) OWASP TOP 10 vulnerabilities (except the broken authentication, and cloud version compatibility). In contrast, other methods like SOA cardiac, and SOD framework has detected 1 out of 10 i.e. 10% detection, Marcelo test could able to detect/fix four (04) vulnerabilities (i.e. 40% of OWASP index). Andrew and Ben's test can detect and/or fix only one (01) or two (02) vulnerabilities (i.e. 10-20% detection ratio). Considering the authors proposed model of 7/9 fixes there is a clear average of 60-70% improvement. Additionally, these six studies found that, neither they are targeting maternal women's health data, nor security vulnerabilities as per OWASP. They are addressing

general patient health data. This demonstrates the effectiveness of the proposed methodology that we provide. While it is effective in detecting known security threats, the proposed method may not be as effective in detecting new or unknown security threats or threats which have no stable pattern or need manual configurations (i.e. broken authentication that needs real-time captcha, security misconfiguration, etc.). Hence, a few of the vulnerabilities have not been fixed.

It is also observed that this security mechanism is not compatible with frameworks developed in programming languages other than PHP, as the presented work is based on the Laravel framework, which is PHP-based. However, it can be used with other different frameworks only if they follow the PHP standards. This can be future scope for new researchers.

6 Conclusion

The sensitive and personal nature of health information in eHealth systems demands a robust security mechanism to safeguard it from unauthorized access, tampering, and theft. This becomes more crucial in the case of the maternal eHealth system. Our principal objective was to address the challenges in maternal treatment and develop a solution by integrating security mechanisms into electronic medical applications for maternal health issues. To achieve this goal; the implementation of a new security mechanism (i.e., MAMATA BOT) was introduced. This mechanism is based on the concept of a rule-based Intrusion Detection System (IDS) that provides a powerful tool to protect against known security threats and detect anomalies in system behavior that may indicate potential security breaches. Compared to other security measures, the proposed security mechanism is capable of identifying attacks that have been known and noticed in OWASP's top 10 threats.

The superior results of this mechanism are due to its ability to identify and adapt to known threats and attack patterns over time, ensuring a higher level of security for eHealth systems. However, it can be less effective in detecting unknown security threats or the threats that need to be listed in OWASP. The threats are only updated when a new threat is identified and considered as a highly risky threat by OWASP. Moreover, we encourage healthcare researchers to prioritize the implementation of proposed security mechanisms as security measures in their eHealth systems to protect the confidentiality, integrity, and availability of sensitive patient data." Subsequently, this generated research report will undoubtedly operate as a ready bookkeeper reference or as a foundation for future research. Finally, in this research article, a short prologue to the MAMATA security mechanism was introduced in the pictorial structure. To achieve high reliability two security test were performed, wherein 2nd Test, findings were acquired, and compared with previous results.

7 Patents

- The research work (security method) is patented in South Africa, in 2022. With Patent registration no. 2022/04698.
- The extracts of the research work are also copyrighted in two (02) other countries named India (Reg no: L-101275/2021 and Canada (Reg no: 1187029).

Author Contributions

Author 1: Conceptualization, Methodology, Software, Field study
Author 2: Data curation, Writing- Review draft preparation, Software validation.
Author 3: Visualization, Investigation, Writing-Reviewing and Editing.

Acknowledgments

This research work-related pen test was performed by two security experts named 1. Company-HACK-X Security Systems Pvt ltd and 2. Certified ethical hacker Mr. Hrishikesh. This research work is registered with Gujarat Technological University, Chandkheda, Ahmedabad, Gujarat (India). The researchers would like to thank all of them who have helped them in the successful completion of the writing of the research paper.

Conflicts of Interest

The authors declare **no conflicts of interest** in the design of the study or the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results".

Appendix A

1

```
apt install-y
libapache2-mod-security2
```

Code Snippet 1: Security Module Configuration

2

```
SecAuditEngine On
```

Code Snippet 2: Enable Audit Engine

3

```
SecAuditLogParts BC
```

Code Snippet 3: Only Log request and Body

4

```
SecAuditLogType Serial
SecAuditLog /var/log/apache2/ modesec_audit.log
```

Code Snippet 4: Use of single/common file for logging to avoid delays in analysis.

References

[1] Alhumud, M. A., Hossain, M. A., & Masud, M. (2016, July). Perspective of health data interoperability on cloud-based medical cyber-physical systems. In *2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICMEW.2016.7574715

[2] Austin, A., & Williams, L. (2011, September). One technique is not enough: A comparison of vulnerability discovery techniques. In *2011 International Symposium on Empirical Software Engineering and Measurement* (pp. 97-106). IEEE. https://doi.org/10.1109/ESEM.2011.18

[3] Chen, J., & Cheng, W. (2016, September). Analysis of web traffic based on HTTP protocol. In *2016 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)* (pp. 1-5). IEEE. https://doi.org/10.1109/SOFTCOM.2016.7772120

[4] Clarke, M., de Folter, J., Verma, V., & Gokalp, H. (2017). Interoperable end-to-end remote patient monitoring platform based on IEEE 11073 PHD and ZigBee health care profile. *IEEE Transactions on Biomedical Engineering*, *65*(5), 1014-1025. https://doi.org/10.1109/TBME.2017.2732501

[5] Farhadi, M., Haddad, H., & Shahriar, H. (2018, July). Static analysis of hippa security requirements in electronic health record applications. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)* (Vol. 2, pp. 474-479). IEEE. https://doi.org/10.1109/COMPSAC.2018.10279

[6] FIRST, E. (2019). Common vulnerability scoring system version 3.1: Specification document. https://www.first.org/cvss

[7] Garkoti, G., Peddoju, S. K., & Balasubramanian, R. (2014, December). Detection of insider attacks in cloud based e-healthcare environment. In *2014 International Conference on Information Technology* (pp. 195-200). IEEE. https://doi.org/10.1109/ICIT.2014.43

[8] Gazzarata, R., Vergari, F., Cinotti, T. S., & Giacomini, M. (2014). A standardized SOA for clinical data interchange in a cardiac telemonitoring environment. *IEEE journal of biomedical and health informatics*, *18*(6), 1764-1774. https://doi.org/10.1109/JBHI.2014.2334372

[9] Haque, S. A., Aziz, S. M., & Rahman, M. (2014). Review of cyber-physical system in healthcare. *international journal of distributed sensor networks*, *10*(4), 217415. https://doi.org/10.1155/2014/217415

[10] https://www.who.int/about

[11] Hu, L., Qiu, M., Song, J., Hossain, M. S., & Ghoneim, A. (2015). Software defined healthcare networks. *IEEE Wireless Communications*, *22*(6), 67-75. https://doi.org/10.1109/MWC.2015.7368826

[12] Kakavand, M., Mustapha, A., Tan, Z., Yazdani, S. F., & Arulsamy, L. (2019). O-ADPI: online adaptive deep-packet inspector using Mahalanobis distance map for web service attacks classification. *IEEE Access*, *7*, 167141-167156. https://doi.org/10.1109/ACCESS.2019.2953791

[13] Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, *2*(1), 1-22. https://doi.org/10.1186/s42400-019-0038-7

[14] Kodric, Z., Vrhovec, S., & Jelovcan, L. (2021). Securing edge-enabled smart healthcare systems with blockchain: A systematic literature review. *Journal of Internet Services and Information Security*, *11*(4), 19-32. https://doi.org/10.22667/JISIS.2021.11.30.019

[15] Lee, I., Sokolsky, O., Chen, S., Hatcliff, J., Jee, E., Kim, B., ... & Venkatasubramanian, K. K. (2011). Challenges and research directions in medical cyber–physical systems. *Proceedings of the IEEE*, *100*(1), 75-90. https://doi.org/10.1109/JPROC.2011.2165270

[16] Lydia, M., Prem Kumar, G. E., & Selvakumar, A. I. (2023). Securing the cyber-physical system: A review. *Cyber-Physical Systems*, *9*(3), 193-223. https://doi.org/10.1080/23335777.2022.2104378

[17] Mahmood, K., Obaidat, M. S., Ghaffar, Z., Alzahrani, B. A., Shamshad, S., Saleem, M. A., & Hussain, S. (2022). Cloud-assisted secure and cost-effective authenticated solution for remote wearable health monitoring system. *IEEE Transactions on Network Science and Engineering*, *10*(5), 2710-2718. https://doi.org/10.1109/TNSE.2022.3164936

[18] Malasinghe, L. P., Ramzan, N., & Dahal, K. (2019). Remote patient monitoring: a comprehensive study. *Journal of Ambient Intelligence and Humanized Computing*, *10*, 57-76. https://doi.org/10.1007/s12652-017-0598-x

[19] Neelima, S., Govindaraj, M., Subramani, D. K., ALkhayyat, A., & Mohan, D. C. (2024). Factors influencing data utilization and performance of health management information systems: A case study. *Indian Journal of Information Sources and Services*, *14*(2), 146-152. https://doi.org/10.51983/ijiss-2024.14.2.21

[20] Odilov, B. A., Madraimov, A., Yusupov, O. Y., Karimov, N. R., Alimova, R., Yakhshieva, Z. Z., & Akhunov, S. A. (2024). Utilizing Deep Learning and the Internet of Things to Monitor the Health of Aquatic Ecosystems to Conserve Biodiversity. *Natural and Engineering Sciences*, *9*(1), 72-83. https://doi.org/10.28978/nesciences.1491795

[21] Pai, M. M., Ganiga, R., Pai, R. M., & Sinha, R. K. (2021). Standard electronic health record (EHR) framework for Indian healthcare system. *Health Services and Outcomes Research Methodology*, *21*(3), 339-362. https://doi.org/10.1007/s10742-020-00238-0

[22] Pradeepkumar Dhage et. al, Copyright No: L-101275/2021 (India), 1187029 (Canada), Patent no. 2022-04698 (South Africa) dtd 05-04-2021, 06-10-2021, 29-06-2022 respectively.

[23] Ragland, A., Yuan, X., & Jones, B. (2013, April). Analyzing the relationship between CCHIT certification criteria and HIPAA. In *2013 Proceedings of IEEE Southeastcon* (pp. 1-5). IEEE. https://doi.org/10.1109/SECON.2013.6567455

[24] Salas, M. I. P., De Geus, P. L., & Martins, E. (2015, June). Security testing methodology for evaluation of web services robustness-case: XML injection. In *2015 IEEE World Congress on Services* (pp. 303-310). IEEE. https://doi.org/10.1109/SERVICES.2015.53

[25] Shamshad, S., Ayub, M. F., Mahmood, K., Kumari, S., Chaudhry, S. A., & Chen, C. M. (2022). An enhanced scheme for mutual authentication for healthcare services. *Digital Communications and Networks*, *8*(2), 150-161. https://doi.org/10.1016/j.dcan.2021.07.002

[26] Shamshad, S., Mahmood, K., Hussain, S., Garg, S., Das, A. K., Kumar, N., & Rodrigues, J. J. (2021). An efficient privacy-preserving authenticated key establishment protocol for health monitoring in industrial cyber–physical systems. *IEEE Internet of Things Journal*, *9*(7), 5142-5149. https://doi.org/10.1109/JIOT.2021.3108668

[27] Shar, L. K., & Tan, H. B. K. (2012). Auditing the XSS defence features implemented in web application programs. *IET software*, *6*(4), 377-390. https://doi.org/10.1049/iet-sen.2011.0084

[28] Smith, B., & Williams, L. A. (2011). *Systematizing security test planning using functional requirements phrases*. North Carolina State University. Dept. of Computer Science.

[29] Soceanu, A., Egner, A., & Moldoveanu, F. (2013, May). Towards interoperability of eHealth system networked components. In *2013 19th International Conference on Control Systems and Computer Science* (pp. 147-154). IEEE. https://doi.org/10.1109/CSCS.2013.69

[30] Sofiene, M., Souhir, C., Yousef, A., & Abdulrahman, A. (2023). Blockchain Technology in Enhancing Health Care Ecosystem for Sustainable Development. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 14*(3), 240-252. https://doi.org/10.58346/JOWUA.2023.I3.018

[31] Veera Boopathy, E., Peer Mohamed Appa, M. A. Y., Pragadeswaran, S., Karthick Raja, D., Gowtham, M., Kishore, R., Vimalraj, P., & Vissnuvardhan, K. (2024). A Data Driven Approach through IOMT based Patient Healthcare Monitoring System. *Archives for Technical Sciences, 2*(31), 9-15. http://dx.doi.org/10.70102/afts.2024.1631.009

[32] Weider, D. Y., Davuluri, L., Radhakrishnan, M., & Runiassy, M. (2014, July). A security oriented design (SOD) framework for ehealth systems. In *2014 IEEE 38th international computer software and applications conference workshops* (pp. 122-127). IEEE. https://doi.org/10.1109/COMPSACW.2014.132

[33] www.OWASP.org, this work is licensed under a Creative Commons Attribution Share Alike 4.0 International License.

[34] Zhang, Y., Qiu, M., Tsai, C. W., Hassan, M. M., & Alamri, A. (2015). Health-CPS: Healthcare cyber-physical system assisted by cloud and big data. *IEEE Systems Journal*, *11*(1), 88-95. https://doi.org/10.1109/JSYST.2015.2460747

Authors Biography

Pradeepkumar C. Dhage, is a research scholar at Gujarat Technological University, Chandkheda, Ahmedabad, Gujarat (India). He has developed more than seven products in IOT, ITES, and healthcare domains. He holds Two Copyrights (India and Canada each) and Three patents to date. He has expertise in Prototype to Product development with GTM skills. He has more than 16 years of experience in Innovation, Product development, and Policy initiatives at national level. He has B.E (ECE), ME (ECE) and pursuing Ph.D.

Rajesh A. Thakker, is presently working as a Director, R & D Cell, GTU, Ahmedabad, India. Before this, he was a Principal and Professor in EC Engineering at Government Engineering College, Rajkot. He received his Ph.D. degree in VLSI from IIT Bombay and M. Tech. Degree in Electrical Engineering from the Indian Institute of Technology (IIT), Bombay. He has more than 26 years of academic experience. His research interests include CMOS Analog Circuit Design, MOS Modelling, Testing and Verification, Image Processing, Deep Learning, and Applications of Evolutionary Algorithms in the field of VLSI.

Krishna K. Warhade, is born in Maharashtra, India. He completed his Ph.D. from IIT Bombay with a specialization in communication and signal processing. He has 25 years of teaching experience. His research interests are Computer vision, biomedical signal processing, precision agriculture, and healthcare. He has contributed papers at various reputed journals and conferences. He has worked as a Dean of Research Development Innovation and Consultancy and currently working as Director for the doctoral program at Dr. Vishwanath Karad MIT World Peace University, Pune.