MyHealthcare™

MyHealthcare Enterprise Application (MHEA)

WASA Whitepaper 2025 v1

Ref: WASA Certificate No: IMS/2025-2026/089 (NABH Assessment — Clause DAC.2.a)

SHA256 Hash: 0ad41814526204c2075422641423e1a9b0d2ecc63835979e70685cff774c5a5d

This is a Controlled Document

In line with MyHealthcare's regulatory obligations, changes to controlled documents must be approved or merged by a control owner. All contributions are welcome and encouraged.

Security Whitepaper
Data Encryption & Protection for MHEA
(MyHealthcare Enterprise Application)

Document Change History:

Version	Date	Author	Reviewer	Changes Done
1.0	25-Nov-2025	Aneesh Nair, CIO, MyHealthcare	Shyatto Raha, CEO, MyHealthcare	WASA White Paper Created as per NABH Audit requirement. NABH Assessment — Clause DAC.2.a. Ref: WASA Certificate No: IMS/2025-2026/089 SHA256 Hash: 0ad41814526204c2075422641423e1a9b0d2ecc63835979e70685cff774c5a5d

Note: This document shall be reviewed only upon significant changes in organisational processes, technology stack, or applicable regulations.

TABLE OF CONTENTS

1. INTRODUCTION

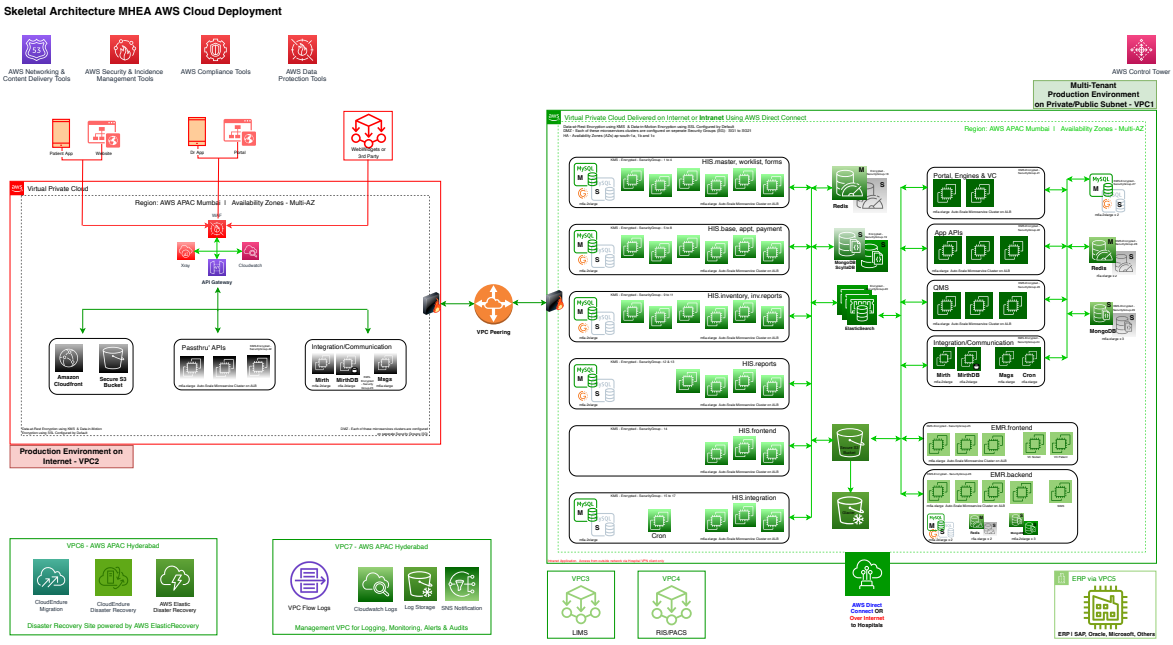
This whitepaper details the data-protection mechanisms implemented in the MyHealthcare Enterprise Application (MHEA) to safeguard all healthcare information. It covers encryption at rest, encryption in transmission, backup protection, cryptographic standards, compliance alignment, and associated risk-mitigation measures.

This paper is submitted in accordance with NABH DAC.2.a requirements, along with the WASA certification report as supporting evidence of security assurance.

2. OVERVIEW OF MHEA ARCHITECTURE

MHEA is a multi-tenant, cloud-hosted Healthcare Information System comprising:

- React.js frontend
- Node.js / Sails.js and .NET Core backend microservices
- Datastores including MariaDB, MongoDB and Redis
- Hosted on AWS EC2 (Amazon Linux) with reverse proxy on Nginx
- All PHI stored within secured VPCs with restricted IAM policies
- Periodic encrypted backups stored in segregated environments



3. DATA ENCRYPTION AT REST

MHEA employs industry-standard, contemporary encryption mechanisms to ensure all PHI stored in databases, logs and backups remains confidential.

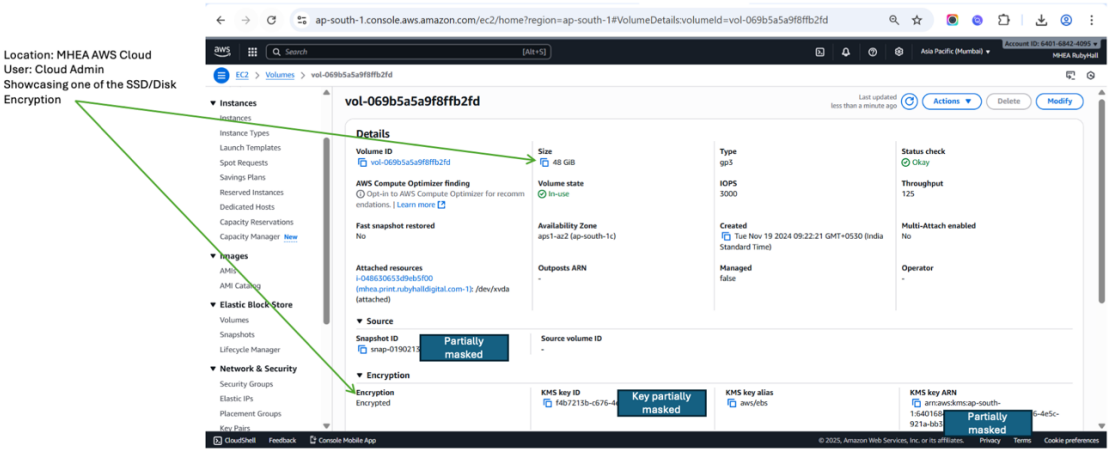


Image-1: Screenshot showing evidence of HDD Encryption for Data-at-Rest Encryption

3.1 Database Encryption

MariaDB (OLTP):

- Disk-level encryption using AES-256 via LUKS/efs encryption (depending on environment).
- Backup files (*.sql, snapshots) encrypted using AES-256-GCM.

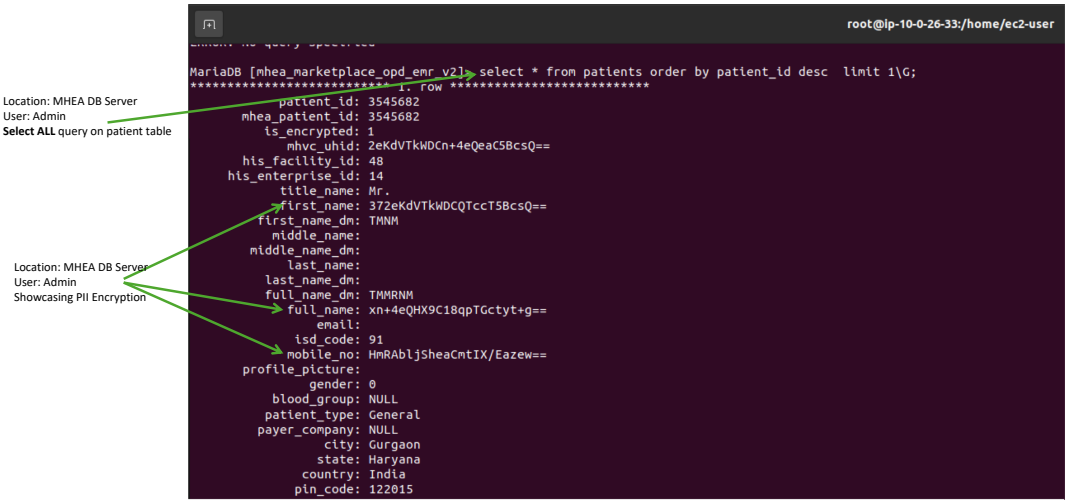
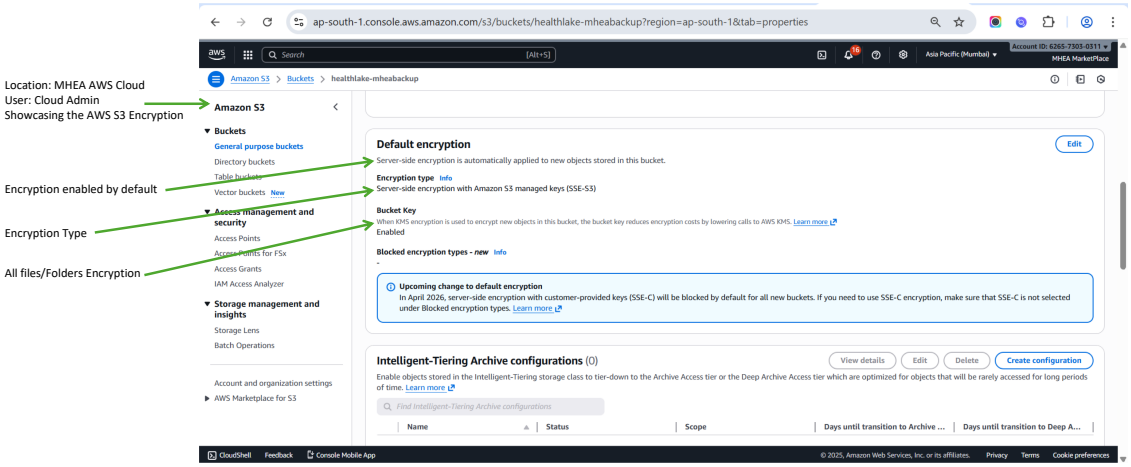


Image-4: Screenshot taken from 'MHEA Database Server' showing evidence of PII Encryption

3.2 File System & Media Encryption

- All uploaded files (lab reports, discharge summaries, imaging metadata) stored on encrypted volumes using AES-256.
- Object storage (if used, e.g., S3) is encrypted using SSE-S3 / SSE-KMS with AWS KMS keys rotated automatically every 365 days.



Screenshot showing evidence of AWS S3 Storage Encryption

3.3 Key Management

- Keys stored and managed in AWS KMS, following:
  - Automatic key rotation
  - Access controlled by IAM policies
  - Strict separation of duties for key administration

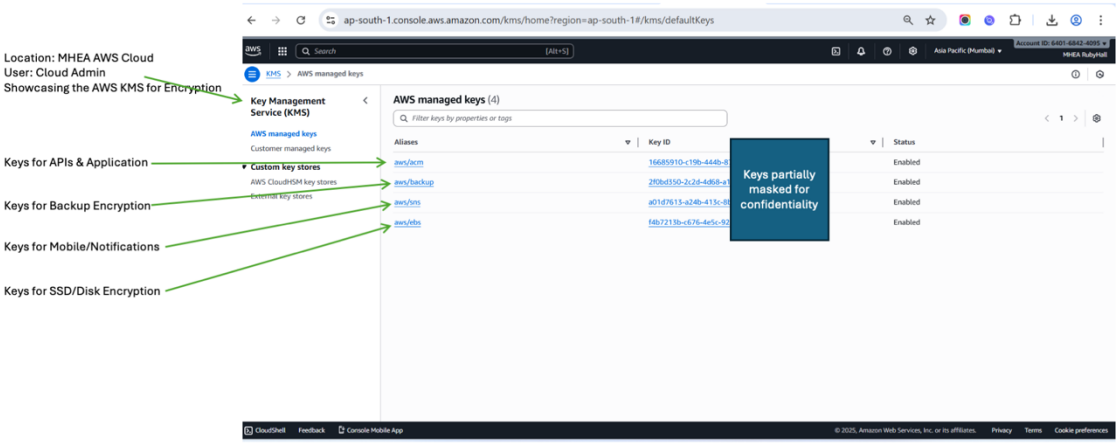


Image-3: Screenshot showing evidence AWS KMS Usage for Encryption Keys

4. DATA ENCRYPTION IN TRANSMISSION

MHEA enforces full end-to-end encryption for all internal, external and API-based communications.

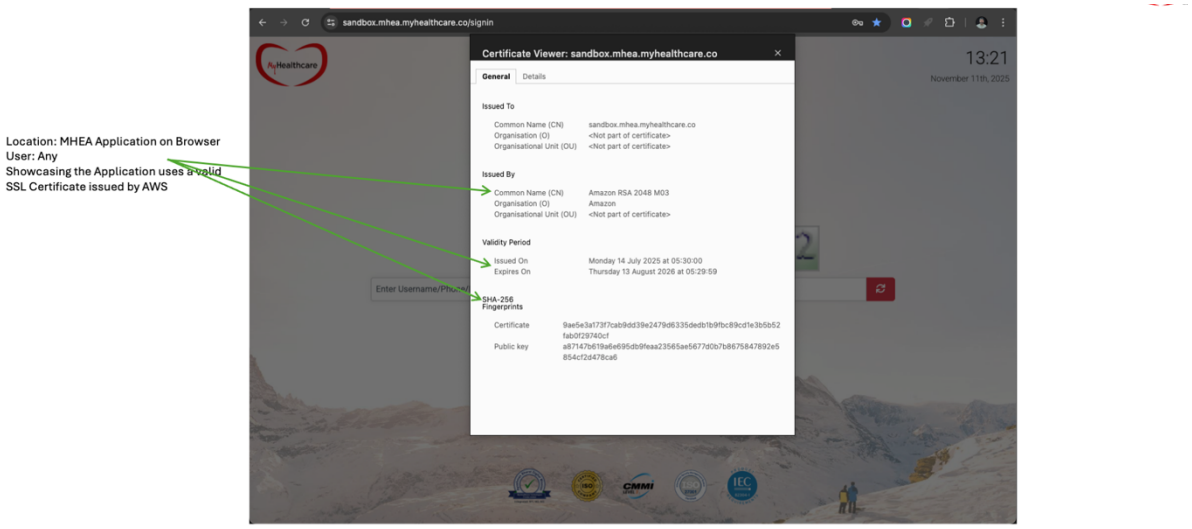


Image-4: Screenshot showing evidence of Data-in-Motion Encryption : SSL Certificate

4.1 TLS/SSL

- All communication between client → backend → APIs is protected using TLS 1.2 / 1.3.
- Certificates issued via AWS Certificate Manager or equivalent CA.
- Strict security headers enforced via Nginx:
    - Strict-Transport-Security (HSTS)
    - X-Frame-Options
    - X-Content-Type-Options

4.2 Internal Microservice Communication

- All service-to-service communication (Node.js, .NET Core, Nginx, Redis, MongoDB, MariaDB) occurs within a private VPC.
- Additional mTLS (mutual TLS) supported where relevant for partner integrations.

4.3 Third-Party Integrations

- All external APIs (pharmacy, lab, AD/SSO, payment gateway) use HTTPS (TLS 1.2/1.3).
- Sensitive parameters protected using industry cryptographic standards.

5. COMPLIANCE ALIGNMENT

MHEA's security controls align with major international standards and healthcare-sector data-protection frameworks:

5.1 ISO 27001:2022

- A.8.24: Cryptographic Key Management
- A.8.25: Secure Coding
- A.5.34: Data Masking & Encryption

5.2 NABH HIS / EMR Standards

- Fully aligned with DAC.2.a requirements on encryption at rest + in transit
- WASA-based security assurance provided as supporting evidence

5.3 HIPAA (Administrative & Technical Safeguards)

- Encryption of PHI at rest (HIPAA 164.312(a)(2)(iv))
- Encryption in transit (HIPAA 164.312(e)(1))
- Access control, audit trails, and emergency mode procedures

5.4 DPDPA/GDPR

- Data minimisation and pseudonymisation
- Encryption as a principal control under Article 32
- Right to erasure and secure destruction of backups

6. SECURE COMMUNICATION PROTOCOLS

To guarantee confidentiality and integrity of data during transmission:

6.1 Protocols Used

- TLS 1.3 with modern cipher suites such as:
  - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- Deprecated protocols (TLS 1.0, 1.1, 1.2, SSLv3) strictly disabled.

6.2 API Security

- All internal and external APIs require:
  - JWT-based authentication
  - Short-lived tokens
  - Role-based access restrictions
  - Input validation to prevent injection attacks

7. RISK MITIGATION MEASURES

To reduce risks relating to encryption failures, data breaches or cyber-attacks, the following safeguards are enforced:

7.1 Technical Safeguards

- Web Application Firewall (WAF) guarding all internet-facing endpoints
- Periodic scanning of SSL configurations
- Strict Content Security Policy (CSP)
- Regular VAPT & WASA cycles with retesting
- Automated log monitoring for unusual access patterns

7.2 Operational Safeguards

- Quarterly key rotation
- Least-privilege access control for DB and storage
- SOC2-style logging and monitoring
- Disaster Recovery backups encrypted and stored separately

7.3 Incident Response & Breach Mitigation

- Defined incident response SOP aligned with CERT-In
- Immediate revocation of compromised keys
- Rapid rotation of service credentials
- Forensic logging for post-incident analysis
- Zero-downtime patching for critical vulnerabilities

8. WASA CERTIFICATION (SUPPORTING EVIDENCE)

A full Web Application Security Assessment was performed on MHEA covering:

- Web UI, Admin Console
- Microservices & APIs
- Authentication/SSO
- Database access controls
- Encryption effectiveness tests
- Transport layer security evaluation

The assessment validated:

- Correct usage of TLS 1.2+
- Enforcement of encryption at rest
- No plaintext PHI storage
- No insecure cryptographic usage

VAPT detailed Scope & Specification document with collaterals.

- 3rd Party Compliance Audit – Scope & Specification Document
- 3rd Party Initial Audit Report
- 3rd Party Final Remediation Audit Report
- WASA Certificate 2025

A copy of the WASA certificate 2025 & summary has been attached separately, as required by NABH.

9. KEY TAKEAWAYS

MHEA employs robust and contemporary cryptographic techniques to ensure the confidentiality and integrity of patient data throughout its lifecycle.
Together with strong operational controls, periodic WASA assessments and compliance with international security standards, MHEA fully meets the expectations of NABH DAC.2.a for encryption at rest and in transmission.

10. APPLICABLE NORMS, FRAMEWORKS, AND POLICIES

MHEA Security Policy is aligned with the following statutory requirements, regulatory directives, and global best-practice frameworks governing cybersecurity, healthcare data protection, and cloud security:

20.1 Indian Laws, Rules, and Regulatory Directives

1. Information Technology Act, 2000
   Including amendments related to:
   - Electronic records
   - Cyber offences
   - Data protection obligations
2. Information Technology (Reasonable Security Practices and Procedures and Sensitive Personal Data or Information) Rules, 2011
   - SPDI definition and handling
   - Notice and consent requirements
   - Data security and breach obligations
3. CERT-In Directions – April 28, 2022
   Mandatory requirements for all service providers, intermediaries, data centres, and cloud entities, including:
   - Six-hour breach notification rule
   - Log retention for 180 days
   - Synchronised time-stamping (NTP)
   - Disclosure of cyber incidents across 20 notified categories
   - Storage of logs within India
4. Digital Personal Data Protection Act (DPDPA), 2023 (forward-looking compliance requirement)
   - Data fiduciary obligations
   - Data breach notifications
   - Rights of data principals
   - Purpose limitation and storage limitation
5. Medical Council of India (MCI) Guidelines
   - Maintenance, retention, and confidentiality of clinical records
   - Handling medico-legal case data
6. Ayushman Bharat Digital Mission (ABDM) Health Data Retention and Management Guidelines
   - Long-term archiving of health data
   - Interoperability and privacy mandates

7. Indian Companies Act, 2013
    - Retention of financial and statutory records
    - Audit and compliance reporting

20.2 International Standards and Frameworks

1. Cloud Security Alliance – Cloud Controls Matrix (CSA CCM v4.0)
   Relevant domains:
    - IVS-09: Incident Response
    - LOG-05: Incident Logging and Monitoring
    - SEF-04: Security Defect and Vulnerability Management
    - GRC-02: Regulatory Alignment
    - BCR-06: Crisis Communications & Post-Incident Processes

2. ITIL v4 Framework
   Specifically:
    - Incident Management
    - Problem Management
    - Service Continuity Management
    - Change Enablement
    - Governance & Continual Improvement

3. NIST Guidelines (Reference Guidance)
    - NIST SP 800-61: Computer Security Incident Handling Guide
    - NIST SP 800-53: Security and Privacy Controls
    - NIST SP 800-88: Media Sanitisation

4. ISO/IEC Standards (Reference Guidance)
    - ISO/IEC 27001: Information Security Management
    - ISO/IEC 27035: Information Security Incident Management
    - ISO/IEC 27701: Privacy Information Management

5. OWASP Security Frameworks
    - OWASP Top 10 (Web Application Risks)
    - OWASP API Security Top 10

20.3 Contractual and Operational Compliance Requirements

1. Data Processing Agreements (DPAs) with hospitals, enterprise customers, and cloud partners

2. Business Associate Agreements (BAAs) where relevant

3. Vendor SLAs and Security Clauses, including breach notification timelines

4. MyHealthcare Internal Security Governance Policies, such as:
    - Incident Response Policy
    - Access Control Policy
    - Cloud Security & DevSecOps Policy
    - Logging & Monitoring Policy
    - Business Continuity and DR Policy
    - Data Classification & Labelling Policy

When in doubt, consult with Security Compliance Team

Team members who have questions about the minimum requirements in this policy or the appropriateness of change procedures that they maintain should consult with your Engineering Manager or DevOps Team as needed.

11. ADDENDUM-1 (SoW & Specs)

Addendums Attached - 1

3rd Party Compliance Audit – Scope & Specification Document

WASA Detailing/Specs. Document

ID	82341
Title	WASA Detailing/Specs — **MHEA** (MyHealthcare Enterprise Application)
Version	1.0
Prepared by	Jointly by MHC Red Team & **IMPERIUM SOLUTIONS**
Prepared for	MHC Security Assessment Team & MHC IT Compliance Teams
Date	16 June 2025

S No.	Head	Product	Product OE	Interpretation	Expected Outcome
DAC.2.	Common	MHEA (MyHealthcare Enterprise Application – Entire Ecosysem)	The system is able to encrypt all the healthcare data at rest and that in transmission.	To safeguard personal and sensitive data from unauthorized access and maintain confidentiality, the system shall ensure that all healthcare data at rest is encrypted (including backup data). Also, all healthcare data in transmission should be encrypted. The system should employ contemporary data encryption techniques. These techniques utilize encryption algorithms and protocols to securely encode sensitive PHI (Personal Health Information).	Submission of detailing documentation, Scope confirmation & for WASA certification.

Executive summary

This document summarizes the Web Application Security Assessment (WASA) performed against the MHEA product between [**10-June-2025**] and [**17-June-2025**]. The assessment was undertaken to evaluate the application's readiness for HIMS/EMR (MHEA) certification and to validate controls protecting patient data (PHI/PII). Testing combined automated scanning and manual verification, mapped to OWASP ASVS v4 and the OWASP WSTG.

Major findings were [1] **High**, [1] **Medium** and [3] **Low** risk items; all risks findings were remediated and retested before final sign-off.

1. Objectives

- Validate confidentiality, integrity, and availability controls for patient data flows.
- Identify exploitable vulnerabilities in the web UI, APIs, authentication, session management, RBAC, and backend integrations.
- Map security posture to OWASP ASVS levels and MHC HIS requirements.
- Provide remediation guidance and re-test verification.

2. Scope

3.1 In-scope Components

- MHEA (full ecosystem) — https://mhea.myhealthcare.life
- White box | Authentication used in production – SAST & DAST.
- Collaterals Shared for Audit:
    - Application code base – GitLab Admin Access
    - Admin Access to AWS Admin Console & Full Unrestricted Server Access for Audit
    - Mobile Apps - .ipa, .apk files and full APIs (Postman Collection)

3.2 Out-of-scope

- Underlying cloud provider IaaS network controls .
- Mobile apps - **Doctor App** - MH Doctor , **Patient Apps** - SPS MyHealth , CKB Patient App , ILS Patient App , Paras Healthcare , Ruby Hall Clinic , Sakra Patient App , Sukoon Health , Sparsh Anubhava , RF Hospital .
- Test environments other than [staging - https://sandbox.mhea.myhealthcare.co/ Prod- / https://mhea.myhealthcare.life] .

3.3 Test type & assurance level

- White-box assessment .
- Assurance mapping: ASVS Level 2 (recommended for healthcare systems processing PHI).

4. Methodology & standards used

4.1 Standards and references

- OWASP ASVS v4.0.x (for control mapping and verification levels).
- OWASP Web Security Testing Guide (WSTG) (for test cases and processes).
- Industry VAPT best practices and reporting templates (pen-test + manual verification).

4.2 Test stages

- Reconnaissance & fingerprinting (enumeration of endpoints, versions, exposed headers).
- Authentication & session management checks (SSO flows, token lifetimes, cookie flags, MFA enforcement).
- Authorization testing (horizontal & vertical privilege escalation).
- Input validation and injection testing (XSS, SQLi, command injection, deserialisation).
- Business logic testing (workflow misuse, patient record access rules).
- API testing (rate limits, object-level access control, JSON/XML parsing).
- Configuration & deployment checks (TLS, HSTS, secure headers, CORS).
- Automated scanning followed by manual verification and proof-of-concept exploitation where safe and authorized.
- Post-test remediation verification (retest).

4.3 Tools & techniques (representative)

- Automated scanners: [e.g., Burp Suite Professional, OWASP ZAP, Nmap . Nikto , Dirb]
- Manual tools: Burp Suite (intruder, repeater), curl, Postman, custom scripts.
- Static code review and dependency analysis: [Snyk/Dependabot/Checkov as applicable].
- Logs & telemetry review to confirm detection of test activities.

(Include tool versions in Appendix B.) *

5. Test environment

- Environment: [staging https://sandbox.mhea.myhealthcare.co/ / prod-https://mhea.myhealthcare.life] —
- Date/time of tests: [10-June-2025 to 17-June-2025]

- Test exclusions & constraints: [e.g., no destructive exploitation, business-impact limitations]

6. Findings — Executive summary

- Total findings: **5** (H: 1; M: 1; L: 3)
- High (H) — [**Hardcoded keys**]
- Medium (M) — [**Misconfigured CORS**]
- Low (L) — [1-**Missing Security Headers**, 2-**Banner Grabbing**, 3-**User Name Enumeration**.]

All high severity issues were fixed by development and verified by **IMPERIUM SOLUTIONS** retest on **17-June-2025**.

7. Detailed findings (sample format)

For detailed finding please find Level 1 Report and for its closure Level 2 Report with relevant POC .

8. Risk assessment & business impact

- Risk rating approach: combination of CVSSv3.1 for technical severity plus business impact scoring (PHI exposure, number of affected records, exploitability).
- Example: High severity with proven read-only access to PHI = High business impact owing to regulatory, privacy and patient-safety consequences.
- Recommended prioritization: Fix all High within 7 days (or immediate hotfix), Medium within 30 days, Low as part of scheduled improvements.

9. Remediation & controls implemented

List of concrete remediations performed within Application find detailed Level 1 Report and for its closure Level 2 Report.

10. Mapping to ISO27001, ISO27035, ISO27701, SOC2T2, CSA CCI, OWASP, DPDPA, ABDM, MCI - compliance frameworks

- ASVS mapping: Controls tested and verified mapped to ASVS sections (e.g., Authentication — V2., Access Control — V4., Cryptography — V7.*).
- HIS/EMR requirements: Map all the HIS clauses defined — e.g., data confidentiality, audit trails, access control.

11. Detection & monitoring verification

- Confirmed logs in SIEM/central logging for: authentication failures, privilege escalations, suspicious API requests.
- Alerting thresholds and examples: [e.g., >5 failed logins within 5 mins triggers alert].
- Evidence: sample alerts, screenshots and log extracts .

12. Retest & validation

- Retest approach: All high and medium findings were retested using the original PoC steps and additional regression test cases.
- Retest results: High findings — closed; Medium — [open/closed] status; see Appendix E (retest evidence). Date of final retest: [17-June-2025].

13. Recommendations & roadmap

Short-term (0–30 days)

- Apply hotfixes for all high severity issues; tighten API access checks; rotate secrets.

Medium-term (30–90 days)

- Adopt ASVS Level 2 as a gating standard for releases; expand automated CI security checks; implement SAST/DAST in CI.

Long-term (90+ days)

- Threat-model critical flows; regular scheduled WASA (quarterly or after major releases); periodic third-party code dependency reviews.

14. Limitations & assumptions

- Testing was performed against the environments and accounts listed in Section 5. Changes after the final test date may affect results.
- No destructive exploitation was carried out; tests were limited to non-destructive PoC where necessary.

15. Conclusion

The WASA identified findings across MHEA. Critical issues were remediated and validated. Based on the ASVS mapping and retest evidence, MHEA meets the expected application security controls for MHC HIS certification at ASVS Level 2, subject to ongoing monitoring and periodic retesting. (Attach a signed statement if required by MHC.)

Signed by:

1. Saba Saifi, Application Security Engineer
2. Mohamed Salman, Engineering Manager
3. Sandeep Singh, Head-IT Infra & DevSecOps and DPO
4. Aneesh Nair, Co-Founder

Date: 28-Nov-2025

Appendices

A. Evidence attachments – See level 1 Report (VAPT report PDF, scan outputs, screenshots -)

B. Tool versions and environment details - Burp Suite Professional - 2025.3.4-38446 ,

OWASP ZAP - 2.16.0 , Nmap - 7.95 , Nikto - Nikto 2.5.0 (LW 2.5) , Dirb - DIRB v2.22 By The Dark Raver .

C. Screenshots and sanitized PoC data – See Level 2 Report (always redact PHI in attachments)

D. Retest reports and timestamps - See Level 2 Report

12. ADDENDUM-2 (Initial Audit Report)

Addendums Attached - 2

3rd Party Initial Audit Report

Imperium Solutions

MHEA VAPT 2025 Level 1 Report

For

Myhealthcare Technologies Pvt Ltd

Prepared By

IMPERIUM SOLUTIONS

10-June-2025

Disclaimer

this document or any of its content cannot account for, or be included in any form of legal advice. The outcome of a vulnerability scan (or security evaluation) should be utilized to ensure that diligent measures are taken to lower the risk of potential exploits carried out to compromise data. Legal advice must be supplied according to its legal context. All laws and the environments, in which they are applied, are constantly changed and revised. Therefore, no information provided in this document may ever be used as an alternative to a qualified legal body or representative. A portion of the information in this report is taken from OWASP's "The Ten Most Critical Web Application Security Vulnerabilities - 2025 Update" document, that can be found at https://www.owasp.org.

The results indicated in this report are reflective of the state of the site and its underlying code at the time of testing. Essential Infosec Pvt Ltd will not be liable for any changes that happen after the submission of this report which might add to vulnerabilities in future.

1. Management Summary

1.1 Document Title

DOCUMENT VERSION CONTROL	
Document Title	Web Application Vulnerability Assessment and Penetration Testing Level 1 Report
Organisation Name	Myhealthcare Technologies Pvt Ltd
Application Name	MyHealthcare Enterprise Application
Document Version	1.0
Last Edit Date	17-June-2025
Auditor Details	Mr. Sanjeev Chavan

Table of Contents

2. Scope, Scan (Test) and Report (Creation/Review) Details

Following Website is considered as scope of work. During the security assessment we have evaluated security of the modules from the application.

Scope of the work	
MHEA	Website
Audit URL	https://sandbox.mhea.myhealthcare.co/
Scope	**Annual Compliance Full Audit – Whitehat Method – Level-1 – Initial Audit Report**
HASH	0ad41814526204c2075422641423e1a9b0d2ecc6383 5979e70685cff774c5a5d

Scan / Test Details	
Start Date	10-June-2025
End Date	17-June-2025
Scan / Test Time	08 Working Days

Report Created	
Report Created By	Mr. Sanjeev Chavan

3. Assessment Methodology

A hybrid approach is followed to perform the assessment that is a combination of tools is used to discover the wide range of vulnerabilities. Additionally, the assessment being adaptive in nature allows us to control the assessment methodology as per the application functionality to focus on the critical areas of the application. The attack vectors are controlled as per the assessment needs and the attack selection ensures maximum coverage of the application.

Following diagram represents the assessment approach:

The table below describes various levels and types of assessment. The type of assessment done for current assessment is available in the "Assessment Scope" section of the document.

Scan/Audit Type		
Level	Type	Information
1	**Safe**	Safe scan discovers minimum types and instances of vulnerabilities. The safe scan mode avoid fault injection such as Java Scripts, HTML tags, crafted SQL queries etc. to ensure that the application retains its state at the end of the assessment. Any fault injections that may trigger Denial of Service situation are avoided in safe scans. Safe scan suits most when the assessment is to be done on a live application instance, and has already undergone either *Standard* or *Destructive* scan/s.
2	**Standard**	Standard scan discovers and exploits most standard checks such as OWASP Top 10 checks. The standard scan performs fault injection such as Java Scripts injection, HTML tag injection, crafted SQL queries etc. Any fault injections that may trigger Denial of Service situation are avoided in standard scans. Standard scan suits most when the assessment is to be done on a staging/pre-prod/testing application instance.
3	**Destructive**	Destructive scan discovers and exploits most comprehensive checks including checks that may trigger Denial of Service Attacks situations for the application. Destructive scan is usually done on staging/pre-prod/testing application instance. A destructive scan on a live environment is avoided on live/production systems unless it is really required.

The vulnerabilities discovered are associated with a risk level that indicates how critical the vulnerability is and helps application owners/developers to prioritize the vulnerabilities and choose an appropriate mitigation approach.

Risk Level Information and Necessary Actions

Risk Level	Risk Description and Necessary Action
High	The high risk level indicates maximum risk associated with a specific vulnerability instance. Such vulnerability may enable an attacker to successfully exploit the underlying application and its data and partially or completely to compromise the application and its data to modify application behaviour to become other than its original intended purpose. The vulnerability marked as "High Risk" is recommended to be handled with utmost priority.
Medium	The medium risk level indicates considerable risk associated with a specific vulnerability instance. Such vulnerability may enable an attacker to exploit the underlying application and its data to a particular level so that the attacker can gain low level information about the application. Such information can be used by an attacker to craft more specific attacks based on the information collected. The vulnerability marked with "Medium Risk" should be mitigated at the earliest or soon after "High Risk" vulnerabilities are mitigated.
Low	The low risk level indicates lowest risk associated with a specific vulnerability instance. Such vulnerability may allow an attacker to gain some information about the application which was not intended to be known otherwise. The attacker may not have exploiting techniques available at that instance based on the information revealed by the system. The vulnerability marked with "Low Risk" can be mitigated soon after high and medium risk vulnerabilities are mitigated.

Severity Level Information and Description

SEVERITY	CVSS SCORE	DESCRIPTION
CRITICAL	**9.00 – 10.00**	Critical Business Impact - e.g., Remote Code Execution, Database Access, System Take Over. Requires fix immediately.
HIGH	**7.00 – 8.99**	High Business Impact - e.g., Bypassing security controls, arbitrary script execution, takeover any user's account, bypass of user accounts on critical parts of the site. Requires fix as soon as possible
MEDIUM	**4.00 – 6.99**	Typically, vulnerabilities that requires the attacker to combine it with another vulnerability to cause serious damage
LOW	**0.01 – 3.99**	Can cause annoyance to the user and be used in a combination with similar vulnerabilities.
INFORMATIONAL	**0.0**	Bugs that do not create a threat directly or indirectly fall under this category.

4. Assessment Constraints

There were no assessment constraints in our security audit.

5. Tools Used

The below tools/scans/scripts were used for security audit:

- BurpSuite Professional
- Zap Proxy
- Nmap
- Nikto
- Kali Linux Tools
- SSL Labs
- Netsparker

6. Standard Followed

Below are the standards followed for VAPT –

1. Open Web Application Security Project (OWASP)
2. SysAdmin, Audit, Network, and Security (SANS)
3. Penetration Testing Execution Standard (PTES)
4. Payment Card Industry Data Security Standard (PCI DSS)
5. Information Systems Security Assessment Framework (ISSAF)
6. Open Source Security Testing Methodology Manual (OSSTMM)

7. OWASP Top 10 and SANS 25 Application Security Risks

The Common Weakness Enumeration (CWE) is a list of software security vulnerabilities found all throughout the software development industry. It's a community-driven project maintained by MITRE, a non-profit research and development group. For each entry, the CWE provides a description of the vulnerability and steps for mitigating it.

MITRE partnered with the SANS Institute to develop the CWE/25, a list of the 25 most critical software vulnerabilities. A similar list is provided in the Open Web Application Security Project (OWASP) Top 10 Project, which is also a community-driven compilation of software vulnerabilities. Although the CWE/25 and OWASP Top 10 are different, they share many of the same vulnerabilities. Here is a list of the OWASP Top 10 entries for 2021 and their corresponding CWEs.

OWASP Top 10	SANS CWE Top 25
A01:2021-Broken Access Control	1. CWE-787 Out-of-bounds Write
A02:2021-Cryptographic Failures	2. CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
A03:2021-Injection	3. CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
A04:2021-Insecure Design	4. CWE-416 Use After Free
A05:2021-Security Misconfiguration	5. CWE-78 Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
A06:2021-Vulnerable and Outdated Components	6. CWE-20 Improper Input Validation
A07:2021-Identification and Authentication Failures	7. CWE-125 Out-of-bounds Read
A08:2021-Software and Data Integrity Failures	8. CWE-22 Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
A09:2021-Security Logging and Monitoring Failures	9. CWE-352 Cross-Site Request Forgery (CSRF)
A10:2021-Server-Side Request Forgery	10. CWE-434 Unrestricted Upload of File with Dangerous Type
	11. CWE-862 Missing Authorization
	12. CWE-476 NULL Pointer Dereference
	13. CWE-287 Improper Authentication
	14. CWE-190 Integer Overflow or Wraparound
	15. CWE-502 Deserialization of Untrusted Data
	16. CWE-77 Improper Neutralization of Special Elements used in a Command ('Command Injection')
	17. CWE-119 Improper Restriction of Operations within the Bounds of a Memory Buffer
	18. CWE-798 Use of Hard-coded Credentials
	19. CWE-918 Server-Side Request Forgery (SSRF)
	20. CWE-306 Missing Authentication for Critical Function
	21. CWE-362 Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')
	22. CWE-269 Improper Privilege Management
	23. CWE-94 Improper Control of Generation of Code ('Code Injection')
	24. CWE-863 Incorrect Authorization
	25. CWE-276 Incorrect Default Permissions

Zero-day (0-day) Application Security Risks

During the security testing, checks were also performed for any zero-day vulnerabilities / attacks.

Below is the brief summary of Zero-day (0-day) security risks –

"Zero-day" is a broad term that describes recently discovered security vulnerabilities that hackers can use to attack systems. The term "zero-day" refers to the fact that the vendor or developer has only just learned of the flaw – which means they have “zero days” to fix it. A zero-day attack takes place when hackers exploit the flaw before developers have a chance to address it.

Zero-day is sometimes written as 0-day. The words vulnerability, exploit, and attack are typically used alongside zero-day, and it’s helpful to understand the difference:

A zero-day vulnerability is a software vulnerability discovered by attackers before the vendor has become aware of it. Because the vendors are unaware, no patch exists for zero-day vulnerabilities, making attacks likely to succeed.

A zero-day exploit is the method hackers use to attack systems with a previously unidentified vulnerability.

A zero-day attack is the use of a zero-day exploit to cause damage to or steal data from a system affected by a vulnerability.

Protection against zero-day attacks –

For zero-day protection and to keep your computer and data safe, it’s essential for both individuals and organizations to follow cyber security best practices. This includes:

Keep all software and operating systems up to date. This is because the vendors include security patches to cover newly identified vulnerabilities in new releases. Keeping up to date ensures you are more secure.

Use only essential applications. The more software you have, the more potential vulnerabilities you have. You can reduce the risk to your network by using only the applications you need.

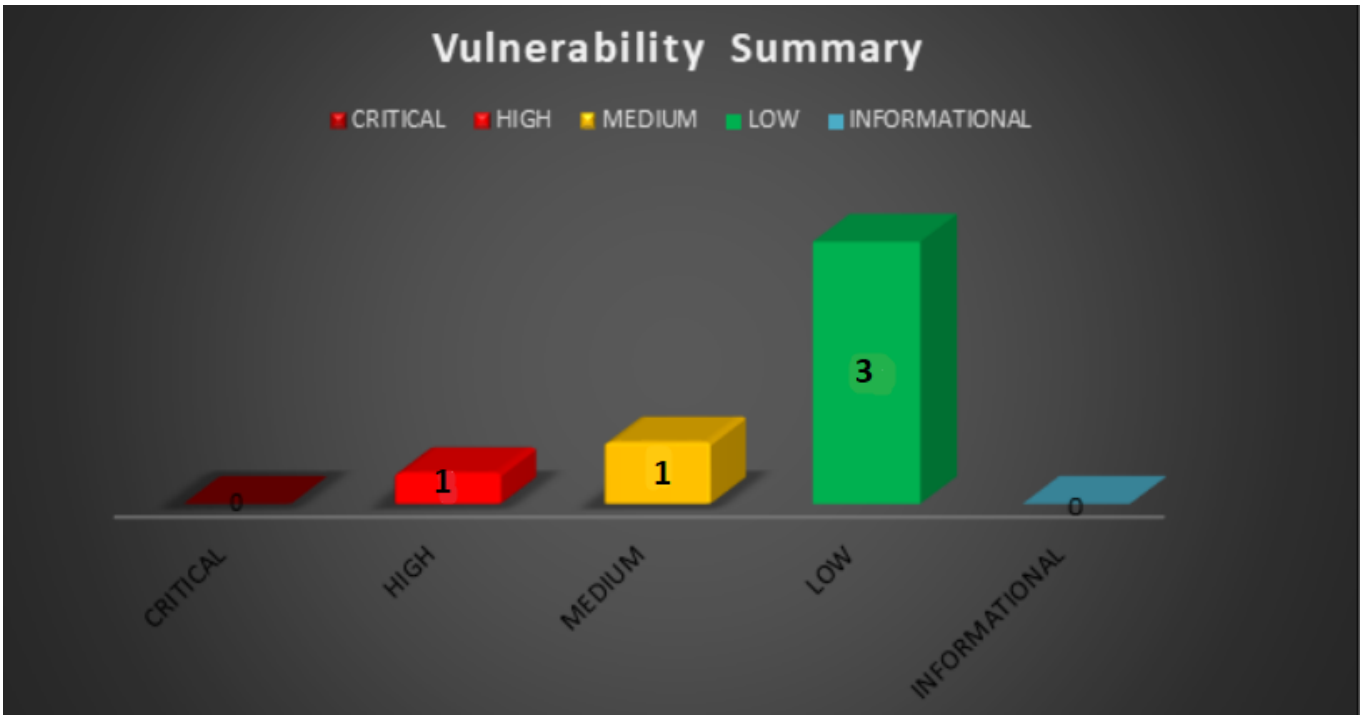
Use a firewall. A firewall plays an essential role in protecting your system against zero-day threats. You can ensure maximum protection by configuring it to allow only necessary transactions.

Within organizations, educate users. Many zero-day attacks capitalize on human error. Teaching employees and users’ good safety and security habits will help keep them safe online and protect organizations from zero-day exploits and other digital threats.

8. Summary of Key Findings

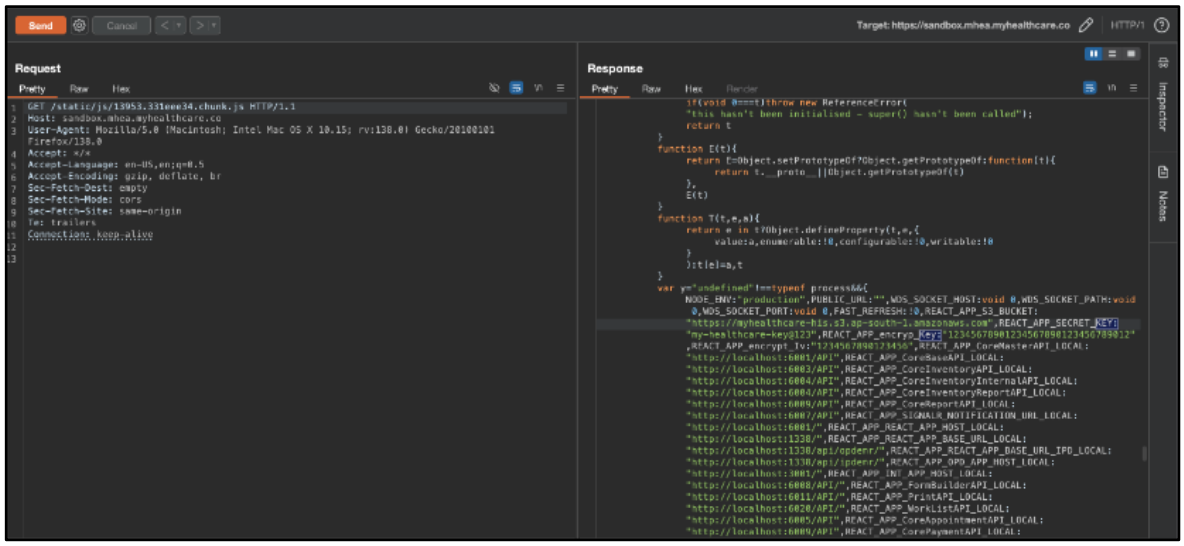
Sr. No.	Vulnerability Name	Severity	Level 1 Status
01	Hardcoded keys	High	OPEN
02	Misconfigured CORS	Medium	OPEN
03	Missing Security Headers	Low	OPEN
04	Banner Grabbing	Low	OPEN
05	Username Enumeration	Low	OPEN

9. Graphical Representation



10. Detailed Technical Report with Recommendations

01: Hardcoded keys	
SEVERITY	**High**
STATUS	**OPEN**
DESCRIPTION	
Application stores sensitive keys (API keys, encryption keys, access tokens, etc.) directly in the source code or configuration files. Anyone with code access or via reverse-engineering can extract these secrets. This leads to unauthorized access to internal services or third-party systems.	
IMPACT	
Attackers can use exposed keys to impersonate the application, access databases, modify data, escalate privileges, or perform financial/transactional abuse. May result in full system	
AFFECTED URL	
GET /static/js/13953.331eee34.chunk.js HTTP/1	
RECOMMENDATIONS	
Remove keys from source code and store them securely using environment variables or secret managers (e.g., AWS Secrets Manager, Azure Key Vault, HashiCorp Vault). Rotate exposed keys	
PROOF OF CONCEPT	



02: Misconfigured CORS

SEVERITY	Medium
STATUS	OPEN

DESCRIPTION

Misconfigured CORS occurs when a web server incorrectly allows unauthorized domains to access resources, exposing sensitive data to malicious websites.

IMPACT

It can lead to data theft, unauthorized access to APIs, and potential exploitation of sensitive user information.

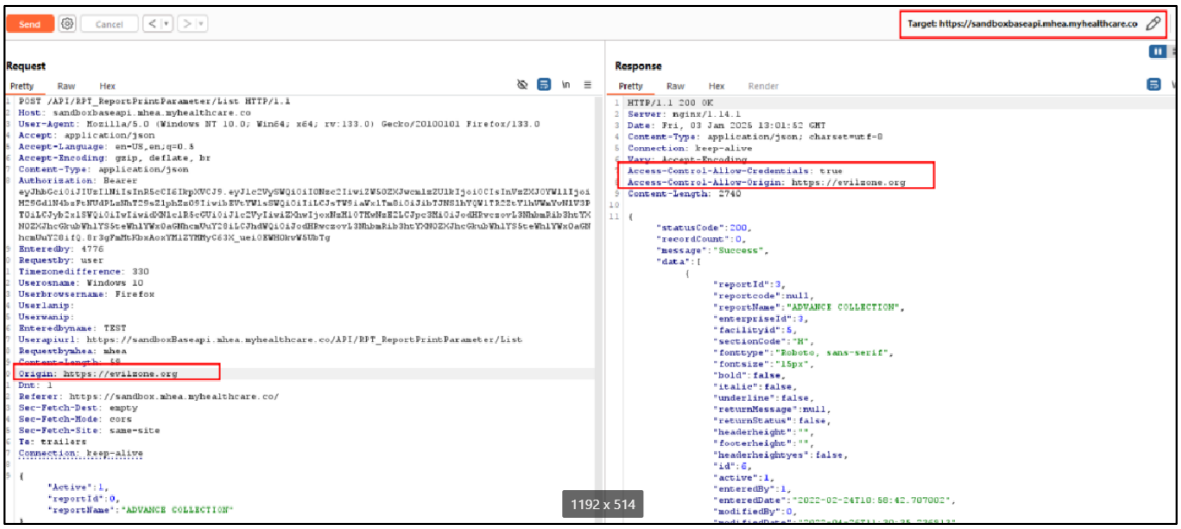
AFFECTED URL

Throughout the Application.

RECOMMENDATIONS

Properly configure CORS to only allow trusted domains, avoid using wildcard (*) for sensitive resources, and validate the origin header.

PROOF OF CONCEPT



03: Missing Security Headers

SEVERITY	Low
STATUS	OPEN

DESCRIPTION

The application is missing important HTTP security headers that help protect against various attacks, such as Cross-Site Scripting (XSS), Clickjacking, and other web vulnerabilities. Key headers such as Content-Security-Policy, X-Frame-Options, X-XSS-Protection, and StrictTransport-Security are not present in the server's response.

IMPACT

The absence of these security headers increases the risk of attacks on the application. Without appropriate headers, the application may be more vulnerable to cross-site scripting, data injection attacks, and unauthorized framing of content, which can compromise user security and privacy

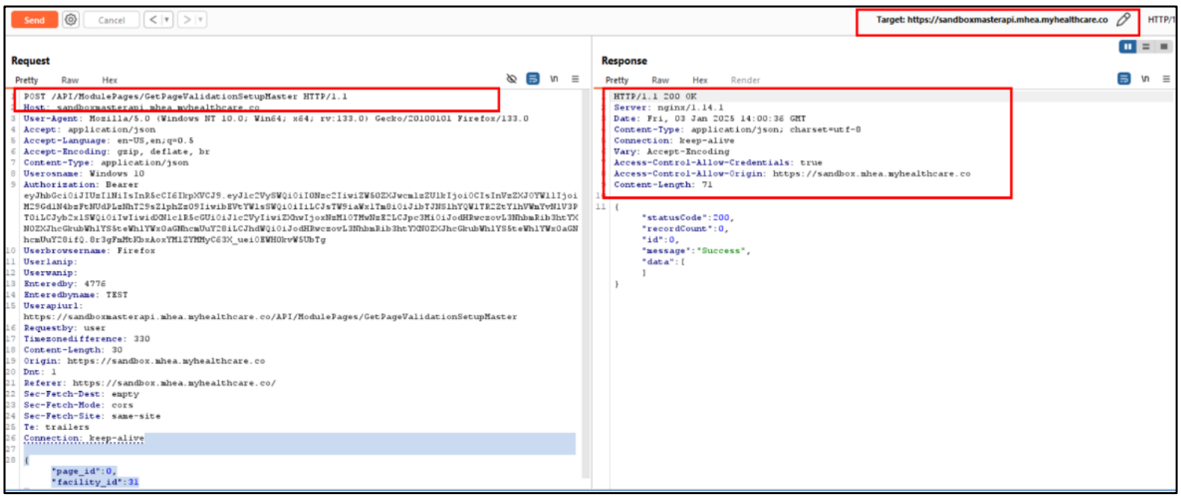
AFFECTED URL

Throughout the Application.

RECOMMENDATIONS

Implement the following security headers to enhance the security posture of the application:
- Content-Security-Policy to control resources the user agent is allowed to load.
- X-Frame-Options to prevent Clickjacking by disallowing the application to be rendered in a frame.
- X-XSS-Protection to enable cross-site scripting filters in web browsers.
- Strict-Transport-Security to enforce secure connections to the server.

PROOF OF CONCEPT



04: Banner Grabbing

SEVERITY	Low
STATUS	OPEN

DESCRIPTION

Banner Grabbing is an enumeration technique used to glean information about computer systems on a network, server information and the services running its open ports

IMPACT

An attacker can access the cookie via non-Http In most cases, banner grabbing does not involve the leakage of critical pieces of information, but rather information that may aid the attacker through the exploitation phase of the attack. For example, if target leaks the version of PHP it is running on the server, and it happens to be vulnerable to Remote Command/Code Execution (RCE) because it wasn't updated, the attackers may exploit the known vulnerability and take full control of the web application. by using client-side script such as JavaScript

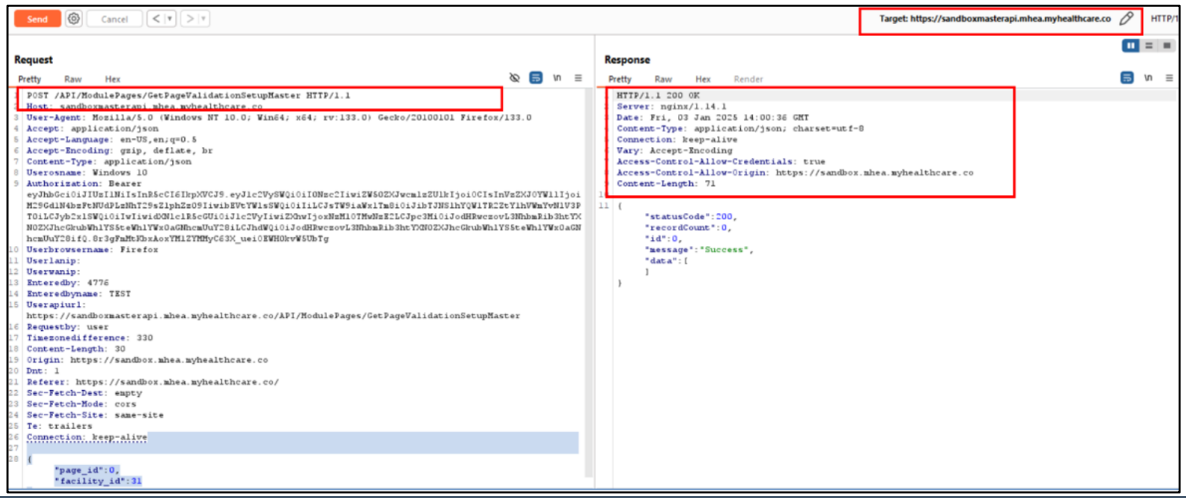
AFFECTED URL

Throughout the Application.

RECOMMENDATIONS

Remove all the unnecessary response headers which contains server information like server name or server version etc.

PROOF OF CONCEPT



05: Username Enumeration

SEVERITY	Low
STATUS	OPEN

DESCRIPTION

Username enumeration is a vulnerability where an attacker can determine whether a username exists in a system by observing different error messages or response times during login
attempts.

IMPACT

It can lead to attackers identifying valid usernames, making it easier to launch further attacks like password guessing or brute force attacks.

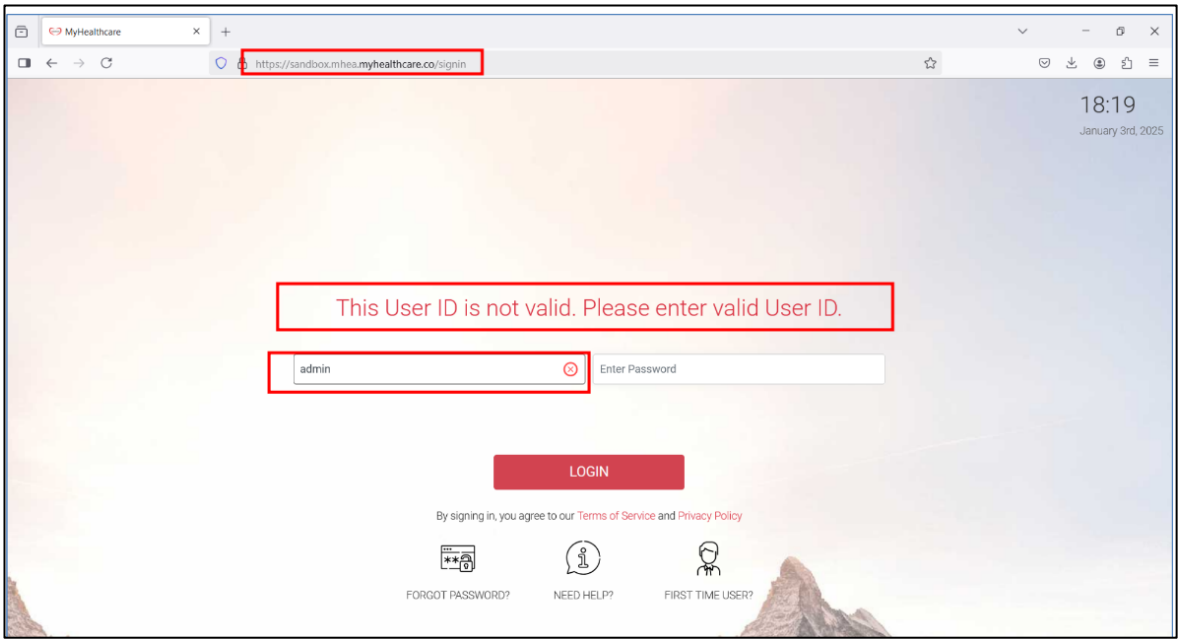
AFFECTED URL

Signing page.

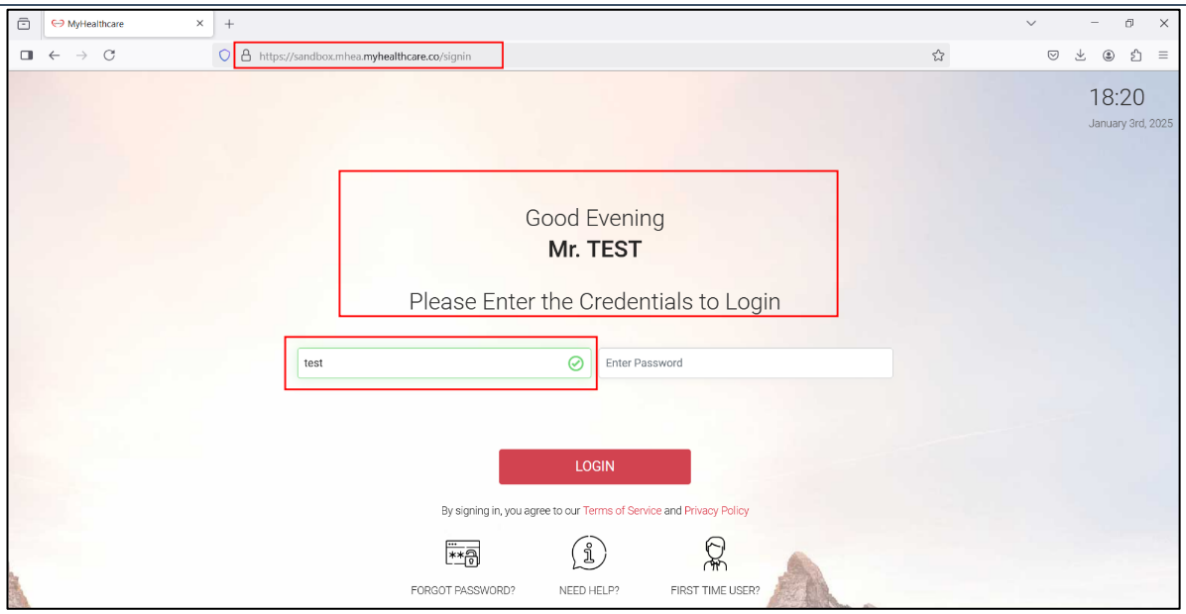
RECOMMENDATIONS

Ensure that error messages are generic and do not reveal information about whether the username or password is incorrect. Implement rate limiting and account lockout mechanisms.

PROOF OF CONCEPT



1.1


MyHealthcare
https://sandbox.mhea.myhealthcare.co/signin
18:20
January 3rd, 2025
Good Evening
Mr. TEST
Please Enter the Credentials to Login
test
Enter Password
LOGIN
By signing in, you agree to our Terms of Service and Privacy Policy
FORGOT PASSWORD?
NEED HELP?
FIRST TIME USER?

1.2

11. General References

Application Security Standard –

https://owasp.org/Top10/

https://www.sans.org/top25-software-errors/

https://cert-in.org.in/

Hardening of Servers –

https://geekflare.com/apache-web-server-hardening-security/

https://geekflare.com/apache-tomcat-hardening-and-security-guide/

https://geekflare.com/nginx-webserver-security-hardening-guide/

https://geekflare.com/ibm-http-server-security-guide/

THE END OF DOCUMENT

13. ADDENDUM-3 (Final Remediation Audit)

Addendums Attached - 3

3rd Party Final Remediation Audit Report

Imperium Solutions

MHEA VAPT 2025 Level 2 Report

For

Myhealthcare Technologies Pvt Ltd

Prepared By

IMPERIUM SOLUTIONS

17-June-2025

Disclaimer

this document or any of its content cannot account for, or be included in any form of legal advice. The outcome of a vulnerability scan (or security evaluation) should be utilized to ensure that diligent measures are taken to lower the risk of potential exploits carried out to compromise data. Legal advice must be supplied according to its legal context. All laws and the environments, in which they are applied, are constantly changed and revised. Therefore, no information provided in this document may ever be used as an alternative to a qualified legal body or representative. A portion of the information in this report is taken from OWASP's "The Ten Most Critical Web Application Security Vulnerabilities - 2025 Update" document, that can be found at https://www.owasp.org.

The results indicated in this report are reflective of the state of the site and its underlying code at the time of testing. Essential Infosec Pvt Ltd will not be liable for any changes that happen after the submission of this report which might add to vulnerabilities in future.

1. Management Summary

1.1 Document Title

DOCUMENT VERSION CONTROL	
Document Title	Web Application Vulnerability Assessment and Penetration Testing Level 2 Report
Organisation Name	Myhealthcare Technologies Pvt Ltd
Application Name	MyHealthcare Enterprise Application
Document Version	1.0
Last Edit Date	17-June-2025
Auditor Details	Mr. Sanjeev Chavan

Table of Contents

2. Scope, Scan (Test) and Report (Creation/Review) Details

Following Website is considered as scope of work. During the security assessment we have evaluated security of the modules from the application.

Scope of the work	
MHEA	Website
Audit URL	https://sandbox.mhea.myhealthcare.co/signin
Scope	**Remediation Audit**
HASH	0ad41814526204c2075422641423e1a9b0d2ecc6383 5979e70685cff774c5a5d

Scan / Test Details	
Start Date	10-June-2025
End Date	17-June-2025
Scan / Test Time	08 Working Days

Report Created	
Report Created By	Mr. Sanjeev Chavan

3. Assessment Methodology

A hybrid approach is followed to perform the assessment that is a combination of tools is used to discover the wide range of vulnerabilities. Additionally, the assessment being adaptive in nature allows us to control the assessment methodology as per the application functionality to focus on the critical areas of the application. The attack vectors are controlled as per the assessment needs and the attack selection ensures maximum coverage of the application.

Following diagram represents the assessment approach:

The table below describes various levels and types of assessment. The type of assessment done for current assessment is available in the "Assessment Scope" section of the document.

Scan/Audit Type		
Level	Type	Information
1	**Safe**	Safe scan discovers minimum types and instances of vulnerabilities. The safe scan mode avoid fault injection such as Java Scripts, HTML tags, crafted SQL queries etc. to ensure that the application retains its state at the end of the assessment. Any fault injections that may trigger Denial of Service situation are avoided in safe scans. Safe scan suits most when the assessment is to be done on a live application instance, and has already undergone either *Standard* or *Destructive* scan/s.
2	**Standard**	Standard scan discovers and exploits most standard checks such as OWASP Top 10 checks. The standard scan performs fault injection such as Java Scripts injection, HTML tag injection, crafted SQL queries etc. Any fault injections that may trigger Denial of Service situation are avoided in standard scans. Standard scan suits most when the assessment is to be done on a staging/pre-prod/testing application instance.
3	**Destructive**	Destructive scan discovers and exploits most comprehensive checks including checks that may trigger Denial of Service Attacks situations for the application. Destructive scan is usually done on staging/pre-prod/testing application instance. A destructive scan on a live environment is avoided on live/production systems unless it is really required.

The vulnerabilities discovered are associated with a risk level that indicates how critical the vulnerability is and helps application owners/developers to prioritize the vulnerabilities and choose an appropriate mitigation approach.

Risk Level Information and Necessary Actions

Risk Level	Risk Description and Necessary Action
High	The high risk level indicates maximum risk associated with a specific vulnerability instance. Such vulnerability may enable an attacker to successfully exploit the underlying application and its data and partially or completely to compromise the application and its data to modify application behaviour to become other than its original intended purpose. The vulnerability marked as "High Risk" is recommended to be handled with utmost priority.
Medium	The medium risk level indicates considerable risk associated with a specific vulnerability instance. Such vulnerability may enable an attacker to exploit the underlying application and its data to a particular level so that the attacker can gain low level information about the application. Such information can be used by an attacker to craft more specific attacks based on the information collected. The vulnerability marked with "Medium Risk" should be mitigated at the earliest or soon after "High Risk" vulnerabilities are mitigated.
Low	The low risk level indicates lowest risk associated with a specific vulnerability instance. Such vulnerability may allow an attacker to gain some information about the application which was not intended to be known otherwise. The attacker may not have exploiting techniques available at that instance based on the information revealed by the system. The vulnerability marked with "Low Risk" can be mitigated soon after high and medium risk vulnerabilities are mitigated.

Severity Level Information and Description

SEVERITY	CVSS SCORE	DESCRIPTION
CRITICAL	**9.00 – 10.00**	Critical Business Impact - e.g., Remote Code Execution, Database Access, System Take Over. Requires fix immediately.
HIGH	**7.00 – 8.99**	High Business Impact - e.g., Bypassing security controls, arbitrary script execution, takeover any user's account, bypass of user accounts on critical parts of the site. Requires fix as soon as possible
MEDIUM	**4.00 – 6.99**	Typically, vulnerabilities that requires the attacker to combine it with another vulnerability to cause serious damage
LOW	**0.01 – 3.99**	Can cause annoyance to the user and be used in a combination with similar vulnerabilities.
INFORMATIONAL	**0.0**	Bugs that do not create a threat directly or indirectly fall under this category.

4. Assessment Constraints

There were no assessment constraints in our security audit.

5. Tools Used

The below tools/scans/scripts were used for security audit:

- BurpSuite Professional
- Zap Proxy
- Nmap
- Nikto
- Kali Linux Tools
- SSL Labs
- Netsparker

6. Standard Followed

Below are the standards followed for VAPT –

1. Open Web Application Security Project (OWASP)
2. SysAdmin, Audit, Network, and Security (SANS)
3. Penetration Testing Execution Standard (PTES)
4. Payment Card Industry Data Security Standard (PCI DSS)
5. Information Systems Security Assessment Framework (ISSAF)
6. Open Source Security Testing Methodology Manual (OSSTMM)

7. OWASP Top 10 and SANS 25 Application Security Risks

The Common Weakness Enumeration (CWE) is a list of software security vulnerabilities found all throughout the software development industry. It's a community-driven project maintained by MITRE, a non-profit research and development group. For each entry, the CWE provides a description of the vulnerability and steps for mitigating it.

MITRE partnered with the SANS Institute to develop the CWE/25, a list of the 25 most critical software vulnerabilities. A similar list is provided in the Open Web Application Security Project (OWASP) Top 10 Project, which is also a community-driven compilation of software vulnerabilities. Although the CWE/25 and OWASP Top 10 are different, they share many of the same vulnerabilities. Here is a list of the OWASP Top 10 entries for 2021 and their corresponding CWEs.

OWASP Top 10	SANS CWE Top 25
A01:2021-Broken Access Control	1. CWE-787 Out-of-bounds Write 2. CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') 3. CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') 4. CWE-416 Use After Free 5. CWE-78 Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') 6. CWE-20 Improper Input Validation 7. CWE-125 Out-of-bounds Read 8. CWE-22 Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') 9. CWE-352 Cross-Site Request Forgery (CSRF) 10. CWE-434 Unrestricted Upload of File with Dangerous Type 11. CWE-862 Missing Authorization 12. CWE-476 NULL Pointer Dereference 13. CWE-287 Improper Authentication 14. CWE-190 Integer Overflow or Wraparound 15. CWE-502 Deserialization of Untrusted Data 16. CWE-77 Improper Neutralization of Special Elements used in a Command ('Command Injection') 17. CWE-119 Improper Restriction of Operations within the Bounds of a Memory Buffer 18. CWE-798 Use of Hard-coded Credentials 19. CWE-918 Server-Side Request Forgery (SSRF) 20. CWE-306 Missing Authentication for Critical Function 21. CWE-362 Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition') 22. CWE-269 Improper Privilege Management 23. CWE-94 Improper Control of Generation of Code ('Code Injection') 24. CWE-863 Incorrect Authorization
A02:2021-Cryptographic Failures	
A03:2021-Injection	
A04:2021-Insecure Design	
A05:2021-Security Misconfiguration	
A06:2021-Vulnerable and Outdated Components	
A07:2021-Identification and Authentication Failures	
A08:2021-Software and Data Integrity Failures	
A09:2021-Security Logging and Monitoring Failures	
A10:2021-Server-Side Request Forgery	

	25. CWE-276 Incorrect Default Permissions

Zero-day (0-day) Application Security Risks

During thesecurity testing, checks were also performed for any zero-day vulnerabilities / attacks.

Below is the brief summary of Zero-day (0-day) security risks –

"Zero-day" is a broad term that describes recently discovered security vulnerabilities that hackers can use to attack systems. The term "zero-day" refers to the fact that the vendor or developer has only just learned of the flaw – which means they have “zero days” to fix it. A zero-day attack takes place when hackers exploit the flaw before developers have a chance to address it.

Zero-day is sometimes written as 0-day. The words vulnerability, exploit, and attack are typically used alongside zero-day, and it’s helpful to understand the difference:

A zero-day vulnerability is a software vulnerability discovered by attackers before the vendor has become aware of it. Because the vendors are unaware, no patch exists for zero-day vulnerabilities, making attacks likely to succeed.

A zero-day exploit is the method hackers use to attack systems with a previously unidentified vulnerability.

A zero-day attack is the use of a zero-day exploit to cause damage to or steal data from a system affected by a vulnerability.

Protection against zero-day attacks –

For zero-day protection and to keep your computer and data safe, it’s essential for both individuals and organizations to follow cyber security best practices. This includes:

Keep all software and operating systems up to date. This is because the vendors include security patches to cover newly identified vulnerabilities in new releases. Keeping up to date ensures you are more secure.

Use only essential applications. The more software you have, the more potential vulnerabilities you have. You can reduce the risk to your network by using only the applications you need.

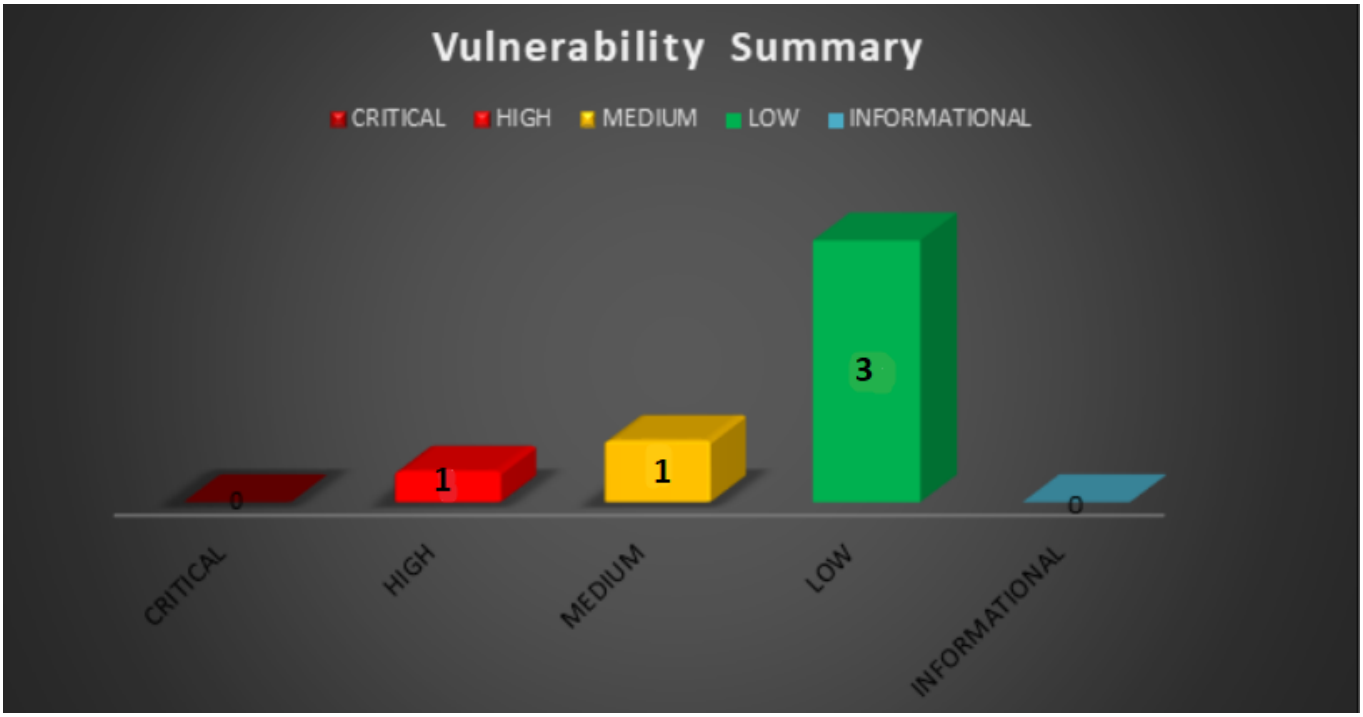
Use a firewall. A firewall plays an essential role in protecting your system against zero-day threats. You can ensure maximum protection by configuring it to allow only necessary transactions.

Within organizations, educate users. Many zero-day attacks capitalize on human error. Teaching employees and users’ good safety and security habits will help keep them safe online and protect organizations from zero-day exploits and other digital threats.

8. Summary of Key Findings

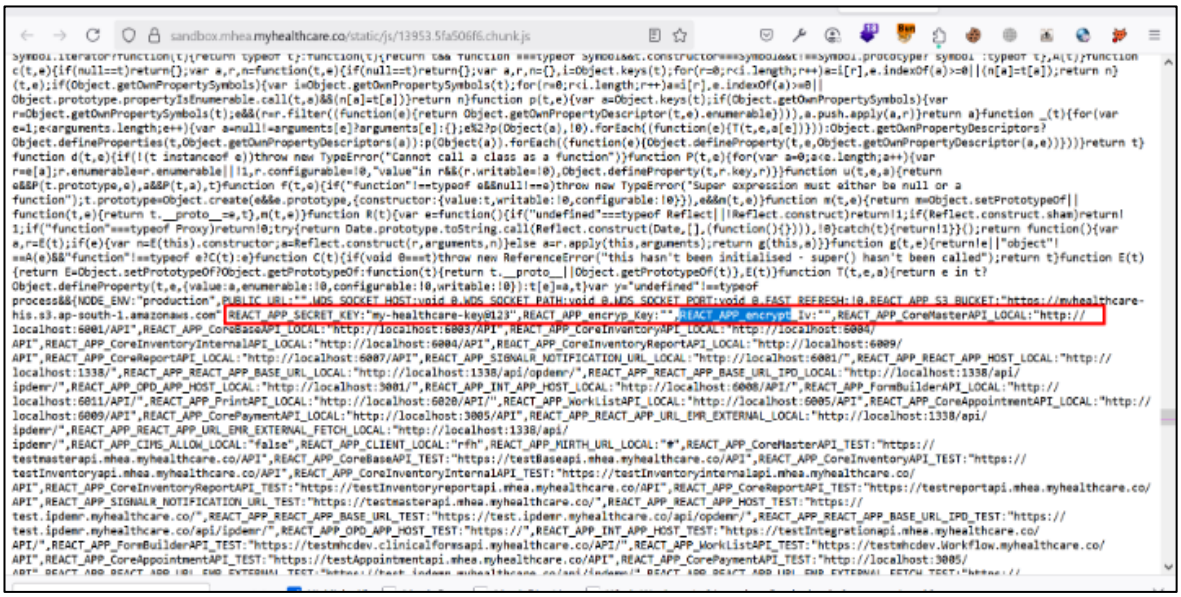
Sr. No.	Vulnerability Name	Severity	Level 2 Status
01	Hardcoded keys	High	Closed
02	Misconfigured CORS	Medium	Closed
03	Missing Security Headers	Low	Closed
04	Banner Grabbing	Low	Closed
05	Username Enumeration	Low	Closed

9. Graphical Representation



10. Detailed Technical Report with Recommendations

01: Hardcoded keys	
SEVERITY	**High**
STATUS	**Closed**
DESCRIPTION	
Application stores sensitive keys (API keys, encryption keys, access tokens, etc.) directly in the source code or configuration files. Anyone with code access or via reverse-engineering can extract these secrets. This leads to unauthorized access to internal services or third-party systems.	
IMPACT	
Attackers can use exposed keys to impersonate the application, access databases, modify	
AFFECTED URL	
GET /static/js/13953.331eee34.chunk.js HTTP/1	
RECOMMENDATIONS	
Remove keys from source code and store them securely using environment variables or secret	
PROOF OF CONCEPT	



02: Misconfigured CORS

SEVERITY	Medium
STATUS	Closed

DESCRIPTION

Misconfigured CORS occurs when a web server incorrectly allows unauthorized domains to access resources, exposing sensitive data to malicious websites.

IMPACT

It can lead to data theft, unauthorized access to APIs, and potential exploitation of sensitive user information.

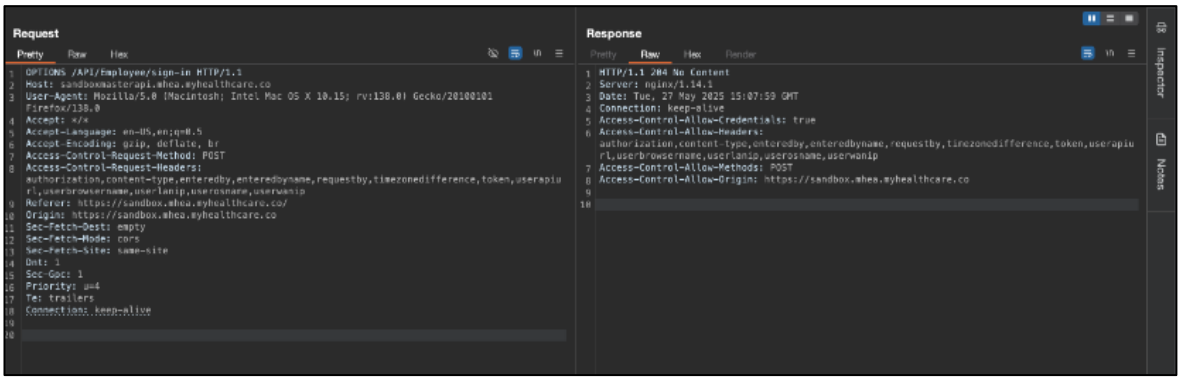
AFFECTED URL

Throughout the Application.

RECOMMENDATIONS

Properly configure CORS to only allow trusted domains, avoid using wildcard (*) for sensitive resources, and validate the origin header.

PROOF OF CONCEPT



03: Missing Security Headers

SEVERITY	Low
STATUS	Closed

DESCRIPTION

The application is missing important HTTP security headers that help protect against various attacks, such as Cross-Site Scripting (XSS), Clickjacking, and other web vulnerabilities. Key headers such as Content-Security-Policy, X-Frame-Options, X-XSS-Protection, and StrictTransport-Security are not present in the server's response.

IMPACT

The absence of these security headers increases the risk of attacks on the application. Without appropriate headers, the application may be more vulnerable to cross-site scripting, data injection attacks, and unauthorized framing of content, which can compromise user security and privacy

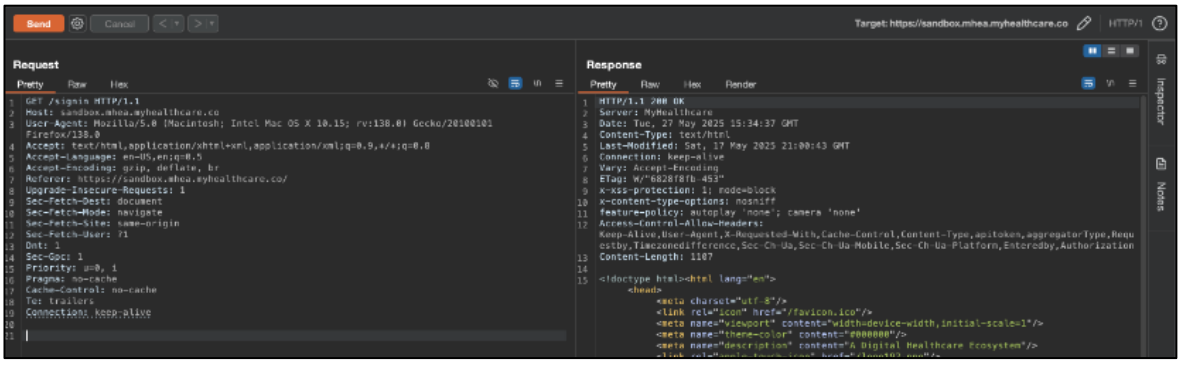
AFFECTED URL

Throughout the Application.

RECOMMENDATIONS

Implement the following security headers to enhance the security posture of the application:
- Content-Security-Policy to control resources the user agent is allowed to load.
- X-Frame-Options to prevent Clickjacking by disallowing the application to be rendered in a frame.
- X-XSS-Protection to enable cross-site scripting filters in web browsers.
- Strict-Transport-Security to enforce secure connections to the server.

PROOF OF CONCEPT



04: Banner Grabbing

SEVERITY	Low
STATUS	Closed

DESCRIPTION

Banner Grabbing is an enumeration technique used to glean information about computer systems on a network, server information and the services running its open ports

IMPACT

An attacker can access the cookie via non-Http In most cases, banner grabbing does not involve the leakage of critical pieces of information, but rather information that may aid the attacker through the exploitation phase of the attack. For example, if target leaks the version of PHP it is running on the server, and it happens to be vulnerable to Remote Command/Code Execution (RCE) because it wasn't updated, the attackers may exploit the known vulnerability and take full control of the web application. by using client-side script such as JavaScript

AFFECTED URL

Throughout the Application.

RECOMMENDATIONS

Remove all the unnecessary response headers which contains server information like server name or server version etc.

PROOF OF CONCEPT

05: Username Enumeration

SEVERITY	**Low**
STATUS	**Closed**

DESCRIPTION

Username enumeration is a vulnerability where an attacker can determine whether a username exists in a system by observing different error messages or response times during login
attempts.

IMPACT

It can lead to attackers identifying valid usernames, making it easier to launch further attacks like password guessing or brute force attacks.

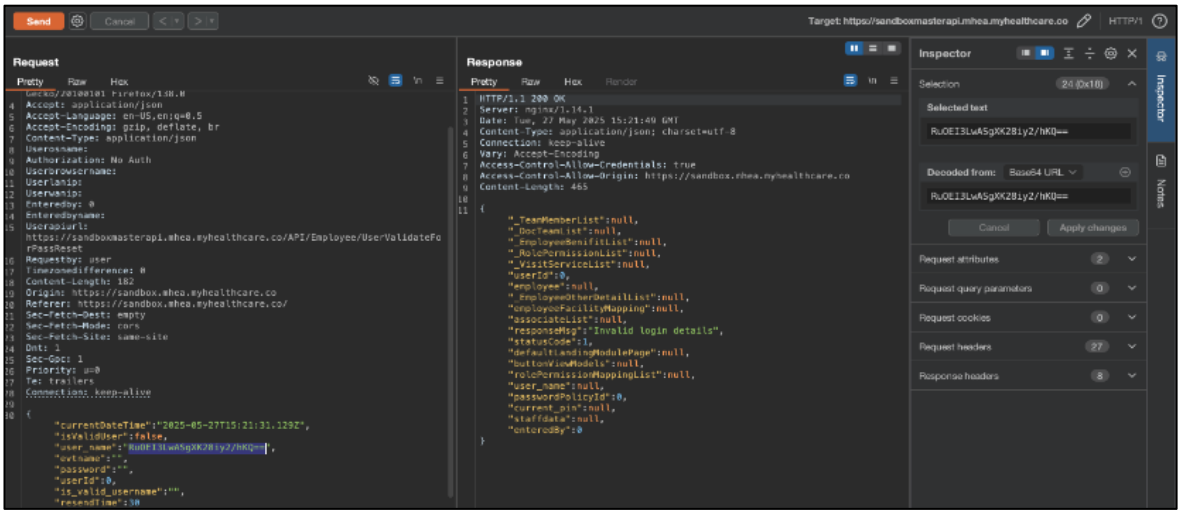
AFFECTED URL

Signing page.

RECOMMENDATIONS

Ensure that error messages are generic and do not reveal information about whether the username or password is incorrect. Implement rate limiting and account lockout mechanisms.

PROOF OF CONCEPT



11. General References

Application Security Standard –

https://owasp.org/Top10/

https://www.sans.org/top25-software-errors/

https://cert-in.org.in/

Hardening of Servers –

https://geekflare.com/apache-web-server-hardening-security/

https://geekflare.com/apache-tomcat-hardening-and-security-guide/

https://geekflare.com/nginx-webserver-security-hardening-guide/

https://geekflare.com/ibm-http-server-security-guide/

THE END OF DOCUMENT

14. ADDENDUM-4 (WASA Certificate 2025)

Addendums Attached - 4

WASA Certificate 2025

Imperium Solutions

Security Audit Certificate

Certificate No: IMS/2025-2026/089

Audit Details

Audit Firm	**:**	IMPERIUM SOLUTIONS
Cert IN Empanelment Reference	**:**	3(15)/2004-CERT-In (Vol. XIII)
Client Name	**:**	Myhealthcare Technologies Pvt Ltd
Scope of activity	**:**	Web Application Vulnerability Assessment and Penetration Testing (VAPT)
Application	**:**	MyHealthcare Enterprise Application
Production URL	**:**	https://mhea.myhealthcare.life
Tested URL	**:**	https://sandbox.mhea.myhealthcare.co/
Audit Duration	**:**	10-June-2025 to 17-June-2025
SHA256 Hash	**:**	0ad41814526204c2075422641423e1a9b0d2ecc6383 5979e70685cff774c5a5d
Audit Methodology	**:**	OWASP, SANS, Cert-IN Advisory, OSSTMM, PTES, ISSAF
Auditor Details	**:**	Mr. Sanjeev Chavan
Certificate Issue Date	**:**	17-June-2025
Certificate Validity	**:**	This certificate is valid till there are no changes in the application

Conclusion

The application has been tested in the UAT Environment, and all the vulnerabilities of the web application has been solved and are defined below.

Sr. No.	Vulnerability Name	Severity	Final Stats
01	Hardcoded keys	High	CLOSED
02	Misconfigured CORS	Medium	CLOSED
03	Missing Security Headers	Low	CLOSED
04	Banner Grabbing	Low	CLOSED
05	User Name Enumeration	Low	CLOSED

The application can be hosted in the production environment after implementing the compensatory controls.

General Recommendations

Production Hosting Environment

1. Deploy a Web Application Firewall ahead of your application facing the Internet and allow only relevant traffic to flow to your web server.
2. Fine tune the firewall rules such that only specific ports and IP addresses are allowed access to your application.
3. Enable, capture and retain application logs so as to be able to trace a security incident, in case it becomes necessary to do so.
4. Utilise services of 3rd party vendors to check for malicious activities like web defacement, etc.
5. Ensure that the Operating system, database and application is hardened.
6. Ensure that the Operating Systems, Database and applications is the latest stable version.
7. Application should undergo security testing annually or whenever any changes are implemented in the application functionality, whichever is earlier.
8. Ensure that all vulnerabilities, irrespective of their criticality, are resolved asap.
9. SOW Black box VAPT Intranet application

For Imperium Solutions,

Ms. Tasneam P
CISA (Certification No – 0977475)
Dated: 17-June-2025