Software Requirements Specification

for

Security on e-health applications (SEC-CAT*)

Version 1.0 approved

Prepared by Software Engineering Research Group

Murcia University

December 2023

Table of Contents

Revision History

Date	Revision	Description	Authors
12/12/2023	Revision 1.0	Initial	Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná

1. Introduction

This document describes a Software Requirements Specification (SRS) for developing secure eHealth applications. This SRS complies with the ISO/IEC/IEEE standard 29148-2018.

1.1 SRS overview

This document adheres to the organizational framework of the ISO/IEC/IEEE 29148:2018 standard (IEEE SA - IEEE/ISO/IEC 29148-2018, n.d.), which supersedes IEEE 830-1998. [2]

The Software Requirements Specification (SRS) requirements are detailed in section 2, specifically called "Requirements." A substantial portion of the subsections in this section are empty because this compendium comprises reusable requirements that pertain to a particular domain. All requirements for the security of this SRS can be found in subparagraph 2.9.3.

OWASP top ten 2021 [3], a complete and standardized awareness project that addresses the various aspects of security and risk in web applications, has been considered to establish ten subcategories containing security requirements.

1.1.1 Metainformation associated with the requirements

This subsection explains the attributes associated with this catalog's requirements. The features marked as “Mandatory” need to be set when the requirement is created. The rest of the attributes can be empty. The IEEE standard defines the following attributes:

- ✓ **PUID (Project Unique IDentification).**
  Mandatory. This identifier corresponds to the requirement's placement within the hierarchical list.

- ✓ **Requirement description.**
  Mandatory. Full description.

- ✓ **Source**
  The attribute signifies the origin of the necessity. This origin may stem from a customer's demand, a technical remedy, or a prescribed guideline, among other possibilities.

- ✓ **Priority**
  The analyst determines this attribute, which signifies the sequence in which the requirements are developed. This numerical representation is derived from the preceding two values, each of which can be classified as low, medium, or high.

- ✓ **Rationale**
  This attribute indicates the motivation for the requirement.

Some other attributes indicate relationships between the requirements:

- ✓ **Child PUIDs and Parent PUIDs.**
  A child PUID, also known as a trace-to relationship, pertains to a stipulation that possesses an inclusive dependency. Conversely, a parent PUID, also referred to as a trace-from relation, relates to a requirement with an inclusive dependence on it.

- ✓ **Exclusion PUIDs.**
  A PUID exclusion refers to a condition that contradicts its requirements. The attribute's value initially comprises the count of PUIDs followed by the actual PUID values.:
  # exclusion PUIDs: PUID, [PUID]*

In addition, the following attributes are also defined:

- ✓ **Critical nature.**
  This attribute denotes the level of significance the requirement holds for the client. The possible options for this attribute include low, medium, and high.

- ✓ **Current state.**
  This attribute indicates the current condition of the requirement. A total of nine distinct states have been identified:
  - ❖ **To_be_determined:** the requirement is included in the document but is not entirely described, or its description is not final.
  - ❖ **Pending_review:** the requirement was determined, but the customers have not reviewed it.
  - ❖ **Discarded:** the requirement is no longer needed or is not feasible.
  - ❖ **Approval:** the requirement is correct and was approved by the customers.
  - ❖ **Analysis_modeling:** the requirement was modelled in the analysis phase.
  - ❖ **Design_modeling:** the requirement was modelled in the design phase.
  - ❖ **Implemented:** the requirement is implemented in the project.
  - ❖ **Verified:** the requirement changes to this state when the technical team and the customer corroborate that the project accomplishes it.

- ✓ **Verification method**.
  This characteristic denotes the approach that must be employed to validate the project's fulfillment of the stipulation. Four options are available: inspection, analysis, demonstration, or evidence.

- ✓ **Validation criteria.**
  This characteristic signifies the prerequisites essential for substantiating the necessity. Typically, these prerequisites are encompassed within the documentation for STS (Software Test Specification).

- ✓ **Requested by.**
  This particular attribute denotes the individual who initiates the solicitation of the necessity.

- ✓ **Responsible.**
  This particular characteristic denotes the individual responsible for fulfilling the specified demand.

- ✓ **Configurable value.**
  This attribute encompasses the potential values of the requirements' parameters. This attribute's initial value comprises the parameter count and the corresponding data types.
  # Parameters: DataType, [DataType]*

- ✓ **Version history.**
  This characteristic encompasses the background information of the individual responsible for the stipulation, the specific date of creation, the iteration number, and a comprehensive explanation.

- ✓ **Author and date.**
  This feature shows the last author and date of the request. Only the required PUID and specification are mandatory, and other identifiers are optional.

The attribute of configurable value enables the inclusion of requirements that can be parameterized. A requirement that can be adjusted according to parameters encompasses a component that possesses the capability to assume various values contingent upon the particular undertaking in which the catalog is being repurposed.

When the catalog is reused on a specific application, all of the requirements are also reused. The attribute values have the potential to be modified with fresh values that are tailored to suit the particular application. The characteristics that may be attributed to a numerical value upon the reuse of the catalog encompass priority, criticality, present condition, method of verification, criteria for validation, originator, accountable entity, modifiable quantity, record of past iterations, and creator and date.

1.2 Purpose

The Software Requirements Specification (SRS) is an elaborate delineation for a particular software product, program, or collection of programs engineered to carry out designated functionalities within a well-defined setting. The principal aim of this SRS is to delineate the essential prerequisites for developing a safeguarded e-health application.

The present SRS pertains to a reusable catalog known as the security requirements catalog (SEC-CAT), which adheres to the guideline proposed by SIREN (SImple REuse of software requiremeNts) [4]. SIREN is a methodology grounded in software engineering standards and advocates reusing requirements. SIREN is characterized as a pragmatic approach encompassing document templates for requirements, a repository of reusable requirements, and a spiral process model. The requirements are systematically organized in catalogs that comply with the international recommendations on requirements, such as IEEE 830-1998 [2] or its updated version ISO/IEC/IEEE 29148:2018 [1], and also align with the guidelines outlined in IEEE 1233 ("IEEE Guide for Developing System Requirements Specifications," 1998).

The utilization of SIREN permits the reutilization and customization of this SRS for various projects. By adhering to the SIREN methodology, this proposed security catalog has been meticulously crafted as a comprehensive, coherent, unequivocal, dependable, provable, attributable, and adaptable document, incorporating international laws and Standards, scientific research, guidelines, good practices, and recommendations on requirements. SEC-CAT was a meticulously designed catalog that can be employed in formulating the specifications for any e-health application.

The target audience for this software requirements specification (SRS) encompasses individuals and sectors engaged in the secure planning, design, development, assessment, and auditing of electronic health (e-health) applications.

1.3 Scope

The scope of this document comprises security on e-health applications.

The following standards provided by the International Organization for Standardization (ISO) were considered for this document:

- ✓ ISO 27001:2022 Information security, cybersecurity, and privacy protection. Information security management systems. Requirements [6]. This standard is the base for establishing, implementing, maintaining, and continuously improving an information security management system.

- ✓ ISO 27002:2022 "Information security, cybersecurity, and privacy protection Information security controls" [7], designed to create industry and organization-specific guidelines for managing information security. This document considers the unique information security risks these industries and organizations face and follows the principles outlined in ISO 27001.

- ✓ ISO 27799:2016 Health informatics - Information security management in health using ISO/IEC 27002 [8]. This standard offers guidance to healthcare entities and other stewards of personal health data on the optimal approaches for safeguarding the confidentiality, integrity, and availability of said data due to the unique stipulations within the healthcare domain that necessitate adherence to privacy, accuracy, and verifiability, as well as accessibility of personal health data.

The following international laws, guides, and considerations were considered for this document:

- ✓ HIPAA [9]. This comprehensive regulation encompasses the requirements stipulated in Title II, subtitle F, sections 261 through 264 of the Health Insurance Portability and Accountability Act of 1996 (HIPAA), Public Law 104-191. These requirements mandate the adoption of protective measures to guarantee the confidentiality and integrity of this information while under the purview of HIPAA-covered entities (covered entities) and during its transmission among covered entities and between covered entities and external parties.

- ✓ HIPAA Security rules [10] This regulation encourages physicians to implement appropriate administrative, physical, and technical safeguards to ensure patient health information's confidentiality, integrity, and security are stored electronically (known as "ePHI").

- ✓ NIST SP 800-53 rev. 5 [11]. Within the confines of this manuscript, there is a delineation of the security and privacy measures that must be adhered to by information systems and organizations. These measures safeguard the operations, assets, individuals, external organizations, and the entire Nation against various perils and risks. Such risks include but are not limited to, malicious assaults, human fallibility, acts of nature, structural collapse, foreign intelligence agencies, and apprehensions regarding personal privacy.

- ✓ Privacy & Security Requirements and Considerations for Digital Health Solutions [12]. This recommendation delineates a collection of privacy and security considerations, which grants digital health solution providers the freedom to establish privacy and security requirements specific to their solutions. The format of the revised standard ISO/IEC 27002:2005 is adhered to in this document. All 11 security control objectives are encompassed in the subsequent sections:

    1. Risk Management
    2. Security Policy;

3. Organizing Information Security;
4. Asset Management
5. Human Resources Security;
6. Physical and Environmental Security;
7. Communications and Operational Management;
8. Access Control;
9. Information Systems Acquisition, Development, and Maintenance;
10. Security Incident Management;
11. Business Continuity Management; and
12. Compliance.

- ✓ DISA STIG Application Security and Development Security Technical Implementation Guide: 2022 [13]. This Guide aims to enhance the security of the information systems belonging to the Department of Defense (DOD) by implementing the necessary measures outlined in the National Institute of Standards and Technology (NIST) 800-53 and other relevant regulations.

- ✓ OWASP Top 10 Web Application Security Risks:2021 [3]. A nonprofit organization has created this Guide to enhance software security by distributing a periodically updated list of the ten most prevalent threats and potential mitigation mechanisms.

- ✓ CWE/SANS TOP 25 Most Dangerous Software Errors: 2021 [14]. In partnership with the National Cyber Security Division of the Department of Homeland Security, MITRE maintains the CWE (Common Weakness Enumeration) website. This online platform comprehensively explains the most critical software flaws alongside authoritative recommendations on preventing and reducing their impact. Additionally, it provides a wealth of information regarding over 700 other software errors, design flaws, and architectural weaknesses that have the potential to be exploited.

A literature review was also performed to get security recommendations for e-health software. Of the entire collection of 886 scholarly papers, a mere three articles [15], [16], [17] were ultimately chosen after undergoing a rigorous and meticulous selection process.

1.4 Product perspective

1.4.1 System interfaces

Not described.

1.4.2 User interfaces

Not described.

1.4.3 Hardware interfaces

Not described.

1.4.4 Software interfaces

Not described.

1.4.5 Communications interfaces

Not described.

1.4.6 Memory constraints

Not described.

1.4.7 Operations

Not described.

1.4.8 Site adaptation requirements

Not described.

1.4.9 Interfaces with services

Not described.

1.5 Product functions

Not described.

1.6 User characteristics

Not described.

1.7 Limitations

Not described.

1.8 Assumptions and dependencies

Not described.

1.9 Apportioning of requirements

Not described.

2. Requirements

2.1 Specified requirements

Not described.

2.2 External Interfaces

Not described.

2.3 Functions

Not described.

2.4 Usability requirements

Not described.

2.5 Performance requirements

Not described.

2.6 Logical database

Not described.

2.7 Design constraints

Not described.

2.8 Standards compliance

Not described.

2.9 Software system attributes

2.9.1 Reliability

Not described.

2.9.2 Availability

Not described.

2.9.3 Security

2.9.3.1 Broken Access Control

AUTHORIZATION

PUID: [SEC-CAT-BAC-001]

Requirement description: The application shall implement and enforce dual access control for [organization-defined: privileged commands and/or other organization-defined actions].

Source:

- ✓ **AC-3 ACCESS ENFORCEMENT**
  (2) ACCESS ENFORCEMENT | DUAL AUTHORIZATION
  Enforce dual authorization for [Assignment: organization-defined privileged commands and/or other organization-defined actions].
  Discussion: Dual authorization, also known as two-person control, reduces risk related to insider threats. Dual authorization mechanisms require the approval of two authorized individuals to execute. To reduce the risk of collusion, organizations consider rotating dual authorization duties. Organizations consider the risk associated with implementing dual authorization mechanisms when immediate responses are necessary to ensure public and environmental safety. [11]

- ✓ **V-222425:** The application must enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies. Design or configure the application to enforce access to application resources. [13]

- ✓ **A01:2021 – Broken Access Control:** Implement access control mechanisms once and re-use them throughout the application, including minimizing Cross-Origin Resource Sharing (CORS) usage. [3]

Priority: Not described

Rationale: Dual access control, or two-person control, is implemented to reduce the risk associated with insider threats. Approval from two authorized individuals is necessary to carry out critical actions, which helps mitigate security risks, especially when immediate responses are crucial to ensure public and environmental safety. The rotation of dual authorization duties will also be considered to reduce the risk of collusion.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Practical Execution:**

  1. Perform practical tests with privileged commands to ensure that authorization from two individuals is required.
  2. Verify that critical commands or actions defined by the organization are only executed after approval by two authorized people.

- ✓ **Dual Approval:**

  1. Confirm that the system requests and validates approval from two authorized individuals before executing critical commands or actions.
  2. Verify that dual approval is consistently required in all necessary instances**.**

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda

José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

- ✓ Verify that dual approval is consistently required in all necessary instances: The functionality was reviewed to eliminate allergy conditions and symptoms of a patient's medical record. However, a request for "Dual Access Control" was not performed before executing the method.

PUID: [SEC-CAT-BAC-002]

Requirement description: The application must incorporate robust access control mechanisms to prevent circumvention of access control checks by manipulating the URL. The following access control bypass tactics should be avoided and counteracted:

- ❖ Modification of the URL (parameter manipulation or forced navigation).
- ❖ Modification of the internal state of the application.
- ❖ Modification of the HTML page.
- ❖ Use of attack tools to modify API requests.

Source:

- ✓ **A01:2021 – Broken Access Control**

  Bypassing access control checks by modifying the URL (parameter tampering or force browsing), internal application state, or the HTML page, or by using an attack tool modifying API requests. [3]

Priority: Not described

Rationale: Access control is critical to application security, and circumvention of these controls can result in unauthorized access to sensitive resources and data. Implementing robust access control mechanisms helps prevent vulnerabilities associated with Broken Access Control.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To_be_determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Verification of Access Controls:**

  1. Access the application using valid credentials.
  2. Try to access resources and features for which you are not authorized.
  3. Verify that unauthorized access is prevented and an error message or access denied page is displayed.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

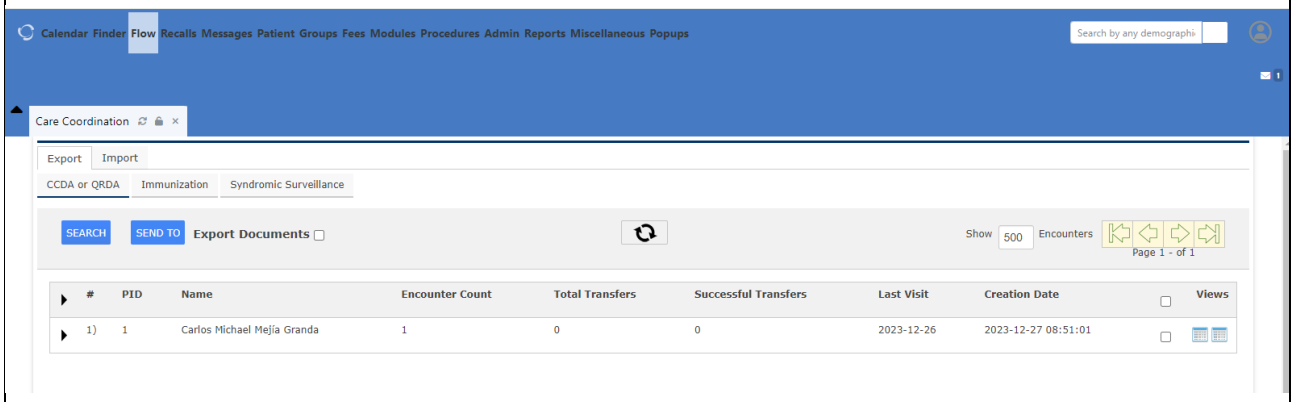
Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

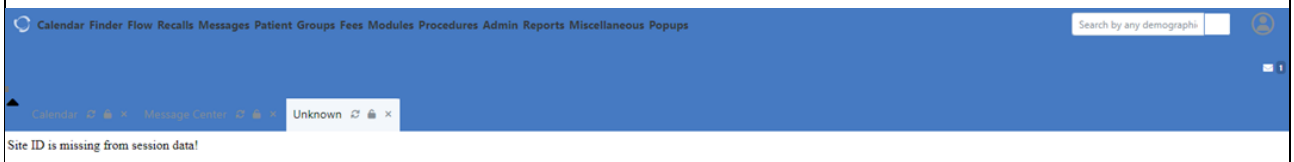
Auditory result:

A DAST has been performed, and there are no findings, so the steps of purchase steps were followed:

1. Log in as an "administrator" user, and you have the option "MODULES -> Care Coordination."



2. When entering with an “Emergency Login” profile user to the URL of point 1, the following error message is presented:



Result: It complies

PUID: [SEC-CAT-BAC-003]

Requirement description: The application must implement security measures to prevent the elevation of privileges. Prevent situations where a user can act without being authenticated, or an administrator can act as a regular user after logging in.

Source:

- ✓ **A01:2021 – Broken Access Control**: Avoid elevation of privilege. Acting as a user without being logged in or acting as an admin when logged in as a user. [3]

- ✓ **CWE-306: Missing Authentication for Critical Function**

  The product does not perform any authentication for functionality that requires a provable user identity or consumes a significant amount of resources. [14]

Priority: Not described

Rationale: Preventing privilege escalation and ensuring proper authentication on critical functions are critical to preventing unauthorized access and protecting application resources.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

1. **Test Scenario: Authentication Bypass Attempt:**

   Description: Attempt to perform actions within the application without authenticating or with invalid credentials.
   Expected Result: The application should reject the action and require proper authentication.

2. **Test Scenario: Privilege Evasion:**

   Description: Attempt to perform actions that require administrator privileges while authenticated as a regular user.
   Expected Result: The application should reject the action and require administrator credentials.

3. **Test Scenario: Authentication in Critical Functions:**

   Description: Attempt to access critical functions without authenticating or with invalid credentials.
   Expected Result: The application should require valid authentication before allowing access to critical functions.

4. **Source Code Review:**

   Description: Review the application source code to identify the presence of access controls and authentication on critical functions.
   Expected Result: There should be evidence of strong authentication and access controls in critical areas of the code.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

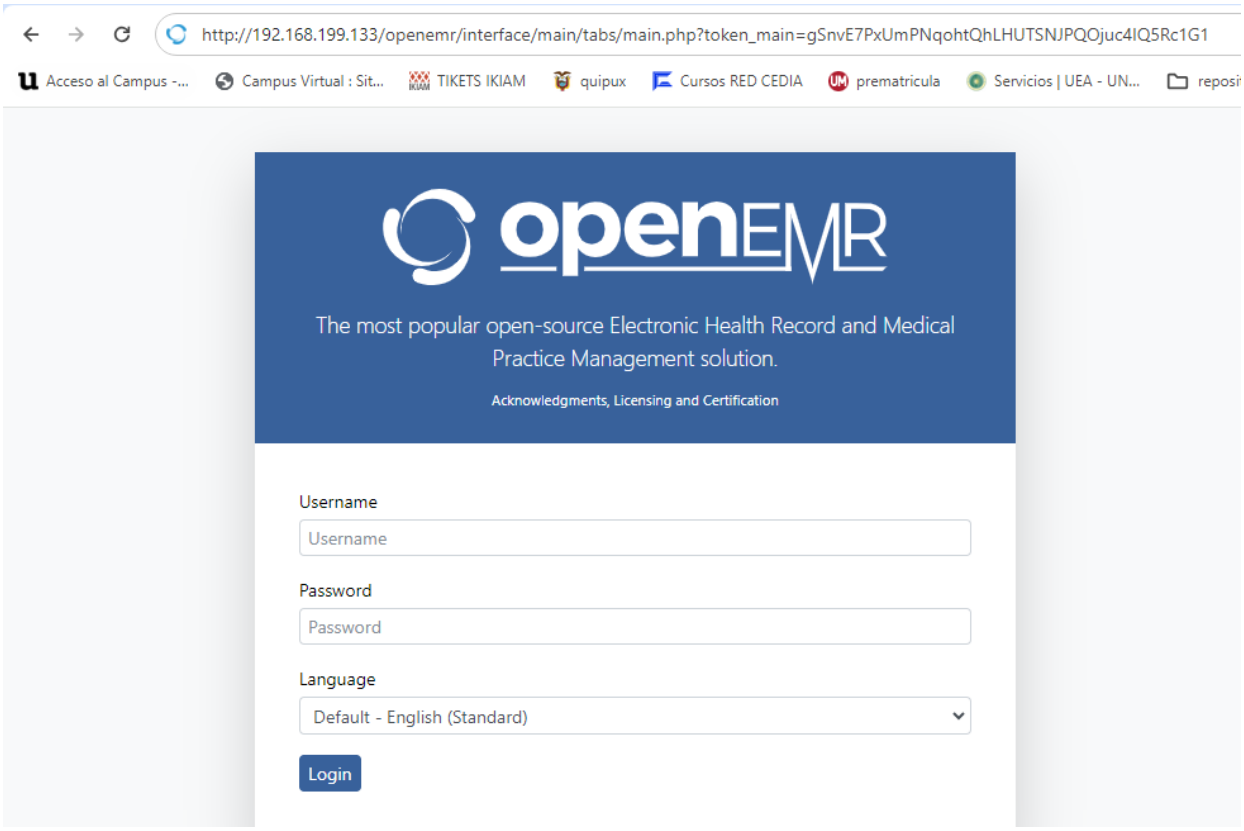
Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are no findings, so the steps of purchase steps were followed:

1. The global application configuration was tried to access without being authenticated, and the application was automatically redirected to the login.



Result: It complies

PUID: [SEC-CAT-BAC-004]

Requirement description:
The application must enforce strict and flexible access controls for authorized users, preventing unauthorized access to the API and ensuring record ownership.

Source:

- ✓ **V-222425**: The application must enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies. Design or configure the application to enforce access to application resources. [13]

- ✓ **V-222511**: The application must enforce access restrictions associated with changes to application configuration.

  Configure the application to limit access to configuration settings to only authorized users. [13]

- ✓ **A01:2021 – Broken Access Control:**
  - Accessing API with missing access controls for POST, PUT, and DELETE
  - Model access controls should enforce record ownership rather than accepting that the user can create, read, update, or delete any record. [3]

- ✓ **CWE-862: Missing Authorization**

The product does not perform an authorization check when an actor attempts to access a resource or perform an action. [14]

- ✓ Access must be carefully controlled and monitored to prevent abuses, and it must be restricted as much as possible but yet flexible [15]

Priority: Not described

Rationale: Implementing strong access controls and enforcing configuration restrictions are critical to preventing unauthorized access and ensuring application security.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To_be_determined

Verification method: Demonstration/Analysis

Validation criteria:

1. **Authorized Access Scenario:**

   Step 1: Access the app with the valid credentials of an authorized user.
   Step 2: Try to perform operations on the API, such as POST, PUT, and DELETE.
   Expected Result: You should be able to perform these operations without problems.

2. **Unauthorized API Access Scenario:**

   Step 1: Try to access the API with invalid credentials or without authentication.
   Expected Result: You should receive an error message or be redirected to an access denied page.

3. **Unauthorized Configuration Change Scenario:**

   Step 1: Try to make app settings changes without the necessary permissions.
   Expected Result: You should receive an error message or be redirected to an access denied page.

4. **Flexible Access Scenario:**

   Step 1: Verify that access controls allow the flexibility to perform permitted actions.
   Expected Result: You should be able to perform permitted actions without excessive restrictions.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The application is not an API that offers methods: PUT, POST, DELETE, or GET; therefore, it is considered not applicable for the audit (N/a).

Result: N/a

DISCRETIONARY ACCESS

PUID: [SEC-CAT-BAC-005]

Requirement description: The application must prevent unauthorized access to data processing devices and assign roles to users to protect the integrity of ePHI against unauthorized alterations.

Source:

§ **164.312** Technical safeguards.

(a)

(1) Standard: Access control: Implement technical policies and procedures for electronic information systems that maintain electronic protected health information to allow access only to those persons or software programs that have been granted access rights as specified in § 164.308(a)(4)

(c)

(2) Implementation specification: Mechanism to authenticate electronic protected health information (Addressable).

Implement electronic mechanisms to corroborate that electronic protected health information has not been altered or destroyed in an unauthorized manner. [10].

- ✓ There should be a mechanism of allowing some users access from only certain locations while allowing others to use any location. A receptionist, for example, only needs access from the reception desk while a physician should be able to log on from any location especially in the event of an emergency [15]
- ✓ Do not admit unauthorized persons to the data processing device [16]
- ✓ Make sure that data are accessible only according to authorization [16]
- ✓ **V-222426**: The application must enforce organization-defined discretionary access control policies over defined subjects and objects. Design and configure the application to enforce discretionary access control policies. [13]
- ✓ 5.18 Access rights: Access rights to information and other associated assets should be provisioned, reviewed, modified and removed in accordance with the organization's topic-specific policy on and rules for access control. It could be amplified in detail, review guidance section. [7]
- ✓ 9.2.3 Management of privileged access rights
  1. Health information systems should associate users (including health professionals, supporting staff and others) with the records of subjects of care and allow future access based on this association.
  2. Systems containing personal health information should support role-based access control capable of mapping each user to one or more roles and each role to one or more system functions.
  3. A user of a health information system containing personal health information shall access its services in a single role. [8]

- ✓ **CWE-269: Improper Privilege Management**

  The product contains a code sequence that can run concurrently with other code, and the code sequence requires temporary, exclusive access to a shared resource, but a timing window exists in which the shared resource can be modified by another code sequence that is operating concurrently.

  The product does not properly assign, modify, track, or check privileges for an actor, creating an unintended sphere of control for that actor. [14]

- ✓ **Security Requirement 61** – Granting Access By Association: The EHRi and all PoS systems connected to the EHRi:
  a) Must be capable of associating users (i.e. healthcare providers) with the records of patients/persons and allowing future access based on this association (i.e., they must be capable of granting discretionary access to records based on a registered user with legitimate and pre-existing access to a patient's record(s) granting access rights for that (those) record(s) to another registered user); and
  b) Must not allow users to grant other users access to a record if the granting users themselves do not possess such access with respect to the record.

  Note that granting other users access to a record does not override the role-based access control restrictions of those other users. [12]

Priority: Not described

Rationale: This requirement ensures the implementation of robust security and access control measures to protect the integrity and confidentiality of electronically protected health information.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Penetration tests:**

  1. Attempting to access data processing devices without authorization.
  2. Verify authentication and confirm the inability to alter ePHI in an unauthorized manner.

- ✓ **User Association Simulation:**

  1. Simulate situations where users try to grant access to records without having the necessary privileges.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

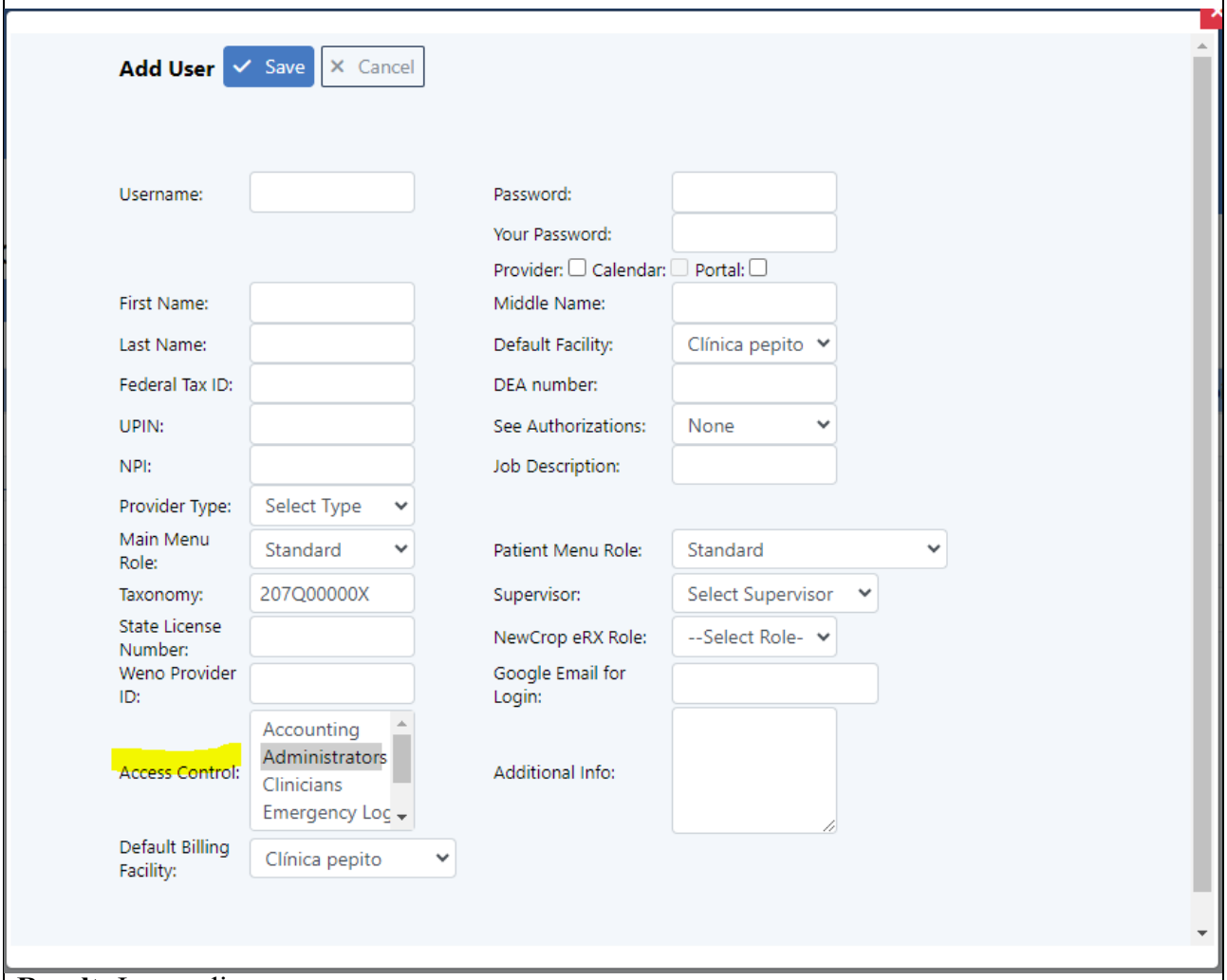
Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná

13/12/2023

Auditory result:

When entering the user manager as an administrator, it is evident that different roles and access levels are established when creating a new user:



Result: It complies

WORK GROUP ACCESS GRANTING

PUID: [SEC-CAT-BAC-006]

Requirement description: The system must allow users to be assigned to workgroups and grant access to records based on these groups. This access may be through a workstation, transaction, program, process, or other mechanisms.

Source:

- ✓ **§ 164.308** Administrative safeguards: (a) A covered entity or business associate must, in accordance with § 164.306:
  (4)

(B) Access authorization (Addressable).
Implement policies and procedures for granting access to electronic protected health information, for example, through access to a workstation, transaction, program, process, or other mechanism [10].

- ✓ Security Requirement 59 – Granting Access to Users in Workgroups: The EHRi and all PoS systems connected to the EHRi must be capable of assigning users to working groups and granting access to records based on working groups. [12]

Priority: Not described

Rationale: This requirement ensures the implementation of administrative measures to manage and control access to electronically protected health information, ensuring that only authorized users can access ePHI.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Assignment to Work Groups:**

  1. Confirm that the system allows the assignment of users to work groups.

- ✓ **Granting Group-Based Access:**

  1. Verify that the system can grant access to records based on workgroup membership.

- ✓ **Group Assignment Capacity and Access by Workgroups:**

  1. Confirm that the ability to assign users to workgroups and grant access to records based on these groups is present in the application, meeting "Security Requirement 59".

Requested by: The organization

Responsible: Developer

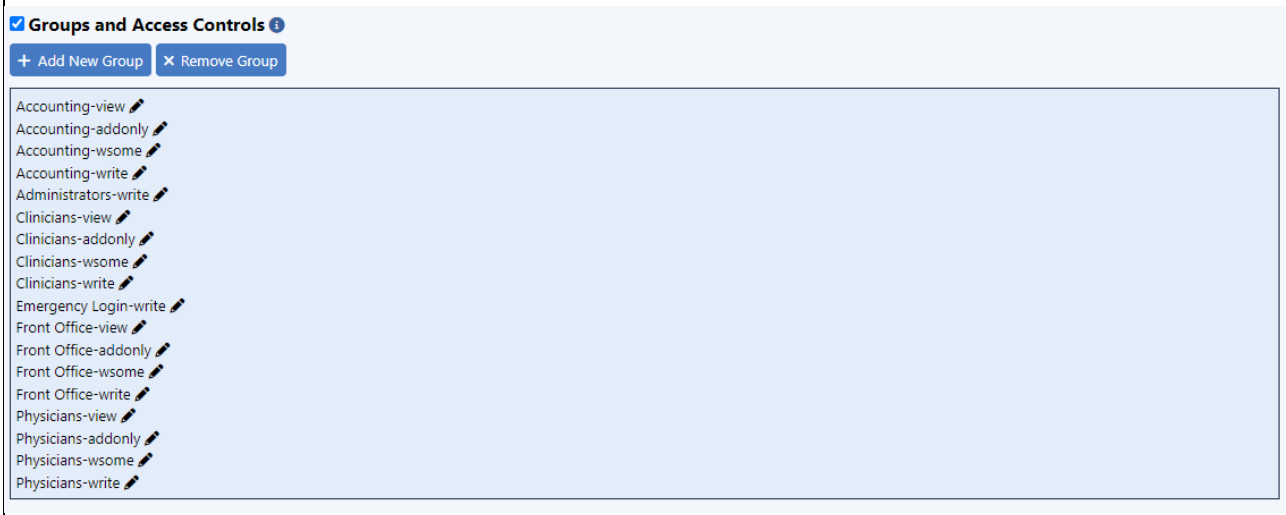
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

There is, in the system, as an administrator user, the option to create groups and assign users to these groups.



Result: It complies

PUID: [SEC-CAT-BAC-007]

Requirement description: The system must implement measures to control who uses the data processing system by ensuring that users are authenticated with an individual authenticator before using a group authenticator

Source:

- ✓ Control who uses the data processing system [16]
- ✓ **V-222529:** The application must ensure users are authenticated with an individual authenticator prior to using a group authenticator. Design and configure the application to individually authenticate group account members prior to allowing access.

  A group authenticator is a shared account or some other form of authentication that allows multiple unique individuals to access the application using a single account.

  If an application allows or provides for group authenticators, it must first individually authenticate users prior to implementing group authenticator functionality. [13]

Priority: Not described

Rationale: This requirement seeks to strengthen the system's security by authenticating individual users before allowing access through group authenticators, guaranteeing the traceability of actions to specific users.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Technical compliance:**

1. It will be verified that the application has been designed and configured according to the guidelines of V-222529, ensuring individual authentication of group account members before allowing access.
2. The implementation of technical measures that ensure that, in group authenticators, users are individually authenticated before enabling the group authenticator functionality will be evaluated.

Requested by: The organization

Responsible: Developer

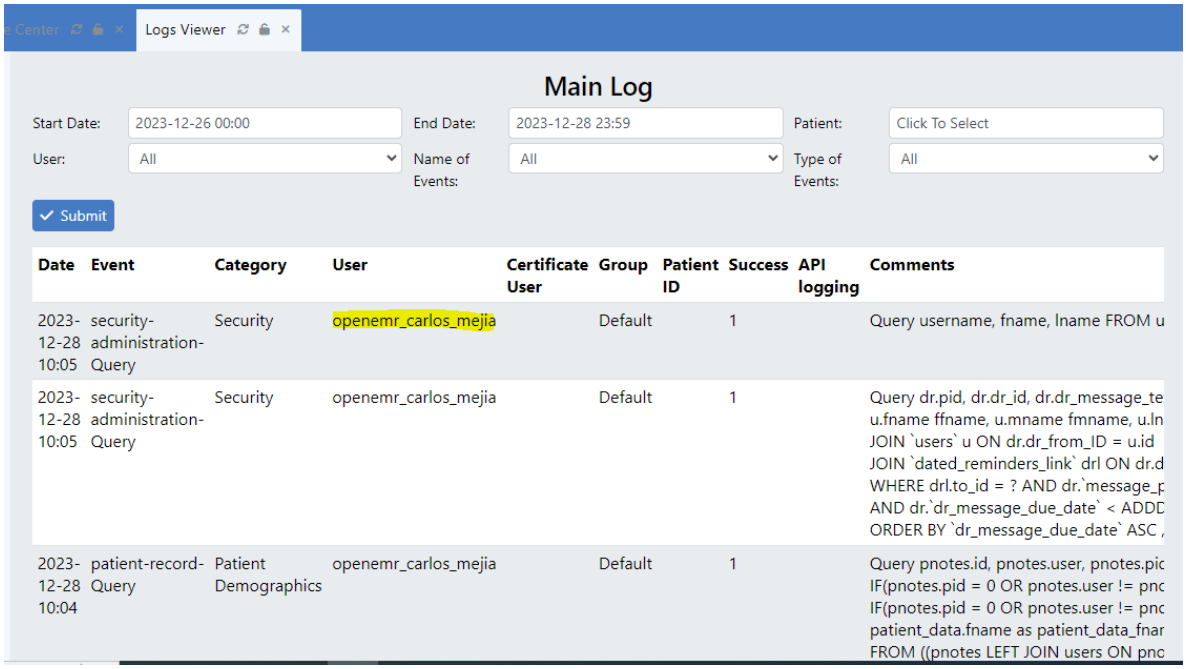
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

It is verified in the application and in the table of registration of activities that there is a single identifier that links the activities carried out by each user and not at a group character.



Result: It complies

USER REGISTRATION

PUID: [SEC-CAT-BAC-008]

Requirement description: The system is required to establish a formal process for user registration and deregistration, facilitating access to health information systems. This process must guarantee a consistent alignment between the authentication level, the mapping of users to roles, roles to system functions, and the level of access to be granted to the user.

Source:
- ✓ 9.2.1 User registration and de-registration: Access to health information systems that process personal health information shall be subject to a formal user registration process. User registration procedures shall ensure that the level of authentication required of claimed user identity is consistent with the level(s) of access that will become available to the user. [8]
- ✓ Security Requirement 57 – Granting Access to Users by Role: The EHRi and all PoS systems connected to the EHRi must support role-based access control (RBAC) capable of mapping each user to one or more roles and each role to one or more system functions. [12]

Priority: Not described

Rationale: This requirement ensures a structured and secure user registration process and effective access control by assigning roles based on specific system functions.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Formal Registration and Deregistration Process:**

  1. The presence of a formal process that allows the registration and departure of users in a structured and secure manner will be verified.

Requested by: The organization

Responsible: Developer

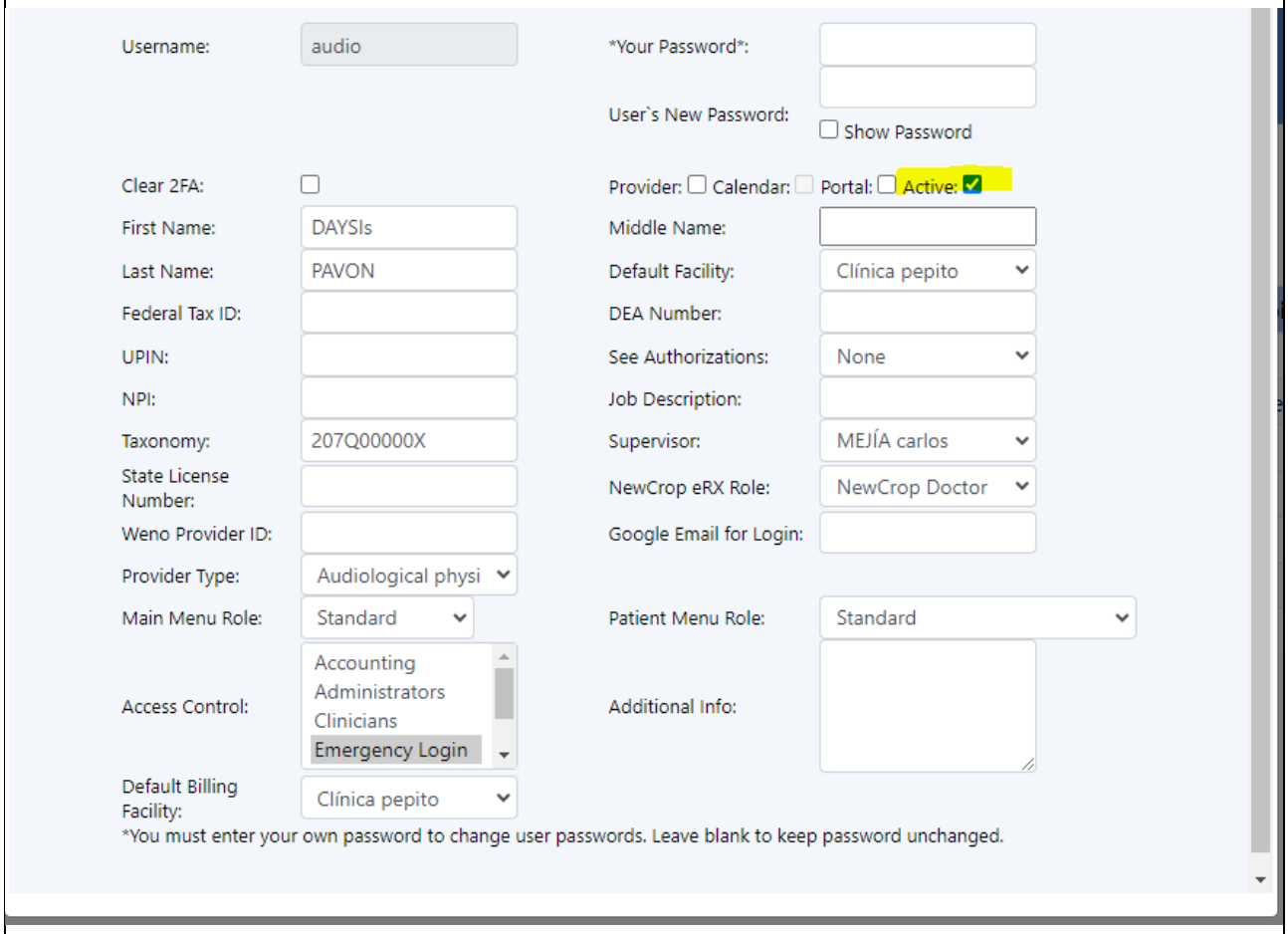
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

There is application functionality to execute a formal process of registration and deactivation of system users.



Result: It complies

2.9.3.2 Cryptographic Failures

PUID: [SEC-CAT-CRF-001]

Requirement description: The system must implement a robust FIPS-validated cryptographic infrastructure to encrypt and decrypt electronically protected health information to ensure adequate protection of sensitive information.

Source:

- ✓ **§ 164.312** Technical safeguards.
  - **(a)**
    - **(2)** Implementation specifications:
      - **(iv)** Encryption and decryption (Addressable): Implement a mechanism to encrypt and decrypt electronic protected health information.
  - **(e)**
    - **(2)** Implementation specifications:
      - **(ii)** Encryption (Addressable): Implement a mechanism to encrypt electronic protected health information whenever deemed appropriate [10]

- ✓ **8.24 Use of cryptography**: Rules for the effective use of cryptography, including cryptographic key management, should be defined and implemented. [7]

- ✓ **A02:2021 – Cryptographic Failures:**

    - Make sure to encrypt all sensitive data at rest.
    - Ensure up-to-date and strong standard algorithms, protocols, and keys are in place; use proper key management. [3]

- ✓ **SC-12** CRYPTOGRAPHIC KEY ESTABLISHMENT AND MANAGEMENT

  Control: Establish and manage cryptographic keys when cryptography is employed within the system in accordance with the following key management requirements: [Assignment: organization-defined requirements for key generation, distribution, storage, access, and destruction].

  Discussion: Cryptographic key management and establishment can be performed using manual procedures or automated mechanisms with supporting manual procedures. Organizations define key management requirements in accordance with applicable laws, executive orders, directives, regulations, policies, standards, and guidelines and specify appropriate options, parameters, and levels. Organizations manage trust stores to ensure that only approved trust anchors are part of such trust stores. This includes certificates with visibility external to organizational systems and certificates related to the internal operations of systems. [NIST CMVP] and [NIST CAVP] provide additional information on validated cryptographic modules and algorithms that can be used in cryptographic key management and establishment. [11].

- ✓ **V-222572:** The application must utilize FIPS-validated cryptographic modules when protecting unclassified information that requires cryptographic protection. Configure the application to use a FIPS-validated cryptographic module.

- ✓ **V-222583:** The application must use the Federal Information Processing Standard (FIPS) 140-2-validated cryptographic modules and random number generator if the application implements encryption, key exchange, digital signature, and hash functionality.

  Use of weak or untested encryption algorithms undermines the purposes of utilizing encryption to protect data. The application must implement cryptographic modules adhering to the higher standards approved by the federal government since this provides assurance they have been tested and validated. [13]

- ✓ **V-222588:** The application must implement approved cryptographic mechanisms to prevent unauthorized modification of organization-defined information at rest on organization-defined information system components.

    - Identify data elements that require protection.
    - Document the data types and specify encryption requirements.
    - Encrypt data according to DoD policy or data owner requirements. [13]

- ✓ **V-222589:** The application must use appropriate cryptography in order to protect stored DoD information when required by the information owner or DoD policy. Identify data elements that require protection.

    - Document the data types and specify encryption requirements.

- Encrypt classified data using Type 1, Suite B, or other NSA-approved encryption solutions. [13]

✓ **A02:2021 – Cryptographic Failures:**

- Store passwords using strong adaptive and salted hashing functions with a work factor (delay factor), such as Argon2, scrypt, bcrypt or PBKDF2.
- Avoid deprecated cryptographic functions and padding schemes, such as MD5, SHA1, PKCS number 1 v1.5 [3]

✓ **V-254803:** The application must implement NSA-approved cryptography to protect classified information in accordance with applicable federal laws, Executive Orders, directives, policies, regulations, and standards.

Configure application to encrypt stored classified information; Ensure encryption is performed using NIST FIPS 140-2-validated encryption.

Encrypt stored, non-SAMI classified information using NIST FIPS 140-2-validated encryption.

Implement NSA-validated type-1 encryption of all SAMI data stored in the enclave.

Use of weak or untested encryption algorithms undermines the purposes of utilizing encryption to protect classified data. The application must implement cryptographic modules adhering to the higher standards approved by the federal government since this provides assurance they have been tested and validated.

Advanced Encryption Standard (AES)
Symmetric block cipher used for information protection
FIPS Pub 197
Use 256-bit keys to protect up to TOP SECRET

Elliptic Curve Diffie-Hellman (ECDH) Key Exchange
Asymmetric algorithm used for key establishment
NIST SP 800-56A
Use Curve P-384 to protect up to TOP SECRET.

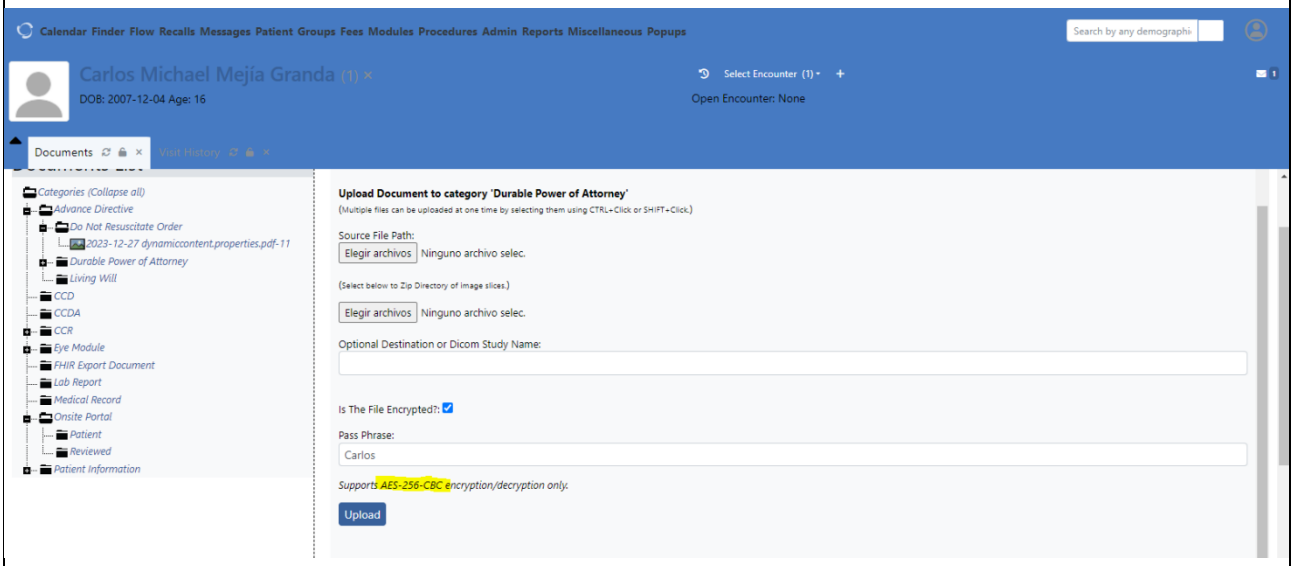
Elliptic Curve Digital Signature Algorithm (ECDSA)
Asymmetric algorithm used for digital signatures
FIPS Pub 186-4
Use Curve P-384 to protect up to TOP SECRET.

Secure Hash Algorithm (SHA)
Algorithm used for computing a condensed representation of information
FIPS Pub 180-4

Use SHA-384 to protect up to TOP SECRET.

Diffie-Hellman (DH) Key Exchange
Asymmetric algorithm used for key establishment
IETF RFC 3526
Minimum 3072-bit modulus to protect up to TOP SECRET

RSA Asymmetric algorithm used for key establishment NIST SP 800-56B rev 1 Minimum 3072-bit modulus to protect up to TOP SECRET RSA Asymmetric algorithm used for digital signatures FIPS PUB 186-4 Minimum 3072 bit-modulus to protect up to TOP SECRET. [13]
Priority: Not described
Rationale: This requirement ensures that the implementation of cryptography in the system complies with recognized standards and guarantees adequate protection of sensitive information, especially that classified as critical and subject to specific regulations.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: **1.** The cryptographic infrastructure must be FIPS validated. **2.** The use of weak or untested encryption algorithms should be avoided. **3.** The application must use FIPS-validated cryptographic modules, as indicated in V-222572 and V-222583. **4.** The application must implement encryption for classified information according to specified standards, such as AES, ECDH, ECDSA, SHA, Diffie-Hellman, and RSA.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The system in the "Patient Documents" option allows you to encrypt and decipher data through the AES-256-CBC algorithm that meets FIPS.



Result: It complies

PUID: [SEC-CAT-CRF-002]

Requirement description: The application must ensure the security of integrity and confidentiality during the transmission of ePHI over electronic networks following the technical safeguards standards established in the regulatory framework

Source:

- ✓ **§ 164.312** Technical safeguards.
  (e)
  (1) Standard: Transmission security.
  Implement technical security measures to guard against unauthorized access to electronic protected health information that is being transmitted over an electronic communications network.
  (2) Implementation specifications:
  (i) Integrity controls (Addressable): Implement security measures to ensure that electronically transmitted electronic protected health information is not improperly modified without detection until disposed of. [10].

- ✓ **A02:2021 – Cryptographic Failures:** Encrypt all data in transit with secure protocols such as TLS with forward secrecy (FS) ciphers, cipher prioritization by the server, and secure parameters. Enforce encryption using directives like HTTP Strict Transport Security (HSTS). [3]

- ✓ **SC-8 TRANSMISSION** CONFIDENTIALITY AND INTEGRITY

  Control: Protect the [Selection (one or more): confidentiality; integrity] of transmitted information.

  Discussion: Protecting the confidentiality and integrity of transmitted information applies to internal and external networks as well as any system components that can transmit information, including servers, notebook computers, desktop computers, mobile devices, printers, copiers, scanners, facsimile machines, and radios. Unprotected communication

paths are exposed to the possibility of interception and modification. Protecting the confidentiality and integrity of information can be accomplished by physical or logical means. Physical protection can be achieved by using protected distribution systems. A protected distribution system is a wireline or fiber-optics telecommunications system that includes terminals and adequate electromagnetic, acoustical, electrical, and physical controls to permit its use for the unencrypted transmission of classified information. Logical protection can be achieved by employing encryption techniques. [11]

- ✓ During transferring and syncing information between networked and connected healthcare devices, data should be encrypted from endpoint to endpoint [17]

- ✓ **Security Requirement 30** – Encrypting PHI During Transmission: The EHRi and PoS systems connected to the EHRi must apply industry-standard cryptographic algorithms and protocols during transmission of PHI to maintain the confidentiality and integrity of this data whenever it is transmitted outside the physical security perimeter that protects information processing facilities supporting EHRi servers, applications or data [12]

- ✓ **Consideration MC22** – Ensure Communications Channel Encryption: Organizations should ensure that all mobile communications channels that transmit or receive confidential information are encrypted. [12].

- ✓ **V-222596:** The application must protect the confidentiality and integrity of transmitted information. Configure all of the application systems to require TLS encryption in accordance with data protection requirements. [13]

- ✓ **V-222599:** The application must maintain the confidentiality and integrity of information during reception. Configure all of the application systems to require TLS encryption.

  Service-Oriented Architecture (SOA) and RESTFUL web services allow for XML-based application data to be transmitted in a manner similar to network traffic wherein the application data is transmitted along multiple servers' hops [13]

Priority: Not described

Rationale: This requirement is intended to ensure that ePHI is transmitted safely and securely, mitigating the risk of unauthorized access and modifications and guaranteeing the confidentiality and integrity of the information throughout the transmission process.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria: Not Described

Requested by: The organization

Responsible: Developer

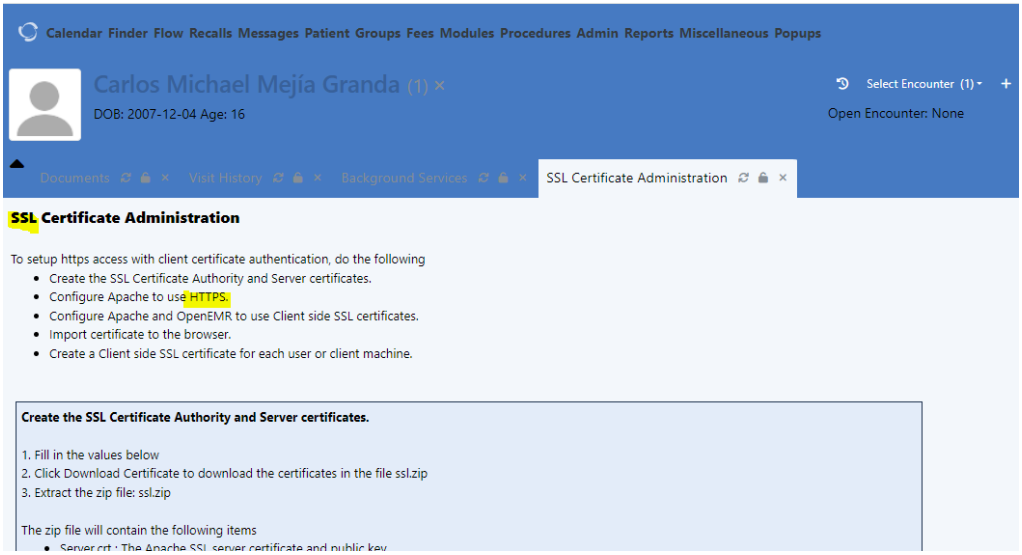
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST has been performed, and there are no findings, so the platform was entered as an administrator user, and there is an option to configure communications with SSL certificates and implement HTTPS.



Result: It complies

PUID: [SEC-CAT-CRF-003]

Requirement description: The application must ensure secure storage of the session identifier, ensuring that at no time is the generated session ID included in requests via the URL

Source:

- ✓ **A07:2021 – Identification and Authentication Failures:**
  - Use a server-side, secure, built-in session manager that generates a new random session ID with high entropy after login. Session identifier should not be in the URL, be securely stored, and invalidated after logout, idle, and absolute timeouts. [3]

- ✓ **V-222581:** Applications must not use URL embedded session IDs. Configure the application to transmit session ID information via cookies.

  Using a session ID that is copied to the URL introduces the risks that the session ID information will be written to log files, made available in browser history files, or made publicly available within the URL.

  Using cookies to establish session ID information is desired. [13]

Priority: Not described

Rationale: This requirement seeks to safeguard the integrity and confidentiality of session information, mitigating potential threats and ensuring secure handling of sessions in the application.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ Compliance with the SEG-CAT-CRF-002 requirement will be evaluated by implementing and verifying technical security measures while transmitting electronic protected health information (ePHI) over electronic networks.
- ✓ SC-8 TRANSMISSION CONFIDENTIALITY AND INTEGRITY: The application must protect the confidentiality and integrity of the transmitted information.
- ✓ V-222599: The application must maintain the confidentiality and integrity of information during reception by configuring all application systems to require TLS encryption
- ✓ A02:2021 – Cryptographic Failures: The application must encrypt all data in transit using secure protocols such as TLS with perfect forward secrecy (FS) ciphers, server-side cipher prioritization, and secure parameters. Encryption must be enforced using HTTP Strict Transport Security (HSTS) policies.
- ✓ **Expected results:**

  The application is expected to comply with the standards above and specifications, thus ensuring the adequate security of ePHI during its electronic transmission. Proper implementation of encryption, secure protocols, and integrity measures will be key factors in evaluating compliance with this requirement.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST has been performed, and there are no findings. Moreover, after analyzing the headers of the requests and observing the URLs of page applications, it is evident that they are sent as a parameter in the URL SESSION IDS.

Result: It complies

PUID: [SEC-CAT-CRF-004]

Requirement description: The application must ensure message encryption when the SessionIndex is linked to privacy data

Source:

- ✓ **V-222406:** The application must ensure messages are encrypted when the SessionIndex is tied to privacy data. Encrypt messages when the SessionIndex is tied to privacy data. [13]
- ✓ **V-222583:** The application must use the Federal Information Processing Standard (FIPS) 140-2-validated cryptographic modules and random number generator if the application implements encryption, key exchange, digital signature, and hash functionality.

- ✓ Use of weak or untested encryption algorithms undermines the purposes of utilizing encryption to protect data. The application must implement cryptographic modules adhering to the higher standards approved by the federal government since this provides assurance they have been tested and validated. [13]

Priority: Not described

Rationale: This requirement is intended to preserve the secrecy and accuracy of sensitive health information when it is communicated beyond physical security; it is essential to utilize widely accepted cryptographic algorithms and protocols commonly used and accepted in the industry.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To_be_determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Message Encryption Verification:**

  1. Send a message with SessionIndex linked to privacy data.
  2. Verify that the message is encrypted according to FIPS 140-2 standards.

- ✓ **Validation of Cryptographic Modules:**

  1. Confirm that the application uses validated FIPS 140-2 cryptographic modules.
  2. Verify that a random number generator is used under FIPS 140-2 requirements.

- ✓ **Encryption Algorithm Testing:**

  1. Evaluate that the application does not use weak or untested encryption algorithms.
  2. Verify that the implementation of cryptographic modules meets standards.

- ✓ **Success Criteria:**

  1. All messages with SessionIndex linked to privacy data must be encrypted.
  2. The application must use FIPS 140-2 validated cryptographic modules and an appropriate random number generator.
  3. Encryption algorithms considered weak or untested should not be used.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are no findings. Moreover, as reviewed, the application does not make "privacy data" shipments Linked to SessionIndex.

Result: It does not apply

CRYPTOGRAPHIC REPRESENTATIONS OF PASSWORDS

PUID: [SEC-CAT-CRF-005]
Requirement description: The application must exclusively store cryptographic representations of passwords, following best practices and using FIPS-validated cryptographic modules
Source: ✓ **V-222542:** The application must only store cryptographic representations of passwords o Use strong cryptographic hash functions when creating password hash values. o Utilize random salt values when creating the password hash. o Ensure strong access control permissions on data files containing authentication data. [13]. ✓ **V-222583:** The application must use the Federal Information Processing Standard (FIPS) 140-2-validated cryptographic modules and random number generator if the application implements encryption, key exchange, digital signature, and hash functionality. Use of weak or untested encryption algorithms undermines the purposes of utilizing encryption to protect data. The application must implement cryptographic modules adhering to the higher standards approved by the federal government since this provides assurance they have been tested and validated. [13] ✓ **V-222571:** The application must utilize FIPS-validated cryptographic modules when generating cryptographic hashes. Configure the application to use a FIPS-validated hashing algorithm when creating a cryptographic hash. Use of weak or untested encryption algorithms undermines the purposes of utilizing encryption to protect data. The application must implement cryptographic modules adhering to the higher standards approved by the federal government since this provides assurance they have been tested and validated. [13]
Priority: Not described
Rationale: This requirement seeks to strengthen the security of passwords stored in the application, mitigating risks associated with unsecured storage and ensuring the use of FIPS-validated cryptographic standards for critical operations.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ Testing should be performed to ensure that the application meets the above implementation criteria. ✓ Stored passwords must be cryptographically represented and generated using robust hash functions with random salt values.

✓ The application must use FIPS-validated cryptographic modules for password, encryption, and hashing operations.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: When I reviewed the application database, it was evident that passwords, as well as the documents of the medical file of a patient, are FIPS-Validated Cryptographic Modules. **Result:** It complies

2.9.3.3 Injection

INPUT DATA VALIDATION

PUID: [SEC-CAT-INJ-001]
Requirement description: The application must implement data entry validation measures before storing them to guarantee the integrity and security of the information processed, stored, or accessible through end-user point devices.
Source: ✓ **8.1 User endpoint devices** Information stored on, processed by or accessible via user endpoint devices should be protected. [7] ✓ **Security Requirement 76** – Validating Input Data: The EHRi and all PoS systems connected to the EHRi must include, wherever feasible, measures to safeguard against user error by validating data input to ensure that it is correct and appropriate. The following controls should be considered: a) Input checks to detect the following errors: i. out-of-range values; ii. invalid characters in data fields; iii. missing or incomplete data; iv. exceeding upper and lower data volume limits; v. unauthorized or inconsistent control data. b) Procedures for responding to validation errors. [12] ✓ **A02:2021 – Cryptographic Failures:** Use positive server-side input validation. This is not a complete defense as many applications require special characters, such as text areas or APIs for mobile applications. [3] ✓ **SI-10 INFORMATION INPUT VALIDATION** Control: Check the validity of the following information inputs: [Assignment: organization-defined information inputs to the system].

Discussion: Checking the valid syntax and semantics of system inputs—including character set, length, numerical range, and acceptable values—verifies that inputs match specified definitions for format and content. For example, if the organization specifies that numerical values between 1-100 are the only acceptable inputs for a field in a given application, inputs of "387," "abc," or "%K%" are invalid inputs and are not accepted as input to the system. Valid inputs are likely to vary from field to field within a software application. Applications typically follow well-defined protocols that use structured messages (i.e., commands or queries) to communicate between software modules or system components. Structured messages can contain raw or unstructured data interspersed with metadata or control information. If software applications use attacker-supplied inputs to construct structured messages without properly encoding such messages, then the attacker could insert malicious commands or special characters that can cause the data to be interpreted as control information or metadata. Consequently, the module or component that receives the corrupted output will perform the wrong operations or otherwise interpret the data incorrectly. Prescreening inputs prior to passing them to interpreters prevents the content from being unintentionally interpreted as commands. Input validation ensures accurate and correct inputs and prevents attacks such as cross-site scripting and a variety of injection attacks. [11].

✓ **V-222609:** The application must not be subject to input handling vulnerabilities.

A common application vulnerability is unpredictable behavior due to improper input validation. This requirement guards against adverse or unintended system behavior caused by invalid inputs, where information system responses to the invalid input may be disruptive or cause the system to fail into an unsafe state.

Data received from the user should always be suspected as being malicious and always validated prior to using it as input to the application.

Some examples of input methods:

- Forms Data
- URL parameters
- Hidden Fields
- Cookies
- HTTP Headers or anything in the HTTP request
- Client data entry fields

Items to validate:

- Out of range values/Boundary
- Data length
- Validate types of characters allowed
- Whitelist validation for known good data input while denying all other input.

Other recommendations include:

- Using drop-down menus for lists
- Validating input on the server, not on the client.

If validating on the client, also validate on the server:

- Using regular expressions to validate input
- Using HTML filter libraries that implement input validation tasks. [13]

✓ **V-222606:** The application must validate all input. Design and configure the application to validate input prior to executing commands. [13]

✓ **CWE-20 Improper Input Validation**

The product receives input or data, but it does not validate or incorrectly validates that the input has the properties that are required to process the data safely and correctly. [14]

Priority: Not described

Rationale: This requirement intends to guarantee the integrity and security of the information in the application by implementing effective input data validation before processing data.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To_be_determined

Verification method: Demonstration/Analysis

Validation criteria:

✓ **The application must include input controls to detect and prevent:**

1. Values out of range.
2. Invalid characters in data fields.
3. Missing or incomplete data.
4. Exceeding upper and lower data volume limits.
5. Unauthorized or inconsistent control data.

✓ **Validation Error Response Procedures:**

1. The application must have defined procedures to respond to validation errors, ensuring proper management of situations where data entry does not meet specified criteria.

✓ **Positive Server-Side Validation:**

1. Positive server-side validation should be used for data entry, avoiding vulnerabilities associated with malicious entries.

✓ **Preventing Input Tampering Vulnerabilities:**

1. The application should not be subject to vulnerabilities in input handling. This includes preventing unpredictable or unwanted behavior due to inadequate input validation.

✓ **Exhaustive Validation of All Types of Entries:**

1. The application must validate all types of input received from users, considering elements such as data forms, URL parameters, hidden fields, cookies, HTTP headers, and other input methods.

✓ **Validation of Specific Elements:**

- ✓ Validation should address out-of-range values, data length, and allowed character types, and whitelisting should be applied for known and safe data.

Requested by: The organization

Responsible: Developer

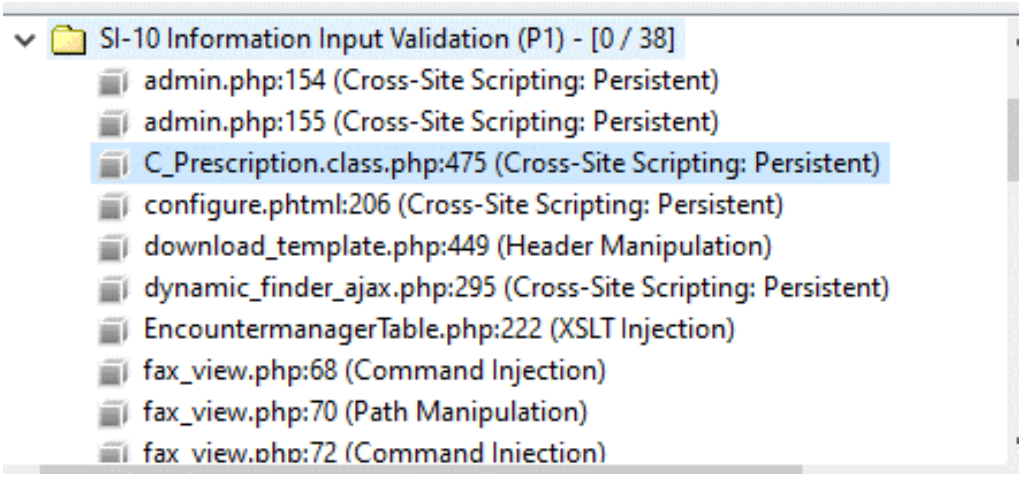
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are findings for XSS, Command injection, and JSON Injection.



Result: It doesn't comply

PUID: [SEC-CAT-INJ-002]

Requirement description: The application must be secure against SQL injection attacks. It is necessary to avoid possible security problems related to SQL injection and adjust the application to eliminate any risk associated with this attack.

Source:

- ✓ **V-222600**: The application must not be vulnerable to SQL Injection. Modify the application and remove SQL injection vulnerabilities.

  SQL Injection is a code injection attack against database applications. Malicious SQL statements are inserted into an application data entry field where they are submitted to the database and executed. This is a direct result of not validating input that is used by the application to perform a command or execute an action.

Successful attacks can read data, write data, execute administrative functions within the database, shutdown the DBMS, and in some cases execute OS commands.

Best practices to reduce the potential for SQL Injection vulnerabilities include:

- Not using concatenation or replacement to build SQL queries.
- Using prepared statements with parameterized queries that have been tested and validated not to be vulnerable to SQL Injection.
- Using stored procedures that have been tested and validated not to be vulnerable to SQL Injection.
- Escaping all user supplied input.

Additional steps to prevent SQL Injection can be found at the OWASP website:

https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet [13[13]

✓ **A02:2021 – Cryptographic Failures**:

- For any residual dynamic queries, escape special characters using the specific escape syntax for that interpreter.

  Note: SQL structures such as table names, column names, and so on cannot be escaped, and thus user-supplied structure names are dangerous. This is a common issue in report-writing software.

- Use LIMIT and other SQL controls within queries to prevent mass disclosure of records in case of SQL injection. [3]

✓ **CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')**

The product constructs all or part of an SQL command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended SQL command when it is sent to a downstream component. It is related to SQL Injection. [14]

Priority: Hight

Rationale: This requirement ensures the application is fully protected against potential SQL injection threats, thereby mitigating risks associated with malicious manipulation of SQL commands.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

✓ **Static Code Analysis:**

1. Perform a static source code analysis in search of patterns susceptible to SQL injection.
2. Identify instances of value concatenation or substitution in SQL queries and ensure safe methods are used.

- ✓ **Penetration tests:**

  1. Perform specific penetration tests to identify possible SQL injection points.
  2. Verify the implementation of prepared queries and ensure that user input is properly escaped.

- ✓ **Validation of Development Practices:**

  1. Review the use of stored procedures and confirm that they have been tested and validated against SQL injection.
  2. Evaluate the use of throttling and other SQL controls to prevent mass disclosure of records in the event of SQL injection.

- ✓ **Success Criteria:**

  1. No SQL injection vulnerabilities should be found during the evaluation.
  2. The application should implement reasonable security practices, such as prepared queries and input validation.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are no findings for Sql Injection.

Result: It complies

PUID: [SEC-CAT-INJ-003]

Requirement description: The application must implement effective restrictions on the types of files that users can upload to prevent the upload of dangerous files that may be automatically processed in the application environment.

Source:

- ✓ **CWE-434: Unrestricted Upload of File with Dangerous Type**

  The product allows the attacker to upload or transfer files of dangerous types that can be automatically processed within the product's environment. The phrase could be interpreted as the lack of restrictions on the size or number of uploaded files, which is a resource consumption issue. This weakness is caused by missing a security tactic during the architecture and design phase. [14]

Priority: Hight

Rationale: This requirement addresses the vulnerability identified as CWE-434, which poses the threat that an attacker could upload or transfer files of potentially dangerous types that could be

automatically processed within the application environment. Establishing file upload restrictions is essential to counteract this vulnerability and safeguard the integrity and security of the application.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **File Upload:**
  1. Try uploading files of known dangerous types and see if the application prevents the upload.
  2. Verify that the application generates clear and descriptive error messages when attempting to upload an illegal file.
- ✓ **Constraint Validation:**
  1. Review the file upload restrictions settings in the application.
  2. Confirm that the application follows security best practices for limiting allowed file types.
- ✓ **Stress Tests:**
  1. Perform stress tests to evaluate the application's ability to handle a normal file load without impacting performance.
- ✓ **Success Criteria:**
  1. The application must block the upload of files of dangerous types.
  2. The error messages provided should be informative and understandable.
  3. File upload restrictions settings must be implemented appropriately and aligned with security best practices.

Requested by: The organization

Responsible: Developer

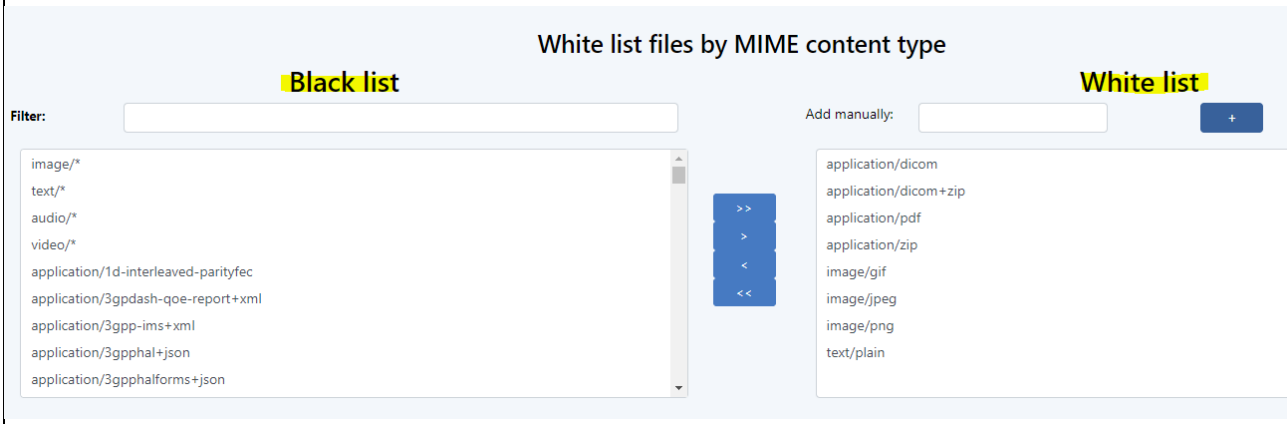
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The audit team reviewed the application, and as an administrator user, the option of managing the types of files the software can handle is presented.



Result: It complies

PUID: [SEC-CAT-INJ-004]

Requirement description: The application must implement protection measures against command injections. Depending on the application's architecture, the application must be modified to escape/sanitize special character input or configure the system to protect against command injection attacks.

Source:

- ✓ **V-222604**: The application must protect from command injection. Modify the application so as to escape/sanitize special character input or configure the system to protect against command injection attacks based on application architecture.

  A command injection attack is an attack on a vulnerable application where improperly validated input is passed to a command shell setup in the application. The result is the ability of an attacker to execute OS commands via the application.

  A command injection allows an attacker to execute their own commands with the same privileges as the application executing.

  The following is an example of a URL based command injection attack.

  Before alteration:
  http://sitename/cgi-bin/userData.pl?doc=user1.txt

  Example URL modified:
  http://sitename/cgi-bin/userData.pl?doc=/bin/ls|

  The result is the execution of the command usr/bin/ls which could allow the attacker to list contents of the directory via the browser.

  The following is a list of functions vulnerable to command injection sorted according to language.

  Language Functions/Characters
  - C/C++ - system(), popen(), execlp(), execvp(), ShellExecute(), ShellExecuteEx(), _wsystem()

- Perl - system, exec, `,open, |, eval, /e
- Python - exec, eval, os.system, os.popen, execfile, input, compile
- Java - Class.forName(), Class.newInstance(), Runtime.exec()"[13]

✓ **CWE-78 Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')**

The product constructs all or part of an OS command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended OS command when it is sent to a downstream component. It is related to command injection in OS. From a weakness standpoint, these variants represent distinct programmer errors. In the first variant, the programmer clearly intends that input from untrusted parties will be part of the arguments in the command to be executed. In the second variant, the programmer does not intend for the command to be accessible to any untrusted party, but the programmer probably has not accounted for alternate ways in which malicious attackers can provide input. [14]

✓ **CWE-77: Improper Neutralization of Special Elements used in a Command ('Command Injection')**

The product constructs all or part of a command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended command when it is sent to a downstream component. Command injection vulnerabilities typically occur when:

1. Data enters the application from an untrusted source.
2. The data is part of a string that is executed as a command by the application.
3. By executing the command, the application gives an attacker a privilege or capability that the attacker would not otherwise have.

Many protocols and products have their own custom command language. While OS or shell command strings are frequently discovered and targeted, developers may not realize that these other command languages might also be vulnerable to attacks. Command injection is a common problem with wrapper programs. [14]

Priority: Not described

Rationale: This requirement seeks to mitigate the risks associated with command injection attacks, which can compromise application security by allowing an attacker to execute operating system commands through the application. By escaping or sanitizing special character input and properly configuring the system, you reduce the chances of attackers executing unauthorized commands and strengthen the application's overall security.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

✓ **Exhaust/Entry Sanitization:**

1. Confirm that the application effectively escapes or sanitizes special characters in user input.

2. Please verify that the application modifies the input so that it cannot be interpreted as operating system commands.

✓ **System configuration:**

1. Ensure the system is configured to protect against command injections based on the application architecture.
2. Verify that controls have been implemented to validate and filter inputs susceptible to command injections.

✓ **Command Injection Prevention:**

1. Confirm that functions and characters prone to command injection, based on the list provided, have been neutralized or handled appropriately.
2. Verify that the measures implemented follow the best security practices to prevent command injection.

✓ **Success Criteria:**

1. The application must successfully pass all stages of the validation procedure.
2. No instances should be found where entering special characters leads to the unauthorized execution of operating system commands.
3. The measures implemented must comply with the standards established in V-222604, CWE-78, and CWE-77.

Requested by: The organization

Responsible: Developer

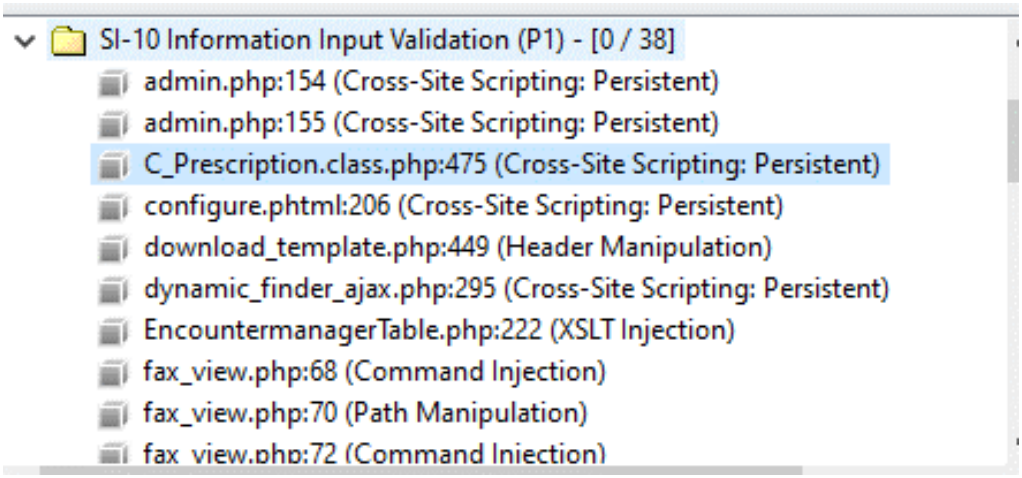
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are findings for XSS, **Command injection**, and JSON Injection.



Result: It doesn't comply

PUID: [SEC-CAT-INJ-005]

Requirement description: The application must implement protection measures against Cross-Site Scripting (XSS) vulnerabilities to prevent malicious code injection attacks

Source:

- ✓ **V-222602**: The application must protect from Cross-Site Scripting (XSS) vulnerabilities.

  "XSS attacks are essentially code injection attacks against the various language interpreters contained within the browser. XSS can be executed via HTML, JavaScript, VBScript, ActiveX; essentially any scripting language a browser is capable of processing.

  XSS vulnerabilities are created when a website does not properly sanitize, escape, or encode user input. For example, ""<"" is the HTML encoding for the ""<"" character. If the encoding is performed, the script code will not execute.

  There are 3 parties involved in an XSS attack, the attacker, the trusted and vulnerable website, and the victim. An attacker will take advantage of a vulnerable website that does not properly validate user input by inserting malicious code into any data entry field.

  When the victim visits the trusted website and clicks on the malicious link left by the attacker, the attackers script is executed in the victims browser with the trust permissions assigned to the site.

  There are several different types of XSS attack and the complete details regarding XSS cannot be described completely here.

  To address the issue of XSS, web application developers must escape, encode or otherwise validate all user input that is processed and output by the web server. They should also use web templates or a web development framework that provides the capability to encode or otherwise validate user input.

  Examples of XSS vulnerabilities can be obtained from the Open Web Application Security Project (OWASP) website.

The site is available by pointing your browser to https://www.owasp.org.

"Verify user input is validated and encode or escape user input to prevent embedded script code from executing.

Develop your application using a web template system or a web application development framework that provides auto escaping features rather than building your own escape logic." [13]

✓ **CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')**

The product does not neutralize or incorrectly neutralizes user-controllable input before it is placed in output that is used as a web page that is served to other users. It is related with XSS.

Cross-site scripting (XSS) vulnerabilities occur when:

1. Untrusted data enters a web application, typically from a web request.
2. The web application dynamically generates a web page that contains this untrusted data.
3. During page generation, the application does not prevent the data from containing content that is executable by a web browser, such as JavaScript, HTML tags, HTML attributes, mouse events, Flash, ActiveX, etc.
4. A victim visits the generated web page through a web browser, which contains malicious script that was injected using the untrusted data.
5. Since the script comes from a web page that was sent by the web server, the victim's web browser executes the malicious script in the context of the web server's domain.
6. This effectively violates the intention of the web browser's same-origin policy, which states that scripts in one domain should not be able to access resources or run code in a different domain.

There are three main kinds of XSS:

- **Type 1: Reflected XSS** (or Non-Persistent) - The server reads data directly from the HTTP request and reflects it back in the HTTP response. Reflected XSS exploits occur when an attacker causes a victim to supply dangerous content to a vulnerable web application, which is then reflected back to the victim and executed by the web browser. The most common mechanism for delivering malicious content is to include it as a parameter in a URL that is posted publicly or e-mailed directly to the victim. URLs constructed in this manner constitute the core of many phishing schemes, whereby an attacker convinces a victim to visit a URL that refers to a vulnerable site. After the site reflects the attacker's content back to the victim, the content is executed by the victim's browser.

- **Type 2: Stored XSS** (or Persistent) - The application stores dangerous data in a database, message forum, visitor log, or other trusted data store. At a later time, the dangerous data is subsequently read back into the application and included in dynamic content. From an attacker's perspective, the optimal place to inject malicious content is in an area that is displayed to either many users or particularly interesting users. Interesting users typically have elevated privileges in the application or interact with sensitive data that is valuable to the attacker. If one of these users executes malicious content, the attacker may be able to perform

privileged operations on behalf of the user or gain access to sensitive data belonging to the user. For example, the attacker might inject XSS into a log message, which might not be handled properly when an administrator views the logs.

- **Type 0: DOM-Based XSS** - In DOM-based XSS, the client performs the injection of XSS into the page; in the other types, the server performs the injection. DOM-based XSS generally involves server-controlled, trusted script that is sent to the client, such as Javascript that performs sanity checks on a form before the user submits it. If the server-supplied script processes user-supplied data and then injects it back into the web page (such as with dynamic HTML), then DOM-based XSS is possible. [14]

✓ **CWE-94: Improper Control of Generation of Code ('Code Injection')**

The product constructs all or part of a code segment using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the syntax or behavior of the intended code segment. [14]

Priority: Not described

Rationale: This requirement is intended to mitigate the threat of malicious code injection into the application's user interface caused by the vulnerability known as Cross-Site Scripting (XSS). This is achieved by implementing security measures that appropriately validate, escape, or encrypt user input, thereby ensuring data integrity and security.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

✓ **Input Validation Check:**

1. Verify that the application effectively validates user input to prevent the insertion of malicious content.
2. Confirm that appropriate encoding or escaping techniques are used for the output of data processed by the server.

✓ **Use of Secure Development Tools:**

1. Verify that the application is developed using a web templating system or a web application development framework that provides auto-escaping capabilities.
2. Ensure that custom escape logic is not implemented but functions provided by the framework or template are used.

✓ **Code Review for Known Vulnerabilities:**

1. Perform a code review for possible known XSS vulnerabilities, following the security guidelines the Open Web Application Security Project (OWASP) provided.
2. Ensure that any identified vulnerabilities are addressed and corrected during development.

✓ **Preventing Persistent and Non-Persistent XSS:**

1. Verify that the application prevents both reflected (non-persistent) XSS and stored (persistent) XSS.
2. Confirm that appropriate controls are implemented to sanitize and validate user input in all application areas.

Requested by: The organization

Responsible: Developer

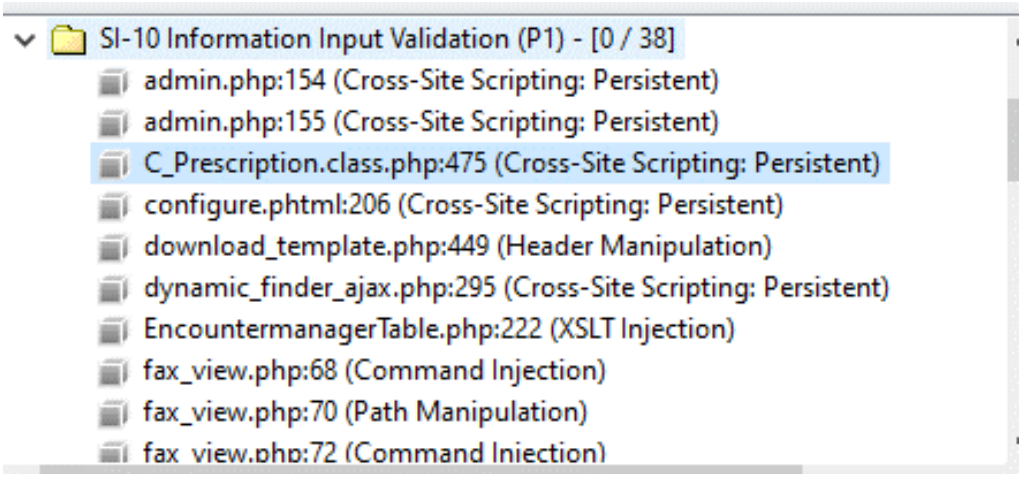
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are findings for XSS, **Command injection**, and JSON Injection.



Result: It doesn't comply

OUTPUT DATA VALIDATION

PUID: [SEC-CAT-INJ-006]

Requirement description: The application must implement an information output validation mechanism for the following software programs and/or applications: [reporting, data tables listing, patient medical records].

Source:

- ✓ SI-15 INFORMATION OUTPUT FILTERING

  Control: Validate information output from the following software programs and/or applications to ensure that the information is consistent with the expected content: [Assignment: organization-defined software programs and/or applications].

  Discussion: Certain types of attacks, including SQL injections, produce output results that are unexpected or inconsistent with the output results that would be expected from software

programs or applications. Information output filtering focuses on detecting extraneous content, preventing such extraneous content from being displayed, and then alerting monitoring tools that anomalous behavior has been discovered. [11].
Priority: Not described
Rationale: This requirement is intended to ensure that the information generated by the application is consistent with the expected content. Implementing an output filter helps prevent attacks, such as SQL injections, that can produce unexpected or inconsistent results with the desired output.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Filter Configuration:** **1.** Verify that the application has output filters configured for software programs and/or applications defined by the organization. ✓ **Consistence of Information:** **1.** Perform information output tests from the defined programs and/or applications. **2.** Confirm that the information displayed is consistent with the expected content. ✓ **Anomaly Detection:** **1.** Evaluate the ability of filters to detect unwanted or unexpected content. **2.** Verify that alerts are generated to the monitoring tools if anomalous behavior is detected.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: A DAST and SAST have been performed, and there are no findings for Output Data Validation.

Result: It complies

2.9.3.4 Insecure Design

ENFORCE DATA FLOW CONTROL

PUID: [SEC-CAT-IND-001]

Requirement description: The system must ensure that data is only processed following the directives provided in the data flow control.

Source:

- ✓ Make sure that data are only processed according to given directives [16]
- ✓ **V-222427:** The application must enforce approved authorizations for controlling the flow of information within the system based on organization-defined information flow control policies. Configure the application to enforce data flow control in accordance with data flow control policies. [13]
- ✓ **V-222428:** The application must enforce approved authorizations for controlling the flow of information between interconnected systems based on organization-defined information flow control policies. Configure the application to enforce data flow control in accordance with data flow control policies. [13]
- ✓ **CWE-863: Incorrect Authorization**
  The product performs an authorization check when an actor attempts to access a resource or perform an action, but it does not correctly perform the check. This allows attackers to bypass intended access restrictions. [14]
- ✓ **CWE-276: Incorrect Default Permissions**
  The product performs an authorization check when an actor attempts to access a resource or perform an action, but it does not correctly perform the check. This allows attackers to bypass intended access restrictions. [14]

Priority: Not described

Rationale: This requirement is established to ensure data manipulation is carried out precisely and safely, avoiding possible safety violations and ensuring the application meets organizational policies and information flow control standards.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Internal information flow control:** 1. Verify that the application applies authorizations approved to control the flow of information within the system under the data flow control policies defined by the organization (V-222427). 2. Confirm that the application configuration complies with specified internal data flow policies. ✓ **External information flow control:** 1. Ensure that the application applies authorizations approved to control the flow of information between interconnected systems according to data flow control policies defined by the organization (V-222428). 2. Confirm that the application configuration meets the specified external data flow control policies. ✓ **Incorrect authorization prevention (CWE-863):** 1. Perform tests to verify that the application correctly performs authorization checks when trying to access a resource or perform an action, avoiding possible omissions that allow attackers to bypass access restrictions. ✓ **Predetermine prevention (CWE-276):** 1. Verify that the application performs correct authorization checks when an actor tries to access a resource or perform an action, avoiding configurations that allow attackers to bypass access restrictions using incorrect predetermined permits.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: **Result:** N/a

PUID: [SEC-CAT-IND-002]
Requirement description: The system must implement measures to prevent unauthorized and involuntary transfer of information through shared system resources
Source:

✓ **V-222592:** Applications must prevent unauthorized and unintended information transfer via shared system resources. Â Configure or design the application to utilize a security control that will implement a boundary that will prevent unauthorized and unintended information transfer via shared system resources If the application does not prevent unauthorized and unintended information transfer via shared system resources, this is a finding. [13]
Priority: Not described
Rationale: The system must implement measures to prevent the unauthorized and involuntary transfer of information through shared system resources. This requirement promises the safety and confidentiality of information by avoiding unauthorized and unintended transfer through shared system resources, thus reducing the risk of sensitive data exposure.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Safety control configuration or design:** 1. Verify that the application is configured or designed to use a security control that establishes limits and prevents unauthorized and involuntary information transfer through shared system resources. ✓ **Effective prevention:** 1. Confirm that the application effectively implements the necessary measures to prevent unauthorized and involuntary information transfer. 2. If it is identified that the application does not meet this security measure, consider it as a finding.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The application does not implement shared sources; therefore, it is considered not applicable for the audit (N/a). **Result:** N/a

PRESENTING INFORMATION TO THE USER

PUID: [SEC-CAT-IND-003]

Requirement description: The application must show a system of use of the system and privacy notification before granting access. This notice must contain information about the property of the system, the monitoring, recording, and auditing of use, the penalties for unauthorized use, and the acceptance of these conditions by the user.

Source:

- ✓ **8.2.2 Labelling of Information**: All health information systems processing personal health information should inform users of the confidentiality of personal health information accessible from the system (e.g., at start-up or log-in) and should label hardcopy output as confidential when it contains personal health information. [8]
- ✓ **9.4.2 Secure** log-on procedures: b) display a general notice warning that the computer should only be accessed by authorized users; [8].
- ✓ **AC-8 SYSTEM USE NOTIFICATION**
  Control:
  a. Display [Assignment: organization-defined system use notification message or banner] to users before granting access to the system that provides privacy and security notices consistent with applicable laws, executive orders, directives, regulations, policies, standards, and guidelines and state that:
  1. Users are accessing a U.S. Government system;
  2. System usage may be monitored, recorded, and subject to audit;
  3. Unauthorized use of the system is prohibited and subject to criminal and civil penalties; and
  4. Use of the system indicates consent to monitoring and recording;
  b. Retain the notification message or banner on the screen until users acknowledge the usage conditions and take explicit actions to log on to or further access the system; and
  c. For publicly accessible systems:
  1. Display system use information [Assignment: organization-defined conditions], before granting further access to the publicly accessible system;
  2. Display references, if any, to monitoring, recording, or auditing that are consistent with privacy accommodations for such systems that generally prohibit those activities; and
  3. Include a description of the authorized uses of the system.

- ✓ PT-5 PRIVACY NOTICE
  Control: Provide notice to individuals about the processing of personally identifiable information that:
  a. Is available to individuals upon first interacting with an organization, and subsequently at [Assignment: organization-defined frequency];
  b. Is clear and easy-to-understand, expressing information about personally identifiable information processing in plain language;
  c. Identifies the authority that authorizes the processing of personally identifiable information;
  d. Identifies the purposes for which personally identifiable information is to be processed; and
  e. Includes [Assignment: organization-defined information]. [11]

- ✓ **V-222434:** The application must display the Standard Mandatory DoD Notice and Consent Banner before granting access to the application. Configure the application to present the standard DoD-approved banner prior to granting access to the application. [13].

- ✓ **V-222436:** The publicly accessible application must display the Standard Mandatory DoD Notice and Consent Banner before granting access to the application. Configure the

application to present the standard DoD-approved banner prior to granting access to the application. [13]

- ✓ **V-222435:** The application must retain the Standard Mandatory DoD Notice and Consent Banner on the screen until users acknowledge the usage conditions and take explicit actions to log on for further access. Configure the application to retain the standard DoD-approved banner until the user accepts the usage conditions prior to granting access to the application. [13]

Priority: Not described

Rationale: This requirement seeks to adequately inform users about the confidential nature of personal health information processed by the system. This is essential to comply with regulations and guarantee data privacy and safety.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

1. Verify that a system is used and privacy notification is presented according to regulations before allowing access.
2. The notice must remain on the screen until users recognize the conditions of use and perform explicit actions to log in and access the system.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

By authenticating the system, no message is presented that the system is monitoring and auditing its use or the penalties associated with unauthorized use.

Result: It doesn't comply

PUID: [SEC-CAT-IND-004]

Requirement description: The system must generate error messages that provide the user with the information necessary for corrective actions without revealing details that can be exploited.

Source:

- ✓ **SI-11 ERROR HANDLING** (log)
  Control:

a. Generate error messages that provide information necessary for corrective actions without revealing information that could be exploited; and
b. Reveal error messages only to [Assignment: organization-defined personnel or roles].

Discussion: Organizations consider the structure and content of error messages. The extent to which systems can handle error conditions is guided and informed by organizational policy and operational requirements. Exploitable information includes stack traces and implementation details; erroneous logon attempts with passwords mistakenly entered as the username; mission or business information that can be derived from, if not stated explicitly by, the information recorded; and personally identifiable information, such as account numbers, social security numbers, and credit card numbers. Error messages may also provide a covert channel for transmitting information. [11].

- ✓ **V-222610:** The application must generate error messages that provide information necessary for corrective actions without revealing information that could be exploited by adversaries.

  Configure the server to not send error messages containing system information or sensitive data to users.

  Use generic error messages [13].

Priority: Not described

Rationale: This requirement is established to ensure that, in error situations, the system provides the essential information to correct problems without disseminating details that adversaries could use. The restriction in the revelation of error messages helps prevent the exposure of sensitive information that could be used for malicious activities.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Generation of error messages:**

  1. Verify that the system generates error messages that provide essential information for corrective actions.
  2. Confirm that error messages do not reveal information that can be exploited by adversaries.

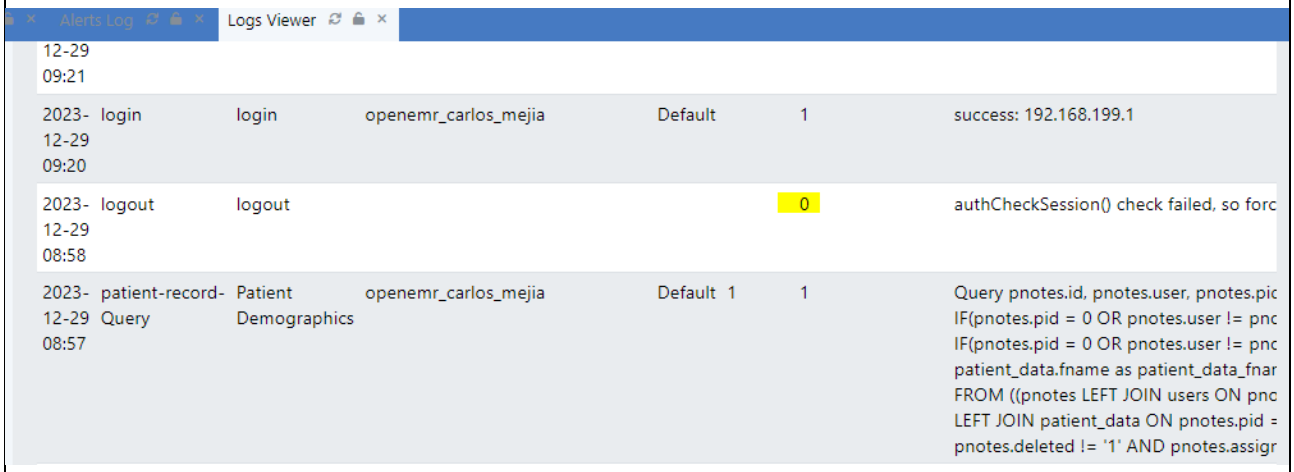
- ✓ **Selective Revelation:**

  1. Evaluate that error messages are only revealed to specific personnel or roles defined by the organization.

- ✓ **Server configuration:**

  1. Verify that the server is configured not to send end users error messages containing system information or sensitive data.

- ✓ **Use of generic messages:**

1. Confirm that generic error messages are used to limit the dissemination of sensitive information.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: After reviewing the system, the log option shows error messages that provide the user with the information necessary for corrective actions without revealing details that can be exploited.  Alerts Log Logs Viewer 12-29 09:21 2023-12-29 09:20 login login openemr_carlos_mejia Default 1 success: 192.168.199.1 2023-12-29 08:58 logout logout 0 authCheckSession() check failed, so forc 2023-12-29 08:57 patient-record-Query Patient Demographics openemr_carlos_mejia Default 1 1 Query pnotes.id, pnotes.user, pnotes.pic IF(pnotes.pid = 0 OR pnotes.user != pnc IF(pnotes.pid = 0 OR pnotes.user != pnc patient_data.fname as patient_data_fnar FROM ((pnotes LEFT JOIN users ON pnc LEFT JOIN patient_data ON pnotes.pid = pnotes.deleted != '1' AND pnotes.assigr **Result:** It complies

PUID: [SEC-CAT-IND-005]
Requirement description: The application must physically or logically separate the functionality of the user, including user interface services and system management functionality.
Source: ✓ **SC-2** SEPARATION OF SYSTEM AND USER FUNCTIONALITY **Control:** Separate user functionality, including user interface services, from system management functionality. **Discussion:** System management functionality includes functions that are necessary to administer databases, network components, workstations, or servers. These functions typically require privileged user access. The separation of user functions from system management functions is physical or logical. Organizations may separate system management functions from user functions by using different computers, instances of operating systems, central processing units, or network addresses; by employing virtualization techniques; or some combination of these or other methods. Separation of system management functions from user functions includes web administrative interfaces that employ separate authentication methods for users of any other system resources.

Separation of system and user functions may include isolating administrative interfaces on different domains and with additional access controls. The separation of system and user functionality can be achieved by applying the systems security engineering design principles in SA-8, including SA-8(1), SA-8(3), SA-8(4), SA-8(10), SA-8(12), SA-8(13), SA-8(14), and SA-8(18). [11]

- ✓ **V-222574:** The application user interface must be either physically or logically separated from data storage and management interfaces. Configure the application so user interface to the application and management interface to the application is separated.

  Application management functionality includes functions necessary for administration and requires privileged user access. Allowing non-privileged users to access application management functionality capabilities increases the risk that non-privileged users may obtain elevated privileges.

  The separation of user functionality from information system management functionality is either physical or logical and is accomplished by using different computers, different central processing units, different instances of the operating system, different network addresses, different TCP/UDP ports, virtualization techniques, combinations of these methods, or other methods, as appropriate.

  An example of this type of separation is observed in web administrative interfaces that use separate authentication methods for users of any other information system resources. This may include isolating the administrative interface on a different security domain and with additional access controls. [13]

- ✓ **V-222590:** The application must isolate security functions from non-security functions. Implement controls within the application that limits access to security configuration functionality and isolates regular application function from security-oriented function.

- ✓ **A04:2021 – Insecure Design:**

  - Establish and use a secure development lifecycle with AppSec professionals to help evaluate and design security and privacy-related controls
  - Segregate tier layers on the system and network layers depending on the exposure and protection needs
  - Segregate tenants robustly by design throughout all tiers [3]

- ✓ **A05:2021 – Security Misconfiguration:**
  - A repeatable hardening process makes it fast and easy to deploy another environment that is appropriately locked down. Development, QA, and production environments should all be configured identically, with different credentials used in each environment. This process should be automated to minimize the effort required to set up a new secure environment.
  - A minimal platform without any unnecessary features, components, documentation, and samples. Remove or do not install unused features and frameworks.
  - A segmented application architecture provides effective and secure separation between components or tenants, with segmentation, containerization, or cloud security groups (ACLs).
  - An automated process to verify the effectiveness of the configurations and settings in all environments. [3]

Priority: Not described

Rationale: Separating user functions from system management functions is essential to mitigate risks and prevent possible vulnerabilities. This measure guarantees that system administration functions, which require privileged access, are isolated from user functions, thus reducing the risk of unauthorized access and obtaining high privileges.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Verification of separation in the user interface:**

  1. Confirm that the application user interface is physically or logically separated from the storage and data management interfaces.

- ✓ **EVALUATION OF THE SEPARATION OF MANAGEMENT FUNCTIONS:**

  1. Confirm that application management functions require privileged access and are physically or logically separated from the user's regular functions.

- ✓ **Isolation of security functions:**

  1. Validate that the application implements controls to limit access to safety configuration functionality and isolate regular functions of the application of security-oriented functions.

- ✓ **Verification of good design practices:**

  1. Verify that a safe development life cycle is established and used with APSEC professionals to evaluate and design security and privacy controls.
  2. Confirm that the level layers in the system and network layers are segregated according to exposure and protection needs.
  3. Validate that the application's architecture is designed to segregate tenants robustly at all levels.

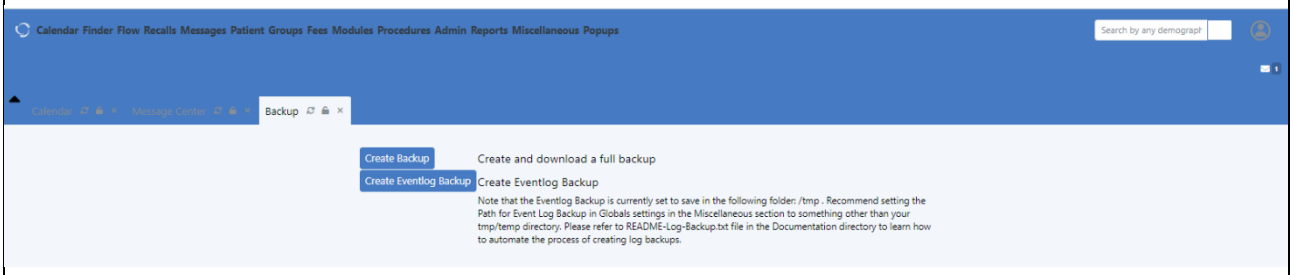
- ✓ **Repeatable hardening process:**

  1. Confirm that a repeatable hardening process facilitates another duly protected environment's rapid and simple implementation.
  2. Validate that development, QA, and production environments are configured identically, with different credentials used in each environment.
  3. Confirm that the platform is minimal, without characteristics, components, documentation, and unnecessary examples.

- ✓ **Automated settings and settings:**

  1. Confirm that the application implements an automated process to verify the effectiveness of configurations and settings in all environments.
  2. Validate that an adequate verification of the configurations and adjustments in development, QA, and production environments are carried out.

Requested by: The organization

Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: After reviewing the system, the functionality and admin functions of the application are in the same combined interface.      **Result:** It doesn't comply.

PUID: [SEC-CAT-IND-006]
Requirement description: The application should be designed not to use emerging or non-categorized mobile code.
Source: ✓ **V-222665:** The designer must ensure uncategorized or emerging mobile code is not used in applications. By definition, mobile code is software obtained from remote systems outside the enclave boundary, transferred across a network, and then downloaded and executed on a local system without explicit installation or execution by the recipient. For a complete list of mobile code categorizations, refer to the overview document included with this STIG. Categorized mobile code includes but is not limited to: o ActiveX o Windows Scripting Host when used as mobile code o Unix Shell Scripts when used as mobile code o DOS batch scripts when used as mobile code o Java applets and other Java mobile code o Visual Basic for Applications (VBA) o LotusScript o PerfectScript o Postscript o JavaScript (including Jscript and ECMAScript variants) o VBScript

- Portable Document Format (PDF)
- Shockwave/Flash
- Rich Internet Applications

The following technologies are not currently designated as mobile code:

- XML
- SMIL
- QuickTime
- VRML (exclusive of any associated Java applets or JavaScript scripts)

The following are outside the scope of the mobile code requirements:

- Scripts and applets embedded in or linked to web pages and executed in the context of the web server. Examples of this are Java servlets, Java Server pages, CGI, Active Server Pages, CFML, PHP, SSI, server-side JavaScript, server-side LotusScript.
- Local programs and command scripts
- Distributed object-oriented programming systems (e.g., CORBA, DCOM).
- Software patches, updates, including self-extracting updates - software updates that must be invoked explicitly by the user are outside the mobile code policy. Examples of technologies in this area include: Netscape SmartUpdate, Microsoft Windows Update, Netscape web browser plug-ins and Linux.

If other types of mobile code technologies are present that are not listed here, a written waiver must be granted by the CIO (allowing use of emerging mobile code technology). Also uncategorized mobile code must be submitted for AO approval.
Remove uncategorized or emerging mobile code from the application or obtain a waiver and risk acceptance to operate. [13].

Priority: Not described

Rationale: The Mobile Code, by definition, is software obtained from remote systems outside the enclave limit, transferred through a network, and then discharged and executed in a local system without explicit installation or execution by the recipient. Using non-categorized or emerging mobile code can introduce significant safety risks since this type of code may not be adequately classified or evaluated in terms of security.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Identification of non-categorized or emerging mobile code:**

  1. Verify that the application does not use non -categorized or emerging mobile code technologies according to the list of categorizations provided in the STIG.

- ✓ **Written exception or exception:**

  1. Confirm that a written exemption from the CIO has been obtained in case of non-categorized or emerging mobile code, with acceptance of the risk to operate.

✓ **AO approval process for non-categorized code:** **1.** Validate that, in the case of non-categorized mobile code, it has undergone the approval of the insurance officer (AO).
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: A DAST and SAST have been performed, and no findings exist for using emerging or non-categorized mobile code. **Result:** It complies.

PUID: [SEC-CAT-IND-007]
Requirement description: The application should not contain authentication data incorporated into its code, configuration files, scripts, HTML files, or any ASCII file
Source: ✓ **V-222642:** The application must not contain embedded authentication data. Authentication data stored in code could potentially be read and used by anonymous users to gain access to a backend database or application servers. This could lead to compromise of application data Remove embedded authentication data stored in code, configuration files, scripts, HTML file, or any ASCII files. [13]. ✓ **CWE-798: Use of Hard-coded Credentials**: The product contains hard-coded credentials, such as a password or cryptographic key, which it uses for its own inbound authentication, outbound communication to external components, or encryption of internal data [14]
Priority: Not described
Rationale: The presence of authentication data incorporated into the application code can represent a significant safety risk. These data could be read and used by anonymous users to obtain unauthorized access to databases or application servers, compromising the integrity and confidentiality of the application data.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To_be_determined
Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Absence of authentication data in the code:**
  1. Verify that there is no authentication data incorporated into the application source code.
- ✓ **Reviewed reviews and files:**
  1. Confirm that no authentication data is stored in configuration files, scripts, HTML files or any ASCII file associated with the application.
- ✓ **Implementation of best security practices:**
  1. Evaluate that best security practices have been followed to eliminate any coded or hardened authentication data in the application code.
- ✓ **CWE-798 compliance analysis:**
  1. Perform an analysis to ensure no hardened credentials, such as passwords or cryptographic keys encoded in the product. This will be carried out following the guidelines established by CWE-798.

Requested by: The organization

Responsible: Developer

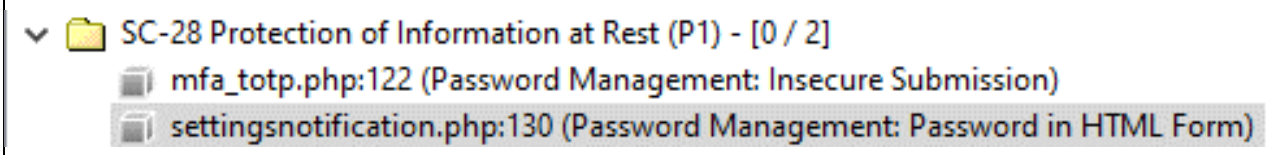
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are findings for passwords in HTM forms.



Result: It doesn't comply.

PUID: [SEC-CAT-IND-008]

Requirement description: The application should not write sensitive data in the application records. The application must be designed or reconfigured to avoid including sensitive data in the records.

Source:

- ✓ **V-222444:** The application must not write sensitive data into the application logs. Design or reconfigure the application to not write sensitive data to the logs.

It is important to identify and exclude certain types of data that is written into the logs. If the logs are compromised and sensitive data is included in the logs, this could assist an attacker in furthering their attack or it could completely compromise the system.

Examples of such data include but are not limited to; Passwords, Session IDs, Application source code, encryption keys, and sensitive data such as personal health information (PHI), Personally Identifiable Information (PII), or government identifiers (e.g., SSN). [13].

Priority: Not described

Rationale: This requirement aims to guarantee the safety of sensitive data in records to prevent potential threats and safeguard the integrity and confidentiality of information. Including sensitive data, such as passwords, session identifiers, application source code, and encryption keys, in the records could be exposed to significant risks, facilitating possible attacks or compromising the entire system in case of a violation of the records.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Application record analysis:**

  1. Verify that the application does not write sensitive data in application records.

- ✓ **Sensitive data exclusion:**

  1. Confirm that specific types of sensitive data have been identified and excluded, such as passwords, session identifiers, application source code, encryption keys, personal health information (PHI), personally identifiable information (PII) or government identifiers (for example, SSN), of the application records.

- ✓ **Reconfiguration of the application:**

  1. Validate that the application has been designed or reconfigured not to include sensitive data in the records.

- ✓ **Committed record analysis:**

  1. Evaluate the resistance of the application to possible registration commitments and verify that, even if the records are compromised, they do not contain sensitive information.

- ✓ **Sensitive data identification:**

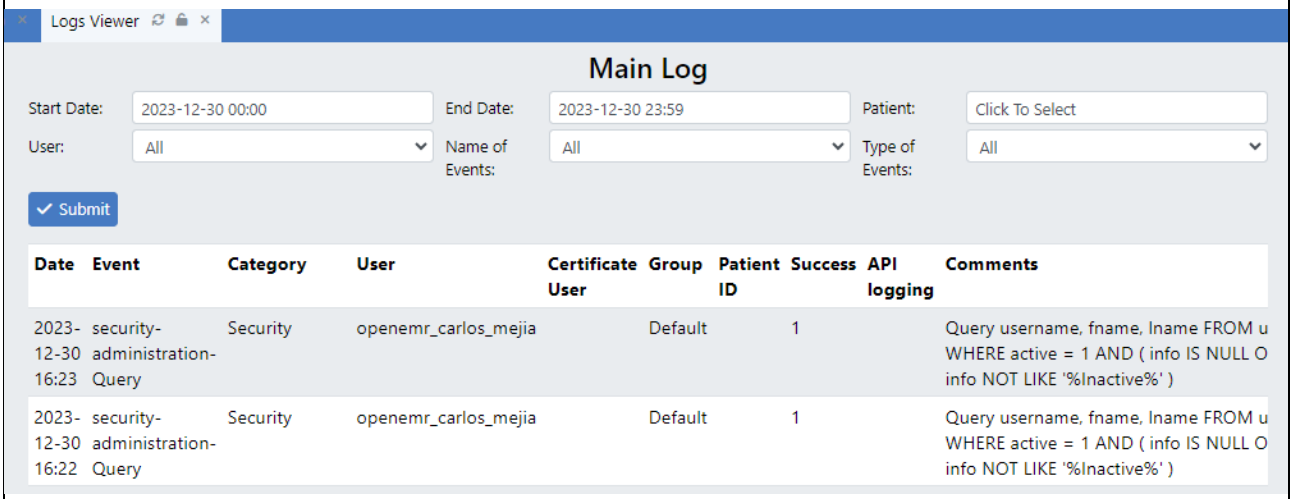
  1. Confirm that the application has effective mechanisms to identify and exclude sensitive data from records automatically.

- ✓ **Compliance audit:**

  1. Perform periodic audits to ensure continuous compliance with the application with the exclusion of sensitive data in the records.

Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023
Auditory result:

After reviewing the log records of the application, it is evident that they are not written sensitively within them.



Result: It complies.

PUID: [SEC-CAT-IND-009]

Requirement description: The application should not unnecessarily store or retain sensitive information in hidden fields. It is imperative to eliminate this information as soon as possible, or instead, use practices such as token or truncation, following the guidelines of PCI DSS to guarantee safe storage.

Source:

- ✓ **A02:2021 – Cryptographic Failures:** Don't store sensitive data unnecessarily. Discard it as soon as possible or use PCI DSS compliant tokenization or even truncation. Data that is not retained cannot be stolen. [3]

- ✓ **V-222444:** The application must not store sensitive information in hidden fields. Design and configure the application to not store sensitive information in hidden fields.

  Encrypt sensitive information stored in hidden fields using DoD-approved encryption and use server side session management techniques for user session management.

"Hidden fields allow developers to process application data without having to display it on the screen. Using hidden fields to pass data in forms is a common practice among web applications and by itself is not a security risk.

However, hidden fields are not secure and can be easily manipulated by users. Information requiring confidentiality or integrity protections must not be placed in a hidden field. If data that is sensitive must be stored in a hidden field, it must be encrypted.

Furthermore, hidden fields used to control access decisions can lead to a complete compromise of access control mechanisms allowing immediate compromise of the user's application session. [13].

Priority: Not described

Rationale: This requirement prevents unnecessary storage of sensitive information since it increases the risk of data exposure and theft. It is essential to follow security practices such as immediately discarding sensitive data and avoiding storing confidential information in hidden fields, which malicious users can easily manipulate.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To_be_determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Not necessary storage verification:**

  1. Confirm that the application discards sensitive information immediately and does not store unnecessary data.

- ✓ **Storage evaluation in hidden fields:**

  1. Validate that the application does not store sensitive information in hidden fields.
  2. Confirm that if it is necessary to store sensitive data in hidden fields, these are encrypted using encryption approved by the security standards.

- ✓ **Use of session management techniques on the server side:**

  1. Verify that the application uses session management techniques on the server side for safe user session management.

- ✓ **Compliance with PCI DSS:**

  1. Confirm that the application follows the practices of tokenization or truncation following the PCI DSS requirements for safe and sensitive information storage.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná

13/12/2023
Auditory result: A DAST and SAST have been performed, and no findings exist for storing or retaining sensitive information in hidden fields. **Result:** It complies.

PUID: [SEC-CAT-IND-010]
Requirement description: The application must ensure that the configuration and control files are not stored in the same directory as the user data. The application user data must be located in a directory different from the application code, and user file permits must be established to restrict user access to application configuration.
Source: ✓ **V-222626:** The designer must ensure the application does not store configuration and control files in the same directory as user data. Separate the application user data into a different directory than the application code and user file permissions to restrict user access to application configuration settings. Application configuration settings and user data are required to be stored in separate locations in order to prevent application users from possibly being able to access application configuration settings or application data files. [13].
Priority: Not described
Rationale: This requirement aims to separate the application configuration and user data, which is an essential action to mitigate computer security risks. When storing the configuration and user data in separate locations, the possibility that the application users inadvertently access the application configuration or application data files is reduced, which could compromise the safety and integrity of the application.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Separate storage verification:** **1.** Confirm that the application stores configuration and control files in a directory different from user data. ✓ **Evaluation of user data location:** **1.** Validate that the application user data is in a separate directory of the application code. ✓ **File permissions:** **1.** Confirm that user file permits have been established to restrict users' access to application configuration.
Requested by: The organization
Responsible: Developer

Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: According to the information reviewed in storing data in the application, the user information is stored in a database, while the configuration files are managed in the server web directory. **Result:** It complies.

PUID: [SEC-CAT-IND-011]
Requirement description: The application should not be housed on a general-purpose server if the application is designated as criticism or high availability by the Information Security Officer (ISSO). Critical applications not shared by other less critical applications must be implemented on servers.
Source: ✓ **V-222635:** The application must not be hosted on a general-purpose machine if the application is designated as critical or high availability by the ISSO. Deploy mission critical applications on servers that are not shared by other less critical applications. Critical applications should not be hosted on a multi-purpose server with other applications. Applications that share resources are susceptible to the other shared application security defects. Even if the critical application is designed and deployed securely, an application that is not designed and deployed securely, can cause resource issues and possibly crash effecting the critical application. [13].
Priority: Not described
Rationale: This requirement tends to separate critical applications in dedicated servers since it minimizes the risks associated with security vulnerabilities of other shared applications. Even if the critical application is designed and deployed safely, the presence of less safe applications on the same server could affect the stability and safety of the critical application.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To_be_determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Verification of accommodation on dedicated servers:** **1.** Confirm that the critical application is housed in dedicated servers that are not shared by other less critical applications. ✓ **Evaluation of resource separation:**

1. Verify that there are no shared resources between critical and less critical applications on the server.

✓ **Confirmation of the designation by the ISSO:**

1. Validate that the application is correctly designated as criticism or high availability by the Information Security Officer.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

According to the official website's default implementation and installation instructions, the database and web application are promoted on a single server.

Result: It doesn't comply.

PUID: [SEC-CAT-IND-012]

Requirement description: The application must implement an adequate limitation of access routes to prevent malicious route manipulation that can lead to unauthorized revelation of files or directories.

Source:

✓ **CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal'):**

The product uses external input to construct a pathname that is intended to identify a file or directory that is located underneath a restricted parent directory, but the product does not properly neutralize special elements within the pathname that can cause the pathname to resolve to a location that is outside of the restricted directory.

Many file operations are intended to take place within a restricted directory. By using special elements such as ".." and "/" separators, attackers can escape outside of the restricted location to access files or directories that are elsewhere on the system. One of the most common special elements is the "../" sequence, which in most modern operating systems is interpreted as the parent directory of the current location. This is referred to as relative path traversal. Path traversal also covers the use of absolute pathnames such as "/usr/local/bin", which may also be useful in accessing unexpected files. This is referred to as absolute path traversal.

In many programming languages, the injection of a null byte (the 0 or NUL) may allow an attacker to truncate a generated filename to widen the scope of attack. For example, the product may add ".txt" to any pathname, thus limiting the attacker to text files, but a null injection may effectively remove this restriction. [14]

Priority: Not described

Rationale: This requirement prevents the inadequate limitation of access routes since it allows attackers to avoid restrictions on directories and access files or directories outside the restricted location. This type of vulnerability, known as 'Path Traversal', can be exploited by introducing special elements on routes, such as ".." and "/", allowing the attacker to navigate unauthorized locations.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Verification of access route limitation:**

  1. Confirm that the application implements adequate mechanisms to limit access routes and avoid malicious manipulation.

- ✓ **CWE-22 vulnerability evaluation:**

  1. Perform specific tests to identify the presence of CWE-22 vulnerability ('traversal path').
  2. Confirm that the application correctly neutralizes special elements on routes to prevent a resolution to unauthorized locations.

- ✓ **Null characters injection prevention:**

  1. Verify that the application is protected against injection of a null character (0 or NUL) to avoid manipulating generated file names.

- ✓ **Sequences protection "../" and "/":**

  1. Confirm that the application protects against sequences such as "../" and "/" that could allow navigation outside the restricted location.

- ✓ **Absolute route injection tests:**

  1. Perform tests to confirm that the application resists the injection of absolute routes, such as "/usr/local/bin", to access unexpected files.

Requested by: The organization

Responsible: Developer

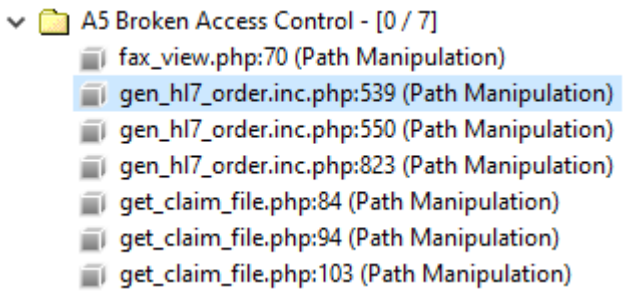
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are findings for “path manipulation.”



Result: It doesn't comply.

PUID: [SEC-CAT-IND-013]

Requirement description: The application must be protected against vulnerabilities of canonical representation. An adequate canonical form must be selected, and the entire user entry must be canonized before authorization decisions are made.

Source:

- ✓ **V-222605:** The application must protect from canonical representation vulnerabilities.

  A suitable canonical form should be chosen and all user input canonicalized into that form before any authorization decisions are performed.

  Security checks should be carried out after decoding is completed. Moreover, it is recommended to check that the encoding method chosen is a valid canonical encoding for the symbol it represents. [13].

Priority: Not described

Rationale: This requirement aims to establish a defense against the vulnerabilities of canonical representation to prevent possible attacks that could take advantage of discrepancies in data representation. The choice of a canonical form and the canonization of the user entry before making authorization decisions contribute to ensuring the coherence and integral security of the application.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Adequate selection of the canonical form:**

  1. Confirm that the application has selected an adequate canonical form for data representation.

- ✓ **User entry canonization:**

  1. Verify that all user entries are canonized in the canonical form chosen before making authorization decisions.
  2. Performing security checks after

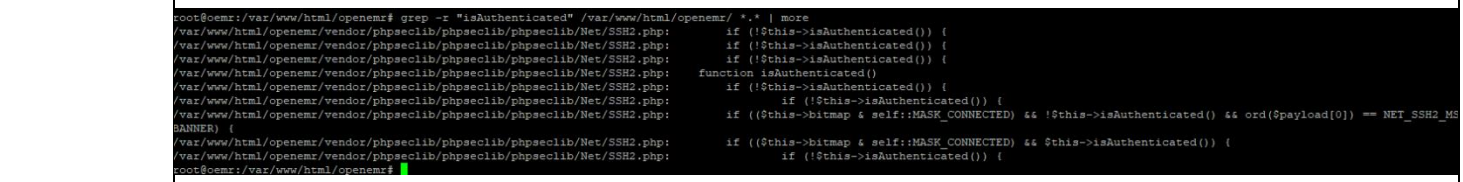
- ✓ **Decoding:**

1. Confirm that safety checks are made after the decoding of the user entry is completed. ✓ **Validation of the canonical coding form:** **1.** Verify that validation is performed to ensure that the chosen coding method is a valid canonical coding for the symbol it represents.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: A DAST and SAST have been performed, and there are no findings for “canonical representations” attacks. **Result:** It complies.

SECURE FAIL

PUID: [SEC-CAT-IND-014]
Requirement description: The application must incorporate a "safe failure" approach through safe error management to prevent vulnerabilities derived from incorrect management of return codes and exceptions in all system components
Source: ✓ **V-222656:** The application must not be subject to error handling vulnerabilities. Ensure proper return code and exception handling is implemented throughout the application. Error handling is the failure to check the return values of functions or catch top level exceptions within a program. Improper error handling in an application can lead to an application failure or possibly result in the application entering an insecure state. The primary way to detect error handling vulnerabilities is to perform code reviews. If a manual code review cannot be performed, static code analysis tools should be employed in conjunction with tests to help force the error conditions by specifying invalid input (such as fuzzed data and malformed filenames) and by using different accounts to run the application. These tests may give indications of vulnerability, but they are not comprehensive. In order to minimize error handling errors, ensure proper return code and exception handling is implemented throughout the application. [13] ✓ **V-222585:** The application must fail to a secure state if system initialization fails, shutdown fails, or aborts fail. Fix any vulnerability found when the application is an insecure state (initialization, shutdown and aborts).

In general, application security mechanisms should be designed so that a failure will follow the same execution path as disallowing the operation. For example, security methods, such as isAuthorized(), isAuthenticated(), and validate(), should all return false if there is an exception during processing. If security controls can throw exceptions, they must be very clear about exactly what that condition means. [13]
Priority: Not described
Rationale: This requirement aims to avoid the incorrect handling of errors, such as the lack of verification of the values of return of functions or the inadequate capture of exceptions that can lead to application failures or even leave it in an insecure state. Guaranteeing proper error management becomes essential to prevent possible vulnerability points that could be exploited.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Code review and static analysis tools:** **1.** Confirm that a manual code review is carried out to detect vulnerabilities in error management. **2.** Validate the use of static code analysis tools in the absence of manual reviews. ✓ **Proper management of return codes and exceptions:** **1.** Verify that the application appropriately manages return codes and exceptions throughout the system. **2.** Confirm that critical functions, such as initialization, off, and abortions, lead to a safe state in case of failure. ✓ **Stress tests and error conditions:** **1.** Execute stress tests with error conditions, such as incorrect data or malformed file names. **2.** Validate that the application responds appropriately to simulated error conditions. ✓ **Initialization, off, and abortion failures:** **1.** Confirm that the application enters a safe state if system initialization, off, or abortions fail. **2.** Identify and correct any vulnerability discovered during such error situations. ✓ **Consistency in the management of errors and security:** **1.** Verify that the application mechanisms follow the same execution path in case of failure, similar to the denial of the operation. **2.** Confirm that security controls return coherent results, such as "false" during processing with exceptions.
Requested by: The organization
Responsible: Developer
Configurable value: Not described

Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: A DAST and SAST have been performed, and there are no findings for avoiding "secure fail"; moreover, security functions such as "isAuthenticated()" are implemented in code files.    **Result:** It complies.

MEMORY PROTECTION

PUID: [SEC-CAT-IND-015]
Requirement description: The application should not be vulnerable to race conditions.
Source: ✓ **V-222567**: The application must not be vulnerable to race conditions. Be aware of potential timing issues related to application programming calls when designing and building the application. Validate those variable values do not change while a switch event is occurring. [13] ✓ **CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition'):** The product contains a code sequence that can run concurrently with other code, and the code sequence requires temporary, exclusive access to a shared resource, but a timing window exists in which the shared resource can be modified by another code sequence that is operating concurrently. [14]
Priority: Not described
Rationale: This requirement seeks to guarantee adequate synchronization and management of shared resources to prevent multiple code sequences simultaneously accessing a shared resource, thus reducing the risk of race conditions and possible safety failures. The race conditions can lead to unexpected results and potentially to security vulnerabilities.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria:

- ✓ **Analysis of possible career conditions:**

  1. Verify that an exhaustive analysis of possible timing problems related to application programming calls during the design and development of the application has been performed.

- ✓ **Validation of the stability of variables:**

  1. Confirm that validations have been implemented to ensure that the values of the variables do not change while a change event occurs.

- ✓ **Vulnerability evaluation:**

  1. Perform specific tests to identify any potential career condition in the application.

- ✓ **Adhesion to good security practices:**

  1. Verify that the application follows the best design and development practices to prevent career conditions according to the guidelines provided by sources V-222567 and CWE-362.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are no findings of "race conditions"; moreover, security functions such as "flock()" are implemented in code files.

```
/var/www/html/openemr/gacl/Cache_Lite/Lite.php:        if ($this->_fileLocking) @flock($fp, LOCK_SH);
/var/www/html/openemr/gacl/Cache_Lite/Lite.php:            if ($this->_fileLocking) @flock($fp, LOCK_UN);
/var/www/html/openemr/gacl/Cache_Lite/Lite.php:            if ($this->_fileLocking) @flock($fp, LOCK_EX);
/var/www/html/openemr/gacl/Cache_Lite/Lite.php:            if ($this->_fileLocking) @flock($fp, LOCK_UN);
/var/www/html/openemr/vendor/smarty/smarty/FAQ:   locking. Comment out the flock() call in _write_file to get around this,
/var/www/html/openemr/vendor/symfony/process/Pipes/WindowsPipes.php:                    if (!flock($h, \LOCK_EX | \LOCK_NB)) {
/var/www/html/openemr/vendor/symfony/process/Pipes/WindowsPipes.php:                        flock($this->lockHandles[$pipe], \LOCK_UN);
/var/www/html/openemr/vendor/symfony/process/Pipes/WindowsPipes.php:                        flock($this->lockHandles[$pipe], \LOCK_UN);
```

Result: It complies.

PUID: [SEC-CAT-IND-016]

Requirement description: The application must implement measures to prevent memory use after being released since referring to memory after release can cause program failures, the use of unexpected values, or execute unwanted code.

Source:

- ✓ **CWE-416 Use After Free:** Referencing memory after it has been freed can cause a program to crash, use unexpected values, or execute code.

The use of previously-freed memory can have any number of adverse consequences, ranging from the corruption of valid data to the execution of arbitrary code, depending on the instantiation and timing of the flaw. The simplest way data corruption may occur involves the system's reuse of the freed memory. Use-after-free errors have two common and sometimes overlapping causes:

Error conditions and other exceptional circumstances.
Confusion over which part of the program is responsible for freeing the memory.
In this scenario, the memory in question is allocated to another pointer validly at some point after it has been freed. The original pointer to the freed memory is used again and points to somewhere within the new allocation. As the data is changed, it corrupts the validly used memory; this induces undefined behavior in the process.

If the newly allocated data happens to hold a class, in C++ for example, various function pointers may be scattered within the heap data. If one of these function pointers is overwritten with an address to valid shellcode, execution of arbitrary code can be achieved. [14]

Priority: Not described

Rationale: The purpose of this requirement is to avoid the use of memory after its release since this practice can have adverse consequences ranging from the corruption of valid data to the execution of arbitrary code. It is essential to address this vulnerability to prevent situations that could compromise the integrity and security of the application.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To_be_determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Identification of possible points of use after release:**

  1. Verify that an exhaustive code review has been carried out to identify possible instances of use after releasing memory.

- ✓ **Implementation of prevention measures:**

  1. Confirm that the application has implemented adequate measures to prevent use after release, including the proper management of memory release.

- ✓ **Handling error verification:**

  1. Evaluate error management mechanisms to ensure that exceptional circumstances and errors are correctly handled without causing use after release.

- ✓ **Memory flow analysis:**

  1. Perform a detailed memory flow analysis to identify any situation where the released memory is referenced again.

- ✓ **Implementation of good development practices:**

1. Verify that good development practices are followed, including careful memory management and appropriate allocation and release of resources.

✓ **Penetration tests:**

1. Perform specific penetration tests to identify and confirm the absence of conditions of use in the application after release.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and no "use after free" findings exist.

Result: It complies.

PUID: [SEC-CAT-IND-017]

Requirement description: The application should prevent null pointer reference, avoiding trying to access a Null pointer, which could result in unexpected failure or exit.

Source:

✓ **CWE-476: NULL Pointer Dereference:** A NULL pointer dereference occurs when the application dereferences a pointer that it expects to be valid, but is NULL, typically causing a crash or exit. NULL pointer dereference issues can occur through a number of flaws, including race conditions, and simple programming omissions. [14]

Priority: Not described

Rationale: This requirement aims to prevent null pointer reference since it is essential to guarantee the stability and safety of the application. The problems of null pointers' reference, such as those described in CWE-476, can lead to unexpected failures, service interruptions, and, in some cases, exploitable vulnerabilities. Implementing measures to avoid null pointer reference helps strengthen the robustness and reliability of the application.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

✓ **Identification of possible null pointer dereferences:**

1. Verify that an exhaustive source code analysis has been conducted to identify possible instances of null pointer dereference.

✓ **Implementation of leading validations:**

1. Confirm that the application includes adequate validations before performing leading operations, verifying the validity of the pointer to avoid Null.

✓ **Stress tests and career conditions:**

1. Perform stress tests and career tests to evaluate the application's resistance to possible cases of null pointer dereference, especially in concurrence situations.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, with no "null dereference" findings.

Result: It complies.

PUID: [SEC-CAT-IND-018]

Requirement description: The application must implement controls defined by the organization to protect the system's memory against unauthorized code execution.

Source:

✓ **SI-16 MEMORY PROTECTION**

Control: Implement the following controls to protect the system memory from unauthorized code execution: [Assignment: organization-defined controls].

Discussion: Some adversaries launch attacks with the intent of executing code in non-executable regions of memory or in memory locations that are prohibited. Controls employed to protect memory include data execution prevention and address space layout randomization. Data execution prevention controls can either be hardware-enforced or software-enforced with hardware enforcement providing the greater strength of mechanism. [11]

✓ **CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer:**

The product performs operations on a memory buffer, but it can read from or write to a memory location that is outside of the intended boundary of the buffer. As a result, an attacker may be able to execute arbitrary code, alter the intended control flow, read sensitive information, or cause the system to crash. [14]

Priority: Not described

Rationale: This requirement seeks the protection of system memory since it is essential to prevent attacks that seek to execute code in non-executable memory regions or prohibited memory locations. These controls, such as data execution prevention and randomization of address space

design, are essential to mitigate risks associated with malicious code executions or memory manipulation attempts.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Implementation of controls defined by the organization:**

  1. Verify that the application implements the specific controls defined by the organization to protect the system's memory.

- ✓ **Unauthorized execution prevention:**

  1. Confirm that effective measures have been implemented, such as data execution prevention (DEP), to avoid unauthorized code execution in non-executable memory regions.

- ✓ **Randomization of addresses space (ASLR):**

  1. Validate the implementation of randomization techniques for the address space design to hinder the prediction of memory locations and reduce the risk of malicious code execution.

- ✓ **EVALUATION OF CWE-119:**

  1. Verify that the application does not perform operations in a memory buffer, resulting in readings or deeds outside the planned limits, as described in CWE-119.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are no findings for “unauthorized code execution”; moreover, security functions such as “filter_input() and prepare()” are implemented in code files.

```
root@oemr:/home/oemr#
root@oemr:/home/oemr# grep -r "filter_input" /var/www/html/openemr/ *.* | more
/var/www/html/openemr/library/ajax/upload.php:$patient_id = filter_input(INPUT_GET, 'patient_id');
/var/www/html/openemr/library/ajax/upload.php:$category_id = filter_input(INPUT_GET, 'parent_id');
/var/www/html/openemr/library/ajax/prescription_drugname_lookup.php:    $term = filter_input(INPUT_GET, "term");
/var/www/html/openemr/library/ajax/document_helpers.php:    'term' => (isset($_GET["term"]) ? filter_input(INPUT_GET, 'term') : ''),
/var/www/html/openemr/library/ajax/document_helpers.php:    'sql_limit' => (isset($_GET["limit"]) ? filter_input(INPUT_GET, 'limit') : 20),
```

Result: It complies.

PUID: [SEC-CAT-IND-019]

Requirement description: The application should not be vulnerable to buffer overflow attacks.

Source:

- ✓ **V-222612:** The application must not be vulnerable to overflow attacks.

  Design the application to use a language or compiler that performs automatic bounds checking.

  Use an abstraction library to abstract away risky APIs. Use compiler-based canary mechanisms such as StackGuard, ProPolice, and the Microsoft Visual Studio/GS flag.

  Use OS-level preventative functionality and control user input validation. Patch applications when overflows are identified in vendor products. [13].

- ✓ **CWE-787 Out-of-bounds Write**

  The product writes data past the end, or before the beginning, of the intended buffer.

  Typically, this can result in corruption of data, a crash, or code execution. The product may modify an index or perform pointer arithmetic that references a memory location that is outside of the boundaries of the buffer. A subsequent write operation then produces undefined or unexpected results**.**

  Memory Corruption:

  Often used to describe the consequences of writing to memory outside the bounds of a buffer, or to memory that is invalid, when the root cause is something other than a sequential copy of excessive data from a fixed starting location. This may include issues such as incorrect pointer arithmetic, accessing invalid pointers due to incomplete initialization or memory release, etc. [14]

- ✓ **CWE-125: Out-of-bounds Read**

  The product reads data past the end, or before the beginning, of the intended buffer.

  Typically, this can allow attackers to read sensitive information from other memory locations or cause a crash. A crash can occur when the code reads a variable amount of data and assumes that a sentinel exists to stop the read operation, such as a NUL in a string. The expected sentinel might not be located in the out-of-bounds memory, causing excessive data to be read, leading to a segmentation fault or a buffer overflow. The product may modify an index or perform pointer arithmetic that references a memory location that is outside of the boundaries of the buffer. A subsequent read operation then produces undefined or unexpected results. [14]

- ✓ **CWE-190: Integer Overflow or Wraparound**

  The product performs a calculation that can produce an integer overflow or wraparound, when the logic assumes that the resulting value will always be larger than the original value. This can introduce other weaknesses when the calculation is used for resource management or execution control. [14]

Priority: Not described

Rationale: This requirement aims to avoid overflow attacks, such as Out-Of-Bounds Write (out-of-limit writing), out-no severe security risks that can result in data corruption, application blockages, or even unauthorized code execution. Using safe practices in the design and implementation of the application is essential to prevent these types of attacks and guarantee the integrity and safety of the application.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Overflow prevention:**

  1. Verify that the application is designed to use a language or compiler that automatically performs limit checks.

- ✓ **Use of abstraction library:**

  1. Confirm that the application uses an abstraction library to hide risky APIS.

- ✓ **Canarian mechanisms based on the compiler:**

  1. Validate Canarian mechanisms based on compilers, such as Stack Guard, Propolice, and the Microsoft Visual Studio/GS indicator, to prevent overflow attacks.

- ✓ **Preventive functionality at SO level:**

  1. Verify that the application uses preventive functionalities at the operating system level.

- ✓ **User input validation:**

  1. Confirm that the application implements user entry validation controls to avoid overflow attacks.

- ✓ **Application patch:**

  1. Verify that patches are applied to the application when overflows are identified in supplier products.

- ✓ **Out-O-Bounds Write prevention:**

  1. Confirm that the application prevents Out-Of-Bounds Write type attacks, according to CWE-787 guidelines.

- ✓ **Out-Of-Bound Precending Read:**

  1. Validate that the application prevents Out-Of-Bounds Read attacks, according to CWE-125 guidelines.

- ✓ **Prevention of Integer Overflow or Wraparound:**

1. Confirm that the application prevents Integer Overflow or Wraparound Type attacks, according to CWE-190 guidelines.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are no findings for "buffer overflow"; moreover, security functions such as "eval() and prepare()" are implemented in code files.

```
/var/www/html/openemr/docker/library/dockers/dev-php-fpm-5-6-redis/php.ini:; error_reporting(0) around the eval().
/var/www/html/openemr/portal/messaging/secure_chat.php:                          scope.$eval(attrs.ngEnter);
/var/www/html/openemr/portal/patient/fwk/libs/util/zip.lib.php:        eval('$hexdtime = "' . $hexdtime . '";');
```

Result: It complies.

XML SECURITY

PUID: [SEC-CAT-IND-020]

Requirement description: The application should be designed to use components that are not vulnerable to XML attacks

Source:

- ✓ **V-222573**: Design the application to utilize components that are not vulnerable to XML attacks. The application must not be vulnerable to XML-oriented attacks.

  Examples include but are not limited to:
  - XML Injection
  - XML related Denial of Service
  - XPATH injection
  - XML Signature attacks
  - XML Spoofing

  Patch the application components when vulnerabilities are discovered. [13].

Priority: Not described

Rationale: This requirement seeks to promote the implementation of components that are not vulnerable to XML attacks, being crucial to prevent the exploitation of possible weaknesses in application security. Since XML attacks can compromise the integrity, confidentiality, and availability of information, ensuring effective protection against these risks is essential.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Evaluation of components used:**

  1. Verify that the application has been designed to use components that are not vulnerable to XML attacks.

- ✓ **Tests against XML attacks:**

  1. Perform specific tests to identify possible vulnerabilities to XML attacks, including but not limited to XML injection, denial of XML-related service, XPath injection, XML signature attacks, and identity supplantation through XML.

- ✓ **PATCH IMPLEMENTATION:**

  1. Confirm that the application has an established process to apply patches to the components when vulnerabilities related to XML are discovered.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, with no "XML attacks" findings.

Result: It complies.

REPLAY-RESISTANT MECHANISM IN AUTHENTICATION

PUID: [SEC-CAT-IND-021]

Requirement description: The application must implement repetition-resistant authentication mechanisms for network access for privileged and non-privileged accounts.

Source:
- ✓ **V-222530**: The application must implement replay-resistant authentication mechanisms for network access to privileged accounts.[13].
- ✓ **V-222531**: The application must implement replay-resistant authentication mechanisms for network access to non-privileged accounts. [13].
- ✓ **A07:2021 – Identification and Authentication Failures:** Ensure registration, credential recovery, and API pathways are hardened against account enumeration attacks by using the same messages for all outcomes [3]

Priority: Not described

Rationale: Implementing repetition-resistant authentication mechanisms is essential to mitigate the risk of reproduction attacks and strengthen the safety of privileged and non-privileged accounts. Guarantee that the registration paths, credentials recovery, and API are hardened against account enumeration attacks to improve the system's resistance to possible threats.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Authentication resistant to repetitions - privileged accounts:**

  1. Verify that the application implements authentication mechanisms resistant to repetitions for access to privileged accounts.

- ✓ **Authentication resistant to repetitions - non-privileged accounts:**

  1. Confirm that the application implements authentication mechanisms resistant to repetitions for access to non-privileged accounts.

- ✓ **Hardening against accounts enumeration attacks:**

  1. Evaluate the registration paths, credentials, and API recovery to ensure they are hardened against account enumeration attacks.

- ✓ **Consistent use of messages:**

  1. Verify that the same messages are used for all results on the registration paths, recovery of credentials, and API to prevent account enumeration attacks.

Requested by: The organization

Responsible: Developer

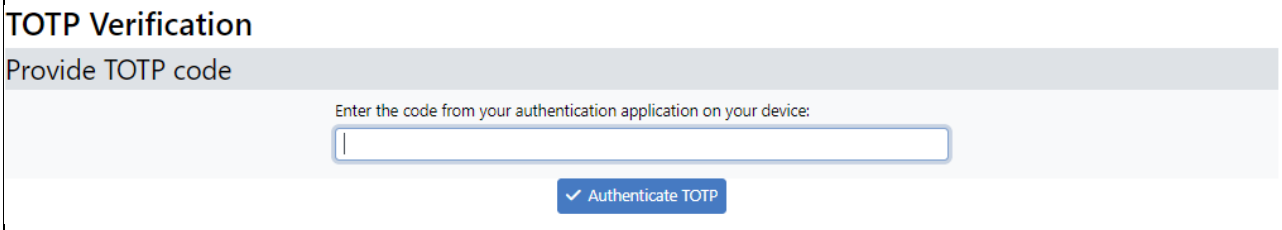
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are no "repetition attacks" findings. Moreover, when any user tries to log in, a time-based one-time password (TOTP) and a token expiration period are requested before conferring access.



Result: It complies.

SOA SECURITY

PUID: [SEC-CAT-IND-022]

Requirement description: Applications with SOAP messages requiring integrity must include the following message elements: -Message ID-Service Request-Timestamp-SAML Assertion (optionally included in messages), and all elements of the message must be digitally signed.

Source:

- ✓ **V-222398:** Applications with SOAP messages requiring integrity must include the following message elements: -Message ID-Service Request-Timestamp-SAML Assertion (optionally included in messages), and all elements of the message must be digitally signed.

  Design and configure the application to sign the following message elements for SOAP messages requiring integrity:

  - Message ID
  - Service Request
  - Timestamp
  - SAML Assertion
  - Message elements [13].

Priority: Not described

Rationale: To preserve the secrecy and accuracy of sensitive health information when it is communicated beyond physical security, it is essential to utilize widely accepted cryptographic algorithms and protocols commonly used and accepted in the industry.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Digital signature in specific elements:**
  Confirm that the application significantly signs the following SOAP messages:
  1. Message identifier
  2. SERVICE APPLICATION
  3. Timestamp
  4. SAML statement (if included in the messages)
  5. All the elements of the message.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The application does not implement SOAP messages; therefore, it is considered not applicable for the audit (N/a). **Result:** N/a

PUID: [SEC-CAT-IND-023]
Requirement description: The application must authenticate all the endpoint devices connected to the network before establishing any connection.
Source: ✓ V-222533: The application must authenticate all network connected endpoint devices before establishing any connection. [13]. ✓ V-222534: Configure the application to utilize mutual authentication when the application is processing non-releasable data. [13].
Priority: Not described
Rationale: This requirement seeks to promote the authentication of endpoint devices since it is essential to guarantee the integrity and safety of the network. Authenticating each device before allowing connections reduces the risk of unauthorized access and strengthens the system's global security.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Device authentication verification:** 1. Confirm that the application implements an authentication mechanism for all devices connected to the network before establishing any connection. ✓ **Use of mutual authentication:** 1. Validate that the application is configured to use mutual authentication. 2. Confirm that mutual authentication applies specifically when the application is processing non-disseminable data. ✓ **Authentication of devices in initial connections:** 1. Verify that the Authentic Application Final point devices are ready before allowing the initial connection.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná

13/12/2023
Auditory result: The application does not implement endpoint device authentication; therefore, it is considered not applicable for the audit (N/a). **Result:** N/a

PUID: [SEC-CAT-IND-024]
Requirement description: Safe messages with WS-Security must have time stamps indicating when they were created and expired. The application must be designed to use these time stamps and sequence numbers in WS-Security messages.
Source: ✓ **V-222399:** Messages protected with WS-Security must use time stamps with creation and expiration times. Design and configure applications using WS-Security messages to use time stamps with creation and expiration times and sequence numbers. [13]
Priority: Not described
Rationale: This requirement promotes the inclusion of temporary stamps with times of creation and expiration in WS-Security messages since it is essential to guarantee the safety and integrity of the transmitted information. This prevents the reproduction of old messages and reinforces the authenticity of the messages, providing an additional layer of protection against threats such as malicious reproduction.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Verification of temporary seals:** **1.** Confirm that messages protected with WS-Security include temporary seals. ✓ **Evaluation of Creation and Expiration Times:** **1.** Validate those temporary seals in WS-Security messages to ensure they contain adequate creation and expiration times. ✓ **Confirmation of sequence numbers:** **1.** Verify that WS-Security messages contain sequence numbers to avoid malicious reproduction. ✓ **Configuration compliance:** **1.** Confirm that the application is designed and configured to use temporary seals with creation and expiration times and sequence numbers in WS-Security messages.
Requested by: The organization
Responsible: Developer
Configurable value: Not described

Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The application does not implement SOAP messages; therefore, it is considered not applicable for the audit (N/a). **Result:** N/a

PUID: [SEC-CAT-IND-025]
Requirement description: The application must verify the validity periods in all messages that WS-SECURITY or SAML statements use. Design and configure the application to use and verify validity periods in all token WS-Security profiles and SAML statements.
Source: ✓ **V-222400:** Validity periods must be verified on all application messages using WS-Security or SAML assertions. Design and configure the application to use validity periods, ensure validity periods are verified on all WS-Security token profiles and SAML Assertions. [13].
Priority: Not described
Rationale: This requirement seeks the verification of periods of validity in messages of the application used by WS-Security or SAML statements since it is crucial to guarantee the safety and integrity of data transmissions. When confirming the validity of these periods, the risk of manipulation or unauthorized access is reduced through non-temporarily valid messages.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Validity period verification implementation:** **1.** Confirm that the application implements mechanisms to verify validity periods in all messages WS-Security or SAML statements use. ✓ **Use of validity periods:** **1.** Verify that the application is designed and configured to explicitly use validity periods in all Token WS-SECURITY profiles and SAML statements. ✓ **Verification of periods in Token WS-Security:** **1.** Validate those validity periods are verified in all messages that use token WS-Security profiles. ✓ **Verification of periods in SAML statements:**

1. Validate those validity periods are verified in all messages that include SAML statements.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The application does not implement SOAP services for exposing; therefore, it is considered not applicable for the audit (N/a). **Result:** N/a

2.9.3.5 Security Misconfiguration

PUID: [SEC-CAT-SEM-001]
Requirement description: The application must define, document, and implement configurations, including security, hardware, software, services, and networks, adjusting them in the most restrictive way consistent with operational requirements.
Source: ✓ **8.9 Configuration management** Configurations, including security configurations, of hardware, software, services, and networks should be established, documented, implemented, monitored, and reviewed (It could be included to fortify the need for implementing security configurations by file codes). [7] ✓ **CM-6 CONFIGURATION SETTINGS** Control: a. Establish and document configuration settings for components employed within the system that reflect the most restrictive mode consistent with operational requirements using [Assignment: organization-defined common secure configurations]; [11]. ✓ **CM-7 LEAST FUNCTIONALITY** Control: a. Configure the system to provide only [Assignment: organization-defined mission essential capabilities]; and b. Prohibit or restrict the use of the following functions, ports, protocols, software, and/or services: [Assignment: organization-defined prohibited or restricted functions, system ports, protocols, software, and/or services]. [11]. ✓ **V-222430:** The application must execute without excessive account permissions. Configure the application accounts with minimalist privileges. Do not allow the application to operate with admin credentials. [13]

- ✓ **A01:2021** – Broken Access Control:

  Avoid violation of the principle of least privilege or deny by default, where access should only be granted for particular capabilities, roles, or users, but is available to anyone. [3]

Priority: Not described

Rationale: This requirement aims to reduce risks and ensure the safe operation of the application through effective configuration management, especially in security aspects. Establishing and documenting configurations aligned with the lower functionality principle helps reduce the attack surface and minimize possible vulnerabilities.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To_be_determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Establishment of documented configurations:**

  1. Verify that the application establishes and documents configurations for components used in the system, reflecting the most restrictive mode of operational requirements.

- ✓ **Principle of lower functionality:**

  1. Confirm that the application is configured to provide only the essential capacities for the mission, as defined by the organization.
  2. Verify that the application prohibits or restricts the use of functions, ports, protocols, software and/or services defined by the organization.

- ✓ **Execution with minimal permits:**

  1. Confirm that the application is executed without excessive permits.
  2. Validate that application accounts are configured with minimal privileges and that the application does not operate with administrator credentials.

- ✓ **Avoid violation of the principle of less privilege:**

  1. Evaluate that the application avoids the violation of the principle of less privilege, ensuring that access is granted only for specific capabilities, roles, or users and is not available for anyone.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are "MISCONFIGURATION SETTINGS" findings according to CM-6 criteria.

Result: It doesn't comply.

PUID: [SEC-CAT-SEM-002]

Requirement description: The application must define and implement due security settings to avoid DDoS attacks

Source:

- ✓ SC-5 DENIAL-OF-SERVICE PROTECTION
  Control:
  a. [Selection: Protect against; Limit] the effects of the following types of denial-of-service events: [Assignment: organization-defined types of denial-of-service events]; and
  b. Employ the following controls to achieve the denial-of-service objective: [Assignment: organization-defined controls by type of denial-of-service event].

  Discussion: Denial-of-service events may occur due to a variety of internal and external causes, such as an attack by an adversary or a lack of planning to support organizational needs with respect to capacity and bandwidth. Such attacks can occur across a wide range of network protocols (e.g., IPv4, IPv6). A variety of technologies are available to limit or eliminate the origination and effects of denial-of-service events. For example, boundary protection devices can filter certain types of packets to protect system components on internal networks from being directly affected by or the source of denial-of-service attacks. Employing increased network capacity and bandwidth combined with service redundancy also reduces the susceptibility to denial-of-service events. [11].

- ✓ **V-222593:** XML-based applications must mitigate DoS attacks by using XML filters, parser options, or gateways.

  Implement:

  - Validation against recursive payloads
  - Validation against oversized payloads
  - Protection against XML entity expansion
  - Validation against overlong element names
  - Optimized configuration for maximum message throughput in order to ensure DoS attacks against web services are limited. [13]

- ✓ **V-222594:** The application must restrict the ability to launch Denial of Service (DoS) attacks against itself or other information systems. Design and deploy the application to utilize controls that will prevent the application from being affected by DoS attacks or being used to attack other systems. This includes but is not limited to utilizing throttling techniques for application traffic such as QoS or implementing logic controls within the application code itself that prevents application use those results in network or system capabilities being exceeded. [13]

- ✓ **A01:2021 – Broken Access Control:** Rate limit API and controller access to minimize the harm from automated attack tooling [3]

- ✓ Authorized personnel should never be refused access and should not have a difficult time obtaining access [15]

- ✓ **A04:2021 – Insecure Design:** Limit resource consumption by user or service. [3]

Priority: Not described

Rationale: This requirement seeks to mitigate the events of denial of service that can originate internally and externally due to various causes, such as adversary attacks or the lack of planning to meet organizational needs regarding capacity and bandwidth. These events can affect multiple network protocols; therefore, it is essential to implement technologies and controls to limit or eliminate the origin and effects of such events. Implementing these controls is crucial to guarantee the continuous availability and functionality of the application against possible service denial events.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Mitigation of two attacks in XML applications:**

  1. Validate that XML-based applications implement mitigation measures against two attacks using XML filters, analyzers analysis, or XML stakes.
  2. Confirm the implementation of validation against recursive payloads, excessive size payloads, expansion protection of XML entities, validation against names of excessively long elements, and optimized configuration for the maximum message performance to limit two attacks against web services.

- ✓ **Restriction of the ability to launch attacks two:**

  1. Please verify that the application restricts the ability to launch denial attacks against itself or other information systems.
  2. Confirm that controls, such as traffic regulation (QOs) or logical rules within the application code, are implemented to prevent the application from being affected by two attacks or used to attack other systems.

- ✓ **Access and access control limitation:**

  1. Validate that the application limits the API and controller access to minimize the damage caused by automated attack tools.
  2. Confirm that authorized personnel are not rejected and have no difficulty obtaining access.

- ✓ **Safe design and limitation of resource consumption:**

  1. Validate that the application design limits the consumption of resources per user or service.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

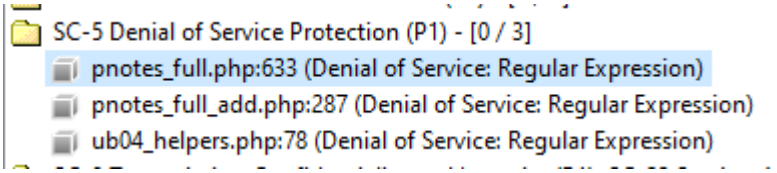
Version history: v1.0

Author and date:

Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are "SC-5 DENIAL-OF-SERVICE PROTECTION" findings.



Result: It doesn't comply.

PUID: [SEC-CAT-SEM-003]

Requirement description: The application must implement measures to validate the deserialization of non-reliable data and ensure the resulting data are valid before processing them.

Source:

- ✓ **CWE-502: Deserialization of Untrusted Data:**

  The product deserializes untrusted data without sufficiently verifying that the resulting data will be valid.

  It is often convenient to serialize objects for communication or to save them for later use. However, deserialized data or code can often be modified without using the provided accessor functions if it does not use cryptography to protect itself. Furthermore, any cryptography would still be client-side security -- which is a dangerous security assumption.

  Data that is untrusted cannot be trusted to be well-formed.

  When developers place no restrictions on "gadget chains," or series of instances and method invocations that can self-execute during the deserialization process (i.e., before the object is returned to the caller), it is sometimes possible for attackers to leverage them to perform unauthorized actions, like generating a shell.

  Suggestion:

  To mitigate this, explicitly define final readObject() to prevent deserialization. [3]

Priority: Not described

Rationale: This requirement aims to combat the deserialization of non-reliable data without adequate verification that can introduce significant vulnerabilities, allowing attackers to modify the data unauthorized or execute malicious code. Implementing a robust validation of deserialization is essential to mitigate security risks and guarantee the data's integrity.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Implementation of restrictions on "Gadget Chains":**

  1. Verify that the application implements "device chains" restrictions during deserialization to prevent possible unauthorized actions, such as malicious code execution.

- ✓ **Explicit definition of readObject():**

  1. Validate that it is explicitly defined readObject() final to prevent unauthorized deserialization.

- ✓ **Deserialization prevention without verification:**

  1. Confirm that the application incorporates measures to prevent non-reliable deserialization without sufficient verification of the validity of the resulting data.

- ✓ **Mitigation of risks associated with deserialization:**

  1. Evaluate the application to ensure effective measures have been implemented to mitigate the risks associated with deserializing non-reliable data.

- ✓ **Compliance with CWE-502 suggestions:**

  1. Verify that the application follows the suggestions provided by CWE-502, including the explicit definition of Final readObject() to prevent deserialization.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and no findings exist for “Deserialization of Untrusted Data”. Moreover, after checking the code files of the application, there are implemented secure functions for serializing and unserializing JSON files such as json_encode(), json_decode();

```
root@oemr:/home/oemr# grep -r "json_decode(" /var/www/html/openemr/ *.* | more
/var/www/html/openemr/src/FHIR/SMART/SMARTLaunchToken.php:        $context = json_decode($jsonEncoded, true, 512, JSON_THROW_ON_ERROR);
/var/www/html/openemr/src/FHIR/SMART/ClientAdminController.php:                $token['scope'] = json_decode($token['scope'], true);
/var/www/html/openemr/src/FHIR/R4/PHPFHIRResponseParser.php:        $decoded = json_decode($input, true);
/var/www/html/openemr/src/PaymentProcessing/PaymentProcessing.php:            $auditData = json_decode($auditData, true);
/var/www/html/openemr/src/MedicalDevice/MedicalDevice.php:        $this->udi_data = json_decode($udi_data, true);
/var/www/html/openemr/src/MedicalDevice/MedicalDevice.php:        $udiData = json_decode($data, true);
/var/www/html/openemr/src/PatientPortal/Chat/ChatModel.php:                $u = json_decode($row['recip_id'], true);
/var/www/html/openemr/src/Common/Auth/OpenIDConnect/JWT/RsaSha384Signer.php:                $jwk = json_decode($key);
/var/www/html/openemr/src/Common/Auth/OpenIDConnect/JWT/JsonWebKeyParser.php:        $refreshTokenData = \json_decode($refreshToken, true);
/var/www/html/openemr/src/Common/Auth/OpenIDConnect/JWT/JsonWebKeySet.php:        $jwks = json_decode($content);
/var/www/html/openemr/src/Common/Auth/OpenIDConnect/Grant/CustomRefreshTokenGrant.php:                $decodedContext = \json_decode($context, true);
/var/www/html/openemr/src/Common/Auth/MfaUtils.php:                $regobj = json_decode($row['varl']);
/var/www/html/openemr/src/Common/Auth/MfaUtils.php:                json_decode($tmprow['login_work_area']), // these are the original challenge requests
/var/www/html/openemr/src/Common/Auth/MfaUtils.php:                json_decode($token)
/var/www/html/openemr/src/Common/Http/oeOAuth.php:            $token = @json_decode(@file_get_contents($file), true);
/var/www/html/openemr/src/Common/Http/HttpRestRequest.php:            return (array) (json_decode($this->requestBody));
/var/www/html/openemr/src/Common/Http/oeHttpResponse.php:        return json_decode($this->response->getBody(), $asArray);
/var/www/html/openemr/src/Pharmacy/Services/ImportPharmacies.php:        $pharmacyObj = json_decode($body, true, 512, 0);
/var/www/html/openemr/src/Cqm/Generator.php:        $oids = json_decode(file_get_contents($oids_file), true);
/var/www/html/openemr/src/Cqm/CqmClient.php:            return json_decode(
/var/www/html/openemr/src/Cqm/CqmClient.php:        return json_decode(
/var/www/html/openemr/src/Cqm/CqmClient.php:            return json_decode(
/var/www/html/openemr/src/Cqm/CqmClient.php:        return json_decode(
/var/www/html/openemr/src/Services/DeviceService.php:            $dataSet = json_decode($json, JSON_THROW_ON_ERROR);
/var/www/html/openemr/src/Services/QuestionnaireService.php:            $q = json_decode($q, true);
/var/www/html/openemr/src/Services/DocumentTemplates/DocumentTemplateService.php:        $q_ob = json_decode($content, true); // only pass array
/var/www/html/openemr/src/Services/QuestionnaireResponseService.php:            $q = json_decode($q, true);
```

Result: It complies.

PUID: [SEC-CAT-SEM-004]

Requirement description: The application must be protected against Vulnerabilities of CROSS-SITE REQUEST FORGERY (CSRF) through the configuration to use unpredictable challenge tokens and verify the HTTP referrer, ensuring that the application was issued from the site itself. In addition, mitigating controls must be implemented as necessary, such as using web reputation services.

Source:

- ✓ **V-222603**: The application must protect from Cross-Site Request Forgery (CSRF) vulnerabilities. Configure the application to use unpredictable challenge tokens and check the HTTP referrer to ensure the request was issued from the site itself. Implement mitigating controls as required such as using web reputation services.

  Cross-Site Request Forgery (CSRF) is an attack where a website user is forced to execute an unwanted action on a website that he or she is currently authenticated to. An attacker, through social engineering (e.g., e-mail or chat) creates a hyperlink which executes unwanted actions on the website the victim is authenticated to and sends it to the victim. If the victim clicks on the link, the action is executed unbeknownst to the victim.

  A CSRF attack executes a website request on behalf of the user which can lead to a compromise of the users data. What is needed to be successful is for the attacker to know the URL, an authenticated application user, and trick the user into clicking the malicious link.

  While XSS is not needed for a CSRF attack to work, XSS vulnerabilities can provide the attacker with a vector to obtain information from the user that may be used in mitigating the risk. The application must not be vulnerable to XSS as an XSS attack can be used to help defeat token, double-submit cookie, referrer and origin-based CSRF defenses. [13]

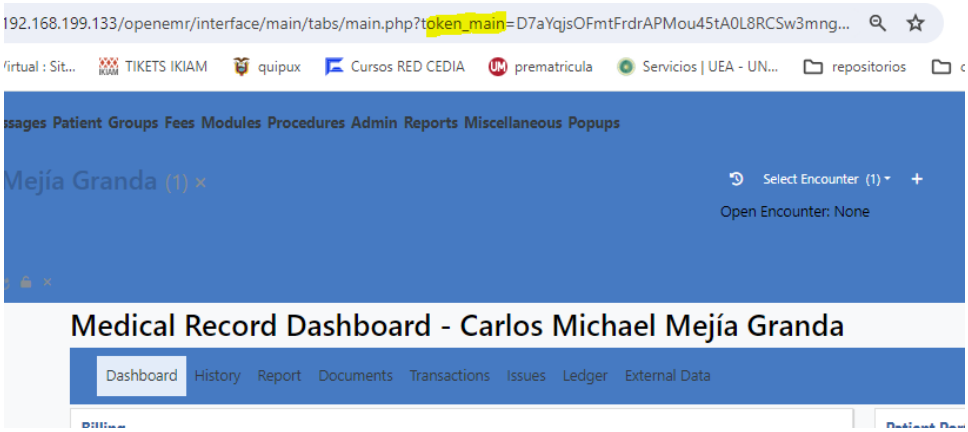
- ✓ **CWE-352: Cross-Site Request Forgery (CSRF)**

  The web application does not, or cannot, sufficiently verify whether a well-formed, valid, consistent request was intentionally provided by the user who submitted the request. When

a web server is designed to receive a request from a client without any mechanism for verifying that it was intentionally sent, then it might be possible for an attacker to trick a client into making an unintentional request to the web server which will be treated as an authentic request. This can be done via a URL, image load, XML HTTP Request, etc. and can result in exposure of data or unintended code execution [14]
Priority: Not described
Rationale: This requirement seeks to promote CSRF protection, which is essential to prevent attacks in which a website user is forced to execute an unwanted action on the site to which it is authenticated. This vulnerability could compromise user data since it allows an attacker to perform requests on behalf of the user without their knowledge.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Use of challenge tokens:** **1.** Verify that the application uses unpredictable challenge tokens to mitigate the vulnerabilities of CSRF. ✓ **REFERRER HTTP VERIFICATION:** **1.** Confirm that the application verifies the HTTP refers to ensure that the application originated from the site itself. ✓ **Implementation of mitigating controls:** **1.** Validate the implementation of mitigating controls, such as web reputation services, as necessary. ✓ **Absence of vulnerabilities of XSS:** **1.** Verify that the application is not vulnerable to Cross-Site Scripting (XSS) attacks since XSS vulnerabilities can provide the attacker with a vector to obtain user information that can be used to mitigate the risk of CSRF. ✓ **CWE-352 Compliance: Cross-Site Request Forgery (CSRF):** **1.** Confirm that the application meets the principles defined in CWE-352 to mitigate the vulnerabilities of CSRF.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023

Auditory result:

A DAST and SAST have been performed, and no findings exist for "CSRF". Moreover, after checking the URL in HTTP requests, an anti-CSRF token is sent for each request operation.



Result: It complies.

PUID: [SEC-CAT-SEM-005]
Requirement description: The application must be configured to deactivate non-essential capabilities.
Source: ✓ **V-222518:** The application must be configured to disable non-essential capabilities. Disable application extraneous application functionality that is not required in order to fulfill the application's mission [13].
Priority: Not described
Rationale: This requirement tends to deactivate non-essential functionalities since reducing the attack surface and mitigating possible safety risks is crucial. When disabled functions are not necessary to meet the mission of the application, exposure to potential vulnerabilities is minimized, and the application security position is reinforced.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Deactivation configuration:** 1. Verify that the application is configured to deactivate non-essential functionalities following the established requirements. ✓ **Effective deactivation:** 1. Confirm that identified non-essential capacities have been effectively deactivated and unavailable.

- ✓ **Disabled functionalities review:**
  1. Evaluate that the deactivation of non-essential functionalities does not negatively affect the mission and main functionality of the application.

Requested by: The organization

Responsible: Developer

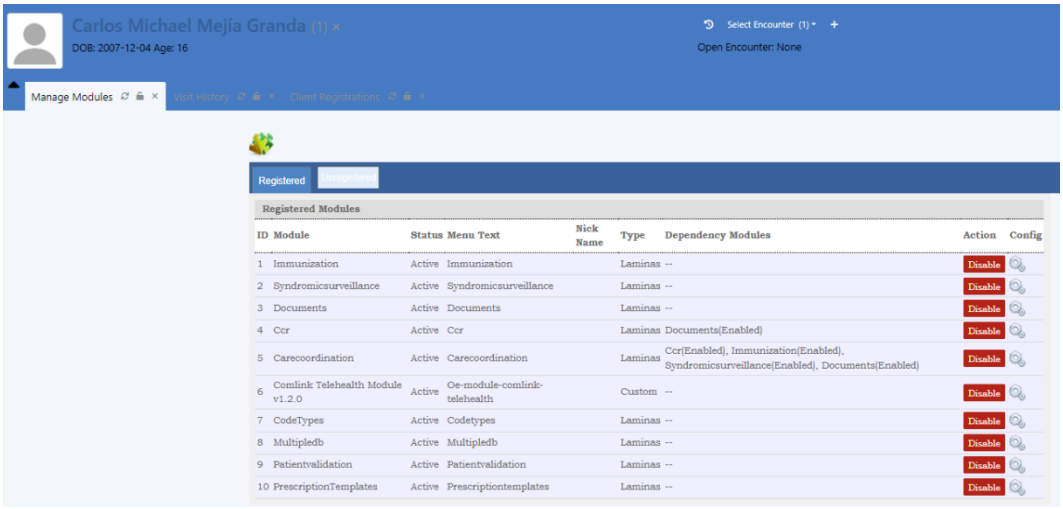
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The application has an administration module that allows the deactivation of modules not necessary for the organization's mission.



Result: It complies.

PUID: [SEC-CAT-SEM-006]

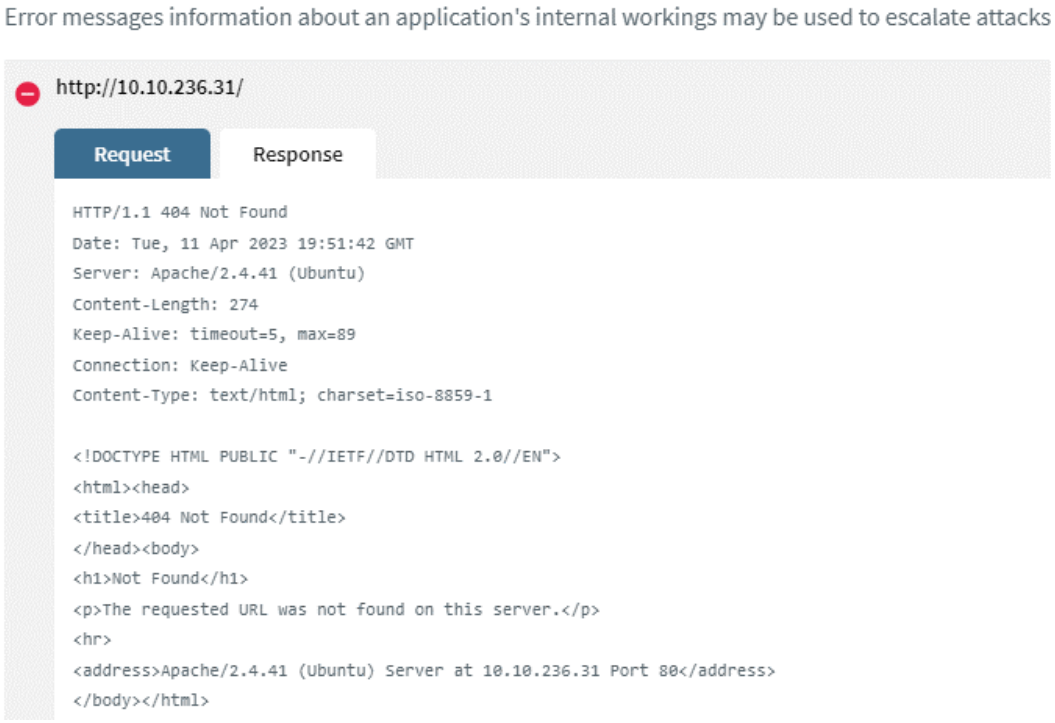
Requirement description: The application should avoid showing users what is not required to users. It is necessary to configure it so that it does not reveal technical details about architecture in error events.

Source:

- ✓ **V-222600:** The application must not disclose unnecessary information to users. Configure the application to not display technical details about the application architecture on error events.

  Applications should not disclose information not required for the transaction. (e.g., a web application should not divulge the fact there is a SQL server database and/or its version).

These events usually occur when the web application has not been configured to send specific error messages for error events. Instead, when a processing anomaly occurs, the application displays technical information about the type of application server, database in use, or other technical details. This provides attackers additional information which they can use to find other attack avenues, or tailor specific attacks, on the application. [13] ✓ **A01:2021 – Broken Access Control:** Disable web server directory listing and ensure file metadata (e.g., git) and backup files are not present within web roots. [3]
Priority: Not described
Rationale: This requirement seeks to reduce the dissemination of unnecessary information, a fundamental pillar of the application's security. The revelation of technical details about the architecture, such as the type of application server or the database used, in error events can provide the attackers with additional information that they could use to identify possible attack points or customize specific attacks.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To_be_determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Non-dissemination configuration:** 1. Verify that the application is configured not to show technical details on the architecture of the application in error events. ✓ **Information dissemination test:** 1. Try to perform operations that generate errors and confirm that the application does not reveal additional technical information beyond what is necessary for the transaction. ✓ **Deactivation of Listing of Web Server Directors:** 1. Validate that the web server does not list directories and ensure that file metadata (for example, git) and backup files are not present in the web roots.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: A DAST and SAST have been performed, and there are “Sensitive data exposure” findings and Application error or warning messages that may expose sensitive information about an application's internal workings to an attacker.



Result: It doesn't comply.

PUID: [SEC-CAT-SEM-007]

Requirement description: The application must identify and authenticate uniquely non-organizational users (or processes that act in the name of non-organizational users).

Source:

- ✓ **V-222556:** The application must uniquely identify and authenticate non-organizational users (or processes acting on behalf of non-organizational users). Configure the application to identify and authenticate all non-organizational users.

  Lack of authentication and identification enables non-organizational users to gain access to the application or possibly other information systems and provides an opportunity for intruders to compromise resources within the application or information system.

  Non-organizational users include all information system users other than organizational users which include organizational employees or individuals the organization deems to have equivalent status of employees (e.g., contractors and guest researchers).

  Non-organizational users must be uniquely identified and authenticated for all accesses other than those accesses explicitly identified and documented by the organization when related to the use of anonymous access, such as accessing a web server. [13]

Priority: Not described

Rationale: This requirement seeks to avoid the lack of authentication and identification that creates vulnerability by allowing non-organizational users to obtain access to the application or possibly other information systems. This measure ensures that non-organizational users are duly identified and authenticated, thus reducing the risk of unauthorized access and protecting resources within the application or information system.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Unique identification:**

  1. Verify that the application implements a mechanism that ensures the unique identification of non-organizational users.

- ✓ **Adequate authentication:**

  1. Confirm that the application carries out an authentication process for all non-organizational users.

- ✓ **Document exceptions:**

  1. Validate that the organization duly documents any exception to identification and authentication for specific access to non-organizational users.

- ✓ **Specific anonymous access:**

  1. Verify that identification and authentication are not required for specific access related to anonymous access, such as access to a web server.

- ✓ **Standard compliance:**

  1. Confirm that the implementation of the identification and authentication of non-organizational users complies with the security standards established by the organization and the requirements of the information source.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

According to the review by the audit team, the application does not allow anonymous or non-organizational users to use the functionality; therefore, it is considered not applicable for the audit (N/a).

Result: N/a

PUID: [SEC-CAT-SEM-008]

Requirement description: The application must identify and authenticate uniquely organizational users (or processes that act in the name of organizational users).

Source:

- ✓ **V-222552:** The application must uniquely identify and authenticate organizational users (or processes acting on behalf of organizational users). Configure the application to uniquely identify and authenticate users and user processes.

  Organizational users (and any processes acting on behalf of users) must be uniquely identified and authenticated for all accesses, except the following:

  - Accesses explicitly identified and documented by the organization. Organizations document specific user actions that can be performed on the information system without identification or authentication; and
  - (ii) Accesses that occur through authorized use of group authenticators without individual authentication. Organizations may require unique identification of individuals in group accounts (e.g., shared privilege accounts) or for detailed accountability of individual activity. [13]

Priority: Not described

Rationale: This requirement ensures that each user and associated process is correctly identified and authenticated, thus reducing the risk of unauthorized access and guaranteeing individual responsibility. The unique identification and authentication of organizational users are fundamental to maintaining the system's integrity and safety.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Uniqueness in identification:**

  1. Verify that the application implements a mechanism to assign unique identifiers to each organizational user.

- ✓ **Unique authentication process:**

  1. Confirm that the application requires that each organizational user pass through a unique authentication process when accessing the system.

- ✓ **Document exceptions:**

  1. Validate that the organization explicitly identifies and documents unique identification and authentication exceptions.

- ✓ **Authorized use of group authenticators:**

  1. Confirm that, in cases allowed by the organization, it can be accessed through the authorized use of group authenticators without individual authentication.

- ✓ **Unique identification requirements in groups:**

1. Verify that the application may require the unique identification of individuals in group accounts (such as shared privileges accounts) when needed for detailed responsibility for individual activity.

✓ **Protection against unauthorized access:**

1. Validate that the application, excluding documented exceptions, prevents unauthorized access by demanding unique identification and authentication in all accesses.

Requested by: The organization

Responsible: Developer

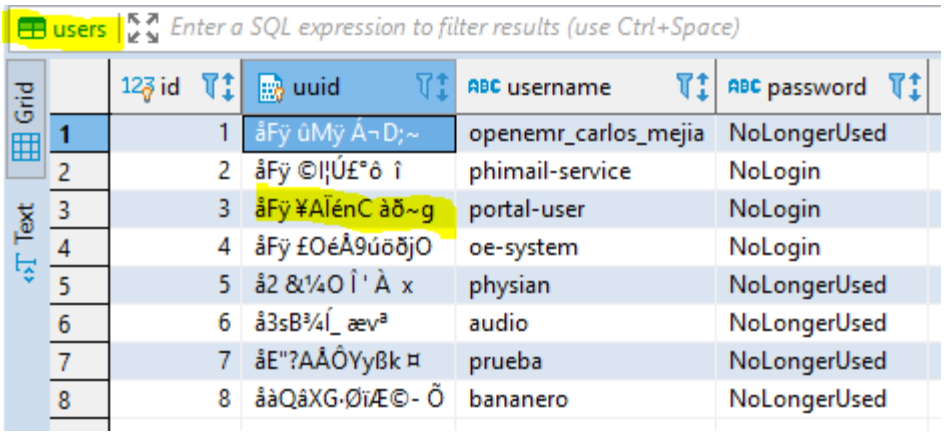
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

According to the review by the audit team, the application identifies each user uniquely in the system.



Result: it complies.

SAML SECURITY

PUID: [SEC-CAT-SEM-009]

Requirement description: The applications that issue SAML statements must use random numbers approved by FIPS to generate SAML session in Saml's AuthnStatement element.

Source:

✓ **V-222573:** Applications making SAML assertions must use FIPS-approved random numbers in the generation of SessionIndex in the SAML element AuthnStatement. Configure the application to use a FIPS-validated cryptographic module. [13].

Priority: Not described

Rationale: When configuring the application to use a cryptographic module validated by FIPS, it is ensured that the random numbers generated comply with recognized and robust safety standards. Using random numbers approved by FIPS in developing SessionIndex in SAML statements is crucial to guarantee the safety and integrity of authenticated transactions.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Use of random numbers approved by FIPS in SAML Assertions:** **1.** Verify that the application issued by SAML statements uses random numbers approved by FIPS to generate SessionIndex in the AuthnStatement element. ✓ **Cryptographic module configuration:** **1.** Confirm that the application is configured to use a cryptographic module that meets FIP Validation standards. ✓ **Compliance with FIPS requirements:** **1.** Validate that the application meets the requirements established by FIPS for generating random numbers in the context of SAML statements.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The installed application for auditing doesn't implement SSO or SAML modules to be analyzed for the audit team; therefore, it is considered not applicable for the audit (N/a). **Result:** N/a

PUID: [SEC-CAT-SEM-010]
Requirement description: The application must ensure that each party that makes statements (Asserting Party) provides unique identifiers for each SAML statement (Security Assertion Markup Language). It is required to design and configure each SAML statement authority to use unique affirmation identifiers.
Source:

✓ **V-222401:** The application must ensure each unique asserting party provides unique assertion ID references for each SAML assertion. Design and configure each SAML assertion authority to use unique assertion identifiers. [13].
Priority: Not described
Rationale: Using unique identifiers for each SAML statement is essential to maintain the integrity and safety of the authentication system. This measure avoids possible conflicts or repetitions of identifiers, guaranteeing the uniqueness of each statement and, therefore, the reliability and validity of the information provided by each entity that makes statements.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Verification of unique identifiers:** **1.** Confirm that the application implements a mechanism that requires unique identifiers for each SAML statement provided by entities that make statements. ✓ **SAML statement authorities configuration evaluation:** **1.** Validate that each SAML statement authority is designed and configured to use unique affirmation identifiers. ✓ **Identifiers repetition tests:** **1.** Perform exhaustive evidence to ensure no repetitions of SAML statement identifiers in different transactions.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The installed application for auditing doesn't implement SSO or SAML modules to be analyzed for the audit team; therefore, it is considered not applicable for the audit (N/a). **Result:** N/a

PUID: [SEC-CAT-SEM-011]
Requirement description: The application must guarantee the use of encrypted statements or equivalent confidentiality protections when the statement data is transmitted through an

intermediary, and the confidentiality of the statements of statements is necessary during the transmission through the intermediary.
Source: ✓ **V-222402:** The application must ensure encrypted assertions, or equivalent confidentiality protections are used when assertion data is passed through an intermediary, and confidentiality of the assertion data is required when passing through the intermediary. [13].
Priority: Not described
Rationale: This requirement is established to safeguard the confidentiality of the statements when transmitted through intermediaries. Proper protection of these data is essential to prevent unauthorized exposure of sensitive information.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Use of encrypted statements:** **1.** Confirm that the application implements encrypted statements or equivalent mechanisms of confidentiality when transmitted through intermediaries. ✓ **Confidentiality protection in intermediaries:** **1.** Verify that the confidentiality of statement data remains properly during transmission through intermediaries.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The installed application for auditing doesn't implement SSO, assertions, or SAML modules to be analyzed for the audit team; therefore, it is considered not applicable for the audit (N/a). **Result:** N/a

PUID: [SEC-CAT-SEM-012]
Requirement description: The application should use both the <NotBefore> y <NotOnOrAfter> elements and the <OneTimeUse> element when using the <Conditions> element in a SAML Assertion.
Source: ✓ **V-222404:** The application must use both the NotBefore and NotOnOrAfter elements or OneTimeUse element when using the Conditions element in a SAML assertion.

Design and configure the application to implement the use of the <NotBefore> and <NotOnOrAfter> or <OneTimeUse> when using the <Conditions> element in a SAML [13] assertion.

- ✓ **V-222405:** The application must ensure if a OneTimeUse element is used in an assertion, there is only one of the same used in the Conditions element portion of an assertion.

  Multiple <OneTimeUse> elements used in a SAML assertion can lead to elevation of privileges, if the application does not process SAML assertions correctly.

  When using OneTimeUse elements in a SAML assertion only allow one, OneTimeUse element to be used in the conditions element of a SAML assertion. [13].

Priority: Not described

Rationale: The proper implementation of the <NotBefore> elements, <NotOnOrAfter>, and <OneTimeUse> contributes to the safety and reliability of the application. Safely using SAML elements in Assertions is essential to prevent the risk of privileges elevation and guarantee the correct processing of SAML Assertions.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Use of elements in SAML Assertions:**

  1. Confirm that the application simultaneously uses the <NotBefore> and <NotOnOrAfter> elements or the <OneTimeUse> element when using the <Conditions> element in a SAML Assertion.

- ✓ **VERIFICATION OF ELEMENTS <OneTimeUse>:**

  1. Ensure that if a <OneTimeUse> element is used in an assertion, only one is allowed in the section of the <Conditions> element of a SAML Assertion.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The installed application for auditing doesn't implement SSO, assertions, or SAML modules to be analyzed for the audit team; therefore, it is considered not applicable for the audit (N/a).

Result: N/a

PUID: [SEC-CAT-SEM-013]

Requirement description: The application should use the NotOnOrAfter condition when using the SubjectConfirmation element in a SAML assertion

Source:

- ✓ **V-222403:** The application must use the NotOnOrAfter condition when using the SubjectConfirmation element in a SAML assertion.

  Design and configure the application to use the <NotOnOrAfter> condition when using the <SubjectConfirmation> element in a SAML assertion.

  When a SAML assertion is used with a <SubjectConfirmation> element, a begin and end time for the <SubjectConfirmation> should be set to prevent reuse of the message at a later time. Not setting a specific time period for the <SubjectConfirmation>, may grant immediate access to an attacker and result in an immediate loss of confidentiality. [13]

- ✓ V-222405: The application must ensure if a OneTimeUse element is used in an assertion, there is only one of the same used in the Conditions element portion of an assertion.

  Multiple <OneTimeUse> elements used in a SAML assertion can lead to elevation of privileges, if the application does not process SAML assertions correctly.

  When using OneTimeUse elements in a SAML assertion only allow one, OneTimeUse element to be used in the conditions element of a SAML assertion. [13].

Priority: Not described

Rationale: Properly using the NotOnOrAfter and OneTimeUse conditions in the SAML statements is essential to prevent unauthorized access and possible elevations of privileges. These controls limit the opportunity window for attacks and guarantee the integrity and confidentiality of SAML statements.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **NotOnOrAfter Use Verification:**

  1. Confirm that the application is designed and configured to use the NotOnOrAfter condition when using the SubjectConfirmation element in a SAML statement.

- ✓ **Validation of OneTimeUse use:**

  1. Verify that the application guarantees that if a OneTimeUse element is used in a statement, there is only one of the same in the Conditions section of a SAML statement.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda

José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The installed application for auditing doesn't implement SSO, assertions, or SAML modules to be analyzed for the audit team; therefore, it is considered not applicable for the audit (N/a). **Result:** N/a

USE OF SECURITY FLAGS

PUID: [SEC-CAT-SEM-014]
Requirement description: The application must configure session cookies with the following properties: ✓ Establish the HTTPOnly flag in session cookies. ✓ Ensure that the safe flag (Secure) is activated in session cookies.
Source: ✓ **V-222575**: The application must set the HTTPOnly flag on session cookies. Configure the application to set the HTTPOnly flag on session cookies. HTTPOnly is a flag included in a Set-Cookie HTTP response header. If the HTTPOnly flag is included in the HTTP response header, the cookie cannot be accessed through client-side scripts like JavaScript. If the HTTPOnly flag is set, even if a cross-site scripting (XSS) flaw in the application exists, and a user accidentally accesses a link that exploits this flaw, the browser will not reveal the cookie to a third party [13]. ✓ **V-222576**: The application must set the secure flag on session cookies. Configure the application to ensure the secure flag is set on session cookies. Many web development frameworks such as PHP, .NET, ASP as well as application servers include their own mechanisms for session management. Whenever possible it is recommended to utilize the provided session management framework. Setting the secure bit on session cookie ensures the session cookie is only sent via TLS/SSL HTTPS connections. This helps to ensure confidentiality as the session cookie is not able to be viewed by unauthorized parties as it transits the network. Setting the secure flag on all cookies may also be warranted depending upon application design but at a minimum, the session cookie must always be secured. [13] ✓ **A05:2021 – Security Misconfiguration:** Sending security directives to clients, e.g., Security Headers. [3]
Priority: Not described
Rationale: Configuring session cookies by activating HTTPOnly and safe flags is crucial to mitigate risks associated with script attacks between sites (XSS) and guarantee the confidentiality of session information during transmission.
Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **HTTP-only flag:**

  1. Confirm that the application configures all session cookies with the HTTPOnly flag.
  2. Verify that the HTTPOnly flag configuration is correctly reflected in HTTP response headers.

- ✓ **Safe Flag (Secure):**

  1. Ensure the application configures all session cookies with the safe flag (Secure).
  2. Confirm that the safe flag (Secure) configuration is activated in all session cookies.

- ✓ **Compliance with security directives:**

  1. Verify that the application complies with security directives by sending adequate security headers, as described in A05: 2021 - Security Misconfiguration.

Requested by: The organization

Responsible: Developer

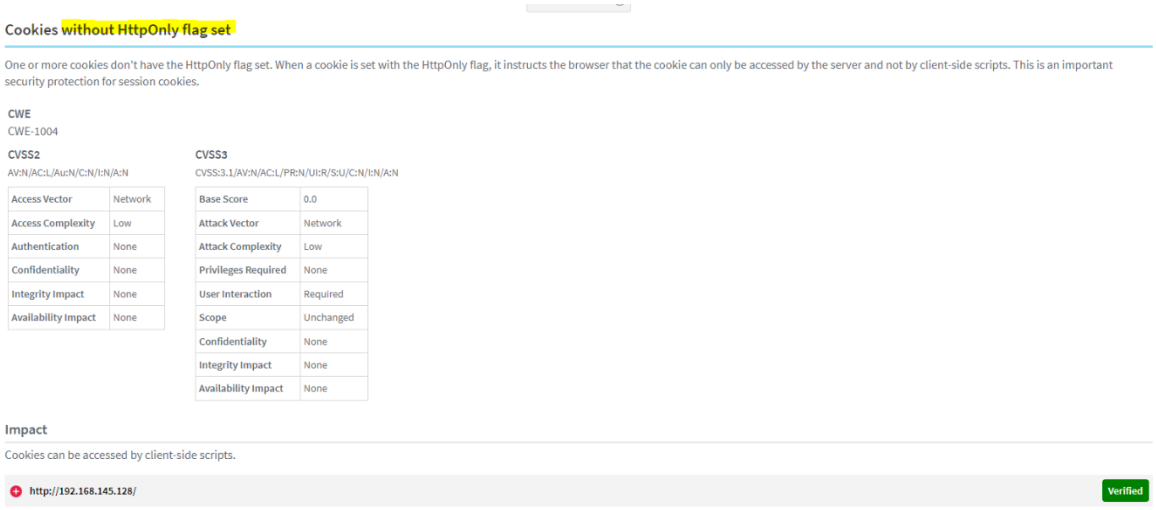
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are findings for “Cookies without HTTPOnly flag set”.



Cookies without HttpOnly flag set

One or more cookies don't have the HttpOnly flag set. When a cookie is set with the HttpOnly flag, it instructs the browser that the cookie can only be accessed by the server and not by client-side scripts. This is an important security protection for session cookies.

CWE
CWE-1004

CVSS2
AV:N/AC:L/Au:N/C:N/I:N/A:N

Access Vector	Network
Access Complexity	Low
Authentication	None
Confidentiality	None
Integrity Impact	None
Availability Impact	None

CVSS3
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:N

Base Score	0.0
Attack Vector	Network
Attack Complexity	Low
Privileges Required	None
User Interaction	Required
Scope	Unchanged
Confidentiality	None
Integrity Impact	None
Availability Impact	None

Impact

Cookies can be accessed by client-side scripts.

http://192.168.145.128/ Verified

Result: It doesn't comply

2.9.3.6 Vulnerable and Outdated Components

Libraries/components update

PUID: [SEC-CAT-VOC-001]

Requirement description: Keep updated on the components or libraries of the application

Source:

- ✓ § 164.308 Administrative safeguards.
  (a) A covered entity or business associate must, in accordance with § 164.306:
  (5)
  (ii) Implementation specifications. Implement:
  (A) Security reminders (Addressable)
  Periodic security updates [10].

- ✓ SA-22 UNSUPPORTED SYSTEM COMPONENTS
  Control:

  a. Replace system components when support for the components is no longer available from the developer, vendor, or manufacturer; or
  b. Provide the following options for alternative sources for continued support for unsupported components [Selection (one or more): in-house support; [Assignment: organization-defined support from external providers]].

  Discussion: Support for system components includes software patches, firmware updates, replacement parts, and maintenance contracts. An example of unsupported components includes when vendors no longer provide critical software patches or product updates, which can result in an opportunity for adversaries to exploit weaknesses in the installed components. Exceptions to replacing unsupported system components include systems that provide critical mission or business capabilities where newer technologies are not available or where the systems are so isolated that installing replacement components is not an option. [11]

- ✓ SI-2 FLAW REMEDIATION

  Control:
  a. Identify, report, and correct system flaws;
  b. Test software and firmware updates related to flaw remediation for effectiveness and potential side effects before installation;
  c. Install security-relevant software and firmware updates within [Assignment: organization-defined time period] of the release of the updates; and
  d. Incorporate flaw remediation into the organizational configuration management process.

  Discussion: The need to remediate system flaws applies to all types of software and firmware. Organizations identify systems affected by software flaws, including potential vulnerabilities resulting from those flaws, and report this information to designated organizational personnel with information security and privacy responsibilities. Security-relevant updates include patches, service packs, and malicious code signatures.

Organizations also address flaws discovered during assessments, continuous monitoring, incident response activities, and system error handling. By incorporating flaw remediation into configuration management processes, required remediation actions can be tracked and verified. [11].

- ✓ **A06:2021 – Vulnerable and Outdated Components**

  - Continuously inventory the versions of both client-side and server-side components (e.g., frameworks, libraries) and their dependencies using tools like versions, OWASP Dependency Check, retire.js, etc. Continuously monitor sources like Common Vulnerability and Exposures (CVE) and National Vulnerability Database (NVD) for vulnerabilities in the components. Use software composition analysis tools to automate the process. Subscribe to email alerts for security vulnerabilities related to components you use.

  - Monitor for libraries and components that are unmaintained or do not create security patches for older versions. If patching is not possible, consider deploying a virtual patch to monitor, detect, or protect against the discovered issue. [3]

- ✓ **V-222614:** Security-relevant software updates and patches must be kept up to date. Check for application updates at least weekly and apply patches immediately or in accordance with POA&Ms, IAVMs, CTOs, DTMs or other authoritative patching guidelines or sources. [13]

- ✓ **V-222658:** All products must be supported by the vendor or the development team. Remove or decommission all unsupported software products in the application.

  Unsupported commercial and government developed software products should not be used because fixes to newly identified bugs will not be implemented by the vendor or development team. The lack of security updates can result in potential vulnerabilities. [13].

- ✓ **A02:2021 – Cryptographic Failures:** Do not use legacy protocols such as FTP and SMTP for transporting sensitive data. [3]

Priority: Medium

Rationale: The effective management of security and components is crucial to maintain the integrity, confidentiality, and availability of the application, as well as to address potential vulnerabilities and guarantee compliance with safety standards.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To_be_determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Verification of security reminders:**

  1. Confirm the implementation of periodic security reminders as security updates.

- ✓ **Non-admitted component replacement evaluation:**

  1. Verify that system components are replaced when the support is no longer available.
  2. Confirm the availability and application of options for alternative support sources.

- ✓ **Failure and updates management evaluation:**

  1. Verify the identification, report, and correction of system failures.
  2. Confirm effective software and firmware update tests.
  3. Validate the timely installation of software and firmware updates related to failure correction.
  4. Verify the incorporation of failure correction in the organizational configuration management process.

- ✓ **Component management evaluation:**

  1. Confirm the implementation of a continuous inventory of component versions.
  2. Validate continuous monitoring of sources of vulnerabilities.
  3. Verify the use of software composition analysis tools.
  4. Confirm the subscription to safety alerts.

- ✓ **Verification of updates and patch maintenance:**

  1. Please confirm the verification of application updates at least weekly.
  2. Validate the timely application of patches according to authoritative guidelines.

- ✓ **Evaluation of development of non-admitted products:**

  1. Confirm the elimination or deactivation of all software products not admitted to the application.

- ✓ **Verification of the use of safe protocols:**

  2. Confirm that inherited protocols such as FTP and SMTP for sensitive data transport are not used.

Requested by: The organization

Responsible: Developer

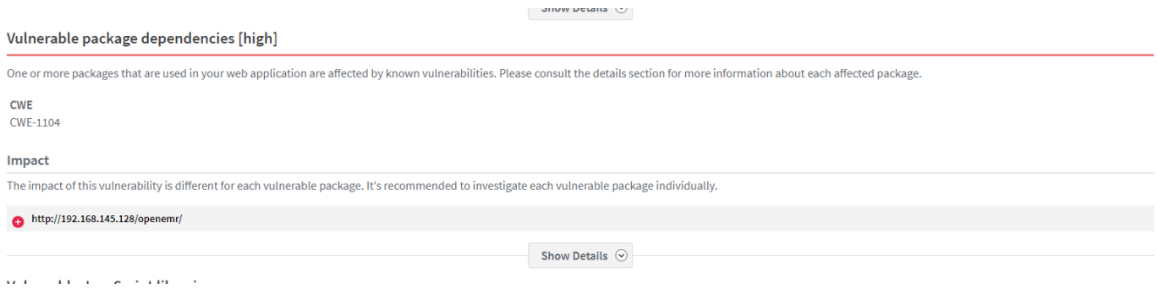
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

- ✓ A DAST and SAST have been performed, and there are findings for “SI-2 FLAW REMEDIATION”.



Result: It doesn't comply

PUID: [SEC-CAT-VOC-002]

Requirement description: The libraries or software components that do not have the support or maintenance must be removed or replaced

Source:

- ✓ **V-222659:** The application must be decommissioned when maintenance or support is no longer available. Ensure there is maintenance for the application.

  Unsupported software products should not be used because fixes to newly identified bugs will not be implemented by the vendor or development team. The lack of security updates can result in potential vulnerabilities.

  When maintenance updates and patches are no longer available, the application is no longer considered supported, and should be decommissioned. [13].

- ✓ **A06:2021 – Vulnerable and Outdated Components:**
  - Remove unused dependencies, unnecessary features, components, files, and documentation. [3]

Priority: Not described

Rationale: Using non-compatible and maintenance software represents a significant safety risk. Software products without support will not receive corrections for possible newly identified safety problems, which can result in potential vulnerabilities. The timely dismantling of the application guarantees mitigating risks associated with the lack of security updates.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

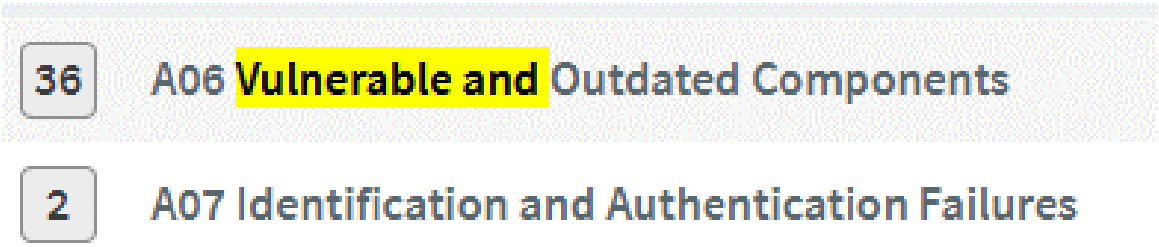
Validation criteria:

- ✓ **Continuous maintenance verification:**

  1. Confirm that the application has a constant maintenance plan.

- ✓ **Removal of unused components:**

  1. Confirm that dependencies, functions, components, archives, and documentation not used from the application have been eliminated.

Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: A DAST and SAST have been performed, and there are findings for "A06:2021 – Vulnerable and Outdated Components".      **Result:** It doesn't comply

PUID: [SEC-CAT-VOC-003]
Requirement description: The application must provide notifications or alerts when product updates and safety-related patches are available.
Source: ✓ **V-222670:** The application must provide notifications or alerts when product update and security related patches are available. Provide a distribution mechanism for obtaining updates to the application. Include a description of the issue, a summary of risk as well as potential mitigations and how to obtain the update. [13].
Priority: Not described
Rationale: This requirement seeks to maintain the updated application with the latest updates and security patches since it is crucial to guarantee integrity and threat resistance. Detailed notifications provide users with the information necessary to address any potential vulnerability proactively.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Update notifications:**

1. Verify that the application issues notifications when there are updates of available products. ✓ **Update distribution:** 1. Confirm that there is an effective mechanism to distribute application updates. ✓ **Notification content:** 1. Validate that notifications include a description of the problem, a summary of risks, possible mitigations, and clear steps to obtain the update.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: When using the auditor equipment, the application did not present messages of possible updates available, even though the official software site already has released the 7.0.2 version. **Result:** It doesn't comply

2.9.3.7 Identification and Authentication Failures

PERMITTED ACTIONS WITHOUT IDENTIFICATION OR AUTHENTICATION

PUID: [SEC-CAT-IAF-001]
Requirement description: The application must identify user actions that can be carried out in the system without requiring identification or authentication.
Source: ✓ AC-14 PERMITTED ACTIONS WITHOUT IDENTIFICATION OR AUTHENTICATION Control: a. Identify [Assignment: organization-defined user actions] that can be performed on the system without identification or authentication consistent with organizational mission and business functions; and b. Document and provide supporting rationale in the security plan for the system, user actions not requiring identification or authentication. [11]
Priority: Not described
Rationale: This requirement seeks to identify and document the user actions that can be carried out without identification or authentication since it is essential to establish limits and safeguard the system's integrity. This measure guarantees a coherent and safe management of user functions.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described

Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Identification of actions without identification or authentication:** **1.** Verify that the application has identified [assignment: user actions defined by the organization] that can be carried out without identification or authentication. ✓ **Documentation and justification:** **1.** Confirm that there is documentation in the system security plan that supports and justifies user actions that do not require identification or authentication.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: A DAST and SAST have been performed, and there are no findings for "AC-14 PERMITTED ACTIONS WITHOUT IDENTIFICATION OR AUTHENTICATION". Additionally, as reviewed by the auditor team, actions that can be carried out without due authentication in the system are not evidenced. AC-14 PERMITTED ACTIONS WITHOUT IDENTIFICATION OR AUTHENTICATION No alerts in this category **Result:** It complies.

USER REGISTRATION

PUID: [SEC-CAT-IAF-002]
Requirement description: All organizations that connect to the Health Electronic Registry Information System (EHRI) must submit to possible users of Systems Point of Sale (POS) that link to EHRI to a formal process of registration of Users
Source: ✓ Security Requirement 53 – Registering Users: All organizations connecting to the EHRi must subject potential users of PoS systems that connect to the EHRi to a formal user-registration process. These user-registration procedures must ensure:

a) The level of user identification that is provided is consistent with the assurance required, given the value of the information assets and the functions that will become available to the user; b) Each potential user has a legitimate relationship with the organization; and c) Each potential user has a legitimate need to access PHI via the EHRi. [12]
Priority: Not described
Rationale: This requirement ensures that only authorized and legitimate users can access sensitive health information. The formal registration process is essential to verify the authenticity and legitimacy of users interacting with the system.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To_be_determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **User identification level verification:** **1.** Confirm that the registration process guarantees the user's identification level is consistent with the required guarantee. ✓ **Validation of the legitimate relationship with the organization:** **1.** Verify that each potential user has a legitimate relationship with the organization that meets the established criteria. ✓ **Confirmation of the legitimate need for access to Phi:** **1.** Validate that each potential user needs to access protected health information (PHI) through EHRI.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The installed application to be audited hasn't implemented "organizations connecting to the EHRi" to be analyzed for the audit team; therefore, it is considered not applicable for the audit (N/a). **Result:** N/a

PUID: [SEC-CAT-IAF-003]
Requirement description: The application, when using PKI-based authentication, must validate certified by building a certification chain (including state information) to an accepted trust anchor,

so the application must be designed to create a certification chain to a Trust Anchor accepted when using PKI -based authentication.

Source:

- ✓ **V-222550:** The application, when utilizing PKI-based authentication, must validate certificates by constructing a certification path (which includes status information) to an accepted trust anchor. Design the application to construct a certification path to an accepted trust anchor when using PKI-based authentication.

  A trust anchor is an authoritative entity represented via a public key and associated data. It is used in the context of public key infrastructures, X.509 digital certificates, and DNSSEC.

  When there is a chain of trust, usually the top entity to be trusted becomes the trust anchor; it can be, for example, a Certification Authority (CA). A certification path starts with the subject certificate and proceeds through a number of intermediate certificates up to a trusted root certificate, typically issued by a trusted CA. [13]

Priority: Not described

Rationale: This requirement tends to properly validate PKI certificates since it is essential to guarantee the authenticity and integrity of safe communications. When building a certification chain to an accepted trust anchor, a confidence structure is established that improves PKI-based authentication safety.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Verification of the construction of the certification chain:**

  1. Confirm that the application can build a certification chain with state information to an accepted trust anchor.

- ✓ **PKI certification evaluation:**

  1. Validate that the application performs the verification of certificates effectively when using PKI-based authentication.

- ✓ **Trust Anchor verification accepted:**

  1. Confirm that the certification chain built by the application reaches an accepted Trust Anchor according to security specifications.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The installed application to be audited hasn't implemented "PKI-based authentication" to be analyzed for the audit team; therefore, it is considered not applicable for the audit (N/a).

Result: N/a

PUID: [SEC-CAT-IAF-004]

Requirement description: The application, when using authentication based on public key infrastructure (PKI), must enforce authorized access to the corresponding private key, so the application or relevant access control mechanism must be configured to guarantee authorized access to the private key (s) of the application.

Source:

- ✓ **V-222551:** The application, when using PKI-based authentication, must enforce authorized access to the corresponding private key. Configure the application or relevant access control mechanism to enforce authorized access to the application private key(s).
  The cornerstone of the PKI is the private key used to encrypt or digitally sign information.

  If the private key is stolen, this will lead to the compromise of the authentication and non-repudiation gained through PKI because the attacker can use the private key to digitally sign documents and pretend to be the authorized user. [13]

Priority: Not described

Rationale: This requirement tends to the security of public key infrastructure (PKI) that depends critically on controlling access to private keys. The loss or commitment of the private key could result in the usurpation of identity and the falsification of digital documents, thus compromising the authenticity and non-repudiation associated with the PKI.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **PKI application:**

  1. Verify that the application uses authentication based on public key infrastructure (PKI).

- ✓ **Private key access control:**

  1. Confirm that the application has mechanisms implemented to enforce authorized access to the corresponding private key.

- ✓ **Configuration of the access control mechanism:**

  1. Validate that the application or relevant access control mechanism is configured to guarantee authorized access to the private key (s).

- ✓ **Loss or theft prevention:**

1. Verify that the security measures implemented prevent the loss or theft of private keys associated with PKI authentication.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The installed application to be audited hasn't implemented "PKI-based authentication" to be analyzed for the audit team; therefore, it is considered not applicable for the audit (N/a). **Result:** N/a

PUID: [SEC-CAT-IAF-005]
Requirement description: The application must implement procedures to verify that the person or entity that seeks access to protected electronic health information is what it claims to be.
Source: ✓ **§ 164.312** Technical safeguards. **(d)** Standard: Person or entity authentication Implement procedures to verify that a person or entity seeking access to electronic protected health information is the one claimed. [10] ✓ Security Requirement 13 – Verifying the Identity of Users All organizations connecting to the EHRi or hosting components of the EHRi must verify the identity and address of each permanent or temporary staff member or contractor who will become a registered user of a PoS system connected to the EHRi or who will have access to hosted components of the EHRi. [12] ✓ **CWE-287: Improper Authentication:** When an actor claims to have a given identity, the product does not prove or insufficiently proves that the claim is correct. This weakness can lead to the exposure of resources or functionality to unintended actors, possibly providing attackers with sensitive information or even execute arbitrary code. [14]
Priority: Not described
Rationale: This requirement seeks to verify identity, help prevent unauthorized access, and protect against possible threats to exposure to sensitive information. The proper authentication of people or entities is essential to guarantee the safety and privacy of protected electronic health information.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To_be_determined
Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Verification of user identity:**

  1. Confirm that the application has procedures implemented to verify the identity of each user who seeks access to protected electronic health information.

- ✓ **Compliance with security requirements:**

  1. Validate that all organizations connected to EHRI, as well as those that house EHRI components, meet the requirement to verify the identity and address of each permanent or temporal member of the personnel or contractor that will become a registered user of a post-connected system to EHRI, or that will have access to components housed from EHRI.

- ✓ **CWE-287 vulnerability mitigation:**

  1. Confirm that the application addresses the vulnerability of inadequate authentication as specified in CWE-287. The application must guarantee that, when an actor claims to have a specific identity, it is properly demonstrated that the statement is correct, thus avoiding the unauthorized exposure of resources or functionalities.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The application requires the user to authenticate before accessing Protected Electronic Health Information.

Result: It complies.

UNSUCCESSFUL LOGON ATTEMPTS

PUID: [SEC-CAT-IAF-006]

Requirement description: The application must:

- ✓ The application must impose a limit of 3 Invalid login attempts by a user during 15 minutes and
- ✓ Automatically block the account for 1 hour; Block the account or node until an administrator releases it; delay the next login notice for 1 hour; Notify the system administrator; Take permanent account blocking when the maximum number of failed attempts is exceeded.

Source:

- ✓ **AC-7 UNSUCCESSFUL LOGON ATTEMPTS**
  Control:
  a. Enforce a limit of [Assignment: organization-defined number] consecutive invalid logon attempts by a user during a [Assignment: organization-defined time period]; and
  b. Automatically [Selection (one or more): lock the account or node for an [Assignment: organization-defined time period]; lock the account or node until released by an administrator; delay next logon prompt per [Assignment: organization-defined delay algorithm]; notify system administrator; take other [Assignment: organization-defined action]] when the maximum number of unsuccessful attempts is exceeded.[11](IE

- ✓ **V-222432**: The application must enforce the limit of three consecutive invalid logon attempts by a user during a 15-minute time period.

  Configure the application to enforce an account lock after 3 failed logon attempts occurring within a 15-minute window. [13].

Priority: Not described

Rationale: This requirement seeks to limit consecutive login consecutive attempts and apply automatic actions in response to exceeding this limit, strengthen application safety, and protect user accounts. This control is essential to mitigate the risk of unauthorized access.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Limit of login attempts:**

  1. Confirm that the application limits three consecutive invalid login attempts per user for 15 minutes.

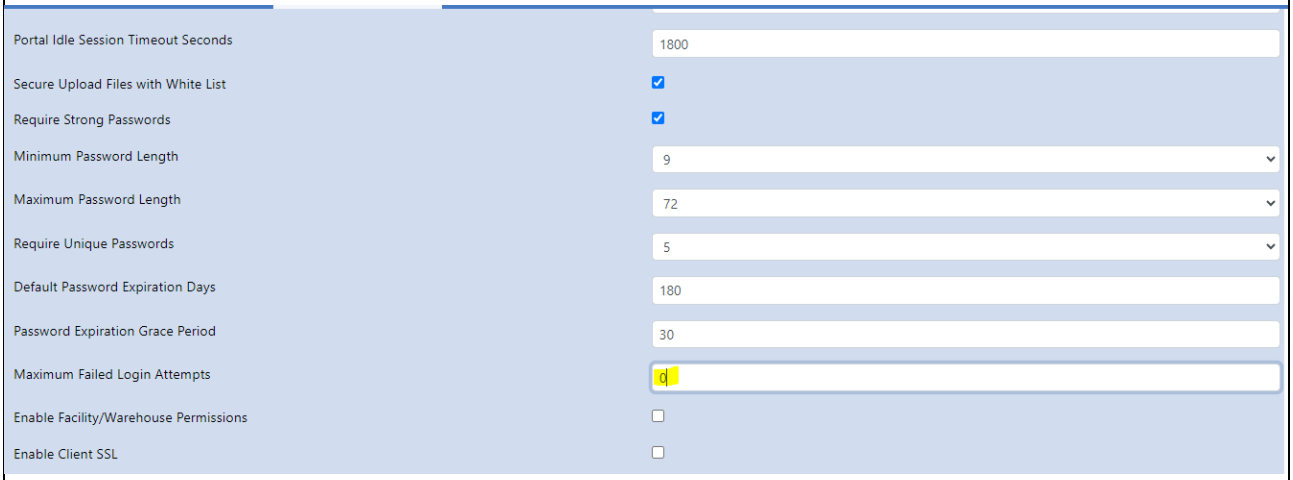
- ✓ **Account block configuration:**

  1. Verify that the application is configured to block the account after three failed login attempts in a 15-minute window.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The auditor team made three login attempts with incorrect passwords, evidenced that a "Limit of login attempts" can be configured from the “Global” option, logged in as an administrator.      **Result:** It complies.

ID ASSIGNMENT FOR SESSIONS

PUID: [SEC-CAT-IAF-007]
Requirement description: The application should generate unique session identifiers using a systemic generation approach to authenticate user sessions. These identifiers must meet the random requirements defined by the organization and should not be reused or recycled.
Source: ✓ § **164.312** Technical safeguards. (a) (2) Implementation specifications: (i) Unique user identification (Required). Assign a unique name and/or number for identifying and tracking user identity. [10] ✓ **SEC. 262**. ADMINISTRATIVE SIMPLIFICATION STANDARDS FOR INFORMATION TRANSACTIONS AND DATA ELEMENTS. ‘‘SEC. 1173. (a) STANDARDS TO ENABLE ELECTRONIC EXCHANGE (b) UNIQUE HEALTH IDENTIFIERS. —

(1) IN GENERAL. —The Secretary shall adopt standards providing for a standard unique health identifier for each individual, employer, health plan, and health care provider for use in the health care system. [9].

- ✓ **Security Requirement 54** – Assigning Identifiers to Users: All organizations connecting to the EHRi must ensure that users of PoS systems that connect to the EHRi are assigned an identifier (User ID) that, in combination with other identifiers (e.g. facility identifiers, jurisdictional identifiers), can uniquely identify the user within the EHRi. PoS systems must support the unique identification of users. [12]

- ✓ **Security Requirement 75** – Uniquely Identifying Patients/Persons: The EHRi and PoS systems connected to the EHRi must:
  a) Ensure that patients/persons are assigned an identifier (patient ID) that can uniquely identify the patient/person within the EHRi or within the PoS system; and
  b) Be capable of merging two or more EHR records if it is determined that multiple records for the same patient/person have been unintentionally created. [12]

- ✓ **Consideration MC17** – Uniquely Identify the Individual Using a Mobile Device: Establish approved mechanisms that will allow the organization to uniquely identify the user of a shared mobile device, prior to granting access to confidential information. [12]

- ✓ **IA-2 IDENTIFICATION AND AUTHENTICATION (ORGANIZATIONAL USERS)**
  Control: Uniquely identify and authenticate organizational users and associate that unique identification with processes acting on behalf of those users. [11].

- ✓ **IA-4 IDENTIFIER MANAGEMENT**
  Control: Manage system identifiers by:
  a. Receiving authorization from [Assignment: organization-defined personnel or roles] to assign an individual, group, role, service, or device identifier;
  b. Selecting an identifier that identifies an individual, group, role, service, or device;
  c. Assigning the identifier to the intended individual, group, role, service, or device; and
  d. Preventing reuse of identifiers for [Assignment: organization-defined time period]. [11]

- ✓ **IA-9 SERVICE IDENTIFICATION AND AUTHENTICATION**

  Control: Uniquely identify and authenticate [Assignment: organization-defined system services and applications] before establishing communications with devices, users, or other services or applications.

  Discussion: Services that may require identification and authentication include web applications using digital certificates or services or applications that query a database. Identification and authentication methods for system services and applications include information or code signing, provenance graphs, and electronic signatures that indicate the sources of services. Decisions regarding the validity of identification and authentication claims can be made by services separate from the services acting on those decisions. This can occur in distributed system architectures. In such situations, the identification and authentication decisions (instead of actual identifiers and authentication data) are provided to the services that need to act on those decisions. [11].

- ✓ **SC-23** SESSION AUTHENTICITY
  Control Enhancements:

(3) SESSION AUTHENTICITY | UNIQUE SYSTEM-GENERATED SESSION IDENTIFIERS

Generate a unique session identifier for each session with [Assignment: organization-defined randomness requirements] and recognize only session identifiers that are system-generated.

Discussion: Generating unique session identifiers curtails the ability of adversaries to reuse previously valid session IDs. Employing the concept of randomness in the generation of unique session identifiers protects against brute-force attacks to determine future session identifiers.
Related Controls: AC-10, SC-12, SC-13. [11].

- ✓ **V-222579:** Applications must use system-generated session identifiers that protect against session fixation. Design the application to generate new session IDs with unique values when authenticating user sessions. [13].

- ✓ **V-222582:** The application must not re-use or recycle session IDs. Design the application to not re-use session IDs.

  Once a user has logged out of the application or had their session terminated, their session IDs should not be re-used. Session IDs should also not be used for other purposes such as creating unique file names and they should also not be re-assigned to other users once the original user has logged out or otherwise quit the application. [13]

Priority: Not described

Rationale: This requirement is intended for the unique identification of sessions since it is essential to guarantee the authenticity and safety of the user sessions, avoiding problems such as setting sessions and improving protection against brute force attacks.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Generation of unique session identifiers:**

  1. Verify that the application uses session identifiers generated by the system to protect against sessions.
  2. Confirm that session identifiers are generated with unique values when authenticating user sessions.

- ✓ **No reuse or recycling of session identifiers:**

  1. Confirm that the application does not reuse or recycle session identifiers.
  2. Validate that once a user has closed or completed a session, session identifiers are not used again for other purposes or assigned to other users.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are no findings for "IA-2 USER IDENTIFICATION AND AUTHENTICATION". Additionally, the Auditor team has shown that the programming framework's methods are used to manage the software sessions.

Result: It complies.

PUID: [SEC-CAT-IAF-008]

Requirement description: The application must implement robust multifactor authentication, using at least two authentication factors (for example, CAC, Alt. Token) to access non-privileged networks and accounts.

Source:

- ✓ 8.5 Secure Authentication
  Control: Secure authentication technologies and procedures should be implemented based on information access restrictions and the topic-specific policy on access control.
  Purpose: To ensure a user or an entity is securely authenticated, when access to systems, applications and services. Is granted.
  Guidance: A suitable authentication technique should be chosen to substantiate the claimed identity of a user, software, messages and other entities. [7]

- ✓ **Security Requirement 70** – Robustly Authenticating Users: The EHRi and all PoS systems connected to the EHRi must robustly authenticate users. [12]

- ✓ **A07:2021 – Identification and Authentication Failures.**

  - Where possible, implement multi-factor authentication to prevent automated credential stuffing, brute force, and stolen credential reuse attacks. [3]

- ✓ **V-222526:** The application must use multifactor (e.g., CAC, Alt. Token) authentication for network access to non-privileged accounts. Configure the application to require CAC or Alt. Token authentication for non-privileged network access to non-privileged accounts.

  Multifactor authentication uses two or more factors to achieve authentication.

  Factors include:

  - (i) Something you know (e.g., password/PIN);

- (ii) Something you have (e.g., cryptographic identification device, CAC/SIPRNet token); or
- (iii) Something you are (e.g., biometric). [13]

✓ **V-222528:** The application must use multifactor (e.g., CAC, Alt. Token) authentication for local access to non-privileged accounts. Configure the application to require CAC or Alt. Token authentication for non-privileged network access.

To assure accountability, prevent unauthenticated access, and prevent misuse of the system, privileged users must utilize multifactor authentication for local access.

Multifactor authentication is defined as: using two or more factors to achieve authentication.

Factors include:
- (i) Something a user knows (e.g., password/PIN);
- (ii) Something a user has (e.g., cryptographic identification device, token); or
- (iii) Something a user is (e.g., biometric).

A non-privileged account is defined as an information system account with authorizations of a regular or non-privileged user. [13]

Priority: Not described

Rationale: Multifactor authentication is essential to strengthen access security, especially for non-non-privileged accounts. Using multiple factors, such as something that the user knows (password), something that the user has (cryptographic identification device, cac/alt. Token), or something that the user is (biometric), provides an additional layer of protection against automated attacks and reuse of stolen credentials.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

✓ **Multifactor authentication for access to networks (V-222526):**

1. Verify that the application requires multifactor authentication (for example, CAC, Alt. Token) for access to networks in non-privileged accounts.

✓ **Multifactor authentication for local access (V-222528):**

1. Confirm that the application requires multifactor authentication (for example, CAC, Alt. Token) for local access in non-privileged accounts.

✓ **Multifactor authentication implementation (A07: 2021):**

1. Validate that, whenever possible, multifactor authentication has been implemented to prevent automated attacks such as credentials, gross force attacks, and reuse of stolen credentials.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: When any user tries to log in, a time-based one-time password (TOTP) and a token expiration period are requested before conferring access. TOTP Verification Provide TOTP code Enter the code from your authentication application on your device: ✓ Authenticate TOTP **Result:** It complies.

PUID: [SEC-CAT-IAF-009]
Requirement description: The application should allow users to use a single identity credential (unique login), whether patients or suppliers when the portal provides access to other applications or services with separate authentication mechanisms.
Source: ✓ Consideration CH27 – Use a single identity credential where possible: Consider the use of a single identity (i.e. single sign-on) for users (i.e. patients or providers) where the portal provides access to other applications/services that have separate authentication mechanisms. [12]
Priority: Not described
Rationale: Implementing a single identity credential simplifies the user experience by providing a single login, improving efficiency and reducing the complexity associated with multiple authentications.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Compatibility with a single login (SSO):** 1. Verify that the application allows patients and suppliers to use a single identity credential to access the portal and other applications or services with different authentication mechanisms. ✓ **Integration with separate authentication mechanisms:**

1. Confirm that the application facilitates access to other applications or services with separate authentication using the same single identity credential.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: OpenEMR in your documentation states that you have compatibility to implement SSO or SAML modules; Therefore, it is considered to meet the SSO integration option. **Result:** It complies

IDENTITY VALIDATION

PUID: [SEC-CAT-IAF-010]
Requirement description: The application must consider and select robust levels of identity assurance following the roles of the users, defining the necessary mechanisms to prove the identity of the record initially.
Source: ✓ **Consideration CH26** – Select identity-proofing level of assurance by role: Consider what robust mechanism(s) should be used to initially prove the identity of the registrant and whether that level of assurance is required and/or adequate for all roles accessing the solution. [12] ✓ **V-222580:** Applications must validate session identifiers. Many web development frameworks such as PHP, .NET, and ASP include their own mechanisms for session management. Whenever possible it is recommended to utilize the provided session management framework. [13]
Priority: Not described
Rationale: The choice of identity assurance levels adapted to user roles is essential to ensure that authentication mechanisms are adequate and robust. This contributes to greater security by aligning the identification requirements with the specific needs of each role.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Appropriate selection of assurance levels:**

1. Confirm that the application has considered and selected adequate identity assurance levels for each user role.
2. Validate that the mechanisms chosen to prove the initial identity are robust and meet the established security requirements.

✓ **Implementation of authentication mechanisms:**

1. Confirm that the application has implemented the necessary mechanisms to ensure the initial identity of the registers according to the defined levels.

✓ **Evaluation Of Compliance with CH26:**

1. Verify that the application has effectively considered and evaluated identity assurance mechanisms depending on the roles of users.

✓ **Validation of compliance with V-222580:**

1. Confirm that the application effectively validates session identifiers using the session management mechanisms recommended by web development frameworks, such as PHP, .net, and ASP.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The Auditor team has shown that the recommended programming framework's methods are used to manage the software sessions.

Result: It complies.

SESSION AUTHENTICITY AND INTEGRITY

PUID: [SEC-CAT-IAF-011]

Requirement description: Ensure the communication session remains authentic and complete from the beginning to the conclusion.

Source:

✓ **SC-23 SESSION AUTHENTICITY**

Control: Protect the authenticity of communications sessions.

Discussion: Protecting session authenticity addresses communications protection at the session level, not at the packet level. Such protection establishes grounds for confidence at both ends of communications sessions in the ongoing identities of other parties and the validity of transmitted information. Authenticity protection includes protecting against

"man-in-the-middle" attacks, session hijacking, and the insertion of false information into sessions [11].

- ✓ **V-222577**: The application must not expose session IDs. Configure the application to protect session IDs from interception or from manipulation.

  Authenticity protection provides protection against man-in-the-middle attacks/session hijacking and the insertion of false information into sessions.

  Application communication sessions are protected utilizing transport encryption protocols, such as SSL or TLS. SSL/TLS provides web applications with a means to be able to authenticate user sessions and encrypt application traffic. Session authentication can be single (one-way) or mutual (two-way) in nature. Single authentication authenticates the server for the client, whereas mutual authentication provides a means for both the client and the server to authenticate each other.

  This requirement applies to applications that utilize communications sessions. This includes, but is not limited to, web-based applications and Service-Oriented Architectures (SOA).

  This requirement addresses communications protection at the application session, versus the network packet, and establishes grounds for confidence at both ends of communications sessions in ongoing identities of other parties and in the validity of information transmitted. Depending on the required degree of confidentiality and integrity, web services/SOA will require the use of SSL/TLS mutual authentication (two-way/bidirectional). [13]

- ✓ No unauthorized reading, manipulation, copying, or deletion during transmission of data [16]

Priority: Not described

Rationale: Protecting the authenticity of communication sessions is essential to guarantee confidence in the parties' identities and the validity of the information transmitted. This addresses threats such as "man in the middle" attacks, kidnapping of sessions, and insertion of false information in the sessions.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria: The application must comply with the following criteria to guarantee the effective protection of the authenticity of the communication sessions:

- ✓ **No Session IDS Exhibition:**

  1. Verify that the application does not expose session IDs.
  2. Confirm that the application is configured to protect session IDs against interception or handling.

- ✓ **Use of encryption protocols:**

  1. Validate transport encryption protocols, such as SSL or TLS, to protect that application communication session.

2. Confirm that session authentication can be unique (unidirectional) or mutual (bidirectional) as necessary.

✓ **Applicability to different types of applications:**

1. Confirm that this requirement applies to applications that use communication sessions, including web applications and service-oriented architectures (SOA).

✓ **Guarantee of confidentiality and integrity:**

1. Verify that the protection of the authenticity of the sessions is carried out at the application level and establishes confidence in the identities of the parties and the validity of the transmitted information.
2. Evaluate whether web and SOA applications that require a high degree of confidentiality and integrity implement mutual authentication SSL/TLS.

✓ **Prevention of readings, manipulations, or unauthorized eliminations:**

1. Confirm that no readings, manipulations, copies, or eliminations are authorized during data transmission.

Requested by: The organization

Responsible: Developer

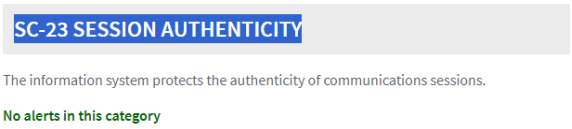
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and no findings exist for "SC-23 SESSION AUTHENTICITY". Additionally, the Auditor team has shown that the application has options for implementing SSL certificates.



Result: It complies.

USER AUTHENTICATION

PUID: [SEC-CAT-IAF-012]

Requirement description: The application must implement safe login procedures that protect against login attempts by brute force.

Source:

✓ **9.4.2 Secure log-on procedures**:

e) protect against brute force log-on attempts. [8]

✓ **A07:2021 – Identification and Authentication Failures**
- Limit or increasingly delay failed login attempts, but be careful not to create a denial-of-service scenario. Log all failures and alert administrators when credential stuffing, brute force, or other attacks are detected. [3].

Priority: Not described

Rationale: Protecting login attempts by brute force is crucial to prevent unauthorized access and guarantee the safety of user credentials. It helps mitigate risks related to credential attacks, such as credential fill attacks or gross force attacks.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

✓ **Limitation or delay of fallen login attempts:**

1. Verify that the application progressively limits or delays the filled login attempts.

✓ **Registration and alert of failed attempts:**

1. Confirm that the application registers all failed login attempts.
2. Verify that administrators receive alerts when attacks such as credentials, brute force, or others are detected.

✓ **Avoid service denial scenarios:**

1. Validate that limitation measures or delay of fallen login attempts do not create a service denial scenario.

Requested by: The organization

Responsible: Developer

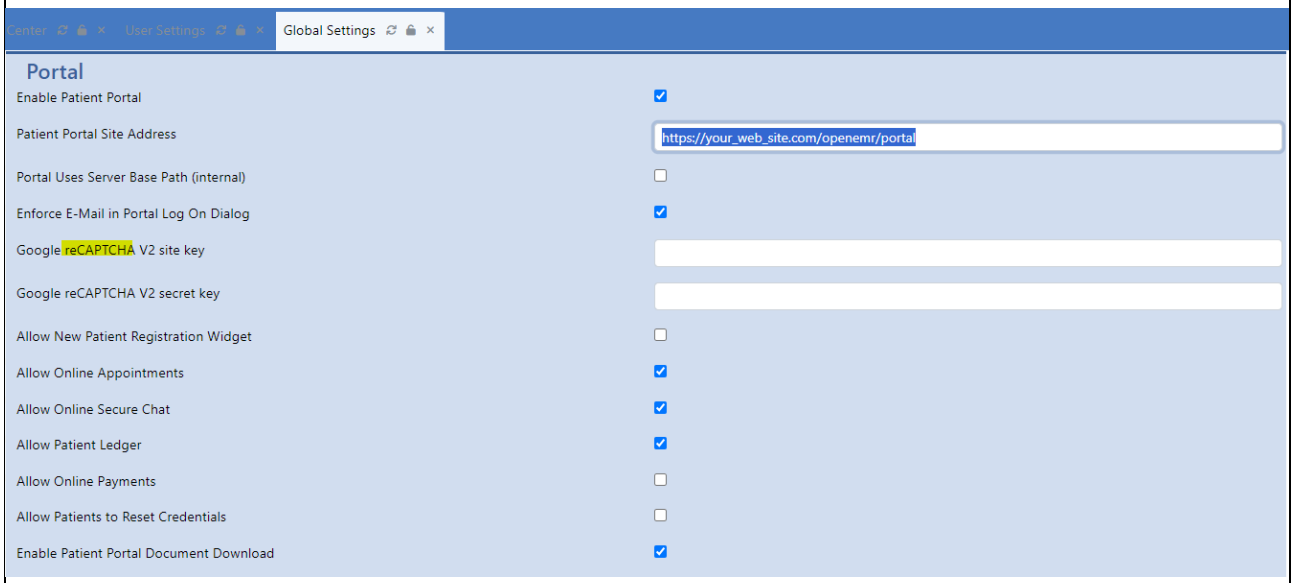
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and some findings exist for "A07 Identification and Authentication Failures". Additionally, the auditor team verified that the application could be configured to implement a CAPTCHA mechanism to avoid DDOS attacks.



Result: It complies.

PUID: [SEC-CAT-IAF-013]

Requirement description: The application should consider that by authenticating the user, at least the two factors of the authentication mechanism are implemented.

Source:

- ✓ IA-10 ADAPTIVE AUTHENTICATION
  Control: Require individuals accessing the system to employ [Assignment: organization-defined supplemental authentication techniques or mechanisms] under specific [Assignment: organization-defined circumstances or situations].

  Discussion: Adversaries may compromise individual authentication mechanisms employed by organizations and subsequently attempt to impersonate legitimate users. To address this threat, organizations may employ specific techniques or mechanisms and establish protocols to assess suspicious behavior. Suspicious behavior may include accessing information that individuals do not typically access as part of their duties, roles, or responsibilities; accessing greater quantities of information than individuals would routinely access; or attempting to access information from suspicious network addresses.

  When pre-established conditions or triggers occur, organizations can require individuals to provide additional authentication information. Another potential use for adaptive authentication is to increase the strength of mechanism based on the number or types of records being accessed. Adaptive authentication does not replace and is not used to avoid the use of multi-factor authentication mechanisms but can augment implementations of multi-factor authentication. [11]

- ✓ 9.4.1 Information access restriction
  Health information systems processing personal health information shall authenticate users and should do so by means of authentication involving at least two factors.
  Access to information and application system functions related to the processing personal health information should be isolated from (and separate to) access to information

processing infrastructure that is unrelated to the processing of personal health information. [8]

Priority: Not described

Rationale: Given the possibility that adversaries compromise individual authentication mechanisms, it is essential to implement supplementary authentication mechanisms in specific conditions or situations. It helps mitigate the risk of supplanting legitimate users when evaluating suspicious behaviors, such as unusual access or access attempts from Suspicious Red Directorates.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Verification of the implementation of supplementary authentication techniques or mechanisms:**

  1. Confirm that the application allows supplementary authentication techniques or mechanisms defined by the organization.

- ✓ **Evaluation of specific conditions or situations:**

  1. Verify that the application requires supplementary authentication in specific circumstances or situations the organization establishes.

- ✓ **Following the information restriction requirement:**

  1. Confirm the authentic system to users through at least two factors, as described in requirement 9.4.1 of access restriction to information.

Requested by: The organization

Responsible: Developer

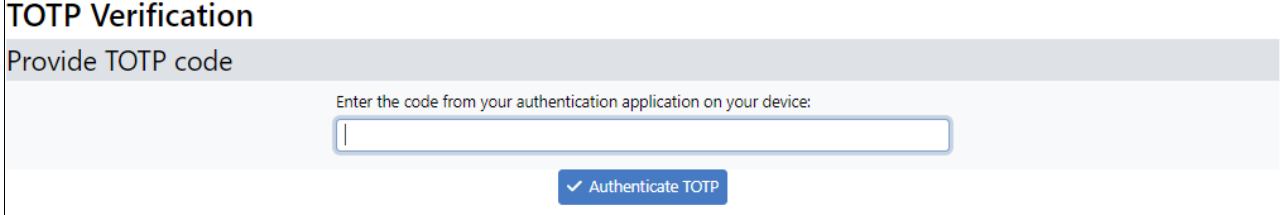
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

When any user tries to log in, a time-based one-time password (TOTP) and a token expiration period are requested before conferring access.



Result: It complies.

PUID: [SEC-CAT-IAF-014]

Requirement description: The application must require the re-authentication of users and devices when information systems' authenticators, roles, or security categories change.

Source:

- ✓ **IA-11 RE-AUTHENTICATION**

  Control: Require users to re-authenticate when [Assignment: organization-defined circumstances or situations requiring re-authentication].

  Discussion: In addition to the re-authentication requirements associated with device locks, organizations may require re-authentication of individuals in certain situations, including when roles, authenticators or credentials change, when security categories of systems change, when the execution of privileged functions occurs, after a fixed time period, or periodically. [11]

- ✓ **V-222520:** The application must require users to reauthenticate when organization-defined circumstances or situations require reauthentication. Configure the application to require reauthentication before user privilege is escalated and user roles are changed.

  When applications provide the capability to change security roles or escalate the functional capability of the application, it is critical the user reauthenticate.

  In addition to the reauthentication requirements associated with session locks, organizations may require reauthentication of individuals and/or devices in other situations, including (but not limited to) the following circumstances:

  - (i) When authenticators change;
  - (ii) When roles change;
  - (iii) When security categories of information systems change;
  - (iv) When the execution of privileged functions occurs;
  - (v) After a fixed period of time;
  - or
  - (vi) Periodically.

- ✓ **V-222521:** The application must require devices to reauthenticate when organization-defined circumstances or situations requiring reauthentication. Configure the application to require reauthentication periodically.

  Without reauthenticating devices, unidentified or unknown devices may be introduced; thereby facilitating malicious activity.

  In addition to the reauthentication requirements associated with session locks, organizations may require reauthentication of devices, including (but not limited to), the following other situations:

  - When authenticators change;
  - (ii) When roles change;
  - (iii) When security categories of information systems change;
  - (iv) After a fixed period of time;

  Or

○ (v) Periodically. Gateways and SOA applications are examples of where this requirement would apply. [13]
Priority: Not described
Rationale: The periodic re-authentication of users and devices is essential to maintain an adequate security level. It ensures that new authentication is carried out in specific or changing situations to prevent unauthorized access or the introduction of unidentified devices.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **User re-authentication:** **1.** Confirm that the application requires user re-authentication according to the circumstances defined by the organization. **2.** Verify that the application is configured to require re-authentication before escalating privileges and changes in user roles. ✓ **Device re-authentication:** **1.** Confirm that the application requires periodic re-authentication of devices according to the circumstances defined by the organization. **2.** Verify that the application is configured to require device re-authentication in situations such as changes in authenticators, roles or categories of information systems, and execution of privileged functions after a fixed period or periodically.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The auditor's team changed a user with the role "Clinicians" to an “Accounting", and re-authentication was not requested. **Result:** It doesn’t comply.

PUID: [SEC-CAT-IAF-015]
Requirement description: In case of an error during the login, the application should not offer specific help messages that can benefit an unauthorized user.
Source:

- ✓ **9.4.2 Secure log-on procedures:**
  c) not provide help messages during the log-on procedure that would aid an unauthorized user. [8]
- ✓ **IA-6 AUTHENTICATION FEEDBACK**

Control: Obscure feedback of authentication information during the authentication process to protect the information from possible exploitation and use by unauthorized individuals.

Discussion: Authentication feedback from systems does not provide information that would allow unauthorized individuals to compromise authentication mechanisms. For some types of systems, such as desktops or notebooks with relatively large monitors, the threat (referred to as shoulder surfing) may be significant. For other types of systems, such as mobile devices with small displays, the threat may be less significant and is balanced against the increased likelihood of typographic input errors due to small keyboards. Thus, the means for obscuring authentication feedback is selected accordingly. Obscuring authentication feedback includes displaying asterisks when users type passwords into input devices or displaying feedback for a very limited time before obscuring it. [11].

Priority: Not described

Rationale: The omission of aid messages during the login is crucial to prevent unauthorized users from obtaining information that facilitates avoiding authentication mechanisms. This reinforces the safety of the authentication process.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Verification of the absence of aid messages:**

  1. Confirm that the application does not provide help messages during the login procedure.

- ✓ **Evaluation of the authentication feedback process:**

  1. Validate that authentication feedback is obscured during the authentication process to protect information against possible farms and use by unauthorized individuals.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

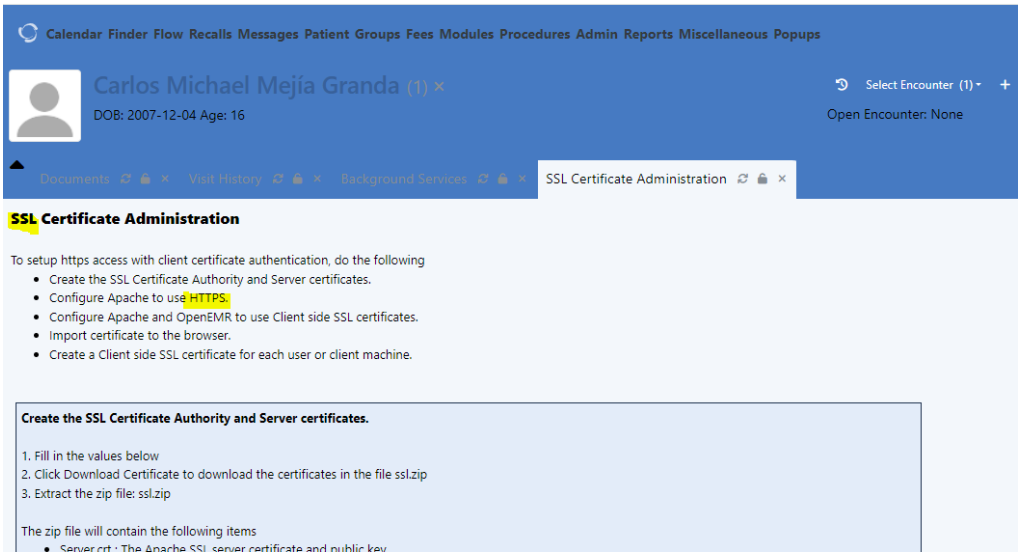
The auditor's team verified the absence of aid messages when the password or username was incorrect, and there was no presence of inadequate information presented to the user.

Result: It complies

PUID: [SEC-CAT-IAF-016]
Requirement description: The application should not show system or application identifiers until the login process has been successfully completed.
Source: ✓ **9.4.2 Secure log-on procedures:** a) not display system or application identifiers until the log-on process has been successfully completed. [8]
Priority: Not described
Rationale: Login security is essential to prevent the exposure of sensitive information. Hiding system or application identifiers until the login process has been completed reduces the risk of unauthorized dissemination and protects the system's integrity.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To_be_determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Identifiers concealment verification:** **1.** Confirm that the application does not show system or application identifiers before completing the login process. **2.** Validate that identifiers visualization occurs only after a successful login.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The auditor's team verified the absence of system or application identifiers before completing the login process. **Result:** It complies

PUID: [SEC-CAT-IAF-017]
Requirement description: The application should not allow passwords to be transmitted in clear text on a network, following section 9.4.2 J) of safe login procedures.
Source: ✓ **§ 164.308** Administrative safeguards.

(a) A covered entity or business associate must, in accordance with § 164.306: **(5)** **(ii)** Implementation specifications. Implement: **(D)** Password management (Addressable) Procedures for creating, changing, and safeguarding passwords [10]. ✓ **9.4.2** Secure log-on procedures: j) not transmit passwords in clear text over a network. [8] ✓ **V-222396:** The application must implement DoD-approved encryption to protect the confidentiality of remote access sessions. Design and configure applications to use TLS encryption to protect the confidentiality of remote access sessions. [13].
Priority: Not described
Rationale: Safe password management is essential to protect access to applications and sensitive data. In addition, the encryption of remote sessions guarantees confidentiality during distance connections.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Safe password transmission:** **1.** Confirm that passwords are not transmitted in clear text on the network during authentication. ✓ **Remote sessions encryption:** **1.** Validate that the application uses TLS to encrypt remote access sessions.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: A DAST has been performed, and there are findings for “User credentials are sent in clear text”. However, after analyzing the platform and logging on as an administrator user, the auditor team found an option to configure communications with SSL certificates and implement HTTPS.



Result: It complies

PUID: [SEC-CAT-IAF-018]

Requirement description: During the login procedures, the application should not show the password that is being entered.

Source:

- ✓ **§ 164.308** Administrative safeguards.
  (a) A covered entity or business associate must, in accordance with § 164.306:
  (5)
  (ii) Implementation specifications. Implement:
  (D) Password management (Addressable)
  Procedures for creating, changing, and safeguarding passwords [10].
- ✓ **9.4.2** Secure log-on procedures:
  i) not display a password being entered; [8]
- ✓ **V-222554**: The application must not display passwords/PINs as clear text:
  Configure the application to obfuscate passwords and PINs when they are being entered so they cannot be read.

  Design the application so obfuscated passwords cannot be copied and then pasted as clear text. [13].

Priority: Not described

Rationale: Sailing password management and non-visualization of passwords during the login are crucial practices to guarantee user information security. These measures prevent unauthorized access and protect the confidentiality of access credentials.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Safe login:**

  1. Validate that the application does not show the password that is being entered during the login procedures, according to the specification of 9.4.2 Secure log-on procedures.

- ✓ **Obfuscation of passwords:**

  1. Please verify that the application is configured to obfuscate passwords and pins during admission, preventing them from being legible.

- ✓ **Copy prevention of obfuscated passwords:**

  1. Confirm that application design prevents obfuscated passwords from being copied and glued as clear text.

Requested by: The organization

Responsible: Developer

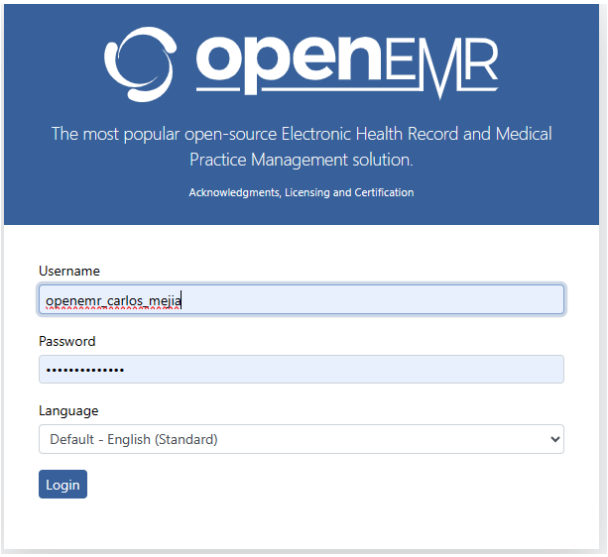
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

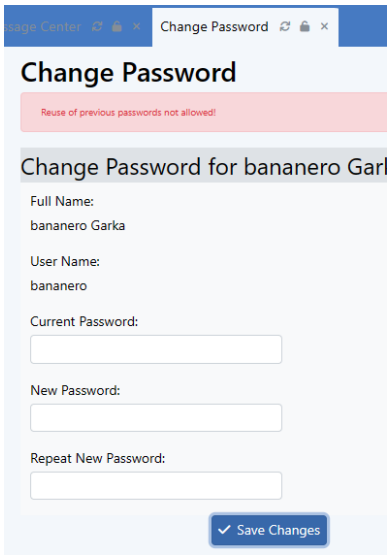
When users type their password in the application, it is not shown while entering because asterisks replace it.



Result: It complies

PUID: [SEC-CAT-IAF-019]

Requirement description: The application must prohibit password reuse for a minimum of five generations
Source: ✓ **V-222546:** The application must prohibit password reuse for a minimum of five generations. Configure the application to prohibit password reuse for up to 5 passwords. [13].
Priority: Not described
Rationale: The prohibition of password reuse contributes to security by preventing users from resorting to old passwords, thus reducing the risk of unauthorized access and strengthening authentication measures.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Prohibition of password reuse:** **1.** Verify that the application is configured to prohibit password reuse for a minimum of five generations. **2.** Confirm that the current configuration prevents users from using the last five previously used passwords.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: Logged in the system as a standard user when the password is changed, the application prohibits password reuse for a minimum of five generations.



Result: It complies

PUID: [SEC-CAT-IAF-020]

Requirement description: The application should allow the use of a temporary password for the session in the system, with an immediate change to a permanent password

Source:

- ✓ **V-222547**: The application must allow the use of a temporary password for system logons with an immediate change to a permanent password. Configure the application to specify when a password is temporary and change the temporary password on the first use.

  Temporary passwords are typically used to allow access to applications when new accounts are created or passwords are changed. It is common practice for administrators to create temporary passwords for user accounts which allow the users to log on, yet force them to change the password once they have successfully authenticated. [13]

Priority: Not described

Rationale: Implementing temporary passwords with an immediate change to permanent improves security by allowing provisional access to applications when creating new accounts or password changes. This common practice facilitates username administration and guarantees rapid password updates.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To_be_determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Temporary password use:**

  1. Verify that the application allows the use of temporary passwords for sessions.
  2. Confirm that the application specifies when a password is temporary.

- ✓ **Immediate change to permanent password:**

  1. Confirm that the application is configured to change the temporary password in the first use after successful authentication.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The administrator user logs in to the system and creates a new user, and then the administrator must type the new user password. Finally, when the new user authenticates on the system, the application doesn't enforce changing passwords.

Result: It doesn't comply

PUID: [SEC-CAT-IAF-021]

Requirement description: The application must, when a successful login is completed, show the following information:

1. Date and time of the last successful login.
2. Details of any login and login attempts since the last successful session.

Source:

- ✓ **9.4.2 Secure log-on procedures**: h) display the following information on completion of a successful log-on:
  1) date and time of the previous successful log-on;
  2) details of any unsuccessful log-on attempts since the last successful log-on. [8]

- ✓ **AC-9 PREVIOUS LOGON NOTIFICATION**
  Control: Notify the user, upon successful logon to the system, of the date and time of the last logon. [11]

- ✓ **V-222437:** The application must display the time and date of the users last successful logon. Design and configure the application to display the date and time when the user was last successfully granted access to the application.

  Providing a last successful logon date and time stamp notification to the user when they authenticate and access the application allows the user to determine if their application account has been used without their knowledge. [13]

Priority: Not described

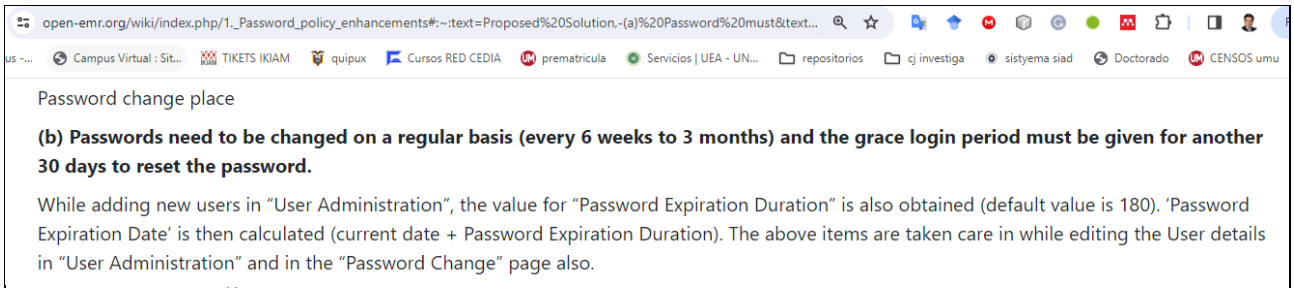
Rationale: Detailed information visualization about the last successful login and failed attempts improves security by allowing users to monitor the activity of their accounts and detect possible unauthorized accesses.

Child PUIDs: Not described

Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Previous login notification:** **1.** Confirm that, when making a successful login, the application notifies the user of the date and time of the last login. ✓ **Falling attempts:** **1.** Verify that the application shows details of any login attempts since the last successful login. ✓ **AC-9 compliance:** **1.** Validate that the application meets AC-9 control, notifying the user of the date and time of the last login during a successful login. ✓ **V-222437 implementation:** **1.** Confirm that the application is designed and configured to display the date and time of the last successful login, providing users with essential information to detect unauthorized access.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: After several failed attempts to access the system as a standard user, the auditor team entered the correct username and password. The Home screen is presented, but the failed authenticating attempts are not detailed. **Result:** It doesn't comply

PUID: [SEC-CAT-IAF-022]
Requirement description: The application must apply a minimum 24 -hour/1 day password life policy.
Source: ✓ **V-222544**: The application must enforce 24 hours/1 day as the minimum password lifetime.

Configure the application to have a minimum password lifetime of 24 hours. Enforcing a minimum password lifetime helps prevent repeated password changes to defeat the password reuse or history enforcement requirement. Restricting this setting limits the user's ability to change their password. Passwords need to be changed at specific policy-based intervals; however, if the application allows the user to immediately and continually change their password, then the password could be repeatedly changed in a short period of time to defeat the organization's policy regarding password reuse. [13].
Priority: Not described
Rationale: Imposing a minimum password period contributes to avoiding repeated password changes to avoid reuse requirements or password history. Restricting this adjustment limits the user's ability to change their password immediately and continuously, thus guaranteeing compliance with the organization's policies.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Configuration of the Minimum Password Life policy:** **1.** Confirm that the application is configured to have a minimum 24-hour password life. ✓ **Repeated change prevention:** **1.** Verify that the application effectively prevents repeated password changes in a short period to avoid password reuse policy.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: As Open-Emr.org's wiki page, the password change is suggested at least every six weeks.



Result: It complies

PUID: [SEC-CAT-IAF-023]

Requirement description: The application must apply a maximum 60-day password restriction

Source:

- ✓ **V-222545**: The application must enforce a 60-day maximum password lifetime restriction. Configure the application to have a maximum password lifetime of 60 days.

  Any password, no matter how complex, can eventually be cracked. Therefore, passwords need to be changed at specific intervals.

  One method of minimizing this risk is to use complex passwords and periodically change them. If the application does not limit the lifetime of passwords and force users to change their passwords, there is the risk that the system and/or application passwords could be compromised. [13].

Priority: Not described

Rationale: A maximum duration limit on the password is essential to mitigate the risk of password commitment. Changing passwords periodically reduces the window of opportunity for possible attacks, even when complex passwords are used.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To_be_determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Verification of password restriction:**

  1. Confirm that the application imposes a maximum 60-day password restriction.

- ✓ **Password configuration:**

  1. Validate that the application is correctly configured to have a maximum password duration of 60 days.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

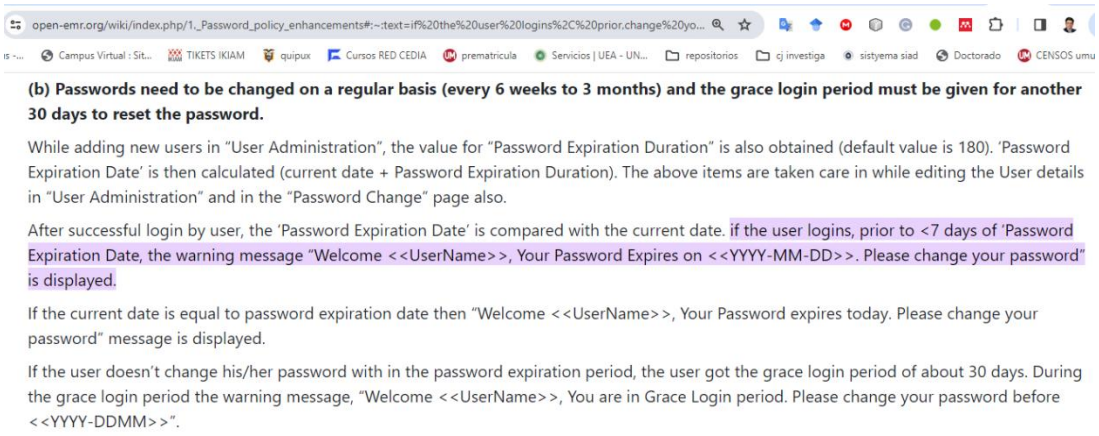
Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán

Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

As Open-Emr.org's wiki page, the maximum 60-day password restriction could be configured.



Result: It complies

PUID: [SEC-CAT-IAF-024]

Requirement description: The application must ensure that changing the password requires changing at least eight characters.

Source:

- ✓ **§ 164.308** Administrative safeguards.
  (a) A covered entity or business associate must, in accordance with § 164.306:
  (5)
  (ii) Implementation specifications. Implement:
  (D) Password management (Addressable)
  Procedures for creating, changing, and safeguarding passwords [10].

- ✓ **V-222541**: The application must require the change of at least 8 of the total number of characters when passwords are changed [13].

Priority: Not described

Rationale: Establish a robust and secure access control and authentication mechanism within the system

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To_be_determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Verification of password change:**

  1. Confirm that the application requires the change of at least eight characters of the total when password changes are made.

- ✓ **Evaluation of password management procedures:**

    1. Review the implementation of procedures for creating, modifying, and protecting passwords, ensuring that they comply with the specifications established by the regulatory entity.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The auditor team entered the "S3ptums3mpr@*" password for logging into the system as a standard user. The change was allowed after accessing the password change module and using the "S3ptums3mpr**" key. Consequently, the requirement is not met.

Result: It doesn't comply

PASSWORD LENGTH AND COMPLEXITY

PUID: [SEC-CAT-IAF-025]

Requirement description: The application must comply with the following password complexity policies:

1. Require at least a special character.
2. Apply a minimal password length of 15 characters.
3. Demand at least one capital character.
4. Make at least a lowercase character.
5. Impose password complexity by requiring at least a numerical character.

Source:

- ✓ **V-222540**: The application must enforce password complexity by requiring that at least one special character be used [13].
- ✓ **V-222536**: The application must enforce a 15-character password length. [13]
- ✓ **V-222537**: The application must enforce password complexity by requiring that at least one upper-case character be used. [13]
- ✓ **V-222538**: The application must enforce password complexity by requiring that at least one lower-case character be used [13].
- ✓ **V-222539**: The application must enforce password complexity by requiring that at least one numeric character be used [13].
- ✓ **A07:2021 – Identification and Authentication Failures**:
    - Implement weak password checks, such as testing new or changed passwords against the top 10,000 worst passwords list.
    - Align password length, complexity, and rotation policies with National Institute of Standards and Technology (NIST) 800-63b's guidelines in section 5.1.1 for Memorized Secrets or other modern, evidence-based password policies. [3]

Priority: Not described

Rationale: Implementing safe password policies is essential to strengthen authentication and protect user accounts against possible attacks. These measures reduce the risk of weak passwords and improve the global security of the system.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Verification of password complexity requirements:**

  1. Confirm that the application requires at least a special password.
  2. Validate that the minimum password length is 15 characters.
  3. Confirm that at least a capital letter is required.
  4. Verify that at least a lowercase character is mandatory.
  5. and. Ensure that password complexity is imposed by requiring at least a numerical character.

- ✓ **Evaluation of password control implementation:**

  1. Confirm that weak password checks are applied, such as testing new passwords or passwords against the list of the 10,000 most vulnerable passwords.
  2. Align the policies of length, complexity, and rotation of passwords with the guidelines of the National Institute of Standards and Technology (NIST) 800-63B, Section 5.1.1 for memorized secrets or other modern password policies based on evidence.

Requested by: The organization

Responsible: Developer

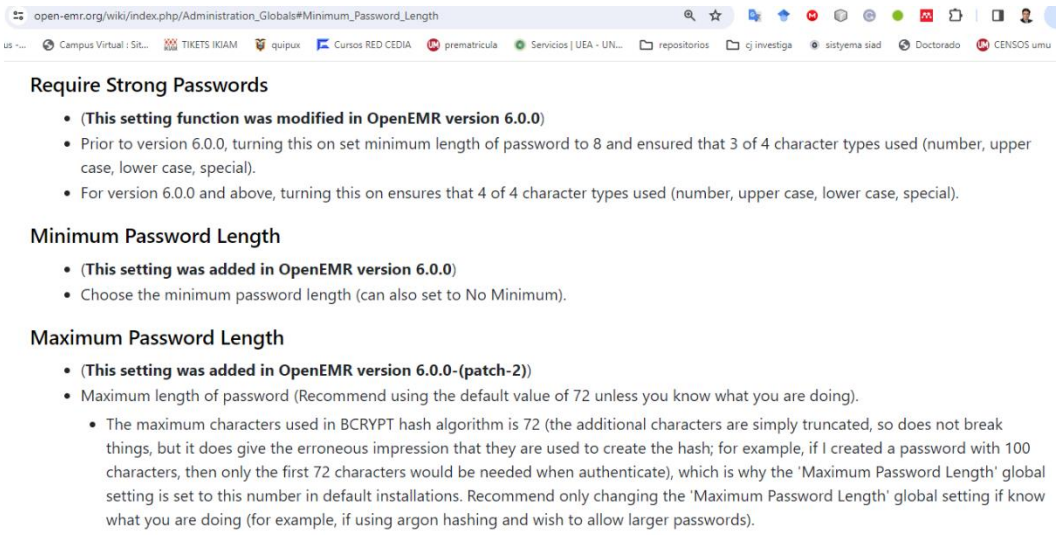
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

As Open-Emr.org's wiki page, the password complexity could be configured.



Result: It complies

PUID: [SEC-CAT-IAF-026]

Requirement description: The application must prevent users, except the administrator or the user associated with the password, from making password changes.

Source:

- ✓ **V-222548:** The application password must not be changeable by users other than the administrator or the user with which the password is associated.

  Protections must be utilized when establishing a password change or reset capability to prevent user A from changing user B's password.

  Protection is usually accomplished by having each user provide an out of bounds (OOB) communication address such as a separate email address or SMS/text address (mobile phone) that can be used to transmit password reset/change information.

  This OOB information is usually provided by the user when the user account is created. The OOB information is validated as part of the user account creation process by sending an account validation request to the OOB address and having the user respond to the request.

  Applications must prevent users other than the administrator or the user associated with the account from changing the account password. [13].

- ✓ **A01:2021 – Broken Access Control.**

  Permitting viewing or editing someone else's account, by providing its unique identifier (insecure direct object references) [3]

Priority: Not described

Rationale: This safety measure is essential to avoid unauthorized access and ensure that only administrators or owner users can control password modification. In addition, it prevents vulnerabilities associated with direct references to insecure objects that could allow the visualization or editing of other people's accounts.

Child PUIDs: Not described

Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria:

- ✓ **Password change restriction:**

  1. Verify that the application does not allow users, except the administrator or the associated user, to make password changes.

- ✓ **Password change protection:**

  1. Confirm that protection measures are implemented by establishing the capacity for change or password restoration, preventing a user (a) from changing the password of another user (b).

- ✓ **Validation of information out of limits (OOB):**

  1. Verify that the application uses OOB (out of limits) information, such as an email address or telephone number, to validate password change or restoration.

- ✓ **Unauthorized access prevention:**

  1. Confirm that unauthorized access to accounts is prevented by preventing the visualization or editing of foreign accounts by the principles established in A01: 2021 - Broken Access Control.

Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023
Auditory result:

The application only allows the change of passwords only to the administrator or the user itself.

Result: It complies.

PUID: [SEC-CAT-IAF-027]
Requirement description: Do not send or implement predetermined credentials, especially for administrators.
Source:

- ✓ **A07:2021 – Identification and Authentication Failures**
  - Do not ship or deploy with any default credentials, particularly for admin users. [3]

✓ **V-222662:** Default passwords must be changed. Default passwords can easily be compromised by attackers allowing immediate access to the applications. Configure the application to use strong authenticators instead of passwords when possible. Otherwise, change default passwords to a DoD-approved strength password and follow all guidance for passwords. [13].
Priority: Not described
Rationale: The elimination of default credentials is essential to mitigate safety risks since default credentials are vulnerable to attacks and can allow unauthorized access to applications
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Verification of elimination of predetermined credentials:** **1.** Confirm that the application is not sent or implemented with default credentials, especially for administrators. ✓ **Change of default passwords:** **1.** Validate that default passwords are changed. **2.** Verify that default passwords are changed to force passwords approved by the Department of Defense (DOD), following all applicable guidelines. ✓ **Use of strong authenticators:** **1.** Confirm that the application is configured to use strong authenticators instead of passwords whenever possible.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The application is configured to use strong authentications and doesn't have default passwords. **Result:** It complies.

CONCURRENT SESSIONS

PUID: [SEC-CAT-IAF-028]

Requirement description: The application must limit the number of concurrent sessions for each account to three.

Source:

- ✓ **AC-10** CONCURRENT SESSION CONTROL
  Control: Limit the number of concurrent sessions for each [Assignment: organization-defined account and/or account type] to [Assignment: organization-defined number] [11].

- ✓ **V-222387:** The application must provide a capability to limit the number of logon sessions per user. Design and configure the application to specify the number of logon sessions that are allowed per user.

  This requirement may be met via the application or by utilizing information system session control provided by a web server or other underlying solution that provides specialized session management capabilities. [13]

Priority: Not described

Rationale: Concurrent session control is essential to mitigate security risks by avoiding improper use of accounts. Limiting the number of active sessions per user reduces the possibility of unauthorized access and reinforces system safety.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Effective limitation of concurrent sessions:**

  1. Verify that the application can limit the number of login sessions per user.
  2. Confirm that the application is configured to specify the number of login sessions allowed per user.

- ✓ **Use of system session control:**

  1. Confirm that the application meets the requirement using session control provided by the application or through the management of specialized sessions provided by a web server or other underlying solution.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

After reviewing the official OpenEMR documentation and the constant options in the "global" menu, the auditor team found no option to limit multiple sessions with a single user account. The audit team was authenticated with a standard user in 4 different browsers, allowing their successful income.

Result: It doesn't comply.

SESSION EXPIRATION AND ENDING

PUID: [SEC-CAT-IAF-029]

Requirement description: The application must implement a user registration system with limited duration for organizations that connect to EHRI. After a specific period, the user's inscription of the POS systems that connect to EHRI must be renewed.

Source:

- ✓ **Security Requirement 55** – Time-Limited User Registration: All organizations connecting to the EHRi must ensure that the registration of users of PoS systems that connect to the EHRi is time-limited (after which, the user's registration must be renewed). [12]
- ✓ **SC-23 SESSION AUTHENTICITY**
  Control Enhancements:
  (1) SESSION AUTHENTICITY | INVALIDATE SESSION IDENTIFIERS AT LOGOUT
  Invalidate session identifiers upon user logout or other session termination.

  Discussion: Invalidating session identifiers at logout curtails the ability of adversaries to capture and continue to employ previously valid session IDs. [11].

Priority: Not described

Rationale: The limited duration of user registration contributes to security by ensuring that user credentials are subject to periodic renewal. It helps mitigate the risk associated with inactive users and reinforces the authenticity of the system.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **User registration implementation:**

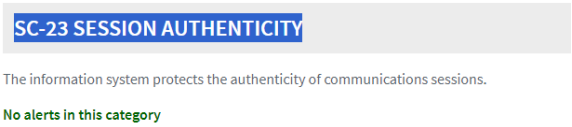
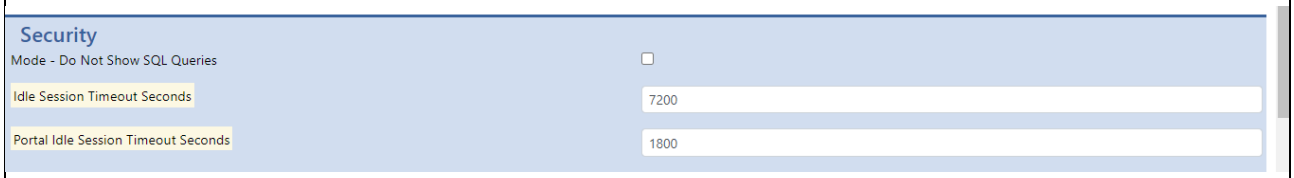
  1. Verify that the application implements a user registration system with a limited duration according to the organization's requirements.

- ✓ **Record renewal:**

  1. Confirm that, after the specified period, the system requires that users of the systems post-renew their records to maintain validity.

- ✓ **Session control (SC-23):**

  1. Validate that the application, as part of the session authenticity control (session authenticity), invalidates session identifiers when closing a user session or at the end of another session termination.

Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: A DAST and SAST have been performed, and no findings exist for "SC-23 SESSION AUTHENTICITY". The auditor team has also shown that the application allows setting timeout sessions.         **Result:** It complies.

PUID: [SEC-CAT-IAF-030]
Requirement description: The EHRI application must restrict the duration of connections to application services to provide additional security in access to those applications.
Source: ✓ **Security Requirement 69** – Restricting Connection Times to EHRi Applications: Where appropriate, the EHRi should restrict connection duration to EHRi application services to provide additional security for access to those applications. [12] ✓ **9.4.2 Secure log-on procedures**: l) restrict connection times to provide additional security for high-risk applications and reduce the window of opportunity for unauthorized access. [8]
Priority: Not described
Rationale: The restriction of connection times is essential to strengthening safety, especially in applications considered high-risk. Limiting the window of opportunity for unauthorized access helps mitigate possible threats and protects the integrity of the user's data and privacy.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria:

- ✓ **Verification of restrictions on the duration of connections:**

  1. Confirm that the EHRI application has implemented effective restrictions in the duration of connections to application services.

- ✓ **Security application analysis at the end of the session:**

  1. Evaluate whether the application properly restricts connection times, providing an additional safety layer, especially for high-risk applications.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

After reviewing the official OpenEMR documentation and the constant options in the "global" menu, the auditor team found no option to limit the duration of connections to application services.

Result: It doesn't comply.

PUID: [SEC-CAT-IAF-031]

Requirement description: The application must implement electronic procedures that automatically finish an electronic session after a default inactivity period.

Source:

- ✓ § **164.312** Technical safeguards.
  (a)
  (2) Implementation specifications:
  (iii) Automatic logoff (Addressable):
  Implement electronic procedures that terminate an electronic session after a predetermined time of inactivity. [10]

- ✓ **AC-2** ACCOUNT MANAGEMENT: Control Enhancements:
  (5) ACCOUNT MANAGEMENT | INACTIVITY LOGOUT
  Require that users log out when [Assignment: organization-defined time period of expected inactivity or description of when to log out].
  Discussion: Inactivity logout is behavior- or policy-based and requires users to take physical action to log out when they are expecting inactivity longer than the defined period. Automatic enforcement of inactivity logout is addressed by AC-11.[11]

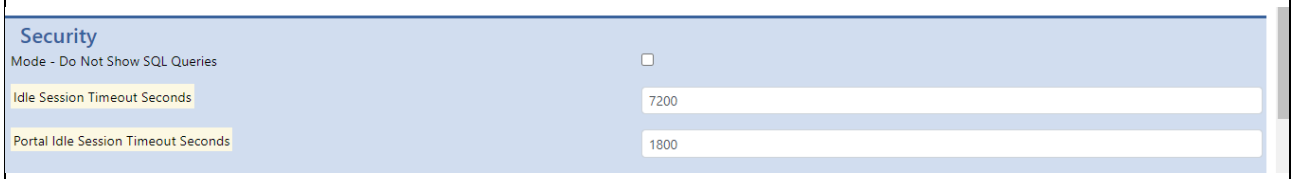
- ✓ Security Requirement 71 – Restricting Access to Unattended Workstations: All PoS systems connected to the EHRi must protect unattended workstations against an unauthorized person using the workstation while the PoS is active, such as with an automatic timeout after a period of inactivity. First, the best approach is to place workstations in a physically secure area in the first place. [12]

✓ Consideration MC21 – Establish and Enforce Session Timeouts: When designing or acquiring mobile applications intended to access or store PHI, organizations should ensure that the application has the ability to enforce a mandatory session timeout when left unattended. [12] ✓ 9.4.2 Secure log-on procedures: k) terminate inactive sessions after a defined period of inactivity, especially in high-risk locations such as public or external areas outside the organization's security management or on mobile devices [8]
Priority: Not described
Rationale: Implementing automatic session closures contributes to security by reducing the risk of unauthorized access in prolonged inactivity cases. It is essential to comply with technical safeguard specifications and improve accounts management.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Implementation of automatic closure procedures:** **1.** Verify that the application implements electronic procedures that automatically end the sessions after a period of inactivity, as specified. ✓ **Compliance with accounts management policies (AC-2):** **1.** Confirm that the application meets the accounts management requirements regarding the closure of inactivity sessions, as established in AC-2. ✓ **Safety of non -supervised work stations (Security Requirement 71):** **1.** Evaluate whether the systems connected to the EHRI protect the workstations not supervised by an automatic closure after a period of inactivity. ✓ **Consideration MC21 - Automatic closure in mobile applications:** **1.** Verify that mobile applications designed or acquired to access or store Phi can enforce a mandatory session closure when left unattended. ✓ **Compliance with safe login procedures (9.4.2):** **1.** Evaluate whether the application automatically ends inactive sessions after a defined period, especially in high-risk locations, such as public or external areas outside the organization's security management or on mobile devices.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná

13/12/2023

Auditory result:

The auditor team has shown that the application can set timeout sessions.



Result: It complies.

PUID: [SEC-CAT-IAF-032]

Requirement description: The application must automatically end the administrator user session and close the administrator's session after a 10-minute inactivity period is exceeded.

Source:

- ✓ **V-222390:** The application must automatically terminate the admin user session and log off admin users after a 10-minute idle time period is exceeded.

  Session termination terminates an individual user's logical application session after 10 minutes of application inactivity at which time the user must re-authenticate and a new session must be established if the user desires to continue work in the application.

  Design and configure the application to terminate the admin user's session after 10 minutes of inactivity. [13]

Priority: Not described

Rationale: The automatic termination of administrator sessions after a period of inactivity is essential to mitigate the safety risks associated with non-supervised active sessions. It ensures that the administrator sessions are protected against unauthorized accesses if an administrator forgets to close the session manually.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Automatic session termination verification:**

  1. Confirm that the application, as designed and configured, automatically ends the administrator user session after 10 minutes of inactivity.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

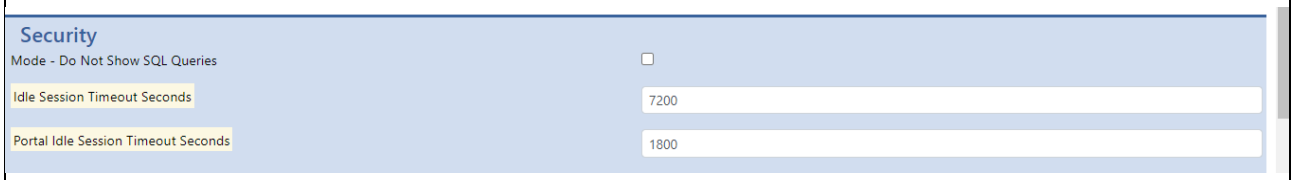
Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea

José A. García-Berná
13/12/2023

Auditory result:

The auditor team has shown that the application can set timeout sessions.



Result: It complies.

PUID: [SEC-CAT-IAF-033]

Requirement description: The application must automatically end the un-privileged user session and close the session of not-privileged users after a 15-minute inactivity period has elapsed.

Source:

- ✓ **V-222389:** The application must automatically terminate the non-privileged user session and log off non-privileged users after a 15-minute idle time period has elapsed.

  Session termination terminates an individual user's logical application session after 15 minutes of application inactivity at which time the user must re-authenticate and a new session must be established if the user desires to continue work in the application.

  Design and configure the application to terminate the non-privileged user's session after 15 minutes of inactivity. [13]

Priority: Not described

Rationale: Automatic session termination is crucial to guarantee the safety and privacy of information. After 15 minutes of inactivity, the risk of unauthorized access is reduced, and user authentication is reinforced when the sessions of non-privileged users are closed.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To_be_determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Session termination:**

  1. Confirm that the application automatically closes the non-privileged user session after 15 minutes of inactivity.

- ✓ **Session restart:**

  1. Verify that, after the automatic termination, the user must be authenticated to establish a new session.

- ✓ **Establishment of new session:**

1. Validate that if the user wishes to continue working in the application after the completion of the session, he must establish a new session through authentication.

Requested by: The organization

Responsible: Developer

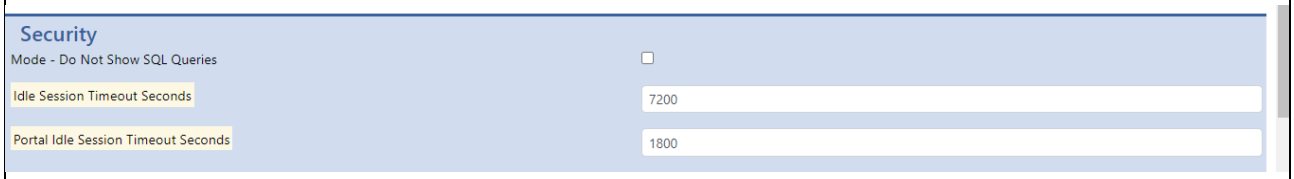
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The auditor team has shown that the application can set timeout sessions.



Result: It complies.

PUID: [SEC-CAT-IAF-034]

Requirement description: The application should allow the user to close the session in the communication sessions that he has started.

Source:

- ✓ **V-222391:** Applications requiring user access authentication must provide a logoff capability for user-initiated communication session.

  If a user cannot explicitly end an application session, the session may remain open and be exploited by an attacker. Applications providing user access must provide the ability for users to manually terminate their sessions and log off.

  Design and configure the application to provide all users with the capability to manually terminate their application session. [13]

- ✓ **V-222568:** The application must terminate all network connections associated with a communications session at the end of the session. Configure or design the application to terminate application network sessions at the end of the session. [13]

- ✓ **V-222578:** The application must destroy the session ID value and/or cookie on logoff or browser close. Configure the application to destroy session ID cookies once the application session has terminated. [13]

- ✓ **A01:2021 – Broken Access Control:** Stateful session identifiers should be invalidated on the server after logout. Stateless JWT tokens should rather be short-lived so that the window of opportunity for an attacker is minimized. For longer lived JWTs it's highly recommended to follow the OAuth standards to revoke access. [3]

- ✓ **V-222388:** The application must clear temporary storage and cookies when the session is terminated. Design and configure the application to clear sensitive data from cookies and local storage when the user logs out of the application.

  Persistent cookies are a primary means by which a web application will store application state and user information. Since HTTP is a stateless protocol, this persistence allows the web application developer to provide a robust and customizable user experience.

  However, if a web application stores user authentication information within a persistent cookie or other temporary storage mechanism, this information can be stolen and used to compromise the user's account. [13]

- ✓ **V-222392:** The application must display an explicit logoff message to users indicating the reliable termination of authenticated communications sessions.

  If a user is not explicitly notified that their application session has been terminated, they cannot be certain that their session did not remain open. Applications with a user access interface must provide an explicit logoff message to the user upon successful termination of the user session.

  Design and configure the application to provide an explicit logoff message to users indicating a successful logoff has occurred upon user session termination. [13]

- ✓ **SC-10** NETWORK DISCONNECT

  Control: Terminate the network connection associated with a communications session at the end of the session or after [Assignment: organization-defined time period] of inactivity.

  Discussion: Network disconnect applies to internal and external networks. Terminating network connections associated with specific communications sessions includes de-allocating TCP/IP address or port pairs at the operating system level and de-allocating the networking assignments at the application level if multiple application sessions are using a single operating system-level network connection. Periods of inactivity may be established by organizations and include time periods by type of network access or for specific network accesses. [11].

Priority: Not described

Rationale: Ensuring that users can close sessions explicitly is crucial to prevent the unwanted persistence of open sessions, which attackers could exploit. It protects the safety of the application and user information.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Session closing capacity:**

  1. Please verify that the application allows all users to close their application sessions manually.

- ✓ **Termination of network connections:**

  1. Confirm that the application ends all network connections associated with a communication session at the end of the session.

- ✓ **ID and/or Cookies session destruction:**

  1. Validate that the application destroys the ID and/or cookies session value when closing or closing the browser.

- ✓ **Invalidation of session identifiers on the server:**

  1. Ensure that state session identifiers are invalidated on the server after the session.

- ✓ **Temporary storage cleaning and cookies:**

  1. Verify that the application cleanses temporary storage and cookies when the user session closes.

- ✓ **Explicit session closure message:**

  1. Confirm that the application shows an explicit message closing message to the user indicating the successful termination of the session.

Requested by: The organization

Responsible: Developer

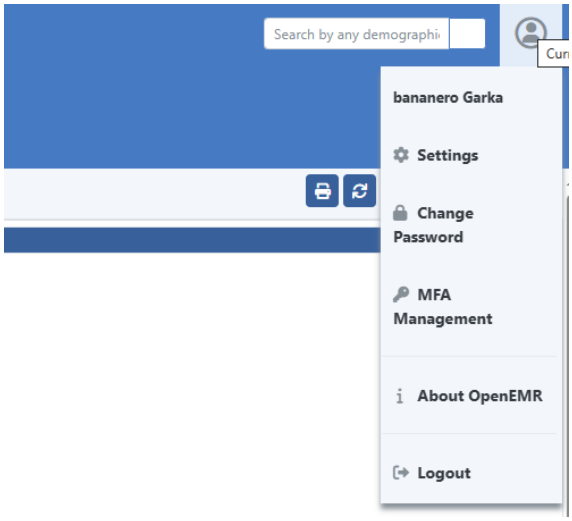
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The auditor team has reviewed the application, and there is an option to close an authenticated user session.



Result: It complies.

PUID: [SEC-CAT-IAF-035]

Requirement description: The application must ensure that, at the end of a session, all network connections linked to that session are closed, and the session ID and cookies data are destroyed.

Source:

- ✓ § **164.312** Technical safeguards.
  (a)
  (2) Implementation specifications:
  (iii) Automatic logoff (Addressable):
  Implement electronic procedures that terminate an electronic session after a predetermined time of inactivity. [10]

- ✓ **SC-10** NETWORK DISCONNECT

  Control: Terminate the network connection associated with a communications session at the end of the session or after [Assignment: organization-defined time period] of inactivity.

  Discussion: Network disconnect applies to internal and external networks. Terminating network connections associated with specific communications sessions includes de-allocating TCP/IP address or port pairs at the operating system level and de-allocating the networking assignments at the application level if multiple application sessions are using a single operating system-level network connection. Periods of inactivity may be established by organizations and include time periods by type of network access or for specific network accesses. [11].

- ✓ **V-222388:** The application must clear temporary storage and cookies when the session is terminated. Design and configure the application to clear sensitive data from cookies and local storage when the user logs out of the application.

  Persistent cookies are a primary means by which a web application will store application state and user information. Since HTTP is a stateless protocol, this persistence allows the web application developer to provide a robust and customizable user experience.

However, if a web application stores user authentication information within a persistent cookie or other temporary storage mechanism, this information can be stolen and used to compromise the user's account. [13]

- ✓ **V-222568:** The application must terminate all network connections associated with a communications session at the end of the session. Configure or design the application to terminate application network sessions at the end of the session. [13]

- ✓ **V-222578:** The application must destroy the session ID value and/or cookie on logoff or browser close. Configure the application to destroy session ID cookies once the application session has terminated. [13]

Priority: Not described

Rationale: Automatically eliminating the data associated with electronic sessions after their closure is essential to guarantee information security. It reduces the risk of unauthorized access to the application if a user forgets to close the session.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Temporary storage cleaning and cookies:**

  1. Confirm that the application cleanses temporary storage and cookies when the session ends.
  2. Validate that sensitive cookies and local storage data are eliminated when session.

- ✓ **Termination of network connections:**

  1. Verify that the application ends all network connections associated with a communication session at the end of the session.

- ✓ **Session ID destruction and cookies:**

  1. Confirm that the application destroys the value of the session ID and/or Cookie when closing the session or closing the browser.
  2. Validate that the application is configured to destroy session ID cookies once the application session is over.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The auditor team reviewed the application, and the destroy session method was found when the user finished his session.

```
root@oemr:/home/oemr# grep -r "session_destroy" /var/www/html/openemr/ *.* | mor
e
/var/www/html/openemr/src/Common/Session/SessionUtil.php:        session_destroy
();
/var/www/html/openemr/portal/patient/fwk/libs/verysimple/Authentication/Authenti
cator.php:        session_destroy();
```

Result: It complies.

PREVENT THE USER FROM LOGGING ON AFTER ACCESS PRIVILEGES HAVE BEEN REVOKED

PUID: [SEC-CAT-IAF-036]

Requirement description: The application must be able to revoke the user's access privileges on time, which implies immediately preventing the user from accessing the system once their privileges have been revoked.

Source:

- ✓ **Security Requirement 60** – Timely Revocation of Access Privileges: The EHRi and all PoS systems connected to the EHRi must support the revocation of user access privileges in a timely manner (i.e., immediately prevent the user from logging on after access privileges have been revoked). [12]

- ✓ **V-222408:** Shared/group account credentials must be terminated when members leave the group. [13].

Priority: Not described

Rationale: The timely revocation of access privileges is crucial to guarantee system safety by preventing unauthorized access. It is imperative when accounts or groups are shared since shared accounts should be guaranteed to be deactivated when members leave the group.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Verification of the timely revocation of privileges:**

  1. Confirm that the application can immediately revoke a user's access privileges after they have been revoked.

- ✓ **Shared/group credential termination:**

  1. Validate that the credentials of shared or group accounts end when the members leave the group.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: Through the administration module, the auditor team has inactivated a user and then tried to authenticate with the aforementioned inactive user, and the system has not allowed it; consequently, the requirement is met. **Result:** It complies.

PUID: [SEC-CAT-IAF-037]
Requirement description: The application must finish the existing user sessions by eliminating an account. Configure the application to close user sessions whose accounts have been deleted.
Source: ✓ **V-222549:** The application must terminate existing user sessions upon account deletion. Configure the application to terminate existing sessions of users whose accounts are deleted. The application must ensure that a user does not retain any rights that may have been granted or retain access to the application after the user's authorization or role within the application has been deleted or modified. This means once a user's role/account within the application has been modified, deleted or disabled, the changes must be enforced immediately within the application [13].
Priority: Not described
Rationale: Immediately terminating user sessions by eliminating an account is essential to guarantee safety and prevent unauthorized access. It ensures that users do not retain rights or access to the application after their authorization or role has been eliminated or modified.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Session termination verification by eliminating accounts:** **1.** Confirm that the application automatically ends the existing user sessions by eliminating an account. **2.** Verify that the application configuration is adjusted to close user sessions with deleted accounts. ✓ **Role's modification evaluation or authorization:**

1. Verify that once the role or account of a user has been modified, eliminated, or disabled, the changes are applied immediately within the application.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: Through the administration module, the auditor team inactivated a user and then tried to List Patients with the aforementioned inactive user, and the system redirected to the login webpage; consequently, the requirement was met. **Result:** It complies.

PUID: [SEC-CAT-IAF-038]
Requirement description: The application must automatically delete or deactivate temporary user accounts 72 hours after creation. Configure the temporary accounts so they are deleted or deactivated automatically after 72 hours of the account's creation.
Source: ✓ **V-222409:** The application must automatically remove or disable temporary user accounts 72 hours after account creation. Configure temporary accounts to be automatically removed or disabled after 72 hours after account creation. [13].
Priority: Not described
Rationale: Proper temporary management is crucial to maintain system safety and prevent possible risks associated with unused accounts. By automatically eliminating or deactivating temporary accounts after a defined period of time, exploitation opportunities are reduced, and the application security position is strengthened.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Automatic removal or deactivation verification:** 1. Confirm that the application automatically eliminates or deactivates temporary user accounts 72 hours after creation.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date:

Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The audit team reviewed the application documentation and did not find temporary accounts management; therefore, it is considered not applicable for the audit (N/a). **Result:** N/a

PUID: [SEC-CAT-IAF-039]
Requirement description: The application must automatically deactivate the user accounts after an inactivity period of 35 days.
Source: ✓ **V-222411:** The application must automatically disable accounts after a 35-day period of account inactivity. Design and configure the application to expire user accounts after 35 days of inactivity. This policy does not apply to either emergency accounts or infrequently used accounts. Infrequently used accounts are local logon administrator accounts used by system administrators when network or normal logon/access is not available. Emergency accounts are administrator accounts created in response to crisis situations. [13].
Priority: Not described
Rationale: Automatic deactivation of inactivity accounts is essential to maintain application security by reducing exposure of unused accounts. It helps prevent possible unauthorized access and guarantees effective username management.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Verification of the inactivity period:** **1.** Confirm that the application automatically deactivates the user accounts after an inactivity period of 35 days. ✓ **Exceptions for specific accounts:** **1.** Validate that the deactivation policy does not apply to emergency accounts or accounts for infrequent use, according to the definition provided. **2.** Confirm that the local login administrator accounts used by system administrators in situations without regular access are not affected. **3.** Confirm that emergency administrator accounts created in response to crisis situations are not deactivated according to the policy.
Requested by: The organization
Responsible: Developer
Configurable value: Not described

Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The audit team reviewed the application documentation and found no option to deactivate user accounts automatically. **Result:** It doesn't comply.

2.9.3.8 Software and Data Integrity Failures

PUID: [SEC-CAT-SDF-001]
Requirement description: The application must acquire components only from official sources through safe links. Preference must be given to signed packages to reduce the possibility of including a modified or malicious component.
Source: ✓ **A06:2021 – Vulnerable and Outdated Components:** o Only obtain components from official sources over secure links. Prefer signed packages to reduce the chance of including a modified, malicious component (See A08:2021-Software and Data Integrity Failures). [3] ✓ **A08:2021 – Software and Data Integrity Failures:** o Use digital signatures or similar mechanisms to verify the software or data is from the expected source and has not been altered. o Ensure libraries and dependencies, such as npm or Maven, are consuming trusted repositories. If you have a higher risk profile, consider hosting an internal known-good repository that's vetted. [3]
Priority: Not described
Rationale: Using software components from unofficial sources or non-safe links can introduce vulnerabilities and integrity failures into the application. This requirement guarantees that the components are obtained from reliable sources, thus reducing the risk of compromising the safety and integrity of the system.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Component origin:**

1. Verify that the application acquires all components from official sources and through safe links.

✓ **Use of signed packages:**

1. Confirm that preference is given to signed packages during the acquisition of components to reduce the possibility of including modified or malicious components.

✓ **Digital signature verification:**

1. Evaluate using digital signatures or similar mechanisms to verify that the software or data comes from the expected source and has not been altered.

✓ **Integrity of libraries and dependencies:**

1. Confirm that libraries and dependencies, such as NPM or Maven, consume trust repositories. Consider using a known and verified internal repository for a higher risk profile.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are findings for "A06:2021 – Vulnerable and Outdated Components" and "A08 Software and Data Integrity Failures".

Result: It doesn't comply

PUID: [SEC-CAT-SDF-002]

Requirement description: The application must incorporate a "software supply chain safety tool" to verify that the components do not contain known vulnerabilities.

Source:

✓ **A08:2021 – Software and Data Integrity Failures:**

○ Ensure that a software supply chain security tool, such as OWASP Dependency Check or OWASP CycloneDX, is used to verify that components do not contain known vulnerabilities. [3]
Priority: Not described
Rationale: Continuously verifying the integrity of the software and the data is essential to mitigate security risks associated with known vulnerabilities. Using safety tools in the supply chain strengthens safety posture and protects against threats derived from components with known failures.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Security tool implementation:** **1.** Confirm that the application has effectively integrated a safety tool for the software supply chain, such as Owasp dependency check or Owasp Cyclonedx. ✓ **Component verification:** **1.** Use the tool to verify that the components used in the application do not contain known vulnerabilities.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: A DAST and SAST have been performed, and the findings are "A08 Software and Data Integrity Failures". Consequently, the software supply chain security tool was ineffective in this case.      **Result:** It doesn't comply

PUID: [SEC-CAT-SDF-003]
Requirement description: The application must have a process for reviewing code and configuration changes to minimize the possibility of malicious code or configuration introduction in the software pipeline.
Source:

- ✓ **A08:2021 – Software and Data Integrity Failures:**
    - Ensure that there is a review process for code and configuration changes to minimize the chance that malicious code or configuration could be introduced into your software pipeline. [3]

Priority: Not described

Rationale: The rigorous review of code and configuration changes is essential to guarantee the integrity of the software and prevent the accidental introduction of malicious elements. This process reduces the risk of vulnerabilities and safeguards the integrity of the data and the system's operation.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To_be_determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **CHANGE REVIEW:**

    1. Confirm that there is a formal process to review code and configuration changes.

- ✓ **Malicious code mitigation:**

    1. Verify that the review process minimizes the possibility of introducing malicious code.

- ✓ **Review Registration:**

    1. Validate that change reviews are appropriately registered and documented.

Requested by: The organization

Responsible: Developer

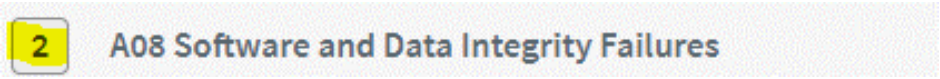
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and the findings are “A08 Software and Data Integrity Failures”. Consequently, the “process for reviewing code and configuration changes” was ineffective.



Result: It doesn’t comply

PUID: [SEC-CAT-SDF-004]

Requirement description: The application must ensure that its continuous integration channel/continuous deployment (CI/CD) has adequate segregation, safe configuration, and access control to guarantee the integrity of the code that flows through construction and implementation processes.

Source:

- ✓ **A08:2021 – Software and Data Integrity Failures:**
  - Ensure that your CI/CD pipeline has proper segregation, configuration, and access control to ensure the integrity of the code flowing through the build and deploy processes. [3]

Priority: Not described

Rationale: The integrity of the software and the data in the CI/CD process is crucial to maintaining confidence in the quality and safety of the implementations. Ensuring adequate segregation, configuration, and access control is essential to prevent malicious failures and manipulations during the code flow.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Segregation in the CI/CD process:**
  1. Verify adequate segregation on the CI/CD channel to avoid unauthorized interference between different code flows.
- ✓ **Safe CI/CD settings:**
  1. Confirm that CI/CD configuration is correctly established to guarantee the integrity of the code during the construction and deployment stages.
- ✓ **CI/CD access control:**
  1. Validate that effective access controls are implemented on the CI/CD channel to restrict unauthorized access to the critical components of the process.

Requested by: The organization

Responsible: Developer

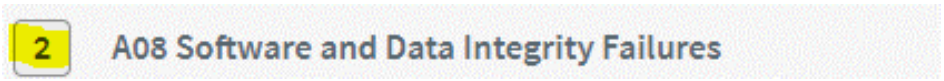
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and the findings are “A08 Software and Data Integrity Failures”. Consequently, the “effective access controls implemented on the CI/CD channel to restrict unauthorized access to the critical components of the process” were ineffective.



Result: It doesn't comply

PUID: [SEC-CAT-SDF-005]

Requirement description: The application must guarantee that the non-signed or non-encrypted serialized data are not sent to non-reliable customers without integrity verification or digital signature to detect manipulations or reproductions of the serialized data.

Source:

- ✓ **A08:2021 – Software and Data Integrity Failures:**
  - Ensure that unsigned or unencrypted serialized data is not sent to untrusted clients without some form of integrity check or digital signature to detect tampering or replay of the serialized data. [3]

Priority: Not described

Rationale: Guaranteeing data integrity and software is crucial to prevent unauthorized manipulation or serialized data reproduction. This requirement is aligned with the best security practices to protect the confidentiality and precision of information.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Serialized data integrity verification:**
  1. Confirm that the application implements mechanisms to verify the integrity of serialized data before sending them to non-reliable customers.
- ✓ **Digital signature review or integrity verification:**
  1. Validate that a digital signature or integrity verification method for serialized data is used before being sent to non-reliable customers.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The audit team reviewed the application documentation and did not find information about "a digital signature or integrity verification method for serialized data"; therefore, it is considered not applicable for the audit (N/a).

Result: N/a

PUID: [SEC-CAT-SDF-006]

Requirement description: The application must be able to show the previous content of a record at any point in the past, as well as the associated details of who entered, agreed, or modified the data and at what time.

Source:

- ✓ **Security Requirement 38** – Preserving the History of PHI in the EHRi: The EHRi must be capable of displaying the former content of a record at any point in the past, as well the associated details of who entered, accessed or modified the data, and at what time. [12]
- ✓ **Security Requirement 39** – Preserving the History of PHI in PoS Systems: All PoS systems connected to the EHRi should be capable of displaying the former content of a record at any point in the past, as well as the associated details of who entered, accessed or modified the data, and at what time. [12]

Priority: Not described

Rationale: Phi history preservation is critical to guarantee the integrity and traceability of personal health information. This capacity allows the precise recovery of previous records, which is essential for the monitoring and auditing of health information over time.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To_be_determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Verification of the visualization capacity of the previous content:**

  1. Confirm that the application can show the previous content of a record anywhere in the past.

- ✓ **Validation of associated details:**

  1. Confirm that the application provides associated details, such as who entered, agreed to, or modified the data and at what time.

- ✓ **Preservation capacity evaluation:**

  1. Verify that the application maintains a complete PHI history that allows the precise recovery of previous records.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán

Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The audit team reviewed the application documentation and tables in the database and found no option to "show the previous content of a record at any point in the past." **Result:** It doesn't comply.

PUID: [SEC-CAT-SDF-007]
Requirement description: The application must implement policies and procedures to protect protected electronic health information (EPHI) against improper alterations or destruction.
Source: ✓ § **164.312** Technical safeguards. **(c)** **(1)** Standard: integrity. ✓ Implement policies and procedures to protect electronic protected health information from improper alteration or destruction. [10]
Priority: Not described
Rationale: The integrity of protected electronic health information (Ephi) is crucial to ensure that data is not altered or destroyed inappropriately, which could compromise the precision and reliability of medical information.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To_be_determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Verification of policies and procedures:** **1.** Confirm that the application has implemented specific policies and procedures to protect the integrity of protected electronic health information (Ephi). ✓ **Alteration's evaluation:** **1.** Verify that there are effective measures to prevent inappropriate alterations of protected electronic health information (Ephi). ✓ **Evaluation against improper destruction:** **1.** Verify that there are effective measures to prevent inappropriate destruction of protected electronic health information (Ephi).
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date:

Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: A DAST and SAST have been performed, and no findings are related to "altered or destroyed data". Additionally, as reviewed by the auditor team, there are options in the system for cyphering medical data. **Result:** It complies.

2.9.3.9 Security Logging and Monitoring Failures

AUDIT RECORDS

PUID: [SEC-CAT-LMF-001]
Requirement description: The application must implement a comprehensive security management process to guarantee the non-repudiation, confidentiality, integrity, and availability of all protected electronic health information (EPHI) created, received, maintained, or transmitted. This process must include the periodic review of system activity records, such as audit records, access reports, and security monitoring reports.
Source: ✓ **§ 164.308 Administrative safeguards**: **(a)** A covered entity or business associate must, in accordance with § 164.306: **(1)** Ensure the confidentiality, integrity, and availability of all electronic protected health information the covered entity or business associate creates, receives, maintains, or transmits. **(i)** Standard: Security management process: Implement policies and procedures to prevent, detect, contain, and correct security violations **(ii)** Implementation specifications: **(D)** Information system activity review (Required) Implement procedures to regularly review records of information system activity, such as audit logs, access reports, and security incident tracking reports [10]. ✓ **8.15 Logging**: Logs that record activities, exceptions, faults and other relevant events should be produced, stored, protected and analyzed. [7] ✓ **AU-10 NON-REPUDIATION** Control: Provide irrefutable evidence that an individual (or process acting on behalf of an individual) has performed [Assignment: organization-defined actions to be covered by non-repudiation]. [11] ✓ A logging mechanism should audit who entered the information and where and how the data originated if it was electronically transferred [15] ✓ Record who entered or manipulated which data. [16]

- ✓ 5.28 Collection of evidence: The organization should establish and implement procedures for the identification, collection, acquisition and preservation of evidence related to information security events [7]

- ✓ **V-222476:** The application must produce audit records that contain information to establish the outcome of the events. Configure the application to include the outcome of application functions or events. [13]

- ✓ **V-222438:** The application must protect against an individual (or process acting on behalf of an individual) falsely denying having performed organization-defined actions to be covered by non-repudiation. Configure the application to provide users with a non-repudiation function in the form of digital signatures when it is required by the organization or by the application design and architecture. [13]

Priority: Not described

Rationale: Avoid non-repudiation and comply with the security standards established by the regulations, such as the requirement of § 164.308 (a) (1) that requires that the entities covered or commercial associates implement policies and procedures to prevent, detect, contain, and correct violations of security. Reviewing system activities is essential to maintain continuous control over the safety of protected electronic health information.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Regular review of system activity records:**

  1. Verify that the application has procedures to regularly review system activity records, such as audit records, access reports, and monitoring reports.

- ✓ **Non -repudiation guarantee:**

  1. Verify that the application provides irrefutable evidence that an individual (or process in the name of an individual) has carried out actions defined by the organization and covered by the principle of non-repudiation, as established in AU-10.

- ✓ **Data input and manipulation record:**

  1. Confirm that the application records who entered or manipulated what data, thus ensuring the traceability of activities related to protected electronic health information.

- ✓ **Non -repudiation function implementation:**

  1. Please verify that the application is configured to protect against the false denial of actions carried out by an individual, covered by non-repudiation, by implementing digital signature functions when required by the organization or by the design and architecture of the application, according to It is specified in V-222438.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The auditor team has identified audit logs and reports in the admin module. **Result:** It complies.

PUID: [SEC-CAT-LMF-002]
Requirement description: The application must perform automatic audits of the following user accounts: **1.** User account creation. **2.** User account modification. **3.** User account deactivation. **4.** User account elimination.
Source: ✓ AC-2 ACCOUNT MANAGEMENT: Control Enhancements: (4) ACCOUNT MANAGEMENT | AUTOMATED AUDIT ACTIONS Automatically audit account creation, modification, enabling, disabling, and removal actions [11]. ✓ **V-222413:** The application must automatically audit account creation. [13] ✓ **V-222467:** The application must generate audit records for all account creations, modifications, disabling, and termination events. Configure the application to log user account creation, modification, disabling, and termination events Configure the application to write a log entry when a new user account is created. At a minimum, ensure account name, date and time of the event are recorded. [13] ✓ **V-222414:** The application must automatically audit account modification. Configure the application to write a log entry when a user account is modified. At a minimum, ensure account name, date and time of the event are recorded. One way for an attacker to establish persistent access is for the attacker to modify or copy an existing account. Auditing of account modification is one method for mitigating this risk [13] ✓ **V-222415:** The application must automatically audit account disabling actions. Configure the application to write a log entry when a user account is disabled. At a minimum, ensure account name, date and time of the event are recorded.

When application accounts are disabled, user accessibility is affected. Accounts are utilized for identifying individual application users or for identifying the application processes themselves [13]

- ✓ **V-222416:** The application must automatically audit account removal actions.

  Configure the application to write a log entry when a user account is removed.

  At a minimum, ensure account name, date and time of the event are recorded. [13]

Priority: Not described

Rationale: Automatic audit of user accounts is crucial to maintaining the integrity and safety of the system. Registering events such as creating, modifying, deactivating, or eliminating user accounts helps to detect and mitigate possible safety risks, such as unauthorized access or persistence attempts.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Account creation audit:**

  1. Confirm that the application automatically records user account creation events.
  2. Verify that at least the account's name, date, and event time are registered.

- ✓ **Account modification audit:**

  1. Validate that the application automatically generates records when a user account is modified.
  2. Ensure that, at least, the name of the account date and time of the event are recorded.

- ✓ **Account deactivation audit:**

  1. Verify that the application automatically records user account deactivation events.
  2. Confirm that at least the account's name, date, and event time are recorded.

- ✓ **Account elimination audit:**

  1. Validate that the application automatically generates records when a user account is deleted.
  2. Ensure that, at least, the name of the account date and time of the event are recorded.

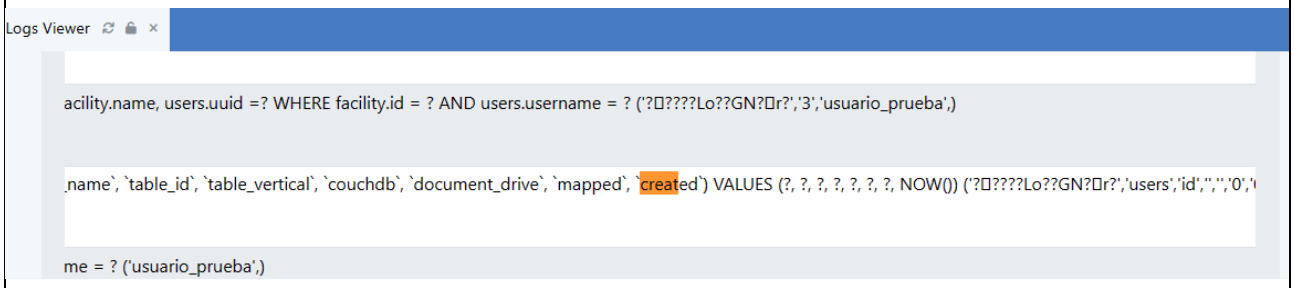
Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná

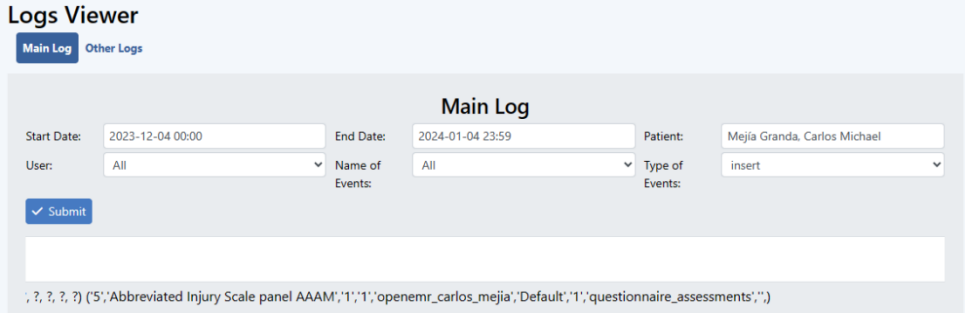
13/12/2023
Auditory result: The auditor team has identified audit logs and reports in the admin module.      **Result:** It complies.

PUID: [SEC-CAT-LMF-003]
Requirement description: The application must have an audit reduction capacity that supports research after security incidents. The application must be configured to offer an audit reduction capacity that supports forensic investigations.
Source: ✓ **V-222491:** The application must provide an audit reduction capability that supports after-the-fact investigations of security incidents. Configure the application to provide an audit reduction capability that supports forensic investigations. [13]
Priority: Not described
Rationale: The audit reduction capacity is essential to analyze security events after events. It allows effective investigation of security incidents and provides valuable information to understand and mitigate potential threats.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Verification of audit reduction capacity:** 1. Confirm that the application has an audit reduction capacity that allows research after security incidents. ✓ **Configuration for Forensic Research:** 1. Validate that the application is configured to provide an audit reduction capacity specifically designed to support forensic investigations.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The auditor team has identified audit logs and reports in the admin module. Moreover, these audit reports can be filtered by user, time, group, action, etc.



Result: It complies.

PUID: [SEC-CAT-LMF-004]

Requirement description: The application must generate audit records when successful/unsuccessful attempts are made to grant privileges. Configure the application to audit successful and unsuccessful attempts to grant privileges.

Source:

- ✓ **V-222450:** The application must generate audit records when successful/unsuccessful attempts to grant privileges occur. Configure the application to audit successful and unsuccessful attempts to grant privileges.

  When a user is granted access or rights to application features and function not afforded to an ordinary user, they have been granted access to privilege and that action must be logged. [13]

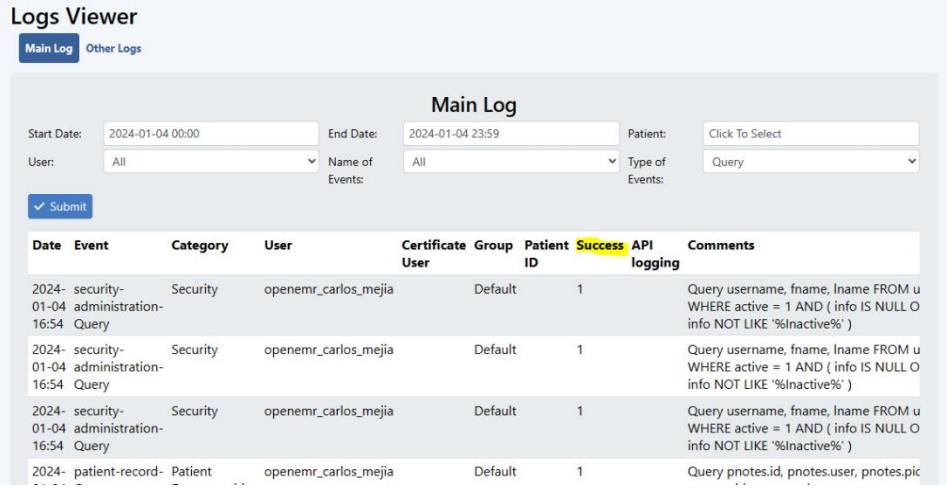
- ✓ **V-222452:** The application must generate audit records when successful/unsuccessful attempts to access security levels occur. Configure the application to create an audit record for both successful and unsuccessful attempts to access security levels. [13]

- ✓ **A01:2021 – Broken Access Control:** Log access control failures, alert admins when appropriate (e.g., repeated failures). [3]

- ✓ **V-222453:** The application must generate audit records when successful/unsuccessful attempts to access categories of information (e.g., classification levels) occur. Configure the application to create an audit record for both successful and unsuccessful attempts to access protected categories of information.

  Categories of information is information that is identified as being sensitive or requiring additional protection from regular user access. The data is accessed on a need-to-know basis

and has been assigned a category or a classification in order to assign protections and track access. [13]
Priority: Not described
Rationale: The registration of actions related to granting privileges is essential for the monitoring and safety of the system. The generation of audit records allows the early detection of suspicious activities and guarantees transparency in privilege management.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Privilege granting audit:** **1.** Verify that the application generates audit records for successful and unsuccessful attempts to grant privileges. **2.** Confirm that the application is appropriately configured to audit successful and unsuccessful grant privileges attempts. ✓ **Access audit to security levels:** **1.** Confirm that the application generates audit records for successful and unsuccessful attempts to access security levels. **2.** Validate that the application configuration allows the creation of audit records for both types of access to security levels. ✓ **Registration of access control failures (A01: 2021):** **1.** Verify that the application records failures in access control as specified in A01: 2021. **2.** Alert administrators appropriately in cases of repeated failures. ✓ **Audit to access information categories:** **1.** Confirm that the application generates audit records for successful and unsuccessful attempts to access information-protected categories. **2.** Validate that the application configuration allows the creation of audit records for both access categories.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The auditor team has identified audit logs and reports in the admin module. Moreover, these audit reports for granting privileges can be filtered by success.



Result: It complies.

PUID: [SEC-CAT-LMF-005]

Requirement description: The application must generate audit records for successful and unsuccessful attempts to modify privileges, security levels, and information categories.

Source:

- ✓ **V-222454:** The application must generate audit records when successful/unsuccessful attempts to modify privileges occur. Configure the application to audit successful and unsuccessful attempts to modify privileges. [13]

- ✓ **V-222456:** The application must generate audit records when successful/unsuccessful attempts to modify security levels occur. Configure the application to create an audit record for both successful and unsuccessful attempts to modify security levels.

  A security level denotes a permissions or authorization capability within the application. This is most often associated with a user role. Attempts to modify a security level can be construed as an attempt to change the configuration of the application so as to create a new security role or modify an existing security role. Some applications may or may not provide this capability. [13]

- ✓ **V-222457:** The application must generate audit records when successful/unsuccessful attempts to modify categories of information (e.g., classification levels) occur. Configure the application to create an audit record for both successful and unsuccessful attempts to modify protected categories of information. [13]

Priority: Not described

Rationale: The generation of audit records for changes in privileges, security levels and information categories are critical to monitor and audit activities related to the modification of sensitive configurations. It contributes to the general security of the system and allows the early detection of possible unauthorized attempts to modify it.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Privilege modification:**

  1. Verify the application generates audit records for successful and unsuccessful attempts to modify privileges.

- ✓ **Modification of security levels:**

  1. Confirm that the application records audit events for successful and unsuccessful attempts to modify security levels.

- ✓ **Modification of information categories:**

  1. Validate that the application registers audit activities for successful and unsuccessful attempts to modify information categories, such as classification levels.

Requested by: The organization

Responsible: Developer

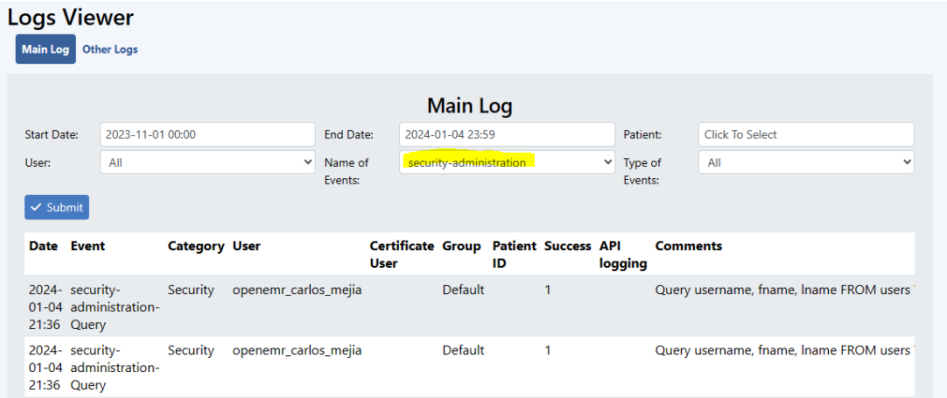
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The auditor team has identified audit logs and reports in the admin module. Moreover, these audit reports can be filtered by success or not for modifying privileges. Additionally, in the option “Name of events”, the auditors' team found security levels and category information.



Result: It complies.

PUID: [SEC-CAT-LMF-006]

Requirement description: The application must generate audit records for successful and failed attempts to eliminate privileges, security levels, database safety objects, and information categories.

Source:

- ✓ **V-222458:** The application must generate audit records when successful/unsuccessful attempts to delete privileges occur. Configure the application to audit successful and unsuccessful attempts to delete privileges. [13]

- ✓ **V-222459:** The application must generate audit records when successful/unsuccessful attempts to delete security levels occur. Configure the application to create an audit record for both successful and unsuccessful attempts to delete security levels.

  A security level denotes a permissions or authorization capability within the application. This is most often associated with a user role. Attempts to delete a security level can be construed as an attempt to change the configuration of the application so as to delete an existing security role. Some applications may or may not provide this capability. [13]

- ✓ **V-222460:** The application must generate audit records when successful/unsuccessful attempts to delete application database security objects occur. Configure the application to create an audit record for both successful and unsuccessful attempts to delete database security objects. [13]

- ✓ **V-222461:** The application must generate audit records when successful/unsuccessful attempts to delete categories of information (e.g., classification levels) occur. Configure the application to create an audit record for both successful and unsuccessful attempts to delete protected categories of information. [13]

Priority: Not described

Rationale: The generation of audit records for critical actions, such as eliminating privileges, security levels, and security objects of the database, is essential for the monitoring of changes and the early detection of inappropriate activities. It contributes to the integral security of the system and guarantees accountability.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Verification of audit records:**

  1. Confirm that the application generates audit records for successful and failed attempts to eliminate privileges.

- ✓ **Validation of security level records:**

  1. Verify the application records audit events for successful and failed attempts to eliminate security levels.

- ✓ **Evaluation of security objects records of the database:**

1. Ensure the application generates audit records for successful and failed attempts to delete database safety objects.

✓ **Confirmation of information categories records:**

1. Validate that the application creates audit records for successful and failed attempts to eliminate protected information categories.

Requested by: The organization

Responsible: Developer

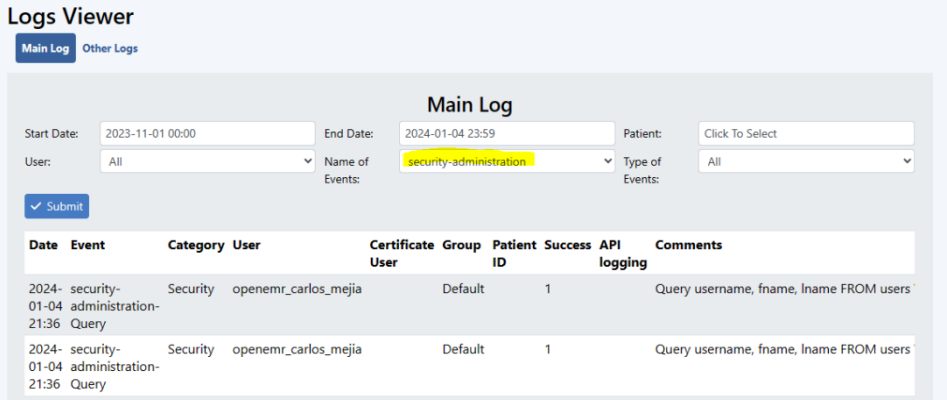
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The auditor team has identified audit logs and reports in the admin module. Moreover, these audit reports can be filtered by success or not for deleting privileges. Additionally, in the option "Name of events", the auditors' team found security levels and category information.



Result: It complies.

PUID: [SEC-CAT-LMF-007]

Requirement description: The application must generate audit records for privileged activities, accesses at the system level, successful/non-success to objects, and all direct access to the information system. The application in registration tickets corresponding to each type of activity mentioned must be configured.

Source:

✓ **V-222463:** The application must generate audit records for privileged activities or other system-level access. Configure the application to write a log entry when privileged activities or other system-level events occur. [13]

- ✓ **V-222465:** The application must generate audit records when successful/unsuccessful accesses to objects occur. Configure the application to log successful and unsuccessful access to application objects. [13]

- ✓ **V-222466:** The application must generate audit records for all direct access to the information system. Configure the application to log all direct access to the system. [13]

Priority: Not described

Rationale: The generation of audit records is essential to monitor and track activities in the application, providing visibility on privileged actions, object access, and any direct access to the system. It is necessary for the early detection of possible threats or security anomalies.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Privileged activities records:**

  1. Verify that the application generates audit records for privileged activities or events at the system level.
  2. Confirm that the application configuration allows the writing of registration inputs for such activities.

- ✓ **Successful/non-successful access records to objects:**

  1. Validate that the application generates audit records for successful and unsuccessful accesses to objects.
  2. Confirm that the application configuration allows the proper record of these accesses.

- ✓ **Records of all direct access to the system:**

  1. Confirm that the application generates audit records for all direct access to the system.
  2. Verify that application configuration guarantees the inclusion of all direct accesses in the records.

Requested by: The organization

Responsible: Developer

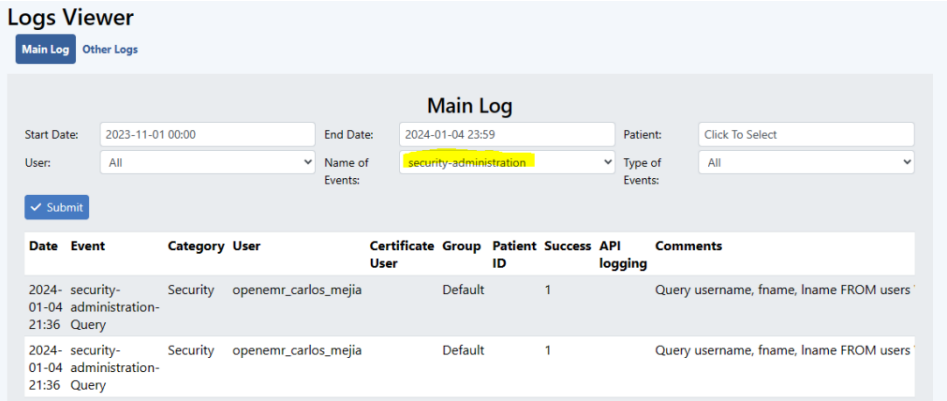
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The auditor team has identified audit logs and reports in the admin module. Moreover, these audit reports can be filtered by success or not for privileged activities or accessing restricted areas. Additionally, in the option “Name of events”, the auditors' team found security levels and category information.



Result: It complies.

PUID: [SEC-CAT-LMF-008]

Requirement description: The application must register the actions of users related to access and data changes. It includes identifying specific data elements that require protection and auditing all access and modification of data.

Source:

- ✓ **V-222471:** The application must log user actions involving access to data. Identify the specific data elements requiring protection and audit access to the data.

  When users access application data, there is risk of data compromise or seepage if the account used to access is compromised or access is granted improperly. To be able to investigate which account accessed data, the account access must be logged. Without establishing when the access event occurred, it would be difficult to establish, correlate, and investigate the events relating to an incident, or identify those responsible for one.

  Associating event types with detected events in the application and audit logs provides a means of investigating an attack; recognizing resource utilization or capacity thresholds; or identifying an improperly configured application. [13]

- ✓ **V-222472:** The application must log user actions involving changes to data. Configure the application to log all changes to application data. [13]

Priority: Not described

Rationale: The user actions record is crucial for data security. It allows traceability and research capacity in case of unauthorized access, undue changes, or account commitment. Identifying access events and data changes is essential for the early detection of possible threats and for investigating security incidents.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To_be_determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Data access:**

  1. Verify that the application generates audit records for all actions of users related to data access.
  2. Confirm that the specific data elements that require protection are duly identified in the audit records.

- ✓ **Changes in data:**

  1. Validate that the application records all the actions of users related to data changes.

Requested by: The organization

Responsible: Developer

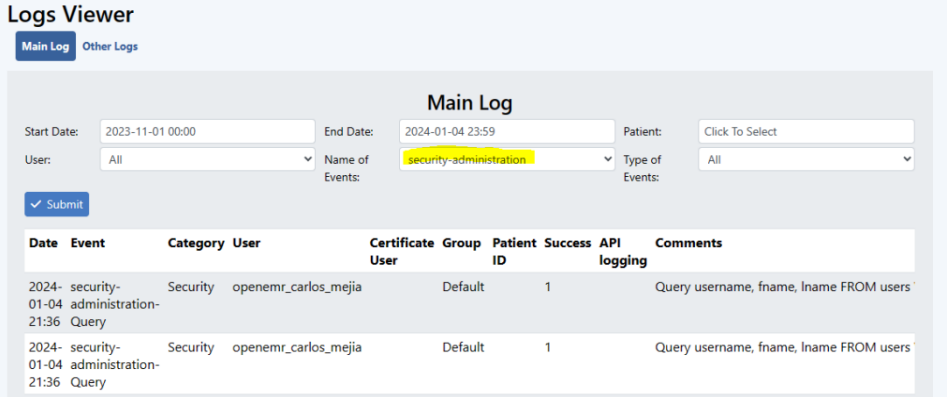
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The auditor team has identified audit logs and reports in the admin module. Moreover, these audit reports can be filtered by success or not for data access.



Result: It complies.

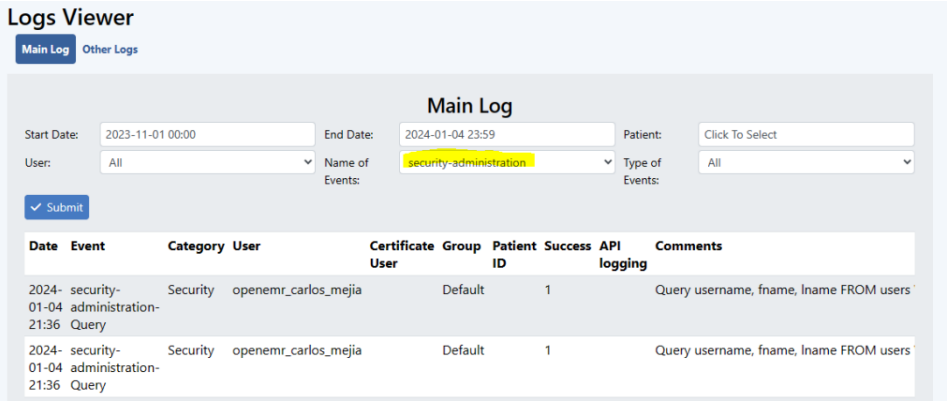
PUID: [SEC-CAT-LMF-009]

Requirement description: The application must audit the execution of privileged functions. Configure the application to record registration inputs when privileged functions are executed. At least ensure the specific action performed on the date and time of the event is registered.

Source:

- ✓ **V-222431:** The application must audit the execution of privileged functions. Configure the application to write log entries when privileged functions are executed. At a minimum, ensure the specific action taken, date and time of event are recorded.

Misuse of privileged functions, either intentionally or unintentionally by authorized users, or by unauthorized external entities that have compromised information system accounts, is a serious and ongoing concern and can have significant adverse impacts on organizations. Auditing the use of privileged functions is one way to detect such misuse, and identify the risk from insider threats and the advanced persistent threat. [13]
Priority: Not described
Rationale: The improper use of privileged functions, intentional or unintentional, by authorized users, or by unauthorized external entities that have compromised information system accounts is a serious and continuous concern. It can have significant adverse impacts on organizations. Auditing privileged functions is a way to detect such improper use and identify the risk of internal and advanced persistent threats.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Audit configuration verification:** **1.** Confirm that the application is configured to audit the execution of privileged functions. ✓ **Review of audit records:** **1.** Validate that the application generates event records when privileged functions are executed. ✓ **Minimum registration content:** **1.** Verify that at least the records contain specific details about the action made, as well as the date and time of the event.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The auditor team has identified audit logs and reports in the admin module. Moreover, these audit reports can be filtered by success or not for privileged actions or privileged functions.



Result: It complies.

PUID: [SEC-CAT-LMF-010]

Requirement description: The application must generate safe audit records every time a user:

1. Access, create, or update the personal health information (PHI) of a patient/person through the application;
2. cancels the consent directives of a patient/person through the application
3. Access, through the application, to data blocked or masked by instruction of a patient/person;
4. Access, create, or update user registration data in the application.

Source:

- ✓ Security Requirement 37 – Logging Transactions in the EHRi: The EHRi must create a secure audit record each time a user:
  a) Accesses, creates or updates PHI of a patient/person via the EHRi;
  b) Overrides the consent directives of a patient/person via the EHRi;
  c) Accesses, via the EHRi, data that is locked or masked by instruction of a patient/person; or
  d) Accesses, creates or updates registration data on an EHRi user. [12]

- ✓ Security Requirement 41 – Logging Access to PHI in PoS Systems: All PoS systems connected to the EHRi must record in an audit log every instance of a user accessing, updating or archiving PHI. [12]

- ✓ 12.4.1 Event logging: In addition to following the guidance given by ISO/IEC 27002, health information systems processing personal health information should create a secure audit record each time a user accesses, creates, updates or archives personal health information via the system. [8]

- ✓ 12.4.1 Event logging: Control: Event logs recording user activities, exceptions, faults and information security events should be produced, kept and regularly reviewed. A secure audit record must be created for:
  e) records of successful and rejected system access attempts;
  f) records of successful and rejected data and other resource access attempts;
  g) changes to system configuration;

h) use of privileges;
i) use of system utilities and applications;
j) files accessed and the kind of access;
l) alarms raised by the access control system;
m) activation and de-activation of protection systems, such as anti-virus systems and intrusion detection systems;
n) records of transactions executed by users in applications.

When personal health information is updated, a record of the former content of the data and the associated audit record (i.e. who entered the data on what date) should be retained.

Messaging systems used to transmit messages containing personal health information should keep a log of message transmissions (such a log should contain the time, date, origin and destination of the message, but not its content). [8]

Priority: Not described

Rationale: Safe audit records are essential to trace and audit users' activities interacting with personal health information. It guarantees transparency, responsibility, and security of transactions and access to sensitive information.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Transaction records:**

  1. Verify that the application generates safe audit records for each instance where a user accesses, believes, or updates the Phi of a patient/person.
  2. Confirm that records include details of canceling consent directives and access to blocked or masked data.
  3. Validate the creation of insurance audit records for access activities creation or update of user registration data.

- ✓ **PHI access records (pos Systems):**

  1. Confirm that all post-connected systems to the application register in an audit register each instance where a user accesses, updates, or archives PHI.

- ✓ **Compliance with ISO/IEC 27002:**

  1. Verify that the application complies with the guidelines ISO/IEC 27002 provided for creating event records.
  2. Validate that safe audit records are generated for successful and rejected records of access to the system and resources, changes in system configuration, use of privileges, profits and system applications, accessed files, alarms, and user transactions.

- ✓ **Phi update:**

  1. Confirm that, when updating personal health information, a record of the previous content of the data and the associated audit record is retained.

- ✓ **Message record (in messaging systems):**

  1. Validate that the messaging systems used to transmit messages containing Phi maintain registration of the transmissions, including the time, date, origin, and destination of the message (not including its content).

Requested by: The organization

Responsible: Developer

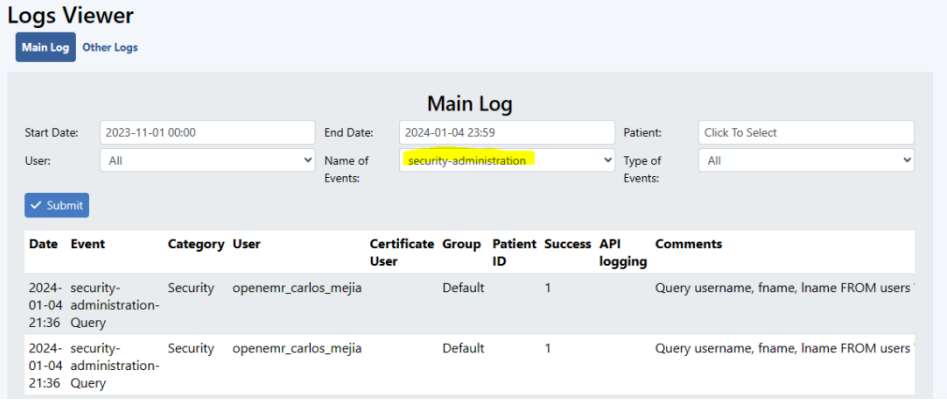
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The auditor team has identified audit logs and reports in the admin module. Moreover, these audit reports can be filtered by success or in many categories in the "Name of Events" option.



Result: It complies.

PUID: [SEC-CAT-LMF-011]

Requirement description: The application must generate audit records that show the start and end of the user's access time to the system and events related to the beginning, closure, and duration of sessions.

Source:

- ✓ **V-222464:** The application must generate audit records showing starting and ending time for user access to the system. Configure the application or application server to record the start and end time of user session activity. [13]

- ✓ **V-222445:** The application must provide audit record generation capability for session timeouts. Configure the application to record session timeout events in the logs. [13]

Priority: Not described

Rationale: Precise user sessions are essential for system monitoring and safety. Capturing the start and completion time of the sessions and session expiration events helps identify suspicious activities and contributes to the system's integrity.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Session starting and completion registration verification:**

  1. Confirm that the application generates audit records that show the start and termination of user access to the system.

- ✓ **Evaluation of generation of registrations for expiration of sessions:**

  1. Validate that the application can generate audit records for session expiration events.

Requested by: The organization

Responsible: Developer

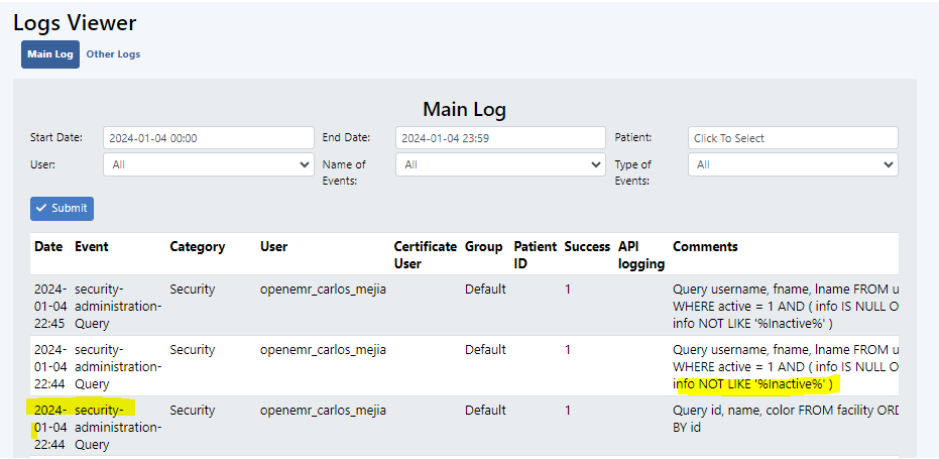
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The auditor team has identified audit logs and reports in the admin module. Moreover, these audit reports have registered events related to the beginning and closure of sessions.



Result: It complies.

PUID: [SEC-CAT-LMF-012]

Requirement description: The application must initiate the audit of sessions at the beginning. Configure the application to start registering application events as soon as the application starts.

Source:

- ✓ **V-222468:** The application must initiate session auditing upon startup. Configure the application to begin logging application events as soon as the application starts up. [13]

Priority: Not described

Rationale: Starting the audit of sessions ensures that applied events are recorded from the beginning, providing a complete and continuous vision of system activities. It is crucial for the security and effective monitoring of the application.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Session Audit Start Verification:**

  1. Confirm that the application begins the audit of sessions as soon as the application starts.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The auditor team has proven that the audit is lifted jointly with the HTTP service. Consequently, the application is started as soon as the application is deployed.

Result: It complies.

PUID: [SEC-CAT-LMF-013]

Requirement description: The application must register closing events of the application. Configure the application or application server to record the closing events of the application in the event records.

Source:

- ✓ **V-222469:** The application must log application shutdown events. Configure the application or application server to record application shutdown events in the event logs. [13].

Priority: Not described
Rationale: The registration of closing events of the application is essential for the monitoring and auditing system operations. It provides a trace of events related to the closure of the application, which can be crucial for the identification of problems and the response to incidents.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Verification of the record of the closing events:** **1.** Confirm that the application records closing events effectively. ✓ **System configuration:** **1.** Validate that the application or server is configured to register closing events in event records.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The auditor team has proven that it is recorded in the Apache server log when there is a stop and the start of the service or when an interruption generates inhibition or unavailability. Consecutively, the requirement is met. **Result:** It complies.

PUID: [SEC-CAT-LMF-014]
Requirement description: The application must generate audit records when successful or failed login attempts occur
Source: ✓ **V-222462:** The application must generate audit records when successful/unsuccessful logon attempts occur. Configure the application or application server to write a log entry when successful and unsuccessful logon events occur. [13] ✓ **V-222476:** The application must produce audit records that contain information to establish the outcome of the events. Configure the application to include the outcome of application functions or events. [13]

- ✓ **§ 164.308** Administrative safeguards.
  (a) A covered entity or business associate must, in accordance with § 164.306:
  (5)
  (ii) Implementation specifications. Implement:
  (d) Log-in monitoring (Addressable)
  Procedures for monitoring log-in attempts and reporting discrepancies. [10]
- ✓ **9.4.2** Secure log-on procedures:
  d) log unsuccessful and successful attempts. [8]

- ✓ **A09:2021 – Security Logging and Monitoring Failures**

  - Ensure all login, access control, and server-side input validation failures can be logged with sufficient user context to identify suspicious or malicious accounts and held for enough time to allow delayed forensic analysis. [3]

Priority: Not described

Rationale: The login events record is critical for the security and effective application monitoring. It provides a detailed trace of access attempts, allowing the early detection of possible threats and the appropriate response.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Generation of login audit records:**

  1. Verify that the application generates audit records when login attempts occur, whether successful or failed.

- ✓ **Inclusion of results information in records:**

  1. Confirm that audit records contain information established by the result of events, including application functions or events.

- ✓ **Login monitoring implementation:**

  1. Validate those procedures to monitor login attempts and report discrepancies have been implemented, complying with the specifications of § 164,308 (a) (5) (ii) (d) and 9.4.2 (d).

- ✓ **Compliance with A09: 2021 specifications:**

  1. Ensure that all login failures, access control, and input validation of the server side can be registered with sufficient user context to identify suspicious or malicious accounts and that these records are retained for the necessary time to allow a deferred forensic analysis.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

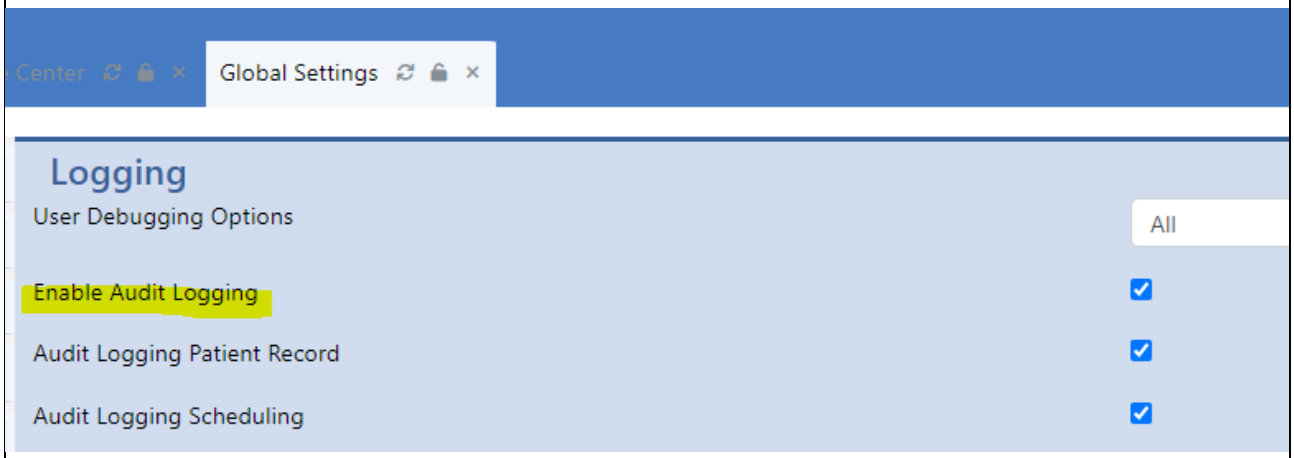
Version history: v1.0

Author and date:

Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The auditor team has proven that the global system configuration option allows audit logging. Moreover, in the official documentation of the OpenEmr webpage related to "3.1 Auditing in OpenEMR", auditing successful and unsuccessful logging attempts is possible. Consequently, the requirement is met.



Result: It complies.

PUID: [SEC-CAT-LMF-015]

Requirement description: The application must ensure that audit records contain information that establishes the following:

1. When using centralized logging; the application must include a unique identifier in order to distinguish itself from other application logs
2. What type of event occurred;
3. The function performed by the accessing user;
4. When the event occurred with a time stamp mark that meets a granularity of one second for a minimum degree of precision
5. Where the event occurred;
6. The organization of the accessing user (at least in those cases where an individual accesses information on behalf of more than one organization);
7. Source of the event that establishes the identity of any individual or process associated with the event;
8. network addresses and protocols
9. destination IP addresses
10. device identity or location, if possible, and system identifier;
11. The outcome of the event; and
12. Identity of any individuals, subjects, or objects/entities associated with the event using the username or user ID of the user related to the event
13. The role the user is exercising;

14. In the case of access override to blocked or masked records or portions of records, a reason for the override, as chosen by the user making the access and
15. In the case of changes to consent directives made by a substitute decision-maker, the identity of the decision-maker.

Source:

- ✓ **5.28 Collection of evidence**: The organization should establish and implement procedures for the identification, collection, acquisition and preservation of evidence related to information security events. [7]

- ✓ Consideration ID-15 – Consider the Definition of Minimum Requirements and Specifications for Audit and Logging Capabilities to be Used by Member Organizations: Member organizations should consider use of common logging and auditing functions to facilitate trust relationships and to meet accountability obligations. [12]

- ✓ **V-222474**: The application must produce audit records containing enough information to establish which component, feature or function of the application triggered the audit event. Configure the application to log which component, feature or functionality of the application triggered the event. [13]

- ✓ **Security Requirement 42** – Minimum Content of Audit Logs: The EHRi audit log and the audit logs of PoS systems connecting to the EHRi must contain:

  a) The user ID of the accessing user;
  b) The role the user is exercising;
  c) The organization of the accessing user (at least in those cases where an individual accesses information on behalf of more than one organization);
  d) The patient ID of the data subject (patient/person);
  e) The function performed by the accessing user;
  f) A timestamp;
  g) In the case of access override to blocked or masked records or portions of records, a reason for the override, as chosen by the user making the access; and
  h) In the case of changes to consent directives made by a substitute decision-maker, the identity of the decision-maker. [12]

- ✓ **12.4.1 Event logging: Control**: Event logs recording user activities, exceptions, faults and information security events should be produced, kept and regularly reviewed.
  The audit log should uniquely identify the user, uniquely identify the data subject (i.e. the subject of care), identify the function performed by the user (record creation, access, update, etc.), and note the time and date at which the function was performed.
  Event logs should include, when relevant:
  a) user IDs;
  b) system activities;
  c) dates, times and details of key events, e.g. log-on and log-off;
  d) device identity or location if possible and system identifier;
  k) network addresses and protocols;[8]

- ✓ **AU-3 CONTENT OF AUDIT RECORDS**

Control: Ensure that audit records contain information that establishes the following:
a. What type of event occurred;
b. When the event occurred;
c. Where the event occurred;
d. Source of the event;
e. Outcome of the event; and
f. Identity of any individuals, subjects, or objects/entities associated with the event. [11]

- ✓ **AU-8 TIME STAMPS**
  Control:
  a. Use internal system clocks to generate time stamps for audit records; and
  b. Record time stamps for audit records that meet [Assignment: organization-defined granularity of time measurement] and that use Coordinated Universal Time, have a fixed local time offset from Coordinated Universal Time, or that include the local time offset as part of the time stamp. [11]

- ✓ **V-222446:** The application must record a time stamp indicating when the event occurred. Configure the application to record the time the event occurred when recording the event. [13]

- ✓ **V-222498:** The application must record time stamps for audit records that can be mapped to Coordinated Universal Time (UTC) or Greenwich Mean Time (GMT). Configure the application to use the underlying system clock that maps to relevant UTC or GMT time zone. [13]

- ✓ **V-222497:** The applications must use internal system clocks to generate time stamps for audit records. Configure the application to use the hosting systems internal clock for audit record generation. [13]

- ✓ **V-222499:** The application must record time stamps for audit records that meet a granularity of one second for a minimum degree of precision. Configure the application to leverage the underlying operating system as the time source when recording time stamps or design the application to ensure granularity of 1 second as the minimum degree of precision. [13]

- ✓ **V-222449:** The application must record the username or user ID of the user associated with the event. Configure the application to record the user ID of the user responsible for the log event entry.

  When users conduct activity within an application, that user's identity must be recorded in the audit log. Failing to record the identity of the user responsible for the activity within the application is detrimental to forensic analysis. [13]

- ✓ **V-222473:** The application must produce audit records containing information to establish when (date and time) the events occurred. Configure the application or application server to include the date and the time of the event in the audit logs. [13]

- ✓ **V-222477**: The application must generate audit records containing information that establishes the identity of any individual or process associated with the event. Configure

the application to log the identity of the user and/or the process associated with the event. [13].

- ✓ **Security Requirement 42** – Minimum Content of Audit Logs: The EHRi audit log and the audit logs of PoS systems connecting to the EHRi must contain:

  a) The user ID of the accessing user;
  b) The role the user is exercising;
  c) The organization of the accessing user (at least in those cases where an individual accesses information on behalf of more than one organization);
  d) The patient ID of the data subject (patient/person);
  e) The function performed by the accessing user;
  f) A timestamp;
  g) In the case of access override to blocked or masked records or portions of records, a reason for the override, as chosen by the user making the access; and
  h) In the case of changes to consent directives made by a substitute decision-maker, the identity of the decision-maker. [12]

- ✓ 12.4.1 Event logging: Control: Event logs recording user activities, exceptions, faults and information security events should be produced, kept and regularly reviewed.
  The audit log should uniquely identify the user, uniquely identify the data subject (i.e. the subject of care), identify the function performed by the user (record creation, access, update, etc.), and note the time and date at which the function was performed.
  Event logs should include, when relevant:
  a) user IDs;
  b) system activities;
  c) dates, times and details of key events, e.g. log-on and log-off;
  d) device identity or location if possible and system identifier;
  k) network addresses and protocols;[8]

- ✓ **V-222470**: The application must log destination IP addresses.

  The IP addresses of the systems that the application connects to are an important aspect of identifying application network related activity. Recording the IP addresses of the system the application connects to in the application logs provides forensic evidence and aids in investigating and correlating the sources of malicious behavior related to security events. Logging this information can be particularly useful for Service-Oriented Applications where there is application to application connectivity.

  Configure the application to record the destination IP address of the remote system. [13]

- ✓ **V-222475**: When using centralized logging; the application must include a unique identifier in order to distinguish itself from other application logs. Configure the application logs or the centralized log storage facility so the application name and the hosts hosting the application are uniquely identified in the logs. [13]

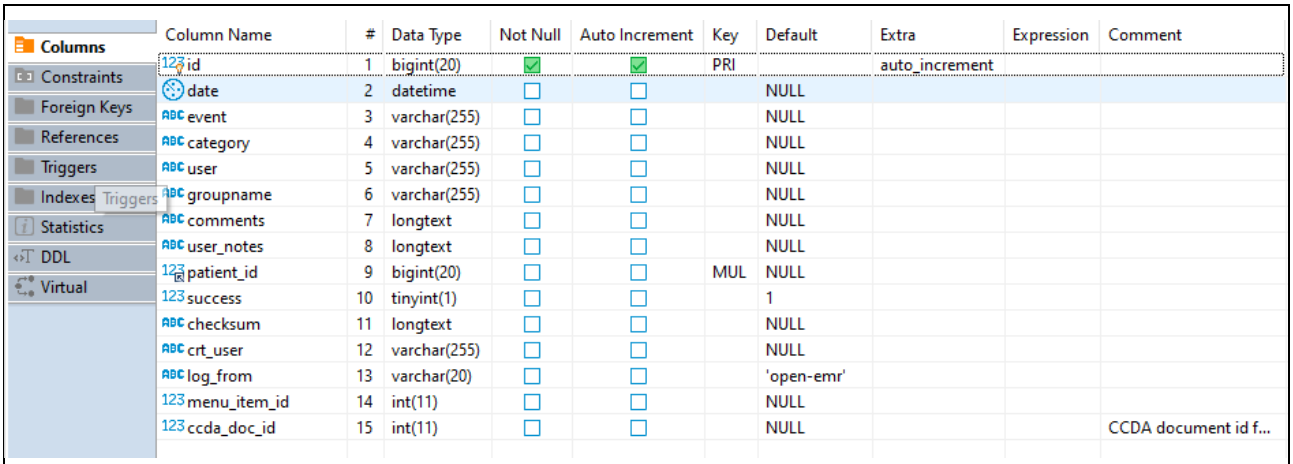
Priority: Not described

Rationale: The generation and registration of audit events are crucial for effective supervision and system action traceability. Guarantee the inclusion of detailed information in audit records. Help in the early detection of unusual events and support forensic research.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Generation and registration of audit events:** **1.** Confirm that the application generates audit records that include the specified information. **2.** Validate that each record contains precise details about the type, moment, place, origin, result, and associated identity. ✓ **Time brand record:** **1.** Verify that the application uses internal system watches to generate time marks in audit records. **2.** Confirm that time brands meet the granularity defined by the organization and use coordinated universal time (UTC) or average time of Greenwich (GMT). ✓ **Specific information record:** **1.** Confirm that the application records the specific information, such as the username, user or ID, IP destination, component, and system function, among others, according to the detailed specifications. ✓ **Safety in the centralized registry:** **1.** Verify that, in case of centralized registration, the application includes a unique identifier to distinguish other application records. ✓ **Compliance with registration specifications:** **1.** Validate that audit records meet the minimum requirements established by the organization, including user identification, user function, date, and time, among others.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The auditor team has proven that the global system configuration option allows audit logging. Moreover, the following statements are not complied with: 8. network addresses and protocols 9. destination IP addresses 10. device identity or location, if possible, and system identifier;



Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra	Expression	Comment
id	1	bigint(20)	☑	☑	PRI		auto_increment		
date	2	datetime	☐	☐		NULL			
event	3	varchar(255)	☐	☐		NULL			
category	4	varchar(255)	☐	☐		NULL			
user	5	varchar(255)	☐	☐		NULL			
groupname	6	varchar(255)	☐	☐		NULL			
comments	7	longtext	☐	☐		NULL			
user_notes	8	longtext	☐	☐		NULL			
patient_id	9	bigint(20)	☐	☐	MUL	NULL			
success	10	tinyint(1)	☐	☐		1			
checksum	11	longtext	☐	☐		NULL			
crt_user	12	varchar(255)	☐	☐		NULL			
log_from	13	varchar(20)	☐	☐		'open-emr'			
menu_item_id	14	int(11)	☐	☐		NULL			
ccda_doc_id	15	int(11)	☐	☐		NULL			CCDA document id f...

Result: It doesn't comply.

PUID: [SEC-CAT-LMF-016]

Requirement description: The application must back up the audit records at least every seven days in a system or component different from the one being audited.

Source:

- ✓ **V-222506**: The application must back up audit records at least every seven days onto a different system or system component than the system or component being audited.

  Configure application backup settings to backup application audit logs every 7 days. [13]

Priority: Not described

Rationale: The periodic support of audit records is essential to guarantee the integrity and availability of audit information. It helps preserve evidence of critical events and mitigate the potential data loss in case of unexpected systems or events.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Backup frequency:**

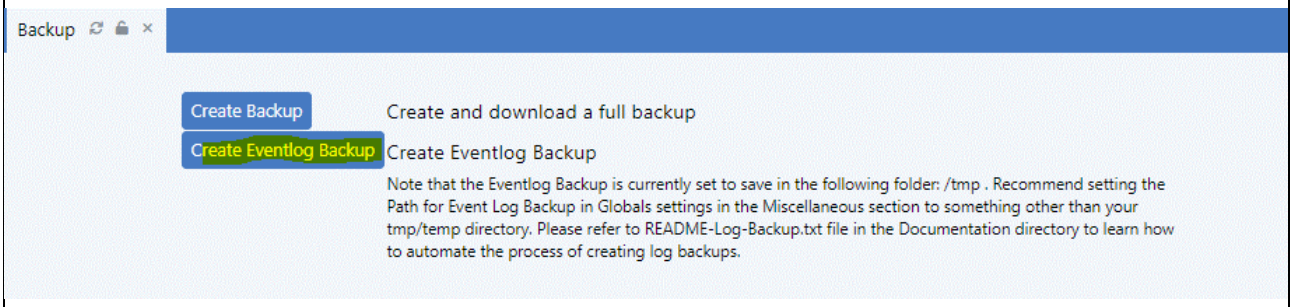
  1. Confirm that the application is configured to support audit records at least once every seven days.

- ✓ **Backup location:**

  1. Verify that backups are stored in a different system or component from the audited system or component.

- ✓ **Backup configuration:**

  1. Check the application backup configuration to ensure you are scheduled to automatically back up the audit records according to the established frequency.

Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The auditor team has proven that an option for backing up the log exists in the platform; however, there are no automatic options for backing up the logs in the system.      **Result:** It doesn't comply.

TIME TO KEEP LOGS

PUID: [SEC-CAT-LMF-017]
Requirement description: The application must maintain audit records for a specific period of time, determined by regulations and medical standards. This retention must cover from 7 to 75 years, depending on the information and legal requirements of the State. Some states may require that medical records be preserved in the hospital to guarantee continuous accessibility.
Source: ✓ Regulations stipulate that various portion of the medical records must be kept from 7 to 75 years depending on the information and state. Some states require that medical records be retained in the hospital so they are always accessible [15] ✓ 12.4.1 Event logging: The organization should carefully assess and determine the retention period for these audit logs, with particular reference to clinical professional standards and legal obligations, in order to enable investigations to be carried out when necessary and to provide evidence of misuse where necessary. [8] ✓ 8.15 Logging: Logs that record activities, exceptions, faults and other relevant events should be produced, stored, protected and analyzed. [7] ✓ Consideration ID-15 – Consider the Definition of Minimum Requirements and Specifications for Audit and Logging Capabilities to be Used by Member Organizations:

Member organizations should consider use of common logging and auditing functions to facilitate trust relationships and to meet accountability obligations. [12]
Priority: Not described
Rationale: Proper retention of audit records is essential to comply with legal and professional standards in the field of health. It allows for conducting research when necessary and serves as evidence in cases of misuse or critical events.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Duration of retention:** **1.** Verify that the application retains audit records according to the periods established by medical and legal regulations, which vary from 7 to 75 years according to the nature of state information and requirements. ✓ **Continuous accessibility:** **1.** Confirm that, when necessary, by state regulations, medical records are preserved in the organization to guarantee continuous accessibility. ✓ **Compliance with professional standards:** **1.** Validate that the retaining duration follows professional standards and legal obligations, as indicated in medical regulations and the 12.4.1 standard of events. ✓ **Complete audit record:** **1.** Verify that the application generates, stores, and protects complete audit records that record activities, exceptions, failures, and other relevant events, as established in consideration 8.15.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The auditor team has no found evidence of politics or information in the documentation about a specific period for conserving log files. **Result:** It doesn't comply.

PUID: [SEC-CAT-LMF-018]

Requirement description: The LOG management application must ensure access and protect against unauthorized modification delete, and guarantee the integrity of audit tools to prevent improper use or commitment.

Source:

- ✓ **Security Requirement 49** – Securing Access to EHRi Audit Logs: The EHRi must secure access to audit records and must safeguard access to system audit tools and audit trails to prevent misuse or compromise. [12]

- ✓ **12.4.2 Protection of log Information**: Audit records shall be secure and tamper-proof. Access to system audit tools and audit trails shall be safeguarded to prevent misuse or compromise. [8]

- ✓ **V-222504**: The application must protect audit tools from unauthorized modification.

  Configure the application to protect audit tools from unauthorized modifications. Limit users to roles that are assigned the rights to edit or update audit tools and establish file permissions that control access to the audit tools and audit tool capabilities and configuration settings. [13].

- ✓ **V-222505**: The application must protect audit tools from unauthorized deletion.

  Configure the application to protect audit tools from unauthorized deletions.

  Limit users to roles that are assigned the rights to edit or delete audit tools and establish file permissions that control access to the audit tools and audit tool capabilities and configuration settings. [13]

- ✓ **V-222509**: The integrity of the audit tools must be validated by checking the files for changes in the cryptographic hash value. Establish a process to periodically check the audit tool cryptographic hashes to ensure the audit tools have not been tampered with. [13]

- ✓ **A09:2021 – Security Logging and Monitoring Failures:** Ensure log data is encoded correctly to prevent injections or attacks on the logging or monitoring systems. [3]

Priority: Not described

Rationale: Guaranteeing the safety of audit tools is essential to maintain the integrity of audit records and prevent unauthorized access or malicious modifications. It helps maintain audit records' reliability and security monitoring effectiveness.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Protection against unauthorized modifications:**

  1. Verify that the application is configured to protect audit tools against unauthorized modifications.

2. Limit access to roles assigned with rights to edit or update audit tools.
3. Establish file permissions that control access to audit tools and their configurations.

✓ **Protection against unauthorized eliminations:**

1. Verify that the application is configured to protect audit tools against unauthorized eliminations.
2. Limit access to roles assigned with rights to edit or eliminate audit tools.
3. Establish file permissions that control access to audit tools and their configurations.

✓ **Validation of the integrity of audit tools:**

1. Verify that the application establishes a process to validate the integrity of audit tools periodically.
2. Establish a process to verify the cryptographic hash values of audit tools and ensure that they have not been manipulated.

Requested by: The organization

Responsible: Developer

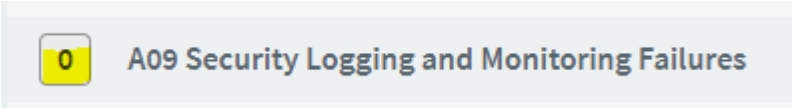
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are no findings for "A09 Security Logging and Monitoring Failures". Moreover, the log is accessed throughout the web application only for view.



Result: It complies.

PUID: [SEC-CAT-LMF-019]

Requirement description: The application must implement strict control measures for the elimination, replication, and consultations of audit records, ensuring total control over the entry, reading, manipulation, or data disposal.

Source:

✓ Deletion, replication and queries must be tightly controlled [15]

✓ Control entry, reading, manipulation or deletion of data [16]

✓ **Security Requirement 49** – Securing Access to EHRi Audit Logs: The EHRi must secure access to audit records and must safeguard access to system audit tools and audit trails to prevent misuse or compromise. [12]

- ✓ **V-222507**: The application must use cryptographic mechanisms to protect the integrity of audit information.

  Configure the application to create an integrity check consisting of a cryptographic hash or one-way digest that can be used to establish the integrity when storing log files. [13]

- ✓ **V-222502**: The application must protect audit information from unauthorized deletion.

  Configure the application to protect audit data from unauthorized deletion. Limit users to roles that are assigned the rights to delete audit data and establish permissions that control access to the audit logs and audit configuration settings [13].

- ✓ **V-222501**: The application must protect audit information from unauthorized modification.

  Configure the application to protect audit data from unauthorized modification and changes. Limit users to roles that are assigned the rights to edit audit data and establish permissions that control access to the audit logs and audit configuration settings. [13]

- ✓ **V-222500**: The application must protect audit information from any type of unauthorized read access.

  Configure the application to protect audit data from unauthorized access. Limit users to roles that are assigned the rights to view, edit or copy audit data, and establish permissions that control access to the audit logs and audit configuration settings. [13]

Priority: Not described

Rationale: The rigorous control of audit records is essential to preserve the integrity and confidentiality of audit information. It avoids unauthorized modifications, ensures the availability of reliable records, and meets the safety requirements for sensitive data protection.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Integrity protection:**

  1. The application should use cryptographic mechanisms to protect the integrity of audit information.
  2. Successful configuration: When storing record files, the application must create integrity control, such as a cryptographic hash or a unidirectional summary.

- ✓ **Protection against unauthorized elimination:**

  1. The application must protect audit information against unauthorized elimination.
  2. Successful configuration: Limit access to roles assigned with rights to eliminate audit data and establish permits that control access to records and audit configuration.

- ✓ **Protection against unauthorized modification:**

  1. The application must protect audit information against unauthorized modifications.

2. Successful configuration: Limit access to roles assigned with rights to edit audit data and establish permits that control access to records and audit configuration.

✓ **Protection against unauthorized access:**

1. The application must protect audit information against any unauthorized access.
2. Successful configuration: limit access to roles assigned with rights to see, edit, or copy audit data and establish permits that control access to records and audit configuration

Requested by: The organization

Responsible: Developer

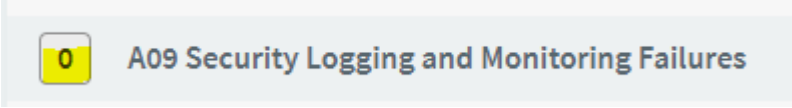
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

A DAST and SAST have been performed, and there are no findings for "A09 Security Logging and Monitoring Failures". However, the log is accessed throughout the web application only for view.



Result: It complies.

PUID: [SEC-CAT-LMF-020]

Requirement description: The application must transfer audit records to a system or medium different from the audited system and be configured to write application records in a centralized repository.

Source:

✓ **V-222481**: The application must off-load audit records onto a different system or media than the system being audited.

Configure the application to off-load audit records onto a different system as per approved schedule. [13]

✓ **V-222482**: The application must be configured to write application logs to a centralized log repository.

Review application documentation and interview application administrator.

Evaluate application log management processes and determine if the system is configured to utilize a centralized log management system for the hosting and management of application audit logs.

If the system is not configured to write the application logs to the centralized log management repository in an expeditious manner, this is a finding.
Priority: Not described
Rationale: Proper audit record management is essential for safety and operational efficacy. Offloading audit records to a separate system and using a centralized repository improves security, integrity, and monitoring capacity.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Audit record transfer:** **1.** Confirm that the application is configured to transfer audit records to a different system according to the approved programming. ✓ **Centralized record repository configuration:** **1.** Review the application documentation and interview the application administrator. **2.** Evaluate application record management processes and determine if the system is configured to use a centralized record management system to administer application audit records. **3.** If the system is not configured to expedite the application records in the centralized record management repository, it is considered a finding.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The auditor team has no found evidence of politics or information in the documentation about transferring audit records to a different system according to the approved programming. **Result:** It doesn't comply.

PUID: [SEC-CAT-LMF-021]
Requirement description: The application must allow the review and centralized analysis of audit records with filtering options defined by the organization
Source: ✓ Auditing information must be collected and intelligently analyzed in order to detect abuses and mistakes [15]

- ✓ **V-222487**: The application must provide the capability to centrally review and analyze audit records from multiple components within the system.

  Configure the application so all of the applications logs are available for review from one centralized location. [13]

- ✓ **V-222488**: The application must provide the capability to filter audit records for events of interest based upon organization-defined criteria.

  Configure the application filters to search event logs based on defined criteria. [13]

- ✓ **A09:2021 – Security Logging and Monitoring Failures:** Ensure that logs are generated in a format that log management solutions can easily consume. [3]

Priority: Not described

Rationale: The effective collection and analysis of audit records are essential to detect undue activities and errors in the system. It provides a critical security layer by allowing the early identification of possible threats and the proactive mitigation of mistakes.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Centralized review capacity:**

  1. Confirm that the application provides the ability to review centrally and analyze multiple component audit records within the system.

- ✓ **Audit record filtering capacity:**

  1. Verify that the application can filter audit records for events of interest according to the criteria defined by the organization.

Requested by: The organization

Responsible: Developer

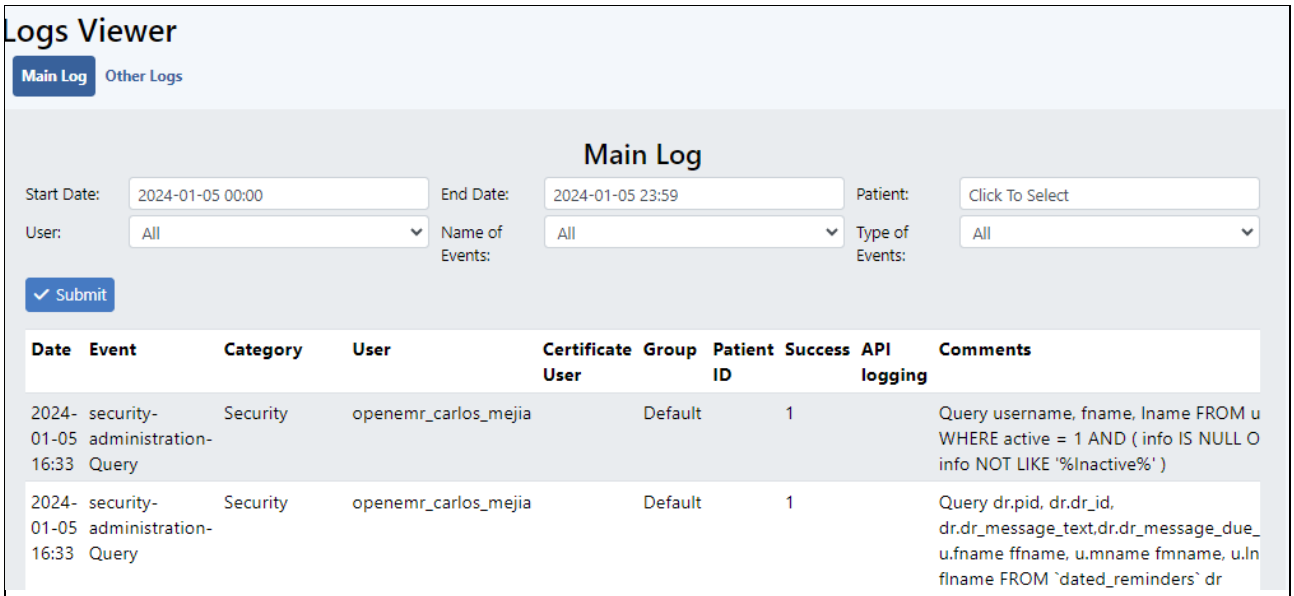
Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The audit team has shown that there is an option within "System -> Logs" for centralized analysis and review of log records with filtering options.



Result: It complies.

MONITORING

PUID: [SEC-CAT-LMF-022]

Requirement description: At least one application administrator must be registered to receive notifications of updates or safety alerts when they are automatically available. Administrators must register to receive update notifications, allowing them to stop and update applications and components of the application.

Source:

- ✓ **V-222669:** At least one application administrator must be registered to receive update notifications, or security alerts, when automated alerts are available. Register administrators to receive update notifications so they can patch and update applications and application components.

  Administrators should register for updates to all COTS and custom-developed software, so when security flaws are identified, they can be tracked for testing and updates of the application can be applied.

  Admin personnel should be registered to receive updates to all components of the application, such as Web Server, Application Servers, and Database Servers. Also, if update notifications are provided for any custom-developed software, libraries or third-party tools, deployment personnel must also register for these updates. [13]

Priority: Not described

Rationale: The effective notification of security updates and alerts is crucial to maintain the integrity and safety of the system. Registering administrators ensure they are informed about the latest updates and can take quick and adequate measures to remedy potential vulnerabilities.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described
Current state: To_be_determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Confirmation of the registration of at least one administrator:** **1.** Verify that at least one application administrator is registered to receive notifications of updates and safety alerts. ✓ **Registration verification for updates of all components:** **1.** Confirm that administrators are registered to receive updates from all application components, such as web servers, application servers, and database servers.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The audit team has shown an administrator registered in the system who can start/stop web servers, databases, and services. **Result:** It complies.

PUID: [SEC-CAT-LMF-023]
Requirement description: The application must generate audit records when a concurrent session is produced from different workstations
Source: ✓ **V-222672:** The application must generate audit records when concurrent logons from different workstations occur. When an application provides users with the ability to concurrently logon, an event must be recorded that indicates the user has logged on from different workstations. It is important to ensure that audit logs differentiate between the two sessions. The event data must include the user ID, the workstation information and application session information that provides the details necessary to determine which application session executed what action on the system Configure the application to log concurrent logons from different workstations. [13]

✓ **V-222476:** The application must produce audit records that contain information to establish the outcome of the events. Configure the application to include the outcome of application functions or events. [13]
Priority: Not described
Rationale: Generating audit records for concurrent access is essential to supervise the application's security and detect possible malicious activities. It provides an additional security layer by identifying the simultaneous session from different locations.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Verification of audit records for concurrent access:** **1.** Confirm the application generates audit records when a concurrent session is produced from different workstations. **2.** Validate those records, including detailed information, such as the user ID, the workstation information, and the application session information. ✓ **Evaluation of the content of the audit records:** **1.** Verify that the application produces audit records that contain information to establish the result of the events or functions of the application. **2.** Confirm that the application is configured to include the result of the functions or events in the audit records.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The audit team previously stated that the requirement [sec-cat-iaf-028] was not met; the requirement was inherent to control the concurrent sessions. Therefore, the current requirement is not completed either. **Result:** It doesn't comply.

PUID: [SEC-CAT-LMF-024]
Requirement description: EHRI must be able to register and safely report each access to the records of a patient/person, complying with the following requirements:

1. Identify all users who have accessed or modified the records of a patient/person in a given period. **2.** Provide functionality to inform for a given user: ✓ The records you can access. ✓ You can access the specific sections of the (the registration (s). ✓ The privileges you have concerning each of these records. **3.** Identify all patients/people whose records have been accessed or modified by a specific user in a given period. **4.** The system must generate a security event in case of an attempt or potential success for violating the login controls.
Source: ✓ **Security Requirement 46** – Reporting Every Access to a Patient/Person's HER: The EHRi must be capable of identifying all users who have accessed or modified a given patient/person's record(s) over a given period of time. [12] ✓ **Security Requirement 62** – Reporting the Access Privileges of a User: The EHRi must, and PoS systems connected to the EHRi should, provide functionality that can report, for a given user: a) Which records the user can access; b) Which portions of the record(s) the user can access; and c) Which privileges (e.g. viewing, modification) the user has with respect to each of these records. [12] ✓ **Security Requirement 47** – Reporting Every Access by a User: The EHRi must be capable of identifying all patients/persons whose records have been accessed or modified by a given user over a given period of time. [12] ✓ **9.4.2 Secure log-on procedures**: g) raise a security event if a potential attempted or successful breach of log-on controls is detected. [8] ✓ **V-222611:** The application must reveal error messages only to the ISSO, ISSM, or SA. Configure the server to only send error messages containing system information or sensitive data to privileged users. Use generic error messages for non-privileged users. [13]
Priority: Not described
Rationale: Establishing a safe and complete access record in EHRI is essential to guarantee transparency and medical information security. Fulfilling these requirements supports the traceability of user actions and facilitates identification and response to possible security problems.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria:

- ✓ **Safe registration and report:**

  1. Confirm that the EHRI safely records each access to the records of a patient/person.
  2. Verify that the report functionality for a given user meets the specified requirements.
  3. Validate EHRI's ability to identify all patients/people whose records have been accessed or modified by a specific user in a given period.

- ✓ **Security in login procedures:**

  1. Verify that a security event is generated in case of attempted or potential success for violating login controls.

- ✓ **Error messages configuration:**

  1. Confirm that the system reveals error messages only to ISSO, ISSM, or SA roles.
  2. Validate that the server is configured to send only privileged user error messages containing system information or sensitive data.
  3. Verify that generic error messages are used for non-privileged users.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The audit team has reviewed the documentation of the official OPENEMR website, and the functionality that allows the requirement in the reporting and log options of the administrator module is available; the data is presented according to what is detailed in the documentation as mentioned above official.

Result: It complies.

PUID: [SEC-CAT-LMF-025]

Requirement description: The application must notify the administrator, audit, and record the modifications made in the application configuration.

Source:

- ✓ **V-222512**: The application must audit who makes configuration changes to the application. Configure the application to create log entries that can be used to identify the user accounts that make application configuration changes. [13]

- ✓ **CM-6 CONFIGURATION SETTINGS**
  Control:
  d. Monitor and control changes to the configuration settings in accordance with organizational policies and procedures.

Discussion: Configuration settings are the parameters that can be changed in the hardware, software, or firmware components of the system that affect the security and privacy posture or functionality of the system. Information technology products for which configuration settings can be defined include mainframe computers, servers, workstations, operating systems, mobile devices, input/output devices, protocols, and applications. Parameters that impact the security posture of systems include registry settings; account, file, or directory permission settings; and settings for functions, protocols, ports, services, and remote connections. [11].
Priority: Not described
Rationale: Auditing configuration changes is essential to guarantee the integrity and safety of the system. Register who makes changes in the configuration allows effective traceability and supports the identification of possible threats or malicious activities.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Generation of audit records:** **1.** Verify that the application is configured to create registration inputs that identify the user accounts responsible for making changes in application configuration. ✓ **Monitoring and control of changes:** **1.** Confirm that the application monitors and controls changes in the configuration in accordance with organizational policies and procedures. ✓ **Relevant parameter registration:** **1.** Validate that audit records include relevant information about changes, such as configuration parameters.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The audit team has reviewed the documentation of the Official Openemr website, and the functionality that allows the configuration and execution of notifications from the administrator module is available. **Result:** It complies.

PUID: [SEC-CAT-LMF-026]

Requirement description: The application must notify the system administrators and system security officers (ISSO) of account enable actions. The application must be configured to send notifications to the system administrator and the ISSO when application accounts are enabled.

Source:

- ✓ **V-222422:** The application must notify System Administrators and Information System Security Officers of account enabling actions. Configure the application to notify the system administrator and the ISSO when application accounts are enabled.

  Once an attacker establishes access to a system, the attacker often attempts to create a persistent method of re-establishing access. One way to accomplish this is for the attacker to simply enable an existing account that has been previously disabled. Notification when account enabling actions occur is one method for mitigating this risk. A comprehensive account management process will ensure an audit trail which documents the enabling of application user accounts and notifies administrators and Information System Security Officers (ISSO) exists. [13].

Priority: Not described

Rationale: The notification of accounts enabling actions is essential for early detection and mitigation of risks associated with unauthorized access. It provides an additional security layer by alerting administrators and the ISSO on possible attempts to establish persistent access to the system through accounts.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Notification of accounts:**

  1. Confirm that the application is configured to notify the administrators of the system and the ISSO every time accounts enable action is carried out.

- ✓ **Audit records:**

  1. Verify that there is a complete audit trace that documents the authorization of application user accounts.

- ✓ **Notification effectiveness:**

  1. Simulate an account authorization action and confirm that notifications are adequately sent to system administrators and ISSO.

Requested by: The organization

Responsible: Developer

Configurable value: Not described

Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea

José A. García-Berná 13/12/2023
Auditory result: The audit team has reviewed the documentation of the Official Openemr website, and the functionality that allows the configuration and execution of account activation notifications from the administrator module is available. In case of notification to the ISSO user, adding it to the distribution list or using the respective message template is enough. **Result:** It complies.

PUID: [SEC-CAT-LMF-027]
Requirement description: The application must notify the system administrators and system information security officers (ISSO) of account deactivation actions. The application configuration must include notification to the system administrator and ISSO when application accounts are deactivated.
Source: ✓ **V-222419:** The application must notify System Administrators and Information System Security Officers of account disabling actions. Configure the application to notify the system administrator and the ISSO when application accounts are disabled. [13]
Priority: Not described
Rationale: The timely notification of account deactivation actions is crucial to maintain the effective safety and supervision of the system. Informing system administrators and ISSO guarantees a quick and adequate response to changes in the status of application accounts, thus strengthening the system's safety position.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To_be_determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Confirmación de la Notificación de Desactivación de Cuentas:** **1.** Verificar que la aplicación genere notificaciones a los Administradores del Sistema e ISSO cuando se realicen acciones de desactivación de cuentas. ✓ **Evaluación de Configuración de Notificación:** **1.** Confirmar que la configuración de la aplicación esté ajustada para notificar automáticamente a los administradores del sistema y a los ISSO cuando se desactiven cuentas de aplicación.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán

Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The audit team has reviewed the documentation of the Official Openemr website, and the functionality that allows the configuration and execution of account deactivation notifications from the administrator module is available. In case of notification to the ISSO user, adding it to the distribution list or using the respective message template is enough. **Result:** It complies.

PUID: [SEC-CAT-LMF-028]
Requirement description: The application must notify the system administrators and the system security officers (ISSO) when application accounts are created.
Source: ✓ **V-222417:** The application must notify System Administrators and Information System Security Officers when accounts are created. Once an attacker establishes access to a system, the attacker often attempts to create a persistent method of re-establishing access. One way to accomplish this is for the attacker to simply create a new account. Notification of account creation is one method for mitigating this risk Configure the application to notify the system administrator and the ISSO when application accounts are created. [13].
Priority: Not described
Rationale: The immediate notification of the creation of application accounts is crucial to mitigate the risk associated with attempts to establish persistent unauthorized access methods. The early identification of the creation of accounts allows a proactive response to prevent unauthorized access.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To_be_determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Confirm account creation notification:** **1.** Verify that the application is configured to notify the system administrators and ISSO when application accounts are created. ✓ **Verification of notification:** **1.** Evaluate that the system administrator and ISSO respond adequately to application account creation notifications according to established policies and procedures.
Requested by: The organization
Responsible: Developer

Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The audit team has reviewed the documentation of the Official Openemr website, and the functionality that allows the configuration and execution of account creation notifications from the administrator module is available. In case of notification to the ISSO user, adding it to the distribution list or using the respective message template is enough. **Result:** It complies.

PUID: [SEC-CAT-LMF-029]
Requirement description: The application must notify the system administrators and system information security officers (ISSO) of account elimination actions. The application configuration must include notification to the system administrator and the ISSO when application accounts are deleted.
Source: ✓ **V-222420:** The application must notify System Administrators and Information System Security Officers of account removal actions. Configure the application to notify the system administrator and the ISSO when application accounts are removed. [13].
Priority: Not described
Rationale: Notification of account elimination actions is essential to maintain adequate visibility on changes in system access levels. Informing system administrators and ISSO about eliminating accounts helps maintain security and take appropriate corrective measures, preventing possible threats.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Notification verification:** **1.** Confirm that the application is configured to notify the administrators of the system and ISSO on accounts elimination actions.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán

Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The audit team has reviewed the documentation of the Official Openemr website, and the functionality that allows the configuration and execution of account elimination notifications from the administrator module is available. In case of notification to the ISSO user, adding it to the distribution list or using the respective message template is enough. **Result:** It complies.

PUID: [SEC-CAT-LMF-030]
Requirement description: The application must notify system administrators and system security officers (ISSO) when user accounts are modified.
Source: ✓ **V-222418:** The application must notify System Administrators and Information System Security Officers when accounts are modified. Configure the application to notify the system administrator and the ISSO when application accounts are modified. [13].
Priority: Not described
Rationale: The application must notify system administrators and security officers (ISSO) when user accounts are modified.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To_be_determined
Verification method: Demonstration/Analysis
Validation criteria: ✓ **Notification confirmation:** **1.** Verify that the application is configured to generate notifications when modifications are made to user accounts. ✓ **Reception by administrators and ISSO:** **1.** Confirm that notifications are sent and received by system administrators and system information security officers. ✓ **Speed of notification:** **1.** Evaluate the speed with which notifications are generated and sent after modifying the user accounts. ✓ **Informative content:** **1.** Verify that notifications contain relevant information on the modifications made to the user accounts.
Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0

Author and date:
Carlos M. Mejía-Granda
José L Fernández-Alemán
Juan Manuel Carrillo-de-Gea
José A. García-Berná
13/12/2023

Auditory result:

The audit team has reviewed the documentation of the Official Openemr website, and the functionality that allows the configuration and execution of account modifying notifications from the administrator module is available. In case of notification to the ISSO user, adding it to the distribution list or using the respective message template is enough.

Result: It complies.

PUID: [SEC-CAT-LMF-031]

Requirement description: The application must provide functions to analyze records and audit traces to identify all users who have accessed or modified patient/people records during a certain period.

Source:

- ✓ **Security Requirement 48** – Analyzing EHRi Audit Logs for Patients/Persons at Elevated Risk: The EHRi must provide functions for analyzing logs and audit trails to allow the identification of all users who have accessed or modified such record(s) over a given period of time. [12]
- ✓ **§ 164.312 Technical safeguards**.
  (b) Standard: Audit controls.
  Implement hardware, software, and/or procedural mechanisms that record and examine activity in information systems that contain or use electronic protected health information. [10]

Priority: Not described

Rationale: This requirement guarantees the application's capacity to perform a detailed analysis of the audit records, which is crucial for identifying and responding to situations where the information of patients/people at high risk may have been accessed or modified inappropriately.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To be determined

Verification method: Demonstration/Analysis

Validation criteria:

- ✓ **Implementation of audit controls:**

  1. Verify that the application has implemented audit mechanisms, as established in the standard § 164,312 (a).

- ✓ **Record analysis functionality:**

  1. Confirm that the application provides specific functions to analyze records and audit traces.

Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The audit team has reviewed the administration module options reports and logging. It corroborates that you have functions for identifying all users who have accessed or modified patients'/people's records during the specific period. **Result:** It complies.

ERROR HANDLING

PUID: [SEC-CAT-LMF-032]
Requirement description: The application must alert the Information Security Officer (ISSO) and the System administration (SA) in case of a failure in audit processing. The application must be configured to send an alarm when the audit system fails or is experiencing failure.
Source: ✓ **V-222485**: The application must alert the ISSO and SA (at a minimum) in the event of an audit processing failure. Configure the application to send an alarm in the event the audit system has failed or is failing. [13].
Priority: Not described
Rationale: The immediate alert in the face of failures in audit processing is essential to guarantee the integrity and effectiveness of the audit system. Notifying the ISSO and SA allows a quick response and the implementation of corrective measures to preserve the security of the application.
Child PUIDs: Not described
Parent PUIDs: Not described
Exclusion PUIDs: Not described
Critical nature: Not described
Current state: To be determined
Verification method: Demonstration/Analysis
Validation criteria: **1.** Confirm that the application is configured to generate alerts in case of failure in audit processing. **2.** Verify that alerts are at least addressed to ISSO and SA. **3.** Perform failure simulation tests in audit processing and confirm that the corresponding alerts are received. **4.** Validate that the alert system is operational and can be tested as necessary.
Requested by: The organization
Responsible: Developer
Configurable value: Not described

Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: The audit team has reviewed the official site documentation for Openemr and finds no options to notify the ISSO and SA of failures in the event auditing process. **Result:** It doesn't comply.

2.9.3.10 Server-Side Request Forgery (SSRF)

PUID: [SEC-CAT-SRF-001]
Requirement description: The application must implement protection measures to prevent SSRF vulnerabilities, including network and application layer controls.
Source: ✓ **A10:2021 – Server-Side Request Forgery (SSRF):** SSRF flaws occur whenever a web application is fetching a remote resource without validating the user-supplied URL. It allows an attacker to coerce the application to send a crafted request to an unexpected destination, even when protected by a firewall, VPN, or another type of network access control list (ACL). As modern web applications provide end-users with convenient features, fetching a URL becomes a common scenario. As a result, the incidence of SSRF is increasing. Also, the severity of SSRF is becoming higher due to cloud services and the complexity of architectures. Suggestions: 1. Implement From Network layer for: o Segment remote resource access functionality in separate networks to reduce the impact of SSRF o Enforce "deny by default" firewall policies or network access control rules to block all but essential intranet traffic. Hints: ~ Establish an ownership and a lifecycle for firewall rules based on applications. ~ Log all accepted and blocked network flows on firewalls 2. Implement From Application layer for: o Sanitize and validate all client-supplied input data o Enforce the URL schema, port, and destination with a positive allow list o Do not send raw responses to clients

- Disable HTTP redirections
- Be aware of the URL consistency to avoid attacks such as DNS rebinding and "time of check, time of use" (TOCTOU) race conditions

3. Do not mitigate SSRF via the use of a deny list or regular expression. Attackers have payload lists, tools, and skills to bypass deny lists [3]

✓ **CWE-918: Server-Side Request Forgery (SSRF)**

The web server receives a URL or similar request from an upstream component and retrieves the contents of this URL, but it does not sufficiently ensure that the request is being sent to the expected destination. By providing URLs to unexpected hosts or ports, attackers can make it appear that the server is sending the request, possibly bypassing access controls such as firewalls that prevent the attackers from accessing the URLs directly. The server can be used as a proxy to conduct port scanning of hosts in internal networks, use other URLs such as that can access documents on the system (using file://), or use other protocols such as gopher:// or tftp://, which may provide greater control over the contents of requests. It can cause XSPA:Cross Site Port Attack [14]

Priority: Not described

Rationale: SSRF vulnerabilities can be exploited to send unwanted requests to unexpected destinations, even when firewalls or other network access controls protect the application. Implementing protection measures from the network and application layers is crucial to mitigate these risks.

Child PUIDs: Not described

Parent PUIDs: Not described

Exclusion PUIDs: Not described

Critical nature: Not described

Current state: To_be_determined

Verification method: Demonstration/Analysis

Validation criteria:

✓ **Network layer:**

1. Segment the functionality of access to remote resources in separate networks to reduce the impact of SSRF.
2. Apply Firewall policies or network access control rules that follow the policy "deny by default" to block all traffic except the essential intranet.
3. Establish the property and life cycle of the application-based firewall rules.
4. Register all network flows accepted and blocked in Firewalls.

✓ **Application layer:**

1. Sanitize and validate all the input data provided by the client.
2. Apply a positive white list for the URL's scheme, port, and destination.
3. Avoid sending without processing customers.
4. Disable HTTP redirections.
5. Maintain coherence in the URLs to avoid attacks such as DNS Rebinding and career conditions "Time of Check, Time of Use."

✓ **Additional considerations:**

1. Do not mitigate SSRF through denial lists or regular expressions since attackers can use payload lists, tools, and skills to avoid these denial lists.

Requested by: The organization
Responsible: Developer
Configurable value: Not described
Version history: v1.0
Author and date: Carlos M. Mejía-Granda José L Fernández-Alemán Juan Manuel Carrillo-de-Gea José A. García-Berná 13/12/2023
Auditory result: A DAST and SAST has been performed, and there are no findings for "A10 Server-Side Request Forgery". Consequently, the requirement is fulfilled. **Result:** It complies

2.9.4 Maintainability

Not described.

2.9.5 Portability

Not described.

3. Verification

- ✓ SAST results (https://github.com/cmejia5486/srs/tree/main/Scan%20results/SAST)
- ✓ DAST results (https://github.com/cmejia5486/srs/tree/main/Scan%20results/DAST)
- ✓ SAST report (https://github.com/cmejia5486/srs/tree/main/Scan%20Reports/SAST)
- ✓ DAST report (https://github.com/cmejia5486/srs/tree/main/Scan%20Reports/DAST)

4. Supporting Information

- ✓ Standard review details (https://github.com/cmejia5486/srs/tree/main/standard%20review)
- ✓ Literature review file (https://github.com/cmejia5486/srs/tree/main/Literature%20review)

5. References

[1] "IEEE SA - IEEE/ISO/IEC 29148-2018." [Online]. Available: https://standards.ieee.org/ieee/29148/6937/
[2] "IEEE SA - IEEE 830-1998." [Online]. Available: https://standards.ieee.org/ieee/830/1222/
[3] "Home - OWASP Top 10:2021." [Online]. Available: https://owasp.org/Top10/en/

[4] A. Toval, J. Nicolás Ros, B. Moros Valle, and F. Garcia, "Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach," *Requir. Eng.*, vol. 6, pp. 205–219, Jan. 2002, doi: 10.1007/PL00010360.
[5] "IEEE Guide for Developing System Requirements Specifications," *IEEE Std 1233, 1998 Edition*, pp. 1–36, 1998, doi: 10.1109/IEEESTD.1998.88826.
[6] "ISO/IEC 27001 Standard – Information Security Management Systems." [Online]. Available: https://www.iso.org/standard/27001
[7] "ISO/IEC 27002:2022 - Information security, cybersecurity and privacy protection — Information security controls." [Online]. Available: https://www.iso.org/standard/75652.html
[8] "ISO 27799:2016 - Health informatics — Information security management in health using ISO/IEC 27002." [Online]. Available: https://www.iso.org/standard/62777.html
[9] "Health Insurance Portability and Accountability Act of 1996 | ASPE." Accessed: Nov. 10, 2022. [Online]. Available: https://aspe.hhs.gov/reports/health-insurance-portability-accountability-act-1996
[10] "eCFR :: 45 CFR Part 164 Subpart C -- Security Standards for the Protection of Electronic Protected Health Information." Accessed: Sep. 27, 2023. [Online]. Available: https://www.ecfr.gov/current/title-45/subtitle-A/subchapter-C/part-164/subpart-C
[11] J. T. Force, "Security and Privacy Controls for Information Systems and Organizations," Jul. 2020, doi: 10.6028/NIST.SP.800-53R5.
[12] "Privacy and Security Requirements and Considerations for Digital Health Solutions | Canada Health Infoway." [Online]. Available: https://www.infoway-inforoute.ca/en/component/edocman/resources/technical-documents/architecture/2154-privacy-and-security-requirements-and-considerations-for-digital-health-solutions
[13] "Application Security and Development Security Technical Implementation Guide." [Online]. Available: https://www.stigviewer.com/stig/application_security_and_development/
[14] "Top 25 Software Errors | SANS Institute." [Online]. Available: https://www.sans.org/top25-software-errors/
[15] D. L. Hamilton, "Identification and evaluation of the security requirements in medical applications," in *[1992] Proceedings Fifth Annual IEEE Symposium on Computer-Based Medical Systems*, 1992, pp. 129–137. doi: 10.1109/CBMS.1992.244954.
[16] H. J. Baur, U. Engelmann, F. Saurbier, A. Schröter, U. Baur, and H. P. Meinzer, "How to deal with security issues in teleradiology," *Comput Methods Programs Biomed*, vol. 53, no. 1, pp. 1–8, 1997, doi: https://doi.org/10.1016/S0169-2607(96)01798-1.
[17] A. Strielkina, O. Illiashenko, M. Zhydenko, and D. Uzun, "Cybersecurity of healthcare IoT-based systems: Regulation and case-oriented assessment," in *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, 2018, pp. 67–73. doi: 10.1109/DESSERT.2018.8409101.
[18] "Home | Canada Health Infoway." [Online]. Available: https://www.infoway-inforoute.ca/en/