Australian Government
Australian Digital Health Agency

My Health Record Connecting Systems Security Conformance Profile

23 July 2024

v1.0

Approved for external use

Document ID: **DH-3981:2024**

Australian Digital Health Agency ABN 84 425 496 912, Level 25, 175 Liverpool Street, Sydney, NSW 2000
Telephone 1300 901 001 or email help@digitalhealth.gov.au
www.digitalhealth.gov.au

Acknowledgements
The Australian Digital Health Agency is jointly funded by the Australian Government and all state and territory governments.

Disclaimer
The Australian Digital Health Agency ("the Agency") makes the information and other material ("Information") in this document available in good faith but without any representation or warranty as to its accuracy or completeness. The Agency cannot accept any responsibility for the consequences of any use of the Information. As the Information is of a general nature only, it is up to any person using or relying on the Information to ensure that it is accurate, complete and suitable for the circumstances of its use.

Document control
This document is maintained in electronic form and is uncontrolled in printed form. It is the responsibility of the user to verify that this copy is the latest revision.

OFFICIAL

Document information

Key information

Owner	Connected Care Branch, Digital Solutions Division
Contact for enquiries	Australian Digital Health Agency Help Centre Phone 1300 901 001 Email help@digitalhealth.gov.au

Product or document version history

Product or document version	Date	Release comments
v1.0	23 July 2024	First release

Table of contents

1 Introduction

The Agency is cognisant of the cyber security risks posed by systems accessing the My Health Record system, as well as potentially vulnerable aspects of the national infrastructure and all services under its care. To address this risk, a set of security requirements for systems connecting to the My Health Record system have been identified. The controls that are most relevant to the development of software for healthcare organisations, have been derived from the Australian Cyber Security Centre's (ACSC) Information Security Manual (ISM) [ACSC2023a].

The focus of this conformance profile is on incorporating the security control functionalities within software systems that are connected to the My Health Record system either directly or indirectly. This conformance profile is intended to set a minimum standard or baseline level of cyber security that is expected of connecting systems, and that is consistently adopted. The requirements in this conformance profile are intended to strike an appropriate balance between strengthening the cyber security posture of all connecting systems and minimising potential impacts on software providers and overall system participation. In doing so, this conformance profile supports the overarching goals of improving security within healthcare software systems and fostering a secure and trusted healthcare ecosystem.

Implementers should refer to the My Health Record System Conformance Assessment Scheme (CAS) [AGENCY2024] for information about declaring conformance to this conformance profile.

1.1 Purpose

This document lists the security requirements that are applicable for healthcare software systems integrating with the My Health Record system.

The profile includes two main sections:

1. Conformance requirements that apply to the healthcare software system
2. Compliance requirements that apply to the software provider organisation.

1.2 Intended audience

The intended audience includes:

- Software developers
- Software provider organisations
- Healthcare organisations and providers

1.3 Scope

This document contains both conformance and compliance requirements that are to be applied to connecting systems that access the My Health Record system directly and indirectly via the Business-to-Business (B2B) gateway services.

2 Security framework

The security framework applicable to this conformance profile is the ISM. As described in the ISM, the ISM is a security manual "*to outline a cyber security framework that an organisation can apply, using their risk management framework, to protect their systems and data from cyber threats.*" [ACSC2023a].

The ISM is general and broad in nature and not all controls can be applied to all software. Of the controls that can be applied to software, the Agency has identified controls that provide direct benefits without imposing unreasonable imposts on the software product.

It is acknowledged that the ISM is updated periodically, and the alignment to ISM controls within this profile is based on a specific version indicated in the document references. Software providers are encouraged to adopt the latest and/or stronger controls in future ISM releases if they choose, provided this does not introduce conflicts with the requirements in this profile. For the avoidance of doubt, the relaxing of controls in a future ISM release is considered a conflict, in which case the requirements specified in this profile continue to prevail.

3 Conformance Requirements

This section lists the security conformance requirements for healthcare software systems. For a healthcare software system to be considered conformant, it must meet the mandatory requirements and all relevant conditional requirements. Conditional requirements are deemed mandatory if the specific conditions are met for the implementation.

While conformance to the recommended requirements is not mandated, it is advisable for healthcare software systems to implement recommended requirements where possible, as these requirements may become mandatory in future releases.

The conformance requirements apply to healthcare software systems which may be a single software or an integration of multiple sub-systems. Integrated multiple sub-systems include systems that interface with the My Health Record system through one or more middleware components.

Given the diverse integration methods of healthcare software with the My Health Record system, it is not expected that a single connecting system alone will fulfill all the requirements outlined in this conformance profile. It is likely that multiple software systems will collaboratively meet the requirements defined in this profile during implementation.

Requirements follow a standard form, utilising the following language:

SHALL: When appearing in a conformance requirement, the verb SHALL indicates a mandatory requirement. Its negative form SHALL NOT indicates a prohibition.
SHOULD: When appearing in a conformance requirement, the verb SHOULD indicates a recommendation. Its negative form SHOULD NOT indicates an option that should not be supported.

Note: the unique numbering of each requirement contained within this section of the profile is not intended to be sequential. Further, any gaps in numbering are intentional and inconsequential. Requirements have been grouped according to the categories of controls outlined in the ISM or a suitable equivalent.

3.1 Authentication hardening

This section includes the requirements for strengthening the system authentication processes used to grant system access. Examples of authentication methods that can be used in multi-factor authentication include passwords, one-time SMS codes, one-time password applications, universal 2nd Factor security keys, physical one-time password tokens, biometrics (such as fingerprint or face identification), and smartcards.

SEC-0081	**Session timeout** The healthcare software system SHALL provide the session timeout functionality to automatically log off an account from the system or require re-authentication after a period of inactivity. The default period of inactivity SHALL be no longer than 15 minutes. If a period of inactivity is configurable by the healthcare organisation, the configurable period of inactivity SHALL be no longer than 2 hours.
Priority:	Mandatory
Rationale:	To prevent unauthorised access to a system after a user has already been authenticated.
Notes:	This requirement is intended for Clinical Information Systems to provide automatic session timeout functionality within the software. Software provider is unaware of the target implementation environment. This function needs to be provided within the software, in addition to any inactivity timeout at the operating system level. Healthcare provider organisations may define the timeout period if the software allows it to be configured. Software-as-a-Service/hosted service providers that are unable to set a unique time period for each organisation may select a default time period no longer than 15 minutes for all organisations and users.

SEC-0086	**Storage of credentials - user** If the healthcare software system stores user credentials in any form, it SHALL ensure that the credentials are stored securely. To securely store credentials, passwords, or passphrases SHALL NOT be stored as plain text and SHALL be stored: • using an irreversible encryption algorithm OR • with salt added and encrypted using an ASD approved hashing algorithm.
Priority:	Conditional
Rationale:	To prevent malicious actors from viewing any stored passwords and gaining unauthorised access to systems using these credentials.
Notes:	It is recommended the salt is unique and randomly generated for each user, with a minimum of 32 bytes. It is understood and acceptable that the algorithm used to generate the salt may result in generating the same salt for different users. Information about ASD approved cryptographic and hashing algorithms can be found in the Australian Cyber Security Centre's Guidelines for Cryptography [ACSC2023b].

SEC-0271	**Multi-factor authentication** The healthcare software system SHALL provide multi-factor authentication capability. The multi-factor authentication function SHALL be configurable by the healthcare provider organisation and SHALL allow the organisation to define the appropriate authentication frequency period.
Priority:	Mandatory
Rationale:	To prevent malicious actors from gaining access and stealing legitimate credentials to the system due to widely used attacks on single-factor username and password. Software provider is unaware of the target implementation environment and hence this function needs to be configurable by the healthcare provider organisation.
Notes:	This requirement does not mandate the compliance of the multi-factor authentication by healthcare provider organisation. Healthcare provider organisation may choose to disable this feature, for example, if the system is integrated with enterprise Single Sign-On (SSO) service. If this feature is enabled, the healthcare provider organisation needs to define the appropriate frequency of authentication based on the organisation's risk profile. It is important that multi-factor authentication is used for privileged accounts. Privileged accounts are more likely to be targeted by threat actors as they could provide full access to systems.

SEC-0083	**Breached credential validation** If the healthcare software system stores user credentials in any form, the system SHOULD validate the user's credentials with a known breached credentials service or against an external known breached credential list. The healthcare software system SHOULD perform the validation when: • user credentials are created • user credentials are updated • user credential is used and has not been validated in the last 24 hours.
Priority:	Recommended
Rationale:	To ensure the user credentials are not compromised passwords from previous data breaches and prevent unauthorised access to the system.
Notes:	This requirement does not apply to healthcare software systems that do not store users' credentials. This include systems that are integrated with an enterprise Single-Sign-On (SSO) service, or systems that verify user's credentials against the organisation's authentication directory, such as Active Directory or an enterprise identity management system. Validation of user credentials is required only once every 24 hours to avoid frequent execution of credential checks, that may adversely impact the breached credentials service performance and interrupt clinical workflow, especially if free services (e.g. "Have I Been Pwned") may be slow to respond during high traffic periods. Software providers should note some breached credential lists are updated infrequently (e.g. once or twice a year). Refer Appendix A.1 for implementation guidance.

SEC-0580	**Notification when credential previously exposed in data breaches** If the healthcare software system performs breached credential validation and the credential was found in a past breach, the system SHOULD alert a responsible person and prompt the user to update the credential before the next login. The alert to the responsible person SHOULD NOT disclose the credential that has been breached.
Priority:	Recommended
Rationale:	To prevent interruption to clinical workflow and not force a user to update their password during the provision of healthcare.
Notes:	A responsible person may be a system administrator or direct supervisor. The alert might be an internal email or text message or some obvious indication to the responsible person inside the healthcare software system. An unobtrusive entry in an audit or transaction log or similar data store is not considered a sufficient alert.

SEC-0075	**System access when no or delayed response from breached credentials service** When the healthcare software system accesses a breached credentials service and the system receives no or delayed response, the system SHOULD permit users to use their credentials and allow access to the system.
Priority:	Recommended
Rationale:	To prevent interruption to clinical workflow and not force a user to receive a response from the credential service during the provision of healthcare.
Notes:	The meaning of “delayed response” is determined by the software provider and will depend on the importance and criticality of the healthcare software system and its target environment. Some free services that provide no Service Level Agreements (e.g. “Have I Been Pwned”) may be slow to respond during high traffic periods.

3.2 Security testing

This section includes the requirements for security testing to assist in identifying security vulnerabilities that adversaries might exploit. This enables software providers and security personnel to develop safeguards to protect against those identified vulnerabilities.

SEC-0220	**Penetration testing** If the healthcare software system is accessible for penetration testing by the software provider organisation, then the system SHOULD be penetration tested periodically at an interval not exceeding 12 months since the last test by a suitably accredited third-party security organisation. Identified vulnerabilities SHOULD be addressed and the system re-assessed so that no residual vulnerabilities remain with a rating score higher than 6.9, as determined by the Common Vulnerability Scoring System (CVSS) v3.1 or v4.0.
Priority:	Recommended
Rationale:	To assist in identifying security vulnerabilities that adversaries might exploit and enable software provider organisation to perform any remediation actions that can protect the software against those identified vulnerabilities.
Notes:	This requirement applies to systems that are hosted and managed by the software provider organisations, including jurisdictional health departments. Example of such systems include Software-as-a-Service (SaaS) systems hosted by a third-party service provider. Due to the evolving cyber threat landscape, this requirement is intended to ensure that the software system performs periodic security testing against newly emerging threats, even if the software functionalities remain unchanged. The Agency deems that an accredited security organisation is an organisation that holds CREST membership. Refer to https://www.crest-approved.org/members/ for a current list of CREST members. Please contact the Agency if the software provider intends to use a security organisation which does not hold CREST membership, prior to committing to and commencing a testing engagement. Refer to Common Vulnerability Scoring System Specification Document [FIRST2019] [FIRST2023] for detailed information and documentation with respect to the Common Vulnerability Scoring System (CVSS) v3.1 or v4.0. Refer to Appendix A.4 for implementation guidance.

SEC-0221	**Vulnerability testing** If the healthcare software system is not accessible for penetration testing by software provider organisation (i.e. hosted by healthcare provider organisation), the system SHOULD be vulnerability tested periodically at an interval not exceeding 12 months since the last test by a suitably accredited third-party security organisation. Identified vulnerabilities SHOULD be addressed and the system re-assessed so that no residual vulnerabilities remain with a rating score higher than 6.9, as determined by the Common Vulnerability Scoring System (CVSS) v3.1 or v4.0.
Priority:	Recommended
Rationale:	To assist in identifying security vulnerabilities that adversaries might exploit and enable software provider organisation to perform any remediation actions that can protect the software against those identified vulnerabilities.
Notes:	This requirement applies to components of systems that cannot be directly accessed by software provider organisations. Examples of such systems include software that is installed on a desktop or applications that are internally hosted (on premise) by healthcare provider organisation. Due to the evolving cyber threat landscape, this requirement is intended to ensure that the software system performs periodic security testing against newly emerging threats, even if the software functionalities remain unchanged. The Agency deems that an accredited security organisation is an organisation that holds CREST membership. Refer to https://www.crest-approved.org/members/ for a current list of CREST members. Please contact the Agency if the software provider intends to use a security organisation which does not hold CREST membership, prior to committing to and commencing a testing engagement. Refer to Common Vulnerability Scoring System Specification Document [FIRST2019] [FIRST2023] for detailed information and documentation with respect to the Common Vulnerability Scoring System (CVSS) v3.1 or v4.0. Refer to Appendix A.4 for implementation guidance.

SEC-0385	**OWASP Application Security Verification Standard (ASVS) Level 1** If the healthcare software system is a web-based system with user input, the system SHOULD follow the Open Worldwide Application Security Project (OWASP) Application Security Verification Standard (ASVS) to Application Security Verification Level 1.
Priority:	Recommended
Rationale:	To ensure that software systems utilise secure coding techniques when building secure healthcare applications, to prevent common vulnerabilities and weaknesses from being exploited, and to develop systems utilising OWASP's guidance as this is considered the security best practice standard for web application development.
Notes:	Level 1 is considered the minimum required for all applications by OWASP. Level 1 is aimed at applications with low protection needs and is formulated in a way that its requirements can be checked in a penetration test or vulnerability test. Testing against this requirement should be included in the scope of penetration or vulnerability testing. See requirement SEC-0220 and SEC-0221. Level 1 controls can also be checked either automatically by tools or manually without requiring access to source code. Refer to the OWASP Application Security Verification Standard [OWASP2019].

3.3 System patching

This section includes the requirements for system patching to ensure the ongoing security of applications, drivers, and systems through effective patch management.

SEC-0020	**Prohibit use of Microsoft Office templates with embedded Flash, Silverlight, or Shockwave controls** The healthcare software system SHALL NOT use or be dependent on Microsoft Office templates in any way that include Flash, Silverlight, or Shockwave content.
Priority:	Mandatory
Rationale:	Flash, Silverlight, and Shockwave products are no longer supported by Microsoft. These unsupported products may contain vulnerabilities that could be exploited by attackers and will not receive any further security patches or updates.
Notes:	Microsoft has stopped support for embedded Flash, Silverlight, and Shockwave content. For more details refer to Microsoft's Announcement.

SEC-0280	**Automated deployment mechanism** If the healthcare software system is hosted by software provider organisation, the system SHOULD provide an automated mechanism that ensures: • security patches and updates are installed, and • operating system patches that are important to the system are installed. If there are any issues with the installation or updates, the system SHALL raise an alert to a responsible person.
Priority:	Recommended
Rationale:	To maintain the integrity of security patches and ensure software systems have fixed newly addressed vulnerabilities.
Notes:	A responsible person may be a system administrator or direct supervisor. The alert might be an internal email or text message or some obvious indication to the responsible person inside the healthcare software system. An unobtrusive entry in an audit or transaction log or similar data store is not considered a sufficient alert.

3.4 Access to systems and their resources

This section includes the requirements for managing access to healthcare software system and My Health Record system to ensure only the users and administrators with appropriate rights and privileges have access to the system.

SEC-0062	**Authorised users for My Health Record** The healthcare software system SHOULD allow only authorised users to access the My Health Record system.
Priority:	Recommended
Rationale:	To prevent inappropriate access to private sensitive consumer information in My Health Record. To restrict the capability of a threat actor in the event of account compromise.
Notes:	Only users authorised by the healthcare provider organisation with specific My Health Record system rights may access My Health Record functionality. This includes viewing, authoring, and uploading My Health Record clinical documents. This authorisation can be achieved by providing role-based access or attribute-based access control. Role-based access enables the principle of least privilege to be adhered to, ensuring users are granted only the minimum level of permission required to perform their tasks.

SEC-0060	**Access to the My Health Record** The healthcare software system SHOULD allow access to the My Health Record system to authorised users (SEC-0062) without the need for the user account to have administrator functions on the operating system.
Priority:	Recommended
Rationale:	To prevent non-administrator users gaining inappropriate privileges and inadvertently compromising the system.
Notes:	Administrator accounts are often targeted as their accounts can potentially give full access to a system. By not requiring the clinical software to run in administration mode, user accounts can be appropriately restricted without impacting the use of the clinical software.

SEC-0160	**Access to backup files** If the healthcare software system provides backup functionality, it SHOULD only allow administrators assigned a dedicated backup administrator role to access, change and erase backup files and data.
Priority:	Recommended
Rationale:	To ensure that the data is protected from unauthorised access, modification, or deletion. By limiting access to only authorised personnel, the risk of data breaches and data loss is reduced.
Notes:	The healthcare provider organisation will determine who the "backup administrator" users are, and the software will ensure only those users can access, change, and erase backup files and data.

SEC-0070	**Privileged account access** The healthcare software system SHOULD restrict software functions associated with advanced or power-users to a role appropriate to those functions.
Priority:	Recommended
Rationale:	To ensure that access to privileged accounts is governed by strict authentication protocols, fine-grained access controls, and immediate privileges revocation mechanisms when needed.
Notes:	Privileged users are those who can alter or circumvent a system's security measures, access and modify system configurations, account privileges, and audit logs, and access important data repositories. This can also apply to users such as software developers, who may have only limited privileges, but can still bypass security measures. The healthcare provider organisation will determine who are the privileged users and which restricted functions are applicable to the privileged user. The software will ensure only those privileged users have access to restricted functions.

SEC-0087	**Disable inactive user accounts** The healthcare software system SHOULD automatically disable a user account after 45 days of inactivity.
Priority:	Recommended
Rationale:	Removing or suspending access for stale accounts can prevent unauthorised access by users who no longer have authority to access the system. Such as when users change duties or leave an organisation.
Notes:	The software provider organisation may set a default period of inactivity, or the healthcare provider organisation may choose to define a period of account inactivity that meets their needs, after which the user account may be considered inactive and vulnerable to misuse in their specific setting, but it should be no longer than 45 days. Software-as-a-Service/hosted service providers that are unable to set a unique period for each organisation may select a period no longer than 45 days for all organisations and users.

3.5 Encryption

This section includes requirements for ensuring data is encrypted both at rest and in transit.

SEC-0110	**Approved cryptographic algorithms and protocols for transmission of information** The healthcare software system SHOULD encrypt all information transmitted using only the Australian Signals Directorate (ASD) approved cryptographic algorithms and ASD-approved cryptographic protocols, except for the data used to support technical operation of the system.
Priority:	Recommended
Rationale:	To prevent unauthorised access to data, data breaches, tampering, and eavesdropping.
Notes:	Support for protocols SSL 1.0/SSL 2.0/SSL 3.0/TLS 1.0/TLS 1.1 must be disabled as they have been deprecated due to exploitable weaknesses. Information about ASD approved cryptographic and hashing algorithms can be found in the Australian Cyber Security Centre's Guidelines for Cryptography [ACSC2023b]. This guideline provides a list of AACAs (ASD approved cryptographic algorithms) and AACPs (ASD Approved Cryptographic Protocols) that must be used when transmitting information. This requirement applies to all patient data transmission (not just the My Health Record) both within the local network and across public networks. The data excluded is the data that is used only for the operation of the system and is unrelated to the patient information, for example, internal record identifiers.

SEC-0084	**Disk encryption for systems hosted by software provider organisation** If the healthcare software system is hosted by software provider organisation (e.g. software-as-a-service, web-based system), the system SHOULD store all data on a partition encrypted with an Australian Signals Directorate (ASD) approved cryptographic algorithm.
Priority:	Recommended
Rationale:	To minimise the risk of unauthorised access to data if an actor gains physical access to the storage device. For example, theft, improper disposal or destruction after decommissioning.

Notes:	Full disk encryption provides a greater level of protection than file-based encryption. While file-based encryption may encrypt individual files, there is the possibility that unencrypted copies of files may be left in temporary locations used by an operating system. Information about ASD approved cryptographic and hashing algorithms can be found in the Australian Cyber Security Centre's Guidelines for Cryptography [ACSC2023b]. Refer to Appendix A.3 for implementation guidance.

SEC-0089	**Disk encryption for systems hosted by healthcare provider organisation** If the healthcare software system is hosted by healthcare provider organisation (e.g. a desktop application), the system SHOULD interrogate the operating system to identify whether disk encryption is enabled. For example, once a day or at the start of each session. If the system detects that the disk encryption is not enabled, the system SHALL allow the operation of the software to continue, and SHOULD alert a responsible person.
Priority:	Recommended
Rationale:	To minimise the risk of unauthorised access to data if an actor gains physical access to the storage device. For example, theft, improper disposal, destruction after decommissioning.
Notes:	Full disk encryption provides a greater level of protection than file-based encryption. While file-based encryption may encrypt individual files, there is the possibility that unencrypted copies of files may be left in temporary locations used by an operating system. It is recommended that the system interrogates the operating system at the startup of the system and then periodically within a period of no more than 24 hours to identify whether disk encryption is currently enabled. If the system identifies through this interrogation of the Operating System that disk encryption is not enabled, the system should notify an appropriate person to take necessary steps to ensure that the disk is encrypted within 72 hours. If not encrypted within 72 hours, then the operation must be ceased. Information about ASD approved cryptographic and hashing algorithms can be found in the Australian Cyber Security Centre's Guidelines for Cryptography [ACSC2023b]. Refer to Appendix A.3 for implementation guidance.

SEC-0125	**Database encryption for systems hosted by software provider organisation** If the healthcare software system is hosted by software provider organisation (e.g. software-as-a-service, web-based system), the system SHOULD store all data in a database (or files) that is encrypted with an Australian Signals Directorate (ASD) approved cryptographic algorithm.
Priority:	Recommended
Rationale:	To prevent network-based attacks where the system is compromised by an attacker, and they have access to the decrypted partition.
Notes:	Database level encryption is offered by many commercial database platforms. Software developers should leverage that functionality to offer protection against data theft. Information about ASD approved cryptographic and hashing algorithms can be found in the Australian Cyber Security Centre's Guidelines for Cryptography [ACSC2023b]. Refer to Appendix A.3 for implementation guidance.

SEC-0126	**Database encryption for systems hosted by healthcare provider organisation** If the healthcare software system is hosted by a healthcare provider organisation (e.g. a desktop application), the system SHOULD interrogate the database to identify whether database encryption is enabled. For example, once a day or at the start of each session. If the system detects that the database encryption is not enabled, the system SHALL allow the operation of the software to continue, and SHOULD alert a responsible person.
Priority:	Recommended
Rationale:	To prevent network-based attacks where the system is compromised by an attacker, and they have access to the decrypted partition.

Notes:	Database level encryption is offered by many commercial database platforms. Software developers should leverage that functionality to offer protection against data theft. It is recommended that the system interrogates the database at the startup of the system and then periodically within a period of no more than 24 hours to identify whether database encryption is currently enabled. If the system identifies through this interrogation of the database that database encryption is not enabled, the system should notify an appropriate person to take necessary steps to ensure that the database is encrypted within 72 hours. If not encrypted within 72 hours, then operation must be ceased. Information about ASD approved cryptographic and hashing algorithms can be found in the Australian Cyber Security Centre's Guidelines for Cryptography [ACSC2023b].] Refer to Appendix A.3 for implementation guidance.

3.6 Application development

This section includes the requirements for making sure all the return values either success or failure of a system call are handled appropriately, and the error return status do not result in a crash or unexpected system behaviour.

SEC-0090	**Return values of system calls** The healthcare software system SHOULD be able to handle all possible return values for all system calls. The system SHOULD capture the return value for every system call and SHOULD have a deliberate course of action.
Priority:	Recommended
Rationale:	To prevent system compromise by attackers via injection attacks or buffer overflow attacks to the system. To prevent exploitation by attackers to execute malicious code or gain unauthorised access to the system.
Notes:	System call describes as a way for the system to request the operating system to perform certain tasks, for example, access a file, allocate memory, or interact with hardware devices. This includes operating system calls and external system calls. This describes a deliberate choice by the software developer to consider each possible documented system response rather than just ignoring the return values. This requirement can be checked in a penetration test or vulnerability test.

3.7 Web application development

This section includes requirements for web-based healthcare software systems (e.g. web pages, web apps etc), and are intended to address vulnerabilities specific to the web environment.

SEC-0180	**HTTP security policies** If the healthcare software system is a web-based system, the system SHOULD implement: • Content-Security-Policy • HTTP Strict Transport Security (HSTS) • X-Frame-Options' response headers OR the 'Frame-Ancestors' directive within the Content-Security-Policy.
Priority:	Recommended
Rationale:	To help protect against various HTTP based attacks and ensure that sensitive data is kept secure.

SEC-0190	**HTTPS exclusively** If the healthcare software system is a web-based system with user input, the healthcare software system SHOULD serve all web application content exclusively on HTTPS.
Priority:	Recommended
Rationale:	To ensure that the data transmitted between the client and server is secure and cannot be intercepted by third parties.
Notes:	Information about ASD approved cryptographic and hashing algorithms can be found in the Australian Cyber Security Centre's Guidelines for Cryptography [ACSC2023b].

SEC-0170	**Output encoding** If the healthcare software system is a web-based system with outputs, the healthcare software system SHOULD perform output encoding on all outputs produced.
Priority:	Recommended
Rationale:	To prevent attackers from injecting malicious code into a web page and executing it. This helps protect against various types of attacks such as cross-site scripting and SQL injection.
Notes:	Output encoding is the process of replacing HTML control characters (e.g. <, >, ", &, etc) into their encoded representatives. This is the best mitigation against cross-site scripting attacks.

SEC-0100	**Input validation for web-based applications** If the healthcare software system is a web-based system, the system SHOULD check all inputs (e.g. datatypes and lengths) to ensure incorrect and inappropriate inputs are captured and managed without compromising the healthcare software system.
Priority:	Recommended
Rationale:	To ensure that the data provided by users is trustworthy and secure. Input validation examines and validates user inputs to prevent malicious attacks and keep the application safe from vulnerabilities.
Notes:	This requirement intends to ensure date fields contain dates, integer fields contain integers etc to protect infrastructure from unnecessary traffic and potential malicious activity. Examples of input validation include: • ensuring a telephone field does not contain letters • ensuring data used in a Structured Query Language query is sanitised properly • ensuring Unicode input is handled appropriately. This requirement can be checked in a penetration test or vulnerability test. Refer to V5.1 Input Validation Requirements in the OWASP Application Security Verification Standard [OWASP2019].

SEC-0390	**OWASP Application Security Verification Standard Level 2** If the healthcare software system is a web-based system with user input, the system SHOULD follow the Open Worldwide Application Security Project (OWASP) Application Security Verification Standard to Application Security Verification Level 2.
Priority:	Recommended
Rationale:	To ensure that software systems utilise secure coding techniques when building secure healthcare applications, to prevent common vulnerabilities and weaknesses from being exploited, and to develop systems utilising OWASP's guidance as this is considered security best practice standard for web application development.
Notes:	Level 2 ensures that security controls are in place, effective, and used within the application. Level 2 is typically appropriate for applications that handle significant business-to-business transactions, including those that process healthcare information, implement business-critical or sensitive functions, or process other sensitive assets, or industries where integrity is a critical facet to protect their business [OWASP2019]. Refer to the OWASP Application Security Verification Standard [OWASP2019].

3.8 Application hardening

This section includes requirements to protect against malicious code executing on systems.

SEC-0030	**Office templates with OLE packages** The healthcare software system SHOULD NOT use or be dependent on in any way Microsoft Office Templates with Object Linking and Embedding (OLE) packages.
Priority:	Recommended
Rationale:	To ensure that threats that exploit the vulnerabilities in Microsoft Office Object Linking and Embedding are prevented.
Notes:	Microsoft Office OLE Packages should be managed carefully by organisations as an adversary can also create macros to perform a variety of malicious activities. By avoiding the use of OLE Packages healthcare provider organisations may place restrictions on the use of Microsoft Office OLE Packages without having an impact on functionality of the clinical software.

SEC-0260	**Trusted macro execution** The healthcare software system SHOULD only permit Microsoft Office Macros that are from trusted locations (refer notes) or restrict all Office Macros.
Priority:	Recommended
Rationale:	To ensure that the threats that exploit the vulnerabilities in Microsoft Office macros are minimised.
Notes:	Refer to the ACSC Microsoft Office Macro Security [ACSC2021a].

SEC-0040	**Office macro signing** If the healthcare software system includes any Microsoft Office macros, the macros SHOULD be digitally signed using a code signing certificate from a commercial third-party Certificate Authority.
Priority:	Recommended
Rationale:	To ensure that threats that exploit the vulnerabilities in Microsoft Office macros are minimised.
Notes:	Microsoft Office Macros should be managed carefully by organisations as an adversary can also create macros to perform a variety of malicious activities. By signing Microsoft Office macros, healthcare provider organisations may place restrictions on the use of Microsoft Office macros while still permitting execution of macros from a third-party software provider organisation.

SEC-0085	**Digital certificate validation** If an external party connects to the healthcare software system and asserts its identity using a digital certificate, the healthcare software system SHOULD validate the digital certificate and its expiry date.
Priority:	Recommended
Rationale:	To establish trust between the healthcare software system and an external connecting party and verify that it is legitimate and not fraudulent. To prevent man-in-the-middle attacks and protect sensitive information from interception and theft by hackers.
Notes:	Certificate validation should be done by: • ensuring the certificate has not been revoked. This may be done by using a Certificate Revocation List (CRL), Online Certificate Status Protocol (OCSP) or other method • verifying that the certificate is from a valid Certificate Authority. Certificate pinning should be considered. Which is where, for specific web addresses a certificate is 'pinned' so that only certificates from a specific Certificate Authority are accepted. Where the network operation to access the CRL or OCSP fails, the certificate validation should not fail as a result. This requirement does not apply to certificates that are issued locally for internal use only. Internal certificates are likely to have other ways to deal with a compromised private key, e.g. reissue new certificates and private keys internally across all impacted systems. Systems connecting to My Health Record are required to use NASH PKI certificates which are X.509. Refer to the NASH Certificates Developer Guide: https://developer.digitalhealth.gov.au/resources/nash-sha-2-certificates-developer-guide Refer to Appendix A.2 for implementation guidance.

SEC-0010	**Prohibit use of Java Applets or Flash** The healthcare software system SHOULD NOT use the following technologies: • Java Applets • Flash.
Priority:	Recommended
Rationale:	To ensure that threats that exploit the vulnerabilities introduced by using the unsupported Java Applets and Flash are minimised. e.g. bad actors running code on a system, to gain access to sensitive information or to install malware
Notes:	This requirement applies specifically to '*Java Applets'* and is not relevant to Java technologies such as J2EE or J2SE. Oracle deprecated Java Applets in Java SE 9 and was removed in Java SE 11. Flash was discontinued in all major web browsers at the end of 2020, meaning that no security patches are available. Refer to https://theblog.adobe.com/adobe-flash-update/.

3.9 Operating system hardening

This section includes the requirements for hardening the security of an operating system to block the unauthorised downloading or running of executable files and preventing malicious actors from executing unauthorised code on the system.

SEC-0290	**Restriction of executables** The healthcare software system SHOULD NOT allow the unauthorised execution of scripts, installers, executables from within the healthcare software system.
Priority:	Recommended
Rationale:	To prevent malicious software from running on a system.
Notes:	Functions that allow uncontrolled access to the hard drive and operating system should be disallowed e.g., providing a Disk Operating System (DOS) prompt within a healthcare software system.

3.10 Data backup and restoration

This section includes the requirements for ensuring appropriate backup, retention, and restoration of all critical information to help ensure business continuity.

SEC-0130	**Minimum storage time** If the healthcare software system provides backup functionality, the system SHOULD NOT automatically delete or overwrite backup files that are within a retention timeframe configured by the healthcare provider organisation.
Priority:	Recommended
Rationale:	This requirement intends to prevent backup files from being automatically removed or deleted from the system.
Notes:	Software providers may choose to configure the time-period to retain the backup files.

SEC-0151	**Backup frequency** If the healthcare software system provides automated backup functionality, the backup frequency SHOULD be configurable by the healthcare provider organisation.
Priority:	Recommended
Rationale:	Frequent backups are essential to ensure that data can be recovered to an appropriate point in time to ensure business continuity.
Notes:	The optimal backup frequency depends on the type of data being backed up, the amount of data, and the criticality of the data. For example, if the data is critical and changes frequently, then it should be backed up more frequently. The backup frequency should align with the business continuity needs. This requirement applies to software systems that have control over the backup functions, for example, the software is hosted by software provider organisation, or integrate with a third-party backup solution that is managed by the software provider.

SEC-0370	**Backup scope** If the healthcare software system provides automated backup and restore functionality, the backup SHOULD contain all important data, software, and configuration settings to enable business continuity when restored.
Priority:	Recommended
Rationale:	To ensure the business continuity of the software system in an event of cyber threat attack.
Notes:	Important information refers to the critical information that is essential for the functioning and operation of the software in an organisation. The specific types of important data are varied, but generally include (but not limited to): customer and healthcare provider data, clinical documents, system configurations and settings.

SEC-0140	**On-screen backup instructions** If the healthcare software system provides the healthcare organisation data backup and restore functionality, the system SHOULD provide on-screen instructions on how to perform the backup and restore functionality.
Priority:	Recommended
Rationale:	To help the healthcare provider organisation ensure their backup/restore functionality is properly configured to enable them to ensure business continuity.
Notes:	This requirement applies to software that is hosted by healthcare provider organisation. The healthcare provider organisation that uses the backup feature within the software, such as desktop application, on premise client/server architectures, and internally hosted application and data that is accessed via a web browser. The end users should be able to find the instructions easily, for example if they can have an indicator or link to find the backup instructions or guidance within the software help files or documentation.

4 Compliance Requirements

This section lists the security compliance requirements for software provider organisations. For a software provider organisation to be considered compliant, it must meet the mandatory requirements and all relevant conditional requirements. Conditional requirements are deemed mandatory if the specific conditions are met for the implementation.

While compliance with recommended requirements is not mandated, it is advisable for software provider organisations to comply with recommended requirements where possible, as these requirements may become mandatory in future releases.

4.1 System patching

This section includes the requirements for system patching.

SEC-0250	**Patch and updating approach** If the healthcare software system is hosted by software provider organisation, the software provider organisation SHALL implement a centralised and managed approach and process that maintains the integrity of patches or updates to ensure that all required system components and patches are in place and current.
Priority:	Mandatory
Rationale:	Applying patches or updates is critical to ensuring the ongoing security of applications, drivers, operating systems, and firmware. In doing so, it is important that patches or updates are applied consistently and in a secure manner.
Notes:	The centralised and managed approach is used to patch or update applications, operating systems, drivers, and firmware, and ensures that they have applied successfully.

SEC-0410	**End of support notifications** The software provider organisation SHALL notify all known customers of a software product and the Agency when the system or version is no longer supported or receiving security updates.
Priority:	Mandatory
Rationale:	To enable the Agency to collaborate with healthcare provider organisations to ensure organisations are aware of potential risks and allow them to take appropriate actions for necessary upgrades or migrations.
Notes:	When applications reach their cessation date for support, an organisation will find it increasingly difficult to protect them against vulnerabilities as patches, updates and other forms of support will no longer be made available by vendors. As such, unsupported applications should be removed or replaced. When the immediate removal or replacement of unsupported applications, compensating controls should be implemented until such time that they can be removed or replaced.

SEC-0460	**Patch and update drivers, firmware and operating system** If the healthcare software system is hosted by the software provider organisation, the software provider organisation SHALL develop and enact a policy where security vulnerabilities in applications, drivers, firmware, and operating system assessed as critical risk are patched, updated, and mitigated. The policy SHALL ensure the patches or updates are applied within 4 weeks of release or within 4 days if it is related to a detected known exploit. The policy SHOULD ensure the patches or updates are applied within 2 weeks of release or within 2 days if it is related to a detected known exploit.
Priority:	Conditional

Rationale: Applying patches or updates is critical to ensuring the ongoing security of applications, drivers, operating systems, and firmware. In doing so, it is important that patches or updates are applied consistently and in a secure manner.

Notes: Known exploits are typically associated with specific software, software libraries, applications, drivers, firmware, operating systems, or network configurations or devices. Once a vulnerability becomes known, security researchers and software vendors work to address and patch the vulnerability to prevent further exploitation. Known exploits can pose critical risk to systems and networks, such as unauthorised access, data loss or theft, service disruption, malware distribution, privacy breaches, financial fraud, etc.

CVSS attempts to assign severity scores to vulnerabilities, allowing responders to prioritise responses and resources according to threat. Scores are calculated based on a formula that depends on several metrics, that approximate ease and impact of an exploit. Scores range from 0 to 10, with 10 being the most severe.

CVSS v3.1 and v4.0 Qualitative Severity Rating Scale:

None	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

Refer to:
Common Vulnerability Scoring System v4.0: Specification Document: https://www.first.org/cvss/v4.0/specification-document [FIRST2023]

Common Vulnerability Scoring System v3.1: Specification Document: https://www.first.org/cvss/v3.1/specification-document [FIRST2019]

SEC-0480	**Supported operating systems and ICT equipment** If the healthcare software system is hosted by the software provider organisation, the software provider organisation SHALL replace or update operating systems for servers and ICT equipment when the operating systems are no longer supported (i.e., patches or updates for security vulnerabilities are no longer available).
Priority:	Conditional
Rationale:	When operating systems, network devices and other ICT equipment reach their cessation date for support, an organisation will find it increasingly difficult to protect them against vulnerabilities as patches, updates and other forms of support will no longer be made available by vendors. As such, unsupported operating systems, network devices and other ICT equipment should be removed or replaced.

4.2 Access to systems and their resources

This section includes the requirements for accessing the software systems and their resources.

SEC-0500	**Privileged access policy** The software provider organisation SHOULD develop and implement a policy of ensuring privileged access to systems, applications and data repositories is validated when first requested and revalidated on an annual or more frequent basis.
Priority:	Recommended
Rationale:	Privileged users, and in some cases privileged service accounts, are often targeted by an adversary as they can potentially give full access to systems. As such, ensuring that privileged accounts are validated minimises opportunities for these accounts to be compromised.
Notes:	This requirement is easiest implemented by having administrator accounts automatically expire after 12 calendar months.

4.3 Application development

This section includes the requirements for application development.

SEC-0430	**Access to source code** The software provider organisation SHOULD ensure that proprietary source code cannot be modified by unauthorised persons.
Priority:	Recommended
Rationale:	Protecting the authoritative source for software is critical to preventing malicious code being surreptitiously introduced into software.

SEC-0440	**Threat modelling and secure design** The software provider organisation SHOULD implement threat modelling and other secure design techniques to ensure that threats to software and mitigations to those threats are identified and accounted for.
Priority:	Recommended
Rationale:	To prevent or mitigate the effects of threats to the system, by identifying the threats, and then defining an appropriate set of security controls to prevent or mitigate them. To ensure that secure-by-design principles, secure-by-default principles, and memory-safe programming languages are used as part of application development to prevent security vulnerabilities in the product.
Notes:	Refer to the Open Worldwide Application Security Project (OWASP) Threat Modelling guide: https://owasp.org/www-community/Application_Threat_Modeling.

SEC-0420 **Security vulnerability notification**

If the software provider organisation provides software that uses third party software, then when security vulnerabilities are discovered that fall into any of the following categories:

- Vulnerabilities that have been assessed as 'High' or 'Critical' (CVSSv3 or CVSSv4 score of above 6.9) by the third-party vendor, or software provider, OR
- Vulnerabilities identified in a software system that is publicly accessible (i.e., expose to the public network), OR
- Vulnerabilities identified in a software system that has a proof-of-concept exploit, OR
- Vulnerabilities that are being actively exploited in the software (regardless of severity),

the software provider organisation SHOULD notify the Australian Digital Health Agency and all customers using the software, within 14 calendar days of security vulnerabilities discovered.

Priority: Recommended

Rationale: To support organisations to address the identified vulnerabilities in a timely manner.

Notes: Notification to customer can be provided using the software, public or private portal.

Notification to the Australian Digital Health Agency will be provided using direct email to "cyber-enquiries@digitalhealth.gov.au".

SEC-0520 **Separate production environment from testing and development environments**

The software provider organisation SHOULD operate the production environment separate from testing and development environments.

Priority: Recommended

Rationale: Segregating development, testing and production environments, and associated data, can limit the spread of malicious code and minimises the likelihood of faulty code being introduced into a production environment.

SEC-0540	**Modifying software in development environment** If the healthcare software system is hosted by the software provider organisation, then they SHOULD only make changes to the software's source code and master data in the development environment prior to testing and migration to production.
Priority:	Recommended
Rationale:	To improve cyber security by reducing the risk of introducing faulty code into a production environment, minimizing the risk of cyber-attacks, and protecting sensitive information from being compromised.
Notes:	System that is hosted by software provider organisation refers to a system in which its server is hosted outside the boundaries and control of the healthcare provider organisation. Examples of such systems include Software-as-a-Service (SaaS), hosted cloud services, and hosted web applications provided by third party service providers.

SEC-0530	**Separate testing environment from development environments** The software provider organisation SHOULD operate development and testing environments as segregated environments.
Priority:	Recommended
Rationale:	Segregating development, testing and production environments, and associated data, can limit the spread of malicious code and minimises the likelihood of faulty code being introduced into a production environment.

SEC-0560	**Independent library testing** The software provider organisation SHOULD ensure that all independent libraries used within their software are tested for security vulnerabilities prior to any release.
Priority:	Recommended
Rationale:	To ascertain that the applications are comprehensively tested for security vulnerabilities, using both static application security testing and dynamic application security testing, prior to their initial release and any subsequent releases

4.4 Data backup and restoration

This section includes the requirements for data backup and restoration.

SEC-0510	**Source code backup** The software provider organisation SHOULD backup their source code regularly.
Priority:	Recommended
Rationale:	Source code repositories are essential to ensure software under development can be restored back to a previous stable version.
Notes:	The source code backup frequency and retention timeframe should be in accordance with the organisation's software development life cycle.

SEC-0380	**Full back up testing** If the software provider organisation offers backup services, the backup SHOULD be tested through a full restoration at least once when initially implemented, and then regularly as part of disaster recovery processes.
Priority:	Recommended
Rationale:	To ensure backups can be restored when the need arises, and that any dependencies can be identified and managed beforehand. It is important that the restoration of important data, software, and configuration settings from backups to a common point in time is tested in a coordinated manner as part of disaster recovery exercises.

Appendix A Implementation guidance

A.1 Breached credential service

Services exist that allow for the checking of user credentials and whether they have been exposed within a security breach. These services differ from simple credential checking as they do not just use an algorithm; they use a database of known breached credential information.

These services are highly effective at reducing compromises to systems that only use single factor for authentication such as username/id and password. However, they should be used in conjunction with other controls such as Multi-factor Authentication mechanisms, since breached credential lists are only updated when the credential breaches are discovered by the breached credential service operators.

Several industries perform this check on their customer accounts at registration and credential change as a good control against password spray and other security attacks.

Some useful guidance links include:

- 2019-130: Password spray attacks – detection and mitigation strategies [ACSC2019]
- Creating Strong Passphrases [ACSC2021b]

One way to check your credentials is by using the service '*Have I been Pwned*'.

This service is referred to by ACSC and has an API for cloud use or a method for offline use that requires manual syncing to the resource.

A risk-based approach should be used to determine how often an organisation should update their breached credential list if they choose the offline method.

API: https://haveibeenpwned.com/API/v3.

Password Lists: https://haveibeenpwned.com/Passwords.

A.2 Digital Certificate Validation

Implementation advice for the validation of digital certificates and use of Certificate Authorities (CAs):

- It is recommended that software developers are using CAs and certificates which implements Certificate Transparency (CT), except when NASH certificates are used.
- The National Authentication Service for Health (NASH) is a PKI that was established for healthcare in Australia and is highly recommended as a PKI solution, (refer https://www.servicesaustralia.gov.au/national-authentication-service-for-health).

Useful links:

• RFC5280: Technical detail for certificate validation (https://www.ietf.org/rfc/rfc5280.txt)

• National Institute of Standards and Technology (NIST) provided resources for testing PKI implementations, including certificate validation and path checking (https://csrc.nist.gov/projects/pki-testing)

A.3 Encryption of data at rest

Disk Encryption:

Modern Desktop and Server Operating Systems include functionality to provide encryption of disks at rest (for example, Windows offers BitLocker disk encryption, and Linux offers LUKS "Linux Unified Key Setup" disk encryption). Similarly, popular Cloud platforms such as Azure and AWS also offer disk encryption functionality.

The system should interrogate the Operating System at the startup of the system and then periodically within a period of no more than 24 hours to identify whether disk encryption is currently enabled.

If the system identifies through this interrogation of the Operating System that disk encryption is not currently enabled, or has been disabled, then the system must notify the individual(s) responsible for supporting and maintaining the infrastructure. In the meantime, the system must be allowed to function, and steps should be taken to ensure that the disk must be encrypted within 72 hours. If not encrypted within 72 hours, then the system must cease operation after that.

Software providers should perform their own technical research or liaise with relevant software developers to identify the most appropriate technical solutions to perform interrogation of disk encryption status within Operating Systems in use by their products, and also for assistance in the management of disk encryption keys. It should be noted however, that it is possible to query Windows via the Command Prompt or via PowerShell to identify whether storage is encrypted. Similarly, there are several tools available for Linux that can be used to determine if a mounted partition is encrypted with LUKS.

Information regarding BitLocker disk encryption for Windows can be found at:
https://learn.microsoft.com/en-us/windows/security/operating-system-security/data-protection/bitlocker/

Information regarding Azure (Cloud) disk encryption can be found at:
https://learn.microsoft.com/en-us/azure/virtual-machines/disk-encryption

Information regarding AWS (Cloud) disk encryption can be found at:
https://aws.amazon.com/blogs/compute/must-know-best-practices-for-amazon-ebs-encryption/

Information regarding LUKS disk encryption for Linux can be found via support pages for all major Linux distributions, however a distribution agnostic overview can be found at:
https://en.wikipedia.org/wiki/Linux_Unified_Key_Setup

Database Encryption:

Many commercial databases include functionality to provide encryption of databases at rest (for example products such as Microsoft's SQL Server, IBM's DB2, MySQL and Oracle offer Transparent Data Encryption "TDE"). Similarly, popular Cloud platforms such as Azure and AWS also offer TDE functionality within their databases.

The system should interrogate the database at the startup of the system and then periodically within a period of no more than 24 hours to identify whether database encryption is currently enabled.

If the system identifies through this interrogation of the database that database encryption is not currently enabled, or has been disabled, then the system must notify the individual(s) responsible

for supporting and maintaining the infrastructure. In the meantime, the system must be allowed to function, and steps should be taken to ensure that the database must be encrypted within 72 hours. If not encrypted within 72 hours, then the system must cease operation after that.

Software providers should perform their own technical research or liaise with relevant software developers to identify the most appropriate technical solutions to perform interrogation of database encryption status within databases in use by their products, and also for assistance in the management of database encryption keys. It should be noted however, that it is possible to query the encryption status of Microsoft SQL Server, IBM's DB2, MySQL and Oracle databases via SQL queries, similarly other products should offer similar mechanisms.

Information regarding Microsoft SQL Transparent data encryption can be found at: https://learn.microsoft.com/en-us/sql/relational-databases/security/encryption/transparent-data-encryption?view=sql-server-ver16

Information regarding Oracle Transparent data encryption can be found at: https://www.oracle.com/database/technologies/faq-tde.html

Information regarding MySQL Transparent data encryption can be found at: https://www.mysql.com/products/enterprise/tde.html

Information regarding Azure SQL (Cloud) Transparent data encryption can be found at: https://learn.microsoft.com/en-us/azure/azure-sql/database/transparent-data-encryption-tde-overview?view=azuresql&tabs=azure-portal

Information regarding AWS (Cloud) Transparent data encryption can be found at: https://aws.amazon.com/rds/features/security/

A.4 Penetration and Vulnerability testing

Penetration testing:

Any penetration test findings with a rating score higher than 6.9, as determined by the Common Vulnerability Scoring System (CVSS) v3.1 or v4.0, are to be addressed within 12 months of the submission of reports.

When defining the scope of a penetration test, the following requirements should be included within the scope, as penetration testing will provide validation of a system's conformance with these requirements:

- SEC-0090 --- Return values of system calls
- SEC-0100 --- Input validation for web-based applications
- SEC-0385 --- OWASP Application Security Verification Standard (ASVS) Level 1

Additionally, within the scope, a requirement to have all identified vulnerabilities rated using CVSS v3.1 or v.4.0 should be included.

NOTE: Please contact the Agency if the software provider intends to use a security organisation which does not hold CREST membership, prior to committing to and commencing a testing engagement and for more information on implementing this requirement.

Common Vulnerability Scoring System version 4.0: Specification Document: https://www.first.org/cvss/v4.0/specification-document [FIRST2023]

Common Vulnerability Scoring System v3.1: Specification Document: https://www.first.org/cvss/v3.1/specification-document [FIRST2019]

OWASP Application Security Verification Standard: https://owasp.org/www-project-application-security-verification-standard/

Vulnerability testing:

Any Vulnerability Test findings with a rating score higher than 6.9, as determined by the Common Vulnerability Scoring System (CVSS) v3.1 or v4.0, are to be addressed within 12 months of the submission of reports.

When defining the scope of a vulnerability test, the following requirements should be included within the scope, as vulnerability testing will provide validation of a system's conformance with these requirements:

- SEC-0090 --- Return values of system calls
- SEC-0100 --- Input validation for web-based applications
- SEC-0385 --- OWASP Application Security Verification Standard (ASVS) Level 1

All identified vulnerabilities rated using CVSS v3.1 or v.4.0 should be included.

NOTE: Please contact the Agency if the software provider intends to use a security organisation which does not hold CREST membership, prior to committing to and commencing a testing engagement and for more information on implementing this requirement.

Common Vulnerability Scoring System version 4.0: Specification Document: https://www.first.org/cvss/v4.0/specification-document [FIRST2023]

Common Vulnerability Scoring System v3.1: Specification Document: https://www.first.org/cvss/v3.1/specification-document [FIRST2019]

OWASP Application Security Verification Standard: https://owasp.org/www-project-application-security-verification-standard/

Acronyms

Acronym	Description
AACA	ASD-Approved Cryptographic Algorithm
AACP	ASD-Approved Cryptographic Protocol
ACSC	Australian Cyber Security Centre
API	Application Programming Interface
ASD	Australian Signals Directorate
B2B	Business-to-business
CA	Certificate Authorities
CAS	Conformance Assessment Scheme
CRL	Certificate Revocation List
CT	Certificate Transparency
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
DOS	Disk Operating System
HSTS	HTTP Strict Transport Security
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over Secure Sockets Layer (SSL)
ICT	Information and Communications Technology
ISM	(Australian Government) Information Security Manual
J2EE	Java 2 Platform Enterprise Edition
J2SE	Java 2 Standard Edition
MFA	Multi-factor authentication
NASH	National Authentication Service for Health
NIST	National Institute of Standards and Technology
OCSP	Online Certificate Status Protocol
OLE	Object Linking and Embedding
OWASP	The Open Worldwide Application Security Project
PKI	Public Key Infrastructure
SaaS	Software as a Service

Acronym	Description
SFA	Single-factor authentication
SSL	Secure Sockets Layer
TLS	Transport Layer Security
VPN	Virtual Private Network
XSS	Cross-site Scripting
UC	Use cases developed to provide a structure to the sections within this document.
URL	Uniform Resource Locator

Glossary

Term	Meaning
Alert	An electronic notification of an exception or event with immediate action required. An alert may be displayed on a user interface and/or communicated to a responsible party through other means (e.g., via a pager, email, or mobile phone). An alert will persist until the underlying exception or event is acknowledged and/or addressed, or the operator explicitly cancels the alert.
Breached credentials service	A known breached credentials service is a service which provides either an application programming interface (API) to check if a password has been included in a known data breach or a list of all known passwords included in known data breaches.
Certificate authority	A trusted organisation that stores, signs, and issues digital certificates to validate the identities of entities such as websites and bind them to cryptographic keys.
Cryptographic algorithm	An algorithm used to perform cryptographic functions such as encryption, integrity, authentication, digital signatures, or key establishment.
CVSS	An open framework for communicating the characteristics and severity of software vulnerabilities. It is a free and open industry standard for assessing the severity of computer system security vulnerabilities.
Clinical information system	A system that deals with the collection, storage, retrieval, communication and optimal use of health related data, information, and knowledge. A clinical information system may provide access to information contained in an electronic health record, but it may also provide other functions such as workflow, order entry, and results reporting.
Digital certificate	A digital certificate is a file or electronic password that proves the authenticity of a website, device, or user. It is tied to a cryptographic key pair that uses public and private keys to exchange communications and data securely over the internet. A digital certificate is verified by a trusted certificate authority.
External endpoints	Access to systems over the public internet (i.e., outside the organisation's network), for example, Software-as-a-Service (SaaS) systems hosted by a third-party service provider with a public network, and internal systems that can be accessed from an external endpoint. Accessing an internal network via VPN is not considered an external endpoint.
Hosted by software provider organisation	A system in which its server is hosted outside the boundaries and control of the healthcare organisation. Examples of such systems include Software-as-a-Service (SaaS), hosted cloud services, and hosted web applications provided by third party service providers.
Healthcare software system	Software and the environment that provides healthcare information to either healthcare providers, healthcare consumers or both. This includes one or more clinical information systems, and potentially the software environment it operates in.

Term	Meaning
Hosted service provider	Business that delivers IT functions such as infrastructure, applications, security, monitoring, storage, web development, website hosting and email, over the Internet or other wide area networks.
Internally hosted system	A system in which its server is hosted within the boundaries and control of the healthcare organisation. Examples of such systems include on premise client/server architectures, and internally hosted application and data that is accessed via a web browser.
Known exploit	A vulnerability or attack method that have already been discovered, documented, and made public, either by security researchers, software vendors, or malicious actors. It can also sometimes include publicly available Proof of Concept (POC) code which may allow attackers to exploit a vulnerability or attack on a system with only very basic knowledge.
OWASP	Open Worldwide Application Security Project which provides comprehensive resources for software developers that should be followed when developing web applications.
Passphrase	A sequence of words used for authentication. [ACSC2023a] Passphrases are made up of four or more random words making them longer than a traditional password. This makes them harder to guess but easy to remember. Changing your passwords to a passphrase is a great way to improve your cyber security. For example, “red house sky train”. [ACSC2021b]
Penetration test	A method of evaluating the security of an ICT system by seeking to identify and exploit vulnerabilities to gain access to systems and data. Also called a ‘pen test’, it is a test using real-world targeted cyber intrusion scenarios in an attempt to achieve a specific goal, such as compromising critical systems or information.
Proof-Of-Concept exploit	An attack against a computer or network that is performed only to prove that it can be done. It is not to cause harm but to show how a hacker can either breach a network or take advantage of a vulnerability that exists in the system.
Responsible person	A responsible person may be a system administrator or direct supervisor. The alert might be an internal email or text message or some obvious indication to the responsible person inside the healthcare software system. An unobtrusive entry in an audit or transaction log or similar data store is not considered a sufficient alert.
Privileged account/user	An account or a user who can alter or circumvent a system’s security measures, access and modify system configurations, account privileges, and audit logs, and access important data repositories. This can also apply to users such as software developers, who may have only limited privileges, but can still bypass security measures.
Salt	A unique, randomly generated string that is added to each password as part of the hashing process. As the salt is unique for every user, an attacker has to crack hashes one at a time using the respective salt rather than calculating a hash once and comparing it against every stored hash. This makes cracking large numbers of hashes significantly harder, as the time required grows in direct proportion to the number of hashes.
SHALL	When appearing in a conformance requirement, this verb SHALL indicates a mandatory requirement. Its negative form SHALL NOT indicates a prohibition.

Term	Meaning
SHOULD	When appearing in a conformance requirement, this verb SHOULD indicates a recommendation. Its negative form SHOULD NOT indicates an option that should not be supported.
Single sign-on	Single-Sign-On service is a centralised authentication service that allows users to access multiple applications and systems within an organisation using a single set of login credentials. This includes healthcare software systems that authenticate verify the user's credentials against the organisation's authentication directory, such as Active Directory or an enterprise identity management system.
Software provider organisation	An organisation that creates an implementation using the My Health Record specifications. A provider may be an organisation that develops a software product, or a provider of digital health services.
Software as a Service	Software that is either supplied as a cloud-based service or deployed over the Internet to run locally. Licenses and support for SaaS systems are commonly provided on a subscription basis, but other models are also used.
Vulnerability test	A documentation-based review of a system's design, an in-depth hands-on assessment, or automated scanning with software tools. In each case, the goal is to identify as many security vulnerabilities as possible.

References

ACSC2019 *2019-130: Password spray attacks – detection and mitigation strategies*, Australian Cyber Security Centre, August 2019

ACSC2021a *Microsoft Office Macro Security*, Australian Cyber Security Centre, October 2021

ACSC2021b *Creating Strong Passphrases*, Australian Cyber Security Centre, October 2021

ACSC2023a *Information Security Manual*, Australian Cyber Security Centre, March 2023

ACSC2023b *Guidelines for Cryptography*, Australian Cyber Security Centre, June 2023

AGENCY2024 *My Health Record System Conformance Assessment Scheme,* Australian Digital Health Agency, 2024

FIRST2019 *Common Vulnerability Scoring System Specification Document*, Forum of Incident Response & Security Teams, August 2019. Available at: https://www.first.org/cvss/v3.1/specification-document

FIRST2023 *Common Vulnerability Scoring System Specification Document*, Forum of Incident Response & Security Teams, November 2023. Available at: https://www.first.org/cvss/v4.0/specification-document

OWASP2019 *Application Security Verification Standard 4.0* (Section V5.1 Input Validation Requirements), Open Worldwide Application Security Project, March 2019. Available at: https://github.com/OWASP/ASVS/raw/v4.0.3/4.0/OWASP%20Application%20Security%20Verification%20Standard%204.0.3-en.pdf