Australian Government
Australian Signals Directorate

ASD AUSTRALIAN SIGNALS DIRECTORATE
ACSC Australian Cyber Security Centre

Information security manual

Last updated: March 2025

Guidelines for software development

Software development fundamentals

Introduction to software development

This section applies to software development activities for traditional applications (including user applications and server applications), mobile applications, web applications and artificial intelligence applications. Additional sections of these guidelines should also be consulted depending on the type of software development being undertaken. For example, the web application development section should be consulted for additional controls applicable to web applications.

Development, testing, staging and production environments

Segregating development, testing, staging and production environments, and their associated data, can minimise the likelihood of faulty or malicious code being introduced into a production environment. Furthermore, protecting the authoritative source for software is critical to preventing malicious code being surreptitiously introduced into software.

Control: ISM-0400; Revision: 6; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Development, testing, staging and production environments are segregated.

Control: ISM-1419; Revision: 1; Updated: Sep-18; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Development and modification of software only takes place in development environments.

Control: ISM-1420; Revision: 5; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Data from production environments is not used in non-production environments unless the non-production environment is secured to at least the same level as the production environment.

Control: ISM-1422; Revision: 3; Updated: Sep-18; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Unauthorised access to the authoritative source for software is prevented.

Control: ISM-1816; Revision: 0; Updated: Dec-22; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Unauthorised modification of the authoritative source for software is prevented.

Secure software development

The use of Secure by Design and Secure by Default principles and practices, including secure programming practices and either memory-safe programming languages (such as C#, Go, Java, Ruby, Rust and Swift) or less preferably memory-safe programming practices, along with threat modelling and mitigation of common security risks, is an important part of secure software development. In addition, providing mechanisms to assist in determining the authenticity and integrity of software, while configuring it in a secure manner, can assist with cyber supply chain security activities.

Control: ISM-0401; Revision: 7; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Secure by Design and Secure by Default principles and practices, including secure programming practices and either memory-safe programming languages or less preferably memory-safe programming practices, are used for software development.

Control: ISM-1780; Revision: 1; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
SecDevOps practices are used for software development.

Control: ISM-1238; Revision: 5; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Threat modelling is used in support of software development.

Control: ISM-1922; Revision: 0; Updated: Jun-24; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
The Open Worldwide Application Security Project (OWASP) Mobile Application Security Verification Standard is used in the development of mobile applications.

Control: ISM-1923; Revision: 0; Updated: Jun-24; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
The OWASP Top 10 for Large Language Model Applications are mitigated in the development of large language model applications.

Control: ISM-1924; Revision: 0; Updated: Jun-24; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Large language model applications evaluate the sentence perplexity of user prompts to detect and mitigate adversarial suffixes designed to assist in the generation of sensitive or harmful content.

Control: ISM-1796; Revision: 1; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Files containing executable content are digitally signed by a certificate with a verifiable chain of trust as part of software development.

Control: ISM-1797; Revision: 1; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Installers, patches and updates are digitally signed or provided with cryptographic checksums as part of software development.

Control: ISM-1798; Revision: 1; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Secure configuration guidance is produced as part of software development.

Software bill of materials

A software bill of materials is a list of open source and commercial software components used in software development. This can assist in providing greater cyber supply chain transparency for consumers of software by allowing for easier identification and management of security risks associated with individual software components.

Control: ISM-1730; Revision: 0; Updated: Dec-21; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
A software bill of materials is produced and made available to consumers of software.

Network application programming interfaces

Network application programming interfaces (APIs) can facilitate the exchange of data between computing devices. As such, common security risks associated with their use should be mitigated during their development, especially for network APIs that are accessible over the internet. In particular, this includes mitigating poorly secured network APIs that facilitate unauthorised modification of data or access to data not authorised for release into the public domain. In such cases, ensuring authentication and authorisation of clients is performed when clients call network APIs can assist in mitigating unauthorised modification of, or access to, data. Finally, centrally logging and analysing network API use can assist in detecting malicious behaviour and contributing to investigations following cybersecurity incidents.

Control: ISM-1818; Revision: 2; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Authentication and authorisation of clients is performed when clients call network APIs that facilitate modification of data and are accessible over the internet.

Control: ISM-2013; Revision: 0; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Authentication and authorisation of clients is performed when clients call network APIs that facilitate modification of data but are not accessible over the internet.

Control: ISM-1817; Revision: 2; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Authentication and authorisation of clients is performed when clients call network APIs that facilitate access to data not authorised for release into the public domain and are accessible over the internet.

Control: ISM-2014; Revision: 0; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Authentication and authorisation of clients is performed when clients call network APIs that facilitate access to data not authorised for release into the public domain but are not accessible over the internet.

Control: ISM-1910; Revision: 1; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Network API calls that facilitate modification of data, or access to data not authorised for release into the public domain, and are accessible over the internet, are centrally logged.

Control: ISM-2015; Revision: 0; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Network API calls that facilitate modification of data, or access to data not authorised for release into the public domain, but are not accessible over the internet, are centrally logged.

Software input handling

Many vulnerabilities in software are caused by a lack of secure input handling. As such, it is essential that software does not trust any input, such as website addresses and their parameters, Hypertext Markup Language (HTML) form data, cookie values, or request headers, without performing validation or sanitisation. Examples of validation and sanitisation include ensuring a telephone form field contains only numerals, ensuring data used in a Structured Query Language query is sanitised properly and ensuring Unicode input is handled appropriately.

Control: ISM-1240; Revision: 4; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Validation or sanitisation is performed on all input received over the internet by software.

Control: ISM-2016; Revision: 0; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Validation or sanitisation is performed on all input received over a local network by software.

Web security policy response headers

Web security policy response header measures, such as Content-Security-Policy, Hypertext Transfer Protocol Strict Transport Security (HSTS) and X-Frame-Options, can be applied by web browsers to help protect themselves. This is achieved by web server software specifying security policy in response headers which web browsers then apply.

Control: ISM-1424; Revision: 5; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Content-Security-Policy, HSTS and X-Frame-Options are specified by web server software via security policy in response headers.

Software interaction with databases

Structured Query Language (SQL) injection attacks, facilitated by the use of dynamically generated queries, are a significant threat to the confidentiality, integrity and availability of database contents. Specifically, SQL injection attacks can allow malicious actors to steal database contents, modify database contents, delete an entire database or even in some circumstances gain control of the underlying database server. Furthermore, when database queries from software fail, they may display detailed error information about the structure of databases. This can be used by malicious actors to further tailor their SQL injection attacks.

Finally, centrally logging and analysing all queries to databases from software that are initiated by users can assist in monitoring the security posture of databases, detecting malicious behaviour and contributing to investigations following cybersecurity incidents.

Control: ISM-1275; Revision: 2; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
All queries to databases from software are filtered for legitimate content and correct syntax.

Control: ISM-1276; Revision: 5; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Parameterised queries or stored procedures, instead of dynamically generated queries, are used by software for database interactions.

Control: ISM-1278; Revision: 5; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Software is designed or configured to provide as little error information as possible about the structure of databases.

Control: ISM-1536; Revision: 3; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
All queries to databases from software that are initiated by users, and any resulting crash or error messages, are centrally logged.

Software security testing

Software security testing can assist software developers in identifying vulnerabilities in software. In doing so, static application security testing and dynamic application security testing should be performed in order to achieve comprehensive test coverage. Furthermore, software developers may choose to use an additional independent party to assist with removing any potential for bias that might occur when they test their own software.

Control: ISM-0402; Revision: 8; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Software is comprehensively tested for vulnerabilities, using static application security testing and dynamic application security testing, prior to their initial release and any subsequent releases.

Vulnerability disclosure program

Implementing a vulnerability disclosure program, based on responsible disclosure, can assist an organisation to improve the security of their products and services as it provides a way for security researchers and other members of the public to responsibly notify them of vulnerabilities in a coordinated manner. Furthermore, following the verification and resolution of reported vulnerabilities, it can assist an organisation in notifying their customers of vulnerabilities that have been discovered in their products and services, and any patches, updates or vendor mitigations that should be applied.

A vulnerability disclosure program should include processes and procedures for receiving, verifying, resolving and reporting vulnerabilities disclosed by internal and external parties. In support of this, a vulnerability disclosure policy should be made publicly available that covers:

- the purpose of the vulnerability disclosure program

- types of security research that are and are not allowed
- how to report any vulnerabilities
- actions, and associated timeframes, upon notification of vulnerabilities
- expectations regarding the public disclosure of vulnerabilities
- any recognition or reward for finders of vulnerabilities.

Finally, the Australian Signals Directorate (ASD) encourages security researchers and other members of the public to responsibly report vulnerabilities directly to an organisation. However, ASD recognises that this is not always practical, initial attempts at communication may be unsuccessful or the person making the report may not wish to do so directly. In such cases, vulnerabilities can be reported to ASD as an independent coordinator. Note, under ASD's limited use obligation, information voluntarily provided to ASD about vulnerabilities cannot be used for regulatory purposes.

Control: ISM-1616; Revision: 0; Updated: Aug-20; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
A vulnerability disclosure program is implemented to assist with the secure development and maintenance of products and services.

Control: ISM-1755; Revision: 1; Updated: Dec-22; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
A vulnerability disclosure policy is developed, implemented and maintained.

Control: ISM-1756; Revision: 1; Updated: Dec-22; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Vulnerability disclosure processes, and supporting vulnerability disclosure procedures, are developed, implemented and maintained.

Control: ISM-1717; Revision: 3; Updated: Sep-24; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
A 'security.txt' file is hosted for each of an organisation's internet-facing website domains to assist in the responsible disclosure of vulnerabilities in the organisation's products and services.

Reporting and resolving vulnerabilities

Following the identification of vulnerabilities, either via internal software security testing or external security researchers, software developers should ensure that such vulnerabilities are reported and resolved in a timely manner. In doing so, software developers should perform root cause analysis and, to the greatest extent possible, seek to remediate entire vulnerability classes.

If vulnerabilities cannot be resolved by software developers in a timely manner via patches or updates, software developers should provide advice on how, to the greatest extent possible, the likelihood of vulnerabilities being exploited can be reduced, the impact of vulnerabilities being exploited can be reduced or both.

Control: ISM-1908; Revision: 1; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Vulnerabilities identified in software are publicly disclosed (where appropriate to do so) by software developers in a timely manner.

Control: ISM-1754; Revision: 3; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Vulnerabilities identified in software are resolved by software developers in a timely manner.

Control: ISM-1909; Revision: 0; Updated: Dec-23; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
In resolving vulnerabilities, software developers perform root cause analysis and, to the greatest extent possible, seek to remediate entire vulnerability classes.

Software event logging

Centrally logging and analysing software crashes and error messages can assist in monitoring the security posture of software, detecting malicious behaviour and contributing to investigations following cybersecurity incidents.

Control: ISM-1911; Revision: 1; Updated: Mar-25; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Software crashes and error messages are centrally logged.

Further information

Further information on a secure software development framework can be found in National Institute of Standards and Technology Special Publication 800-218, *Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities*.

Further information on Secure by Design and Secure by Default principles and practices can be found in the following publications:

- ASD's *Secure by Design foundations*
- ASD's *IoT Secure by Design guidance for manufacturers*
- United Kingdom's National Cyber Security Centre's *Secure development and deployment guidance*
- United Kingdom's Central Digital and Data Office's *Secure by Design Principles* and *Secure by Design Activities*
- United States' Cybersecurity & Infrastructure Security Agency's *Safe Software Deployment: How Software Manufacturers Can Ensure Reliability for Customers*
- United States' Cybersecurity & Infrastructure Security Agency's *Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Secure by Design Software*.

Further information on secure programming practices is available from the Carnegie Mellon University's Software Engineering Institute.

Further information on the need for memory-safe programming languages can be found the following publications:

- United States' Cybersecurity & Infrastructure Security Agency's *The Case for Memory Safe Roadmaps*
- United States' Cybersecurity & Infrastructure Security Agency's *Exploring Memory Safety in Critical Open Source Projects*
- United States' National Security Agency's *Software Memory Safety*.

Further information on mobile application security can be found in the *OWASP Mobile Application Security Verification Standard version 2.1.0* publication.

Further information on large language model application security risks can be found in the *OWASP Top 10 for Large Language Model Applications version 2025* publication.

Further information on artificial intelligence security risks can be found in ASD's *An introduction to artificial intelligence* and *Engaging with artificial intelligence* publications.

Further information on artificial intelligence security risks can also be found in the following publications:

- MITRE’s *Adversarial Threat Landscape for Artificial-Intelligence Systems*
- National Institute of Standards and Technology AI 100-2 E2023, *Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations*
- United Kingdom’s National Cyber Security Centre and United States’ Cybersecurity & Infrastructure Security Agency’s *Guidelines for secure AI system development*
- United States’ National Security Agency’s *Deploying AI Systems Securely: Best Practices for Deploying Secure and Resilient AI Systems*.

Further information on cyber supply chain transparency, and recommended content for a software bill of materials, can be found in the United States’ National Telecommunications and Information Administration’s *The Minimum Elements For a Software Bill of Materials (SBOM)* publication.

Further information on strong authentication can be found in the authentication hardening section of the *Guidelines for system hardening*.

Further information on implementing a vulnerability disclosure program can be found in the following publications:

- Google’s *Starting a Vulnerability Disclosure Program*
- Carnegie Mellon University’s Software Engineering Institute’s *The CERT Guide to Coordinated Vulnerability Disclosure*
- International Organization for Standardization/International Electrotechnical Commission 29147:2018, *Information technology – Security techniques – Vulnerability disclosure*
- International Organization for Standardization/International Electrotechnical Commission 30111:2019, *Information technology – Security techniques – Vulnerability handling processes*.

Further information on developing a vulnerability disclosure policy is available from the disclose.io project to assist an organisation with their implementation.

Further information on recommended contents for a ‘security.txt’ file is available to assist an organisation with their implementation.

Further information on reporting vulnerabilities to ASD as an independent coordinator, including ASD’s limited use obligation, is available from ASD.

Further information on event logging can be found in the event logging and monitoring section of the *Guidelines for system monitoring*.

Web application development

Introduction to web application development

This section describes the controls applicable to web application development and extends upon the prior software development fundamentals section.

Secure web application design and development

OWASP provides comprehensive resources for software developers that should be followed when developing web applications.

Control: ISM-0971; Revision: 8; Updated: Mar-23; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
The OWASP Application Security Verification Standard is used in the development of web applications.

Control: ISM-1849; Revision: 0; Updated: Mar-23; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
The OWASP Top 10 Proactive Controls are used in the development of web applications.

Control: ISM-1850; Revision: 0; Updated: Mar-23; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
The OWASP Top 10 are mitigated in the development of web applications.

Web application frameworks

Web application frameworks can be leveraged by software developers to enhance the security of web applications while decreasing development time. These resources can assist in securely implementing complex software functions, such as session management, input handling and cryptographic operations.

Control: ISM-1239; Revision: 4; Updated: Mar-22; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Robust web application frameworks are used in the development of web applications.

Web application interactions

Hypertext Transfer Protocol Secure (HTTPS) is the Hypertext Transfer Protocol secured by Transport Layer Security (TLS) encryption. The use of HTTPS for web applications can assist in ensuring that interactions with web applications are confidential and that the integrity of such interactions are also maintained.

Control: ISM-1552; Revision: 0; Updated: Oct-19; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
All web application content is offered exclusively using HTTPS.

Web application programming interfaces

Web APIs can facilitate the exchange of data between computing devices. As such, common security risks associated with their use should be mitigated during their development.

Control: ISM-1851; Revision: 0; Updated: Mar-23; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
The OWASP API Security Top 10 are mitigated in the development of web APIs.

Web application output encoding

The likelihood of cross-site scripting and other content injection attacks can be reduced through the use of output encoding. In particular, output encoding is useful when external data sources, which may not be subject to the same level of input filtering, are output to users. The most common example of output encoding is the conversion of potentially dangerous HTML characters into their encoded equivalents, such as ‘<’, ‘>’ and ‘&’ into ‘&lt;’, ‘&gt;’ and ‘&amp;’.

Control: ISM-1241; Revision: 4; Updated: Mar-22; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
Output encoding is performed on all output produced by web applications.

Web application firewalls

When using a web application firewall (WAF), care should be taken with their configuration to ensure that the Internet Protocol (IP) addresses of an organisation's web servers (referred to as origin servers) are not identifiable by malicious actors, as knowledge of origin server IP addresses could allow for protections provided by a WAF to be bypassed. Additionally, appropriate controls should be applied to only allow communication between origin servers, the WAF and authorised management networks.

Control: ISM-1862; Revision: 0; Updated: Jun-23; Applicability: NC, OS, P, S, TS; Essential Eight: N/A
If using a WAF, disclosing the IP addresses of web servers under an organisation's control (referred to as origin servers) is avoided and access to the origin servers is restricted to the WAF and authorised management networks.

Further information

Further information on web application security can be found in the *OWASP Application Security Verification Standard 4.0.3* and *OWASP Top 10 Proactive Controls 2024* publications.

Further information on web application security risks can be found in the *OWASP Top 10 2021* publication.

Further information on implementing HTTPS can be found in ASD's *Implementing certificates, TLS, HTTPS and opportunistic TLS* publication.

Further information on using TLS in HTTPS can be found in the Transport Layer Security section of the *Guidelines for cryptography*.

Further information on web API security can be found in the *OWASP API Security Top 10 2023* publication.

Disclaimer

The material in this guide is of a general nature and should not be regarded as legal advice or relied on for assistance in any particular circumstance or emergency situation. In any important matter, you should seek appropriate independent professional advice in relation to your own circumstances.

The Commonwealth accepts no responsibility or liability for any damage, loss or expense incurred as a result of the reliance on information contained in this guide.

Copyright

Use of the Coat of Arms

For more information, or to report a cybersecurity incident, contact us:

cyber.gov.au | 1300 CYBER1 (1300 292 371)

AUSTRALIA

Australian Government

Australian Signals Directorate