Web Applications Minimum Security Requirements Policy

Prepared By:

National Data Management Authority
March 2023

Document Status Sheet

	Signature	Date
Policy Coordinator (Cybersecurity)	**Muriana McPherson**	**31-03-2023**
General Manager (NDMA)	**Christopher Deen**	**31-03-2023**

Document History and Version Control

Date	Version	Description	Authorised By	Approved By
31-03-2023	**1.0**		**General Manager, NDMA**	**National ICT Advisor**

Summary

1. This policy addresses the acceptable use of information technology resources.
2. This is a living document which will be updated annually or as required.
3. Submit all inquiries and requests for future enhancements to the Policy Coordinator, NDMA.

1.0. Purpose and Benefits

The purpose of the policy is twofold:

a) To outline the minimum requirements for assuring the security of web applications that fall within the purview of all government ministries and agencies in Guyana.
b) To outline a secure design with minimum security requirements in each phase to the Software Development Life Cycle (SDLC).

2.0. Authority

The Permanent Secretary, Administrative Head, Head of Human Resources or their designated representative of the Public Sector Organisation is responsible for the implementation of this policy. For further information regarding the foregoing, please contact the Policy Coordinator - National Data Management Authority (NDMA).

3.0. Scope

These requirements apply to all web applications that fall within the purview of all government ministries and agencies in Guyana, whether acquired off-the-shelf, developed or customized.

4.0. Information Statement

The goal of the Web Applications Minimum Security Requirements Policy is to ensure that the minimum-security requirements are sufficient to assure the confidentiality, integrity and availability of all web applications. It is to also ensure that security requirements are considered in the planning, analysis design, development, testing, and maintenance stages of web applications.

This policy is important to help the Government of Guyana to implement Security Requirements.

5.0. Policy

5.1 Minimum Security Requirements for Web Applications

5.1.1 All applications must be free from vulnerabilities including those referenced as critical in the OWASP Top 10 https://www.owasp.org/images/b/b0/OWASP_Top_10_2017_RC2_Final.pdf.

5.1.2 Applications must be developed in accordance with OWASP Secure Coding Practices https://www.owasp.org/images/0/08/OWASP_SCP_Quick_Reference_Guide_v2.pdf.

5.1.3 All components (web server(s), database(s) and other back-end servers (s), web content management framework, etc.) must be configured according to the relevant vendor/distributor security recommendations.

5.1.4 All connections from web application front-end to back-end systems must be configured to use minimal privileges.

5.1.5 All user-provided input must be validated before it is passed on to back-end systems or returned to the user.

5.1.6 Websites that allow the uploading of files (images, documents, etc.) must verify the file type, validate file size, and be scanned for malicious code.

5.1.7 Web applications must not display error or system messages that reveal information about the underlying configuration.

5.1.8 Components (HTTP verbs, widgets, plugins, add-ons, etc.) that are not necessary for the functioning of the web application must be disabled or uninstalled.

5.1.9 Relevant activity on the server and in the application must be monitored and tracked by appropriate logging mechanisms for auditing and accountability purposes.

5.1.10 Require authentication for all pages and resources, except those specifically intended to be public, all authentication controls must be enforced on a trusted system (e.g., The server).

5.1.11 All sites must use the "secure hypertext transfer protocol" (HTTPS) to ensure that user credentials and other potentially confidential content cannot be intercepted during transmission.

5.1.12 Passwords must not be stored in "clear text", but in a manner that protects them even in case of a compromise to the application.

5.1.13 Users shall be able to change their passwords without the intervention of another person.

5.1.14 Controls that prevent brute-force attacks against user accounts must be implemented, e.g., by" locking out" accounts after a pre-defined number of invalid login attempts, or by displaying a CAPTCHA test (or alternative mechanisms) to prevent automated login attempts. Recommended number of attempts is dependent on the classification of the application, however, for default cases, the application should block users for sixty (60) minutes after five (5) failed login attempts and permanently block users after ten (10) failed login attempts.

5.2 Secure Software Development Life Cycle (SDLC) Recommendations

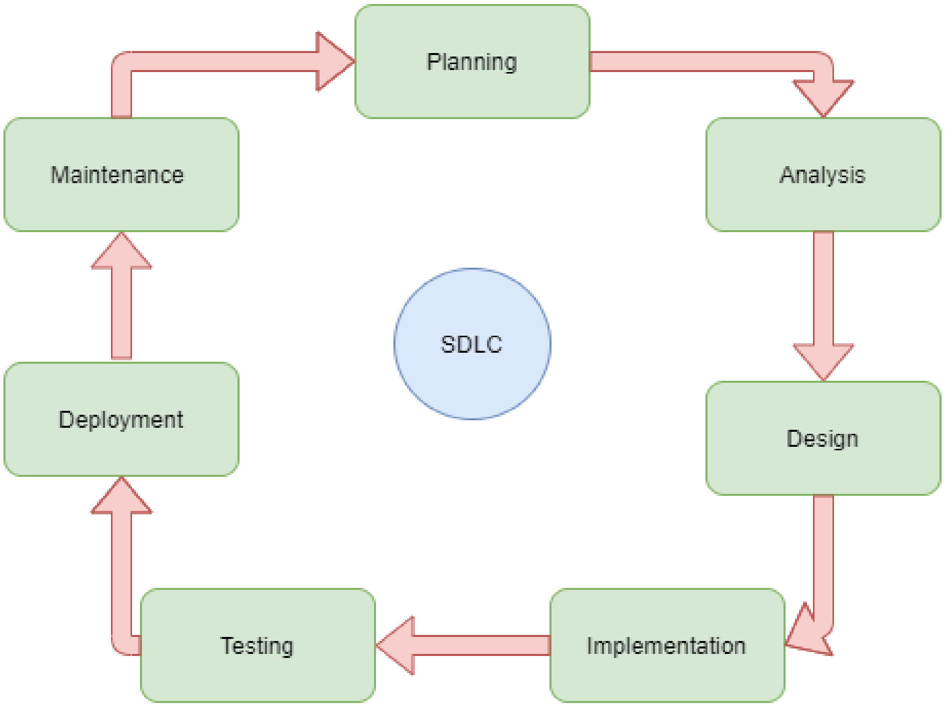
5.3 Security within the SDLC is a critical issue, confidential data can be compromised if an application is exploited due to the lack of security characteristics. Traditionally security is introduced in the final phase of the SDLC and therefore, security implementation (design/code changes) incurs additional cost and time to implement which affects project profitability. If security is implemented effectively throughout the SDLC, it provides an effective mechanism of delivering an application software with information security standards. Security integrated within the SDLC ensures that the software guarantees a high level of confidentiality, integrity, and availability.

SDLC

The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software. There are many different models of the SDLC, however, for this document, the following phases in the SDLC are identified:

1. Planning
2. Analysis
3. Design
4. Implementation
5. Testing

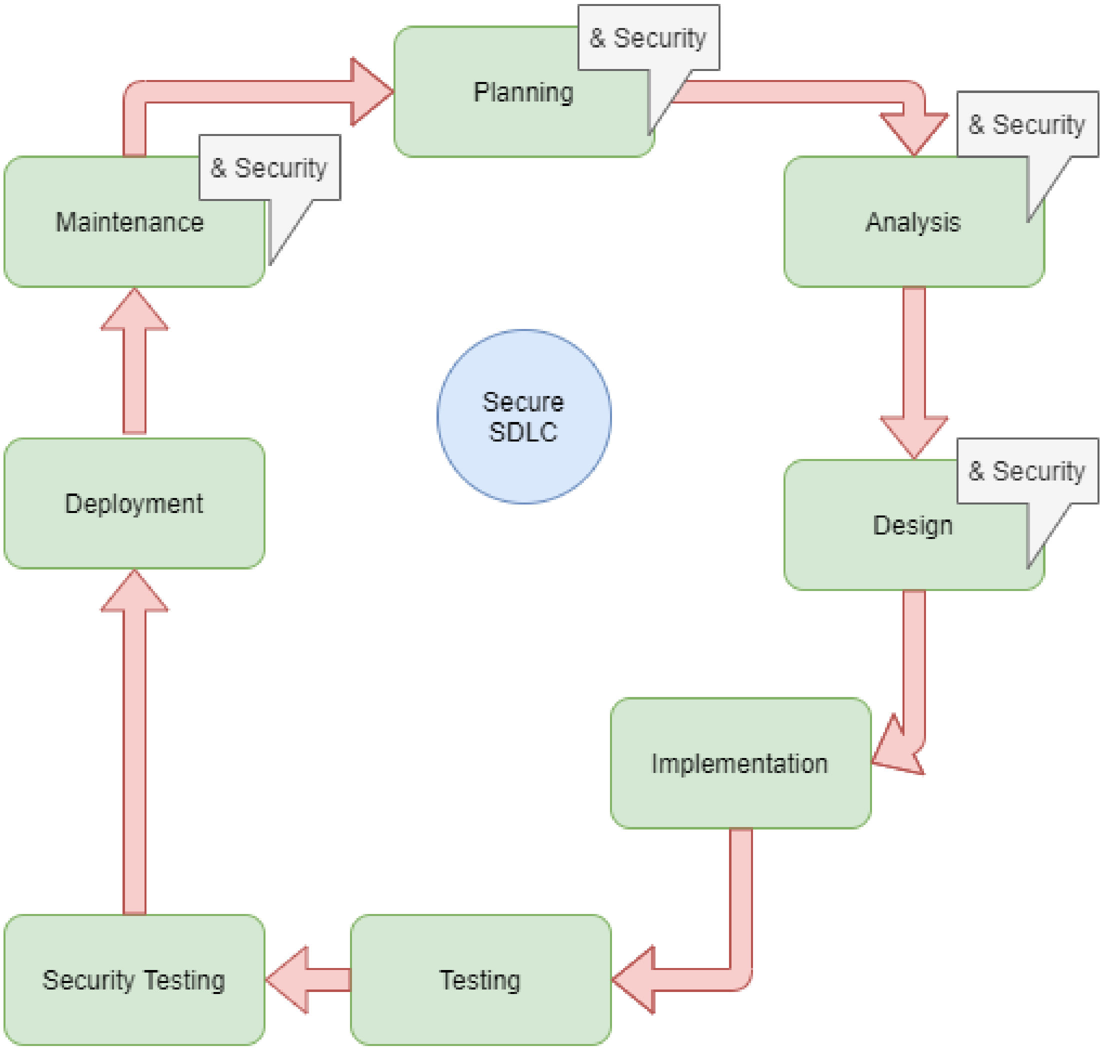
6. Deployment
7. Maintenance



Secure SDLC

A secure SDLC involves integrating security into an existing development process. SDLC generally focuses on functional requirements throughout the initial phases of the SDLC and security is usually introduced as an afterthought and in most cases too late within the software Lifecycle. A secure SDLC methodology includes the following phases (in brackets are possible security activities in secure SDLC):

1. Planning and Security (Involving Security Liaison)
2. Analysis and Security (Data Protection considerations)
3. Design and Security (Data flow and Access Control Design)
4. Implementation (Secure Coding Practices)
5. Testing (Functional Testing)
6. Security Testing (Testing Security Controls)
7. Deployment
8. Maintenance and Security (Periodic Risk Assessment)



5.4 Minimum-Security Requirements that should be incorporated within each phase of the Secure Software Development Life Cycle (SDLC)

5.4.1 **Planning (Security)**

5.4.1.1 Nominate a security liaison with the project.

5.4.1.2 Identify potential data and security considerations. (identify the type of data and privacy concerns)

5.4.2 **Analysis (Security)**

5.4.2.1 Information to be processed by the application must be classified by the development team with the approved Government-wide data/information classification schema.

5.4.2.2 Document vulnerabilities that might threaten the security of the chosen tools (technology, frameworks, languages) to make the appropriate security choices throughout design and development.

5.4.2.3 Document all potential security and privacy issues, these include:

5.4.2.3.1 Applications that store, transmit, and/or process personal information must take into consideration data privacy principles.

5.4.2.3.2 Applications that store, transmit and process Personal Identifiable Information (PII) of Guyanese citizens must comply with all local regulations such as the Data Protection Act.

5.4.2.3.3 Applications that store, transmit and process Personal Identifiable Information (PII) of international citizens must comply with all international regulations such as European General Data Protection Regulation (GDPR).

5.4.2.3.4 Application security controls must be defined, identified, and documented in keeping with industry recognised good practices.

5.4.3 Design (Security)

5.4.3.1 Document architecture security design that demonstrates data flow and access control designs.

5.4.3.1.1 Security controls must restrict access to application functionality to authorised users in accordance with employees' roles and responsibilities.

5.4.3.1.2 Security controls must include control of data input, output, and processing within the application to ensure that data is protected from compromise of confidentiality and integrity.

5.4.3.1.3 Applications must be designed with controls to authenticate users and create audit trails by logging system events and user activities.

5.4.4 Implementation

5.4.4.1 Applications must be developed in accordance with OWASP Secure Coding Practices. https://www.owasp.org/images/0/08/OWASP_SCP_Quick_Reference_Guide_v2.pdf

5.4.4.2 Production environments shall be separate from development and test environments.

5.4.5 Testing

5.4.5.1 This phase is purely functionality testing.

5.4.6 Security Testing

5.4.6.1 A vulnerability assessment should be conducted in this phase to assure all security features are effectively implemented. Vulnerability assessments typically tests for the following minimum-security risks:

5.4.6.1.1 Functional Security Testing (Data Leakage)

5.4.6.1.2 Validation Security Testing (XSS)

5.4.6.1.3 Communication Security Testing (API and HTTP verbs testing)

5.4.6.1.4 Performance/Stress security testing (overflows)

5.4.6.1.5 Environment Specific security testing (Technology specific exploitation)

5.4.6.1.6 Penetration testing

5.4.6.1.7 Monitor testing, this includes checks to assure all security mechanisms such as logging of security incidents are correctly configured.

5.4.6.2 Security testing must not be performed by the originating code authors and should be done by qualified personnel.

5.4.6.3 Security vulnerabilities identified during testing should be addressed prior to implementation. Any untreated security vulnerabilities must be documented, and the documentation reviewed by Cybersecurity and approved by the identified business owner or the head of the agency.

5.4.6.4 Refer to the latest version of the OWASP Web Security Testing Guide for the most current testing practices.

5.4.7 **Deployment**

5.4.7.1 Web application components and supporting services with known or published high risk or critical vulnerabilities must not be used or must be patched within an acceptable timeframe of the vulnerability prior to deployment.

5.4.7.2 All documentation must be adequately protected from unauthorised access.

5.4.7.3 All unnecessary application content should be removed prior to application acceptance into production. This includes removing all test and default files, test user accounts, and other unnecessary content.

5.4.7.4 Web applications must be configured to use a service account assigned the least privileges necessary to run the application.

5.4.7.5 All web application data must have an appointed data custodian who is responsible for maintaining the integrity and protection of the data. This custodian can be the same as the appointed Contracting Agency/Development Team.

5.4.7.6 All hosting agreements must adequately concisely define security requirements and responsibilities to reduce potential misunderstandings.

5.4.7.7 Mechanisms must be established for monitoring hosted applications to ensure agreed service levels are maintained and security controls are operating effectively.

5.4.7.8 Security Incident management responsibilities must be established to ensure that incidents and weaknesses are reported and actioned according to existing agency procedures.

5.4.8 Maintenance and Security

5.4.8.1 Periodic security assessment must be conducted with frequency determined by previous risk assessment and application classification. Security assessments of the application should be conducted to ascertain that:

5.4.8.1.1 All changes to applications, including updates and patches are tested to ensure that there is no adverse impact on operation or security.

5.4.8.1.2 Periodic penetration testing should be performed to ensure the ongoing effectiveness of the application security controls against emerging threats.

5.4.8.1.3 Ensure that all technologies are up to date and void of any version-specific vulnerabilities

5.4.8.2 Version control must be maintained for all application updates and changes.

5.4.8.3 The reference copy of the source code must be stored in a source code library approved by the contracting agency.

5.4.8.4 Source code libraries must be adequately secured to protect against unauthorized or inappropriate access or changes.

5.4.8.5 Old versions of source programs should be archived, with a clear indication of the precise dates and times when they were operational.

5.5 Minimum Security Requirements for Web Developer

5.5.1 Software developers provide a document that lists all the secure coding guidelines they follow.

5.5.2 Developers produce documents that prove they have received security awareness and secure coding training.

5.5.3 A document is provided, that lists the security tools utilized to uncover software vulnerabilities throughout the SDLC and solutions used to protect the development environment from malware.

5.5.4 Software development occurs in a secure, controlled environment. Access should be limited to development personnel only. Developers should not have access to the production and operational environment.

5.5.5 Source integrity checks are performed to ensure that no tampering has occurred with the application code.

5.5.6 In cases where compliance with an Industry security standard is required, for example, Payment Card Industry (PCI), Data Security Standard (DSS), the entity must explicitly state that developers must be familiar with implementing security controls related to that standard.

5.5.7 As part of the contract, software developers sign a non-disclosure agreement (NDA). It must outline in detail what information must remain private and what information can be shared or released to the authorized third party. NDAs are used to protect sensitive information and intellectual property. If the NDA is broken and information is leaked, it is considered a breach of contract. Nefarious individuals can use leaked information to do reconnaissance on an organization. In the case of in-house software developers, the signing of a confidentially statement is recommended.

6.0 Compliance

This policy shall take effect upon publication. Compliance is expected with all organisational policies and standards. Failure to comply with the policy may, at the full discretion of the Permanent Secretary, Administrative Head, or Head of Human Resources of the Public Sector Organisation, may result in the suspension of any or all privileges and further action may be taken by the Ministry of Public Service.

7.0 Exceptions

Requests for exceptions to this policy shall be reviewed by the Permanent Secretary, Administrative Head, Head of Human Resources of the Public Sector Organisation, or the Policy Coordinator, NDMA. Departments requesting exceptions shall provide written requests to the relevant personnel. The request should specifically state the scope of the exception along with justification for granting the exception, the potential impact or risk attendant upon granting the exception, risk mitigation measures to be undertaken by the IT Department, initiatives, actions and a time-frame for achieving the minimum compliance level with the policies set forth herein.

8.0 Maintenance

The Policy Coordinator, NDMA shall be responsible for the maintenance of this policy.

9.0 Definitions of Key Terms

Term	**Definition**
Software Development Lifecycle (SDLC)	A process/methodology for planning, creating, testing, and deploying a web application.
Web Applications	A computer program that utilizes web browsers and web technology to perform tasks over the Internet.
OWASP	The Open Web Application Security Project is an online community that produces freely available articles, methodologies,

Term	Definition
	documentation, tools, and technologies in the field of web application security.
Confidentiality	Requires that data, objects, and resources be protected from unauthorised viewing and other access.
Integrity	Requires that data be protected from unauthorized changes to ensure that it is reliable and correct.
Availability	Requires that authorized users have access to the systems and the resources when needed.
Security Controls	Involves the security practice that blocks or restricts unauthorized applications from executing in ways that put data at risk.

10.0 Contact Information

Submit all inquiries and requests for future enhancements to the Policy Coordinator, NDMA.