NATIONAL DATA
MANAGEMENT AUTHORITY

Secure Website Development and Post Deployment Maintenance Guidelines

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 guide outline concrete recommendations for the secure development, maintenance, and monitoring of Government websites.
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

The purpose of these guidelines is to outline concrete recommendations for the secure development, maintenance, and monitoring of Government websites.

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 guideline. For further information regarding the foregoing, please contact the Policy Coordinator - National Data Management Authority (NDMA).

3.0 Scope

These guidelines encompass all good practices that government of Guyana website developers must implement to ensure secure development, maintenance, and monitoring of Government's online presence.

4.0 Information Statement

Securing websites are of paramount importance in the maintenance of citizen trust as well as for the confidentiality and integrity of information.

The frequency of security compromises faced by Government Ministries and Agencies whose websites are hosted by the NDMA is cause for concern and must be remedied expeditiously. As the authority with responsibility for the provision of secure online presence for Government agencies, we must ensure prudent measures are implemented to curb this issue.

The development and maintenance of websites is of paramount importance to minimize financial losses, protect reputations, and to recover from and continue business operations in the event of a cybersecurity breach. This is usually accomplished by taking a three-pronged approach to security.

These guidelines seek to outline best practices for securing websites within the Government Ministries and Agencies.

5.0 Guideline

5.1 Minimum recommendation for websites development

No	Recommendations	Implications
1.	Application must be free from Injections attacks listed as number one in the OWASP top 10 attacks. A code injection occurs when a malicious attacker sends malicious and invalid data to the application with the intention of making the application do something it was not designed to. Injection attacks include SQL injection and Command injection attacks. All user inputs must be correctly sanitized to protect against injection attacks.	Code injections pose a severe threat to website owners if no safeguards are in place. These attacks take advantage of security flaws to carry out a hostile takeover or leak sensitive information.
2.	Application must be free from Broken Authentication Vulnerabilities which is listed as number two in OWASP top 10. An attacker can use manual and/or automatic methods to try to obtain control of any account they desire in a system. Attackers could also gain entire control of the system if an authentication vulnerability is exploited. This can be prevented by: ➢ Implement multi-factor authentication (if possible) to prevent automated brute-forced attacks. ➢ Change all default credentials, particularly for admin users. ➢ Ensure registration and credential recovery are hardened against account enumeration attacks. ➢ Session IDs should be invalidated after logout, idle, and absolute timeouts.	Broken Authentication mechanism could result in account compromise resulting in unauthorised access to confidential information.
3.	Sensitive data exposure is listed as number three on OWASP 10 vulnerabilities and websites must be resilient to this vulnerability. Sensitive data exposure consists of comprising data that should have been protected. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser. These include implementing HTTPS and not displaying passwords in clear text.	The repercussions of a successful attack would be the disclosure of confidential information.

No	Recommendations	Implications
4	Application must be protected from XML External Entities attacks. Listed as number four on the OWASP top 10 list, an XML External Entity attack is a type of attack against an application that parses XML input. This attack occurs when XML input containing a reference to an external entity is processed by a weakly configured XML parser.	External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.
5.	Access control mechanisms must be effectively implemented. Listed as broken access control in OWASP top 10, restrictions on what authenticated users can do are often not properly enforced. For example, privileged access is required to upload or add new content to a website. However administrative access, which can add users etc or modify website settings is not required to upload website contents, and therefore appropriate access control mechanisms must be enforced. The following recommendations can reduce the risks of broken access control: ➢ Employ least privileged concepts – apply a role appropriate to the task and only for time necessary to complete said task and no more. ➢ Get rid of accounts that are not required. ➢ Audit your servers and websites to determine who is doing what, when, and why. ➢ If possible, apply multi-factor authentication to all your access points.	Attackers can leverage these weaknesses to get unauthorized access to unauthorized functionality and/or data, such as other users' accounts, sensitive files, other users' data, changing access rights etc.

No	Recommendations	Implications
6.	Secure configurations are vital for website security. Security misconfigurations is cited as number six in OWASP top 10. Common security attacks are most times automated and rely on weaknesses found in websites default settings. These misconfigurations can occur in any component of the website such as the database server, webserver, plugins, and CMS's. An example of a misconfiguration is that during the development of an application the developer allows the website to print detailed error messages to the browser so that development hiccups can be easily rectified. However, enabling this option whilst the website is in the production environment would allow an attacker to obtain sensitive information on the website architecture. Which can be used to launch crafted attacks on the website. When developing a website, a production environment is not recommended to develop, test, or push updates without testing.	Allows the website to be vulnerable to common security threats which can compromise the confidentiality, integrity and availability of the website.
7.	Application must be secure from Cross-site scripting (XSS) vulnerabilities. XSS vulnerabilities is listed as number seven in the OWASP top 10 vulnerabilities. XSS attacks consist of injecting malicious client-side scripts into a website and using the website as a propagation method. XSS attacks can be prevented by: ➢ Using frameworks that automatically escape XSS by design, such as React JS, Laravel or CMSs. Learn the limitations of each framework's XSS protection and appropriately handle the use cases which are not covered. ➢ Validating user input. ➢ Encoding output.	The risks behind XSS is that it allows an attacker to inject content into a website and modify how it is displayed, forcing a victim's browser to execute the code provided by the attacker while loading the page.

No	Recommendations	Implications
8.	Avoid using components such as libraries and plugins with known vulnerabilities. These include: ➢ Have an inventory of all website components so that security auditors can quickly identify vulnerable components. ➢ Get rid of components not actively maintained. ➢ Patch all components as directed by vendor guidelines. ➢ Update all components to the latest version.	Such an attack can result in catastrophic data loss or server takeover if a susceptible component is exploited.
9.	Relevant activity on the application must be monitored and tracked by appropriate logging mechanisms for auditing and accountability purposes. These include: ➢ Authentication successes and failures ➢ All CRUD activities ➢ Validation failures ➢ Authorization (access control) failures ➢ Application errors and system events The logging mechanism of the application must capture the following details when recording log events: ➢ When - Log date and time (international format) ➢ Where – Includes all information of the event, which includes identifiers, service names or URLS. ➢ Who – Source address (IP address), user identity (if it is an authenticated user)	Efficient logging mechanisms detect security violations and flaws in application and allows security experts to reconstruct user activities for forensic analysis in a post-breach scenario.
10.	Websites that allow the uploading of files (images, documents, etc.) must verify the file type, validate file size, and be scanned for malicious code.	This assures that the servers are not vulnerable to malicious file upload attacks.
11.	All user-provided input must be validated before it is passed on to back-end systems or returned to the user.	This mitigates the attack surface for common security threats such as SQL injections and XSS attacks.
12.	Components (HTTP verbs, widgets, plugins, add-ons, etc.) that are not necessary for the functioning of the web application must be disabled or uninstalled.	In most cases, particularly in CMS's, developers tend to test multiple extensions during development and are left on production systems. Unused extensions are not tested or maintained effectively and therefore are vulnerable to security threats.
13.	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.	HTTPS enabled secure communication with the webserver and web client. This prevents man-in-the-middle attacks on the web application.

No	Recommendations	Implications
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. 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.	Brute-forced attacks are common in technological landscapes. Protecting against brute-force attacks allows the web application to protect confidentiality and integrity by strengthening the authorized access control mechanisms.

5.2 Recommendations for Post-Deployment maintenance of Government Websites.

The post deployment maintenance phase involves making changes/updates to the website to enhance operational effectiveness. It includes making changes to improve a system's performance, correct problems, enhance security, or address user requirements.

No.	Recommendations	Implications
1.	Whilst the application is in production, the application must undergo periodic security assessments every year. These include monthly external vulnerability scans and bi- annual penetration testing. External scans and penetration tests must occur whenever a change is made to the application N.B Currently the Cybersecurity Division conducts periodic assessments of all websites hosted on the NDMA platform.	Periodic security assessment discovers new and emergent threats that an application may be susceptible to which was not present in the initial assessment. This assessment determines whether the security mechanisms are sufficient against old and newly emerging threats.
2.	All security issues found during vulnerability assessments and Penetration testing must be addressed and mitigated. A tracking mechanism must be implemented to document and track the remediation of all security issues.	This ensures that the application is resilient to common security threats and assures confidentiality, integrity and availability.
3.	Keep/Update a record of initial and post configurations and components implemented on the websites	This is necessary to ensure key stakeholders are aware of the components that may require monitoring and updating, and it is a key record when conducting post-breach analysis.

No.	Recommendations	Implications
4.	Implement a defined process for software updates and patch management. this includes but not limited to the following Assign a resource/entity the responsibility of updating components with vendor-provided software updates and patches released for the various components of the website.	Vendors periodically release software updates and patches to address the vulnerability in their software. Failure to apply same can result in website compromises. Once updates/patches are released, they should be tested and applied to the related component without delay. This resource/entity would therefore keep abreast with vendor and user community announcements.
5.	Limit and Monitor all remote administration sessions to the webserver using TFA with VPN, and audit trails enabled	This is another layer to prevent compromises.

5.3 Recommendations for the monitoring of Government Websites

This phase is considered the regular functioning of the website. The day-to-day activities performed on government websites must be done securely to assure confidentiality, integrity and availability.

No.	Recommendations	Implications
1.	Assign a resource/entity the responsibility of day-to-day monitoring of the website and establish clear procedures for the resource/entity to ensure the website is properly monitored	This will ensure that monitoring is not overlooked and provides a proactive mechanism to prevent compromises. Monitoring procedures may be specific to the CMS and other functionalities provided on the website. 1. Examples of the things to monitor daily include: a. Analyze the raw access logs daily for suspicious activities. b. Check the last login IP for association with a pool of known IP addresses. Block any Suspicious IP addresses if necessary.
2.	Ensure administrative and content provider access are properly managed including the use of provisioning and deprovisioning procedures for the issuance of accounts to administrators and employees/contractors who require login access to interact with some aspect of the website to post content or fix issues.	It is vital to keep current the users who require access to the website and to deprovision users who no longer require access. Deactivate all accounts associated with the website that is not in use and disable access privileges when employees leave.

No.	Recommendations	Implications
3.	Ensure all default passwords are changed. Implement a policy to use complex passwords, change passwords at least quarterly (every three months)	An inactive admin account could be used by an attacker to perpetuate attacks. Further, in the case of a disgruntled former employee, it could be used to compromise systems.
4.	Limit and Monitor all remote administration sessions to the webserver. Ensure TFA with VPN is used during remote administration.	This is another layer to prevent compromises.
5.	Credentials must be changed every time they are shared with external parties. These include instances where credentials are shared with developers to conduct maintenance.	Failure to change credentials after it has been shared with external parties can lead to sabotage if that party decides to go rogue.
6.	Backup the website (including all applications and databases) at least weekly and after major changes have been made. Backups must be encrypted. N.B This is accomplished automatically on the NDMA hosting plan.	This will ensure the ability to restore the website in the event of a major compromise.

6.0 Compliance

These guidelines shall take effect upon publication. Compliance is expected with all organisational guidelines, policies, and standards. Failure to comply with the guidelines 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 guideline 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 timeframe 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 guideline.

9.0 Definitions of Key Terms

Term	Definition
OWASP[1]	Open Web Application Security Project .
XML[2]	A flexible text format designed to describe data for electronic publishing.

10.0 Contact Information

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

[1] Retrieved from NIST Information Technology Laboratory Computer Security Resource Center https://csrc.nist.gov/glossary/term/owasp

[2] Retrieved from NIST Information Technology Laboratory Computer Security Resource Center https://csrc.nist.gov/glossary/term/xml