TECHNICAL GUIDELINES FOR WEB APPLICATION SECURITY

SRI LANKA CERT | CC

An Agency under the Ministry of Technology

Draft - Version 1.0

December 2020

TECHNICAL GUIDELINES FOR WEB APPLICATION SECURITY

This publication is a set of guidelines on secure web application development, hosting and maintenance, in order to ensure the confidentiality, integrity and availability of web applications that is to be followed by government organizations where applicable. It can also be adopted by the private sector to enhance web application security.

The guideline is developed for Senior Officials in charge of the subject of IT (Chief Innovation Officer, Director IT, IT Officer, Network Administrator, Information Security Officer) of the public sector and service providers who develop and host web applications on behalf of government organizations.

[Draft version 1.0]

Revision Date : 11/12/2020

Table of Contents

TECHNICAL GUIDELINES FOR WEB APPLICATION SECURITY

INTRODUCTION

The functional and security requirements for a web application will always vary depending on the type and the purpose of the portal. Web applications are generally more focused on functionality than security. Web application development should focus on usability, functionality and security. All the above three components need to go hand in hand to develop a secure user-friendly Web Application. Failing to focus on security at the right stage or leaving security towards the end of development of the web application have resulted in insecure web applications and frequent web application compromises.

Public web servers can be accessed by any party on the Internet as they are open to public access, thus there is a high probability of the respective servers being compromised. Web Application security should comprise of implementing multitudes of security controls in different layers. As depicted in the Figure 1, securing web applications should focus on following (1) secure web application development, (2) secure hosting, (3) perimeter and network security, (4) database defenses, and (5) securing content.

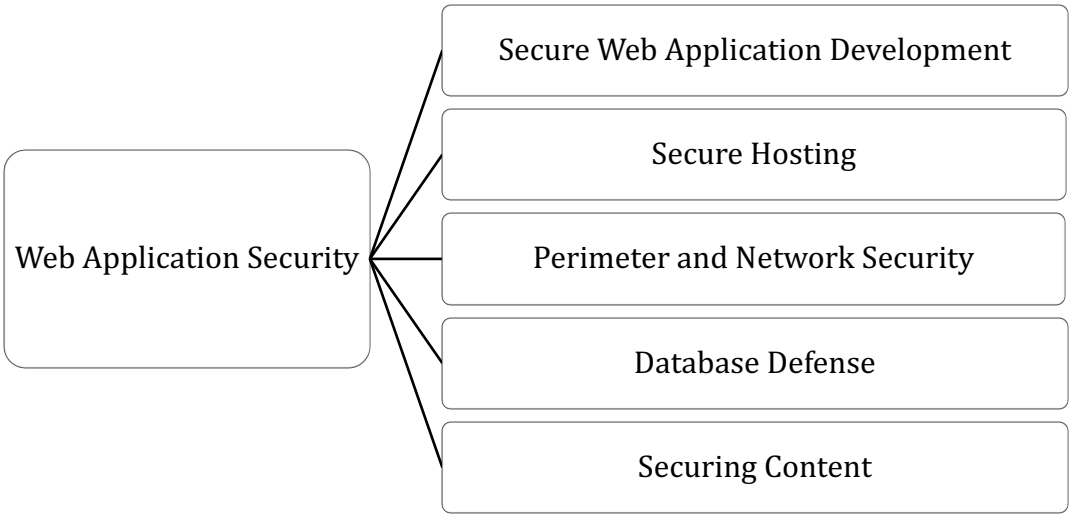

Figure 1 - An overview on Web Application Security

OBJECTIVE

The primary objective of this guideline is to recommend government organizations on the, secure web application development, hosting and maintenance, in order to ensure the confidentiality, integrity and availability of government web applications.

SCOPE

This document describes in detail on the following practices in relation to web application security:

a. Guidelines for secure development of web applications
b. Guidelines for secure hosting of web application
c. Guidelines for securing network and perimeter
d. Guidelines for securing, databases
e. Guidelines for securing of web content

TARGET AUDIENCE

The target audience for this guideline include Chief Innovation Officers [CIO] of Government organizations, IT Officers, Project Managers, Web Application Testers, Information Security Auditors and Engineers and Web Developers.

WEB APPLICATION SECURITY CHECKLIST

This checklist provides an overview of the web application security guidelines that government organizations should comply with.

Action	Activity	Sec No.	Compliance
SECTION 01: SECURE WEB APPLICATION DEVELOPEMNT			
Secure Web Application Development	Initiation Phase 1.1. Identify which security measures need to be incorporated during the initial planning stages of application development	1.1.	
	Design Phase 1.2. Incorporate best practices to the functional and design specification 1.3. Perform risk analysis to identify threats and vulnerabilities	1.2.	
	Development Phase 1.4. Establish best practices to detect and remove security issues	1.3.	
	Testing Phase 1.5. Ensure that application complies with the design security requirements	1.4.	
	Operations & Maintenance Phase 1.6. Respond to post-release security vulnerabilities	1.5.	
SECTION 02 : SECURE HOSTING			
Planning & Managing of Web Servers	2.1. Identify functions of the Web server(s), the types of information that will be stored, processed, transmitted and information is published to the Web server	2.1.	
	2.2. Determining an appropriate Operating System for Web servers	2.1.1. (a)	
	2.3. Determining an appropriate platform for Web servers	2.1.1. (b)	
Securing the Web Server Operating System	2.4. Planning the installation and deployment of the host OS	2.2.1.	
	2.5. Identify, patch and update the host Operating System	2.2.2.	
	2.6. Limit the execution privilege of system tools to system administrators	2.2.4.	
	2.7. Disconnect respective servers from networks or work on an isolated network	2.2.6.	
	2.8. Hardening and configuring the host Operating System.	2.2.3.	
	2.9. Select, install, and configure additional software to provide additional security (Virus Guards)	2.2.5	
	2.10. Install and Harden the updated Web server software on a dedicated host or a dedicated virtualized guest	2.3.1.	
	2.11. Configure OS and Web server access controls to run as a user with a strictly limited set of privileges	2.3.2.	
	2.12. Ensure that log files are stored in a location that is sized appropriately	2.3.3.	

Securing the Web Server	2.13. Configure a secure Web content directory via dedicating a single hard drive or logical partition for Web content	2.3.4.	
	2.14. Disable the execution of scripts that are not exclusively under the administrators	2.3.3.	
	2.15. Define a complete Web content access matrix.	2.3.2.	
	2.16. Configure Uniform Resource Identifiers and Cookies to use the robots.txt file, and anti-spambot protection	2.3.5.	
Administration of Web Servers [Hosting]	Logging 2.17. Establish different log file names for different virtual Web Applications 2.18. Ensure there is sufficient capacity for the logs, review & archive	2.4.1. 2.4.2.	
	Perform Web server backups 2.19. Deployment of a Web server backup policy & Backing up of Web servers	2.4.3.	
	Recover from a compromise 7.5. Report the incident to the organization's computer incident response capability 7.6. Isolate the compromised system(s) to contain the attack and Investigate similar hosts 7.7. Analyze the intrusion, restore the system, test system 2.20. Document lessons learned	2.4.6.	
	7.8. Periodically conduct vulnerability scans penetration testing	2.4.7.	
	Remotely Administration 7.9. Use a strong authentication mechanism 7.10. Restrict which hosts can be used to remotely administer 7.11. Use secure protocols that can provide encryption of both passwords and data 2.21. Enforce the concept of least privilege on remote administration and content updating	2.4.8.	
SECTION 03 : SECURE HOSTING			
Securing Network Infrastructure	3.1. Identify the network location and assess firewall configurations	3.1. 3.4.	
	3.2. Evaluate and configure intrusion detection and prevention systems	3.5.	
	3.3. Evaluate and configure appropriate access control	3.2.	
	3.4. Evaluate and configure routers	3.3	
SECTION 04 : DATABASE DEFENSE			
4. Database Security	4.1. Ensure database servers are adequately protected in terms of hardening, installing patches and is not assigned with a publicly accessible IP	04	
SECTION 05 : SECURING CONTENT			
Securing Web Content	5.1. Ensure that sensitive information is not made available on or through a public Web server.	5.1.	
	5.2. Establish an organizational-wide documented formal policy and process for approving public Web content	5.1.	
	5.3. Maintain a published Web user privacy policy	5.2.	
	5.4. Maintain server-side active content security	5.3.	

01. SECURE WEB APPLICATION DEVELOPMENT

Integrating the security by design is a foundational part of building secure web applications. Regardless of the development method, security of the application is a fundamental aspect. Security requirements must be updated continually when systems functionalities and threat landscape is changed. Ideal time to define the security requirements is during the initial design and planning stages of web applications as this allows development teams to integrate security.

While Legal, industry requirements factors, internal standards and coding best practices, previous incidents, and known threats will influence security requirements to be included. Secure Software Development Lifecycle introduces the security throughout all phases of the web development process as indicated below.

Security by design approach emphasizes the importance of considering the security aspects of the web development life cycle with respect to the (1) Initiation (2) Design, (3) Development, (4) Testing and (5) Operations and Maintenance phases. Refer Figure [2].

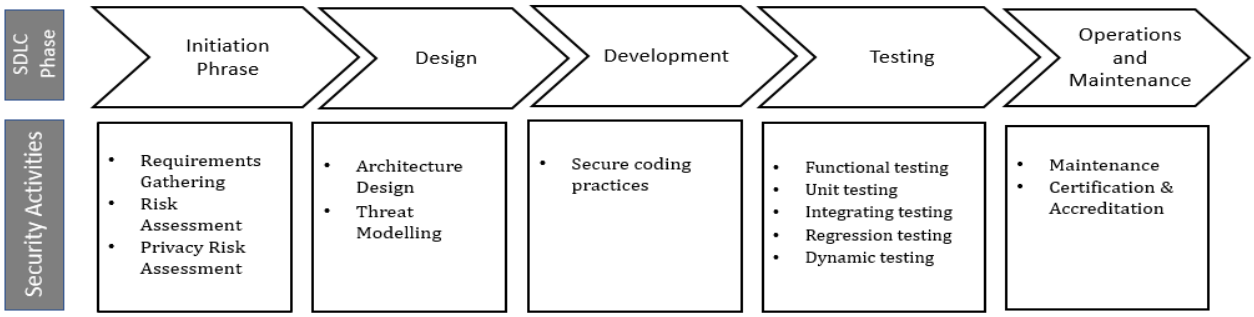

Figure 2 - Overview of Security by design lifecycle - Web Application Development

1.1. Initiation Phase

Software lifecycle beings with requirements gathering. The stakeholders need to understand the scope of the web application to be developed by considering the security, risk and privacy aspects.

1.1.1. Requirements Gathering

During the requirement gathering stage, the functional requirements should be identified and documented on a System Requirements Specification. All functionality of the web application needs to be agreed upon and verified before moving ahead with the initiation process.

The security requirements for the main functions of the web application should be identified with respect to confidentiality, integrity and availability aspects. The level of security requirements for any Web Application will always vary depending on the type and the nature of the Web Application.

The functional requirements shall be revisited each time a change is made to the design specifications. Any major functional changes are likely to have changes with security requirements. Therefore, changes to the functional requirements shall be verified before moving to the next stage of development.

1.1.2. Risk Assessment

The purpose of risk assessment is to identify potential threats on the Web Application at an early stage to minimize or control them to ensure they are maintained at an acceptable level of risk.

During this phase, vulnerabilities and threats alongside the probability and impact of exploitation should be assessed and appropriate controls should be identified to be built onto the web application in order to prevent the impact of the possible exploitation.

Risk assessment results should indicate the platform to develop the Web Application, database platform & model, hosting provider platform, identified files & directories, etc. Security will alone be insufficient therefore privacy also need to be considered

1.1.3. Privacy impact assessment

The Web Application may display, process and store different sensitive level of data. The developer is to store the gathered information to ensure that privacy of the client is not violated. Thus, the type of information collected, where and how it would be stored alongside the credential storing mechanism. The privacy impact assessment is to identify possible impact to privacy and find possible ways to reduce, remove or transfer them.

1.2. Design Phase

The design phase is about transforming the identified requirements to a workable Web Application design.

1.2.1. Architecture Design

The overall structure of the web application (architecture) shall be designed by taking into account the security requirements of the Web Application.

1.2.2. Threat Modeling

A systematic approach is important to understand the different types of threats that would be applicable to the Web Application and how compromise could possibly take place.

All threats related to the design need to be identified and addressed appropriately before moving on to the development phase of the Web Application. For example, applications should be designed to thwart brute force attacks, different type of XSS attack, buffer overflow attack.

1.3. Development Phase

The Development phase focuses on transforming the design to an operable web application. During this phase, the developers will be required to adhere to secure coding practices to avoid application, database and server-side attacks through exploitation. In absence of secure coding methods, the application will be vulnerable for various cyberattacks.

1.3.1. Coding standards and Conventions

The simple rule behind coding standards is to reduce errors and latency by following less complex process. The coding standard shall include collections of rules that determine the programming style, procedure, and methods for each programming language.

1.3.2. Generic Practices

a. The code must be developed in a low complexity to make it efficient.
b. Must be easy to read and understand the code.
c. Use tested and approved code than creating new unmanaged code for common tasks.
d. Maintain naming conventions of the variables throughout the code.
e. Function should be named according to what they would do.
f. For all comments, a specific method must be used.

1.3.3. Input Validations

a. Input validation is performed to ensure only accurate data is entering into the workflow, preventing malformed data from persisting in the database and triggering malfunctions of various downstream components.
b. Input validation should happen as early as possible in the data flow, preferably as soon as the data is received from the external party.
c. Data from all potentially untrusted sources should be subjected to input validation.
d. Input validation should be applied on both Syntactic and Semantic level:
   i. Syntactic validation should enforce correct syntax of structured fields (e.g. NIC, Date, Currency symbol).
   ii. Semantic validation should enforce correctness of their values in the specific business context (e.g., start date is before end date, price is within expected range).
e. Minimum and maximum of characters should be validated.
f. Whitelisting and Blacklisting:
   i. **Whitelisting** – Only accepted characters are allowed in the application [e.g.: - Email field – Alphabets, Numbers and special character (@.) must only be allowed.
   ii. **Blacklisting** – All characters that are disallowed are listed during the phrase of the blacklisting approach, leaving any special characters could lead to security vulnerabilities [e.g. Name field – disallowing special characters (@, #, $, *) & Numbers (0-9)]
g. Client side & Server-side validation:
   i. Client-side validation should be performed to provide a better user experience at a browser level, to prevent heavy network traffic flow on the server side. The client-side validation can be bypassed using proxies.
   ii. Perform both Server side and Client-side input validation (Client-side validation is inadequate for input validation).
h. File upload Validation:
   i. Extension type should be validated.
   ii. Max file size for the upload file should be defined.
   iii. If ZIP files are allowed, do validation check before unzipping the file. The target path, level of compress and unzip size shall be defined.

1.3.4. Authentication

User authentication is the first line of defense and it's very important to ensure the right user has access to the information on the Web Application. The following controls are to be exercised to ensure that a successful authentication strategy is in place

a. All passwords must be sent through a secure connection (TLS 1.3 or latest).
b. The authentication controls must be enforced on a trusted server.
c. Use only HTTP POST requests to transmit authentication credentials.
d. Define minimum length of 14 characters for passwords.
e. Enforce a password complexity policy.
f. Enforce 'Password' field to:
   i. Use of combination of alphanumeric (A-Z, a-z, 0-9) and special characters (@, $, #, &, *)
   ii. Must be at least fourteen (14) characters' long.
   iii. Password should be hashed using an appropriate password hashing algorithm and stored in the database.
   iv. Password entry should be obscured on the user's screen.
   v. Enforce password changes based on requirements established in the policy.
g. Provide a password reset option and use an alternate channel to communicate the method of reset
h. Return a generic message for both existent and non-existent accounts or incorrect passwords.

i. Ensure that generated tokens or codes are randomly generated using a cryptographically safe algorithm.
j. Storing Database Credentials: All Credentials must be stored using hashing algorithms. Ex- PBKDF2, bcrypt or scrypt and avoid using MD5 or a weaker hashing algorithm.
k. All applications or systems shall use appropriate a Multi Factor Authentication mechanism by combing two or more of the following.
    i. Type 1- Something you know. E.g.: - Password, pin
    ii. Type 2 –Something you have. E.g.: - Token based authentication.
    iii. Type 3- Something you are. E.g.: - Biometrics such as fingerprint, voice.
l. All user account must be locked out after a certain number of failed logins attempts.
m. An effective CAPTCHA must be implemented to prevent dictionary and brute-force attacks.

1.3.5. Session Management

It is recommended that the developers implement the following measures.

a. Creating a unique session ID: Use CSPRNG (Cryptographically Secure Pseudorandom Number Generator).
b. The session ID must be meaningless and must not include sensitive information or Personally Identifiable Information (PII). ID must simply be an identifier to the client side. The logic associated with the session ID generally includes client IP address, User-Agent, e-mail, username, user ID, role, privilege level, access rights, language preferences, account ID, current state, last login, session timeouts, and other internal session details. If the session objects include credit card or any sensitive information, it is highly recommended to use SHA256 or higher cryptographic hash functions.
c. Multiple Cookie utilization: There are times when multiple cookies can be used to identify a client. If multiple session cookies are used in the application, all cookies must be verified before allowing access to the application sessions.
d. All sessions must be implemented with idle or inactivity timeout. The business requirements could be taken into consideration while defining the session timeouts.
e. Concurrent sessions and session bypassing should be disallowed.
f. Renewing the sessions are important aspects of session management. After a user has been created the web application should regenerate a new session ID for the user session and renew it on the client. Once the new session is validated, the previous session should be invalidated.
g. User must be completely logged out after clicking on the logout button. The user shouldn't be able to go back using the previous page.
h. Same Site attribute: This attribute provides protection against risk of cross-origin information leakage and CSRF attack. Possible values that could be provided are Strict, Lax, and None. If the attribute is set to none then the secure flag attribute should be set.
    i. Set-Cookie: "SameSite=Strict" The cookie will not be sent along with the request initiated by the third-party websites.
    ii. Set-Cookie: "SameSite=Lax" The cookie is to be sent through a get request initiated by a third-party website.
    iii. Set-Cookie: "SameSite=None" The cookie will be sent in all context therefore secure attribute should be set.

1.3.6. Use of Third-Party Components (TPC)

The following should be considered in the utilization of Third-Party Components (plugins & codes),

a. Choose established and proven TPC to defend from identified threats
b. All TPC used must be listed along with the version
c. Prior to the utilization of TPCs, a risk assessment should be performed
d. Patch or Update the TPCs to the latest stable version

e. Once the TCP is incorporated to the web application, the security impact should be carefully assessed.

1.3.7. Error Handling and Logging

Errors are quite common in Web Application but how errors are detected and handled by the application is very important. Most importantly the unexpected errors in Web Application is a challenge for developers hence its crucial how the Web Application responds to the error.

a. Generic error messages must be created regardless of the user logged into the application. All potential errors and unanticipated errors shall display a generic message. The application error must not leak any sensitive information.
b. Default errors must be customized to generic errors.
c. The developers shall determine the errors that need to be logged. This shall include authentication, session management, admin activities, access to sensitive activities. All necessary information shall be defined and logged properly.
d. Logs related to the application which includes but is not limited to Access logs, transaction logs, security logs shall be stored in a read only medium. Replicating the logs are important and should be followed according to the organization's data retention policy.
e. The retention period needs to be set according to the organization's information security policy. The Log file shall not be destroyed before the required duration of the retention period.
f. All logs shall be reviewed regularly for better security.
g. Only defined individuals shall be allowed to access the log files. Access to the log files should be monitored, recorded and reviewed regularly.
h. Log Transition shall be performed with a secure transmission protocol and the origin should be verified.
i. A separate log server should be maintained.

1.3.8. Data Protection

Developers shall follow and exercise all appropriate measures to protect confidentiality, integrity and availability of the data.

a. Events such as authentication verification data shall always be hashed and stored.
b. Source code shall be obfuscated.
c. Server-side code shall always be protected from end users.
d. Limit the use and storage of sensitive data.
e. Least privilege shall be followed to allow access to applications.
f. All passwords shall be stored using a hash function in a trusted server.
g. An event log should include time, user info, error message and other useful information.

1.3.9. Static Analysis

The static analysis should be performed by the developers on the web application code and server-side code after completing the code development process. The process shall highlight the poor coding practices, programming flaws and vulnerabilities. The code review could be performed manually by going through the code or using automated tools. Following are the instances where the said process needs to be followed.

a. **Planning** – The objectives of the planning stages are to identify the code that needs to be reviewed, a team to perform the code review, the schedule to review the code and process, follow up activities involved need to be clearly identified.
b. **Overview** –The respective code and other related material is to be distributed along with the inspection materials to the code reviewers.

c. **Preparation** –The code and other related material needs to be studied by the code reviewers. The role and the corresponding responsibilities of the reviewing team should be assigned. Also, the recent error types and reviewing techniques should be adopted.
d. **Inspection** - As the name suggests, the errors in the code need to be identified at this stage. The author of the code showing the implementation of the design will make it easier for others to inspect the code. True errors should be identified and noted along with severity level identifications. All findings should be documented in a report.
e. **Rework** – All errors should be remediated, and responses are to be provided. The author should fix code and revert with a response.
f. **Follow-up** - Once rework is done, the moderator follows up with the author to check if the changes have been made as mentioned. Unresolved observations should be documented.

1.4. Testing Phase

Testing the web application during the development phase is an essential process as part of secure web development. Some of the important parts of the testing phase are performing dynamic testing, unit testing, integration testing, functionality testing, regression testing and vulnerability assessments.

1.4.1. Functional Testing

Functionalities of the web application are defined at the beginning of the web development lifecycle. These functionalities need to be tested to verify the requirements of the SRS. A strict process should be followed to ensure all functionalities associated with the web application meets the expected results.

1.4.2. Unit Testing

Unit testing is important to the overall stability of the project, therefore its essential to ensure each unit is tested appropriately before being integrated. The validation of the data structure, logic, and boundary conditions must be part of the unit testing. Performing unit by unit tests will assist the developer in finding bugs efficiently.

1.4.3. Integration Testing:

Each unit tested against the web application requirements and specifications at the unit testing stage are brought together to complete the full web application. This test ensures that the there is no compromise in security.

1.4.4. Regression Testing

Changes are common and expected during the development phase. Testers should ensure regression testing is followed carefully after any changes in the web application to ensure functionality, performance and protection.

1.4.5. Dynamic Testing

The Web application should be tested on an identical staging environment prior to commissioning. All applications must undergo with the certification process with Sri Lanka CERT prior to the launch of the application. When a vulnerability assessment is conducted, the report will include identified vulnerabilities, observations and the mitigation techniques in detail. The report will enable the developers to fix the identified issues.

1.5. Operations and Maintenance Phase

a. Updates and maintenance of a website is important to stay current and secure. The administrator shall ensure the web application platform, Database platform, operating system and webserver platform need to be patched and updated with security patches.

b. Websites are mostly updated with new functionalities for improvements. The changes need to be done without any compromise with security. Each time a new website functionality is added the website shall go through a Vulnerability Assessment.

c. The change control shall be followed by re-certification and re-accreditation process. The Government organization shall ensure that the re-certification process is completed after any functionality changes on the web application. The Reassessment shall be followed by a Vulnerability assessment to ensure that all issues identified are fixed by the developers. The management of the Government organization shall ensure that the re-accreditation is only given after the successful completion of fixes in the reassessment report.

d. The website shall be monitored continuously to ensure the latest security updates are installed. The Government organization will be responsible for securing their web applications. The organizations need to ensure the website is always safe and secure.

e. Any Government web application should follow a Vulnerability Assessment process every year regardless of any changes in the web application. The Government Organization needs to ensure that a Vulnerability assessment is performed by Sri Lanka CERT.

02. SECURE HOSTING

2.1. Planning and Managing Web Servers

The most critical aspect of deploying a secure Web server is careful planning prior to installation, configuration, and deployment. A well designed and detailed deployment plan should be developed by taking into account the following [NIST, 2007a.]

a. Identify the Purpose of the Web Server
   i. Classification of the Information (secure, confidential, limited sharing, public) to be stored, processed and transmitted on the Web server & corresponding services to be provided
   ii. Security requirements for any other hosts involved
   iii. Requirements for continuity of services
   iv. Nature of the network will be used to host web server
b. Identify the network services that will be provided
   i. Categorize the services that will utilize the following protocols: HTTP, HTTPS, Internet Caching Protocol (ICP), Hyper Text Caching Protocol (HTCP), Web Cache Coordination Protocol (WCCP) SOCKS, Database services
c. Identify any network service software, both client and server, to be installed on the Web server and any other support servers
d. Identify the users or types of users of the Web server and determine the privileges that each type of user will have on the Web server
e. Determine how the Web server will be managed (e.g., locally, remotely from the internal network, remotely from external networks)
f. Determine whether appropriate physical security protection mechanisms are in place (e.g. locked rooms, card reader access, security guards)
g. Availability of Redundant power supplies and Internet connections.
h. Adequate precautions in terms of security of the location (Contingency sites)
i. Determine the availability of a redundant web server at the DR Site

2.1.1. Evaluation of appropriate Operating Systems and Platforms for Web Servers

The following measures are to be considered when opting for an appropriate platform and an operating system for Web Servers

a. Determine whether the Platform should be a general purpose, Trusted, Web server appliance, virtualized or pre-hardened operating system.
b. Ensure that the operating system has;
   i. Minimal exposure to vulnerabilities
   ii. Ability to restrict administrative or root level activities to authorized users.
   iii. Ability to control access to data on the server
   iv. Ability to disable unnecessary network services that may be built into the OS or server software
   v. Ability to control access to various forms of executable programs, such as CGI scripts and server plug-ins
   vi. Ability to log appropriate server activities to detect intrusions and attempted intrusions
   vii. Provision of a host-based firewall capability
   viii. Availability of experienced staff to install, configure, secure, and maintain

2.2. Securing the Web Server Operating System

The techniques for hardening different OSs vary greatly; therefore, this section includes the generic procedures common in securing most OSs [NIST, 2007a]. There are 5 main steps to be followed in determining O or S Security;

a. Update default security settings
b. Patching and updating the host OS as required

c. Hardening and configuring the host OS to address security adequately
d. Installing and configuring additional security controls, if needed
e. Testing the host OS to ensure that the previous four steps adequately addressed all security issues.
f. Planning the installation and deployment of the host OS and other components for the Web server

2.2.1. Installation and Deployment of the host OS and Components for the Web Server

In securing of a Host, the general consideration would be

a. Security Certification Level of the chosen platform
b. Level of support provided by the Vendor
c. Compatibility and support concerns of the Software to be used on the platform
d. Support of Security features on the platform (Authentication, Levels of Access control, Remote logging and administration)
e. Minimize the operating system with only essential services by removing all operating system and network services that is not required
f. Keep operating systems and application software up to date with the latest service packs and patches
g. Strong password policies to be enforced
h. Enabling detailed logging including failed logging, etc.
i. Configure Operating Systems with appropriate object, device and file access controls

2.2.2. Patching and Updating the host Operating System

During the patch updating process, following should be considered

a. Create, document, and implement a patching process
b. Keep the servers disconnected from networks or connect them only to an isolated network until patches have been installed to the servers through out-of-band means (e.g. CDs).
c. Identify and install all necessary patches and upgrades to the OS, applications and services whilst mitigating any unpatched vulnerabilities
d. Administrators should not apply patches to web servers without first testing them on another identically configured system.

2.2.3. Hardening and Configuring the Host Operating System

In hardening the Host Operating System, it is essential to Remove or Disable unnecessary Services and Applications.

Some common types of services and applications that should usually be disabled if not required include the following:

a. File and printer sharing services (e.g., Windows Network Basic Input or Output System [NetBIOS] file and printer sharing, Network File System [NFS], File Transfer Protocol [FTP])
b. Wireless networking services
c. Remote control and remote access programs, particularly those that do not strongly encrypt their communications (e.g., Telnet)
d. Directory services (e.g., Lightweight Directory Access Protocol [LDAP], Kerberos, Network Information System [NIS])
e. Email services (e.g., Simple Mail Transfer Protocol [SMTP])
f. Language compilers and libraries, System development tools, System and network management tools and utilities, including Simple Network Management Protocol (SNMP)

2.2.4. Configure Operating System User Authentication

In addition to the Access Control Policy of the government organizations, the following steps should also be taken to ensure the appropriate user authentication.

a. Remove or Disable Unnecessary Default Accounts and Groups
b. Disable Non-Interactive Accounts
c. Create the User Groups—Assign users to the appropriate groups.
d. Create the User Accounts—The deployment plan is to identify who will be authorized to use each computer and its services.
e. Organizations should implement authentication and encryption technologies, such as Secure Sockets Layer (SSL) or Transport Layer Security (TLS), Secure Shell (SSH), or virtual private networking (VPN), to protect passwords during transmission.

2.2.5. Installing and Configuring Additional Security Controls

Anti-malware software, such as antivirus software, anti-spyware software, and rootkit detectors can be installed to protect the local OS from malware and to detect and eradicate any infections that may occur.

2.2.6. Security Testing of the Operating System

Vulnerability scanning entails using an automated vulnerability scanner to scan a host on a network for identifying OS vulnerabilities and Penetration testing is a testing process designed to compromise a network using the tools and methodologies of an attacker. It involves identifying and exploiting the weakest areas of the host or networks.

Vulnerability Assessments and Penetration Testing is recommended for securing operating systems.

2.3. Securing the Web Server

Following actions are recommended for securing the Web Server.

2.3.1. Secure Installation of Web Servers

During the installation of the Web server, the following steps should be performed:

a. Install only the services required for the Web server and to eliminate any known vulnerabilities through patches or upgrades.
b. Any unnecessary applications, services, or scripts that are installed should be removed immediately once the installation process is complete.
c. Install the Web server software either on a dedicated host or on a dedicated guest OS if virtualization is being employed.
d. Apply any patches or upgrades to correct for known vulnerabilities. Create a dedicated physical disk or logical partition for Web content.
e. Remove or disable all services installed by the Web server application but not required, all unneeded default login accounts created by the Web server installation & all manufacturers' documentation alongside all example or test files from the server, including scripts and executable code.

2.3.2. Configuring Access Controls

Web server administrators should consider how best to configure access controls to protect information stored on public Web servers from two perspectives:

i. Limit the access of the Web server application to a subset of computational resources.
ii. Limit the access of users through additional access controls enforced by the Web server, where more detailed levels of access control are required.

The proper setting of access controls can help prevent the disclosure of sensitive or restricted information that is not intended for public dissemination.

Typical files to which access should be controlled are as follows:

a. Application software and configuration files

b. Files related directly to security mechanisms: (Password hash files and other files used in authentication. Files containing authorization information used in controlling access, Cryptographic key material used in confidentiality, integrity, and non-repudiation services)
c. Server log and system audit files
d. System software and configuration files
e. Web content files.

2.3.3. Configuring the Permissions of the Web Server Application

The following should be enforced in terms of the Web server host OS access controls [NIST, 2007a]

a. Service processes are configured to run as a user with a strictly limited set of privileges (i.e., not running as root, administrator, or equivalent).
b. Web content files can be read but not written by service processes.
c. Service processes cannot write to the directories where public Web content is stored.
d. Only processes authorized for Web server administration can write Web content files.
e. The Web server application can write Web server log files, but log files cannot be read by the Web server application.
f. Only root or system or administrative level processes can read Web server log files.
g. Temporary files created by the Web server application, such as those that might be generated in the creation of dynamic Web pages or by users uploading content, are restricted to a specified and appropriately protected subdirectory (if possible).
h. Access to any temporary files created by the Web server application is limited to the Web server processes that created the files (if possible).
i. Installing Web content on a different hard drive or logical partition than the OS and Web server application.

2.3.4. Configuring Secure Web Content Directory

The following steps are required to restrict access to a specific Web content file directory tree:

a. Dedicate a single hard drive or logical partition for Web content and establish related subdirectories exclusively for Web server content files, including graphics but excluding scripts and other programs.
b. Define a single directory exclusively for all external scripts or programs executed as part of Web content (e.g., CGI, Active Server Page [ASP], PHP).
c. Disable the execution of scripts that are not exclusively under the control of administrative accounts. This action is accomplished by creating and controlling access to a separate directory intended to contain authorized scripts.
d. Disable the use of hard or symbolic links.
e. Define a complete Web content access matrix. Identify which folders and files within the Web server document should be restricted and which should be accessible (and by whom).
f. Do not use links, aliases, or shortcuts in the public Web content file directory tree that point to directories or files elsewhere on the server host or the network file system.

2.3.5. Configuration of Uniform Resource Identifiers, Cookies and Web “Bots

Uniform Resource Identifiers (URI) are the address technology from which URLs are created. Publicly served Web content should not include sensitive URIs hidden in the source code.

Collecting cookies should be disabled unless there is a need to gather the data on the site, and only with the appropriate approvals, notifications, and safeguards in place [OWASP, 2002].

Web bots (Crawlers or spiders) are software applications used to collect, analyze, and index Web content. Spambots searching for Web forms to submit spam-related content are a direct threat to the Web Application and affect the availability by making it difficult for users to find necessary content. There are several techniques available to reduce the amount of spam submissions, including the following.

a. Web administrators who wish to limit bots' actions on their Web server need to create a plain text file named "robots.txt." in the Web server's root document directory, as malicious bots ignore this file while scanning.
b. Blocking form submissions that use spam-related keywords
c. Using the **rel= "nofollow"** keyword in all submitted links, which will cause search engines to omit the links in their page-ranking algorithms, directly affecting the goals of a spambot.
d. Requiring submitters to solve a Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) prior to being allowed to submit content.

2.4. Administering the Web Server Applications

Web server administrators need to maintain its security continuously.

2.4.1. Logging

Logging is a foundation of a sound security posture. The following configuration is an essential for logging into public Web servers:

a. Use the combined log format for storing the Log, or manually configure the information described by the combined log format to be the standard format for the Log.
b. Use the remote user identity as specified in RFC 1413 whilst ensuring procedures or mechanisms are in place so that log files do not fill up the hard drive.

2.4.2. Reviewing and Retaining Log Files

a. Reviews should take place regularly (e.g., daily) and when a suspicious activity has been noted or a threat warning has been issued.
b. Log files should be protected to ensure that if an attacker does compromise a Web server, the log files cannot be altered to cover the attack.
c. Depending on the criticality, logging can be performed using a syslog or an and event management (SIEM) software.
d. Log files should be backed up and archived regularly.

2.4.3. Web Server Backup Procedures

The Web server [Data and the O or S] should be backed up periodically for legal and financial reasons as well as to ensure business continuity.

2.4.4. Maintain a Test Web Server

A test server is to be maintained on a Test or Development Web server which should to identical to the production or live web server on the organization's intranet.

2.4.5. Maintain an Authoritative Copy of Organizational Web Content

a. All organizations should maintain an authoritative copy of their public Web Applications on a host that is inaccessible to the Internet.
b. Consider performing automatic updates from the authoritative copy to the Web server periodically as this will overwrite a Web Application defacement automatically [NIST, 2007a].

2.4.6. Recovering from a Security Compromise

Steps to be performed after discovering a successful compromise are as follows:

a. Report the incident to the Director IT or CIO, Isolate the compromised systems or take other steps to contain the attack so that additional information can be collected.
b. Consult expeditiously, as appropriate, with management.
c. Investigate similar hosts to determine if the attacker also has compromised other systems.
d. Analyze the intrusion with the support of experienced and qualified experts.
e. Restore the system.

i. Either install a clean version of the OS, applications, necessary patches, and Web content; or restore the system from backups
ii. Disable unnecessary services and apply all patches.
iii. Change all passwords
iv. Reconfigure network security elements (e.g., firewall, router, IDPS) to provide additional protection and notification.
v. Test system to ensure security & reconnect it to network.

f. Monitor system and network for signs that the attacker is attempting to access the system or network again.
g. Document lessons learned.

2.4.7. Scanning of Web Servers

Periodic security testing of public Web servers is critical. This can be done in terms of Vulnerability Scanning as well as Penetration Testing.

2.4.8. Remotely Administering a Web Server

It is strongly recommended that remote administration and remote updating of content for a Web server is to be allowed only after careful consideration of the risks. The most secure configuration is to disallow any remote administration or content update. Remote Administration should only be performed through secure connections

03. PERIMETER AND NETWORK DEFENSE

3.1. Designed Screened Subnet

a. The network architecture can be designed as a single or multiple layer, as per the requirement of the organization.
b. A Web Hosting Network should have at least following segments.
   i. Internet Segment or Public Server Segment (Web, Mail, DNS Servers)
   ii. Internal Segment
c. The Web Server should be placed in the secure server security segment (DMZ or screened subnet) isolated from the public network and organization's internal network. Web Servers should be placed in the Internet Segment.

3.2. Access Controls

As per the Access Control Policy of the Organization, access to the network resources shall be restricted.

3.3. Routers

The router is the first line of defense to the network of an organization and hence the router itself should be secured. Necessary control should be applied on the router to stop unwanted traffic and attacks at the perimeter. In the secure configuration of a router, the following should be considered.

a. Deploy proper access management and preferably disable remote administration.
b. Enable a secure password
c. Change default SNMP community string
d. ACLs (Access Control Lists) should include Applying egress or ingress filters, filtering all RFC 1918, 3330 Address space & special or reserved addresses and permit the required services for the required IP Addresses only
e. Turn on logging to a central syslog server

3.4. Firewalls

A firewall is a combination of hardware and software, located at a network gateway, protecting the resources of a network from users of other networks. It enforces a boundary between two or more networks and limits access between networks and network segments in accordance with the local security policy. It filters all network packets to determine whether to forward them towards their destination or discard them.

Following should be considered in configuring Firewalls.

a. Update Default Security Settings

i. Default Firewall settings should be updated.
ii. Default vendor supplied user accounts should be disabled after setting their password to a complex value.
iii. The firewall should not have any additional services running that can be accessed remotely.

b. Firewall Interfaces

Listed below are guidelines to adhere to while determining the number of firewall segments and servers or applications to be hosted within that segment;

i. Sensitive and critical web applications or servers that are accessed only internally should be hosted on the most protected segment of the firewall.
ii. Applications accessed internally as well as by external sources should be hosted on a separate segment of the firewall, preferably the Demilitarized Zone (DMZ). Additionally, for better manageability, these systems can further be classified into business application and

infrastructure support applications (e.g., DNS, Web mail, Proxy), with each category hosted in a separate DMZ. Connection links from third parties should terminate on a separate interface of the firewall.

iii. Wide Area Network (WAN) links connecting to the data center should terminate on a separate interface of the firewall.

iv. Administrators usually require unrestricted access to systems and networks they manage. In case these administrators have their machines configured as part of the user LAN, there is the strong possibility that a malicious user may sniff the administrative communication and thereby gain unauthorized administrative access. To avoid this, it is necessary to group all administration terminals on to a separate interface of the firewall with access restricted only to administrators. Additional security measures such as multi factor authentication should be considered for protecting these terminals.

c. Rule Base Creation

The Network Administrator is responsible for designing and testing the firewall rule base before deployment in production. The following guidelines should be adhered to while adding or modifying the rule base,

i. By default, the firewall MUST have a DENY ALL policy, with access granted on a need to do basis. The firewall should have a rule to deny all access that is not explicitly allowed.

ii. Only required services or ports must be opened between specific source and destination IP addresses or subnets. Use of the "ANY" literal either in the source, destination or service or ports must be strictly avoided.

iii. The firewall rule base should restrict access to required ports on the target machine. The source field in the rule base should be restricted to specific IP addresses or subnet addresses wherever feasible. In the case of applications where the number of individual IP addresses or subnets is very large the source address can be made generic to make the rule base more manageable.

iv. Application or servers which are directly accessed from a public network such as the internet should be moved to a separate segment (Demilitarized Zone) of the firewall. The IP address of the server should be NATed with a public IP address.

v. For connections with third parties, NATing should be performed using any available private or public address slots.

vi. Access to administrative ports including SSH and Microsoft Windows Terminal services on protected servers should have user ID based authentication at the firewall in addition to the source IP address.

vii. The firewall user-database, needed for rules that are configured for user authentication, can be stored either locally on the firewall or in an external directory server.

viii. Password policies for these user accounts including password expiry, password history, and password complexity should be enforced. Account lockout should be configured to prevent password cracking attempts. It should be ensured that these user credentials are transmitted in encrypted format from the user PC to the firewall.

ix. For certain applications such as MySQL, Active FTP uses random ports for data transfer between the client and server, after the initial handshake has taken place over a standard port. For such access requirements, the following steps should be followed to avoid exposing all standard TCP or UDP ports,

- Create a service group for the application consisting of the standard port for initial communication and all non-standard TCP or UDP ports (1024 and above)
- Open access for the service group between the desired client and server
- Enable logging on the individual rules judiciously.
- The comments column for each rule MUST have the following information duly entered, Purpose of the rule, Expiry date, for temporary rules.
- The LAST rule for each segment MUST be a "DENY ALL" rule denying all traffic not explicitly allowed. Logging should be enabled on this rule.

d. Rule Base Change

i. After the firewall goes into production, all changes to the rule base should be done after proper authorization, to ensure that the security level is maintained.
ii. Users should contact the application owner for any access requirement. Application owners should validate the request, translate the user request to specific IP addresses and port numbers and pass it to the CIO or Director IT.
iii. A backup or recovery strategy should be in place to ensure that an implementation failure does not adversely impact availability of other systems and firewalls, in general.

e. Administrative Access

iv. Administrative access to firewalls is required for activities including rule base modification, firewall-user account management, firewall-administrator account management and log monitoring.
v. Super-user privileges should be provided to a relevant officer on a need to have and need to do basis.
vi. Default passwords for all vendor-supplied user accounts should be changed to complex combinations.
vii. Logical access to the firewalls should be limited.
viii. Access to firewall administration programs should be through encrypted channels. If the firewall software itself does not provide this facility, then additional mechanisms such as IPSec should be used for this purpose.

f. Audit Logging

i. Logging needs to be enabled to ensure that all critical access is tracked. Logging should be enabled for rules enabling administrative access (e.g., SSH access to web server). Logging should not be enabled for normal user access (e.g., HTTP access to web server).
ii. Logging should be enabled for the last rule that blocks all access that is not explicitly allowed by the other rules.
iii. Logging should be enabled to track any changes done to firewall configurations including changes to the rule base.
iv. Logs should be monitored periodically for the following activities,
    - Port scans
    - Authentication failures
    - Denial of service attempts
    - Failed connections

g. Performance Monitoring

i. Resource utilization should be tracked to ensure that the firewall is performing at the optimum level.
ii. Any surge in utilization of any of these parameters might be an indication of a system under attack.
iii. The IT Security team should determine threshold levels for peak and average usage for the following parameters: CPU utilization, Memory utilization, Hard disk free space, Concurrent connections

h. Change Control

Changes to the following should adhere to the change management process

i. OS Upgrade or installing a new patch on the firewall

ii. Firewall Application upgrade
iii. Installation or removal of additional components
iv. Integration of Firewall with third party components
v. Adding a new segment or modifying existing segments
vi. Adding a new Firewall Rule
vii. All the Firewall changes should be approved by the CIO or Director

i. Backup and Recovery

The IT Personnel appointed by CIO will be responsible for backup and recovery of the firewall. The following should be backed up soon after installation and successful testing of the firewall, and securely stored, which include:

i. Firewall OS files, Firewall application files, Configuration files, Firewall rule base, Routing table and Firewall log
ii. A full backup of firewall application or operating system files should be taken before any major changes to the firewall, including;
   - Upgrade of firewall OS or application
   - Installation of any additional component on the firewall (e.g., VPN, Hard disk drive)
   - Integration of the firewall with third party components
   - Backup of firewall logs and audit trails should be taken on a daily basis and archived for a period based on statutory requirements and for forensic analysis.
   - Backup of the firewall policy or rule base should be taken before and after the addition of new rules or modification to any existing rule.
   - By default, a backup of the firewall configuration and policies should be taken on a monthly basis, irrespective of whether changes are made to the firewall.

j. High Availability

i. Firewall redundancy should be configured based on the criticality of the applications and network segments or zones being protected.
ii. For critical applications or zones, firewalls should be configured in high availability mode to ensure minimum downtime for the respective applications.
iii. All communication between the primary and secondary firewall appliance should be secure using supported encryption technologies and dedicated communication channels such as cross-over cables.

k. Documentation

CIO should maintain detailed documentation of the firewall architecture and administration tasks. Firewall architecture documentation should include the following,

i. Network diagram with firewall segments or interfaces
ii. IP addresses of firewall interfaces and network devices connected to the firewall
iii. Routing table of firewall and connected devices
iv. Documentation on firewall administration tasks should include the following,
   - Installation and configuration of the firewall
   - Adding or deleting or modifying the firewall rule base, routing table, users & administrators
   - Backup or recovery of the firewall OS or application files

3.5. Intrusion Detection Systems [IDS]

An IDS is an application that monitors the events occurring in a system or network and analyzes them for signs of potential incidents, which are violations or imminent threats of violation of computer

security policies, acceptable usage policies, or standard security practices. An IPS has all the capabilities of an IDS and can also attempt to stop potential incidents.

To successfully protect a Web server using an IDPS, ensure that the IDPS is configured to;

a. Monitor network traffic to and from the Web server
b. Monitor changes to critical files on the Web server (file integrity checking capability)
c. Monitor the system resources available on the Web server host (host-based)
d. Block (in conjunction with the firewall) IP addresses or subnets that are attacking the organizational network
e. Notify the appropriate parties (e.g., IDPS administrator, Web server administrator, incident response team) of suspected attacks through appropriate means according to the organizational incident response policy and procedures
f. Detect as wide a variety of scanning and attacks as possible with an acceptable level of false positives
g. Log events, including the following details:
    i. Time or date
    ii. Sensor IP address
    iii. Manufacturer-specific attack name
    iv. Standard attack name (if one exists)
    v. Source and destination IP addresses
    vi. Source and destination port numbers
    vii. Network protocol

3.6. Antivirus

a. An anti-virus package should be installed on a Web Server System, if available on the platform.
b. All clients who access the web server for the purpose of administration and content management should use an antivirus package with latest signatures.
c. All documents and files hosted on the web server should be uploaded only after being checked for Virus and Trojans.
d. If the web server has provisions for uploading of files from users, appropriate mechanisms should be in place at the server side to ensure that the files are virus free.

04. DATABASE SECURITY

The following should be considered for securing a database system attached to the web application

a. Update latest Service Packs and Patches
b. Remove unnecessary services and protocols
c. Depending on importance of data, consider encryption
d. Secure the database server behind a firewall and use IDS to detect any intrusion attempts
e. The database server process should run as a user with minimum privileges, not on administrator level privileges.
f. Enforce a strict access control policy and secure coding practices for application developers
g. Audit trails logs on the database servers should be enabled
h. The database sever should not be assigned publicly accessible IP, and access to the database should be allowed only from the Web Server on a particular port only
i. Depending upon importance of data, fine grained record or row level auditing should be considered

05. SECURING CONTENT

5.1. Publishing Information on Public Web Applications

An organization should create a formal policy and process for determining and approving the information to be published on a Web server. Such a process should include the following steps [NIST, 2007b.]:

a. Identify type of information, the target audience and any negative ramifications of publishing the information on the Web.
b. Identify who should be responsible for creating, publishing, and maintaining content
c. Create or format information for Web publishing
d. Review the information for sensitivity and distribution or release controls
e. Determine the appropriate access and security controls.
f. Verify information to be published
g. Periodically review published information to confirm continued compliance with organizational guidelines.

5.2. Observing Regulations on the Collection of Personal Information

Governmental organizations with Web Applications should be aware of the appropriate and applicable laws, regulations, and agency guidelines.

5.3. Securing Active Content and Content Generation Technologies

Server-side content generators can create the following security vulnerabilities at the server:

i. Leakage of information that can assist a malicious user attacker thus allowing outsiders to deface or modify site content,
ii. When processing user-provided input there may be a vulnerability that can be exploited as the user guiles the application into executing commands supplied in the input stream

The Server-side content generator security considerations that are to be followed

a. The codes or content received from other programs or applications should be analyzed to identify security vulnerabilities.
b. In terms of defining the location of storing the Server-Side Content Generators, writable files and executable files placed in separate folders. Such writable files should not include scripts. Files created for code reusability should be placed in separate directories. include files should have an ".asp" extension instead of ".inc".

REFERENCES

NIST, 2007a, National Institute of Standards and Technology, Guidelines on Securing Public Web Servers, NIST Special Publication No. 800-44 Version 2, September 2007

NIST, 2007b, National Institute of Standards and Technology, Guide to Secure Web Services, NIST Special Publication No. 800-95, August 2007

OWASP, 2002, The Open Web Application Security Project [OWASP], A Guide to Building Secure Web Applications and Web Services, September 2002

CERT-In, 2004, Indian Computer Emergency Response Team [CERT-In], Web Server Security Guidelines, August 2004

[THIS PAGE WAS INTENTIONALLY LEFT BLANK]