

TECHNICAL GUIDELINES FOR WEB APPLICATION & WEBSITE SECURITY

This publication is a set of guidelines on secure web application and web site (herein referred to as web applications) 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 utilized by the private sector to enhance web application security.

The guideline shall be followed by the Senior Officials in charge of the subject of IT (Chief Innovation Officer, Director IT), IT Officers, Network Administrator, Information Security Officer, Associate Information Security Officer, Web Developers and other applicable Technical Staff of the public sector as well as the service providers who develop and host web applications on behalf of the government.

Through the guideline, government organizations must identify a prioritized, flexible, repeatable, performance-based, and cost-effective approach, including information security measures and controls to identify, assess, and manage cyber risks related to web applications and websites.

This document provides detailed guidance for developing individual organizational profiles pertaining to securing web applications and websites. The guide is not a one-size-fits-all approach to managing cybersecurity risk and the decision on how to apply it is left to the discretion of the implementing organization. If required, further guidance and assistance can be obtained through Sri Lanka CERT in terms of deploying the said control measures.

Version 1.0

Revision Date: 21/03/2022

Table of Contents

TECHNICAL GUIDELINES FOR WEB APPLICATION & WEBSITE SECURITY	2
INTRODUCTION	2
OBJECTIVE	2
SCOPE.....	3
TARGET AUDIENCE	3
1. SECURE WEB APPLICATION DEVELOPMENT	5
2. SECURE HOSTING	18
3. PERIMETER AND NETWORK DEFENSE.....	27
4. DATABASE SECURITY	33
5. SECURING CONTENT.....	35
REFERENCES	36

TECHNICAL GUIDELINES FOR WEB APPLICATION & WEBSITE SECURITY

INTRODUCTION

The functional and security requirements for a web application and website (hereinafter web applications and websites shall be referred to as web applications) 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 shall 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 shall comprise of implementing multitudes of security controls in different layers. As depicted in the Figure 1, securing web applications shall focus on following (1) secure web application development, (2) secure hosting, (3) perimeter and network security, (4) database defenses, and (5) securing content.

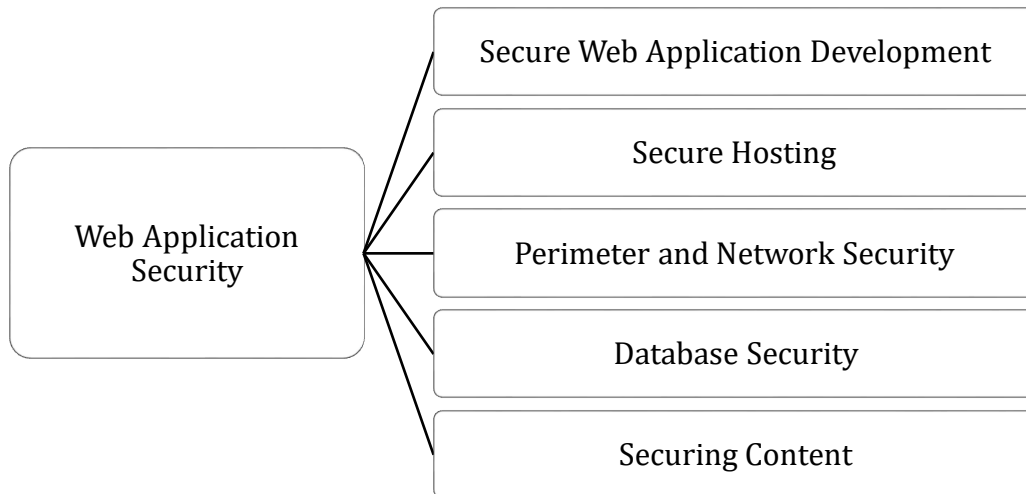


Figure 1 - An overview on Web Application Security

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. Government organizations are requested to follow the guidelines prescribed in this document.

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, Quality Assurance Officers, Information Security Auditors, Engineers and Web Developers.

SECTION ONE

SECURE WEB APPLICATION DEVELOPMENT

1. 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.

Proper understanding of the client's requirement by the developer is critical for the successful completion of application development. It is the client's responsibility to correctly specify the software requirements related to the application.

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

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

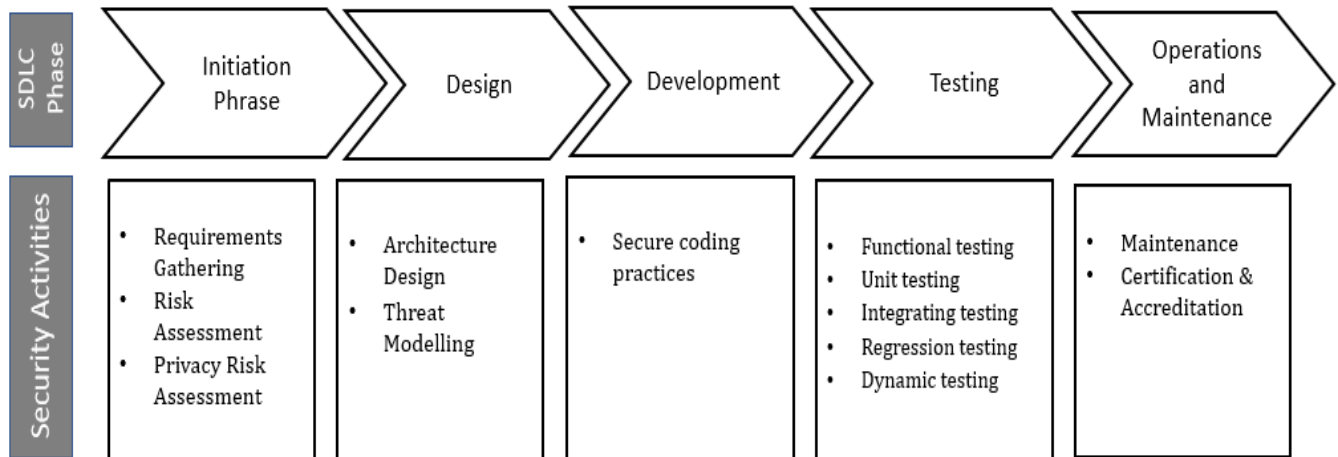


Figure 2 - Overview of Security by design lifecycle - Web Application Development

1.1. Initiation Phase

Software lifecycle begins 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 and security requirements of the web application shall be identified and documented on a System Requirements Specification (SRS). All functionality of the web application needs to be agreed upon and verified before moving ahead with the initiation process.

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.

The security requirements for the main functions of the web application shall 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.

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 shall be assessed and appropriate controls shall be identified to be built into the web application in order to mitigate the impact of the possible exploitation.

Risk assessment results shall 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 needs to be taken into consideration conducting the assessment. 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.

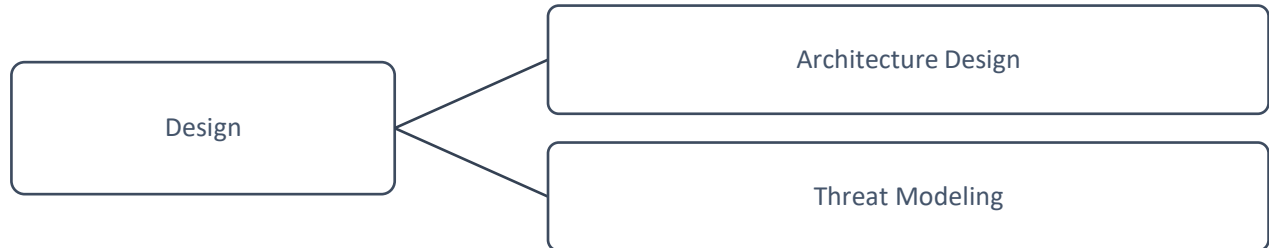


Figure 3 - Structuring of the Design Phase

1.2.1. Architecture Design

The overall structure of the web application (architecture) shall be designed by taking into account the functional and 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. The risk ratings may vary based on the features offered by the application, complexity, the purpose of the application, etc. Hence, the risk shall be analyzed according to it. For example, applications shall be designed to thwart brute force attacks, different types 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.

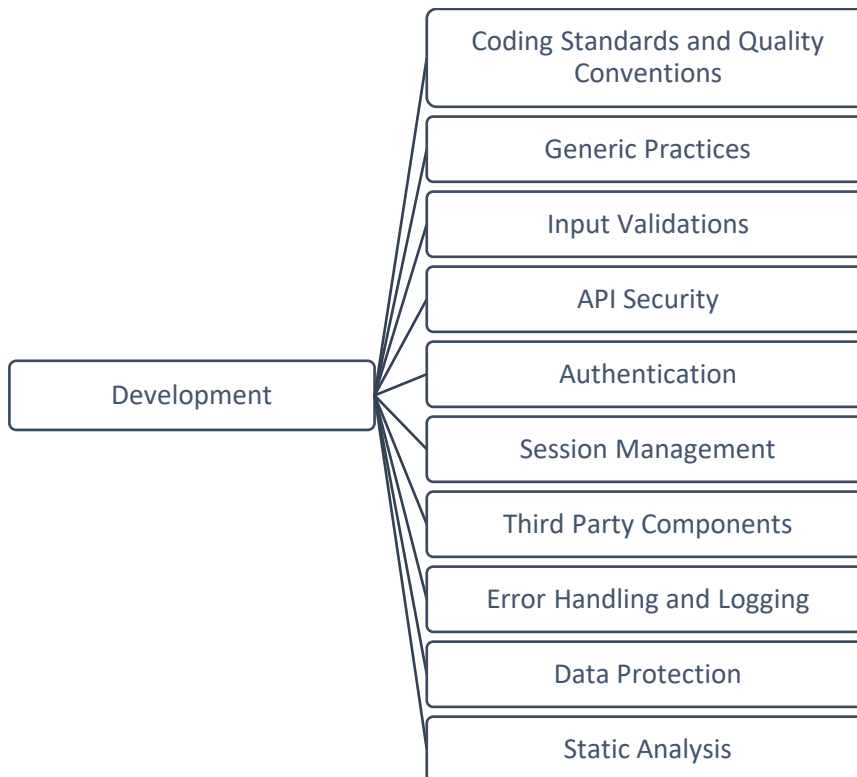


Figure 4 - Components of the Development Phase

1.3.1. Coding standards and Quality conventions

The simple rule behind coding standards and quality conventions 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 rather than creating new unmanaged code for common tasks.
- d. Maintain naming conventions of the variables throughout the code.
- e. Functions shall 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/relevant data is entered into the workflow, preventing malicious or undesirable data from persisting in the database and triggering malfunctions of various downstream components.
- b. Input validation shall happen as early as possible in the data flow, preferably as soon as the data is received from the external party. Input shall be validated as strictly as possible on arrival, given the kind of content which it is expected to contain. For example, personal names shall consist of alphabetical and a small range of typographical characters, and be relatively short; a year of birth shall consist of exactly four numerals; email addresses shall match a well-defined regular expression. Input that fails the validation shall be rejected and shall properly notified via error message.
- c. User input shall be HTML-encoded at any point where it is copied into application responses. All HTML meta-characters, including < > " ' and =, shall be replaced with the corresponding HTML entities (< > etc). In cases where the application's functionality allows users to author content using a restricted subset of HTML tags and attributes (for example, blog comments which allow limited formatting and linking), it is necessary to parse the supplied HTML to validate that it does not use any dangerous syntax; this is a non-trivial task.
- d. Data from all potentially untrusted sources shall be subjected to input validation.
- e. Input validation shall be applied on both Syntactic and Semantic level:
 - i. Syntactic validation shall enforce correct syntax of structured fields (e.g. NIC, Date, Currency symbol).
 - ii. Semantic validation shall enforce correctness of their values in the specific business context (e.g., start date is before end date, price is within expected range).
 - iii. If compressed / encrypted files are allowed, do validation check before decompressing / decrypting the file. The target path, level of compress and the uncompressed size shall be defined.
- f. Minimum and maximum of characters shall be defined and validated.
- g. 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 phase of the blacklisting approach, leaving any special characters could lead to security vulnerabilities [e.g. Name field – disallowing special characters (@, #, \$, *) & Numbers (0-9)].
- h. Client side & Server-side validation:
 - i. Client-side validation shall 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).
- i. File upload validation:
 - i. Extension type shall be validated.
 - ii. Max file size for the upload file shall be defined.

- iii. If compressed / encrypted files are allowed, do validation check before decompressing / encrypting the file. The target path, level of compress and uncompressed size shall be defined.
- iv. The file types allowed to be uploaded shall be restricted to those that are necessary for business functionality via whitelisting.
- v. The application shall perform filtering and content checking on any files, which are uploaded, to the server. Files shall be thoroughly scanned and validated before being made available to other users. If in doubt, the file shall be discarded.
- vi. If files shall be saved in a file system, consider using an isolated server with a different location to serve the uploaded files.
- vii. Uploaded directory shall not have any “execute” permission and all the script handlers shall be removed from these directories.
- viii. All the control characters and Unicode ones shall be removed from the filenames and their extensions without any exception.
- ix. Use input validation to prevent the metadata from being exploited. For example, remove any unnecessary metadata such as **exif** data from images and remove control characters from filenames and extensions.
- x. File types shall also be validated through MIME type. Further, the allowed file types shall be displayed on the UI before the upload and not via an error message.
- xi. It is highly recommended to only accept Alpha-Numeric characters and only 1 dot as an input for the file name and the extension; in which the file name and also the extension shall not be empty. When not filtered by the MIME type files shall be filtered for two extensions.
- xii. All necessary validations and values shall be configured in both server and client sides.

1.3.4. API Security

The Web Application shall use the appropriate API security measures and the respective protocols mentioned below.

- Use of the right API Security Protocol: Industry standard authentication protocols help reduce the effort of securing the API. The following can be considered.
 - Basic API Authentication: Basic authentication that uses TLS or SSL.
 - OAuth 1.0a: Cryptography signature value that combines the token secret, nonce and other request based information. Recommended for sensitive data application.
 - OAuth 2.0: All encryption is handled by TSL. Recommended for less sensitive data application.
 - JWT (JSON Web Token): This is a Security token which acts as a container for claims about the user, it can be transmitted easily between the Authorization server - Token Issuer, and the Resource server - Audience.

Furthermore, API Security Testing must be carried out on a periodic basis, at least annually.

1.3.5. 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 8 characters for passwords.
- e. Enforce a password policy.
 - i. Use of combination of alphanumeric (A-Z, a-z, 0-9) and special characters (@, \$, #, &, *)
 - ii. Must be at least fourteen (8) characters' long.
 - iii. Password shall be hashed using an appropriate password hashing algorithm and stored in the database.
 - iv. Password entry shall be obscured on the user's screen.
 - v. Enforce password changes periodically based on the established requirements in the policy.
 - vi. Request the current password in changing the current password
- f. Provide a password reset option and use an alternate channel to communicate the method of reset
- g. Return a generic message for both existent and non-existent accounts or incorrect passwords and usernames.
- h. Ensure that generated tokens or codes are randomly generated using a cryptographically safe algorithm.
- i. Storing Database Credentials: All Credentials must be stored using hashing algorithms. Ex-PBKDF2, bcrypt or scrypt and use a strong hashing algorithm.
- j. All applications or systems shall use appropriate 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.
- k. All user accounts must be locked out after a certain number of failed logins attempts.
- l. Use Key based authentication and disable generic administrative accounts
- m. An effective CAPTCHA must be implemented where applicable to prevent dictionary and brute-force attacks.

1.3.6. Session and Cookie Management

It is recommended that the developers implement the following measures.

- a. Generate a unique session ID
- 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 cookies are used in the application, all cookies must be verified before allowing access to the application sessions.

- d. HttpOnly and Secure flags must be enabled on every cookie and an expiration time has to be defined.
- e. All sessions must be implemented with idle or inactivity timeout. The business requirements could be taken into consideration while defining the session timeouts.
- f. Concurrent sessions and session bypassing shall be disallowed.
- g. Renewing the sessions are important aspects of session management. After a user has been created the web application shall regenerate a new session ID for the user session and renew it on the client. Once the new session is validated, the previous session shall be invalidated.
- h. User must be completely logged out after clicking on the logout button. The user shalln't be able to go back using the previous page.
- i. Implement CSRF token if the web application includes any type of form.

1.3.7. Use of Third-Party Components (TPC)

Followings shall be considered in the utilization of Third-Party Components (plugins, libraries, APIs, codes and etc.),

- 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 shall 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 shall be carefully assessed.

1.3.8. 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 shall 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 shall be monitored, recorded and reviewed regularly.

- h. Log Transition shall be performed with a secure transmission protocol and the origin shall be verified.
- i. The best practice is to maintain a separate log server where applicable.

1.3.9. 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 shall include time, user info, error message and other useful information.

1.3.10. Static Analysis

The static analysis shall 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 shall be assigned. Also, the recent error types and reviewing techniques shall 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 shall be identified and noted along with severity level identifications. All findings shall be documented in a report.
- e. **Rework** – All errors shall be remediated, and responses are to be provided. The author shall 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 shall be documented.

1.4. Testing and Deployment 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 functionality testing, unit testing, integration testing, regression testing, security testing and application deployment testing.

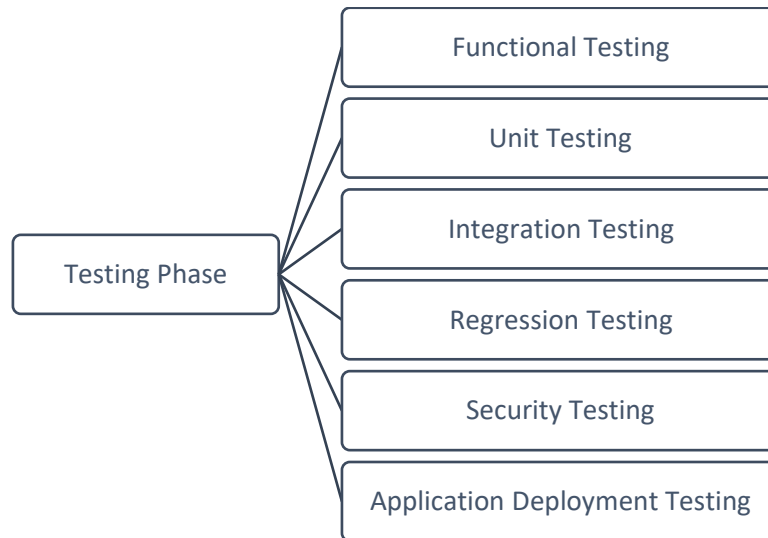


Figure 5 - Types of Testing

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 shall 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 shall ensure regression testing is followed carefully after any changes in the web application to ensure functionality, performance and protection.

1.4.5. Security Testing

Security Testing is to be performed to identify the threats in the system and measure its potential vulnerabilities, so the threats can be encountered and the system does not stop functioning or can avoid being exploited. It also helps in detecting possible security risks in the system and helps developers to fix the problems through coding. Testing shall be performed according to a standard such as OWASP top 10, SANS 25, etc.

1.4.6. Application Deployment Testing

Once functional and security testing is performed in both the application and server environment, it needs to be deployed from the staging environment to the production environment which is your live environment. It is essential to ensure that both the staging and production environments are identical when pushing changes or updates from one deployment environment to another.

1.5. Operations and Maintenance Phase

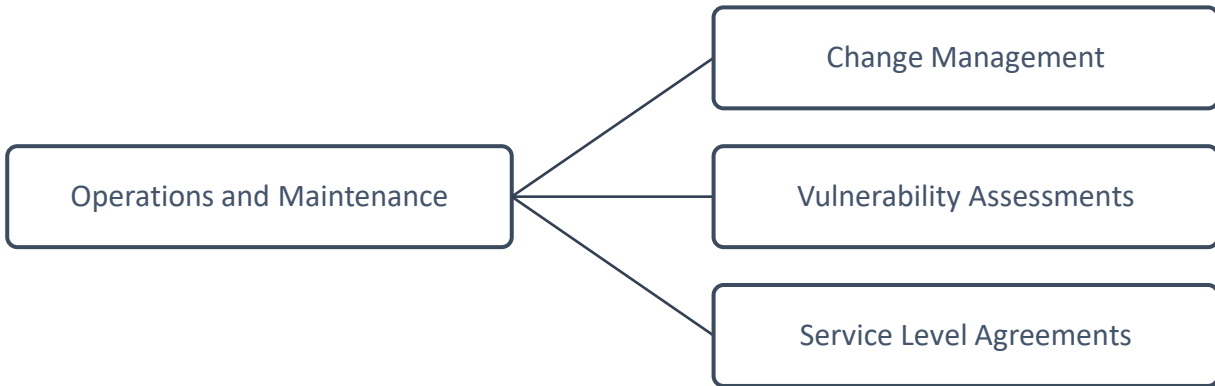


Figure 6 - Components of Operations and Maintenance

- 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 are 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 of security. Each time a new website functionality is added the website shall go through a Vulnerability Assessment.
- c. 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 and need to ensure the website is always safe and secure.
- d. Any Government web application shall 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.
- e. Government organizations must maintain a Service Level Agreement (SLA) with the external service providers/suppliers such as web developers and web hosting service providers supported by a documented SLA or contract which commits both parties to the agreed terms of services, including details of authorization, level of information security and service delivery, issues of liability, reliability of services and response times for the provision of services. For confidential information, the specific mechanisms used for the transfer of such information shall be consistent and Non-Disclosure Agreements need to be in place.
- f. The Government organization shall regularly monitor, review and audit supplier service delivery to ensure that the supplier delivery and information security terms and conditions of the agreements are being adhered to and managed properly.

SECTION TWO

SECURE HOSTING

2. 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 shall 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 that will be used to host web server
 - v. Identify the network services that will be provided
 - vi. 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
- b. Identify any network service software, both client and server, to be installed on the Web server and any other support servers
- c. 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
- d. Determine how the Web server will be managed (e.g., locally, remotely from the internal network, remotely from external networks)
- e. Determine whether appropriate physical security protection mechanisms are in place at the hosting location (e.g. locked rooms, card reader access, security guards)
- f. Availability of Redundant power supplies and Internet connections.
- g. Adequate precautions in terms of security of the location including the DR site.
- h. Determine the availability of a redundant web server at the DR Site

2.1.1. Selecting a Hosting Platform (Co-Location)

- a. Ensure the hosting uptime and availability is in par with organizational requirements
- b. Ensure Physical security and monitoring requirements are met (E.g. Whether the database server is on a third party data center or in LGC Data center, it must be located within a secure, climate-controlled environment.)
- c. Ensure High Availability (Power, Connectivity)
- d. Technical support and services (Assistance during downtime)
- e. Ensure Bandwidth requirement in between production and DR site
- f. Availability of hardware firewall and web application firewall (WAF)
- g. Protect data at rest and in transit with encryption and VPN
- h. Ensure data backup and recovery process meet the organization's needs.
- i. Availability of efficient patch management practices
- j. Identity and Access Management (IAM) to gain access to organization resources
- k. Perform server, database and application audits on a regular basis
- l. Perform vulnerability and system configuration assessments.

It is strongly recommended that apart from the aforementioned requirements, the organization's security policy is also to be taken into consideration when selecting an appropriate hosting platform

2.1.2. 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 shall be a general purpose, Trusted, Web server appliance, virtualized or pre-hardened operating system.
- b. Ensure that the operating system has;
 - o Minimal exposure to vulnerabilities
 - o Ability to restrict administrative or root level activities to authorized users.
 - o Ability to control access to data on the server
 - o Ability to disable unnecessary network services that may be built into the OS or server software
 - o Ability to control access to various forms of executable programs, such as CGI scripts and server plug-ins
 - o Ability to log appropriate server activities to detect intrusions and attempted intrusions
 - o Provision of a host-based firewall capability
 - o 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

- o Security Certification Level of the chosen platform
- o Level of support provided by the Vendor
- o Compatibility and support concerns of the Software to be used on the platform
- o Support of Security features on the platform (Authentication, Levels of Access control, Remote logging and administration)
- o Minimize the operating system with only essential services by removing all operating system and network services that are not required
- o Keep operating systems and application software up to date with the latest service packs and patches
- o Strong password policies to be enforced

- Enabling detailed logging including failed logging, etc.
- 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 shall 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 shall 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 shall 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 communication (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)
- g. Ports used for common services (Change default port numbers and limit IP binding)

2.2.4. Configure Operating System User Authentication

The following steps shall 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 shall 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.
- f. Use .htaccess to restrict access to files and directories of critical resources
- g. Configure file permissions according to security standards. Ensure that configuration files such as “.htaccess” or “web.config” cannot be replaced using file uploaders. Ensure that appropriate

settings are available to ignore the “.htaccess” or “web.config” files if uploaded in the upload directories. (Need to be in file permission section)

- h. Monitor the crontab and /tmp for malicious activities

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 of the Web Server

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 shall 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 shall 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 test files from the server, including scripts and executable code.

2.3.2. Configuring Access Controls

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

- Limit the access of the Web server application to a subset of computational resources.
- 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 shall 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 shall 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 shall be restricted and which shall 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 shall not include sensitive URIs hidden in the source code.

Collecting cookies shall be disabled unless there is a need to gather the data on the site. This shall only be enabled 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 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 shall take place regularly (e.g., daily) and when a suspicious activity has been noted or a threat warning has been issued.
- b. Log files shall 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 and an event management (SIEM) software.
- d. Log files shall be backed up and archived regularly.

2.4.3. Web Server Backup Procedures

The Web server [Data and the OS] shall 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 shall to identical to the production or live web server on the organization's intranet.

2.4.5. Maintain a Formal Copy of Organizational Web Content

- a. The government organization shall maintain a formal copy of their public Web Applications on a host that is inaccessible to the Internet.
- b. Consider performing automatic updates from the copy to the Web server periodically as this will overwrite a Web Application defacement automatically [NIST, 2007a].

2.4.6. Incident Handling

- a. Government organizations, including contractors, service providers must report all suspected information security incidents to the relevant government officers without any delay. Upon the reporting of an Incident, the respective Information Security Officer activate the incident response plan upon the approval of the Head of the organization.
- b. Thereafter, the respective government officers must perform an initial analysis to determine the incident's scope, attack methods, targeted vulnerabilities, and the criticality of the Incident with reference to the Incident Classification Framework and is to prioritize subsequent activities, including Containment of the incident.
- c. All suspected incidents must be logged in the Incident Register, and perform further analysis as stipulated in the Handbook of Information Security guidelines for government organizations. Government organizations shall identify, collect, acquire and preserve data related to the incident as evidence.
- d. As determined by the respective officers, the government organization is advised to report critical information security incidents to Sri Lanka CERT immediately for technical advice and handling. The Information Security Officer of the Government organization shall report all the incidents to the Head of the Organization. At the closure of the Incident Response Process, the relevant government officials shall submit a detailed report to the management to take further necessary action if any.

2.4.7. 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 and isolate the compromised systems or take other steps to contain the attack.
- b. Take a backup of the file system of the compromised application which will help in investigations
- c. Consult the appropriate personnel in the management immediately.
- d. Investigate similar hosts to determine if the attacker also has compromised other systems.
- e. Analyze the intrusion with the support of experienced and qualified experts.
- f. Perform a security assessment on the compromised application and the underlined system and remediate identified vulnerabilities.
- g. Restore the system.
 - Either install a clean version of the OS, applications, necessary patches, and Web content; or restore the system from backups
 - Disable unnecessary services and apply all patches.
 - Change all passwords
 - Recommend to use a Password manager to store passwords.
 - Reconfigure network security elements (e.g., firewall, router, IDPS) to provide additional protection and notification.
 - Restore the most trusted and immediate backup.
 - Test system to ensure security & reconnect it to network.
- h. Monitor system and network for signs that the attacker is attempting to access the system or network again.
- i. Document lessons learned.

2.4.8. Backup & Restoration

- a. The government organization shall have a backup policy and a strategy to ensure that critical information is recoverable if lost. It shall include backup data, audit logs, system information, configurations or any other information that are necessary to restore normal operations in an event of a disaster.
- b. The secondary storage media includes but is not limited to Tape cartridges, CDs, DVDs, Mirror Disks (onsite or offsite), or network storage shall be used for data backup purposes.
- c. Data written to backup media shall be preserved as per regulatory requirements of the government.
- d. Backups shall be stored offsite in a safe location, physically distant from the data processing center to facilitate disaster recovery efforts. Offsite backup storage location must comply with the organization's policy on physical and logical controls. It shall be stored as per the asset classification scheme.
- e. The government organization must ensure that backups must be scheduled regularly. The standard backup schedule (frequency of taking backups) shall be based on the RPO defined by the organization.
- f. Backups shall be periodically tested and restored, at least annually, to ensure that they are recoverable.
- g. The government organization must ensure that a catalog of information regarding the version and location of data file is maintained for the specified retention time and protecting this catalog against unauthorized disclosure.
- h. Based on the classification of information systems with reference to the impact of failure, the organization shall maintain an alternative data processing facility at geographically distance location.

2.4.9. 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.

2.4.10. 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 shall only be performed through secure connections. Implement secure administrative hosts (Jump Servers) to adhere to secure development and administrative practices.

SECTION THREE

PERIMETER AND NETWORK DEFENCE

3. PERIMETER AND NETWORK DEFENCE

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 shall have at least following segments.
 - i. Internet Segment or Public Server Segment (Web, Mail, DNS Servers)
 - ii. Internal Segment
- c. The Web Server shall be placed in the secure server security segment (DMZ or screened subnet) isolated from the public network and organization's internal network. Web Servers shall 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. 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 shall be considered in configuring Firewalls.

- Update default security settings
- Default Firewall settings shall be updated.
- Default vendor supplied user accounts shall be disabled after setting their password to a complex value.
- The firewall shall not have any additional services running that can be accessed remotely.

3.3.1. 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;

- Sensitive and critical web applications or servers that are accessed only internally shall be hosted on the most protected segment of the firewall.
- Applications accessed internally as well as by external sources shall 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 shall terminate on a separate interface of the firewall.
- Wide Area Network (WAN) links connecting to the data center shall terminate on a separate interface of the firewall.

- 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 shall be considered for protecting these terminals.

3.3.2. Rule Base Creation

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

- By default, the firewall MUST have a DENY ALL policy, with access granted on a need to do basis. The firewall shall have a rule to deny all access that is not explicitly allowed.
- 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.
- The firewall rule base shall restrict access to required ports on the target machine. The source field in the rule base shall 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.
- Application or servers which are directly accessed from a public network such as the Internet shall be moved to a separate segment (Demilitarized Zone) of the firewall. The IP address of the server shall be NATed with a public IP address.
- For connections with third parties, NATing shall be performed using any available private or public address slots.
- Access to administrative ports including SSH and Microsoft Windows Terminal services on protected servers shall have user ID based authentication at the firewall in addition to the source IP address.
- 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.
- Password policies for these user accounts including password expiry, password history, and password complexity shall be enforced. Account lockout shall be configured to prevent password cracking attempts. It shall be ensured that these user credentials are transmitted in encrypted format from the user PC to the firewall.
- 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 shall 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.
 - 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 shall be enabled on this rule.

3.3.3. Rule Base Change

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

3.3.4. Administrative Access

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

3.3.5. Audit Logging

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

3.3.6. Performance Monitoring

- i. Resource utilization shall 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 shall determine threshold levels for peak and average usage for the following parameters: CPU utilization, Memory utilization, Hard disk free space, Concurrent connections.

3.3.7. Change Control

Changes to the following shall 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 shall be approved by the CIO or Director IT

3.3.8. Backup and Recovery

The IT Personnel appointed by CIO or the Head of the organization will be responsible for backup and recovery of the firewall. The following shall 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 shall 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 shall 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 shall 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 shall be taken on a monthly basis, irrespective of whether changes are made to the firewall.

3.3.9. High Availability

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

3.3.10. Documentation

The CIO or the appointed officer shall maintain detailed documentation of the firewall architecture and administration tasks. Firewall architecture documentation shall 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 shall 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

3.4. Intrusion Detection and Prevention Systems [IDPS]

An Intrusion Detection and Prevention System (IDPS) 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 IDPS 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 a wide 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

3.5. Antivirus

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

SECTION FOUR

DATABASE SECURITY

4. DATABASE SECURITY

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

- a. Update to the latest and stable Service Packs and Patches.
- b. Remove unnecessary services and protocols.
- c. Secure the database server behind a firewall and use IDS to detect any intrusion attempts.
- d. The database server process shall run as a user with minimum privileges, not on administrator level privileges.
- e. Enforce strict access control and secure coding practices for application developers.
- f. Audit trails logs on the database servers shall be enabled.
- g. The database sever shall not be assigned publicly accessible IP, and access to the database shall be allowed only from the Web Server on a particular port only.
- h. Depending upon importance of data, row level auditing shall be considered.
- i. Depending on importance of data, consider encryption. Protected data is to be encrypted during transmission over the network using strong encryption measures. Backups of the database shall also be in an encrypted format. Key management procedures for decrypting backups shall be documented, available to more than one person and approved by the data owner.
- j. The backup and recovery procedures are documented and meet data owner's requirements. Backup and recovery procedures are periodically tested and Backup retention intervals are documented and sufficient to meet the business resumption requirements and expectations of the data owner.
- k. Only protected data required for the business function is kept within the database. Protected data is never used as a key in a table. When possible, historical information is purged when no longer required. Hashing functions are applied to protected data elements before storing if the data is only required for matching purposes. If possible, disassociate protected data from personally identifiable information and keep offline until needed. If data transfers are required for other applications, notify them of protected data and its security requirements.
- l. Protected data in non-production environments is held to the same security standards as production systems.

SECTION FIVE

SECURING CONTENT

5. SECURING CONTENT

5.1. Publishing Information on Public Web Applications

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

- a. Identify type of information, information classification category, the target audience and any negative ramifications of publishing the information on the Web.
- b. Identify who shall 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 shall 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 shall 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 shall be placed in separate folders. Such writable files shall not include scripts. Files created for code reusability shall be placed in separate directories. include files shall 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