Security Related Guidelines

Recommendations & Best Practices

Security related Recommendations & Best Practices:

- Web server audit certificate, web server and OS level hardening need to be in place for the production server before making the application live.
- Website audit should be done atleast once a year or when there is any change in the application.
- No new web pages are to be added without proper security audit.
- Server-side issues should be taken care by hosting provider.
- Security Socket layer (SSL) should be implemented on the main hosting URL (including sub folders).
- Site's 'root' folder must be hosted with the 'Read' permissions.
- The uploaded files (images) are stored in file system. 'Read' and 'Write' permissions should be granted to 'folders' where files are stored. "Execute" permission should not be granted to this above folder.
- The application should connect with the database using a low privileged database user having minimum required database privileges.
- Directory browsing should be disabled.
- Default and sample applications should be removed from the web server.
- Credentials used during testing should be mandatorily changed in production environment.
- The application should be upgraded to latest stable version of Apache during production deployment.
- WAF (Web Application Firewall) can be implemented for additional protection.
- Housekeeping on regular interval.
- It is recommended to implement and use latest version of TLS for website.
- It is recommended to patch all the issues as mentioned in final audit report once the application is deployed in production server.
- The server should be physically protected from unauthorised access.
- Web server should go through VA/PT on regular intervals as server-side vulnerabilities can be used to exploit the web applications.
- Any change in the source code of the audited web applications requires a VA/PT.
- Entire website may be deployed over TLS 1.1 or TLS 1.2 or higher.
- Web- app/site may be considered safe for hosting, with read only permission, except write permission on "Uploads" folder where files may be allowed to be uploaded and script execute permission where scripts may be allowed to be used for execution. It needs to be ensured that no combined Write+execute permission is give on any folder/site.
- In the production environment, it needs to be ensured that any component with known vulnerabilities/old versions is upgraded to the latest stable versions for the same.
- As a best practice, web-app/Sites may be got audited again at periodic intervals, or as per organisation's policy/regulatory guidance. To enhance user experience,

appropriate User Acceptance Test (UAT) and Functional Testing of the web-app/Sites vis-à-vis SRS (System Requirement Specification) may also be got carried out, if not already done.

- Never trust user input.
- Never insert untrusted data except in allowed locations.
- HTML escape before inserting untrusted data into HTML element content.
- Use whitelists in place for Black lists for input filtering.
- Unless the web server is being utilized to share static and non -sensitive files, enabling directory listing is considered a poor security practice.
- This can typically be done with a simple configuration change on the server. The steps to disable the directory listing will differ depending on the type of server being used (IIS, Apache, etc). If directory listing is required, and permitted, then steps should be taken to ensure that the risk of such configuration is reduced.
- Requiring authentication to access affected pages.
- Adding the affected path to the 'robots.txt' file to prevent the directory contents being searchable via search engine.
- Ensuring that sensitive files are not stored within the web or document root.
- Removing any files that are not required for the application to function.
- To effectively prevent framing attacks, the applications should return a response header with the name X- Frame-Options and the value DENY to prevent framing all together, or the value SAMEORIGIN to allow framing only by pages on the sameorigin as the response itself. Note that the SAMEORIGIN can be partially bypassed if the application itself can be made to frame untrusted websites.
- WordPress admin page should be disabled to public view.
- Error message should be disabled.
- Sanitizes this kind of information from the output.
- During Production Hosting ○ Security Harden the OS/Web Server/DB Server ○ Install and configure SSL Certificate on the server ○ Set Complex alphanumeric Login Password ○ Set SECURE & HTTP Only params ON, for website cookies ○ Read-Write-Execute permission to be given to the folder path

  "/var/www/html/sites/default/files/"
- Disable OPTIONS HTTP method on the server
- Disable server and X-Powered-by header on the server.
- After Hosting ○ Regularly Update OS, Web & DB Server and Application ○ Re-Audit the web application, at-least annually.

- User latest version of web server, Database server and Application such as PHP, JSP, ASP, JBoss etc. Ensuring that web servers run the most up-to-date software is necessary to address these risks and reduce the attack surface.
- Apply appropriate updates/patches on the OS and Application software when available. Maintain up-to-date antivirus signatures and engines.

- Adopt latest components of secure software development, establish a baseline of security awareness for design, build and deploy secure software and applications.
- Validate and sanitize all user input, and present error messages that reveal little or no useful information to the user to prevent SQL injection attacks.
- When selecting plugins and applications, consider their age, amount of installs and updates. Install software only from trustworthy sources to protect from possible malware infections. Change default settings, such as login credentials, to prevent them from being used in hacking attempts.
- Enable, maintain and preserve logs of different devices and servers and maintain the same for all the levels. Logs should be rotated periodically.
- Scan and test databases, network, web server and web applications throughout the development lifecycle from design to development to testing. Further, conduct complete security audit periodically, after every major configuration change and plug vulnerabilities found.
- Different user categories must be permitted with only one bare minimum level of access privileges they need for their purposes. Never allow unrestricted files uploads and limit these uploads only to what users need.
- Apply Security Information and Event Management (SIEM) and/or Database Activity Monitoring (DAM) solutions.
- Search all the websites hosted on web server or sharing the same DB server for the malicious webshells or any other artefact.
- Periodically check the webserver directories for any malicious/unknown web shell files and remove as and when noticed.
- Perform regular backups of all critical information to limit the impact of data or system loss and to help expedite the recovery process. Ideally, these data should be kept on a separate device, and backup should be stored offline.
- Host sensitive ICT infrastructure on dedicated servers and avoid shared hosting. Networking segmentation and segregation into security zones-help protect sensitive information and critical services. Separate administrative network from business processes with physical controls and virtual area networks.
- Ensure strict applications of security control at the data centres. In addition, perform periodic and regular compliance of such security controls.
- Disable Remote Desktop Connections, employ least-privileged accounts.
- Use a web application firewall (WF) to control access to web applications using ruled designed to recognise and restrict suspicious activities, such as SQLi, XSS and exploitation of vulnerabilities. Choose a WAF that has bot detection and mitigation capabilities to prevent bad bots from accessing application data.
- Disable unnecessary services on agency workstations and servers.
- Restrict users' abilities (permissions) to install and run unwanted software applications.