CYPRESS
DATA DEFENSE

The Do-it-Yourself Web Application Testing Checklist

www.cypressdatadefense.com

Web application testing helps discover potential vulnerabilities and weaknesses in the application before the application is moved into the production environment or is deployed.

Many aspects of the application are exercised during web application testing such as input validation, authentication, authorization, etc. This helps determine potential vulnerabilities that could be exploited by attackers and ensure the web application is safe for use.

Security professionals perform thorough testing which encompasses much more breadth and in-depth testing. However, you can perform some basic security testing by yourself to get things started before you bring them in.

This might include simple tests such as checking whether the site is storing sensitive data in an encrypted format, or if you have a proper backup in place.

By performing these security tests yourself, you'll get a leg up on security testing and have an idea if there are any glaring vulnerabilities that could be exploited by attackers.

Here is a do-it-yourself web application testing checklist:

Functional Security Testing

To start with, you'll want to do some very basic functional security testing. Are the most basic controls in place?

Here are a few examples of test cases for functional security testing:

- ❏ Are all of the mandatory fields validated?
- ❏ Are the user credentials protected properly while in transit (encrypted) and in storage (hashed)?
- ❏ Does the site prompt for sensitive information such as credit card details, birth date, etc.? Does it need to be stored? If so, is it stored in an encrypted format? Are there any specific requirements that must be followed (PII, PCI, HIPAA, etc)?
- ❏ Are the data limits functioning as intended?
- ❏ In the event of a failed login, is the application leaking information as to whether or not the password was incorrect or if the user doesn't exist?
- ❏ Are proper password policies implemented in the system?
- ❏ Are all the fields for special characters functioning correctly? Can a user enter unexpected characters?
- ❏ Are the numeric fields allowing negative numbers or zero when they shouldn't?
- ❏ If an action fails, is the user being redirected to a custom error page?

Security Testing

Once you've done some of the most basic functional security testing, you'll want to look at some more security test cases.

Here are a few examples of test cases for security testing:

Configuration Management

- ❏ Is the administrative interface hosted separately from the application? Is it secure?
- ❏ Do you have a backup that regularly stores data? Is it encrypted? Have you restored it?
- ❏ Is the site secured with the TLS protocol? What key lengths are used (2048 bit symmetric, 256 bit symmetric)? What cryptographic algorithms are used (AES, SHA256)?
- ❏ Are the servers hardened?
- ❏ Are appropriate rules in place for the firewall and web application firewall?
- ❏ Are you only handling expected request types and rejecting all others?
- ❏ Do you have limits on request sizes?

Auditing

- ❏ Is a log created for every record add/update/delete operation?
- ❏ Are all user actions audited with enough information to trace all of their activities?
- ❏ Is the audit data stored in a safe format that cannot be altered?

Data Transmission

- ❏ Is the HTTPS certificate valid? What is the duration of the certificate?
- ❏ Are all pages being delivered only via HTTPS?
- ❏ Are the session tokens being delivered only via HTTPS (marked Secure)?
- ❏ Are the session tokens protected with HTTPOnly?

Session Management

- ❏ Are values in URLs and cookies handled securely?
- ❏ Is there a maximum lifetime for a session? Does it sufficiently limit the attack window (i.e. 20 minutes)?
- ❏ Is the session terminating once the maximum lifetime has been reached?
- ❏ Is sensitive data displayed after session timeout?
- ❏ Is the user able to navigate the site when logged out of the system or the user session has expired?
- ❏ Can users have multiple simultaneous sessions? Should this be allowed?
- ❏ Are new session tokens being issued when the user logs in and logs out?
- ❏ Is there consistent session management throughout the application?

Authentication

- ❏ Is autocomplete turned off for sensitive information?
- ❏ Can the user reset or recover their password securely?
- ❏ Is the user locked out after multiple failed attempts to login? How can they try to login again?
- ❏ Is multi-factor authentication enabled in the application?
- ❏ Is the CAPTCHA functionality enabled and working properly?
- ❏ Is the logout functionality working as intended?

Authorization

- ❏ Can the user bypass authorization by forced navigation or other methods?
- ❏ Are the access privileges implemented correctly?
- ❏ Do you have proper vertical access control?
- ❏ Do you have a proper horizontal access control that ensures users can only access their authorized information (one user can’t access another user’s account)?
- ❏ Does the application have missing authorization in any locations?

File Uploads

- ❏ Does the application have a maximum file upload size limit?
- ❏ What are the permitted upload frequency and the total number of files that can be uploaded by a user?
- ❏ Are all file uploads checked with anti-virus scanning software?
- ❏ Are the files stored in the database or filesystem and are appropriate protective measures in place?
- ❏ Does the application sanitize unsafe filenames?
- ❏ Are the uploaded files downloadable by others?
- ❏ Are all files integrated with authentication and authorization policies in place?

Cryptography

- ❏ Is there any weak algorithm usage?
- ❏ Does the application use individual salts for hashes?
- ❏ Does the application use individual Initialization Vectors (IVs) for encrypted items?