www.certifiedsecure.com
info@certifiedsecure.com
Tel.: +31 (0)70 310 13 40
Loire 128-A
2491 AJ The Hague
The Netherlands

Certified Secure Web Application Secure Development Checklist

About

Certified Secure exists to encourage and fulfill the growing interest in IT security knowledge and skills. We stand for openness, transparency and the sharing of knowledge; making sure everybody can experience and enjoy IT security. Security is serious fun!

All Certified Secure certifications, products and training are developed by IT security professionals with international recognized expertise. Our involvement in the IT security community worldwide, ensures relevant and high-quality standards. Delivering a wide variety of online challenges, videos, tools and more, Certified Secure is the authoritative source for practical IT security know-how.

Scope

This checklist can be used as a standard when developing or auditing a web based application. For security testers a separate Web Application Security Test Checklist is available from https://www.certifiedsecure.com/checklists.

Usage

Developers should use this checklist when developing a web application. Software auditors should use this checklist when performing a white box source code audit of a web application. A risk analysis for the web application should be performed before starting with the checklist. Every control on the checklist should be completed or explicitly marked as being not applicable. Once a control is completed the checklist should be updated with the appropriate result icon and a document cross-reference.

The completed checklist should never be delivered standalone but should be incorporated in a report detailing the risk analysis and checklist results. When performing a white box source code audit the report should also include the scope and context of the performed audit.

License

Result Icon Legend

Icon	Explanation
✓	Test was performed and results are okay
✗	Test was performed and results require attention
■	Test was not applicable

#	Certified Secure Web Application Secure Development Checklist	Result	Ref
1.0	**Generic**		
1.1	Write clean, maintainable and readable code		
1.2	Use assertions to double-check critical (security) assumptions		
1.3	Separate the presentation layer from the data/business logic layer		
2.0	**Deployment**		
2.1	Install security updates for all deployed software		
2.2	Never use unsupported or end-of-life software versions		
2.3	Disable the HTTP TRACK and TRACE methods		
2.4	Remove all extraneous functionality		
2.5	Complete the Server Configuration Checklist for all servers		
3.0	**Information Disclosure**		
3.1	Remove all extraneous files and directories		
3.2	Prevent extraneous directory listings		
3.3	Implement access controls for debug functionality		
3.4	Only display generalized log and error messages to users		
3.5	Prevent direct propagation of errors and exceptions to users		
3.6	Never store sensitive information in the robots.txt file		
3.7	Remove sensitive information from publicly accessible source code		
3.8	Never disclose internal addresses		
4.0	**Privacy and Confidentiality**		
4.1	URLs must never contain sensitive information		
4.2	Never store unencrypted sensitive information at the client side		
4.3	Disallow indexing of pages with sensitive information		

#	Certified Secure Web Application Secure Development Checklist	Result	Ref
4.4	Never include content from untrusted (external) sources		
4.5	Implement anti-caching measures for pages with sensitive information		
4.6	Transmit all sensitive information via secure connections		
4.7	Deny access to sensitive information via insecure connections		
4.8	Configure SSL/TLS for all pages on sites processing sensitive information		
4.9	Never include non-SSL/TLS resources in SSL/TLS pages		
4.10	Emit the HSTS header for full SSL sites		
4.11	Configure SSL/TLS to use only strong keys, ciphers and protocols		
4.12	Ensure "cache keys" cover all information used to generate cached content		
5.0	**State Management**		
5.1	Implement state management at the server side		
5.2	Explicitly validate all state transitions		
5.3	Prevent race conditions		
6.0	**Authentication and Authorization**		
6.1	Configure mandatory authentication for all non-public services		
6.2	Implement a default-deny policy for all authorization mechanisms		
6.3	Implement authorization for every privileged operation		
6.4	Implement authorization and authentication at the server side		
6.5	Implement authorization and authentication at a centralized location		
6.6	Remove all test and default accounts from production systems		
6.7	Use a cryptographically strong PRNG for generating tokens		
6.8	Use a sufficiently large size for generated tokens (at least 128 bits)		
6.9	Never implement authentication or authorization based on obscurity		

#	Certified Secure Web Application Secure Development Checklist	Result	Ref
6.10	Enforce the usage of strong passwords		
6.11	Sufficiently hash, salt and stretch stored passwords		
6.12	Implement rate limiting for all authentication functionality		
6.13	Require re-authentication when changing credentials		
6.14	Always implement logout functionality		
7.0	**Cryptography**		
7.1	Use only widely accepted and proven cryptographic algorithms		
7.2	Use existing, well-tested implementations of cryptographic algorithms		
7.3	Never use weak, untrusted or expired certificates		
7.4	Use a sufficiently large size for cryptographic secrets (at least 128 bits)		
7.5	Ensure cryptographic signatures cover all relevant data		
7.6	Prevent replay attacks		
8.0	**User Input**		
8.1	Encode data in the context where it is used		
8.2	Accept only specific data types (string, integer, etc.) for user input		
8.3	Never process unexpected user input		
8.4	Use parameterized SQL queries		
8.5	Encode data before it is used in SQL statements		
8.6	Encode data before it is used in filenames or directories		
8.7	Encode data before it is used in a HTML context		
8.8	Encode data before it is used in a JavaScript context		
8.9	Encode data before it is used in system commands		
8.10	Encode data before it is used in XML documents		

#	Certified Secure Web Application Secure Development Checklist	Result	Ref
8.11	Encode data before it is used in XPath queries		
8.12	Encode data before it is used in XSL(T) style sheets		
8.13	Encode data before it is used in SSI statements		
8.14	Encode data before it is used in HTTP headers		
8.15	Encode data before it is used in HTTP parameters		
8.16	Encode data before it is used in LDAP queries		
8.17	Encode data before it is used in regular expressions		
8.18	Encode data before it is used in eval functions		
8.19	Encode data before it is used in expression languages		
8.20	Encode data before it is used in other contexts		
8.21	Prevent insecure deserialization of data		
9.0	**Sessions**		
9.1	Deny all data/state mutation requests without a valid CSRF token		
9.2	Deny all privileged operation requests without a valid CSRF token		
9.3	Use a cryptographically strong PRNG for generating CSRF tokens		
9.4	Use a sufficiently large size for CSRF tokens (at least 128 bits)		
9.5	Revoke the session on logout		
9.6	Regenerate the session identifier on login		
9.7	Regenerate the session identifier when changing credentials		
9.8	Revoke existing sessions when changing credentials		
9.9	Set the Secure flag for session cookies		
9.10	Set the HttpOnly flag for session cookies		
9.11	Set a restrictive "SameSite" attribute for session cookies		

#	Certified Secure Web Application Secure Development Checklist	Result	Ref
9.12	Set a restrictive "Domain" attribute for session cookies		
9.13	Set a restrictive "Path" attribute for session cookies		
9.14	Set the "__Host-" or "__Secure-" cookie prefix for session cookies		
9.15	Use a cryptographically strong PRNG for generating session identifiers		
9.16	Use a sufficiently large size for session identifiers (at least 128 bits)		
9.17	Reject unknown session identifiers received from the client side		
9.18	Only transmit session identifiers via secure connections		
9.19	Enforce expiry and revocation of sessions and tokens		
10.0	**File Uploads**		
10.1	Store uploaded files outside the document root		
10.2	Never execute or evaluate uploaded files		
10.3	Enforce a conservative file size limit for uploaded files		
10.4	Accept only specific file types (image, text, etc.) for uploaded files		
11.0	**Content**		
11.1	Explicitly specify a specific content type for all served resources		
11.2	Explicitly specify a character set for all served resources		
11.3	Implement measures against content sniffing for all served resources		
12.0	**XML Processing**		
12.1	Disable XML external entity expansion		
12.2	Disable XML external DTD parsing		
12.3	Disable all extraneous or dangerous XML extensions		
12.4	Use an XML parser that does not recursively expand entities		
13.0	**Miscellaneous**		

#	Certified Secure Web Application Secure Development Checklist	Result	Ref
13.1	Implement anti-clickjacking measures		
13.2	Never redirect to unvalidated URLs		
13.3	Never open unvalidated URLs		
13.4	Restrict cross-domain access using a whitelist based approach		
13.5	Implement rate limiting for all email based functionality		
13.6	Implement rate limiting for all resource-intensive functionality		
13.7	Prevent unintended denial of service when implementing rate limiting		
13.8	Prevent HTTP request smuggling		
13.9	Check for and mitigate application- or setup-specific problems		