EUROPEAN
COMMISSION

Brussels, 26.10.2018
C(2018) 7283 final

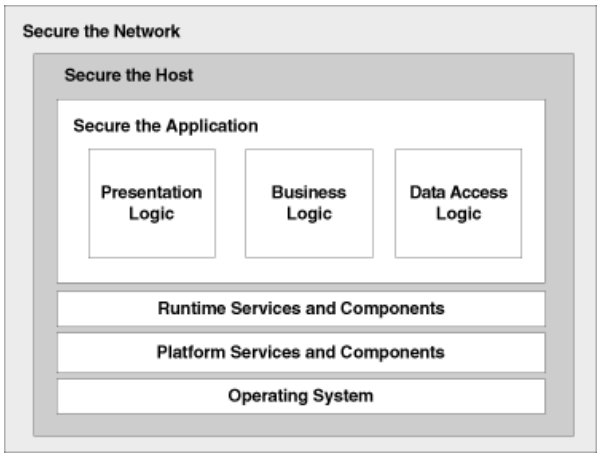
IT Security Standard

Web Application Security Standard

TABLE OF CONTENTS

1. INTRODUCTION

Building secure web applications, a holistic approach to application security is required and security must be applied at all three layers: application, host and network. This approach is shown below



This security standard covers the design, development and deployment of a web application. It establishes principles that developers should respect over the different phases of the web application lifecycle. Certain requirements have been suggested as recommended, and compliance with them may be evaluated on a case-by-case basis, based on a risk assessment. Mandatory requirements, however, apply to any web application, regardless of its exposure, nature, functionalities or published information.

2. PURPOSE

The purpose of this document is to define the minimum set of security requirements to comply with in order to design, build and configure secure web applications, to ensure resilience over attacks and to limit the extent of damage should an attack occur.
This document sets the baseline of the security requirements for any new or existing web application within the European Commission.

3. SCOPE

The following requirements apply to all IT Systems in the Commission, either hosted on-premises or in the cloud, providing the minimum set of requirements with which the applications shall comply. Mandatory requirements are subject to compliance verification. Some requirements in this standard are recommended, hence not subject to compliance verification, which means they shall be used as a guidance and they are based on market best practices. These recommended requirements shall be under consideration to become mandatory in the future versions of this standard. It shall be noted that any updated will be based on customer feedback, further market best practice analysis and compliance analysis.

4. EXCEPTIONS

Mandatory requirements within this document need a documented exception request in the security plan by the System Owner. For recommended requirements an exception request is not needed. Exceptions to mandatory requirements must be handled according to the article 8.3.d.iv of the Implementing Rules of the Commission Decision 2017/46.

5. DEFINITIONS

TERM	DESCRIPTION
Anomaly Detection	Technique used to identify unpredicted or expected behaviour.
Data Anonymization	Data anonymization is a type of information sanitization whose intent is privacy protection. It is the process of either encrypting or removing personally identifiable information from data sets, so that the people whom the data describe remain anonymous.
Content Management System	Software program or a set of related programs that are used to manage the structure and the design of digital content.
Cross Site Request Forgery	Attack where a malicious third party website visited by the victim executes unwanted actions on the web application where the victim is currently authenticated to.
HTTP Strict Transport Security	Opt-in security enhancement that is specified by a web application using a special response header. Once a supported browser receives this header that browser will prevent any communications from being sent over HTTP to the specified domain and will instead send all communications over HTTPS.
Perfect Forward Secrecy	Property of secure communication protocols in which compromises of long-term keys do not compromise past session keys.
Pseudonymization	Pseudonymization is a data management and de-identification procedure by which personally identifiable information fields within a data record are replaced by one or more artificial identifiers, or pseudonyms.
Sensitive non-classified Information	Information defined in the Commission Decision (EU, Euratom) 2015/443 that the Commission must protect because of legal obligations laid down in the Treaties or in acts adopted in implementation thereof.
Strong Authentication	Authentication based on the use of two or more independent authenticators as defined in the Password Technical Specification and Access Control and Authentication Standard.
Web Application Framework	Software framework designed to support the development of web applications including web services, web resources, and web APIs. Web application frameworks provide a standard way to build and deploy web applications. Web frameworks aim to automate the overhead associated with common activities performed in web development and provide a certain level of security by embedding and enforcing security mechanisms.
Web-Facing	Any system that is directly accessible from the Internet.
Session ID	Unique identifier that a website assigns to a specific user for some predetermined duration of time, or session, to keep track of visitor activity.

ken.ducatel@ec.europa.eu

6. GENERAL REQUIREMENTS

Adequate security practices adoption in the design, development and implementation of a web application are necessary to ensure it is secure by design.

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
1.	A risk assessment **shall** be conducted by the System Owner, in accordance with an IT Risk Management Methodology approved by the European Commission and its outcome (and any other relevant constituent) shall be part of the relevant Security Plan.	M
2.	A new web application **shall** use a robust and recognized development framework.	M
3.	The development team **should** use a development framework recommended by DIGIT.	R
4.	A maintained version of the framework **shall** be used; by default, this should be the latest stable version of the framework.	M
5.	Unsupported or deprecated client-side technologies **shall not** be used.	M
6.	All client side technologies not natively supported by browsers **shall** not be used.	M

7. AUTHENTICATION

Authentication mechanisms prevent access by unauthorised users as well as leakage of information that could enable an unauthorised user to gain access.

7.1. Use of mutualized authentication services

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
7.	Web applications **should** use EU Login to enforce authentication with a strength that satisfies the risk analysis and/or the required level of security.	R
8.	If EU Login support is not implemented the web applications **shall** implement authentication mechanisms, with a strength that satisfies the risk analysis and/or the required level of security, as well as authorization checks that meet requirements established by the Password Technical Specification and the Access Control and Authentication Standard.	M

7.2. Implement authentication controls to protect sensitive non-classified information.

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
9.	Web applications handling sensitive non-classified information **shall** use EU Login[1] to enforce authentication with a strength that satisfies the risk analysis and/or the required level of security.	M
10.	Administration pages of an application containing sensitive non-classified information **shall** have stricter security requirements which may justify the need for a re-authentication and/or for a stronger authentication.	M
11.	Sensitive non-classified information, such as incorrect passwords, in case of failed login attempts, **shall** not be stored.	M

7.3. Protect privileged functionalities

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
12.	Administrator functionality and/or interfaces **shall not** be accessible to unauthorized users.	M
13.	Administration functionality and/or interfaces **shall not** be accessible from the Internet except for specific source IP addresses.	M
14.	Administration interfaces accessible from the Internet **should** use a random path instead of the default path (e.g. /admin_7e63b9c132).	R
15.	Default accounts **shall** be disabled.	M
16.	Administrators **shall** have a separate user and administrator role. Privileged functionalities **shall** only be available when the administrator is active.	M

[1] DGs with their own Single-Sign-On (SSO) authentication mechanism can use their solution as an alternative to EU Login.

7.4. Sanitize authentication error messages

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
17.	Information provided in authentication error messages **shall not** reveal technical details about the underlying security mechanisms.	M
18.	Information provided in authentication error messages **shall not** provide information about the existence of an account on the application.	M

7.5. Securely store and transmit credentials

(M=Mandatory, R=Recommended)

S/N	Requirement	User Centric
19.	Security secrets used by the web application itself (e.g. passwords, API keys, encryption keys, etc.) **shall not** be included in the source code or online repositories so that if the source code is leaked these secrets do not become public.	M
20.	User passwords used to authenticate **shall** be stored in a secure way, being at least hashed using a strong cryptographic hash function[2] and the passwords **shall** be salted per user before hashing.	M
21.	Authentication controls **shall** be checked on the server side.	M
22.	Credentials used to authenticate **shall** be sent over HTTPS.	M

7.6. Protect renewal of credentials

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
23.	Modification of a user's credential **shall** require the user to enter the old password, new password and a confirmation of the new password[3]; The password change functionality **shall** be accessible only by a secured logged-in session.	M
24.	Account/password recovery controls **should** make use of a time-based one-time authentication token. The validity period shall be based on the business needs with a max of 24h validity.	R

[2] For more details on Standard on Cryptography and Public Key Infrastructure

[3] For password requirements see on Password Technical Specification.

ken.ducatel@ec.europa.eu

25.	Initial credentials for the users of the application **shall** be unique and the application **shall** enforce the users to modify their initial credentials.	M
26.	Recovery tokens and initial credentials **should** be delivered over an encrypted side-channel to the affected users.	R

8. SESSION MANAGEMENT

Session IDs are protected both in storage and in transport to prevent account takeover through session ID exposure or predictability.

8.1. Session IDs

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
27.	Session IDs **shall** be unique and contain at least 128 bits of entropy so that brute-forcing or guessing the session ID of an authenticated user is not feasible.	M
28.	Session IDs contents (or value) **shall not** contain meaningful data like username or e-mail address and other user's personal information.	M
29.	Sessions IDs **shall** never be displayed in URLs, logs, and error messages.	M
30.	Session IDs stored in cookies **shall** have the "Secure" flag set to prevent the browser from sending the cookie over an unsecured channel.	M
31.	Session IDs stored in cookies **should** have the domain attribute blank to avoid that the cookie is also sent to subdomains.	R
32.	Session IDs stored in cookies **shall** have the path attribute set to the web directory path of the application that needs to receive the cookie rather than the root directory.	M
33.	Session IDs stored in cookies **shall** have the "HttpOnly" flag set, thus making it impossible for an attacker to access this cookie by client-side APIs such as JavaScript.	M
34.	Web application **shall** only accept cookies as a means for session ID exchange management, and shall ensure that no other exchange mechanism is possible.	M

8.2. Session lifecycle

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
35.	Sessions **shall** be automatically terminated on the server when a user is	M

	no longer active for a specified amount of time.	
36.	Session idle timeouts **should** be no longer than 5 minutes for applications handling sensitive non-classified information and 30 minutes for the rest of the applications depending on a risk assessment.	R
37.	Sessions **shall** be automatically terminated when the user logs out of the web application.	M
38.	Sessions **shall** be automatically terminated on the client when the user closes the browser, by creating cookies without an expiration date.	M
39.	Successful authentications **shall** generate a new session and therefore a new session ID.	M
40.	Web applications **shall** use the session management features implementation from the selected web development framework, rather than building such mechanism from scratch.	M

9. ACCESS CONTROL

Access control is used to allow access to those resources which are appropriate to that entity's identity and to prevent the intentional or unintentional execution of unauthorized actions in the application.

9.1. Enforce access control

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
41.	Access control checks performed at client-side **shall** also be checked at server-side.	M
42.	Directory listing and browsing **shall** be disabled.	M
43.	File or directory metadata in the web applications **shall** be sanitized.	M
44.	The web application **shall** make use of anti-Cross Site Request Forgery (CSRF) tokens in order to prevent the user from executing unwanted actions on the web application they are currently authenticated to.	M

9.2. Minimize privileges

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
45.	All user accounts and resources (such as processes) **shall** only have the lowest level of rights needed to perform their tasks.	M
46.	Unapproved self-registered accounts **shall not** be allowed to post any public contents.	M

ken.ducatel@ec.europa.eu

47.	Accounts supporting automated application functionalities **should** prevent interactive login, making it impossible to use these accounts for non-automated operations.	R

10. INPUT VALIDATION & OUTPUT SANITIZATION

All input and output is handled securely to prevent any adverse effects when processed or rendered by the application.

10.1. Input validation checks

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
48.	Input validation controls **shall** be implemented for every web application which allows a user to input data.	M
49.	Input validation **shall** be performed at server-side.	M
50.	Input validation **should** also be performed at client-side in addition to server-side checks.	R

10.2. Prevent injection attacks

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
51.	The web application and all its backend services **shall** make use of a safe API that allows the use of a parameterized interface.	M
52.	If a parameterized API is not available, special characters **shall** be escaped using the specific escape syntax for that interpreter.	M

10.3. Render content safely

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
53.	Web applications **shall** use development frameworks mechanisms for rendering content safely and escaping reserved characters.	M

11. COMMUNICATIONS

All communication between the client and the web server is protected to prevent modification or disclosure of sensitive non-classified information.

11.1. Secure traffic

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
54.	Communication between applications and underlying services[4] **should** be encrypted.	R
55.	Communication of sensitive non-classified information between applications and underlying services **shall** be encrypted.	M
56.	Web application **shall** make use of encrypted traffic for the entire web session on every web page including content from third party domains, in compliance with the SSL/TLS Technical Standard.	M
57.	All sensitive non-classified information **shall** be kept out of the URL.	M
58.	The HTTP Strict Transport Security (HSTS) header **shall** be set on all requests and for all subdomains.	M
59.	The HSTS header **should** be pre-loaded into browsers with a long max-age flag (ideally one year).	R

11.2. Certificates

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
60.	Strong, non-deprecated algorithms, ciphers and protocols **shall** be used throughout the whole certificate hierarchy.	M
61.	Web facing applications **shall** use certificates delivered by a trusted Certificate Authority[5].	M
62.	Perfect Forward Secrecy **shall** be supported.	M

11.3. HTTP Secure Configuration

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
63.	The web application **shall** only accept the standard HTTP request methods (e.g. GET, POST). Other protocols or methods shall be	M

[4] underlying services are any type of IT services (e.g. web services, database services) used by the application

[5] HR.DS provides guidance on how to use Trusted Certificate Authorities for the applications. Internal applications may use CommisSign-2 SSL certificates.

	blocked.	
64.	All HTTP responses **shall** contain a Content-Type header with the correct MIME type.	M
65.	HTTP headers **shall not** disclose version or any other internal information about the underlying system or technology.	M
66.	Content Security Policy (CSP) header **shall** be used and strictly configured to prevent injections such as XSS or HTML injection.	M
67.	The X-XSS-Protection header **shall** be used in order to protect against reflected XSS attacks.	M
68.	The X-Frame-Options header **shall** be used in order to protect against clickjacking attacks	M

12. DATA PROTECTION

Data is handled securely by the application to prevent leakage of data, including of sensitive non-classified information.

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
69.	Forms handling sensitive non classified information **shall not** make use of autocomplete features for those fields of information and **shall** disable client-side caching.	M
70.	When filling out forms, sensitive non-classified information **should** be masked while typed. (e.g. *****)	R
71.	Data stored in client-side cache **shall not** contain any sensitive non-classified information.	M
72.	Sensitive non-classified information **shall** only be sent over HTTPS.	M
73.	All sensitive non-classified information maintained in memory **should** be overwritten with zeros or random data once it is no longer necessary to be kept in memory	R

13. SECURE HANDLING OF RESOURCES

Input and content provided by the user is handled safely by the application to prevent application failure, data leakage or abuse of its functionalities.

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
74.	Data (e.g. files, variables) submitted by a user to the web application **shall not** be used as input for operating system commands.	M

75.	Files uploaded by users **shall not** be stored under the web directory (webroot) of the webserver.	M
76.	Uploaded files **shall** be scanned by antivirus software.	M
77.	The web application **shall not** execute files uploaded by the user.	M
78.	All URL redirects **shall** be validated at the input time.	M
79.	If URL redirects based on a pre-defined list (e.g. whitelist) of allowed domain is not possible, a warning **shall** be shown firstly to the users notifying them that they are going off of the site, and a link shall be clicked by them for confirmation.	M

14. ERROR & EXCEPTION HANDLING

Error messages must not contain information that can be used to compromise the application or reveal sensitive non-classified information.

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
80.	Information provided in error messages **shall** be generic: it shall not reveal technical details about the underlying security or any other system internal mechanisms, except for a unique identifier which can be used in troubleshooting.	M

15. LOGGING

Logging is the act of keeping records of events that occur in an operating system or software applications. Logging enables anomaly detection as well as investigation and response to incidents.

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
81.	The web application **shall** log all necessary information (e.g. access control decisions) needed to begin a thorough investigation.	M
82.	Authentication attempts, both successful and failed, **shall** be logged.	M
83.	Access to sensitive non-classified information **shall** be logged.	M
84.	Changes to web application configuration, including changes to privileges assigned to users and security parametrization, **shall** be logged.	M
85.	Logs **shall not** include sensitive non-classified information.	M
86.	Logs **shall not** be accessible to unauthorized users.	M

ken.ducatel@ec.europa.eu

87.	Controls **shall** be in place to prevent that logs are overwritten or tampered with.	M

16. Specific Considerations for Mobile Applications

Appropriate mechanisms are implemented to mitigate the impact on stolen or compromised devices.

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
88.	Login credentials **shall not** be stored onto the device.	M
89.	Sensitive non-classified information that must be stored onto the device **shall** be encrypted.	M
90.	The mobile application **shall** only require the least privileges necessary to function properly.	M
91.	The mobile application **shall** check authentication when exposing non-public functions to other local mobile applications on the same device.	M
92.	Security controls **shall** be implemented both in the mobile application and the backend with which it interacts.	M

17. Host & Network Security

Host and network supporting the application have to be secure to prevent host-level attacks and network-level attacks resulting in a compromise of the application through the host. As far as the network is concerned, this section describes a high-level approach for the network security. Network Security Standard provides more in-depth details.

17.1. Host Security on Premise or in Private Cloud

All hosts shall be configured based on hardened security baselines reflecting best security configuration practices.

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
93.	Communication between components (e.g. web application server - database server) **shall** require an authenticated connection, using an account with the least privileges necessary to operate.	M
94.	The application and all underlying components and middleware **shall** run with minimal privileges and **shall** not use (default) administration accounts shipped with systems.	M
95.	All hosts and software supporting the web application **shall** be updated timely after publication of security patches.	M

96.	All hosts and software supporting the web application **should** be updated timely after publication of functional patches.	R
97.	All management platforms that allow interaction with the hosts and software supporting the web application – including cloud consoles or similar management platforms – **shall** adhere to at least the same set of requirements as the actual web application and its supporting software.	M
98.	Access to such management platforms **shall** be considered as privileged access, and the accounts for this **shall** adhere to applicable requirements of the Password Technical Specification and the Access Control and Authentication Standard.	M
99.	Cloud Service Providers **shall** be able to support all the security requirements applicable to the web application.	M

17.2. Network Security

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
100.	External facing web applications **shall** be deployed in demilitarized zone (DMZ) to permit only limited connectivity to specific hosts in the internal network, reducing the attack surface.	M
101.	External facing web application containing sensitive non-classified information **shall** be protected by a Web Application Firewall.	M

18. DEPLOYMENT

Proper security testing prior to implementation in production prevents the deployment of vulnerable applications.

Development releases should be digitally signed by the development team to ensure there is no modification after the final acceptance and waiting for deploying in the production environment.

18.1. Debug mode

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
102.	Debug mode **shall** be disabled in production.	M

18.2. Acceptance and Security Testing

(M=Mandatory, R=Recommended)

S/N	Requirement	Implementation
103.	For web applications handling, sensitive non-classified information, pseudonymization and/or anonymization of data **shall** be assured in the test environments.	M
104.	For web applications handling sensitive non-classified information, pseudonymization and/or anonymization of data **should** be assured in the acceptance environments.	R
105.	In accordance with the priority levels[6] as described in IT Vulnerability Management Security Standard every web application **shall** undergo regularly a web application vulnerability assessment to identify common vulnerabilities.	M
106.	Important vulnerabilities based on a vulnerability assessment **shall** be fixed prior to going into production.	M
107.	If the web application handles sensitive non-classified information, a penetration test **should** be executed to find vulnerabilities in the web application.	R
108.	Test environments **shall not** be publicly accessible.	M

19. FINAL PROVISION

This standard shall enter into force on the day following the date of the adoption.

[6] For more information on priority levels of IT systems and their IT assets refer to the IT Asset Management Security Standard.

REFERENCES

[1] SSL/TLS Technical Standard, "European Commission INFOSEC Technical Standards, SSL/TLS" [Online]. Available: https://myintracomm.ec.europa.eu/corp/digit/itsecurity/Documents/DIGITCINFOSECTechnicalStandards.pdf

ken.ducatel@ec.europa.eu