Draft ETSI EN 304 617

Cybersecurity (CYBER); CRA; Cybersecurity requirements for Browsers

Reference
< WI-0000 >

Keywords
< CRA >

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from the ETSI Search & Browse Standards application.

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on ETSI deliver repository.

Users should be aware that the present document may be revised or have its status changed, this information is available in the Milestones listing.

If you find errors in the present document, please send your comments to the relevant service listed under Committee Support Staff.

If you find a security vulnerability in the present document, please report it through our Coordinated Vulnerability Disclosure (CVD) program.

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI IPR online database.
Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organisations associated with those trademarks.
The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organisations associated with those trademarks.
DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organisational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This draft Harmonised European Standard (EN) has been produced by ETSI Technical Committee Cyber Security (CYBER), and is now submitted for the combined Public Enquiry and Vote phase of the ETSI Standardisation Request deliverable Approval Procedure (SRdAP).
The present document has been prepared under the Commission's standardisation request C(2025)618 [i.3] to provide one voluntary means of conforming to the requirements of Regulation (EU) 2024/2847 [i.1] of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828, known as the Cyber Resilience Act (CRA).
Once the present document is cited in the Official Journal of the European Union under that Regulation, compliance with the normative clauses of the present document given in table A.1 confers, within the limits of the scope of the present document, a presumption of conformity with the corresponding requirements of that Regulation and associated EFTA regulations.
Transposition table
The Harmonised Standard shall have appropriate transposition periods specified. A Harmonised Standard confers presumption of conformity when it has been published in the Official Journal of the European Union (OJEU) and transposed by a member state.

Proposed national transposition dates	
Date of latest announcement of this EN (doa):	3 months after ETSI publication
Date of latest publication of new National Standard or endorsement of this EN (dop/e):	6 months after doa
Date of withdrawal of any conflicting National Standard (dow):	18 months after doa

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot** are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).
"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.
=

Introduction

The present document provides the technical cybersecurity requirements for the products in scope, following a risk-based approach in support of the Cyber Resilience Act (CRA) [i.1]. The technical cybersecurity requirements are thereby proportionate to the intended purpose, reasonably foreseeable use, deployment context, and threat exposure of the products.

Clause 4 does not contain technical requirements; it describes the product architecture and intended purpose in their context.

Clause 4 also defines Use Cases (UCs) that represent the main deployment scenarios reflecting the intended purpose and reasonably foreseeable use of the product, which serve as the basis for identifying relevant cybersecurity risks.

Clause 5 specifies technical cybersecurity requirements for the product to mitigate the identified risks, including their applicability conditions.

Clause 6 specifies the assessment criteria and compliance verification procedures with the requirements of Clause 5.

Annex A maps the technical requirements of the present document with the essential requirements of the CRA [i.1] regulation.

Annex B informs about the methodology used to assess the security risks of the products in their context. Where a product does not clearly correspond to one of the defined Use Cases of Clause 4, the risk assessment methodology of Annex B may be used to determine the applicable cybersecurity requirements.

Annex C refers to other standards that can potentially have an impact on this document.

Annex G provides further information on guidance for the application of the present document.

Annex K supports the definition of the cryptographic requirements and assessment criteria used by the present document.

1 Scope

The present document specifies technical requirements and corresponding assessment criteria for web browsers related to cybersecurity. The products with digital elements in scope, thereafter "the products" are specified within the "technical description" of the "category of product" number "2" by the Commission Implementing Regulation (EU) 2025/2392 of 28 November 2025 on the technical description of the categories of important and critical products with digital elements pursuant to Regulation (EU) 2024/2847 of the European Parliament and of the Council. [i.2] as:

“Software products with digital elements that enable end users to access, render, and interact with web content and services hosted on servers that are connected to networks such as the Internet. They typically include a browser engine for interpreting and displaying content written in markup language (e.g. HTML), support for web protocols (e.g. HTTP, HTTPS), the ability to execute scripts and manage user inputs as well as storage of temporary or persistent data from websites (cookies).

This category includes but is not limited to standalone applications that fulfil the functions of browsers, embedded browsers intended for integration into another system or application as well as browsers with AI agent integration.”

The products are only covered within the product context described in clause 4. The present document specifies technical characteristics and methods of assessment for:

- **Standalone web browsers**: standalone applications that fulfill the functions of web browsers
- **Embedded web browsers**: embedded browsers intended for integration into another system or application

The present document covers those products to demonstrate compliance with essential cybersecurity requirements in the Regulation (EU) 2024/2847 [i.1] Annex I Part I under the conditions identified in annex A.

Browsers intended for use in the industrial OT (Operational Technology) domain are excluded from the scope of the present document, see prEN 50770 series [i.x].

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies. Referenced documents which are not found to be publicly available in the expected location might be found in the ETSI docbox.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ENISA Report 1747792503 (version 2 - April 2025): "European Cybersecurity Certification Group Sub-group on Cryptography Agreed Cryptographic Mechanisms.

[2] prEN 40000-1-3: "Cybersecurity requirements for products with digital elements - Vulnerability Handling" [Version and date to be added upon its publication by CEN CENELEC]

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

[i.1] Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act).

[i.2] Commission Implementing Regulation (EU) 2025/2392 of 28 November 2025 on the technical description of the categories of important and critical products with digital elements pursuant to Regulation (EU) 2024/2847 of the European Parliament and of the Council.

[i.3] Standardisation request M/606 - C(2025)618: "Commission Implementing decision of 3.2.2025 on a standardisation request to the European Committee for Standardisation (CEN), the European Committee for Electrotechnical Standardisation (Cenelec) and the European Telecommunications Standards Institute (ETSI) as regards products with digital elements in support of Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020and Directive (EU) 2020/1828 (Cyber Resilience Act)".

[i.4] prEN 40000-1-1: "Cybersecurity requirements for products with digital elements - Vocabulary" (produced by CEN CENELEC)

NOTE: Version and date to be added upon its publication by CEN CENELEC.

[i.5] prEN 40000-1-2: “Cybersecurity requirements for products with digital elements – Principles for cyber resilience” (produced by CEN CENELEC)

NOTE: Version and date to be added upon its publication by CEN CENELEC.

[i.6] Web Platform Design Principles, World Wide Web Consortium Group Note, 27 January 2026

[i.7] RFC 5246: TLS 1.2, IETF, August 2008

[i.8] RFC 8446: TLS 1.3, IETF, August 2018

[i.9] RFC 6979: HSTS, IETF, November 2012

[i.10] Mixed Content (W3C Candidate Recommendation Draft), W3C, 23 February 2023

[i.11] Threat Model for the Web (W3C Group Draft Note), W3C, 11 May 2026

[i.12] HTML (WHATWG Living Standard), WHATWG, 11 May 2026

[i.13] Fetch (WHATWG Living Standard), WHATWG, 08 May 2026

[i.14] Content Security Policy Level 2 (W3C Recommendation), W3C, 15 December 2016

[i.15] Secure Contexts (W3C Candidate Recommendation Draft), W3C, 10 November 2023

[i.16] Subresource Integrity (W3C Working Draft), W3C, 20 March 2026

[i.17] RFC 8996: Deprecating TLS 1.0 and TLS 1.1, IETF, March 2021

[i.18]]RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, IETF, May 2008

[i.19] RFC 6962: Certificate Transparency, IETF, June 2013

[i.20] Threat Modeling Guide (W3C Group Draft Note), W3C, 4 June 2026

[i.21] OWASP Risk Rating Methodology (OWASP Community Pages), OWASP, 2 December 2025

[i.22] Secure Contexts, W3C, 10 November 2023

[i.23] TLS_CHACHA20_POLY1305_SHA256

[i.24] TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256

[i.25] TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA

[i.26] TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256

[i.27] TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA

[i.28] TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

[i.29] TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA

[i.30] TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256

[i.31] TLS_RSA_WITH_3DES_EDE_CBC_SHA

[i.32] TLS_RSA_WITH_AES_128_CBC_SHA

[i.33] https://www.rfc-editor.org/info/rfc3268/TLS_RSA_WITH_AES_256_CBC_SHA

[i.34] ecdsa_secp256r1_sha256

[i.35] rsa_pkcs1_sha512

[i.36] rsa_pss_rsae_sha384

[i.37] rsa_pss_rsae_sha512

[i.38] AES-GCM-SIV

[i.39] https://www.rfc-editor.org/info/rfc3268/AES-OCB

[i.40] Argon2d

[i.41] Argon2i

[i.42] Argon2id

[i.43] Blake2b

[i.44] Blake2s

[i.45] ChaCha20

[i.46] Ed25519

[i.47] SHAKE128

[i.48] SHAKE256

[i.49] TurboSHAKE128

[i.50] TurboSHAKE256

[i.51] X-Wing

[i.52] X25519

[i.53] X25519MLKEM768

[i.54] X448

[i.55] XChaCha20

[i.56] XChaCha20-Poly1305

[i.57] XSalsa20

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in Regulation (EU) 2024/2847 [i.1], prEN 40000-1-1 [i.4] and the following apply:

Browser profiles: user managed profiles that may be used for separating principals or use-cases

Effective Top-Level Domain: a domain suffix, under which independent tenants may register subdomains

Origin: the compound of the scheme, host and port of a web resource. When a web resource defined as valid in the user documentation cannot be formed into a meaningful combination of host, scheme and port, the product constructs a synthetic origin uniquely identifying the resource.

Powerful Web Platform Feature: A web platform API the web browser provides to a website that directly exposes cameras, microphones, location information, clipboard contents, screen contents, system notifications, external peripherals, or payment methods.

Private mode: a context within the browser interface offered to users to provide enhanced privacy protections

Public suffix list: list of all eTLDs

Registrable domain: subdomain exactly one level below an eTLD, such that the domain and all further sub-domains are registered to a single tenant

Rendering processes: processes or other execution contexts, that handle the execution or rendering of website assets

Same-origin policy: a fundamental browser security model in browser products used to determine whether assets and state should be shared or not between web page execution contexts.

Sandboxing: isolation mechanism that confines a product component to a restricted execution environment, which limits its access to operating system resources, to other components, and to data, to the level for that component to perform its tasks.

Secure Context: a web page that meets minimum standards of authentication and confidentiality by delivering content over an encrypted and authenticated channel like HTTPS, as fully defined by Secure Contexts [i.22]

Site: logical security boundary defined by scheme and registrable domain

Storage data: cookies, Web Storage, IndexedDB, and any other site-stored data automatically made available to sites

Third-party cookies: cookies keyed to a site other than the top-level site

Trusted root store: a collection containing a set of "root" certificates used as trust anchors

3.2 Symbols

VOID

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ACM ENISA Agreed Cryptographic Methods
APT Advanced persistent threats
C2 Command and Control
CC Common Criteria
CPU Central Processing Unit
CRA Cyber Resilience Act
CVE Common Vulnerabilities and Exposures
DB Database
eTLD Effective Top-level Domain
EDR Endpoint Detection and Response
DHCP Dynamic Host Configuration Protocol
DOM
GDPR General Data Protection Regulation
GUI Graphical User Interface
HSM Hardware Security Module
HSTS HTTP Strict Transport Security

HTTP Hypertext Transport Protocol
HTTPS Hypertext Transport Protocol Secure
IAM Identity and Access Management
ICMP Internet Control Message Protocol
ICT Information and Communication Technology
IoT Internet of Things
ISO International Organization for Standardization
MITM Man-In-The-Middle
MP Main Product
MTTD Mean Time to Detect
MTTR Mean Time to Respond
NIST National Institute of Standards and Technology
OS Operating System
OT Operational Technology
PC Personal Computer
PIA Privacy Impact Assessments
PII Personally Identifiable Information
PoLP Principle of Least Privilege
RDPS Remote Data Processing Solution
RTO Recovery Time Objective
SBOM Software Bill of Materials
SCC Security Category Classes
SDK Software Development Kit
SIEM Security Information and Event Management
SIF Social Interactive Function
SOAR Security Orchestration Automation and Response
SOP Same-origin policy TLS Transport Layer Security
TUF The Update Framework
UC Use Case
UI User Interface
XDR Extended Detection and Response

4 Product context

4.1 Product Functions

The intended purpose of a web browser is to access, retrieve, and render web-based resources on behalf of an end user and display them via a Graphical User Interface. This includes everything from reading the news and online encyclopedias to accessing critical systems related to finance, government, healthcare, etc.

Browsers are general-purpose software: When presented with a web browser, including an embedded one, user expectations are generally that they can trust it as their user agent to mediate the interaction between them and the web.

The breadth of reasonably foreseeable uses for all web browsers implies that there are many risks in common across use cases. To access websites and make them available to users, web browsers need components to handle the following:

- Retrieve arbitrary web content
- Parse HTML and build DOM

- Style CSS and compute resource layout
- Execute JavaScript and WebAssembly
- Render images, media, and other embedded content
- Manage user-directed navigation and sessions
- Establish client-server connections
- Store and manage data
- Receive and process user input
- Deliver and apply software updates

4.2 Product Architecture

This section highlights the components, functionalities, and trust boundaries of web browsers that have significant security implications. While the specific architecture of web browsers can vary based on factors such as vendor-specific implementation and deployment environment, this section presents a generic high-level web browser architecture diagram derived from the W3C Threat Model for the Web Group Draft Note [i.10].

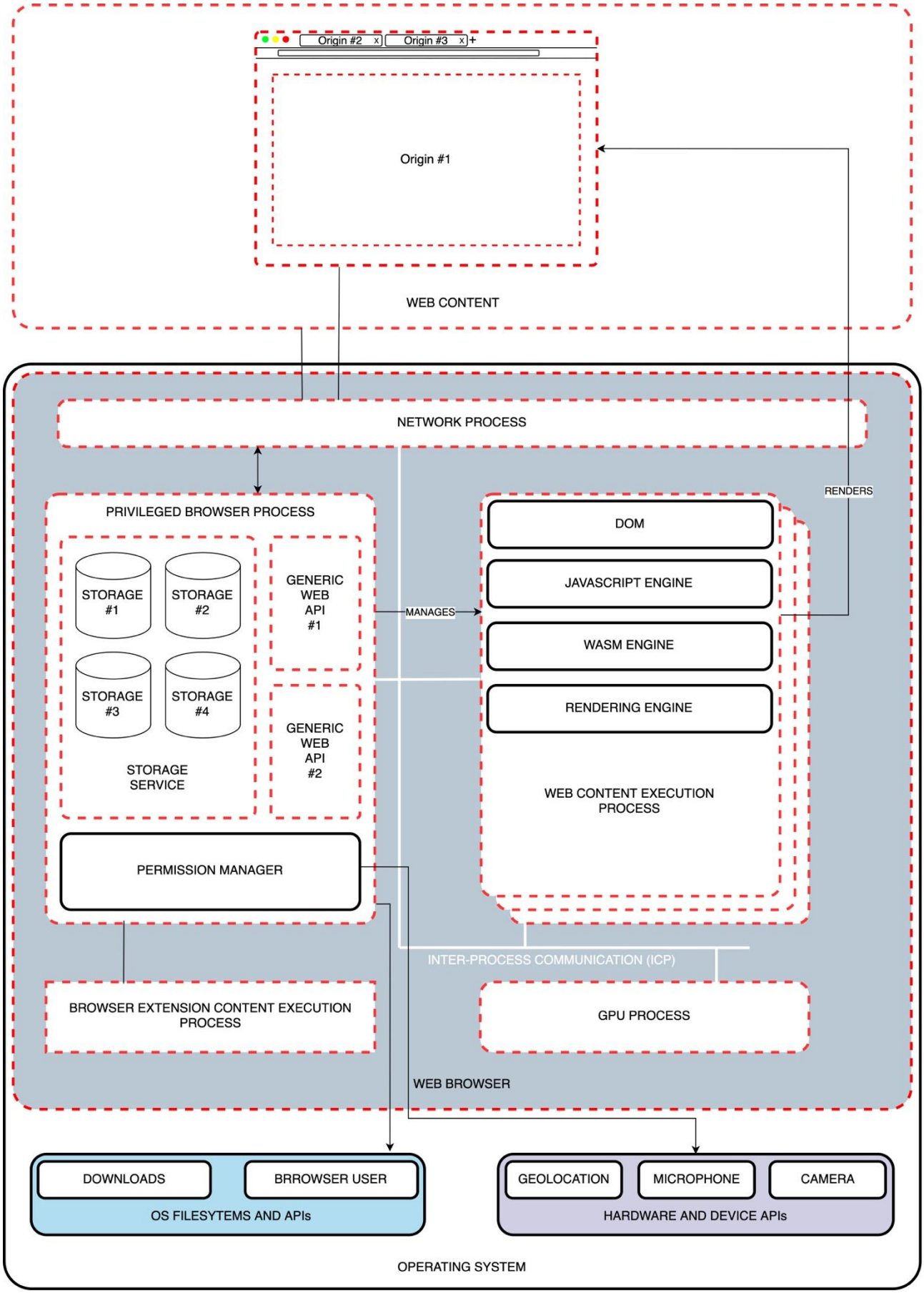

Diagram 4.2.1: Reference Web Browser Architecture

Web browsers are often characterized by a multi-process structure where different processes handle different functions.

- **Privileged web browser process.** The Privileged web browser process is the generally initial process launched by the OS and is typically responsible for managing subsequent processes, sessions, navigation, User Interface, and policies. This high-privilege process also generally serves as a central broker for permissions and other OS APIs, as well as access to local resources and OS filesystem (It is worth noting that, in some implementations, these functions may not be carried out by the privileged web browser process alone but might instead be assigned to multiple processes). With direct OS access and elevated privileges, the Privileged web browser process enforces the trust boundary between the OS and the web browser itself.
- **web-content execution.** Web resources are parsed and executed inside dedicated child processes. These processes handle untrusted web resources and should therefore be modeled under the assumption that the input they receive and interpret cannot be trusted. For this reason, each website is typically assigned a dedicated sandboxed process that is isolated from all others. The resulting trust boundary separates web-executed content from the privileged web browser process.
- **Storage services.** Web browser typically also isolate and separate storage by origin. This includes cache, site data, session data, cookies, user credentials, and any other type of persistent or session-specific storage. This separation enforces the trust boundary between each origin-specific storage, preventing any origin from accessing another's stored data.
- **I/O.** Modern web browsers typically also isolate helper functionalities such as, networking stack, and other utilities into separate isolated processes to improve security and reliability.

Common trust boundaries include:

- The origin/origin trust boundary separates each web origin from one another.
- The browser/OS trust boundary separates the general purpose web browser application from the underlying operating system, but also from remote web content.

Inside the web browser's architecture, trust boundaries separate:

- The Privileged Browser process from the unprivileged Web Content Execution Process;
- The Browser Extension Content Execution Process from the Privileged Browser Process;
- The Web Content Execution Process responsible for executing content from one origin or site from execution processes responsible for other origins or sites;
- The elements within the Web Content Execution Process from one another. These include Document Object Model, JavaScript Engine, Rendering Engine, and WebAssembly Engine;
- Helper services and utility functionality processes such as the Network Process and the GPU Process from other processes within the web browser architecture;
- The site data Storage Service from other elements within the browser architecture.

The specifics of a web browser architecture may differ between use cases and deployment contexts, thus changing the final threat model of the product. This section highlights some key architectural additions and differences for two common web browser use cases: enterprise and embedded browsers.

Enterprise Browsers Architecture

Enterprise browsers are a specific implementation of general purpose web browsers deployed in enterprise environments and modified to fit organisational objectives and functions. These objectives and functions typically include providing centralised security policy enforcement, extended logging and monitoring, data loss prevention controls, and access management.

For the purposes of this standard, the enterprise browser use case shares the same high-level architecture and trust boundaries as the general purpose web browser (See General Purpose Browser Architecture Diagram), but with additional use-case specific components:

- An enterprise policy enforcement engine that checks some or all web browser actions against a predefined enterprise security policy;
- A centralized management interface that handles policy configuration, software updates, and extension deployments;
- A Data Loss Prevention (DLP) engine that monitors user activity such as downloads, uploads, clipboard operations, and printing against enterprise security policies;
- A logging component that records web browser activity for auditing, compliance, and forensics purposes;
- An enterprise identity and access management component that handles SSO flows and enterprise authentication tokens.

Embedded Browser Architecture

Embedded browsers are integrated as components within a host application and therefore execute and render untrusted web content within the host application context itself. Despite sharing the same trust boundaries as general purpose web browsers (See General Purpose Web Browser Architecture Diagram), embedded browsers' architecture (See Embedded Browser Architecture Diagram) and trust model differs from standalone web browsers in the following respects:

Depending on the actual technological implementation, the user's active sessions, passwords, and cookies may or may not be shared with the application.

- A JavaScript Bridge separates untrusted web content execution from the host application context, mediating embedded browser access to host application source code;
- Unlike standalone browsers, embedded browsers typically handle web content loading from the host application itself through the Native API gateway, as opposed to direct user interaction;
- Web browser extensions are typically absent in embedded browser contexts and as a result, the Browser Extension Content Execution Process is not present in the embedded browser use case architecture (See Embedded Browser Architecture Diagram).
- Permission management is handled by the host application itself, rather than presenting interactive prompts to the user via a GUI;
- Web content is rendered in the Embedded Browser View: a user interface within the host application's window hierarchy that presents the executed web content to the user.

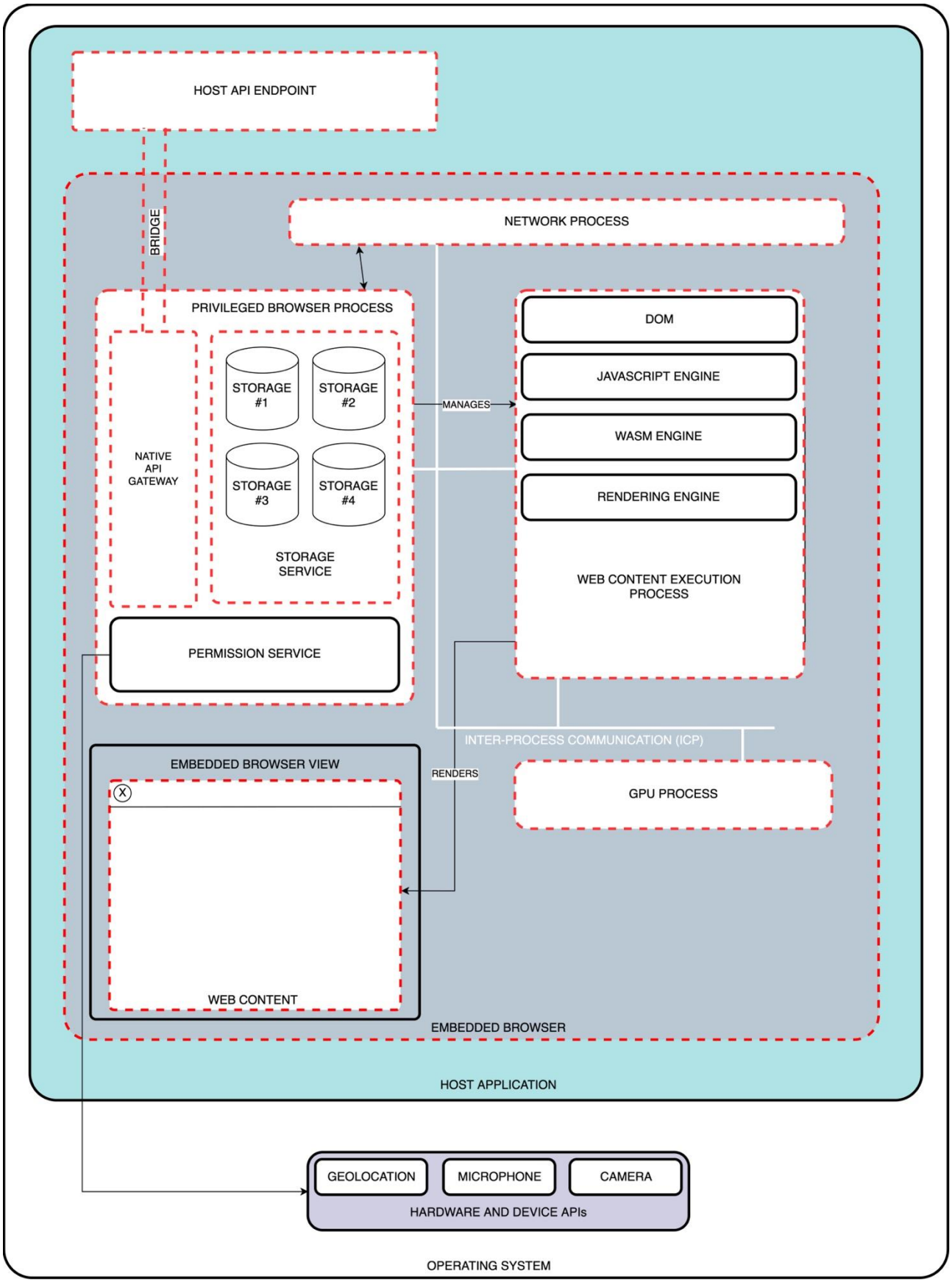

Diagram 4.2.2: Embedded Web Browser Architecture Diagram

4.3 Operational Environment

The product serves as a medium for interaction between a system and the web, retrieving web resources and executing them client-side. Because of this, the operational environment in which it operates assumes that the resources it handles are inherently untrusted.

4.3.1 General description

The product operates as a kind of user-agent that mediates between a local system and remote web resources. Its operational environment spans a range of local host devices, deployment contexts and user populations that range from unmanaged consumer devices to centrally administered enterprise contexts. Across all of these, the product assumes that the content it has retrieved is potentially hostile and that the surrounding system itself may be partially compromised, misconfigured, or operated in an insecure way.

4.3.2 Physical/Hardware environment

The product runs on general purpose computing hardware, which includes desktops, laptops, mobile devices, wearables, televisions and larger embedded devices. Unless purpose built for a specific operating system or hardware platform, the product does not necessarily presume any dedicated or trusted hardware component. If it is purpose built for a particular platform, it may make use of platform-provided features such as hardware-backed key storage, memory protection, or process isolation primitives if these are available. Physical access to the host device is outside the product's control and is generally treated as a property of the deployment context rather than a guarantee that the product provides, which is discussed in Clause 4.4.

4.3.3 Logical/Software environment

The product is executed as an application on top of an underlying operating system and that system's associated libraries. It depends upon the the host platform for services such as process and memory isolation, filesystem access, networking, and cryptographic primitives, and certain security functions may be delegated to these abstraction layers as described in Clause 4.4.1. The product coexists with other applications, extensions, and embedded host applications with different levels of trustworthiness. It enforces logical isolation between web origins and between its own components regardless of the surrounding software.

4.3.4 Connectivity aspects

The product is, by its very nature, persistently connected to untrusted networks and establishes client-server connections to arbitrary remote endpoints on behalf of the user. It is exposed to hostile networking conditions, including potential interception, manipulation and even the impersonation of web traffic. To this end, it enforces transport security and certificate validation with cryptographic methods described in Annex K. The connectivity might be direct from browser to server, or mediated by intervening infrastructure, and the product does not assume that the network path between it and a remote origin is trustworthy. Mediating elements can include:

- Forward proxies and secure web gateways for enforcing content filtering
- Reverse proxies and content delivery networks
- Carrier-side proxies and transcoding services on mobile networks
- VPNs and zero-trust network access brokers

- Parental control or enterprise management of policies that deny or allow access to specific content types or sources
- System or network-level proxy configurations
- Installed browser extensions that can tamper with the web content
- Captive portals on public or guest networks that intercept traffic and reroute in order to provide an authentication opportunity

4.4 Distribution of Security Functions

4.4.1 Security Functions

Regardless of use cases, vendor-specific implementations, and browser architecture specifics, all products with digital elements classified as web browsers that fall within the scope of the present document should provide the security functions laid out in this section. The security functions of a web browser include:

- **Same Origin Policy.** A security function whereby scripts from one origin cannot access, read or manipulate data from another origin, as described in WHATWG HTML Living Standard [i.11], and WHATWG Fetch Living Standard [i.12].
- **Controlled relaxation of SOP.** A security property whereby the Same Origin Policy can be relaxed based on server-defined policies, for example, in accordance with Cross Origin Resource Sharing as defined in WHATWG Fetch Living Standard [i.12].
- **Web content execution isolation.** A security property whereby web content from different origins is isolated into separate execution contexts.
- **Web resource content restriction.** A security control used to declare a set of content restrictions for a web resource as defined in W3C Content Security Policy Level 2 [i.13] and successor standards.
- **Isolation and sandboxing.** A security function whereby product components are sandboxed and operate with minimal OS privileges in accordance with the principle of least-privilege, and logical isolation is enforced between different components (such as sites and execution contexts).
- **Permission management of Powerful Web Platform Features.** A security function whereby access to powerful features, as defined in W3C Secure Contexts [i.14], is mediated through a permission system that requires express user permission.
- **Web resource integrity verification.** A security function whereby the integrity of retrieved web resources is verified at product level, in accordance with the Subresource Integrity feature defined in W3C Subresource Integrity [i.15].
- **Automated updates.** An security feature whereby the rollout of security updates is automated, ensuring that vulnerabilities are remediated in a timely manner.
- **TLS enforcement.** A security function that enforces support for current TLS versions as defined by IETF RFC 8446 [i.7] and successor standards, and the rejection of connections which use protocol versions identified as deprecated in IETF standards track RFCs, including RFC 8996 [i.16] and successor standards.
- **Certificate validation.** A security function that includes validation of certificate chains, including certificate path as defined in IETF RFC 5280 [i.17], and enforces Certificate Transparency as defined in IETF RFC 6962 [i.18].
- **Interaction with Browser Extensions.** Those products that provide an interface to manage browser extensions have the security functionality that formalises the browser extension's interactions with the websites that the user visits.

4.4.2 Decision tree

The aforementioned security functions inform the product-specific security requirements laid out in Clause 5.

For each security function, a web browser can:

- **Provide it itself.** In this case, the security function is provided as a native component within the web browser's codebase.
- **Require that it be provided by other parts of its context.** In this case, the browser delegates the provision of security functions to other components within its execution environment.

In the latter case, the product's technical documentation would be an appropriate place to explain deviations from the requirements in Clause 5.

4.4.3 Security functions provided outside of the product's context

Across web browsers and operating systems, different choices are made in different contexts about what is provided by the web browser and what is implemented in the surrounding operating system and system libraries, with security functions performed at those other levels. These can include for example:

- Some standalone browsers (e.g., on mobile platforms) are implemented with the use of a platform-provided embedded browser, subsuming most security functions.
- Networking, including certificate validation and management, selection of algorithms, TLS broadly, etc.
- Image decoding, text rendering and Unicode handling (no inherent security function, but a historical source of exploitable vulnerabilities).
- Integrated password manager (may also be provided via an extension).
- Handling links which may be interpreted as references to native applications, whether via OS-registered origins or custom schemes.
- Scoping permission to access camera, microphone, geolocation or other sensitive information.
- Web processes isolation may be enforced at OS kernel level.

In such cases, when functionality is implemented outside of the browser in such system libraries or operating systems, the responsibility for handling the risk is transferred to this library, following a risk assessment by the browser which integrates the OS/library that this is reasonable.

Web browsers typically reduce the privileges of their components with respect to the operating system to the level each component requires to perform its tasks. By granting each component only the access it needs, the product limits what an attacker can reach should that component be compromised. This least-privilege distribution reinforces the trust boundaries described in Clause 4.2.

For example, in a browser that separates a renderer process from the privileged browser process, the renderer process, which handles untrusted web content, is commonly denied direct access to various

operating system calls. Access to such capabilities is instead mediated by the privileged browser process acting as a broker.

The web platform maintains a surface area such that "It should be safe to visit a web page" [i.5]. Going to a website does not grant that website risky capabilities over the user's computer. By contrast, other platforms presenting a direct user interface typically involve either a platform curation step, or some risk being taken on by the user. Many mitigations in Clause 5 relate to ensuring that web browsers meet this guarantee.

4.5 Users

Almost all computer users interact with web browsers at some point. Users may include for example:

- General public
- Children and protected users
- Vulnerable adults
- Users of the web with respect to sensitive data
- Professionals in all fields of work
- Workers in critical infrastructure
- Users with accessibility needs
- Enterprise administrators
- Developers and technical users

Accessibility, health, and safety considerations apply across all user categories and are not confined to any one of them.

In consideration of the fact that there are many types of users with accessibility needs, it might be appropriate for the product to also undergo Module A type self-assessment under EN 301549 [C.4].

4.6 Use Cases

4.6.1 UC-CONS General Purpose Consumer Browser

Goal: Provide an individual consumer with general access to the whole web, including sensitive and critical applications.

- Product type: Standalone web browser.
- Function: Renders and navigates arbitrary content across the open web, including sensitive and critical applications.
- Users: Individual consumers. In general, not educated in or aware of cybersecurity issues.
- Architecture: Installed on mobile phones, desktop and laptop computers, televisions, larger embedded devices, and similar. Exposes flexible settings, including high-risk features such as developer mode.
- Operational environment: Uncontrolled. No central administration. Configuration is managed by the user.

UC-INST Enterprise or Institutional Browser

Goal: Provide managed web access within institutional or enterprise environments, including critical infrastructure.

- Product type: Standalone web browser.
- Function: Renders and navigates web content, including internal websites developed with varying attention to security or with special authentication needs.
- Users: Institutional or enterprise users, supported by central IT administration.
- Architecture: Supports an "enterprise policy" allowing central IT to configure security-related policies.
- Operational environment: Managed. Often complemented by other precautions at the network or VPN level, physical access controls, and similar.

UC-ETAB Embedded Browser Component

Goal: Allow a host application to present external web content while keeping the user oriented toward the parent application.

- Product type: Browser-like tab component for embedding in a larger application.
- Function: Facilitates access to typically one website, omitting any UI to enter URLs or searches. Grants access via links to the rest of the World Wide Web.
- Users: Users of the embedding application. They may forget that they are inside another application.
- Architecture: Typically no settings UI and no integration into enterprise policy. Could share state with the user's default browser.
- Operational environment: Determined by the host application. Uncontrolled with respect to browser configuration.

5 Technical requirements for products

5.1 Introduction - Applicability of the requirements

The technical requirements of the present document apply under the product context described in Clause 4, which shall be in accordance with its intended use. The product shall comply with all applicable technical requirements of the present document at all times when operating in such a product context.

Not all requirements are universally applicable: The applicability of requirements may be based on use cases or specific capabilities of the product.

The extension-related requirements apply to products that offer the installation of third-party extensions. Products not offering third-party extension support are out of scope for those requirements.

Each requirement in Clause 5 has additional information about its applicability. The following list displays them.

- All products
- One or more Use Cases
- Product has a private mode
- Potentially any product
- Products that maintain their own root store, rather than using the OS's root store.
- Product supports extensions
- Product supports cross-origin embedding.

Please note, that they can be additive, such as for example UC-CONS + Product supports extensions.

This document uses several abbreviations within the labels of the requirements to identify topics that are present across the various categories of the requirements. This is their meaning:

- TLS: Transport Layer Security
- EXT: Browser extension
- MEM: Memory
- STORE: Storage
- ISO: Isolation
- PWR: Powerful Web Platform Features
- SOP: Same Origin Policy

5.2 Appropriate level of cybersecurity

5.2.0 Introduction

This clause addresses the requirements in the CRA [i.1] Annex 1 Part 1 (1).

5.2.1 REQ-ALC-1

The product shall show evidence of full compliance with all requirements of this standard that are applicable to its use-case.

Applicability: All products

Note: This first requirement is also the last requirement.

5.2.2 REQ-ALC-2

If the product has one of the use-cases as its baseline, but introduces novel features on top, then it shall display cybersecurity properties that mitigate those risks, potentially through the application of other harmonised standards.

Applicability: Potentially any product

Note: Please consult Annex B.3 for an introduction to risk acceptance in this standard.

5.3 No known exploitable vulnerabilities

5.3.0 Introduction

This clause addresses the requirements in the CRA [i.1] Annex 1 Part 1 (2) (a).

5.3.1 REQ-KEV-1

The product shall incorporate only components, including third-party and open-source elements, for which no unmitigated known exploitable vulnerabilities exist at the time of release.

Applicability: All products

NOTE 1: The manufacturer may rely on documentation in the form of a Software Bill of Materials (SBOM) as well as reasoning of why known vulnerabilities are not exploitable under the applicable, expected operational environment.

NOTE 2: For the purposes of the present requirement, a known exploitable vulnerability is considered mitigated where appropriate technical or operational measures, including measures described in the product guidance where relevant, sufficiently prevent or reduce the feasibility of exploitation in the product context, taking into account the intended purpose and reasonably foreseeable use of the product.

NOTE 3: A known exploitable vulnerability may be considered mitigated where its exploitation is heavily constrained by the product context. An example is a vulnerability in a library used only to process trusted data supplied by the manufacturer. The vulnerability should still be fixed, but its presence need not block a release.

NOTE 4: A manufacturer may become aware of a known exploitable vulnerability for which developing and qualifying a fix takes longer than the established patch cycle. In such cases, releasing a scheduled update that delivers other security fixes need not be blocked solely by the pending fix, where interim technical or operational measures sufficiently reduce the feasibility of exploitation.

5.3.2 REQ-MEM-KEV-1

The product’s source code that parses untrusted, non-trivial data, shall undergo ongoing automated dynamic analysis to identify vulnerabilities.

Applicability: All products

Example: Code that parses media files is subjected to fuzz testing in an environment that enables the detection of memory safety errors, such as LLVM sanitisers.

5.3.3 REQ-MEM-KEV-2

The product's source code shall undergo ongoing automated static analysis to identify vulnerabilities.

Applicability: All products

Example: Via LLM code analysis, static application security testing tools.

5.3.4 REQ-MEM-KEV-3

The product shall be implemented using programming languages, language features, and/or automation-enforced coding conventions that limit the introduction of memory safety vulnerabilities.

Applicability: All products

Example: Via the use of memory safe languages, bounds checked container classes, restrictions on the use of raw pointers.

5.4 Secure by default configuration

5.4.0 Introduction

This clause addresses the requirements in the CRA [i.1] Annex 1 Part 1 (2) (b).

Note: The CRA essential requirement laid out in Annex 1 Part 1 (2) (b) foresees an applicability exception to the secure configuration by default in case there is an agreement between manufacturer and business user in relation to a tailor-made product with digital elements.

5.4.1 REQ-TLS-SBD-1

The product shall, by default, reject TLS protocol versions, ciphers and security-relevant configurations that are not recommended by the normative TLS specifications referenced in this document.

Applicability: All products

5.4.2 REQ-TLS-SBD-2

For validation of TLS server certificates, the product shall be configured by default to use a trusted root store whose trust scope and inclusion and removal criteria are documented in and justified by the manufacturer's root-store policy and risk assessment.

Applicability: Products that maintain their own root store, rather than using the OS's root store.

5.4.3 REQ-EXT-SBD-1

The product shall execute extensions with minimal privileges by default.
Example: The extension execution process runs with no greater operating system privileges or capabilities than are required for the extension execution context.

Applicability: Product supports extensions

5.4.4 REQ-EXT-SBD-2

The product shall execute extensions in an isolated context.

Applicability: Product supports extensions

Note: This covers the default configuration - the exploitation mitigation is covered in Clause 5.12.

5.4.5 REQ-PWR-SBD-1

The product's technical documentation shall describe all web platform APIs it implements that fall under the definition of a Powerful Web Platform Feature, detailing the specific APIs and the mechanism by which express user permission is obtained before they can be used by a website.

Applicability: All products

5.4.6 REQ-PWR-SBD-2

The product shall provide a mechanism for users to select the default behaviour when a web page wishes to use a given Powerful Web Platform Feature, which shall at least include denying by default.

Applicability: All products

5.5 Security updates

5.5.0 Introduction

This clause addresses the requirements in the CRA [i.1] Annex 1 Part 1 (2) (c).

5.5.1 REQ-TLS-SU-1

The product's trusted root store shall be kept up to date in accordance with the manufacturer's documented root-store policy and risk assessment, which shall define the criteria and timeframe for adding and removing trust anchors.

Applicability: Products which maintain their own root store, rather than using the OS's root store.

5.5.2 REQ-EXT-SU-1

The product shall support automatic updates of extensions, and before installing an update shall cryptographically verify the update.

Applicability: Product supports extensions

Note: Extensions installed by enterprise policy, by the developer as unpacked extensions, or sideloaded directly by the user are out of scope of this requirement.

5.5.3 REQ-STORE-SU-1

The product shall maintain the validity of data stored to disk across updates.

Applicability: All products

5.5.4 REQ-STORE-SU-2

The product shall update the Public Suffix List regularly.

Applicability: All products

5.6 Authentication and access control

5.6.0 Introduction

This clause addresses the requirements in the CRA [i.1] Annex 1 Part 1 (2) (d).

5.6.1 REQ-STORE-ACC-1

The product shall store data and enforce access according to the Same Origin Policy.

Applicability: All Products

5.6.2 REQ-STORE-ACC-2

The product shall not share or make data available across browser profiles.

Applicability: All Products

5.6.3 REQ-PRIV-AAC-1

When in private mode, the product shall not expose storage data and history from other modes of browser usage.

Applicability: Product has a private mode

5.6.4 REQ-PRIV-CON-1

When in private mode, by default the product shall not support 3rd party cookies unless the third-party cookies are consistent with the `Partitioned` keyword.

Applicability: Product has a private mode

5.6.5 REQ-PRIV-CON-2

Upon ending the private mode, the product shall delete storage data and history that was created during its activities.

Applicability: Product has a private mode

5.6.6 REQ-EXT-AAC-1

The product shall enforce a granular permission model for the extension.
Applicability: Product supports extensions

Example: Permissions are decomposed into capability groupings rather than a single all-or-nothing grant.

5.6.7 REQ-EXT-AAC-2

The product shall grant an extension only the permissions declared in its manifest and granted in accordance with the product's permission model.
Applicability: Product supports extensions

Note: Some permissions are considered low-impact and auto-granted without an explicit user-visible prompt. The user-prompt requirement is covered separately by REQ-EXT-AAC-3.

5.6.8 REQ-EXT-AAC-3

The product shall prompt the user with the manifest-declared permissions prior to installation, listing the capabilities and implications of each permission, and allow the user to approve or decline the installation.

Applicability: Product supports extensions

Note: Extensions installed via the product's extension distribution channel. Extensions installed by enterprise policy, by the developer as unpacked extensions, or sideloaded directly by the user are out of scope of this requirement.

5.6.9 REQ-EXT-AAC-4

The product shall allow the user to review and revoke extension access to specific origins after installation.

Applicability: Product supports extensions

5.6.10 REQ-EXT-AAC-5

The product shall ensure isolation between the execution and data contexts of different extensions.

Applicability: Product supports extensions

5.6.11 REQ-SOP-AAC-2

The product shall isolate origin-specific data: cookies, Web Storage, IndexedDB and any other product-specific storage, ensuring it is not available to any other origin, except through protocol-determined methods.

Applicability: All Products

5.6.12 REQ-PWR-AAC-1

The product shall require express user permission before allowing a web page to use Powerful Web Platform Features.

Applicability: All Products

5.7 Confidentiality protection

5.7.0 Introduction

This clause addresses the requirements in the CRA [i.1] Annex 1 Part 1 (2) (e).

5.7.1 REQ-TLS-CON-1

The product shall, by default, support and prefer TLS versions and configurations classified as Recommended for TLS in Annex K. Legacy TLS versions and configurations may be supported only for interoperability where justified by the manufacturer's risk assessment.

Applicability: All products

NOTE 1: These algorithms are listed on page 37-38 of [1]. TLS 1.3 [i.7] is R, and TLS 1.2 [i.6] is L. The L[2025] algorithms (with CBC) are not considered state of the art and can be ignored.

NOTE 2: A minimal set of cipher suites includes TLS_CHACHA20_POLY1305_SHA256, TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA, TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256, TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA

NOTE 3: A minimal set of named groups includes X25519MLKEM768 and x25519

5.7.2 REQ-TLS-CON-2

The product shall treat a TLS connection as authenticated only if the TLS server certificate chain has been validated in accordance with RFC 5280 [i.17].

Applicability: All products

5.7.3 REQ-TLS-CON-3

The product shall verify that the TLS server certificate identifies the requested server before treating the TLS connection as authenticated.

Applicability: All products

Example: When presenting content served with cryptographic methods with a certain risk of exploitation, a web browser presents a user with a user interface element representing a broken lock, or an interstitial requiring user interaction to proceed.

5.7.4 REQ-TLS-CON-4

The product shall not present content as secure when delivered over HTTP, an unauthenticated TLS connection, or a TLS configuration that is not permitted by Annex K.

Applicability: All products

5.7.5 REQ-TLS-CON-5

For certificate validation failures or disallowed TLS configurations, the product shall warn the user and require explicit user action before allowing access, or block access where user override is not permitted.

Example: When presenting content served with cryptographic methods with a certain risk of exploitation, a web browser presents a user with a user interface element representing a broken lock, or an interstitial requiring user interaction to proceed.

Applicability: All products

5.7.6 REQ-TLS-CON-6

The product shall, by default, prefer HTTPS over HTTP where HTTPS is available.

Applicability: All products

5.7.7 REQ-EXT-CON-1]

The product shall prevent secrets stored by extensions from being read by other extensions or by web content.

Applicability: Product supports extensions

NOTE: This requirement addresses the platform-enforced isolation boundary between extensions, and between extensions and web content. It does not address application-level leaks within an extension's own code.

5.7.8 REQ-STORE-CON-1]

The product shall be able to be configured so as not to send third-party cookies by default. They may be supported consistent with the Partitioned attribute.

Applicability: All products

NOTE: The product can provide users the ability to share third-party cookies, whether on a case-by-case basis through interaction as in Storage Access API or throughout their browser profile through configuration.

5.7.9 REQ-STORE-CON-2

The product shall make use of OS access control, encryption methods or other of the product's or operating system's mechanisms to ensure confidentiality of disk-stored data.

Applicability: All products

5.7.10 REQ-STORE-CON-3

The product shall store browser cache data such that they are keyed to both top-level site and resource.

Applicability: All products

5.7.11 REQ-SOP-CON-1

The product shall only allow scripts to access the body of a cross-origin HTTP response where the source origin has opted to allow access by the requesting origin.

Applicability: All products

Example: Implementation of Cross-Origin Resource Sharing (CORS)

5.7.12 REQ-SOP-CON-2

Where the product allows cross-origin embedding of a resource in documents, the product shall limit the information about that resource exposed to the embedding document, except where the source origin permits broader access.

Applicability: Product supports cross-origin embedding.

Example: Cross-origin embedding of the height and width of an image are exposed to scripts, while the contents of the image are not exposed unless CORS is enabled by the source origin of the image.

5.8 Integrity protection

5.8.0 Introduction

This clause addresses the requirements in the CRA [i.1] Annex 1 Part 1 (2) (f).

Note: TLS-related clauses contribute to integrity.

5.8.1 REQ-EXT-INT-1

By default, the product shall warn or obstruct the user from installation or update of extensions that are not cryptographically verified.
Applicability: Product supports extensions

5.8.2 REQ-PWR-INT-1

The product shall protect permission prompts against manipulation by the web page.
Applicability: All products.

5.8.3 REQ-PWR-INT-2

The product shall only enable Powerful Web Platform Features for use within Secure Contexts.
Applicability: All products

5.8.4 REQ-ISO-MAS-1

Each stabilised web-exposed interface implemented by the product shall conform to a publicly available specification or be specified in sufficient technical detail to permit an independent implementation of that interface.

Applicability: All products

NOTE 1: Referencing web specifications is impossible due to their transient nature.
NOTE 2: Examples of specifications are MDN Documentation, WEBIDL, user guides.
NOTE 3: Web-exposed interfaces are broad (JavaScript, WASM, CSS, etc.)

5.9 Data minimisation

5.9.0 Introduction

This clause addresses the requirements in the CRA [i.1] Annex 1 Part 1 (2) (g).

Note: TLS-related clauses contribute to data minimisation.

5.9.1 REQ-PWR-DM-1

Powerful Web Platform Feature permissions shall be origin-scoped by default, with broader scoping permitted only via express user action, or opt-in by the origins.

Applicability: All products

5.9.2 REQ-PWR-DM-2

Powerful Web Platform Feature permissions decisions shall apply to each Browser Profile separately.

Applicability: All products

5.10 Availability protection

5.10.0 Introduction

This clause addresses the requirements in the CRA [i.1] Annex 1 Part 1 (2) (h).

5.10.1 REQ-EXT-AP-1

The product shall make the best effort to prevent the ability of an extension to make the product unavailable.

Applicability: Product supports extensions

5.10.2 REQ-STORE-AP-2

The product shall retain data stored to disk in case of a crash and make it available upon product restart.

Applicability: All products

5.11 Non-interference

5.11.0 Introduction

This clause addresses the requirements in the CRA [i.1] Annex 1 Part 1 (2) (i).

5.11.1 REQ-EXT-IM-1

If the product is running with elevated system privileges, extensions shall be prevented from executing with those elevated privileges.

Applicability: Product supports extensions

5.11.2 REQ-EXT-IM-2

The product shall permit extensions to communicate with native applications only when declared in the extension manifest and the native application is configured according to the product requirements for doing so.

Applicability: Product supports extensions

5.11.3 REQ-PWR-IM-1

The product shall by default deny cross-origin iframes access to Powerful Web Platform Features.

Applicability: All products

5.11.4 REQ-ISO-IM-1

[REQ-ISO-IM-1]: Each stabilised public network protocol implemented by the product shall conform to a publicly available specification, or be specified in sufficient technical detail to permit an independent implementation of that protocol.

Applicability: All products

5.12 Attack surface minimisation

5.12.0 Introduction

This clause addresses the requirements in the CRA [i.1] Annex 1 Part 1 (2) (j).

5.12.1 REQ-MEM-MAS-1

The product shall use operating system functionality to restrict the capabilities and privileges of architectural components of the product.

Applicability: All products

Example: Via sandboxed processes, system call filtering.

5.12.2 REQ-EXT-MAS-1

For each of the product's extension APIs, the following shall be defined: its purpose, its inputs and outputs, the permissions it requires and its security-related behaviour. This documentation shall accurately reflect the actual behaviour of each extension API.

Applicability: Product supports extensions

5.12.3 REQ-EXT-MAS-2]

The product shall support enterprise policy controls allowing administrators to disable the extension feature entirely, or specify an allow-list of extensions.

Applicability: Enterprise browsers (UC-INST) that support extensions

5.12.4 REQ-EXT-MAS-3

The product shall enforce the scope of extension resources available to web content, as declared in the manifest.

Applicability: Product supports extensions

Example: Extensions may bundle assets in their packages, and browsers may allow them provide web pages with access to static assets, such as images, scripts and styles. Extensions declare these assets in their manifest, and browsers restrict access to only the declared assets.

5.12.5 REQ-SOP-MAS-1]

The product shall enforce restrictions from a source origin on how its resources may be used across origins, as defined in the user documentation.
Applicability: All products

Example: Usage of cross-origin options such as X-Frame-Options or Cross-Origin-Resource-Policy allow source origins to specify usage constraints for products to receive and enforce.

5.13 Exploit mitigation

5.13.0 Introduction

This clause addresses the requirements in the CRA [i.1] Annex 1 Part 1 (2) (k).

5.13.1 REQ-EXT-EMM-1]

The product shall enforce a default Content Security Policy for all extension-origin pages and background contexts that prevents the execution of inline scripts and remote code.

Applicability: Product supports extensions

5.13.2 REQ-EXT-EMM-2]

The product shall validate an extension's manifest before installation and update, reject manifests that are malformed or contain disallowed content, and ignore unrecognised optional fields.

Applicability: Product supports extensions

5.13.3 REQ-ISO-EMM-1]

The product shall separate certain product components from each other to reduce the scope of exploits, using process isolation or similar industry standard mitigations.

Applicability: All products

Example: Containing the renderer and networking in separate processes

5.13.4 REQ-ISO-EMM-2]

The product shall isolate different sites from each other.

Applicability: All products

Example: Process isolation of sites

5.13.5 REQ-MEM-EMM-1]

The product shall include runtime assertions as to the correctness of program state, which fail safely when violated.

Applicability: All products

Example: Via the use of ASSERT() or CHECK() statements that result in an unexploitable crash.

5.13.6 REQ-MEM-EMM-2]

The product shall use compiler features designed to reduce the risk of exploitation.

Applicability: All products

Example: stack canaries, source code fortification, variable initialisation.

5.13.7 REQ-MEM-EMM-3]

The product shall use mitigation technologies provided by the operating systems and/or hardware on which the web browser is designed to run.

Applicability: All products

Example: hardware-assisted memory tagging, Address Space Layout Randomization.

5.14 Monitoring

5.14.0 Introduction

This clause addresses the requirements in the CRA [i.1] Annex 1 Part 1 (2) (l).

5.14.1 REQ-TLS-LOG-1

The product shall provide a user interface indicating the origin of the current top-level document and whether its transport is encrypted and authenticated.

Applicability: All products

5.14.2 REQ-EXT-LOG-1

The product shall provide the user the ability to identify which user-installed extensions are currently running, and the permissions in effect for each.

Applicability: Product supports extensions

5.14.3 REQ-STORE-LOG-1

The product shall provide an interface for viewing information about stored data at a granularity of site or narrower (origin).

Applicability: All products

5.14.4 REQ-PRIV-LOG-1

When in private mode, the product shall provide visual cues so that the user can easily distinguish that they are browsing in a private mode.

Applicability: Browser has a private mode.

5.15 Factory reset and data portability

5.15.0 Introduction

This clause addresses the requirements in the CRA [i.1] Annex 1 Part 1 (2) (m).

5.15.1 REQ-PWR-DRT-1

The product shall provide a user interface allowing revocation of previously granted permissions for Powerful Web Platform Features.

Applicability: All products

5.15.2 REQ-TLS-DRT-1]

Where the product allows TLS-related settings to be changed, it shall provide a mechanism to restore browser-controlled TLS settings to the manufacturer's default secure configuration, including enabled protocol versions, cipher suite or algorithm preferences, certificate-validation settings, and, where applicable, the current browser-managed trusted root store.

Applicability: Web browsers which allow changing TLS-related settings.

5.15.3 REQ-EXT-DRT-1

The product shall enable the removal of individual extensions, which shall delete all data associated with the extension, and revoke all permissions granted to it.

Applicability: Product supports extensions

5.15.4 REQ-STORE-DRT-1

When storage data is deleted from the product's internal data stores, the product shall purge it so that it is no longer retrievable.

Applicability: All products

5.15.5 REQ-STORE-DRT-2

The product shall provide reset functionality that removes all stored data across all sites and browser profiles.

Applicability: All products

5.15.6 REQ-STORE-DRT-3

The product shall have a user interface for deleting storage at a granularity of site or narrower (origin).

Applicability: All products

6 Assessment criteria for compliance with technical requirements

6.1 Introduction to the assessment and compliance criteria

This clause provides objective and reproducible assessment criteria to determine whether a product complies with the technical security requirements of clause 5.
For each cybersecurity requirements defined in clause 5, the following clauses specify assessment criteria to determine if the technical requirement is met.

Where Assessment Activities are presented as a numbered list, they are to be carried out sequentially. Where the Assessment Activities are presented as bullet points, the order of execution is not critical to the assessment and can be undertaken in any order.

The assessment criteria for each security requirements are described in a structured manner, as follows:

- **Assessment reference:** Refers to the identifier of the concerned technical requirement.

- **Assessment objective:** Defines the security property or capability that shall be verified, ensuring that the assessment remains focused on the intent of the requirement.

- **Assessment preparation:** Describes the environment, setup, and preconditions required before executing the test with a view to guaranteeing consistent evaluation results. It includes the following elements as applicable:
  - Test environment: Describe the hardware, software, and network setup used for the assessment, including versions, topology, and any relevant dependencies.
  - Preconditions: Specify any configurations, credentials, or operational states that should be established before the test (e.g. product initialized, certificates loaded, user roles created).
  - Required tools: Identify the tools or software necessary to perform the assessment (e.g. vulnerability scanners, protocol fuzzers, traffic analysers, static code analysers, cryptographic test suites).
  - Required information/documentation for the assessment: Specify all information that is necessary to perform the assessment.
  - Reference any vendor provided setup guides, configuration instructions, or operational manuals, as well as any relevant standards or technical notes, that define how the product shall be configured or operated for the assessment.
- **Assessment activities:** Provides execution steps to be performed such that the same assessment verdict would be reached when repeating these steps now or at a later point in time. Assessment activities may include, as applicable:
  - Review information/documentation for the assessment to confirm that the described implementation matches the requirement (e.g. verify that the security architecture document specifies TLS 1.2 or higher for all external interfaces, or that the password policy aligns with the defined threshold).
  - Perform security functional tests to verify the completeness and correctness of the information/documentation for the assessment.
  - Perform security functional or penetration tests to verify that implemented controls are correctly implemented e.g. to prevent unauthorized access or data modification (e.g. via

attempting to log in with invalid credentials to test lockout enforcement or trying to modify protected configuration files without administrative privileges).
- Analyse code or binaries to identify potential security weaknesses or misconfigurations (e.g. perform static analysis to detect hardcoded credentials or use dynamic analysis tools to identify buffer overflow or injection vulnerabilities).
- Inspect configurations to ensure that required security parameters are correctly applied (e.g. check that weak cipher suites are disabled, two-factor authentication is enabled, and least-privilege access controls are configured in the system).
- Observe runtime behaviour to confirm that protections such as encryption, authentication, and integrity verification operate as intended (e.g. monitor network traffic to ensure data in transit is encrypted or observe system logs to verify successful validation of digital signatures during startup).

- **Assessment verdict:** Defines the pass/fail criteria.
  - **Pass:** The assessment is considered passed if the product demonstrably fulfils the requirement and meets the defined security thresholds. Examples of such thresholds include:
    - Minimum cryptographic strength (e.g. AES-128 or higher);
    - Password policy limits (e.g. minimum of 12 characters);
    - Login protection mechanisms (e.g. account lockout after five consecutive failed attempts);
    - Resistance to a specified attack potential (e.g. equivalent to CSA High/AVA_VAN.3 or higher).
  - **Fail:** The assessment is considered failed if the requirement is not fulfilled, or if the defined security thresholds are not achieved (e.g. insufficient key length, missing authentication enforcement, or inadequate resistance to the required attack potential).
- **Assessment evidence**: Defines the artefacts and documentation collected to demonstrate that the requirement has been assessed and fulfilled. The evidence shall be sufficient to enable independent verification of the assessment results and to demonstrate compliance with the relevant CRA essential requirements. The supporting evidence includes, where applicable:
  - Test or assessment reports showing the steps performed and results obtained;
  - Logs, configuration files, or audit traces demonstrating the implementation of the requirement;
  - Screenshots, captures, or console outputs confirming the correct execution or protection behaviour;

6.2 Appropriate level of cybersecurity

6.2.1 ACC-ALC-1

6.2.1.1 Reference

The product shall show evidence of full compliance with all requirements of this standard that are applicable to its use-case.

6.2.1.2 Objective

Verify that the applicable requirement set has been correctly derived for the product and that each applicable requirement has been assessed under Clause 6.

6.2.1.3 Preparation

- Identification of the product's use case or use cases per Clause 4.6.
- Where the product does not map to a defined use case, a risk assessment performed under the methodology of Annex B.
- The manufacturer's cybersecurity risk assessment for the product.
- A mapping of the requirements of Clause 5 to the product, identifying each requirement as applicable or excluded.
- A substantiated justification for each requirement marked as excluded.
- The set of assessment records produced for each applicable requirement under Clause 6.
- Product documentation sufficient to confirm the use case and deployment context.

6.2.1.4 Activities

- Confirm that the product's use case or use cases are identified, or that a risk assessment has been performed under Annex B.
- Confirm that the applicable requirement set has been derived from the identified use case or risk assessment.
- For each requirement marked as excluded, confirm that the exclusion is substantiated by the risk assessment and consistent with the product's use case.
- For each applicable requirement, confirm that an assessment record exists under the corresponding part of Clause 6.
- Confirm that no applicable requirement is left without an assessment record.

6.2.1.5 Verdict

- Pass: The applicable requirement set is correctly derived from the product's use case or risk assessment, every applicable requirement has an assessment record under Clause 6, and every exclusion is substantiated.
- Fail: The applicable requirement set is not derived or not justified, OR one or more applicable requirements lack an assessment record, OR one or more exclusions are not substantiated.

6.2.1.6 Evidence

- Use case identification per Clause 4.6, or the Annex B risk assessment record.
- The requirement applicability mapping, with justification for each excluded requirement.
- An index of the per-requirement assessment records under Clause 6.
- The manufacturer's cybersecurity risk assessment with version.

6.2.2 ACC-ALC-2

6.2.2.1 Reference

If the product has one of the use-cases as its baseline, but introduces novel features on top, then it shall display cybersecurity properties that mitigate those risks, potentially through the application of other harmonised standards.

6.2.2.2 Objective

Verify that risks introduced by novel features beyond the appropriate baseline use case have been identified and mitigated, including through other harmonised standards where applicable.

6.2.2.3 Preparation

- Identification of the baseline use case per Clause 4.6.
- Technical documentation describing the novel features introduced on top of the baseline use case.
- A risk assessment under the methodology of Annex B covering those novel features.
- A mapping of each novel-feature risk to its mitigating cybersecurity property or control.
- Where a mitigation relies on another harmonised standard, identification of that standard and the relevant assessment records produced under it.
- Technical documentation sufficient to confirm the presence and behaviour of the novel features.

6.2.2.4 Activities

- Confirm that the baseline use case is identified and that the novel features are documented.
- Confirm that a risk assessment under Annex B has been performed for the novel features.
- For each identified risk, confirm that a mitigating cybersecurity property or control is specified.
- Where a mitigation relies on another harmonised standard, confirm that an assessment record exists under that standard.
- Confirm that no identified novel-feature risk is left without a mitigation.

6.2.2.5 Verdict

- Pass: The novel features are identified, their risks are assessed under Annex B, and each identified risk has a specified mitigation, with any reliance on another harmonised standard supported by an assessment record under that standard.
- Fail: The novel features are not identified or not risk-assessed, OR one or more identified risks lack a mitigation, OR a mitigation relying on another harmonised standard lacks a supporting assessment record.

6.2.2.6 Evidence

- The baseline use case identification per Clause 4.6.
- Documentation of the novel features.
- The Annex B risk assessment covering the novel features, with version.
- The mapping of novel-feature risks to mitigations.
- Identification of any other harmonised standards relied upon, with an index of the relevant assessment records under them.

6.3 No known exploitable vulnerabilities

6.3.1 ACC-KEV -1

6.3.1.1 Reference

The product shall incorporate only components, including third-party and open-source elements, for which no unmitigated known exploitable vulnerabilities exist at the time of release.

6.3.1.2 Objective

Verify that no component carries a known exploitable vulnerability that is unmitigated at the assessed release. "Known exploitable" means flagged as exploited or exploitable by an authoritative source. "Unmitigated" means neither fixed, nor excluded by substantiated VEX, nor neutralised by an applied mitigation.

6.3.1.3 Preparation

- Clean target host and installation package from the manufacturer's official channel
- Software Bill of Materials (SBOM) for the product and bundled components
- Source code
- Vulnerability Exchange (VEX) document or equivalent mitigation register aligned to the assessed product version
- SBOM correlation tooling with vulnerability database synchronised within 24 hours
- Access to vulnerability intelligence sources (NVD, EUVD, GHSA, vendor advisories)
- Access to exploitation intelligence sources (CISA KEV, EUVD exploitation data, EPSS, public exploit and PoC registers)

6.3.1.4 Activities

- Install the product using defaults
- Correlate all SBOM components against vulnerability database entries to produce the candidate vulnerability set
- For each candidate, determine known exploitable status from exploitation intelligence sources, and retain only vulnerabilities with confirmed exploited or exploitable status
- For each known exploitable vulnerability, check the VEX or mitigation record for a fixed, not_affected, or applied-mitigation status tied to the assessed version
- Verify each claimed mitigation by code, configuration, or controlled test

6.3.1.5 Verdict

- Pass: No SBOM component carries a known exploitable vulnerability, OR every known exploitable vulnerability is covered by a verified mitigation (fixed, substantiated not_affected, or applied mitigation) effective at the assessed release
- Fail: One or more known exploitable vulnerabilities remain unmitigated at the assessed release, OR a mitigation claim cannot be substantiated, OR SBOM coverage is insufficient to support the correlation

6.3.1.6 Evidence

- SBOM with hash
- Raw correlation output identifying candidate vulnerabilities per component, pre-suppression
- Exploitability determination for each candidate, with source and retrieval timestamp establishing known exploitable status
- VEX or mitigation document with version tied to the assessed product
- Verification record for each fixed, not_affected, or applied-mitigation claim
- Vulnerability and exploitation database snapshots used during correlation, with retrieval timestamps

6.3.2 ACC-MEM-KEV-1

6.3.2.1 Reference

Assessment of [REQ-MEM-KEV-1]
The web browser's source code that parses untrusted, non-trivial data shall undergo ongoing automated dynamic analysis to identify vulnerabilities.

6.3.2.2 Objective

The manufacturer conducts ongoing automated dynamic analysis to identify vulnerabilities in the web browser.

6.3.2.3 Preparation

- Documentation describing how code is selected to undergo automated dynamic analysis.
- Documentation describing how, and how often, the automated dynamic analysis is conducted.
- Documentation describing a methodology for measuring and improving the code coverage of their dynamic analysis tools over time.
- Code coverage reports.
- A sample of bug tracker entries generated by each dynamic analysis technique described in the documentation.

6.3.2.4 Activities

The following steps are to be carried out in order:

1. Review the documentation.
2. Review the code coverage reports.
3. Review the bug tracker entries.

6.3.2.5 Verdict

- **Pass**:
  - The documentation describes a methodology for how code is selected to undergo automated dynamic analysis.
  - The documentation describes one or more dynamic analysis techniques and how they are automated and integrated into the web browser's software development lifecycle.
  - The documentation describes a robust methodology for measuring and improving the code coverage of their dynamic analysis tools over time.
  - Code coverage reports demonstrate that the dynamic analysis achieves the coverage targets defined in the manufacturer's methodology, or show documented progress toward those targets.
  - The bug tracker entries show evidence of being filed in an automated fashion as described in the documentation.
  - The bug tracker entries describe identified vulnerabilities.
  - The bug tracker entries show the identified vulnerabilities have been fixed.
- **Fail**:
  - Any of the above are not fulfilled.

6.3.2.6 Evidence

- Log of documentation and bug tracker entry review.

6.3.3 ACC-MEM-KEV-2

6.3.3.1 Reference

Assessment of [REQ-MEM-KEV-2]
The web browser's source code shall undergo ongoing automated static analysis to identify vulnerabilities.

6.3.3.2 Objective

The manufacturer conducts ongoing automated static analysis to identify vulnerabilities in the web browser.

6.3.3..3 Preparation

- Documentation describing the automated static analysis that is conducted.
- A sample of bug tracker entries generated by each static analysis technique described in the documentation.

6.3.3.4 Activities

The following steps are to be carried out in order:
1. Review the documentation.
2. Review the bug tracker entries.

6.3.3.5 Verdict

- **Pass**:
  - The documentation describes one or more static analysis techniques and how they are automated and integrated into the web browser's software development lifecycle.
  - The bug tracker entries show evidence of being filed in an automated fashion as described in the documentation.
  - The bug tracker entries describe identified vulnerabilities.
  - The bug tracker entries show the identified vulnerabilities have been fixed.
- **Fail**:
  - Any of the above are not fulfilled.

6.3.3.6 Evidence

- Log of documentation and bug tracker entry review.

6.3.4 Assessment ACC-MEM-KEV-3

6.3.4.1 Reference

Assessment of [REQ-MEM-KEV-3]

The manufacturer shall use programming languages, language features, and/or automation-enforced coding conventions that limit the introduction of memory safety vulnerabilities.

6.3.4.2 Objective

The web browser's source code is hardened against the introduction of memory safety vulnerabilities.

6.3.4.3 Preparation

- Documentation describing the techniques the manufacturer uses to limit the introduction of memory safety vulnerabilities into the web browser.
- Documentation describing how those methods are applied to the architectural components of the web browser.
- For architectural components to which those methods are not applied, documentation describing the mitigations in place to reduce the risk of memory safety vulnerabilities from being successfully exploited.
- Documentation describing the policies by which those methods are applied to the web browser's software development lifecycle.
- Relevant software development lifecycle artefacts, such as design documents, bug tracker entries, security review outcomes.
- The web browser's source code.

6.3.4.4 Activities

The following steps are to be carried out in order:

1. Review the documentation.
2. For each of the methods used to limit the introduction of memory safety vulnerabilities, confirm via spot checks of the source code that the method is applied in accordance with the documentation.
3. Confirm the relevant software development lifecycle policies are being followed via spot checks of artefacts.

6.3.4.5 Verdict

- **Pass**:
  - The documentation describes methods that limit the introduction of memory safety vulnerabilities.
  - The documentation maps those methods to the architectural components of the web browser.
  - The mitigations applied to relevant architectural components convincingly reduce the risk of memory safety vulnerabilities from being successfully exploited.
  - The relevant software development lifecycle policies are being followed.
- **Fail**:
  - Any of the above are not fulfilled.

6.3.4.6 Evidence

- Log of documentation, source code, and artefact review.

6.4 Secure by default configuration

Proposed ESR code SBD

6.4.1 ACC-TLS-SBD-1

6.4.1.1 Reference

Assessment of [REQ-TLS-SBD-1]
The product shall, by default, reject TLS protocol versions, ciphers and security-relevant configurations that are not recommended by the normative TLS specifications referenced in this document.

6.4.1.2 Objective

The product is configured by default to reject TLS protocol versions, ciphers and security-relevant configurations that are not recommended by the normative TLS specifications referenced in this document.

6.4.1.3 Preparation

- Install the product from scratch, or reset it to its default settings.

Configure a web server to expose websites served with several protocol versions, ciphers or configurations that are not recommended by the normative TLS specifications referenced in this document.

6.4.1.4 Activities

The following steps are to be carried out in order:

1. Navigate to each website and observe how it loads.

6.4.1.5 Verdict

- **Pass**: Each navigation attempt to the vulnerable website is blocked by the product.
- **Fail**: Any of the above are not fulfilled.

6.4.1.6 Evidence

- Logs or screenshots showing the product's response to the navigation.

6.4.2 ACC-TLS-SBD-2

6.4.2.1 Reference

Assessment of [REQ-TLS-SBD-2]

For validation of TLS server certificates, the product shall be configured by default to use a trusted root store whose trust scope and inclusion and removal criteria are documented in and justified by the manufacturer's root-store policy and risk assessment.

6.4.2.2 Objective

Assess that, for validation of TLS server certificates, the product is configured by default to use a trusted root store whose trust scope and inclusion and removal criteria are documented in and justified by the manufacturer's root-store policy and risk assessment.

6.4.2.3 Preparation

- Install the product from scratch, or reset it to its default settings.
- Locate the technical documentation indicating the product's trusted root store policy.
- Prepare a web server with TLS signed from a trusted root.

6.4.2.4 Activities

The following steps are to be carried out in order:
1. Check that the trusted root policy exists
2. Navigate to the website and observe the results

6.4.2.5 Verdict

- **Pass**:
  - The trusted root policy exists
  - Navigating to the trusted site succeeds
- **Fail**: Any of the above are not fulfilled.

6.4.2.6 Evidence

- Logs or screenshots showing the product's response to the navigation
- Root store policy and listing of trusted roots

6.4.3 ACC-EXT-SBD-1

6.4.3.1 Reference

Assessment of [REQ-EXT-SBD-1]
The product shall execute extensions with minimal privileges by default.

6.4.3.2 Objective

Failure mode: extensions execute at full product privilege, or successfully invoke privileged browser or OS APIs.

6.4.3.3 Preparation

- A test extension declaring no permissions.
- Tools for inspecting processes on the operating system.
- Code executing privileged browser or OS APIs from inside the extension.

6.4.3.4 Activities

The following steps are to be carried out in order:
1. Install the test extension
2. Inspect the OS-level process the product runs the extension in, record privilege level
3. Inspect the log of the extension's attempt to run privileged APIs

6.4.3.5 Verdict

- **Pass**:
  - OS privilege level of the extension process is less than the product process.
  - The extension invocation of privileged APIs failed.
- **Fail**:
  - OS privilege level of the extension process equals the product process, or the extension's invocation of privileged APIs succeeded.
  - The extension escalates privilege at runtime via a privileged API not enumerated by activities.
  - The extension is initially low-privilege but escalates after the inspection point.

6.4.3.6 Evidence

- Process inspection log.
- Extension console log.

6.4.4 ACC-EXT-SBD-2

6.4.4.1 Reference

Assessment of [REQ-EXT-SBD-2]
The product shall execute extensions in an isolated context.

6.4.4.2 Objective

Failure mode: an extension reads or modifies state belonging to the product, a loaded web page, or the operating system.

6.4.4.3 Preparation

- Product private state is selected, such as a unique string, or a configuration value.
- A text file on the operating system, with test data.
- A web page loaded in the product, with test data.
- A test extension declaring no permissions, and executing attempts to read and modify state in the product, the loaded web page, and the operating system.
- Console log access.

6.4.4.4 Activities

The following steps are to be carried out in order:
1. Install the test extension
2. Load a web page in the product

6.4.4.5 Verdict

- **Pass**:
  - Product private state data access was denied.
  - File system text file access was denied.
  - Web page test data access was denied.

- **Fail**:
  - Cross-context state access was possible in the product, the loaded web page, or the operating system.

6.4.4.6 Evidence

- Console log output from the test extension demonstrated state access or denial.

6.4.5 ACC-PWR-SBD-1

6.4.5.1 Reference

Assessment of [REQ-PWR-SBD-1]
The product's technical documentation shall describe all web platform APIs it implements that fall under the definition of a Powerful Web Platform Feature, detailing the specific APIs and the mechanism by which express user permission is obtained before they can be used by a website.

6.4.5.2 Objective

The Powerful Web Platform Features supported by the product are documented.

6.4.5.3 Preparation

- The product's technical documentation.

6.4.5.4 Activities

- Review the documentation.

6.4.5.5 Verdict

- **Pass**:
  - The product's technical documentation correctly lists the Powerful Web Platform Features supported by the product.
  - For each Powerful Web Platform Feature listed, the mechanism by which express user permission is obtained is detailed.
- **Fail**:
  - Any of the above are not fulfilled.

6.4.5.6 Evidence

- Log of documentation review.

6.4.6 ACC-PWR-SBD-2

6.4.6.1 Reference

Assessment of [REQ-PWR-SBD-2]

The product shall provide a mechanism for users to select the default behaviour when a web page wishes to use a given Powerful Web Platform Feature, which shall at least include denying by default.

6.4.6.2 Objective

The product provides a mechanism for users to configure default behaviour when a web page requests to use a Powerful Web Platform Feature.

6.4.6.3 Preparation

- The product installed from scratch, or reset to its default settings.
- Test web page or pages that exercises every Powerful Web Platform Feature identified in the product's technical documentation.

6.4.6.4 Activities

The following steps are to be carried out in order:

1. Locate the product's UI that allows configuring the default behaviour of individual Powerful Web Platform Features.
2. Verify the existence of options to deny or block access by default to every Powerful Web Platform Features supported by the product.
3. For each Powerful Web Platform Feature, set the default behaviour to block or deny.
4. Navigate to the test web page or pages and trigger the request to access every Powerful Web Platform Feature.

6.4.6.5 Verdict

- **Pass**:
  - The product settings provide an option to individually deny access to all supported Powerful Web Platform Features by default.
  - Applying the setting automatically blocks the test web page's access to Powerful Web Platform Features without presenting a permission prompt to the user.
- **Fail**:
  - The deny-by-default option does not exist.
  - Any Powerful Web Platform Feature is still accessible, or still prompts the user despite the deny-by-default setting being applied.

6.4.6.6 Evidence

- Screenshots of the settings UI presenting the option.
- Console logs and/or UI captures from the test web page showing the denied request.

6.5 Security updates

Proposed ESR code: SU

6.5.1 ACC-TLS-SU-1

6.5.1.1 Reference

Assessment of [REQ-TLS-SU-1]
The product's trusted root store shall be kept up to date in accordance with the manufacturer's documented root-store policy and risk assessment, which shall define the criteria and timeframe for adding and removing trust anchors.

6.5.1.2 Objective

Assess that the product's trusted root store is kept up to date in accordance with the manufacturer's documented root-store policy and risk assessment.

6.5.1.3 Preparation

- Install the product from scratch, or reset it to its default settings.
- Locate the documented policy describing how and when the trusted root store is updated.
- Capture the initial trusted root store contents.

6.5.1.4 Activities

The following steps are to be carried out in order:

1. Exercise the documented root store update mechanism (e.g., product update, separate channel).
2. Capture the trusted root store contents after the update.
3. Compare the change against the documented policy.

6.5.1.5 Verdict

- **Pass**:
  - The documented update mechanism functions as documented
  - The resulting change to the root store is consistent with the documented policy.
- **Fail**: Either of the above are not fulfilled.

6.5.1.6 Evidence

- Documented root store update policy.
- List of trusted roots before and after the update.

6.5.2 ACC-EXT-SU-1

6.5.2.1 Reference

Assessment of [REQ-EXT-SU-1]
The product shall support automatic updates of extensions, and before installing an update shall cryptographically verify the update. Applicability: Extensions installed via the product's extension distribution channel. Extensions installed by enterprise policy, by the developer as unpacked extensions, or sideloaded directly by the user are out of scope of this requirement.

6.5.2.2 Objective

The product supports automated updates of extensions, and cryptographically verifies extension updates prior to installation.

6.5.2.3 Preparation

- A correctly signed test extension installed via the product's extension distribution channel.
- Update variations of the test extension: signed correctly, signed incorrectly, unsigned, and malformed.
- A test extension update server.

6.5.2.4 Activities

The following steps are to be carried out in order:

1. Install the test extension.
2. Configure the update server with an update variation.
3. Perform update per product documentation.
4. Repeat steps 2-3 with each variation.

6.5.2.5 Verdict

- **Pass**:
  - Correctly signed update is installed.
  - All other variations are not installed.
- **Fail**: Any of the above are not fulfilled.

6.5.2.6 Evidence

- Product user interface per the documentation, showing the update status for an extension.

6.5.3 ACC-STORE-SU-1

6.5.3.1 Reference

Assessment of [REQ-STORE-SU-1]
The product shall maintain the validity of data stored to disk across updates.

6.5.3.2 Objective

Assess whether the product maintains the validity of data stored to disk across updates.

6.5.3.3 Preparation

- Reset browser to factory default settings.
- Prepare tooling to initiate an update.
- Prepare tooling that will provide visibility into data stored in the browser.
- Load data into browser to cover available storage mechanisms, including LocalStorage, Cookie storage and IndexedDB.

6.5.3.4 Activities

1. Making use of tooling, verify the data that has been added to storage.
2. Perform the update of the browser.
3. Verify whether the data is available in storage after the update.

6.5.3.5 Verdict

- **Pass**:
  - The data is available prior to browser update.
  - The data available subsequent to browser update is identical.
- **Fail**: Any of the above are not fulfilled.

6.5.3.6 Evidence

- Screenshot(s) or log output from tooling to demonstrate each verdict

6.5.4 ACC-STORE-SU-2

6.5.4.1 Reference

Assessment of [REQ-STORE-SU-2]
The product shall update the Public Suffix List regularly.

6.5.4.2 Objective

Assess whether the product updates the public suffix list regularly.

6.5.4.3 Preparation

- Technical documentation containing public suffix list update policy.
- Browser with out-of-date public suffix list, with known missing suffix.
- Identify methodology or tooling to test Public Suffix List entries, e.g.
  - Set a cookie at the PSL level, or
  - Tooling to inspect and output PSL.

6.5.4.4 Activities

1. Verify state of the missing PSL suffix.
2. Await update according to policy.
3. Verify the updated state of the missing PSL suffix.

6.5.4.5 Verdict

- **Pass**:
  - Testing initially shows known-missing entry as not present in internal list of eTLDs.
  - After update period, testing shows known-missing entry is present in internal list of eTLDs.
- **Fail**: Any of the above are not fulfilled.

6.5.4.6 Evidence

- Screenshot(s) or log output from tooling to demonstrate each verdict

6.6 Authentication and access control

6.6.1 ACC-STORE-AAC-1

6.6.1.1 Reference

Assessment of [REQ-STORE-AAC-1]
The product shall isolate storage data, ensuring it is not available outside the origin creating it, except through protocol determined methods.

6.6.1.2 Objective

Assess whether the product presents stores and enforces access within an origin.

6.6.1.3 Preparation

- Prepare tooling that will provide visibility into data available for a given context.
- Website hosted on origin A, that will provide storage data and does not provide access to origin B.
- Website hosted on origin B, that will provide storage data and does not provide access to origin A.
- Reset browser to factory default settings.

6.6.1.4 Activities

1. Navigate to the website on origin A.
2. Making use of tooling, verify the data is added to storage, and is available in the context.
3. Navigate to the website on origin B.
4. Verify that the data set when accessing the first site is unavailable in the new context.
5. Verify that the data set when accessing the second site is available in the new context.
6. Navigate back to the website on origin A.
7. Verify that the data set by the site is still available.

6.6.1.5 Verdict

8. **Pass**:
    - When site adds data to storage it is available.
    - When the site looks for data, it does not see data from the other origin
    - When the second site sets data it is also available.
    - When navigating back to the first site, the data set when visiting initially will still be available.
9. **Fail**: Any of the above are not fulfilled.

6.6.1.6 Evidence

- Screenshot(s) or log output from tooling to demonstrate each verdict

6.6.2 ACC-STORE-AAC-2

6.6.2.1 Reference

Assessment of [REQ-STORE-ACC-2]
The product shall not share or make data available across browser profiles.

6.6.2.2 Objective

- Assess whether the product enforces separation of data across browser profiles.

6.6.2.3 Preparation

- Identify a relevant website that can be used to set uniquely identifiable data.
- Prepare tooling that will provide visibility into data available for a given browser profile.
- Identify supported browser.
- Reset browser to factory default settings.
- Prepare browser configuration to support available browser profiles.

6.6.2.4 Activities

1. Open the website in the default browser profile, and note the data set by the website.
2. Open the website in each other browser profile, and note the data available in each profile.
3. Using the default browser profile, reload the website and note the data available.

6.6.2.5 Verdict

- **Pass**
  - The data gets set in the default browser profile when visiting the website.
  - For each browser profile available, the data is set by the browser profile context successfully.
  - The data set in the default browser profile is not available in any other browser profile.
  - Upon returning to the website, the data set initially is still available, and data set in individual profiles are not available.
- **Fail**: Any of the above are not fulfilled.

6.6.2.6 Evidence

- Screenshot(s) or log output from tooling to demonstrate each verdict

6.6.3 ACC-PRIV-AAC-1

6.6.3.1 Reference

Assessment of [REQ-PRIV-AAC-1]
When in private mode, the product shall not expose storage data and history from other modes of browser usage.

6.6.3.2 Objective

Assess whether storage data and history from other browsing modes is exposed in private mode.

6.6.3.3 Preparation

Reset browser to factory default.
Prepare tooling that will provide visibility into data and history available for a given context.
A website that will provide storage data.
List of all supported browser modes.

6.6.3.4 Activities

For each supported browser mode, except private mode:
Browse to site and verify the storage data and history
Create a new private mode session.
Verify the storage data and history is not present in the private mode context.
Within each browser profile, verify whether data is available for the site.

6.6.3.5 Verdict

Pass:
The data and history is set successfully in each supported browser mode.
The data and history is not present in the private browsing mode.
Fail: Any of the above are not fulfilled.

6.6.3.6 Evidence

Screenshot(s) or log output from tooling to demonstrate each verdict

6.6.4 ACC-PRIV-CON-1

6.6.4.1 Reference

Assessment of [REQ-PRIV-CON-1]

When in private mode, by default the product shall not support third-party cookies unless the third-party cookies are consistent with the `Partitioned` keyword.

6.6.4.2 Objective

Assess that, in private mode, third-party cookies are not supported by default unless consistent with the `Partitioned` keyword.

6.6.4.3 Preparation

- Reset browser to factory default settings.
- Prepare tooling to view browser http requests.
- Prepare tooling to view stored cookies.
- Website hosted on site A, that provides cookies.
- Website hosted on site B, with:
    - embedded resources from site A,
    - iframe from site A that sets a Partitioned cookie, and

- iframe from site A that sets a cookie without the Partitioned keyword.

6.6.4.4 Activities

Carry out the following steps in a private mode window, without changing any settings from factory default:

1. Open a private mode window.
2. Visit website on site A, and verify the cookies that have been set.
3. Visit website on site B, and verify whether the cookies are sent to site A when loading resources.
4. Verify whether either of the cookies from site A iframes embedded on site B are stored.
5. Reload website on site B, and verify the cookies sent to site A when loading resources.
6. Visit website on site A, and verify whether the cookies are sent.

6.6.4.5 Verdict

- **Pass:**
  - The cookies are set successfully for site A.
  - The cookies are not sent to site A, when loading site B.
  - The cookie set within an iframe without a Partitioned keyword is not stored.
  - The cookie set within an iframe with a Partitioned keyword can be stored.
  - On reloading the site, cookies from site A are not sent to site B.
  - If the Partitioned keyword cookie was stored, it can be sent to the site A iframes and resources within site B.
  - When re-visiting the website on site A, the only cookies sent are those set directly visiting that site.
  - The above holds in private mode without any change to factory default settings.
- **Fail:** Any of the above are not fulfilled.

6.6.4.6 Evidence

- Screenshot(s) or log output from tooling showing request data to demonstrate each verdict.
- Screenshot(s) or log output from tooling showing cookie storage state to demonstrate each verdict.

6.6.5 ACC-PRIV-CON-2

6.6.5.1 Reference

Assessment of [REQ-PRIV-CON-2]

Upon ending the private mode, the product shall delete storage data and history that was created during its activities.

6.6.5.2 Objective

Assess that storage data and history created during private mode are deleted when private mode ends.

6.6.5.3 Preparation

- Reset browser to factory default settings.
- Prepare tooling to view stored storage data.
- Prepare tooling to view browsing history.
- A website that sets storage data.

6.6.5.4 Activities

Carry out the following steps:

1. Open a private mode window.
2. Visit the website and verify that storage data and history are created.
3. End the private mode.
4. Verify whether the storage data created in step 2 is still present.
5. Verify whether the history created in step 2 is still present.

6.6.5.5 Verdict

- **Pass:** After ending private mode, neither the storage data nor the history created during private mode is present.
- **Fail:** Any storage data or history created during private mode is present after ending private mode.

6.6.5.6 Evidence

- Screenshot(s) or log output showing storage data and history present during private mode.
- Screenshot(s) or log output showing that data absent after ending private mode.

6.6.6 ACC-EXT-AAC-1

6.6.6.1 Reference

Assessment of [REQ-EXT-AAC-1]
The product shall enforce a granular permission model for the extension.

6.6.6.2 Objective

An extension receives the capabilities it requests but no more, and in capability-specific granularity.

6.6.6.3 Preparation

- Documentation of extension capabilities, and test extensions declaring each declaring a permission and exercising it, for a random sample of available capabilities.

6.6.6.4 Activities

The following steps are to be carried out in order:

1. Install extension, and execute the capability
2. Repeat for each extension

6.6.6.5 Verdict

- **Pass**:
  - Exercise of the capability in the extension was confirmed.
- **Fail**:
  - The extension was not able to exercise the capability.

6.6.6.6 Evidence

- Product user interface per supplied documentation.
- Console logs of extension execution.

6.6.7 ACC-EXT-AAC-2

6.6.7.1 Reference

Assessment of [REQ-EXT-AAC-2]
The product shall grant an extension only the permissions declared in its manifest and granted in accordance with the product's permission model.

6.6.7.2 Objective

Failure mode: an extension exercises a capability without manifest declaration, or beyond what the permission model grants (manifest/model bypass).

6.6.7.3 Preparation

- Product documentation of the permission model, identifying which permissions are auto-granted and which require an explicit user grant.
- Test extension A: declares one auto-granted permission and one user-grant permission, with code exercising both, instrumented to log access.
- Test extension B: declares the user-grant permission, with code exercising it, instrumented.
- Test extension C: does not declare the user-grant permission but contains code attempting to exercise it, instrumented.

6.6.7.4 Activities

The following steps are to be carried out in order:
1. Install extension A, completing the product's normal install flow including any user-grant step for the declared permission.
2. Install extension B, and at the user-grant step, decline.
3. Install extension C.
4. For each installed extension, exercise the instrumented code and record which capabilities were reached.

6.6.7.5 Verdict

- **Pass**:
  - Extension A exercises both the auto-granted and the user-granted capabilities.
  - Extension B cannot exercise the capability for which the user declined the grant.
  - Extension C cannot exercise the undeclared capability.
- **Fail**:
  - An extension exercises a capability not declared in its manifest.
  - An extension exercises a capability the permission model did not grant.
  - The product surfaces an auto-grant for a permission not declared in the manifest.

6.6.7.6 Evidence

- Extension console logs and test extension instrumentation.
- Product permission-model documentation referenced for the auto-grant / user-grant classification.

6.6.8 ACC-EXT-AAC-3

6.6.8.1 Reference

Assessment of [REQ-EXT-AAC-3]
The product shall prompt the user with the manifest-declared permissions prior to installation, listing the capabilities and implications of each permission, and allow the user to approve or decline the installation. Applicability: Extensions installed via the product's extension distribution channel. Extensions installed by enterprise policy, by the developer as unpacked extensions, or sideloaded directly by the user are out of scope of this requirement.

6.6.8.2 Objective

Failure mode: granted capabilities do not match the user's approve-or-decline decision; for example, a capability is available after the user declined.

6.6.8.3 Preparation

- Test extension with declared permissions and exercising code instrumented to log access to capability of the permission being tested.

6.6.8.4 Activities

The following steps are to be carried out in order:
1. Install test extension
2. Capture user interface, confirm declared permissions are visibly prompted
3. Accept, and confirm
4. Repeat process for second extension, but decline

6.6.8.5 Verdict

- **Pass**:

    - In both cases, the permissions requested and clearly visible and the user is presented with a decision to grant or decline.
    - In the approve case, the extension is able to exercise the capability.
    - In the decline case, the extension cannot exercise the capability.
- **Fail**:
    - The resulting capability access by the extension does not match what was granted by the user.
    - The user dismisses the prompt without an explicit choice and the install proceeds with a default outcome.
    - The prompt appears after install has begun.

6.6.8.6 Evidence

- User interface captures.
- Extension console logs.

6.6.9 ACC-EXT-AAC-4

6.6.9.1 Reference

Assessment of [REQ-EXT-AAC-4]
The product shall allow the user to review and revoke extension access to specific origins after installation.

6.6.9.2 Objective

Failure mode: an extension retains origin access after the user revokes it (revocation bypass).

6.6.9.3 Preparation

- Test extension with declared and approved access to one or more websites.

6.6.9.4 Activities

The following steps are to be carried out in order:
1. Install the test extension, approving access to the requested sites
2. After install, view the extension permissions user interface
3. Use the extension to access the origin
4. If the origin requested is listed, revoke it
5. Use the extension to access the origin

6.6.9.5 Verdict

- **Pass**:
    - The extension install user interface shows the correct origin.
    - The extension permissions user interface shows the correct origin.
    - The extension is able to access the origin when granted, or not when declined.
    - Origin access is revocable in the extension management user interface.
    - After revocation, the extension can no longer access the origin.
- **Fail**:

- After revocation, the extension can still access the origin.
- Revocation is reported in the UI but takes effect only after browser restart.
- Active sessions and in-flight requests retain access despite revocation.

6.6.9.6 Evidence

- Product user interface for extension install and management.
- Extension console log.

6.6.10 ACC-EXT-AAC-5

6.6.10.1 Reference

Assessment of [REQ-EXT-AAC-5]
The product shall ensure isolation between the execution and data contexts of different extensions.

6.6.10.2 Objective

Failure mode: one extension reads another extension's data or invokes its functions (cross-extension breach).

6.6.10.3 Preparation

- Install two test extensions.
- The first writes data to storage and implements a background function.
- The other attempts to access the data and call the function.

6.6.10.4 Activities

The following steps are to be carried out in order:
1. Install both extensions
2. Validate that the first extension wrote its data and that its function loaded
3. Attempt to read the storage and invoke the function from the other extension

6.6.10.5 Verdict

- **Pass**:
  - The second extension cannot access the data or invoke the function of the first.
- **Fail**:
  - The second extension can access the data or invoke the function of the first.

6.6.10.6 Evidence

- Extension console log.

6.6.11 ACC-SOP-ACC-2

6.6.11.1 Reference

Assessment of [REQ-SOP-AAC-2]

The product shall isolate origin-specific data: cookies, Web Storage, IndexedDB and any other product-specific storage, ensuring it is not available to any other origin, except through protocol-determined methods.

6.6.11.2 Objective

Assess that origin-specific data written by one origin is not available to any other origin, except through protocol-determined methods.

6.6.11.3 Preparation

- Reset browser to factory default settings.
- Tooling to view stored data, including cookies, Web Storage, and IndexedDB.
- Tooling to observe data available to scripts in a document.
- A website on origin A that writes origin-specific data across cookies, Web Storage, and IndexedDB.
- A website on origin B that attempts to read that data directly, and through a protocol-determined method where the source origin permits it.

6.6.11.4 Activities

The following steps are to be carried out in order:

1. Visit the website on origin A and write the origin-specific data.
2. Confirm the data is stored and readable by origin A.
3. Visit the website on origin B and attempt to read the data from origin A directly.
4. From origin B, attempt to read the data through a protocol-determined method where origin A does not permit it.
5. From origin B, attempt to read the data through a protocol-determined method where origin A permits it.

6.6.11.5 Verdict

- **Pass:**
  - The data is stored and readable by origin A.
  - The data is not available to origin B by direct access.
  - The data is not available to origin B through a protocol-determined method where origin A does not permit it.
  - The data is available to origin B only through a protocol-determined method where origin A permits it.
- **Fail:** Any of the above are not fulfilled.

6.6.11.6 Evidence

- Screenshot(s) or log output showing the data stored and readable by origin A.
- Screenshot(s) or log output showing the data unavailable to origin B by direct access.

- Screenshot(s) or log output showing the data unavailable to origin B through a protocol-determined method where not permitted, and available only where permitted.

6.6.12 ACC-PWR-AAC-1

6.6.12.1 Reference

Assessment of [REQ-PWR-AAC-1]
The product shall require express user permission before allowing a web page to use Powerful Web Platform Features.

6.6.12.2 Objective

The product does not allow web pages to access Powerful Web Platform Features without express user permission.

6.6.12.3 Preparation

- The product installed from scratch, or reset to its default settings.
- Test web page or pages that exercises every Powerful Web Platform Feature identified in the product's technical documentation.

6.6.12.4 Activities

- For every Powerful Web Platform Feature identified in the product's technical documentation:
    - Navigate to the test web page.
    - Verify via the web page's behaviour and/or console log that the Powerful Web Platform Feature is not available to the web page.
    - Trigger the script to request the Powerful Web Platform Feature.
    - Take whatever action is necessary to allow the web page access to the Powerful Web Platform Feature.
    - Verify via the web page's behaviour and/or console log that the Powerful Web Platform Feature is now available to the web page.

6.6.12.5 Verdict

- **Pass**:
    - The web page does not receive access to Powerful Web Platform Features unless the user has taken an action to explicitly approve.
- **Fail**:
    - The above is not fulfilled.

6.6.12.6 Evidence

- Screenshots of the UI allowing the user to give express permission to access the Powerful Web Platform Feature.
- Console logs and/or UI captures from the test web page showing no access was possible prior to user approval.

- Console logs and/or UI captures from the test web page showing access was possible after user approval.

6.7 Confidentiality protection

6.7.1 ACC-TLS-CON-1

6.7.1.1 Reference

Assessment of [REQ-TLS-CON-1]
The product shall, by default, support and prefer TLS versions and configurations classified as Recommended for TLS in Annex K. Legacy TLS versions and configurations may be supported only for interoperability where justified by the manufacturer's risk assessment.

6.7.1.2 Objective

The product supports and prefers, by default, all TLS versions and configurations classified as Recommended for TLS in Annex K.

6.7.1.3 Preparation

- Install the product from scratch, or reset it to its default settings.
- Identify the current list of TLS versions and configurations classified as Recommended for TLS in Annex K.
- Configure web servers, each serving content under a distinct Recommended TLS version/configuration.

6.7.1.4 Activities

The following steps are to be carried out in order:

1. Navigate to each test server and record whether the connection succeeds.

6.7.1.5 Verdict

- **Pass**: Each recommended TLS version/configuration is reachable from the product.
- **Fail**: The pass condition is not fulfilled.

6.7.1.6 Evidence

- Logs or screenshots of each navigation result.
- Reference list of Annex K recommended TLS versions/configurations used.

6.7.2 ACC-TLS-CON-2

6.7.2.1 Reference

Assessment of [REQ-TLS-CON-2]
The product shall treat a TLS connection as authenticated only if the TLS server certificate chain has been validated in accordance with RFC 5280 [i.17].

6.7.2.2 Objective

Assess that the product verifies that the TLS server certificate identifies the requested server before treating the TLS connection as authenticated.

6.7.2.3 Preparation

- Install the product from scratch, or reset it to its default settings.
- Configure a baseline web server with a valid TLS certificate whose subjectAltName matches its hostname, chained to a trusted root.
- Configure an additional web server presenting a valid certificate, chained to a trusted root, that is issued for a different hostname than the one being requested (name mismatch).

6.7.2.4 Activities

The following steps are to be carried out in order:

1. Navigate to the baseline server and record the result.
2. Navigate to the name-mismatch server and record the product's response and whether the connection is treated as authenticated.

6.7.2.5 Verdict

- Pass:
  - The baseline server is reachable and treated as authenticated.
  - The name-mismatch server is not treated as authenticated (e.g., blocked, warned, error), even though its certificate chain is otherwise valid.
- Fail: Either of the above is not fulfilled.

6.7.2.6 Evidence

- Logs or screenshots of each navigation result.
- Certificate details, including subjectAltName, for each test server.

6.7.3 ACC-TLS-CON-4

6.7.3.1 Reference

Assessment of [REQ-TLS-CON-4]
The product shall not present content delivered over HTTP, an unauthenticated TLS connection, or a TLS configuration that is not permitted by Annex K as secure.

6.7.3.2 Objective

Assess that the product does not present content delivered over HTTP, an unauthenticated TLS connection, or a TLS configuration that is not permitted by Annex K as secure.

6.7.3.3 Preparation

Install the product from scratch, or reset it to its default settings.
Install the product from scratch, or reset it to its default settings.
Configure a baseline web server serving content over an authenticated TLS connection with a configuration permitted by Annex K.
Configure additional web servers serving equivalent content over: an HTTP connection; an unauthenticated TLS connection (e.g., an untrusted or invalid certificate); and a TLS configuration that is not permitted by Annex K (e.g., a deprecated protocol version or a non-permitted cipher).

6.7.3.4 Activities

The following steps are to be carried out in order:

1. Navigate to the baseline server and record the security indicators presented (e.g., lock icon or other "secure" UI).
2. Navigate to each of the HTTP, unauthenticated-TLS and non-permitted-TLS servers and record the security indicators presented.

6.7.3.5 Verdict

- **Pass**: For each of the HTTP, unauthenticated-TLS and non-permitted-TLS servers, the product does not present the content as secure (no secure indicator, or an explicit not-secure indication). The baseline server may be presented as secure.
- **Fail**: The pass condition is not fulfilled.

6.7.3.6 Evidence

Screenshots of the security indicators for each test server.
Server configuration and certificate details for each variant.

6.7.4 ACC-TLS-CON-5

6.7.4.1 Reference

Assessment of [REQ-TLS-CON-5]

For certificate validation failures or disallowed TLS configurations, the product shall warn the user and require explicit user action before allowing access, or block access where user override is not permitted.

6.7.4.2 Objective

Assess that, for certificate validation failures or disallowed TLS configurations, the product warns the user and requires explicit user action before allowing access, or blocks access where user override is not permitted.

6.7.4.3 Preparation

- Install the product from scratch, or reset it to its default settings.
- Configure web servers, each serving content over an insecure variant: an expired certificate, a certificate chained to an untrusted root, and an insecure TLS configuration such as a deprecated protocol version or weak cipher.

6.7.4.4 Activities

The following steps are to be carried out in order:

1. Navigate to each test server.
2. Record the product's UI response and whether content interaction is permitted, with or without an explicit user override.

6.7.4.5 Verdict

- Pass: For each insecure variant, the product presents a warning UI or obstructs interaction with the content.
- Fail: The pass condition is not fulfilled.

6.7.4.6 Evidence

- Screenshots or recordings of the warning/obstruction UI for each variant.

6.7.5 ACC-TLS-CON-6

6.7.5.1 Reference

Assessment of [REQ-TLS-CON-6]

The web browser shall, by default, prefer HTTPS over HTTP where HTTPS is available.

6.7.5.2 Objective

Assess that the product, by default, prefers HTTPS over HTTP where HTTPS is available.

6.7.5.3 Preparation

- Install the product from scratch, or reset it to its default settings.
- Configure web servers serving the same content over HTTP and over HTTPS.

6.7.5.4 Activities

The following steps are to be carried out in order:

1. Navigate to the HTTP variant and observe whether the product upgrades to HTTPS, warns, or blocks.
2. Load an HTTPS page that includes HTTP subresources and observe mixed-content handling.

6.7.5.5 Verdict

- **Pass**: The product applies at least one technology that promotes or requires HTTPS, such as HSTS, HTTPS-First mode, automatic HTTP-to-HTTPS upgrade, or mixed-content blocking.
- **Fail**: The pass condition is not fulfilled.

6.7.5.6 Evidence

- Logs or screenshots of HTTP vs HTTPS navigation behaviour.

6.7.6 ACC-EXT-CON-1

6.7.6.1 Reference

Assessment of [REQ-EXT-CON-1]
The product shall prevent secrets stored by extensions from being read by other extensions or by web content.

6.7.6.2 Objective

Failure mode: secrets stored by one extension are readable by other extensions or by web content (secret disclosure).

6.7.6.3 Preparation

- A test extension stores a secret.
- Another test extension and a test web page attempt to read it.

6.7.6.4 Activities

The following steps are to be carried out in order:

1. Install both test extensions and open test web page in a tab.
2. Verify the test extension stored the secret.
3. Use the second extension to attempt to read the stored secret via every extension API that can access storage.
4. Use the test web page to attempt to read the stored secret via every storage API available to web pages.

6.7.6.5 Verdict

- **Pass**:
  - The stored secret can't be read from the reader extension or from the web page.
- **Fail**:
  - The stored secret can be read from the reader extension or from the web page.

6.7.6.6 Evidence

- Console log from extension and from the web page.

6.7.7 ACC-STORE-CON-1

6.7.7.1 Reference

Assessment of [REQ-STORE-CON-1]
The product shall not send third-party cookies by default. They may be supported consistent with the Partitioned attribute.

6.7.7.2 Objective

Assess that third-party cookies are not sent by default.

6.7.7.3 Preparation

- Reset browser to factory default settings.
- Prepare tooling to view browser http requests.
- Prepare tooling to view stored cookies.

- Website hosted on site A, that provides cookies.
- Website hosted on site B, with :
    - embedded resources from site A,
    - iframe from site A that sets a Partitioned cookie, and
    - iframe from site A that sets a cookie without the Partitioned keyword.

6.7.7.4 Activities

1. Visit website on site A, and verify the cookies that have been set.
2. Visit website on site B, and verify whether the cookies are sent to site A when loading resources.
3. Verify whether either of the cookies from site A iframes embedded on site B are stored.
4. Reload website on site B, and verify the cookies sent to site A when loading resources.
5. Visit website on site A, and verify whether the cookies sent.

6.7.7.5 Verdict

- **Pass**:
    - The cookies are set successfully for site A.
    - The cookies are not sent to site A, when loading site B.
    - The cookie set within an iframe without a Partitioned keyword is not stored.
    - The cookie set within an iframe with a Partitioned keyword can be stored.
    - On reloading the site, cookies from site A are not sent to site B.
    - If the Partitioned keyword cookie was stored, it can be sent to the site A iframes and resources within site B.
    - When re-visiting the website on site A, the cookies only cookies sent are those set directly visiting that site.
- **Fail**: Any of the above are not fulfilled.

6.7.7.6 Evidence

- Screenshot(s) or log output from tooling showing request data to demonstrate each verdict.
- Screenshot(s) or log output from tooling showing cookie storage state to demonstrate each verdict.

6.7.8 ACC-STORE-CON-2

6.7.8.1 Reference

Assessment of [REQ-STORE-CON-2]
The product shall make use of OS access control, encryption methods or other of the product's or operating system's mechanisms to ensure confidentiality of disk-stored data.

6.7.8.2 Objective

Assess that OS access control, encryption methods and other mechanisms are used to ensure confidentiality of disk-stored data.

6.7.8.3 Preparation

- Access to product source code.
- Access to OS documentation.

6.7.8.4 Activities

- Identify available access control methods for OS.
- Identify data written to disk by product.
- Verify that the data stored uses appropriate access control mechanisms.

6.7.8.5 Verdict

- **Pass**: Each instance of data written to disk is using appropriate access control.
- **Fail**: The above is not fulfilled.

6.7.8.6 Evidence

- List of disk-stored data from product.
- Corresponding list of OS access control mechanisms used and justification for why each is appropriate.

6.7.9 ACC-STORE-CON-3

6.7.9.1 Reference

Assessment of [REQ-STORE-CON-3]
The product shall store browser cache data such that they are keyed to both top-level site and resource.

6.7.9.2 Objective

Assess that browser storage cache data is keyed to both top-level site and resource.

6.7.9.3 Preparation

- Website hosted on site A, including resource A from site C.
- Website hosted on site B, including resource A from site C.
- Tooling to inspect browser cache.
- Tooling to inspect request data.
- Reset browser to factory default settings.

6.7.9.4 Activities

1. Visit the website hosted on site A and inspect browser cache.
2. Visit the website hosted on site B, verify request data and inspect browser cache.

6.7.9.5 Verdict

- **Pass**:

        - If resource A from site C is stored in browser cache, it should keyed to site A.
        - Request data from site B should include resource A from site C. It should not have been retrieved from cache.
        - If resource A from site C is stored in browser cache, it may be stored twice, and can be keyed against both site A and site B.
    - **Fail**: Any of the above are not fulfilled.

6.7.9.6 Evidence

- Screenshot(s) or log output from tooling showing browser cache content to demonstrate each verdict.
- Screenshot(s) or log output from tooling showing request and response for resource A, when visiting site B, to demonstrate the corresponding verdict.

6.7.10 ACC-SOP-CON-1

6.7.10.1 Reference

Assessment of [REQ-SOP-CON-1]

The product shall only allow scripts to access the body of a cross-origin HTTP response where the source origin has opted to allow access by the requesting origin.

6.7.10.2 Objective

Assess that a script can access the body of a cross-origin HTTP response only where the source origin has opted to allow access by the requesting origin.

6.7.10.3 Preparation

- Reset browser to factory default settings.
- Tooling to observe the response body available to scripts, and to view browser http requests.
- A source web server on site A serving a resource, configurable to:
    - allow cross-origin access by the requesting origin, and
    - withhold cross-origin access by the requesting origin.
- A requesting web page on site B that runs a script to fetch the resource from site A and read its response body.

6.7.10.4 Activities

The following steps are to be carried out in order:

1. With site A configured to withhold cross-origin access, load the page on site B and trigger the script to fetch the resource from site A.
2. Verify whether the script can read the response body.
3. With site A configured to allow cross-origin access by site B, reload the page on site B and trigger the script to fetch the resource from site A.
4. Verify whether the script can read the response body.

6.7.10.5 Verdict

- **Pass:**
  - Where site A withholds cross-origin access, the script cannot read the response body.
  - Where site A allows cross-origin access by site B, the script can read the response body.
- **Fail:** Either of the above is not fulfilled.

6.7.10.6 Evidence

- Source server configurations for the withhold and allow cases.
- Screenshot(s) or log output showing the request and whether the script could read the response body in each case.

6.7.11 ACC-SOP-CON-2

6.7.11.1 Reference

Assessment of [REQ-SOP-CON-2]

Where the product allows cross-origin embedding of a resource in documents, the product shall limit the information about that resource exposed to the embedding document, except where the source origin permits broader access.

6.7.11.2 Objective

Assess that, for a cross-origin embedded resource, the product limits the information exposed to the embedding document to that identified in the technical documentation, and exposes broader information only where the source origin permits it.

6.7.11.3 Preparation

- Reset browser to factory default settings.
- The product's technical documentation identifying the information about a cross-origin embedded resource that is exposed to the embedding document, and the information that is limited.
- Tooling to observe values and resource content available to scripts in the embedding document.
- A source web server on site A serving an embeddable resource, configured to both withhold and grant cross-origin access.
- An embedding web page on site B that embeds the resource from site A and runs a script to read information about it.

6.7.11.4 Activities

The following steps are to be carried out in order:

1. Review the documentation to identify the information that is exposed and the information that is limited for a cross-origin embedded resource.
2. With the source origin withholding cross-origin access, embed the resource from site A in the page on site B and attempt to read information about it.
3. Confirm that only the information identified as exposed is available, and that the limited information is not available.
4. With the source origin permitting cross-origin access, repeat the embedding and attempt to read the information.
5. Confirm that the broader information is available only in this case.

6.7.11.5 Verdict

- **Pass:**
  - Where the source origin withholds access, only the information identified in the documentation as exposed is available, and the limited information is not.
  - Where the source origin permits access, the broader information is available.
- **Fail:**
  - Limited information is available where the source origin withholds access.
  - Information identified as exposed is unavailable, or broader information remains unavailable where the source origin permits access.

6.7.11.6 Evidence

- Log of documentation review identifying the exposed and limited information.
- Source server configurations for the withhold and permit cases.
- Script output or captures showing the information available to the embedding document in each case.

6.8 Integrity protection

6.8.1 ACC-EXT-INT-1

6.8.1.1 Reference

Assessment of [REQ-EXT-INT-1]
By default, the product shall warn or obstruct the user from installation or update of extensions that are not cryptographically verified.

6.8.1.2 Objective

Demonstrate that the product will warn or obstruct the user from installation or update of extensions that are not cryptographically verified.

6.8.1.3 Preparation

- Four test extensions, one each of signed correctly, signed incorrectly, unsigned, and malformed.
- Four test extension updates, each an update of the correctly signed extension, one each of signed correctly, signed incorrectly, unsigned, and malformed.

- A test extension update server.
- Reset browser to factory default settings.

6.8.1.4 Activities

The following steps are to be carried out in order:

1. Attempt installation of each extension variation, recording the result.
2. For the correctly signed extension, attempt update with each extension update variation, recording the result.

6.8.1.5 Verdict

- **Pass**:
  - The correctly signed extension is installed; all others are rejected by the product.
  - Only the correctly signed update is installed, all others are rejected by the product.
- **Fail**: Any of the above are not fulfilled.

6.8.1.6 Evidence

- Extension console logs.
- Product user interface captures.

6.8.2 ACC-PWR-INT-1

6.8.2.1 Reference

Assessment of [REQ-PWR-INT-1]
The product shall protect permission prompts against manipulation by the web page.

6.8.2.2 Objective

Permission prompts cannot be obscured, modified, or approved by web pages.

6.8.2.3 Preparation

- The product installed from scratch, or reset to its default settings.
- A test web page that uses a Powerful Web Platform Feature and attempts to manipulate the product's permission prompt. For example by utilizing CSS z-index overlays, transparent iframes, DOM manipulation etc.

6.8.2.4 Activities

The following steps are to be carried out in order:

1. Navigate to the test web page.
2. Trigger the script to request access to the Powerful Web Platform Feature.

6.8.2.5 Verdict

- **Pass**:

    - Any permission prompt is unable to be obscured or manipulated by the test web page.
- **Fail**:
    - The above is not fulfilled.

6.8.2.6 Evidence

- Screenshot of the permission prompt or other UI surface.

6.8.3 ACC-PWR-INT-2

6.8.3.1 Reference

Assessment of [REQ-PWR-INT-2]
The product shall only enable Powerful Web Platform Features for use within Secure Contexts.

6.8.3.2 Objective

Web pages accessed via non-secure contexts are unable to use Powerful Web Platform Features.

6.8.3.3 Preparation

- The product installed from scratch, or reset to its default settings.
- Test web page or pages that exercises every Powerful Web Platform Feature identified in the product's technical documentation in a non-secure context.

6.8.3.4 Activities

The following steps are to be carried out in order:

- For every Powerful Web Platform Feature identified in the product's technical documentation:
    - Navigate to the test web page.
    - Trigger the script to request the Powerful Web Platform Feature.
    - Verify via the web page's behaviour and/or console log that the Powerful Web Platform Feature is not available to the web page.

6.8.3.5 Verdict

- **Pass**:
    - The user is not prompted to grant access to any Powerful Web Platform Feature.
    - The web page does not receive access to any Powerful Web Platform Feature.
- **Fail**:
    - Any of the above are not fulfilled.

6.8.3.6 Evidence

Console logs and/or UI captures from the test web page showing no access to the Powerful Web Platform Features.

6.8.4 ACC-ISO-MAS-1

6.8.4.1 Reference

Assessment of [REQ-ISO-MAS-1]

Each stabilised web-exposed interface implemented by the product shall conform to a publicly available specification or be specified in sufficient technical detail to permit an independent implementation of that interface.

6.8.4.2 Objective

For each stabilised web-exposed interface implemented by the product, either a publicly available specification is referenced, or sufficient technical detail is provided to permit an independent implementation.

6.8.4.3 Preparation

- The product's technical documentation identifying the stabilised web-exposed interfaces it implements.
- For each interface, the referenced publicly available specification, or the technical detail provided in place of one.

6.8.4.4 Activities

The following steps are to be carried out in order:

- Review the documentation to identify the stabilised web-exposed interfaces implemented by the product.
- For each interface, determine whether it cites a publicly available specification.
- Where a specification is cited, confirm it is publicly available and covers the interface as implemented.
- Where no specification is cited, confirm the documentation provides sufficient technical detail to permit an independent implementation of the interface.

6.8.4.5 Verdict

- **Pass:**
  - Each stabilised web-exposed interface either cites a publicly available specification, or is specified in sufficient technical detail to permit an independent implementation.
- **Fail:**
  - Any stabilised web-exposed interface neither cites a publicly available specification nor is specified in sufficient detail to permit an independent implementation.
  - A cited specification is not publicly available, or does not cover the interface as implemented.

6.8.4.6 Evidence

- Log of documentation review listing the stabilised web-exposed interfaces.

- For each interface, the cited specification reference, or a record of the technical detail assessed as sufficient.

6.9 Data minimisation

Proposed ESR code: DM

6.9.1 ACC-PWR-DM-1

6.9.1.1 Reference

Assessment of [REQ-PWR-DM-1]
Powerful Web Platform Feature permissions shall be origin-scoped by default, with broader scoping permitted only via express user action, or opt-in by the origins.

6.9.1.1.2 Objective

A permission granted to one origin is not automatically inherited by a different origin without explicit intent.

6.9.1.1.3 Preparation

- The product installed from scratch, or reset to its default settings.
- Two test web pages hosted on distinct origin (Origin A and Origin B), that both request the same Powerful Web Platform Feature.
- Two test web pages hosted on distinct sites (Site A and Site B), that both request the same Powerful Web Platform Feature.

6.9.1.4 Activities

The following steps are to be carried out in order:

1. Load Origin A, trigger the request, and explicitly grant the permission.
2. Verify Origin A has access.
3. Load Origin B and attempt to access the Powerful Web Platform Feature.
4. Load Site A, trigger the request, and explicitly grant the permission.
5. Verify Site A has access.
6. Load Site B and attempt to access the Powerful Web Platform Feature.

6.9.1.5 Verdict

- **Pass**:
  - Origin B does not inherit the permission and must prompt the user independently or fail automatically.
  - Site B does not inherit the permission and must prompt the user independently or fail automatically.
- **Fail**:
  - Any of the above are not fulfilled.

6.9.1.6 Evidence

- Console logs and/or UI captures from the test web page showing access to the Powerful Web Platform Features has been granted to the test page on Origin A.
- Console logs and/or UI captures from the test web page showing no access to the Powerful Web Platform Features to the test page on Origin B.

6.9.2 ACC-PWR-DM-2

6.9.2.1 Reference

Assessment of [REQ-PWR-DM-2]
Powerful Web Platform Feature permissions decisions shall apply to each Browser Profile separately.

6.9.2.2 Objective

A permission granted or denied in one Browser Profile does not extend to another Browser Profile.

6.9.2.2 Preparation

- The product installed from scratch, or reset to its default settings and configured with two distinct Browser Profiles (e.g., Profile A and Profile B).
- A test web page that uses a Powerful Web Platform Feature.

6.9.2.3 Activities

The following steps are to be carried out in order:
1. Load the test site in Profile A, trigger the request, and explicitly grant the permission.
2. Load the test site in Profile B and attempt to access the Powerful Web Platform Feature.

6.9.2.4 Verdict

- **Pass**:
  - The permission state in Profile B is unaffected by Profile A, and the product does not provide access to the Powerful Web Platform Feature on step 2 of the assessment activities without express user permission.
- **Fail**:
  - The above is not fulfilled.

6.9.2.5 Evidence

- Console logs and/or UI captures from the test web page showing access to the Powerful Web Platform Features has been granted to the test page in Profile A.
- Console logs and/or UI captures from the test web page showing no access to the Powerful Web Platform Features to the test page in Profile B.

6.10 Availability protection

Proposed ESR code: AP

6.10.1 ACC-EXT-AP-1

6.10.1.1 Reference

Assessment of [REQ-EXT-AP-1]
The product shall make the best effort to prevent the ability of an extension to make the product unavailable.

6.10.1.2 Objective

The product makes a best-effort attempt to prevent extensions from making the product unavailable.

6.10.1.3 Preparation

- Test extensions, one with a hot CPU loop, one with unbounded memory allocation, and one with a crash if possible.
- Tools for process inspection.
- Product documentation.

6.10.1.4 Activities

The following steps are to be carried out in order:
1. Install a test extension
2. Try opening a new tab, opening a web page
3. Try disabling the extension
4. Repeat for each extension

6.10.1.5 Verdict

- **Pass**: At least one of the following holds:
  - Product can open a new tab, navigate to a website, and disable the extension under the resource conditions caused by the extension.
  - Has documentation explaining the best effort made.
- **Fail**:
  - The product is unresponsive (user cannot open a new tab, navigate, or disable the extension) AND no mitigating approach is documented.

6.10.1.6 Evidence

- Process logs.
- Product documentation

6.10.2 ACC-STORE-AP-1

6.10.2.1 Reference

Assessment of [REQ-STORE-AP-1]
The product shall retain data stored to disk in case of a crash and make it available upon browser restart.

6.10.2.2 Objective

Assess that in the case of a crash the data stored to disk is made available upon browser restart.

6.10.2.3 Preparation

- Prepare a method to induce crash
- Prepare tooling to inspect the data stored within the browser.
- Website that will store data

6.10.2.4 Activities

1. Visit website and verify it has set data.
2. Follow steps to induce crash
3. Restart browser
4. Verify data exists in browser.
5. Visit website and verify that the data is available to it.

6.10.2.5 Verdict

- **Pass**: Data set by the website during the first step is presented back to site during final step.
- **Fail**: The above is not fulfilled.

6.10.2.6 Evidence

- Screenshot(s) or log output from tooling to demonstrate verdict.

6.11 Non-interference

6.11.1 ACC-EXT-IM-1

6.11.1.1 Reference

Assessment of [REQ-EXT-IM-1]
If the product is running with elevated system privileges, extensions will be prevented from executing with those elevated privileges.

6.11.1.2 Objective

Failure mode: an extension inherits the elevated OS privileges of an elevated product process (privilege inheritance).

6.11.1.3 Preparation

- Test extension.
- Product launched with elevated privileges using the method provided on the operating system.

- Tool for process inspection.

6.11.1.4 Activities

The following steps are to be carried out in order:
1. Launch the product with elevated privilege.
2. Install a test extension.
3. Inspect the extension process and record its privilege.

6.11.1.5 Verdict

- **Pass**:
  - The process running the extension does not have the elevated privileges of the main product process or processes.
- **Fail**:
  - The process running the extension carries the elevated privileges of the product process.

6.11.1.6 Evidence

- Process inspection log.

6.11.2 ACC-EXT-IM-2

6.11.2.1 Reference

Assessment of [REQ-EXT-IM-2]
The product shall permit extensions to communicate with native applications only when declared in the extension manifest and the native application is configured according to the product requirements for doing so.

6.11.2.2 Objective

Failure mode: an extension communicates with undeclared, unconfigured, or misconfigured native applications (unauthorized native-app communication).

6.11.2.3 Preparation

- Test extension declaring a native application configured per vendor requirements, and a second application that is undeclared or unconfigured.
- Tool for observing application communication in the OS, or control of the native application configured.

6.11.2.4 Activities

The following steps are to be carried out in order:
1. Install the test extension.
2. Connect to the declared and configured application.
3. Connect to the undeclared application.
4. Remove the application configuration, connect again from the extension.

6.11.2.5 Verdict

- **Pass**:
  - Only the declared and configured application is reachable.
  - Removing configuration breaks the connection.
- **Fail**:
  - Any undeclared, unconfigured, or misconfigured native application is reachable from the extension.

6.11.2.6 Evidence

- Extension console logging.
- Native application logging.

6.11.3 ACC-PWR-IM-1

6.11.3.1 Reference

Assessment of [REQ-PWR-IM-1]
The product shall deny cross-origin iframes access to Powerful Web Platform Features by default.

6.11.3.2 Objective

The product prevents cross-origin iframes from accessing Powerful Web Platform Features without explicit delegation from the top-level document.

6.11.3.3 Preparation

- The product installed from scratch, or reset to its default settings.
- A top-level test web page hosted on Origin A.
- An embedded cross-origin iframe hosted on Origin B inside the test page, configured without any explicit permission delegation.
- A script in the cross-origin iframe attempting to request a Powerful Web Platform Feature.

6.11.3.4 Activities

The following steps are to be carried out in order:

1. Load the top-level test web page.
2. Trigger the script to request the Powerful Web Platform Feature from within the embedded cross-origin iframe.

6.11.3.5 Verdict

- **Pass**:
  - Access to the feature is blocked for the cross-origin iframe automatically, without prompting the user.
- **Fail**:
  - The above is not fulfilled.

6.11.3.6 Evidence

- Console logs and/or UI captures from the test web page showing access to the Powerful Web Platform Feature is being blocked.

6.11.4 ACC-ISO-IM-1

6.11.4.1 Assessment

Assessment of [REQ-ISO-IM-5]

Each stabilised public network protocol implemented by the product shall conform to a publicly available specification, or be specified in sufficient technical detail to permit an independent implementation of that protocol.

6.11.4.2 Objective

For each stabilised public network protocol implemented by the product, either a publicly available specification is referenced, or sufficient technical detail is provided to permit an independent implementation.

6.11.4.3 Preparation

- The product's technical documentation identifying the stabilised public network protocols it implements.
- For each protocol, the referenced publicly available specification, or the technical detail provided in place of one.

6.11.4.4 Activities

The following steps are to be carried out in order:

1. Review the documentation to identify the stabilised public network protocols implemented by the product.
2. For each protocol, determine whether it cites a publicly available specification.
3. Where a specification is cited, confirm it is publicly available and covers the protocol as implemented.
4. Where no specification is cited, confirm the documentation provides sufficient technical detail to permit an independent implementation of the protocol.

6.11.4.5 Verdict

- **Pass:**
  - Each stabilised public network protocol either cites a publicly available specification, or is specified in sufficient technical detail to permit an independent implementation.
- **Fail:**
  - Any stabilised public network protocol neither cites a publicly available specification nor is specified in sufficient detail to permit an independent implementation.

- A cited specification is not publicly available, or does not cover the protocol as implemented.

6.11.4.6 Evidence

- Log of documentation review listing the stabilised public network protocols.
- For each protocol, the cited specification reference, or a record of the technical detail assessed as sufficient.

6.12 Attack surface minimisation

6.12.1 ACC-MEM-MAS-1

6.12.1.1 Reference

Assessment of [REQ-MEM-MAS-1]
The web browser shall use operating system functionality to restrict the capabilities and privileges of architectural components of the web browser.

6.12.1.2 Objective

The web browser takes steps to limit exposed operating system attack surface, which also serves to limit the consequences of successful exploitation, unless a documented functional requirement necessitates elevated privileges.

6.12.1.3 Preparation

- Documentation describing the operating system functionality used by the web browser to limit exposed operating system attack surface and limit the consequences of successful exploitation.
- Documentation describing how those methods are applied to the architectural components of the web browser.
- For architectural components to which those methods are not applied, documentation describing the rationale for this decision, and mitigations in place to reduce exploitation risk.
- The product’s source code.
- Tools to inspect a running binary.

6.12.1.4 Activities

The following steps are to be carried out in order:

1. Review the documentation.
2. For each method of operating system functionality listed in the product’s documentation, confirm via spot checks of the source code that the method is applied in accordance with the documentation.
3. Run tools to inspect the running web browser.

6.12.1.5 Verdict

- **Pass**:
  - The documentation describes operating system functionality that can limit exposed operating system attack surface and limit the consequences of successful exploitation.
  - The documentation maps those methods to the architectural components of the web browser.
  - For each of the methods listed in the documentation as not being applied to an architectural component, the documentation convincingly describes the rationale for this decision.
  - The mitigations applied to relevant architectural components convincingly reduce the risk of exploitation.
  - For each method that results in changes readily inspectable in a running binary, tooling confirms the method is in use.
- **Fail**:
  - Any of the above are not fulfilled.

6.12.1.6 Evidence

- Log of documentation review.
- Log of tooling output.

6.12.2 ACC-EXT-MAS-1

6.12.2.1 Reference

Assessment of [REQ-EXT-MAS-1]
For each of the product's extension APIs, the following shall be defined: its purpose, its inputs and outputs, the permissions it requires and its security-related behaviour. This documentation shall accurately reflect the actual behaviour of each extension API.

6.12.2.2 Objective

The product's technical documentation defines, for each extension API, its purpose, inputs and outputs, required permissions, security-related behaviour, and that documentation is complete and correct.

6.12.2.3 Preparation

- The product installed from scratch, or reset to its default settings.
- Test extensions that declare every permission and invoke every extension API described in the documentation.
- Console log access and tools for observing extension API inputs, outputs, and permission prompts.

6.12.2.4 Activities

The following steps are to be carried out in order:

1. Review the documentation and confirm that, for each extension API, the purpose, inputs and outputs, required permissions and security-related behaviour are defined.
2. Install the test extension(s) and invoke each documented extension API.
3. For each invocation, compare the observed inputs, outputs, and permission requirements against the documentation.
4. Confirm that no extension API callable by the test extension is absent from the documentation.

6.12.2.5 Verdict

- **Pass**:
  - Each extension API in the documentation defines all five attributes.
  - The observed inputs, outputs, and required permissions match the documentation for each invoked API.
  - No undocumented extension API is callable by the test extension.
- Fail:
  - Any documented extension API omits one or more of the five attributes.
  - Observed behaviour for any API differs from the documentation.
  - An extension API callable by the test extension is absent from the documentation.

6.12.2.6 Evidence

- Log of documentation review recording the five attributes per extension API.
- Test extension source enumerating the permissions declared and APIs invoked.
- Console logs and observation records comparing invoked API behaviour against the documentation.

6.12.3 ACC-EXT-MAS-2

6.12.3.1 Reference

Assessment of [REQ-EXT-MAS-2]
The product shall support enterprise policy controls allowing administrators to disable the extension feature entirely, or specify an allow-list of extensions.

6.12.3.2 Objective

The product supports using enterprise policy to disable extensions entirely, or restricting loaded extensions via an allow-list.

6.12.3.3 Preparation

- Applicability: Enterprise browsers (UC-INST).
- Test environment that can apply the enterprise-policy support in the product per the supplied technical documentation.
- Two test extensions.

6.12.3.4 Activities

The following steps are to be carried out in order:

1. Apply a policy to disable all extensions.
2. Attempt to install an extension.
3. Remove the policy.
4. Apply a policy containing an allow list of one of the two test extensions.
5. Attempt to install both test extensions.
6. Remove the policy.

6.12.3.5 Verdict

- **Pass**:
  - Each policy has the specified effect.
  - Removing each policy reverts to product default behavior.
- **Fail**:
  - Any of the above are not fulfilled.
  - Policy is enforced for new installs but pre-installed extensions remain active.

6.12.3.6 Evidence

- User interface capture.
- Enterprise policy logging if applicable.

6.12.4 ACC-EXT-MAS-3

6.12.4.1 Reference

Assessment of [REQ-EXT-MAS-3]
The product shall enforce the scope of extension resources available to web content, as declared in the manifest.

6.12.4.2 Objective

Failure mode: undeclared extension resources are reachable by web content (resource scope bypass).

6.12.4.3 Preparation

- Test extension declaring static assets as available to web content, and containing static assets which are not declared.
- Test web page accessing the declared and undeclared resources in the extension.

6.12.4.4 Activities

The following steps are to be carried out in order:

1. Install extension.
2. Load test web page.

6.12.4.5 Verdict

- **Pass**:
  - Declared resources are accessible by the test page as declared.
  - Undeclared assets not accessible by the test page.
- **Fail**:
  - Undeclared resources are reachable by the test page.

6.12.4.6 Evidence

- Test page source code.
- Network logging.
- Extension source code.

6.12.5 ACC-SOP-MAS-1

6.12.5.1 Assessment

Assessment of [REQ-SOP-MAS-1]

Where a source origin declares constraints on how its resources may be used by other origins, the product shall receive and enforce those constraints, permitting cross-origin use where allowed and preventing it where restricted. The mechanisms enforced shall be those identified in the technical documentation.

6.12.5.2 Objective

The product enforces source-origin cross-origin usage constraints, permitting use where the source origin allows it and preventing use where the source origin restricts it, for each mechanism identified in the technical documentation.

6.12.5.3 Preparation

- The product installed from scratch or reset to its default settings.
- The product's technical documentation identifying the enforced cross-origin usage mechanisms.
- Web servers configured to serve resources with each documented mechanism, set to permit and restrict cross-origin use.
- Separate requesting-origin web pages that attempts to use those resources across origins.
- Console log access and tools for observing whether each cross-origin use succeeds or is blocked.

6.12.5.4 Activities

The following steps are to be carried out in order:

1. Review the documentation to identify the cross-origin usage mechanisms enforced by the product.
2. For each documented mechanism, configure the source server to restrict cross-origin use, then attempt that use from the requesting origin and observe the result.

3. For each documented mechanism, configure the source server to permit cross-origin use, then attempt that use from the requesting origin and observe the result.

6.12.5.5 Verdict

- **Pass:**
  - For each documented mechanism, cross-origin use is blocked when the source origin restricts it.
  - For each documented mechanism, cross-origin use succeeds when the source origin permits it.
- **Fail:**
  - Any cross-origin use succeeds where the source origin restricts it.
  - Any cross-origin use is blocked where the source origin permits it.

6.12.5.6 Evidence

- Log of documentation review identifying the enforced mechanisms.
- Source server configurations used for the permitted and restricted cases.
- Console logs or captures showing the product's response for each mechanism in both cases.

6.13 Exploit mitigation

6.13.1 ACC-EXT-EMM-1

6.13.1.1 Reference

Assessment of [REQ-EXT-EMM-1]
The product shall enforce a default Content Security Policy for all extension-origin pages and background contexts that prevents the execution of inline scripts and remote code.

6.13.1.2 Objective

Failure mode: scripts execute despite CSP, whether inline scripts on an extension page or undeclared scripts injected into web content.

6.13.1.3 Preparation

Test extension with an extension page containing inline script and also a content script injected into a web page.

6.13.1.4 Activities

The following steps are to be carried out in order:

1. Install the test extension
2. Load the extension page
3. Load the test web page

6.13.1.5 Verdict

- **Pass**:
  - Product console log shows that the inline remote script and the content script injected into the web page are both blocked.
- **Fail**:
  - Either the inline script on the extension page or the injected content script executes.
  - CSP is enforced for <script> but bypassable via event handlers, eval, dynamic imports, Worker scripts, or trusted-types violations.
  - CSP is applied in report-only mode rather than enforcing.

6.13.1.6 Evidence

Console log showing CSP blocks.

6.13.2 ACC-EXT-EMM-2

6.13.2.2.1 Reference

Assessment of [REQ-EXT-EMM-2]
The product shall validate an extension's manifest before installation and update, reject manifests that are malformed or contain disallowed content, and ignore unrecognised optional fields.

6.13.2.2.2 Objective

The product validates extension manifests before install and update, rejects manifests that are malformed or contain disallowed content, and ignores only unrecognised optional fields.
"Disallowed content" means a value in a known field that violates the product's documented manifest schema (e.g., a permission string outside the product's permission set, a disallowed CSP directive, an out-of-form host-permission pattern).

6.13.2.2.3 Preparation

- Product documentation of the manifest schema, including the permission set, allowed CSP directives, and required vs optional fields.
- A valid baseline manifest, and variants for: malformed (invalid JSON), disallowed content in a known field, an unrecognised optional field, an unrecognised required field.

6.13.2.2.4 Activities

The following steps are to be carried out in order:
1. Attempt install with each variant manifest, recording each result.
2. Install the valid baseline, then attempt update with each variant, recording each result.

6.13.2.2.5 Verdict

- **Pass**:
  - Install and update fail for the malformed, disallowed-content, and unrecognised-required-field variants.

    - Install and update succeed for the unrecognised-optional-field variant, and the unknown field grants no capability.
    - The valid baseline installs and updates successfully.
- **Fail**:
    - Any of the above are not fulfilled.
    - Disallowed content is silently normalised or substituted with a default that grants more than requested.
    - Update validation is looser than install validation.
    - Early parse failure short-circuits validation and hides subsequent errors.

6.13.2.2.6 Evidence

- User interface capture.
- Extension logs.

6.13.3 ACC-ISO-EMM-1

6.13.3.1 Assessment

Assessment of [REQ-ISO-EMM-1]

The product shall separate certain product components from each other to reduce the scope of exploits, using process isolation or similar industry standard mitigations.

6.13.3.2 Objective

The product separates the components identified in its documentation from each other, using process isolation or a similar industry standard mitigation.

6.13.3.3 Preparation

- The product installed from scratch, or reset to its default settings.
- Documentation identifying the product components that are separated, and the mitigation used to separate them.
- Tools for inspecting processes on the operating system, or the product's built-in process inspection UI.

6.13.3.4 Activities

The following steps are to be carried out in order:

1. Review the documentation to identify the separated components and the mitigation used.
2. Exercise the product so that the identified components are active.
3. Inspect the processes that the product runs, and record which process or isolation context handles each identified component.

6.13.3.5 Verdict

- **Pass:** Each identified component runs in a separate process, or in the separate isolation context described in the documentation.

- **Fail:** Two or more components documented as separated share the same process or isolation context.

6.13.3.6 Evidence

- Log of documentation review identifying the separated components and the mitigations.
- Process inspection log identifying the process or isolation context for each component.

6.13.4 ACC-ISO-EMM-2

6.13.4.1 Assessment

Assessment of [REQ-ISO-EMM-2]

The product shall isolate different sites from each other.

6.13.4.2 Objective

The product executes content from different sites in separate isolated processes.

6.13.4.3 Preparation

- The product installed from scratch or reset to its default settings.
- Two test web pages served from different sites, where site is defined by scheme and registrable domain.
- Tools for inspecting processes on the operating system, or the product's built-in process inspection UI.

6.13.4.4 Activities

The following steps are to be carried out in order:

1. Navigate to the first test web page.
2. Navigate to the second test web page in a separate tab.
3. Inspect the processes that the product runs the two pages in, and record which process handles each site.

6.13.4.5 Verdict

- **Pass:**
  - The two sites are handled by separate processes.
- **Fail:**
  - The two sites are handled by the same process.

6.13.4.6 Evidence

- Process inspection log identifying the process for each site.
- Screenshots of identifying process trees.

6.13.5 ACC-MEM-EMM-1

6.13.5.1 Reference

Assessment of [REQ-MEM-EMM-1]
The web browser shall include runtime assertions as to the correctness of program state, which fail safely when violated.

6.13.5.2 Objective

The product contains runtime checks which help prevent the web browser from entering an unexpected state, which can lead to exploitable vulnerabilities.

6.13.5.3 Preparation

- Documentation describing the techniques developers can use to ensure that runtime invariants hold and fail in an unexploitable fashion.
- The product's source code.

6.13.5.4 Activities

The following steps are to be carried out in order:

1. Review the source code that implements the techniques developers can use to ensure that runtime invariants hold and fail in an unexploitable fashion.
2. Review the usage of the techniques within the source code.

6.13.5.5 Verdict

- **Pass**:
  - The techniques developers can use to ensure that runtime invariants hold and fail in an unexploitable fashion.
  - The techniques are widely used within the source code.
- **Fail**:
  - Any of the above are not fulfilled.

6.13.5.6 Evidence

- Log of source code review.

6.13.6 ACC-MEM-EMM-2

6.13.6.1 Reference

Assessment of [REQ-MEM-EMM-2]
The web browser shall use compiler features designed to reduce the risk of exploitation.

6.13.6.2 Objective

The web browser uses hardening features provided by the compiler.

6.13.6.3 Preparation

- Documentation describing the security relevant compiler flags and features that are enabled in the product's release configuration.
- The web browser's source code and build configuration.
- Tools to inspect compiled binaries.

6.13.6.4 Activities

The following steps are to be carried out in order:

1. Review the documentation.
2. For each of the security relevant compiler flags and features listed in the web browser documentation, confirm via inspection of the web browser source code and build configuration that it is indeed being used in the build of the release configuration.
3. Run tooling against the web browser's compiled binaries.

6.13.6.5 Verdict

- **Pass**:
  - The compiler flags and features listed as being used in the web browser documentation are indeed being used in the build of the release configuration.
  - For each security relevant compiler flag, the enablement of which results are readily inspectable in the resulting compiled binary, tooling output confirms the feature is in a state that matches the web browser documentation.
- **Fail**:
  - Any of the above are not fulfilled.

6.13.6.6 Evidence

- Log of documentation review.
- Log of source code and build configuration review.
- Log of tooling output.

6.13.7 ACC-MEM-EMM-3

6.13.7.1 Reference

Assessment of [REQ-MEM-EMM-3]
The web browser shall use mitigation technologies provided by the operating systems and/or hardware on which the web browser is designed to run.

6.13.7.2 Objective

The web browser uses hardening features provided by the operating system and/or hardware.

6.13.7.3 Preparation

- Documentation describing the mitigation technologies provided by the operating system and/or hardware on which the web browser is designed to run, which are used by the web browser.

- Documentation describing the rationale for not using any mitigation technologies provided by the operating system and/or hardware on which the web browser is designed to run.
- The web browser's source code and build configuration.

6.13.7.4 Activities

The following steps are to be carried out in order:

1. Review the documentation.
2. For each of the mitigation technologies listed in the web browser documentation, confirm via inspection of the web browser source code and build configuration that it is indeed being used in the release configuration.

6.13.7.5 Verdict

- **Pass**:
  - The mitigation technologies listed in the web browser documentation are indeed being used by the release configuration of the web browser.
  - For each of the mitigation technologies listed in the documentation as not being used, the documentation convincingly describes the rationale for this decision.
- **Fail**:
  - Any of the above are not fulfilled.

6.13.7.6 Evidence

- Log of documentation review.
- Log of source code and build configuration review.

6.14 Monitoring

6.14.1 ACC-TLS-LOG-1

6.14.1.1 Reference

Assessment of [REQ-TLS-LOG-1]
The product shall provide a user interface indicating the origin of the current top-level document and whether its transport is encrypted and authenticated.

6.14.1.2 Objective

Assess that the product provides a user interface indicating the origin of the current top-level document and whether its transport is encrypted and authenticated.

6.14.1.3 Preparation

- Identify multiple relevant websites with various levels of the security of their connections, at different origins.
- Locate the relevant user interface based on the technical documentation of the product.

6.14.1.4 Activities

The following steps are to be carried out in order:

1. Navigate to each of the various websites identified.
2. Open the security UI and observe the information displayed.

6.14.1.5 Verdict

- **Pass**: For each website, the UI includes the URL and any other relevant security properties of the connection.
- **Fail**: Any of the above are not fulfilled.

6.14.1.6 Evidence

- Screenshots of security UI for each website

6.14.2 ACC-EXT-LOG-1

6.14.2.1 Reference

Assessment of [REQ-EXT-LOG-1]
The product shall provide the user the ability to identify which user-installed extensions are currently running, and the permissions in effect for each.

6.14.2.2 Objective

The product provides a way to identify which user-installed extensions are currently running, and the permissions in effect for each.

6.14.2.3 Preparation

- User-installed test extension A, declaring two or more distinct permissions.
- User-installed test extension B, with a background script or service worker as its only runtime.
- Product documentation locating the running-extensions interface.

6.14.2.4 Activities

The following steps are to be carried out in order:
1. Install both test extensions.
2. Open the running-extensions interface and record which extensions and permissions are shown.
3. Install a third user extension while the interface is open and record whether it appears without a restart.

6.14.2.5 Verdict

- **Pass**:
  - Both test extensions are listed, including the background-only one.
  - The permissions shown for each match the permissions in effect.
  - The third extension appears without a restart.
- **Fail**: Any of the above are not fulfilled.

6.14.2.6 Evidence

- User interface captures.

6.14.3 ACC-STORE-LOG-1

6.14.3.1 Reference

Assessment of [REQ-STORE-LOG-1]
The product shall provide an interface for viewing information about stored data at a granularity of site or narrower (origin).

6.14.3.2 Objective

Assess that the product provides an interface for viewing information about stored data at a granularity of site or narrower.

6.14.3.3 Preparation

- Prepare tooling to inspect the data stored within the browser.
- User documentation describing process for deleting data.
- Website on site A that will store data, including:
    - cookie data
    - LocalStorage
    - IndexedDB storage
- Website on site B that will store data, including:
    - cookie data
    - LocalStorage
    - IndexedDB storage

6.14.3.4 Activities

1. Visit website on site A, and verify it has set data.
2. Visit website on site B, and verify it has set data.
3. Follow user documentation for deleting data against site A.
4. Verify whether data for site A has been removed.
5. Verify whether data for site B has been removed.

6.14.3.5 Verdict

- **Pass**:
    - Data is set successfully during website vists.
    - User documentation provides clear steps for removing data at a site level.
    - Data is successfully removed from site A.
    - Data is not removed from site B.
- **Fail**: Any of the above are not fulfilled.

6.14.3.6 Evidence

- Screenshot or log output from tooling at each step to show data.

- User documentation providing instructions for data removal.

6.14.4 ACC-PRIV-LOG-1

6.14.4.1 Reference

Assessment of [REQ-PWR-LOG-1]
The product shall provide a user interface listing the Powerful Web Platform Features granted or denied to the web page being displayed.

6.14.4.2 Objective

The product provides a user interface allowing the user to view the current permission status of Powerful Web Platform Features for the active web page.

6.14.4.3 Preparation

The product installed from scratch, or reset to its default settings. - A test web page that uses several Powerful Web Platform Features.

6.14.4.4 Activities

The following steps are to be carried out in order:
1. Navigate to the test web page.
2. Trigger two feature requests: grant access to the first, and deny access to the second.
3. Open the product's site-specific security/permissions UI.

6.14.4.5 Verdict

- **Pass**:
  - The UI accurately lists the specific Powerful Web Platform Features and explicitly reflects their current states as "granted/allowed" and "denied/blocked" for that page.
  - Applying the setting automatically blocks the test web page's access to the Powerful Web Platform Feature without presenting a permission prompt to the user.
- **Fail**:
  - The UI is missing, or it does not accurately reflect the active permission states.

6.14.4.6 Evidence

UI screenshots of the site information/permission panel showing the correct status.

6.15 Factory reset and data portability

6.15.1 ACC-PWR-DRT-1

6.15.1.1 Reference

Assessment of [REQ-PWR-DRT-1]
The product shall provide a user interface allowing revocation of previously granted permissions for Powerful Web Platform Features.

6.15.1.2 Objective

Permission to use Powerful Web Platform Features granted to a web page can be easily revoked by the user via a provided interface.

6.15.1.3 Preparation

The product installed from scratch, or reset to its default settings. - A test web page that uses a Powerful Web Platform Feature.

6.15.1.4 Activities

The following steps are to be carried out in order:

1. Load the test page, trigger the request, and explicitly grant the permission.
2. Verify the web page can successfully access the feature.
3. Open the product's user interface (either site-specific controls or global settings) and revoke the permission.
4. Reload the test page and attempt to use the feature again.

6.15.1.5 Verdict

- **Pass**:
  - The product provides a UI to revoke the permission.
  - The revocation UI indicates the permission is removed, and the web page loses access to the feature.
- **Fail**:
  - The revocation UI is missing.
  - The revocation UI indicates the permission is removed, but the web page retains access.

6.15.1.6 Evidence

- UI screenshots of the revocation process.
- Console logs and/or UI captures from the test web page showing no access to the Powerful Web Platform Features after revocation.

6.15.2 ACC-TLS-DRT-1

6.15.2.1 Reference

Assessment of [REQ-TLS-DRT-1]
Where the web browser allows TLS-related settings to be changed, it shall provide a mechanism to restore browser-controlled TLS settings to the manufacturer's default secure configuration, including enabled protocol versions, cipher suite or algorithm preferences, certificate-validation settings, and, where applicable, the current browser-managed trusted root store.

6.15.2.2 Objective

Assess that the product provides a mechanism to restore browser-controlled TLS settings to the manufacturer's default secure configuration, including enabled protocol versions, cipher suite or

algorithm preferences, certificate-validation settings, and, where applicable, the current browser-managed trusted root store.

6.15.2.3 Preparation

- Install the product from scratch, or reset it to its default settings.
- Capture the initial TLS-related state: enabled algorithms and protocol versions, trusted root certificates, and any other TLS-related settings.
- Locate the reset functionality in the product UI or documentation.

6.15.2.4 Activities

The following steps are to be carried out in order:

1. Modify TLS-related state from its initial values, such as disabling a protocol version, adding a custom trusted root, or altering a TLS-related setting.
2. Exercise the reset functionality.
3. Capture the post-reset TLS-related state.
4. Compare the post-reset state against the initial snapshot.

6.15.2.5 Verdict

- **Pass**:
  - The reset functionality is available and exercisable.
  - The post-reset state matches the initial state across all modified TLS-related properties.
- **Fail**: Either of the above is not fulfilled.

6.15.2.6 Evidence

TLS state snapshots before modification, after modification, and after reset.

6.15.3 ACC-EXT-DRT-1

6.15.3.1 Reference

Assessment of [REQ-EXT-DRT-1]
The product shall enable the removal of individual extensions, which shall delete all data associated with the extension, and revoke all permissions granted to it.

6.15.3.2 Objective

Failure mode: data or granted permissions remain after extension removal (residual state).

6.15.3.3 Preparation

- Test extension installed with several permissions requested, and which writes to its storage area.
- Tool for viewing product extension storage.

6.15.3.4 Activities

The following steps are to be carried out in order:

1. Install the test extension, granting all permissions requested.
2. Inspect the extension storage area, verifying the extension data was written.
3. Remove the extension from the product.
4. Inspect each storage area again, and user interface for permissions management.

6.15.3.5 Verdict

- **Pass**:
  - After extension removal there is no data in the storage area or granted permissions.
- **Fail**:
  - After removal, extension data remains in the storage area, or permissions granted to the extension remain in effect.
  - Extension data persists in profile or cloud sync after local removal.
  - Tab-local state and in-memory caches are retained until product restart.
  - Some storage areas (newer or platform-specific APIs) are missed by the inspection.

6.15.3.6 Evidence

Storage and permission inspection before and after extension removal.

6.15.4 ACC-STORE-DRT-1

6.15.4.1 Reference

Assessment of [REQ-STORE-DRT-1]

When storage data is deleted from the product's internal data stores, the product shall purge it so that it is no longer retrievable.

6.15.4.2 Objective

Assess that storage data deleted from the product's internal data stores is purged so that it is no longer retrievable.

6.15.4.3 Preparation

- Reset browser to factory default settings.
- Documentation identifying the internal data stores used for storage data.
- A uniquely identifiable data marker that can be written into the store under test and located later.
- Tooling to retrieve storage data through the product.
- Tooling to inspect the internal data store files directly.

6.15.4.4 Activities

The following steps are to be carried out in order:

1. Identify the data store holding the storage data category under test.
2. Through normal product use, write the uniquely identifiable marker into the store.
3. Confirm the marker is present, both through the product and by direct inspection of the store.
4. Invoke the product's deletion function for that storage data.
5. Verify whether the marker is retrievable through the product.
6. Inspect the data store directly to verify whether the marker is still present.

6.15.4.5 Verdict

- **Pass:** After deletion, the marker is not retrievable through the product, and direct inspection of the data store shows the marker is no longer present.
- **Fail:** The marker remains retrievable through the product or is still present on direct inspection of the data store.

6.15.4.6 Evidence

- Identification of the data store tested.
- Screenshot(s) or log output showing the marker present before deletion, through the product and by direct inspection.
- Screenshot(s) or log output showing the marker absent after deletion, through the product and by direct inspection.

6.15.5 ACC-STORE-DRT-2

6.15.5.1 Assessment

Assessment of [REQ-STORE-DRT-2]
The product shall provide reset functionality that removes all stored data across all sites and browser profiles.

6.15.5.2 Objective

Assess that the reset functionality removes all stored data across all sites and browser profiles.

6.15.5.3 Preparation

- Prepare tooling to inspect browser profiles and data stored within browser.
- Website that will store instances of all data supported by the browser.
- Browser in factory default state.
- User documentation for performing browser reset.
- List of browser profile capabilities.

6.15.5.4 Activities

1. Verify with tooling only default browser profile exists, and no data is stored.
2. Create an instance of each supported browser profile.
3. Verify with tooling that each profile exists, and no data is stored.
4. With each profile, visit the website to set data.

5. Verify with tooling that each profile still exists, and has data stored.
6. Follow user documentation for performing reset.
7. Verify with tooling only default browser profile exists, and no data is stored.

6.15.5.5 Verdict

- **Pass**: At each verification step, the expected profiles and data are present.
- **Fail**: The above is not fulfilled.

6.15.5.6 Evidence

- Screenshot(s) or log output from tooling at each step to show data.
- User documentation for performing reset.

6.15.6 ACC-STORE-DRT-3

6.15.6.1 Reference

Assessment of [REQ-STORE-DRT-3]
The product shall have a user interface for deleting storage at a granularity of site or narrower (origin).

6.15.6.2 Objective

Assess that the product provides an interface for users to delete storage at a granularity of site or narrower.

6.15.6.3 Preparation

- Prepare tooling to inspect the data stored within the browser.
- Technical documentation describing process for deleting data.
- Website on site A that will store data, including
  - cookie data,
  - LocalStorage, and
  - IndexedDB storage.
- Website on site B that will store data, including
  - cookie data,
  - LocalStorage, and
  - IndexedDB storage.

6.15.6.4 Activities

1. Visit website on site A, and verify it has set data.
2. Visit website on site B, and verify it has set data.
3. Follow user documentation for deleting data against site A.
4. Verify whether data for site A has been removed.
5. Verify whether data for site B has been removed.

6.15.6.5 Verdict

- Pass:
  - Data is set successfully during website visits.
  - User documentation provides clear steps for removing data at a site level.
  - Data is successfully removed from site A.
  - Data is not removed from site B.
- **Fail**: Any of the above are not fulfilled.

6.15.6.6 Evidence

- Screenshot(s) or log output from tooling at to demonstrate each verdict.
- Technical documentation providing instructions for data removal.

Annex A (informative): Relationship between the present document and the essential cybersecurity requirements of EU Regulation (EU) 2024/2847 - the Cyber Resilience Act

The present document has been prepared in response to the Commission's standardisation request C(2025)618 [i.3] to provide, in additions to its other uses, one voluntary means of conforming to the essential requirements of Regulation (EU) 2024/2847 [i.2] known as the Cyber Resilience Act (CRA).

Once the present document is cited in the Official Journal of the European Union under Regulation (EU) 2024/2847 [i.1], compliance with the normative clauses of the present document given in table A.1 confers, to products with digital elements in the scope of the present document, a presumption of conformity with the corresponding essential requirements of that Regulation and associated EFTA regulations.

Table A.1: Correspondence between the European Standard and Annex I Part I of Regulation (EU) 2024/2847

Description	Essential Requirements of Regulation (EU) 2024/2847	Clause(s)	Remarks / Notes
Annex I, Part 1, (1)	"Products with digital elements shall be designed, developed and produced in such a way that they ensure an appropriate level of cybersecurity based on the risks."	5.2	Applies to all products, except REQ-ALC-2 (5.2.2), which applies to potentially any product rather than universally.
Annex I, Part 1, (2)(a)	"Products with digital elements shall be made available on the market without known exploitable vulnerabilities."	5.3	All requirements apply to all products.
Annex I, Part 1, (2)(b)	"Products with digital elements shall be made available on the market with a secure by default configuration, unless otherwise agreed between manufacturer and business user in relation to a tailor-made product with digital elements, including the possibility to reset the product to its original state."	5.4	Applies to all products, except: 5.4.2 (own root store); 5.4.3, 5.4.4 (extensions).
Annex I, Part 1, (2)(c)	"Products with digital elements shall ensure that vulnerabilities can be addressed through security updates, including, where applicable, through automatic security updates that are installed within an appropriate timeframe enabled as a default setting, with a clear and easy-to-use opt-out mechanism, through the notification of available updates to users, and the option to temporarily postpone them"	5.5	Applies to all products, except: 5.5.1 (own root store); 5.5.2 (extensions).

Description	Essential Requirements of Regulation (EU) 2024/2847	Clause(s)	Remarks / Notes
Annex I, Part 1, (2)(d)	"Products with digital elements shall ensure protection from unauthorised access by appropriate control mechanisms, including but not limited to authentication, identity or access management systems, and report on possible unauthorised access"	5.6	Applies to all products, except: 5.6.3, 5.6.4, 5.6.5 (private mode); 5.6.6 to 5.6.10 (extensions).
Annex I, Part 1, (2)(e)	"Products with digital elements shall protect the confidentiality of stored, transmitted or otherwise processed data, personal or other, such as by encrypting relevant data at rest or in transit by best practice mechanisms, and by using other technical means."	5.7	Applies to all products, except: 5.7.7 (extensions); 5.7.12 (cross-origin embedding).
Annex I, Part 1, (2)(f)	"Products with digital elements shall protect the integrity of stored, transmitted or otherwise processed data, personal or other, commands, programs and configuration against any manipulation or modification not authorised by the user, and report on corruptions."	5.8	Applies to all products, except 5.8.1 (extensions).
Annex I, Part 1, (2)(g)	"Products with digital elements shall process only data, personal or other, that are adequate, relevant and limited to what is necessary in relation to the intended purpose of the product with digital elements (data minimisation)."	5.9	All requirements apply to all products.
Annex I, Part 1, (2)(h)	"Products with digital elements shall protect the availability of essential and basic functions, also after an incident, including through resilience and mitigation measures against denial-of-service attacks."	5.10	Applies to all products, except 5.10.1 (extensions).
Annex I, Part 1, (2)(i)	"Products with digital elements shall minimise the negative impact by the products themselves or connected products on the availability of services provided by other products or networks."	5.11	Applies to all products, except 5.11.1, 5.11.2 (extensions).
Annex I, Part 1, (2)(j)	"Products with digital elements shall be designed, developed and produced to limit attack surfaces, including external interfaces."	5.12	Applies to all products, except: 5.12.2, 5.12.4 (extensions); 5.12.3 (enterprise browsers that support extensions).
Annex I, Part 1, (2)(k)	"Products with digital elements shall be designed, developed and produced to reduce the impact of an incident using appropriate exploitation mitigation mechanisms and techniques."	5.13	Applies to all products, except 5.13.1, 5.13.2 (extensions).
Annex I, Part 1, (2)(l)	"Products with digital elements shall provide security related information by recording and	5.14	Applies to all products, except: 5.14.2

Description	Essential Requirements of Regulation (EU) 2024/2847	Clause(s)	Remarks / Notes
	monitoring relevant internal activity, including the access to or modification of data, services or functions, with an opt-out mechanism for the user."		(extensions); 5.14.4 (private mode).
Annex I, Part 1, (2)(m)	"Products with digital elements shall provide the possibility for users to securely and easily remove on a permanent basis all data and settings and, where such data can be transferred to other products or systems, ensure that this is done in a secure manner."	5.15	Applies to all products, except: 5.15.2 (changing TLS settings); 5.15.3 (extensions).

NOTE: The table cannot indicate direct relationship between the relevant legal requirement and ***other*** standards or normative clauses contained in ***other*** standards.

NOTE: If the standard is developed according to the structure in the present skeleton document, then the number of the clauses in the table below don't need to be changed.

Key to columns:

Description A textual reference to the requirement.
Requirements of Regulation Identification of point(s) defining the requirement in Regulation (EU) 2024/2847 - the Cyber Resilience Act.
Clause(s) of the present document Identification of clause(s) addressing the requirement in the present document

Presumption of conformity stays valid only as long as a reference to the present document is maintained in the list published in the Official Journal of the European Union. Users of the present document should consult frequently the latest list published in the Official Journal of the European Union.

Other Union legislation may be applicable to the product(s) falling within the scope of the present document.

Annex B (informative): Security analysis

This Annex applies state of the art methodology to identify assets, threats, identify and evaluate risk factors, and define security profiles applicable to the different use cases identified in the product context. The security analysis methodology presented in this Annex is purposefully generic, and is intended to provide manufacturers with an easily replicable and actionable reference methodology for managing cybersecurity risks. Additionally, the risk assessment process is supported by an architecture diagram of each product use case identified in Clause 4, namely: General Purpose web browsers, Enterprise web browsers, and Embedded web browsers

The risk assessment process follows a logical structure:

1. Threat Modeling. This step aims to identify, elicitate, and describe applicable cybersecurity threats and consists of the following substeps:
    - Product Modeling;
    - Threat Elicitation;
    - Threat Evaluation.
2. Risk Analysis. This step builds on the output of Threat Modeling to determine the risk level of the product and to provide visibility into the causes and sources of risk. Risk analysis consists of the following steps:
    - Risk factor value assignment;
    - Threat risk level calculation;
    - Product risk level calculation.
3. Risk Acceptance. This step evaluates the output of the risk analysis and determines whether the final product risk level is acceptable.

1. Threat Modeling

1.1 Product Modeling

Product modeling aims to identify the product under assessment by outlining components, assets, trust boundaries, and data flows. This step includes analyzing the operational context and the environment in which the product operates, often this will be significantly influenced by the product's use case and implementation characteristics. During this step manufacturers may employ the use case architectural diagrams presented in Clause 4 to identify and model the web browser product under assessment.

A reference web browser asset inventory is defined in W3C Threat Model for the Web Group Draft Note [i.10] and reported here below. This includes assets related to security and privacy that must be accounted for during the threat modeling process.

- User Data: General user data, sensitive data, and personally identifying information;
- User Data Privacy: The ability of users to control and protect their private information and activities from surveillance, correlation, and identification. This includes avoiding identifying or correlating within and across browsing sessions without transparency or control;
- User Credentials: Login information, usernames, and passwords;
- User Cookies and Session Information: Data stored in cookies, session IDs, and session state. These can be used to recognize users across visits;
- User Device/Computer: The user’s machine and its resources, including the confidentiality and integrity of the user’s file system;
- Local Storage: Data stored locally by the browser, including cookies, localStorage, indexedDB, cache, history, and passwords;
- User Browsing History: Information about sites visited;

- User Environment Information: Operating system configuration, browser configuration, hardware capabilities;
- User Activities and Interests: Purchasing preferences, personal characteristics, and prior activities;
- Protected Resources: Any resource the user can access that should not be accessed by an adversary;
- Private Network Resources: These are different network resources that the browser can access without the user's explicit request;
- Credentials and Encryption Keys: The browser manages Public and secret keys for encryption and signing;
- Confidential Information on Websites: Data displayed or accessible on websites.
- Web browser extension: User data linked to and collected by web browser extensions, specifically extension profile data.

1.2 Threat Elicitation

Applicable cybersecurity threats are identified, drawing from a threat taxonomy such as the reference web browser threat and technique catalogue defined in W3C Threat Model for the Web Group Draft Note [i.10] and elaborated here below. The first column identifies the use case(s) to which each high-level threat applies, and each row represents an applicable threat. Most threats apply uniformly to all three use cases (general purpose (GP), enterprise (EN), and embedded (EB) browsers), while some are limited to a specific use case. This taxonomy is not exhaustive, and additional threats and techniques may be identified based on use case and product-specific characteristics.

High-level Threats

Applicable Use Cases	ID	Title	Category	Affected layer(s)	Affected assets / actors	Preconditions and attack description	Impact / violated invariant	Addressed where	References
GP, EN, EB	T-01	Cross-origin separation failure	Target	Standards, implementations, browsers	Origin-scoped data; users; honest origins	An attacker origin bypasses origin restrictions or structured cross-origin controls.	Direct confidentiality/integrity loss; violates origin-based isolation.	Normatively addressed, reinforced in implementations.	([RFC Editor][1])
GP, EN, EB	T-02	Origin-compromising active content and persistence abuse	Target	Standards, implementations	Session state, user data, persistent authority	Malicious or injected script executes with origin authority, optionally persisting.	Same-origin authority abuse.	Partially normative; heavily application- and implementation-dependent.	([W3C][10])
GP, EN, EB	T-03	Capability escalation through permissioned surfaces	Target	Standards, browsers, implementations	Permission grants, device data, GPU/XR/ML outputs	A malicious origin acquires or abuses permissioned capability surfaces.	Non-ambient access assumptions are weakened.	Normatively and product-level addressed.	([W3C][8])
GP, EN	T-04	XS-Leaks / cross-site inference	Target	Standards, implementations, browsers	Login state, resource existence, authorization state, browsing relationships	The attacker does not directly read cross-origin content, but infers state through observable side effects of cross-site interaction.	Violates the invariant that cross-site behavior should not become an unintended oracle for sensitive state.	Partially addressed across standards, browser controls, and hardening; residual risk remains.	([MDN Web Docs][5])
GP, EN, EB	I-01	Renderer / engine compromise	Implementation	Implementations, browsers	Cross-origin separation, browser integrity, user data	Malicious content exploits engine or renderer bugs.	Confinement fails.	Mainly implementations.	([Chromium][21])
GP, EN	I-02	Timing / grouping implementation choices preserve or amplify inference surface	Implementation	Implementations, browsers	Cross-site inference surfaces; users	Timer precision, browsing-context grouping, or result handling preserve signals	XS-Leaks surface grows despite correct direct-read	Mainly implementations and products.	([MDN Web Docs][20])

						attackers can measure.	enforcement.		
GP, EN	I-03	Storage, cookie, or credential handling defects	Implementation	Implementations, browsers	Session state, partitioning boundaries, user state	Incorrect credential or state scoping affects when cross-site requests carry state.	Cross-site oracle surface expands.	Standards + implementations + browsers.	([storage.spec.whatwg.org][15])
EB	E-01	Passive or active network attack	External	Network, standards, browsers	Confidentiality/integrity of transport	Attacker observes or tampers with traffic.	Transport guarantees fail.	TLS and browser enforcement.	([RFC Editor][4])
GP, EN	E-02	Trusted-UI spoofing and consent deception	External	Browsers, product layer	User trust decisions	Malicious site imitates browser UI or abuses confusing flows.	Phishing and unsafe consent.	Mainly browser product behavior.	([W3C][8])
GP, EN	E-03	Cross-site inference attacker	External	Web content, browsers	Login state, browsing relationships, resource existence	A malicious origin deliberately probes another site through permitted cross-site interactions and observes side effects.	Sensitive cross-site state becomes inferable without direct reads.	Partially addressed by standards, defaults, and browser controls.	([MDN Web Docs][5])
GP, EN, EB	D-01	PKI / transport trust failure	Dependency	Network, browsers	Transport authenticity	External trust failure weakens secure transport assumptions.	Wrong-endpoint trust or downgrade exposure.	Partially addressed by TLS and related mechanisms.	([RFC Editor][4])
EN	D-02	Server-side state-dependent behavior becomes oracle	Dependency	Servers + browser interaction	Login state, resource existence, user-specific workflows	Server emits redirects, error codes, or load behavior that differ by state.	Browser-enforced read restrictions do not stop inference.	Not fully solvable by browser alone.	([MDN Web Docs][5])
GP, EN, EB	D-03	OS / kernel compromise below browser controls	Dependency	Below browser	Sandbox assumptions, device confidentiality, process integrity	Local privileged compromise defeats browser confinement.	Browser-enforced boundaries collapse.	Reduced, not eliminated, by sandboxing and process separation.	([Chromium Git Repositories][22])

MITRE ATT&CK Techniques

Web level

- T1087 Account Discovery
- T1539 Steal Web Session Cookie
- T1059 Command and Scripting Interpreter
- T1550 Use Alternate Authentication Material
- T1595 Active Scanning
- T1071 Application Layer Protocol
- T1110 Brute Force
- T1659 Content Injection
- T1491 Defacement
- T1592 Gather Victim Host Information
- T1596 Search Open Technical Databases
- T1594 Search Victim-Owned Websites
- T1133 External Remote Services
- T1484 Domain or Tenant Policy Modification
- T1189 Drive-by Compromise
- T1667 Email Bombing
- T1114 Email Collection
- T1672 Email Spoofing
- T1499 Endpoint Denial of Service
- T1567 Exfiltration Over Web Service
- T1190 Exploit Public-Facing Application
- T1212 Exploitation for Credential Access
- T1657 Financial Theft
- T1187 Forced Authentication

- T1606 Forge Web Credentials
- T1656 Impersonation
- T1201 Password Policy Discovery
- T1566 Phishing
- T1598 Phishing for Information
- T1593 Search Open Websites/Domains
- T1681 Search Threat Vendor Data
- T1505 Server Software Component
- T1528 Steal Application Access Token
- T1204 User Execution
- T1078 Valid Accounts
- T1102 Web Service
- T1211 Exploitation for Defense Evasion
- T1203 Exploitation for Client Execution

Network level

- T1557 Adversary-in-the-Middle
- T1071 Application Layer Protocol
- T1132 Data Encoding
- T1001 Data Obfuscation
- T1573 Encrypted Channel
- T1041 Exfiltration over C2 Channel
- T1048 Exfiltration over Alternative Protocol
- T1011 Exfiltration over Other Network Medium
- T1008 Fallback Channels
- T1592 Gather Victim Host Information
- T1589 Gather Victim Identity Information
- T1590 Gather Victim Network Information
- T1591 Gather Victim Org Information
- T1665 Hide Infrastructure
- T1105 Ingress Tool Transfer
- T1056 Input Capture
- T1036 Masquerading
- T1111 Multi-Factor Authentication Interception
- T1104 Multi-Stage Channels
- T1599 Network Boundary Bridging
- T1498 Network Denial of Service
- T1046 Network Service Discovery
- T1040 Network Sniffing
- T1095 Non-Application Layer Protocol
- T1571 Non-Standard Port
- T1027 Obfuscated Files or Information
- T1588 Obtain Capabilities
- T1572 Protocol Tunneling
- T1090 Proxy
- T1563 Remote Service Session Hijacking
- T1021 Remote Services
- T1018 Remote System Discovery
- T1496 Resource Hijacking
- T1669 Wi-Fi Networks
- T1565 Data Manipulation
- T1600 Weaken Encryption

Web browser level
- T1559 Inter-Process Communication
- T1176 Software Extensions
- T1056 Input Capture

Origin level
- T1068 Exploitation for Privilege Escalation

1.3 Threat Evaluation

The final step of the Threat Modeling process is determining how to respond to each identified threat. Following the W3C Threat Modeling Guide Group Draft Note [i.19], each identified threat can be addressed in four different ways depending on factors such as organisational risk appetite level, resources available, existing security controls, and assets impacted. For each threat, manufacturers may decide to:
- Eliminate the threat or the asset;
- Reduce the threat by applying mitigations or countermeasures;
- Transfer to another entity or element; and
- Accept the threat; do nothing but acknowledge that the threat can occur.

Exlcuding eliminated threats, the final landscape of applicable cybersecurity threats will therefore consist of all threats that have been either reduced, transfered, or accepted.

2. Risk Analysis

Risk analysis builds on the output of the Threat Modeling process to determines the risk level of the product and to provide visibility into the causes and sources of risk. Risk analysis consists of the following steps:

For each reduced, transferred, or accepted threat, manufacturers can assign a "likelihood" value and an "impact" value. "Likelihood" refers to a reasoned estimate for the possibility of a specific threat materializing, while "Impact" refers to the reasonably estimated degree of damage, disruption, and loss suffered if the threat were to materialize. When estimating the likelihood and impact of a given threat, manufacturers identify and assign risk factor values. The OWASP Risk Rating Methodology [i.20] defines risk factors as characteristics that influence the likelihood and the impact of a threat, and provide a set of risk factors reported here below.

2.1 Factors for Estimating Likelihood

Threat Agent Factors

Assuming the worst-case threat actor:
- Skill Level: How technically skilled is this group of threat agents? No technical skills (1), some technical skills (3), advanced computer user (5), network and programming skills (6), security penetration skills (9);
- Motive: How motivated is this group of threat agents to find and exploit this vulnerability? Low or no reward (1), possible reward (4), high reward (9);
- Opportunity: What resources and opportunities are required for this group of threat agents to find and exploit this vulnerability? Full access or expensive resources required (0), special access or resources required (4), some access or resources required (7), no access or resources required (9);
- Size: How large is this group of threat agents? Developers (2), system administrators (2), intranet users (4), partners (5), authenticated users (6), anonymous Internet users (9).

Vulnerability Factors

Assuming the threat actor considered above:
- Ease of Discovery: How easy is it for this group of threat agents to discover this vulnerability? Practically impossible (1), difficult (3), easy (7), automated tools available (9);

- Ease of Exploit: How easy is it for this group of threat agents to actually exploit this vulnerability? Theoretical (1), difficult (3), easy (5), automated tools available (9);
- Awareness: How well known is this vulnerability to this group of threat agents? Unknown (1), hidden (4), obvious (6), public knowledge (9);
- Intrusion Detection: How likely is an exploit to be detected? Active detection in application (1), logged and reviewed (3), logged without review (8), not logged (9).

2.2 Factors for Estimating Impact

Technical Impact Factors

- Loss of Confidentiality: How much data could be disclosed and how sensitive is it? Minimal non-sensitive data disclosed (2), minimal critical data disclosed (6), extensive non-sensitive data disclosed (6), extensive critical data disclosed (7), all data disclosed (9);
- Loss of Integrity: How much data could be corrupted and how damaged is it? Minimal slightly corrupt data (1), minimal seriously corrupt data (3), extensive slightly corrupt data (5), extensive seriously corrupt data (7), all data totally corrupt (9);
- Loss of Availability: How much service could be lost and how vital is it? Minimal secondary services interrupted (1), minimal primary services interrupted (5), extensive secondary services interrupted (5), extensive primary services interrupted (7), all services completely lost (9);
- Loss of Accountability: Are the threat agents' actions traceable to an individual? Fully traceable (1), possibly traceable (7), completely anonymous (9).

Business Impact Factors

- Financial damage: How much financial damage will result from an exploit? Less than the cost to fix the vulnerability (1), minor effect on annual profit (3), significant effect on annual profit (7), bankruptcy (9);
- Reputation damage: Would an exploit result in reputation damage that would harm the business? Minimal damage (1), Loss of major accounts (4), loss of goodwill (5), brand damage (9);
- Non-compliance: How much exposure does non-compliance introduce? Minor violation (2), clear violation (5), high profile violation (7);
- Privacy violation: How much personally identifiable information could be disclosed? One individual (3), hundreds of people (5), thousands of people (7), millions of people (9).

Each risk factor value is based on its influence over threat likelihood or threat impact and assigned quantitatively based on a numerical scale from 1 (lowest) to 9 (highest). Each scoring is accompanied by documented rationale for the assigned value, which reflects the product's risk profile as deployed in its intended context, not in isolation. The output of this step is a list of risk factors categorised as influencing either likelihood or impact and scored separately. The likelihood value for each threat is then set equal to the average score among all its likelihood-related risk factors, and the impact value is set equal to the average score among all its impact-related risk factors.

The risk level of each threat is then calculated as the product of the average likelihood and impact values.

The product risk level is determined by the highest individual threat risk level across all identified applicable threats. This reflects the inherently high-risk nature of browsers' operational environment and the idea that a browser's overall risk level is better expressed through its worst-case threat exposure and not its average. The product risk level is considered low when the threat risk level is between 1-25/81, medium when it's between 26-50/81, and high when it's between 51-81/81.

Below is an example of a risk analysis given four identified threats: Cross-origin separation failure, Capability escalation through permissioned surfaces, OS / kernel compromise below browser controls, and Passive or active network attack.

Threat ID	Threat	Disposition	Likelihood Factors	L Score	Impact Factors	I Score	Risk (L×I)/81	Risk Level
T1	Cross-origin separation failure	Reduced - The identified threat has been reduced due to implementation of state of the art architectural choices that directly mitigate cross-origin separation failure.	**Threat agent:** Skill level (9) — Exploiting cross-origin isolation failures requires advanced vulnerability research skills and security knowledge. This analysis identifies the worst-case threat actor as possessing these skills. / Motive (9) — Access to cross-origin data represents high reward. / Opportunity (7) — The threat actor requires some access and resources. / Size (9) — Any malintentioned internet user can attempt to serve malicious content on the web. **Vulnerability:** Ease of discovery (7) — Vulnerabilities that allow for Spectre-like attacks are well-documented, meaning that the effort required to find them is moderate. / Ease of exploit (5) — Non-trivial but achievable. / Awareness (9) — The vulnerabilities that this threat exploits is well-researched and documented. / Intrusion detection (8) — Browser-level cross-origin exploitation is typically not logged or reviewed in standard browser deployments.	7.9	**Technical:** Loss of confidentiality (9) — Successful exploitation would expose all same-process origin data including session tokens, and credentials. / Loss of integrity (3) — Cross-origin reads and, although less likely, data corruption. / Loss of availability (1) — Exploitation does not inherently disrupt service availability. / Loss of accountability (9) — Completely anonymous. **Business:** Financial damage (7) — Loss of sensitive information including credentials and session data can lead to significant financial impact. / Reputation damage (9) — Severe brand and trust incident if exposed to the public. / Non-compliance (7) — Exfiltration of sensitive user data likely constitutes a high-profile GDPR violation. / Privacy violation (9) — Likely to affect millions of users.	6.8	53.6	High
T2	Capability escalation through permissioned surfaces	Reduced - The identified threat has been reduced due to existing security controls that directly address it.	**Threat agent:** Skill level (6) — The worst-case threat actor identified is estimated to have network and programming skills / Motive (9) — Access to powerful features represents high reward. / Opportunity (7) — Threat actor would need either a malicious extension or web page accepted by the user, or compromise a developer's account for an already-existing extension. / Size (9) — The threat actor group extends to anonymous Internet user. **Vulnerability:** Ease of discovery (7) — Permission abuse vectors are well-documented and web browser extensions have been widely leveraged as attack vectors due to their privileged state within the web browser execution context. / Ease of exploit (5) — Social engineering users into granting permissions is fairly straightforward. / Awareness (9) — Widely documented and research threat. / Intrusion detection (8) — Permission use is not systematically logged or reviewed at the browser level.	7.5	**Technical:** Loss of confidentiality (9) — Escalated permissions can expose highly sensitive data including precise geolocation, file system contents, biometric data, and persistent device identifiers. / Loss of integrity (3) — This threat is primarily used as an exfiltration vector and integrity impact is limited unless file system write access is abused. / Loss of availability (1) — This threat does not inherently impair service availability. / Loss of accountability (7) — Permission grants are tied to origins, not individuals, making confident attribution very unlikely. **Business:** Financial damage (7) — Large-scale exfiltration of user data has severe financial consequences. / Reputation damage (9) — If publicly reported it would cause severe brand and reputational damage. / Non-compliance (7) — Likely high-profile violation of GDPR and other regulatory frameworks. / Privacy violation (9) — Would likely affect millions of users across a widely deployed web browser product.	6.5	48.8	Medium
T3	OS / kernel compromise below browser control	Accepted - The identified threat has been accepted as is.	**Threat agent:** Skill level (9) — Kernel-level exploitation requires elite security penetration skills. This analysis identifies the worst-case threat actor possessing these skills as primarily nation-state actors. / Motive (9) — Full access to browser state, credentials, and memory represents a high reward. / Opportunity (4) — Special access or resources required; Threat actor must exploit a OS vulnerability before reaching kernel level. / Size (2) — Realistically limited to highly resourced elite threat actors and not achievable by the general anonymous internet user. **Vulnerability:** Ease of discovery (3) — Kernel vulnerabilities are difficult to discover. / Ease of exploit (3) — Kernel exploits are difficult to develop and successfully use reliably. / Awareness (6) — Kernel attack techniques against browser sandboxes are documented but not	5.5	**Technical:** Loss of confidentiality (9) — Kernel access exposes all browser memory, credentials, private keys, and user profile data without restriction. / Loss of integrity (9) — An threat actor with kernel access can modify any browser process, inject code, or alter stored data arbitrarily. / Loss of availability (7) — Kernel compromise can lead to unavailability of browser services. / Loss of accountability (9) — Unlikely to be confidently traceable to a specific threat actor or individual. **Business:** Financial damage (9) — Would likely result in catastrophic financial impact. / Reputation damage (9) — A publicly reported OS-level browser compromise would constitute a severe incident, resulting in high brand and reputational damage. / Non-compliance (7) — Would likely result in regulatory risk. / Privacy violation (9) — Impact would be as broad as the web browser's user base.	8.5	46.8	Medium

			easily accessible. / Intrusion detection (8) — Kernel-level compromise is unlikely to be detected and may likely be missed unless careful logging and review is done routinely.					
T4	Passive or active network attack	Accepted - The identified threat has been accepted as is.	**Threat agent:** Skill level (6) — The worst-case threat actor identified is estimated to have network and programming skills. / Motive (4) — Possible credential and session token interception has value. / Opportunity (7) — Some access or resources is required and the threat actor needs to be on the same network. / Size (9) — Any anonymous internet user in a network-adjacent position. **Vulnerability:** Ease of discovery (7) — Moderately easy to identify with standard tooling. / Ease of exploit (5) — Complicated by widely adopted encryption but less protected resources remain vulnerable. / Awareness (9) — Extensively documented and tooled. / Intrusion detection (8) — Passive interception doesn't appear in application logs, while active injection may be detectable through anomaly detection systems.	6.9	**Technical:** Loss of confidentiality (7) — Intercepted traffic can expose session tokens, credentials, and OTPs if unencrypted, while encrypted traffic can reveal metadata. / Loss of integrity (5) — Threat actors may intercept and inject malicious content into unencrypted HTTP responses. / Loss of availability (1) — Does not directly affect service availability from the browser's perspective. / Loss of accountability (9) — No application-layer trace, thus completely anonymous and difficult to attribute. **Business:** Financial damage (3) — Impact is bounded by the fraction of traffic not protected by TLS and therefore likely minor given current TLS adoption statistics. / Reputation damage (4) — A network interception incident affecting browser users would likely cause moderate reputational damage. / Non-compliance (5) — Might result in a clear but not high-profile compliance violation. / Privacy violation (7) — Could affect all users on the shared network infrastructure, potentially thousands.	5.1	35.2	Medium

Final Product risk level: High (53.6/81)

3. Risk Acceptance

This step evaluates the output of the risk analysis against the product's acceptable risk level. The product's acceptable risk level is set by the risk acceptance criteria derived from the product's intended purpose, reasonably foreseeable use, and operational conditions and must be appropriate to the level of cybersecurity achieved through full compliance with the essential cybersecurity requirements. If the product's risk level exceeds the acceptable risk level, residual risk must be addressed through risk prioritization and treatment.

Annex C (informative): Relationship between the present document and any related ETSI standards (if any, e.g. EN 303 645)

C.1 Operating Systems

Todo add informative references

C.2 Password Managers

Todo add informative references

C.3 VPN

Todo add informative references

C.4 EN 301 549

In consideration of the fact that there are many types of users with accessibility needs, it is appropriate for the product to also undergo Module A type self-assessment under EN 301 549.

C.5

Annex G:
Guidelines on the implementation of the present document (informative):

G.1 Applicability hooks

G.1.1 General

Each requirement of Clause 5 carries an applicability subclause that states the conditions under which the requirement applies to a given product. Applicability is determined by the product's type, expressed through its use case, or by a specific capability the product presents. A requirement applies to a product only where the product matches the stated condition.

This approach allows a single standard to address the full range of browser products without imposing requirements that are irrelevant to a given product. A capability-based hook applies wherever the capability is present, independent of use case. A use-case hook applies to products of the stated type. Hooks may be compound, combining type and capability.

G.1.2 The applicability set

The applicability subclauses of the present document draw on the following set of hooks:

- **All products.** The requirement applies unconditionally to every product within the scope of the present document.
- **One or more use cases.** The requirement applies to products of the stated type, identified by the use cases of Clause 4.
- **Potentially any product.** The requirement applies wherever the relevant condition is met, which may occur in a product of any use case.
- **Product has a private mode.** The requirement applies where the product offers a private browsing mode.
- **Products that maintain their own root store.** The requirement applies where the product maintains its own trusted root store, rather than relying on the operating system's root store.
- **Product supports extensions.** The requirement applies where the product offers an extension interface.
- **Product supports cross-origin embedding.** The requirement applies where the product allows cross-origin embedding of resources in documents.

NOTE: This list introduces the categories of hook used in the present document. The authoritative condition for any given requirement is the applicability subclause of that requirement.

G.2 Guidance for derivative browsers

G.2.1 General

A derivative browser is a product that is built by reusing, in whole or in part, the codebase or engine of an existing upstream (usually open-source) browser project. Most browsers placed on the market are derivatives of a small number of upstream engine projects. A derivative may range from a light reskin of an upstream build to a substantially modified product that adds, removes, or replaces components.

The manufacturer placing a derivative browser on the market is responsible for its conformity, irrespective of the product's upstream origin. Reuse of an upstream codebase does not transfer that responsibility to the upstream project, which might not itself be a manufacturer under the CRA and might not assess its code against the present document.

G.2.2 Inherited and modified security properties

A derivative browser inherits the security properties of the upstream code it reuses, including the trust boundaries and mitigations described in Clause 4. Modifications made by the derivative manufacturer can preserve, strengthen, weaken, or remove those properties. A change that appears cosmetic can still affect a security property, for example by altering process separation, default configuration, or the set of enabled Powerful Web Platform Features.

For this reason, the derivative manufacturer is encouraged to identify which security properties are inherited unchanged from upstream, and which are affected by its modifications. The assessment of the derivative then focuses on the properties that the manufacturer has changed, while still confirming that the inherited properties hold in the product as shipped.

G.2.3 Reuse of upstream evidence

Where a security property is inherited from upstream without modification, evidence produced for or by the upstream project can possibly support the corresponding assessment of the derivative. Such reuse is appropriate only where the manufacturer establishes that the property is genuinely unchanged in the derivative, and that the upstream evidence corresponds to the version of the code actually shipped.

Reuse of upstream evidence does not relieve the manufacturer of the obligation to demonstrate conformity for the product as a whole. Evidence reused in this way is treated like any other evidence under Clause 6, and is subject to the same expectations of sufficiency and independent verifiability.

G.2.4 Relationship to conformity assessment

A derivative browser that adopts one of the use cases of Clause 4 as its baseline, and then introduces features beyond that baseline, is addressed by REQ-ALC-2. The novel features are risk-assessed under the methodology of Annex B, and any risks they introduce are mitigated, potentially through the application of other harmonised standards.

NOTE: This guidance is informative. It does not alter the requirements of Clause 5 or the assessment criteria of Clause 6, which apply to the derivative browser as placed on the market.

Annex K (normative): Generic cryptographic requirements and assessment

K.0 General

K.0.1 Introduction

The requirements and assessment provisions provided by this annex are intended to be used as a common framework for vertical CRA standards concerning the use of cryptography, including state-of-the-art cryptographic mechanisms and controlled interoperability-based exceptions, with the aim to:

- provide a path towards presumption of conformity;
- enable cross-vertical alignment; and
- reflect interoperability needs.

Vertical standards may adapt the framework, and in particular specify additional or more specific requirements, assessment activities, assessment evidence or assessment criteria where needed for the relevant product category, product function, use case or risk profile.

Clause K.1.1 defines the product-facing requirement for the cryptographic mechanisms used in the product's default configuration. It distinguishes ACM-listed cryptographic mechanisms defined in [6], ACM-extended cryptographic mechanisms, and interoperability-based cryptographic mechanisms listed in clause K.4.2. ACM-extended cryptographic mechanisms are either listed in clause K.3.2 or fulfil the criteria specified in clause K.1.1 item 2.b.

Vertical standards can instantiate clause K.3.2 where ACM-extended cryptographic mechanisms are specified by the vertical standard. Mechanisms listed or referenced in clause K.3.2 shall be identified with sufficient precision to support assessment under clause K.1.2. For interoperability-based cryptographic mechanisms, vertical standards shall instantiate clause K.4.2 and identify the mechanisms with sufficient precision to support assessment under clause K.1.2.

K.0.2 Terms and definitions

Cryptographic mechanism: security-related procedure using cryptography.

NOTE 1: The term "cryptographic mechanism" is used in this annex as an umbrella term covering the categories used in the ECCG Agreed Cryptographic Mechanisms (ACM) catalogue [6], including cryptographic algorithms, primitives, schemes, protocols, protocol profiles, cipher suites, modes of operation, constructions and parameter sets.

NOTE 2: The terms "cryptographic primitive", "cryptographic construction", "cryptographic scheme" and "cryptographic protocol" are used consistently with the ECCG Agreed Cryptographic Mechanisms (ACM) catalogue [6].

NOTE 3: A cryptographic mechanism can be specified at different levels of abstraction. For example, AES is a cryptographic primitive, AES-GCM is a mode of operation / authenticated encryption construction, TLS 1.3 is a protocol, a TLS 1.3 restricted cipher-suite profile is a protocol profile, and TLS_AES_128_GCM_SHA256 is a cipher suite.

Cryptographic property: property provided or supported by a cryptographic mechanism.

NOTE 1: Cryptographic properties include, for example, confidentiality, integrity, authenticity, entity authentication, message authentication, key establishment, key confirmation, replay protection, collision resistance, second-preimage resistance, pre-image resistance, signature unforgeability, non-repudiation, forward secrecy and password-verifier protection. Where relevant, a cryptographic property can be expressed using a formal cryptographic notion, for example IND-CPA, IND-CCA, IND-CCA2, EUF-CMA, SUF-CMA or authenticated key exchange security.

NOTE 2: A cryptographic property is distinct from a product-level technical requirement, although it can support such a requirement. For example, a secure communication requirement can rely on cryptographic properties such as confidentiality, integrity and authentication.

K.1 Cryptography

K.1.1 Requirement

The product's default configuration shall only use cryptographic mechanisms that meet at least one of the following criteria:

1. ACM-listed: the cryptographic mechanism is listed in the ECCG Agreed Cryptographic Mechanisms (ACM) catalogue [6];
2. ACM-extended: the cryptographic mechanism is not listed in the ECCG Agreed Cryptographic Mechanisms (ACM) catalogue [6] and meets at least one of the following conditions:
   a. the cryptographic mechanism is listed in clause K.3.2 as an ACM-extended cryptographic mechanism for the specific product function(s);
   b. where the cryptographic mechanism is not listed in clause K.3.2, the cryptographic mechanism meets all the following criteria:
      - the cryptographic mechanism, or where applicable the ACM-listed cryptographic mechanism on which it is based, is not deprecated per the ECCG Agreed Cryptographic Mechanisms (ACM) catalogue [6];
      - the cryptographic mechanism has been specified, developed or maintained through a transparent process by a recognised European, international or sector-specific standards development organisation, or by an industry specification organisation accountable for the relevant specification, including CEN, CENELEC, ETSI, ISO, IEC, ISO/IEC JTC 1, IETF, IRTF, IEEE, ITU-T, NIST, W3C, FIDO Alliance, OIDF, SECG, AAIF, OASIS*;* or the cryptographic mechanism is listed as suitable in a publicly available cryptographic catalogue maintained by a recognised national or governmental cybersecurity authority, where the catalogue is maintained under a documented revision and retirement process, including those listed in clause K.3.2.6;
      - the cryptographic mechanism is described in a valid, publicly available and uniquely referenceable specification;
      - the cryptographic properties of the cryptographic mechanism are known;
      - no known weakness affects the cryptographic mechanism in a way that affects its cryptographic properties;
      - the cryptographic mechanism is required for a specific set of product functions;

3. Interoperability-based: the cryptographic mechanism is listed in clause K.4.2 as an interoperability-based cryptographic mechanism for specific product function(s) and external specification(s) or external requirement(s).
4. Pilot: the cryptographic mechanism is not listed in the ECCG Agreed Cryptographic Mechanisms (ACM) catalogue [6], but is used in the product's default configuration to evaluate, mature or accelerate the adoption of an emerging cryptographic mechanism, provided the cryptographic mechanism meets all of the following criteria:
   - the cryptographic mechanism is described in a publicly available specification, a draft submitted to a recognised standards development organisation (such as those listed in item 2.b), or is a precursor designed to gather technical data for such a standard;
   - the cryptographic properties of the cryptographic mechanism are known, and no known weakness affects the cryptographic mechanism, or any ACM-listed cryptographic mechanism on which it is based, in a way that affects those cryptographic properties, and any ACM-listed cryptographic mechanism on which it is based is not deprecated per the ACM catalogue;
   - where the cryptographic mechanism is used in a negotiated protocol, the product ensures that the cryptographic mechanism is used only with a communicating peer that supports it, and that a cryptographic mechanism meeting item 1) or item 2) is used otherwise, without reducing the cryptographic protection provided to the connection;
   - the implementation incorporates documented risk mitigation measures to protect against potential vulnerabilities in the cryptographic mechanism. For example, implementing the mechanism in a hybrid construction alongside an approved cryptographic mechanism meeting criterion 1) or 2), restricting the deployment to a limited time period, to defined product functions, or to a defined deployment scope.

- the documentation defines a maximum period of not more than 2 years after which its continued use under this item shall be re-justified, or the cryptographic mechanism shall be transitioned to a cryptographic mechanism meeting item 1) or item 2), updated, disabled or withdrawn.

NOTE 1: The reference to the product's default configuration is intended to define a clear and assessable baseline, corresponding to the configuration in which the product is placed on the market. The product can provide several configurations that fulfil the requirement.
NOTE 2: For products supplied as hardware platforms or components to be configured by an integrator, references in this annex to the product's default configuration refer to the configuration specified for the intended operational use of the product after integration, including the relevant integration assumptions and configuration constraints.
NOTE 3: The applicable ACM catalogue version is identified in the normative references of the present document. The lifecycle treatment of mechanisms affected by ACM deprecation dates, expiry dates, migration conditions or usage limitations is addressed in clause K.2.
NOTE 4: In this clause, an external specification or external requirement means a specification or requirement that is imposed on the product, and which requires the use of a specific cryptographic mechanism for the product to interoperate with an identified system, platform or operational context. An external requirement can be a regulatory requirement, operational constraint, technical interoperability constraint, or platform compatibility requirement.
NOTE 5: Inclusion of a cryptographic mechanism under the interoperability-based criterion does not classify that mechanism as state-of-the-art cryptography.
NOTE 6: Pilot cryptographic mechanisms should preferentially be limited to unfinished software, however it is recognised that in some cases that may not be sufficient to obtain real-world data.
EXAMPLE: Examples of product functions include secure communication based on TLS 1.3; authenticated encryption of communicated data based on AES-GCM; storage confidentiality based on AES-XTS; firmware signature verification based on Ed25519; and key derivation based on HKDF.

K.1.2 Assessment of product cryptographic configuration

K.1.2.0 General

The assessment in clause K.1.2 verifies the cryptographic mechanisms used in the product's default configuration against clause K.1.1.

For ACM-extended cryptographic mechanisms, the assessment verifies that the cryptographic mechanism is either listed in clause K.3.2 or fulfils the criteria specified in clause K.1.1 item 2.b. For interoperability-based cryptographic mechanisms, the assessment verifies that the cryptographic mechanism is listed in clause K.4.2. The assessment also verifies that the product uses the cryptographic mechanism in accordance with the relevant characteristics, product functions, use cases where applicable, external specifications or external requirements, and conditions specified in the present document.

K.1.2.1 Assessment of ACM-listed cryptographic mechanisms

K.1.2.1.1 Assessment objective

The purpose of this assessment case is to verify that cryptographic mechanisms claimed to comply with clause K.1.1 item 1) and used in the product's default configuration are listed in the ECCG Agreed Cryptographic Mechanisms (ACM) catalogue [6].

K.1.2.1.2 Assessment preparation

- Preconditions for the assessment: The product's default configuration shall be used for the assessment.
- The documentation shall provide a list of cryptographic mechanisms used in the product's default configuration and claimed under clause K.1.1 item 1), including the related product function, relevant parameters where applicable, and the relevant ACM entry.

K.1.2.1.3 Assessment activities

The assessment shall include verification that:

- each cryptographic mechanism claimed under clause K.1.1 item 1) is identified by reference to a relevant ACM entry;

- the ACM entry corresponds to the cryptographic mechanism used in the product's default configuration, including relevant parameters where applicable;
- where the ACM entry includes lifecycle information, such as expiry date, deprecation date, migration condition or usage limitation, this information is reflected in the documentation.

K.1.2.1.4 Assessment evidence

The assessment evidence shall include, as applicable:

- description of the product's default configuration;
- list of cryptographic mechanisms claimed under clause K.1.1 item 1);
- references to the relevant ACM entries;
- relevant parameters, profiles, cipher suites or configuration constraints, where applicable;
- relevant lifecycle information from the ACM catalogue, where applicable.

K.1.2.1.5 Assessment verdict

- The verdict PASS shall be assigned if the required evidence has been provided, is complete, is applicable to the assessed configuration, and demonstrates compliance with clause K.1.1 item 1).
- The verdict FAIL shall be assigned otherwise.

K.1.2.2 Assessment of ACM-extended cryptographic mechanisms

K.1.2.2.1 Assessment objective

The purpose of this assessment case is to verify that cryptographic mechanisms claimed to comply with clause K.1.1 item 2) and used in the product's default configuration are either listed as ACM-extended cryptographic mechanisms in clause K.3.2 or fulfil the criteria specified in clause K.1.1 item 2.b, and are used for the claimed product function(s).

K.1.2.2.2 Assessment preparation

- Preconditions for the assessment: The product's default configuration shall be used for the assessment.
- The documentation shall provide a list of cryptographic mechanisms used in the product's default configuration and claimed under clause K.1.1 item 2), including the related product function, use case where applicable, relevant parameters where applicable, and at least one of the following:
  1. the relevant entry in clause K.3.2; or
  2. evidence that the cryptographic mechanism fulfils all applicable criteria in clause K.1.1 item 2.b.

K.1.2.2.3 Assessment activities

The assessment shall include verification that:

- each cryptographic mechanism claimed under clause K.1.1 item 2) is either listed in clause K.3.2 as an ACM-extended cryptographic mechanism or supported by evidence demonstrating fulfilment of the applicable criteria in clause K.1.1 item 2.b;
- the cryptographic mechanism used by the product corresponds to the mechanism listed in clause K.3.2 or described in the evidence provided under clause K.1.1 item 2.b;
- the product function using the cryptographic mechanism, and the use case where applicable, correspond to the related product function and use case specified in clause K.3.2 or justified under clause K.1.1 item 2.b;
- the relevant characteristics, parameters, profiles, cipher suites or configuration constraints of the cryptographic mechanism correspond to those specified in clause K.3.2 or justified under clause K.1.1 item 2.b, where applicable;
- where clause K.3.2 or the justification under clause K.1.1 item 2.b defines conditions or limitations for the use of the cryptographic mechanism, these conditions or limitations are reflected in the product's default configuration;
- where technically feasible, the cryptographic mechanism, parameters and configuration identified in the documentation correspond to the assessed product configuration.

K.1.2.2.4 Assessment evidence

The assessment evidence shall include, as applicable:

- description of the product's default configuration;
- list of cryptographic mechanisms claimed under clause K.1.1 item 2);

- reference to the relevant entry in clause K.3.2, where the mechanism is listed in clause K.3.2;
- evidence that the cryptographic mechanism fulfils the criteria in clause K.1.1 item 2.b, where compliance with clause K.1.1 item 2 is claimed on the basis of those criteria;
- identification of the related product function using the cryptographic mechanism and the use case where applicable;
- relevant characteristics, parameters, profiles, cipher suites or configuration constraints, where applicable;
- evidence that the cryptographic mechanism is used in accordance with the conditions or limitations specified in clause K.3.2 or in the justification under clause K.1.1 item 2.b, where applicable.

K.1.2.2.5 Assessment verdict

- The verdict PASS shall be assigned if the required evidence has been provided, is complete, is applicable to the assessed configuration, and demonstrates compliance with clause K.1.1 item 2).
- The verdict FAIL shall be assigned otherwise.

K.1.2.3 Assessment of interoperability-based cryptographic mechanisms

K.1.2.3.1 Assessment objective

The purpose of this assessment case is to verify that cryptographic mechanisms claimed to comply with clause K.1.1 item 3) and used in the product's default configuration are listed as interoperability-based cryptographic mechanisms in clause K.4.2 and are used in accordance with the characteristics, product functions, use cases where applicable, external specifications or external requirements, and conditions specified in clause K.4.2.

K.1.2.3.2 Assessment preparation

- Preconditions for the assessment: The product's default configuration shall be used for the assessment.
- The documentation shall provide a list of cryptographic mechanisms used in the product's default configuration and claimed under clause K.1.1 item 3), including the related product function, use case where applicable, relevant parameters where applicable, the external specification or external requirement requiring its use, and the relevant entry in clause K.4.2.

K.1.2.3.3 Assessment activities

The assessment shall include verification that:

- each cryptographic mechanism claimed under clause K.1.1 item 3) is listed in clause K.4.2 as an interoperability-based cryptographic mechanism;
- the cryptographic mechanism used by the product corresponds to the cryptographic mechanism listed in clause K.4.2;
- the product function using the cryptographic mechanism, and the use case where applicable, correspond to the related product function and use case specified in clause K.4.2;
- the external specification or external requirement requiring the use of the cryptographic mechanism corresponds to the external specification or external requirement specified in clause K.4.2;
- the relevant characteristics, parameters, profiles, cipher suites or configuration constraints of the cryptographic mechanism correspond to those specified in clause K.4.2, where applicable;
- the conditions or limitations specified in clause K.4.2 for the use of the cryptographic mechanism are reflected in the product's default configuration;
- where technically feasible, the cryptographic mechanism, parameters and configuration identified in the documentation correspond to the assessed product configuration.

K.1.2.3.4 Assessment evidence

The assessment evidence shall include, as applicable:

1. description of the product's default configuration;
2. list of cryptographic mechanisms claimed under clause K.1.1 item 3);
3. reference to the relevant entry in clause K.4.2;
4. identification of the related product function using the cryptographic mechanism and the use case where applicable;
5. identification of the external specification or external requirement requiring use of the cryptographic mechanism;

6. relevant characteristics, parameters, profiles, cipher suites or configuration constraints, where applicable;
7. evidence that the cryptographic mechanism is used in accordance with the conditions or limitations specified in clause K.4.2.

K.1.2.3.5 Assessment verdict

- The verdict PASS shall be assigned if the required evidence has been provided, is complete, is applicable to the assessed configuration, and demonstrates compliance with clause K.1.1 item 3).
- The verdict FAIL shall be assigned otherwise.

NOTE 1: The assessment of the external specification or external requirement itself is outside the scope of this assessment case. The assessment verifies that the external specification or external requirement is identified in clause K.4.2 and that the cryptographic mechanism is used by the product for the related product function.

NOTE 2: Documentation review is a minimum assessment activity under the present annex. Functional and/or resistance testing activities can be added by vertical standards where relevant for the product category, product function or cryptographic mechanism.

K.1.2.4 Assessment of pilot cryptographic mechanisms

K.1.2.4.1 Assessment objective

The purpose of this assessment case is to verify that cryptographic mechanisms claimed to comply with clause K.1.1 item 4) and used in the product's default configuration are deployed to evaluate, mature or accelerate the adoption of an emerging cryptographic mechanism, and fulfil all criteria specified in clause K.1.1 item 4).

K.1.2.4.2 Assessment preparation

- Preconditions for the assessment: The product's default configuration shall be used for the assessment.
- The documentation shall provide a list of cryptographic mechanisms used in the product's default configuration and claimed under clause K.1.1 item 4), including the related product function, use case where applicable, relevant parameters where applicable, and evidence that the cryptographic mechanism fulfils all criteria in clause K.1.1 item 4).

K.1.2.4.3 Assessment activities

The assessment shall include verification that:

- each cryptographic mechanism claimed under clause K.1.1 item 4) is described in a publicly available specification, a draft submitted to a recognised standards development organisation, or is a precursor designed to gather technical data for such a standard;
- the cryptographic properties of the cryptographic mechanism are known, and no known weakness affects the cryptographic mechanism, or any ACM-listed cryptographic mechanism on which it is based, in a way that affects those cryptographic properties, and any ACM-listed cryptographic mechanism on which it is based is not deprecated per the ACM catalogue;
- where the cryptographic mechanism is used in a negotiated protocol, the product ensures that the cryptographic mechanism is used only with a communicating peer that supports it, and that a cryptographic mechanism meeting clause K.1.1 item 1) or item 2) is used otherwise, without reducing the cryptographic protection provided to the connection;
- the implementation incorporates documented risk mitigation measures (such as a hybrid construction, restricting the deployment to a limited time period, to defined product functions, or to a defined deployment scope) to protect against potential vulnerabilities in the cryptographic mechanism;
- the documentation defines a maximum period of not more than 2 years after which its continued use under this item shall be re-justified, or the cryptographic mechanism shall be transitioned to a cryptographic mechanism meeting clause K.1.1 item 1) or item 2), updated, disabled or withdrawn;

- where technically feasible, the cryptographic mechanism, parameters and configuration identified in the documentation correspond to the assessed product configuration.

K.1.2.4.4 Assessment evidence

The assessment evidence shall include, as applicable:

- description of the product's default configuration;
- list of cryptographic mechanisms claimed under clause K.1.1 item 4);
- reference to the publicly available specification, standard draft, or documented precursor;
- justification of known cryptographic properties and the absence of relevant known weaknesses;
- evidence demonstrating the protocol negotiation and fallback capabilities, where applicable;
- description of the implemented risk mitigation measures;
- documentation defining the maximum period (not exceeding 2 years) and the subsequent transition, update, disablement, withdrawal, or re-justification requirements.

K.1.2.4.5 Assessment verdict

The verdict PASS shall be assigned if the required evidence has been provided, is complete, is applicable to the assessed configuration, and demonstrates compliance with clause K.1.1 item 4). The verdict FAIL shall be assigned otherwise.

K.2 Crypto agility

K.2.1 Requirement

Where the product's default configuration uses a cryptographic mechanism for which the ACM catalogue, or the present document, specifies a deprecation date, expiry date, migration condition or usage limitation falling within the intended lifetime of the product, the product shall provide means for addressing the affected cryptographic mechanism by one or more of the following:

1. updating the cryptographic mechanism;
2. using another cryptographic mechanism that complies with clause K.1.1 and is not subject to the relevant deprecation date, expiry date, migration condition or usage limitation;
3. disabling the use of the affected cryptographic mechanism; or
4. limiting the use of the affected product functions accordingly.

EXAMPLE: Where a security mechanism uses a hybrid cryptographic construction, for example a hybrid key-encapsulation mechanism combining a classical and a post-quantum primitive, the lifecycle status of each constituent primitive is relevant to the lifecycle status of the construction. Hybridization can be used as part of a planned migration strategy where permitted by the ACM catalogue or by the present document. However, hybridization does not, by itself, extend the recommended usage lifetime of a constituent primitive that has been deprecated.

NOTE 1: Where a hybrid cryptographic construction includes multiple cryptographic primitives, the lifecycle status of each constituent primitive is relevant to the lifecycle status of the construction. Hybridization is not assumed to mitigate the deprecation of a constituent primitive unless the ACM catalogue or the present document classifies the hybrid construction as suitable for the corresponding product function.

NOTE 2: Crypto agility supports maintaining appropriate cryptographic protection within the intended lifetime of the product, in addition to the capability of updating cryptographic mechanisms on the product in accordance with secure update and secure communication mechanisms.

NOTE 3: Deprecation information from sources other than the ACM catalogue or the present document can be considered as input to vulnerability management or security risk assessment but does not by itself trigger the requirement in clause K.2.1.

NOTE 4: Where the product provides cryptographic capabilities for use by other products, components or layers, the mechanism required by clause K.2.1 can consist of update, configuration, disabling, deprecation or limitation capabilities usable by the integrating product, system operator or user, where applicable.

NOTE 5: The applicability condition of this requirement can express exceptions based on product characteristics, e.g. cryptographic implementation based on immutable hardware co-processor(s) or hardware-based root of trust, or long-lived silicon used in a long-lived non-updateable environment.

K.2.2 Assessment of crypto-agility

K.2.2.1 Assessment objective

The purpose of this assessment case is to verify whether, where the product's default configuration uses a cryptographic mechanism for which the ACM catalogue, or the present document, specifies a deprecation date, expiry date, migration condition or usage limitation falling within the intended lifetime of the product, the product provides means to address the affected cryptographic mechanism in accordance with clause K.2.1.

K.2.2.2 Assessment preparation

- Preconditions for the assessment: The product's default configuration shall be used for the assessment.
- The documentation shall identify:
  1. the intended lifetime of the product;
  2. the cryptographic mechanisms used in the product's default configuration;
  3. the related product function(s) for each cryptographic mechanism;
  4. the lifecycle information applicable to each cryptographic mechanism, where specified by the ACM catalogue or the present document, including any deprecation date, expiry date, migration condition or usage limitation.

K.2.2.3 Assessment activities

The assessment shall include verification that:

1. for each cryptographic mechanism used in the product's default configuration, the lifecycle information specified by the ACM catalogue or the present document has been identified, where such information is specified;
2. where the ACM catalogue or the present document specifies a deprecation date, expiry date, migration condition or usage limitation for a cryptographic mechanism, this information is reflected in the documentation;
3. where a deprecation date, expiry date, migration condition or usage limitation falls within the intended lifetime of the product, the product provides an applicable mechanism for updating the cryptographic mechanism, using another cryptographic mechanism that complies with clause K.1.1 and is not subject to the relevant deprecation date, expiry date, migration condition or usage limitation, disabling the use of the affected cryptographic mechanism, or limiting the use of the affected product functions accordingly;
4. where another cryptographic mechanism is used, that cryptographic mechanism complies with clause K.1.1 and is not subject to the relevant deprecation date, expiry date, migration condition or usage limitation.

K.2.2.4 Assessment evidence

The assessment evidence shall include, as applicable:

1. documentation of the intended lifetime of the product;
2. list of cryptographic mechanisms used in the product's default configuration;
3. identification of the related product function(s) for each cryptographic mechanism;
4. references to the ACM catalogue entry or to the provision of the present document used to determine lifecycle information, where applicable;
5. identification of any deprecation date, expiry date, migration condition or usage limitation falling within the intended lifetime of the product;
6. description of the means provided by the product, such as update of the cryptographic mechanism, use of another cryptographic mechanism that complies with clause K.1.1 and is not subject to the relevant lifecycle constraint, disabling the use of the affected cryptographic mechanism, or limitation of the use of the affected product functions.

K.2.2.5 Assessment verdict

- The verdict PASS shall be assigned if the required evidence has been provided, is complete, is applicable to the assessed configuration, and demonstrates compliance with clause K.2.1.
- The verdict FAIL shall be assigned otherwise.

K.3 ACM-extended cryptographic mechanisms

K.3.1 Requirement

Where the product's default configuration uses ACM-extended cryptographic mechanisms, these mechanisms shall comply with the ACM-extended criterion in clause K.1.1 item 2.

K.3.2 List of ACM-extended cryptographic mechanisms

ACM-extended cryptographic mechanism	Type of cryptographic mechanism	Specification / reference
TLS_CHACHA20_POLY1305_SHA256	Cipher suite	https://www.rfc-editor.org/info/rfc8439/
TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256	Cipher suite	https://www.rfc-editor.org/info/rfc7905/
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	Cipher suite	https://www.rfc-editor.org/info/rfc8422/
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	Cipher suite	https://www.rfc-editor.org/info/rfc7905/
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	Cipher suite	https://www.rfc-editor.org/info/rfc8422/
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	Cipher suite	https://www.rfc-editor.org/info/rfc8422/
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	Cipher suite	https://www.rfc-editor.org/info/rfc8422/
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	Cipher suite	https://www.rfc-editor.org/info/rfc7905/
TLS_RSA_WITH_3DES_EDE_CBC_SHA	Cipher suite	https://www.rfc-editor.org/info/rfc3268/
TLS_RSA_WITH_AES_128_CBC_SHA	Cipher suite	https://www.rfc-editor.org/rfc/rfc3268.html
TLS_RSA_WITH_AES_256_CBC_SHA	Cipher suite	https://www.rfc-editor.org/rfc/rfc3268.html
ecdsa_secp256r1_sha256	Construction	https://www.ietf.org/archive/id/draft-mattsson-tls-compact-ecc-00.html
rsa_pkcs1_sha512	Construction	https://www.rfc-editor.org/info/rfc8446/
rsa_pss_rsae_sha384	Construction	https://www.rfc-editor.org/info/rfc8017/
rsa_pss_rsae_sha512	Construction	https://www.rfc-editor.org/info/rfc8017/
AES-GCM-SIV	Primitive	https://www.rfc-editor.org/info/rfc8452/
AES-OCB	Primitive	https://www.rfc-editor.org/info/rfc7253/
Argon2d	Primitive	https://www.rfc-editor.org/info/rfc9106/
Argon2i	Primitive	https://www.rfc-editor.org/info/rfc9106/
Argon2id	Primitive	https://www.rfc-editor.org/info/rfc9106/
Blake2b	Primitive	https://datatracker.ietf.org/doc/html/rfc7693.html

Blake2s	Primitive	https://datatracker.ietf.org/doc/html/rfc7693.html
ChaCha20	Primitive	https://www.rfc-editor.org/info/rfc8439/
Ed25519	Primitive	https://www.rfc-editor.org/info/rfc8032/
SHAKE128	Primitive	https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.202.pdf
SHAKE256	Primitive	https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.202.pdf
TurboSHAKE128	Primitive	https://www.rfc-editor.org/info/rfc9861/
TurboSHAKE256	Primitive	https://www.rfc-editor.org/info/rfc9861/
X-Wing	Primitive	https://www.ietf.org/archive/id/draft-connolly-cfrg-xwing-kem-10.html
X25519	Primitive	https://www.rfc-editor.org/info/rfc7748/
X25519MLKEM768	Primitive	https://www.ietf.org/archive/id/draft-ietf-tls-ecdhe-mlkem-05.html
X448	Primitive	https://www.rfc-editor.org/info/rfc7748/
XChaCha20	Primitive	https://datatracker.ietf.org/doc/html/draft-arciszewski-xchacha-03
XChaCha20-Poly1305	Primitive	https://datatracker.ietf.org/doc/html/draft-arciszewski-xchacha-03
XSalsa20	Primitive	https://cr.yp.to/snuffle/xsalsa-20110204.pdf

K.3.3 Assessment

Compliance with the ACM-extended criterion in clause K.1.1 item 2 is assessed in clause K.1.2.2.

History

The following table will automatically be filled in by the ETSI Secretariat.

Document history		
V0.0.1	16 Feb 2026	Preliminary draft distributed to all ETSI-EUSR Rapporteurs for comments
V0.0.2	24 Feb 2026	Contributed and presented to CYBEREUSR#30, including comments received from consumer IoT and Hypervisor rapporteurs
V0.0.3	5 March 2026	Contributed to CYBEREUSR#31, including comments from previous meeting and additional guidance received from EC and new introduction clause
V0.0.4	10 April 2026	Resolution of inconsistencies, links errors and editorials, agreed changes at skeleton meeting held on 10 April 2026, v0.0.7 of Annex K
V0.0.5	19 Mai 2026	Inclusion of additional guidance (editor's notes) provided by the EC, editorial corrections, resolution of comments raised by the EC on v0.0.4 during meetings held on 18 and 19 may 2026, Exclusion of Annex K that will be included once endorsed by the TB

Last update on 2026-05-19