Application Software Extended Package for Web Browsers

Version: 2.0

2015-06-16

National Information Assurance Partnership

Revision History

Version	Date	Comment
v 1.0	2014-03-31	Release - Protection Profile for Web Browsers
v 2.0	2015-06-16	Update as Extended Package of the Protection Profile for Application Software

Contents

1. Introduction

1.1 Overview

Web browsers are client applications that retrieve and render content provided by web servers, primarily using the hypertext transfer protocol (HTTP) or HTTP Secure (HTTPS). Browsers have grown in complexity over the years, starting as tools used to display simple, unchanging web pages and becoming sophisticated execution environments for web content. The use of browsers to administer accounts, servers or embedded systems remotely requires them to handle sensitive information securely. Innovations such as tabs, extensions and HTML5 have not only increased browser functionality, but also introduced new security concerns. Being the principal method for accessing the Internet, and due to their complexity and the information that they process, browsers are a natural target for attackers. As a result, it is paramount that the security of web browsers be improved to reduce the risk to client machines and enterprise networks.

This Extended Package (EP) along with the Protection Profile for Application Software ([AppPP]) provide a baseline set of Security Functional Requirements (SFRs) for web browsers running on any operating system regardless of the composition of the underlying platform. The requirements are intended to improve the security of browsers by encouraging the use of operating system security services and requiring the use of sandboxing technologies and environmental mitigations provided by the underlying platform. Additionally, these requirements define security functionality that browsers must provide.

The terms web browser, browser, and TOE are interchangeable in this document.

1.2 Terms

The following sections provide both Common Criteria and technology terms used in this Extended Package.

1.2.1 Common Criteria Terms

Common Criteria (CC)	Common Criteria for Information Technology Security Evaluation.
Extended Package (EP)	An implementation-independent set of security requirements for a category of products, which extends those in a Protection Profile.
Protection Profile (PP)	An implementation-independent set of security requirements for a category of products.
Security Target (ST)	A set of implementation-dependent security requirements for a specific product.
Target of Evaluation (TOE)	The product under evaluation. In this case, a web browser and its supporting documentation.
TOE Security Functionality (TSF)	The security functionality of the product under evaluation.
TOE Summary Specification (TSS)	A description of how a TOE satisfies the SFRs in a ST.
Security Functional Requirement (SFR)	A requirement for security enforcement by the TOE.
Security Assurance Requirement (SAR)	A requirement to assure the security of the TOE.

1.2.2 Technology Terms

Add-on	Capabilities or functionality added to an application. This term includes plug-ins, extensions, and other controls.
Administrator	The Administrator is responsible for management activities, including setting the policy that is applied by the enterprise on the browser. This administrator is likely to be acting remotely. If the platform is unmanaged by an enterprise, the user can act as the administrator.
CSRF	Cross Site Request Forgery - Vulnerability where an attacker gets a target user to execute a script with that user's privileges.
Domain	A realm of administrative autonomy, authority or control on the Internet (e.g., cnn.com).
Extension	Bundle of code added to the browser to add specific functionality that the browser does not provide by default.
HTML	HyperText Markup Language - Language used by web servers to present content to browsers.
HTML5	HyperText Markup Language version 5, a new version of HTML that incorporates many new features that enrich the browsing experience.
HTTP	HyperText Transfer Protocol - Protocol for communicating on the web.
HTTPS	HyperText Transfer Protocol Secure; secure version of HTTP that runs over an encrypted channel (SSL/TLS).
JavaScript	Scripting language commonly integrated into web pages to generate dynamic, interactive content.
Mobile Code	Software transmitted from a remote system for execution within a limited execution environment on the local system. Typically, there is no persistent installation and execution begins without the user's consent or even notification. Examples of mobile code technologies include Java applets, Adobe ActionScript, and Microsoft Silverlight. **Note:** JavaScript is not included in references to mobile code in this browser EP.
Plug-in	Browser add-on to handle specific types of web content.
Pop-up	Piece of web code that causes a browser to open a window outside the window that is currently in focus.
Port	An application-specific construct that functions as a communications endpoint in a computer's host OS; in a web environment, port 80 is the default port for HTTP

	communications, although other ports can be used. In a web address, the port follows the domain or sub-domain name (e.g., http://www.cnn.com:80).
Protocol	A system of digital rules for data exchange within or between computers; in a web environment, the typical protocols are HTTP and HTTPS.
Sandbox	Security mechanism for separating running processes, most often used to run untrusted or vulnerable processes by reducing their privileges to such an extent that they should not be able to harm the host system.
Sensitive Data	Sensitive data may include all user or enterprise data or may be specific application data such as data transferred to submit a form or complete a transaction. Sensitive data must minimally include personally identifiable information (PII), credentials, and keys. Sensitive data shall be identified in the application's TSS by the ST author.
Sub-domain	An Internet domain which is part of a primary domain, denoted by a prefix before the primary domain (e.g., news.cnn.com).
Tabs	Allow the browsers to display content from multiple web sites in the same window.
Web Browser	Application that retrieves and renders content provided by a web server. The terms web browser, browser, and TOE are interchangeable in this document.
XSS	Cross Site Scripting - Injection of untrusted content into a vulnerable web application to render or execute that content on a victim's system.

1.3 Compliant Targets of Evaluation

The Target of Evaluation (TOE) in this EP is any web browser client capable of running on any operating system or platform and rendering web content using HTTP and HTTPS.

This EP describes the extended security functionality of web browsers in terms of [CC]. As an extension of the App PP, it is expected that the content of this EP will be appropriately combined with the App PP to include selection-based requirements in accordance with the selections and/or assignments made, and any optional and/or objective components to include: FCS_CKM.1.1, FCS_CKM.2.1, FCS_COP.1.1(*), FCS_DTLS_EXT.1.*, FCS_HTTPS_EXT.1.*, FCS_RBG_EXT.2.*, FCS_TLSC_EXT.1.*, FIA_X509_EXT.1.*, FIA_X509_EXT.2.*.

An ST must identify the applicable version of the App PP and this EP in its conformance claims.

1.4 Use Cases

Requirements in this extended package are designed to address the security problems in the use cases below. These use cases are intentionally very broad, as web browsers can be used to perform many tasks.

[USE CASE 1] Surfing the Web

Browsers are used to retrieve, display and render content from the web, such as web pages, streaming media, images and specialized formats (e.g., Java, Flash, PDF). They can also be used to write content to web sites (web 2.0 – e.g., Facebook). Web surfing can be done over the Internet or within an Intranet.

[USE CASE 2] Remote Administration Client

Browsers are used to provide remote administration interfaces for systems such as servers, network devices and embedded systems, to include supervisory control and data acquisition (SCADA) systems, smart TVs and thermostats. As opposed to surfing the web, where the browser may be interacting with untrusted content, the browser, acting as a Remote Administration Client, is connecting to a server that the user trusts.

[USE CASE 3] Content Creation

Browsers are used to create content via an increasing number of Software as a Service (SaaS) offerings, including Microsoft Office 365, Google Drive, and Adobe Creative Cloud, where user data and records are stored online.

2. Conformance Claims

Conformance Statement

The Protection Profile for Application Software ([AppPP]) defines the baseline Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) for application software products. This EP serves to extend the App PP baseline with additional SFRs and associated Assurance Activities specific to a web browser. Assurance Activities are the actions that the evaluator performs in order to determine a web browser's compliance to the SFRs.

This EP conforms to Common Criteria [CC] for Information Technology Security Evaluation, Version 3.1, Revision 4. It is CC Part 2 extended and CC Part 3 conformant. In order to be conformant to this EP, the ST must include all components in this EP and the associated App PP that are:

- unconditional (which are always required)
- selection-based (which are required when certain selections are chosen in the unconditional requirements)

and may include optional and/or objective components that are desirable but not required for conformance.

In accordance with CC Part 1, dependencies are not included when they are addressed by other SFRs. The assurance activities provide adequate proof that any dependencies are also satisfied.

3. Security Problem Description

The security problem is described in terms of the threats that the web browser is expected to address, assumptions about the operational environment, and any organizational security policies that it is expected to enforce.

This Extended Package does not repeat the threats, assumptions, and organizational security policies identified in the App PP, though they all apply given the conformance and hence dependence of this EP on it. Together the threats, assumptions and organizational security policies of the App PP and those defined in this EP describe those addressed by a web browser as the Target of Evaluation.

Notably, browsers are particularly at risk from the Network Attack threat identified in the App PP. Attackers can use phishing or another social engineering technique to persuade a user to visit a malicious site. Users may also unintentionally visit malicious sites in the course of web browsing. Such sites then present malicious content to the user's browser to exploit it and perform installation of malware, often with no indication to the user.

3.1 Threats

The following threats are specific to web browsers, and represent an addition to those identified in the App PP.

T.FLAWED_ADDON

Web browser functionality can be extended through the integration of third-party utilities and tools. Malicious or vulnerable add-ons could result in attacks against the system. Such attacks can allow unauthorized access to sensitive information in the browser, unauthorized access to the platform's file system, or even privilege escalation that enables unauthorized access to other applications or the operating system.

T.SAME-ORIGIN_VIOLATION

Violating the same-origin policy is a specialized type of network attack (covered generally as T.NETWORK_ATTACK in the App PP) which involves web content violating access control policies enforced by a web browser to separate the content of different web domains. It is specifically identified as a threat to web browsers, since they implement the access control policies that are violated in these attacks.

Attacks which involve same origin violations include:

- Insufficient protection of session tokens can lead to session hijacking, where a token is captured and reused in order to gain the privileges of the user who initiated the session.
- Cross-site scripting (XSS) and Cross-Site Request Forgery (CSRF) attacks are methods used to compromise user credentials (usually by stealing the user's session token) to a web site. These attacks are more likely a result of server security problems, but some browsers incorporate technologies that try to detect the attacks.
- Inadequate sandboxing of browser windows/tabs or a faulty cross domain communications model can lead to leakage of content from one domain in one window/tab to a different domain in a different window/tab. Such attacks leverage the ability of browsers to display content from multiple domains simultaneously.

4. Security Objectives

This Extended Package adds security objectives to those identified in the Protection Profile for Application Software (App PP).

4.1 Security Objectives for the TOE

O.INTEGRITY

Addressed by: FPT_DNL_EXT.1, FPT_MCD_EXT.1

O.MANAGEMENT

Addressed by: FDP_TRK_EXT.1, FMT_MOF_EXT.1

O.PROTECTED_STORAGE

Addressed by: FDP_COO_EXT.1, FDP_PST_EXT.1

O.PROTECTED_COMMS

Addressed by: FCS_STS_EXT.1, FDP_STR_EXT.1, FPT_INT_EXT.1, FPT_INT_EXT.2

O.DOMAIN_ISOLATION

To address the network attack associated with content leakage between different web domains, the browser must ensure that content originating from different domains (e.g., in a tab or iFrame) is properly isolated.

Addressed by: FDP_ACF_EXT.1.1, FDP_SBX_EXT.1, FDP_SOP_EXT.1

O.ADDON_INTEGRITY

To address issues associated with malicious or flawed add-ons, conformant browsers implement mechanisms to ensure their integrity. This includes verification and validation at installation time and update.

Addressed by: FPT_AON_EXT.1, FPT_AON_EXT.2

5. Security Requirements

This chapter describes the security requirements which have to be fulfilled by the browser. The browser must not rely on any third party add-ons, or vendor supplied add-ons which do not adhere to the browser's sandbox, to accomplish these requirements. Security requirements comprise functional components from Part 2 of [CC]. The following notations are used:

- **Selection** (denoted by *italicized text*): is used to select one or more options provided by the [CC] in stating a requirement. Selections must be captured in the ST.
- **Assignment** operation (denoted by *italicized text*): is used to assign a specific value to an unspecified parameter, such as the length of a password. Showing the value in square brackets indicates assignment.
- **Iteration** operation: are identified with a number inside parentheses (e.g. "(1)").

5.1 Security Functional Requirements

The Security Functional Requirements included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, with additional extended functional components.

5.1.1 Cryptographic Support (FCS)

FCS_STS_EXT.1 Strict Transport Security

FCS_STS_EXT.1.1

The browser shall implement HTTP Strict-Transport-Security according to RFC 6797.

This is an objective requirement.

FCS_STS_EXT.1.2

The browser shall retain persistent data signaling HSTS enablement for the time span declared by the website in a max-age directive.

This is an objective requirement.

FCS_STS_EXT.1.3

The browser shall cache the "freshest" Strict Security policy information.

This is an objective requirement.

Application Note: Freshness refers to the length of time between generation by the origin server and the expiration time when the origin server specifies that a stored response can no longer be used by a cache without further validation (RFCs 6797 and 7234). If a browser receives the HSTS header from a website, all future HTTP sessions between the browser and the domain or superdomain of that website must occur over TLS 1.2 (RFC 5246) or greater by utilizing HTTPS (RFC 2818) negotiating the strongest cipher possible.

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure that it documents how the browser supports HSTS.

Guidance

The evaluator shall examine the operational guidance to ensure it contains instructions on how to use HSTS.

Tests

The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall connect to a HSTS-compliant website while running a network protocol analyzer to monitor the traffic. The evaluator shall examine the captured network traffic and verify that a Strict Transport Security header is received and that there is a directive for the max-age of the HSTS relationship.*
- ***Test 2:*** *The evaluator shall reconnect to the HSTS website again over HTTP and shall verify that the session is redirected to HTTPS.*
- ***Test 3:*** *The evaluator shall reconnect to the HSTS website after the max-age has expired, and verify that the website and browser reestablish an HSTS relationship.*
- ***Test 4:*** *The evaluator shall update the website HSTS information, and verify that when the browser reconnects to the website, that information is updated by the browser.*

5.1.1 User Data Protection (FDP)

FDP_ACF_EXT.1 Local and Session Storage Separation

FDP_ACF_EXT.1.1

The browser shall separate local (permanent) and session (ephemeral) storage based on domain, protocol and port:

- Session storage shall be accessible only from the originating window/tab;
- Local storage shall only be accessible from windows/tabs running the same web application.

Application Note: The separation of local and session storage is described in World Wide Web Consortium (W3C) Proposed Recommendation: "Web Storage".

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure it describes how the browser separates local and session storage.

Guidance

The evaluator shall examine the operational guidance to verify that it documents the location on the file system that will be used for local storage and the location used for session storage.

Tests

The evaluator shall obtain or create JavaScript-based scripts that store and retrieve information from local and session storage and shall set up a web server with two or more web pages from different domains using different protocols and/or ports. The evaluator shall incorporate the scripts into the web pages and shall perform the following tests:

- ***Test 1:*** *The evaluator shall open two or more browser windows/tabs and navigate to the same web page. The evaluator shall verify that the script for accessing session storage that is running in one window/tab cannot access session storage associated with a different window/tab.*
- ***Test 2:*** *The evaluator shall open windows/tabs and navigate to different web pages. The evaluator shall verify that a script running in the context of one domain/protocol/port in a browser window/tab cannot access information associated with a different domain/protocol/port in a different window/tab.*

FDP_COO_EXT.1 Cookie Blocking

FDP_COO_EXT.1.1

The browser shall provide the capability to block the storage of third party cookies by websites.

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure it describes how the browser blocks third party cookies and when the blocking occurs (e.g., automatically, when blocking is enabled).

Guidance

The evaluator shall examine the operational guidance to verify that it provides a description of the configuration option for blocking of third party cookies.

Tests

The evaluator shall perform the following tests which may require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products:

- ***Test 1:*** *The evaluator shall clear all cookies and then configure the browser so that storage of third party cookies is allowed. The evaluator shall load a web page that stores a third party cookie. The evaluator shall navigate to the location where cookies are stored and shall verify that the cookie is present.*
- ***Test 2:*** *The evaluator shall clear all cookies and then configure the browser so that storage of third party cookies is blocked (i.e. not allowed). The evaluator shall load a web page that attempts to store a third party cookie and shall verify that the cookie was not stored.*

FDP_PST_EXT.1 Storage of Persistent Information

FDP_PST_EXT.1.1

The browser shall provide the capability to operate without storing persistent data to the file system with the following exceptions: [**selection**: *credential information, administrator provided configuration information, certificate revocation information, no exceptions*] .

This is an optional requirement. It may be required by Extended Packages of this Protection Profile.

Application Note: Any data that persists after the browser closes, including temporary files, is considered to be persistent data.

Assurance Activity ⮟

TSS

The evaluator shall examine the TSS to verify it describes how the browser operates without storing persistent user data to the file systems.

Guidance

N/A

Tests

The evaluator shall perform the following test which may require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products:

- ***Test 1:*** *The evaluator shall operate the browser for a period of time, ensuring that a wide variety of browser functionality has been exercised. The evaluator shall then examine the browser and the underlying platform to ensure that no files have been written to the file system other than the exceptions identified in FDP_PST_EXT.1.1.*

FDP_SBX_EXT.1 Sandboxing of Rendering Processes

FDP_SBX_EXT.1.1

The browser shall ensure that web page rendering is performed in a process that is restricted in the following manner:

- The rendering process can only directly access the area of the file system dedicated to the browser.
- The rendering process can only directly invoke inter-process communication mechanisms with its own browser processes.
- The rendering process has reduced privilege with respect to other browser processes [**selection**: *[**assignment**: other methods by which the principle of least privilege is implemented for rendering processes] , in no other ways*]

Application Note: Web browsers implement a variety of methods to ensure that the process that renders HTML and interprets JavaScript operates in a constrained environment in order to reduce the risk that the rendering process can be corrupted by the HTML or JavaScript it is processing. This component requires the browser to lower the privileges of rendering processes by ensuring that it cannot directly access the file system of the host, and that it cannot use IPC mechanisms provided by the host to communicate with non-browser processes on the host. Typically, if a rendering process needs to access a file or communicate with a non-browser process, it must request such access through the TSF (which is allowed by the requirement).

In addition to the two required measures, other measures can be implemented depending on the browser and the host platform. These may involve such actions as changing the owner of the rendering process to a low-privileged account or dropping platform-defined privileges in the rendering process. The ST author fills in the additional measures implemented by the browser.

Assurance Activity ⮟

TSS

The evaluator shall examine the TSS to ensure it describes how the rendering of HTML and interpretation of JavaScript is performed by the browser in terms of the platform processes that are involved (with "process" being an active entity that executes code). For the processes that render HTML or interpret JavaScript, the evaluator shall examine the TSS to check that it describes how these processes are prevented from accessing the platform file system. The evaluator shall check the

TSS to ensure it describes each platform-provided IPC mechanism, and details for each mechanism how the rendering process is unable to use it to communicate with non-browser processes. The evaluator shall also confirm that the TSS describes how IPC and file system access is enabled (if this capability is implemented); for instance, through a more privileged browser process that does not perform web page rendering. The evaluator shall ensure that these descriptions are present for all platforms claimed in the ST.

For each additional mechanism listed in the third bullet of this component by the ST author, the evaluator shall examine the TSS to ensure 1) the mechanisms are described; 2) the description of the mechanisms are sufficiently detailed to determine that it contributes to the principle of least privilege being implemented in the rendering process; and 3) appropriate supporting information is provided in the TSS (or pointers to such information are provided) that provides context for understanding the claimed least privilege mechanisms.

Guidance

The evaluator shall examine the operational guidance to determine that it provides a description of the restrictions available on rendering processes. Additionally, if such mechanisms are configurable (for instance, if a user can choose which mechanisms to "turn on"), the evaluator shall examine the operational guidance to ensure that the method for enabling and disabling the mechanisms are provided, and the consequences of such actions are described.

Tests

The evaluator shall perform the following test on each platform claimed in the ST:

- ***Test 1:*** *The evaluator shall execute a form of mobile code within an HTML page that contains instructions to modify or delete a file from the file system and verify that the file is not modified for deleted.*

FDP_SOP_EXT.1 Same Origin Policy

FDP_SOP_EXT.1.1

The browser shall only permit scripts contained in one web page to access data in a second web page if both pages are from the same origin.

FDP_SOP_EXT.1.2

The browser shall enforce the same origin policy for all domains.

Application Note: The Same Origin Policy concept is described in RFC 6454, "The Web Origin Concept".

Origin is defined as the combination of domain, protocol and port. Two URIs sharing the same domain, protocol and port are considered to have the same origin.

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure it describes its implementation of a same origin policy and explains how it complies with RFC 6454. If the browser allows the relaxation of the same origin policy for subdomains in different windows/tabs, the TSS shall describe how these exceptions are implemented.

Guidance

N/A

Tests

The evaluator shall obtain or create scripts that can retrieve content from designated locations and shall set up a web server with two or more web pages representing different domains. The evaluator shall incorporate the scripts into the web pages. The evaluator shall *associate each page with a different protocol and/or port and shall perform the following tests:*

- ***Test 1:*** *The evaluator shall open two or more browser windows/tabs and navigate to a different page on the website in each window/tab. The evaluator shall run the scripts and shall verify that the script that is running in one window/tab cannot access content that was retrieved in a different window/tab.*
- ***Test 2:*** *The evaluator shall verify that the scripts can retrieve content from another window/tab at a different subdomain.*

FDP_STR_EXT.1 Secure Transmission of Cookie Data

FDP_STR_EXT.1.1

The browser shall ensure that cookies containing the *secure* attribute in the set-cookie header are sent over HTTPS.

Application Note: The set-cookie header functionality is described in RFC 6265, "HTTP State Management Mechanism".

Assurance Activity ⮟

TSS

The evaluator shall examine the TSS to verify it describes the browser's support for the "secure" attribute of the set-cookie header in accordance with RFC 6265, including the required sending of cookies containing this attribute over HTTPS.

Guidance

N/A

Tests

The evaluator shall perform the following tests which may require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products:

- ***Test 1:*** *The evaluator shall connect the browser to a cookie-enabled test website implementing HTTPS and have the website present the browser with a "secure" cookie. The evaluator shall examine the browser's cookie cache and verify that that it contains the secure cookie.*
- ***Test 2:*** *The evaluator shall reconnect to the cookie-enabled website over an insecure channel and verify that no "secure" cookie is sent.*

FDP_TRK_EXT.1 Tracking Information Collection

FDP_TRK_EXT.1.1

The browser shall provide notification to the user when tracking information for [**selection**:

geolocation,
browser history,
browser preferences,
browser statistics

] is requested by a website.

Assurance Activity ⮟

TSS

The evaluator shall examine the TSS to ensure it describes the browser's support for tracking information and specifies the tracking information that the browser allows websites to collect about the browser user.

Guidance

The evaluator shall examine the operational guidance to ensure it *describes any notifications that the user will receive when tracking information is requested by a website and the options that the user has upon receiving the notification.*

Tests

The evaluator shall perform the following tests for each type of tracking information listed in the TSS:

- ***Test 1:*** *The evaluator shall configure a website that requests the tracking information about the user and shall navigate to that website. The evaluator shall verify that the user is notified about the request for tracking information and that, upon consent, the web browser retrieves the tracking information.*
- ***Test 2:*** *The evaluator shall verify that the user is notified about the request for tracking information and that, when rejected, the browser does not provide the tracking information.*

5.1.2 Security Management (FMT)

FMT_MOF_EXT.1 Management of Functions Behavior

FMT_MOF_EXT.1.1

The browser shall be capable of performing the following management functions, controlled by the administrator or user as shown:

- X = Mandatory
- O = Optional

Management Function	**Administrator**	**User**
Enable/disable storage of third party cookies	O	X
Enable/disable use of OCSP for obtaining the revocation status of X.509 certificates	O	O
Configure inclusion of user-agent information in HTTP headers	O	O
Enable/disable ability for websites to collect tracking information about the user through [**selection**: *zombie cookies, add-on based tracking (e.g. Flash cookies), browsing history, [**assignment**: other tracking mechanisms]*]	O	O
Enable/disable deletion of stored browsing data (cache, web form information)	O	X
Enable/disable storage of sensitive information (e.g., auto-fill, auto-complete) in persistent storage	O	O
Configure size of cookie cache	O	O
Configure size of cache	O	O
Enable/disable interaction with Graphic Processing Units (GPUs)	O	O
Configure the ability to advance to a web site with an invalid or unvalidated X.509 certificate	O	O
Enable/disable establishment of a trusted channel if the browser cannot establish a connection to determine the validity of a certificate	O	O
Configure the use of an application reputation service to detect malicious applications prior to download	O	O
Configure the use of a URL reputation service to detect sites that contain malware or phishing content	O	O
Enable/disable automatic installation of software updates and patches	O	O

Enable/disable ability for websites to register protocol handlers	O	O
Enable/disable display notification when unsigned, untrusted or unverified mobile code is encountered	O	O
Enable/disable user's ability to select default actions upon download of a file (e.g., always open, or always save, a downloaded file)	O	O
Enable/disable launching of downloaded files outside the browser	O	O
Enable/disable JavaScript	O	O
Enable/disable [**selection**: *ActiveX*, *Flash*, *Java*, *[**assignment**: other mobile code types supported by the browser]*] mobile code	O	O
Enable/disable support for add-ons	O	O
Enable/disable individual add-ons	O	O
Enable/disable HSTS mode	O	O

Application Note: For these management functions, the term "Administrator" refers to the administrator of a non-mobile device or the device owner of a mobile device. The intent of this requirement is to allow the user and administrator of the platform to configure the browser with configuration policies. If the administrator has not set a policy for a particular function, the user may still perform that function. Enforcement of the policy is done by the browser itself, or the browser and its platform in coordination with each other.

Disabling OCSP shall only be permitted if CRL was selected in FIA_X509_EXT.1.1 ([AppPP]).

Assurance Activity

TSS

The evaluator shall verify that the TSS describes those management functions which may only be configured by the browser platform administrator and cannot be over-ridden by the user when set according to policy.

Guidance

The evaluator shall examine the operational guidance to verify that it includes instructions for a browser platform administrator to configure the functions listed in FMT_MOF.1.1.

Tests

The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall verify that functions perform as intended by enabling, disabling, and configuring the functions.*
- ***Test 2:*** *The evaluator shall create policies that collectively include all management functions controlled by the browser platform administrator and cannot be over-ridden by the user as defined in FMT_MOF.1.1. The evaluator shall apply these policies to the browser, attempt to override each setting as the user, and verify that the browser does not permit it.*

5.1.3 Protection of the TSF (FPT)

FPT_DNL_EXT.1 File Downloads

FPT_DNL_EXT.1.1

The browser shall prevent downloaded content from launching automatically.

FPT_DNL_EXT.1.2

The browser shall present the user with the option to either save or discard downloaded files.

Application Note: This requirement ensures that if the user intentionally (via clicking on a link) or unintentionally initiates the download of a file, the browser will intervene by, for example, opening a dialog box that presents the user with the option to either save the file to the file system or not download the file.

In this context, an executable is a file containing code for a software program that is invoked independent of and outside the context of the browser. It does not include mobile code, scripts, or add-ons.

Assurance Activity ⮟

TSS

The evaluator shall examine the TSS to ensure that it describes the behavior of the browser when a user initiates the download of a file.

Guidance

The evaluator shall examine the operational guidance to ensure it describes the dialog box that appears when a download is initiated and the implications of the options presented by the dialog box.

Tests

The evaluator shall perform the following test:

- ***Test 1:*** *The evaluator shall navigate to a website that hosts files for download including executables and shall attempt to download and open several of these files. The evaluator shall verify that the browser always presents a dialog box with the option to either download the file to the file system or to discard the file.*

FPT_INT_EXT.1 Interactions with Application Reputation Services

FPT_INT_EXT.1.1

The browser shall utilize an application reputation service to prevent downloading of malicious applications.

This is an objective requirement.

Application Note: An application reputation service is an online service that identifies malicious applications; it is used to detect such applications prior to downloading them. Using a reputation service would require configuration of the trusted service to be used. The quality of the reputation service may fall outside of the scope of the evaluation.

Assurance Activity ⮟

TSS

The evaluator shall examine the TSS to ensure it describes the browser's use of application reputation services in detecting malicious applications.

Guidance

The evaluator shall examine the operational guidance to ensure it describes the browser's support for use of an application reputation service, including which services the browser supports by default (if any) and whether additional services can be configured. The operational guidance shall include steps for how to configure the application reputation service.

Tests

The evaluator shall perform the following test:

- ***Test 1:*** *The evaluator shall configure the browser to enable the use of one or more application reputation services per the operational guidance. The evaluator shall initiate a connection with a website that attempts to download an application to the browser while sniffing the network traffic using a network protocol analyzer. The evaluator shall inspect the captured network traffic and shall verify that the browser initiates a connection to the configured application reputation service(s) before initiating the download.*

FPT_INT_EXT.2 Interactions with URL Reputation Services

FPT_INT_EXT.2.1

The browser shall utilize a URL reputation service to prevent connections with malicious websites.

This is an objective requirement.

Application Note: A URL reputation service is an online service that identifies websites with malicious or phishing content applications; it is used to detect such websites prior to allowing users to access them. The goal of this requirement is to ensure that the browser is prevented from establishing connections with known-bad sources of malware on the Internet. The specifics of the sequence of actions taken before a block decision is made may depend upon the specific implementation of the browser. For example, some browsers might implement the check for malicious content by checking against the list of bad URLs provided by the URL reputation service in real time; others may download updated lists of bad URLs at browser startup, updating the list periodically from the URL reputation service(s) until the browser is terminated. Ultimately, the result should be that the browser blocks the connection to the bad URL.

Assurance Activity ⮟

TSS

The evaluator shall examine the TSS to ensure it describes the browser's use of a URL reputation service in detecting malicious websites.

Guidance

The evaluator shall examine the operational guidance to ensure it describes the browser's support for use of URL reputation services, including which services the browser supports by default (if any) and whether additional services can be configured. The operational guidance shall include steps for how to configure the URL reputation service.

Tests

The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall configure the browser to enable the use of one or more URL reputation services per the operational guidance. The evaluator shall initiate a connection with a known good website while sniffing the network traffic using a network protocol analyzer. The evaluator shall inspect the captured network traffic and shall verify that the browser initiates a connection to the configured URL reputation service(s).*
- ***Test 2:*** *The evaluator shall configure the browser to enable the use of one or more URL reputation services per the operational guidance. The evaluator shall initiate a connection with a known malicious website that is identified by one or more of the URL reputation services while sniffing the network traffic using a network protocol analyzer. The evaluator shall verify that a warning appears alerting that the website is known to be malicious and the browser is not allowed to connect. The evaluator shall inspect the captured network traffic and shall verify that the browser initiates a connection to the configured*

URL reputation service(s) and retrieved an updated list of malicious URLs with the tested website being on the list.

FPT_MCD_EXT.1 Mobile Code

FPT_MCD_EXT.1.1

The browser shall support the capability to execute signed [**selection**:

ActiveX,
Flash,
Java,
ActionScript,
*[**assignment**: other mobile code types supported by the browser]* ,
no

] mobile code.

FPT_MCD_EXT.1.2

The browser shall provide the user with the option to discard unsigned, untrusted or unverified [**selection**:

ActiveX,
Flash,
Java,
ActionScript,
*[**assignment**: other mobile code types supported by the browser]*

] mobile code without executing it.

Application Note: The ST author must specify all mobile code types for which the browser provides this support.

An authorized signer may directly sign the code itself, or the code may be delivered over an authenticated HTTPS connection with an authorized entity.

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure it lists the types of signed mobile code that the browser supports. The TSS shall describe how the browser handles unsigned mobile code, mobile code from an untrusted source, and mobile code from an unverified source.

Guidance

The evaluator shall examine the operational guidance to verify it provides configuration instructions for each of the supported mobile code types. The operational guidance shall also describe the alert that the browser displays to the user when unsigned, untrusted, or unverified mobile code is encountered and the actions the user can take.

Tests

The evaluator shall perform the following test for each mobile code type specified in the TSS:

- ***Test 1:*** *The evaluator shall construct web pages containing unsigned, correctly authenticated, and incorrectly authenticated mobile code and ensure that the browser alerts the user when it encounters mobile code that fails to authenticate and provides the user with the option to discard the mobile code without executing it, but does execute signed mobile code that properly authenticates.*

FPT_AON_EXT.1 Support for Only Trusted Add-ons

FPT_AON_EXT.1.1

The browser shall include the capability to load [**selection**: *trusted add-ons, no add-ons*] .

Application Note: FPT_AON_EXT.2 depends upon the selection made here. If the browser does not include support for installing only trusted add-ons, this requirement can be met by demonstrating the ability to disable all support for add-ons as specified in FMT_MOF_EXT.1. Cryptographic verification (i.e., trust) of add-ons is tested in FPT_AON_EXT.2.1

Assurance Activity ⮟

TSS

The evaluator shall verify that the TSS describes whether the browser is capable of loading trusted add-ons.

Guidance

The evaluator shall examine the operational guidance to verify that it includes instructions on loading trusted add-on sources.

Tests

The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall create or obtain an untrusted add-on and attempt to load it. The evaluator shall verify that the untrusted add-on is rejected and cannot be loaded.*
- ***Test 2:*** *The evaluator shall create or obtain a trusted add-on and attempt to load it. The evaluator shall verify that the trusted add-on loads.*

FPT_AON_EXT.2 Trusted Installation and Update for Add-ons

FPT_AON_EXT.2.1

The browser shall [**selection**: *provide the ability, leverage the platform*] to provide a means to cryptographically verify add-ons using a digital signature mechanism and [**selection**: *published hash, no other functions*] prior to installation and update.

This is a selection-based requirement. Its inclusion depends upon selection in FPT_AON_EXT.1.1.

FPT_AON_EXT.2.2

The browser shall [**selection**: *provide the ability, leverage the platform*] to query the current version of the add-on.

This is a selection-based requirement. Its inclusion depends upon selection in FPT_AON_EXT.1.1.

FPT_AON_EXT.2.3

The browser shall prevent the automatic installation of add-ons.

This is a selection-based requirement. Its inclusion depends upon selection in FPT_AON_EXT.1.1.

Assurance Activity ⮟

TSS

The evaluator shall examine the TSS to verify that it states that the browser will reject add-ons from untrusted sources.

Guidance

The evaluator shall examine the operational guidance to verify that it includes instructions on how to configure the browser with trusted add-on sources.

Tests

The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall create or obtain an add-on signed by a trusted source and attempt to install it. The evaluator shall verify that the signature on the add-on is valid and that the add-on can be installed.*
- ***Test 2:*** *The evaluator shall create or obtain an add-on signed with an invalid certificate and attempt to install it. The evaluator shall verify that the signed add-on is rejected and cannot be installed.*
- ***Test 3:*** *The evaluator shall create or obtain an add-on signed by a trusted source, modify the add-on without re-signing it, and attempt to install it. The evaluator shall verify that the signed add-on is rejected and cannot be installed.*

A. References

Identifier	Title
[CC]	Common Criteria for Information Technology Security Evaluation - • Part 1: Introduction and General Model, CCMB-2012-09-001, Version 3.1 Revision 4, September 2012. • Part 2: Security Functional Components, CCMB-2012-09-002, Version 3.1 Revision 4, September 2012. • Part 3: Security Assurance Components, CCMB-2012-09-003, Version 3.1 Revision 4, September 2012.
[AppPP]	Protection Profile for Application Software

B. Acronyms

Acronym	Meaning
CRL	Certificate Revocation List
CSRF	Cross Site Request Forgery
GPU	Graphics Processing Unit
HTML	HyperText Markup Language
HTML5	HyperText Markup Language version 5
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IETF	Internet Engineering Task Force
IPC	Inter-process communication
OCSP	Online Certificate Status Protocol
PDF	Portable Document Format
RFC	Request for Comment (IETF)
SaaS	Software as a Service
SSL	Secure Sockets Layer
TLS	Transport Layer Security
W3C	World Wide Web Consortium
XSS	Cross Site Scripting