MOBILE BROWSERS AND CLOUD GAMING

Appendix A: Browser comparison

22 November 2024

Competition & Markets Authority

Website: www.gov.uk/cma

The Competition and Markets Authority has excluded from this published version of the final report information which the inquiry group considers should be excluded having regard to the three considerations set out in section 244 of the Enterprise Act 2002 (specified information: considerations relevant to disclosure). The omissions are indicated by [✂].Non-sensitive wording is also indicated in square brackets.

Contents

Tables

Figures

1. Introduction and summary of findings

Introduction

1. This document is an Appendix to Section 4: The requirement to use Apple's WebKit browser engine on iOS of the Provisional Decision Report. It includes analysis of data gathered from a variety of public sources to demonstrate differences between browser engines and browsers, in relation to web compatibility and feature support, known browser vulnerabilities, and comparison of types of bugs and their resolution by each vendor. It also includes analysis of data provided by browser engine providers and browser vendors on security issues and stability.

2. The main browsers considered in this document are Chrome, Firefox, and Safari. Unless specified, this includes their associated browser engine Blink (Chrome), Gecko (Firefox) and WebKit (Safari).

3. Due to the constraints of available public test data, it is not possible to consistently represent data from solely desktop or mobile devices, or to consistently compare data for solely browser engines or browsers.

Summary of findings

Feature compatibility and support in browsers and browser engines

4. This Appendix considers several sources of data related to web feature compatibility in browsers, which indicate Safari offers the lowest level of support.

   (a) Web browsers offer differing levels of support for features included in HTML, CSS, and JavaScript scripting languages. Data from the 'Can I Use' and 'Web Platform Tests Project' websites show that Safari has the lowest count of supported features overall but has been increasing this number in recent years.

   (b) The 'Interop Project' and 'Microsoft Edge Top Web Developer Needs' websites focus on a subset of features identified as being important to browser engine vendors and web developers. The results from these websites show that Safari has improved its performance in the 2022 and 2023 Interop tests but offers the lowest level of support for features identified by Microsoft as being important to web developers.

Quantitative comparisons of security and bugs between web browsers

5. This Appendix considers several metrics relating to security vulnerabilities and bugs identified in each browser engine, including the time taken to fix the most severe issues, and the frequency of browser updates available to users.

6. Whilst WebKit had fewer identified vulnerabilities than Blink or Gecko, the time taken to fix vulnerabilities and bugs in WebKit was longer, and updates to WebKit were less frequent.

7. However, limitations around measuring vulnerabilities and comparability of publicly available bug data mean that it is difficult to draw firm conclusions on the relative security outcomes of different browser engines.

Mobile browser stability analysis

8. This Appendix considers data on crash rates for mobile browsers on iOS and Android, and therefore for the WebKit and non-WebKit versions of those mobile browsers. The analysis shows that crash rates across platforms are similar. There is therefore no evidence to suggest that WebKit performs significantly better or worse than other mobile browser engines with respect to stability.

Mobile browser privacy analysis

9. This Appendix includes results from PrivacyTests.org, an open-source initiative that tests the level of privacy of different mobile browsers, for mobile browsers on iOS and Android, and therefore for the WebKit and non-WebKit versions of those mobile browsers.

10. The results show that some mobile browsers perform better (pass more privacy tests) on iOS than on Android. No mobile browser passed more than 78 tests on iOS, whilst on Android, one mobile browser, Brave, passed 88. Another mobile browser, Firefox Focus, passed 77 tests on Android, compared to 60 on iOS. Safari on iOS passed 36 tests, whilst six mobile browsers on Android matched or exceeded this score.

11. A quantitative analysis of this kind may not capture that some tests could be more important than others, and therefore passing more tests does not necessarily indicate greater privacy overall. We also note that the test is developed by an employee of Brave and may therefore be influenced by its view of privacy.

2. Feature compatibility and support in browsers and browser engines

2.1 In web development, compatibility refers to the ability of a website to function and render as intended across various web browsers. It ensures that users with different browser preferences experience the website consistently and efficiently. This is achieved by adhering to established web standards for scripting languages like HTML, CSS, and JavaScript, while employing techniques to address potential browser-specific rendering differences.

2.2 These browser-specific rendering differences can be identified by examining feature support in browsers. The presence of features indicates a browser's capability to execute the specific functionalities implemented within a website. For example, a website might leverage an innovative video codec not supported by an older browser. This would result in the video being unavailable for users on that browser.

2.3 There are several publicly available sources for identifying compatibility and feature support in different browsers.

2.4 Not all features reported as available and supported will necessarily be in active use by developers on websites. Therefore, volume of features available in a browser is not an indicator of functionality being made available to, or used by, end users of websites.

Web Platform Tests Project (WPT)

2.5 The Web Platform Tests Project[1] runs tests for various browser technologies and based on the test results, provides assessments of compatibility and feature support of different browsers. It comprises a group of test suites for many web platform specifications including over 55,000 individual checks.

2.6 The main users of WPT are browser developers, who can check development versions of their browser against existing test suites and contribute new tests to the project which emulate new or changed features and functionality to see if these are supported by other browsers.

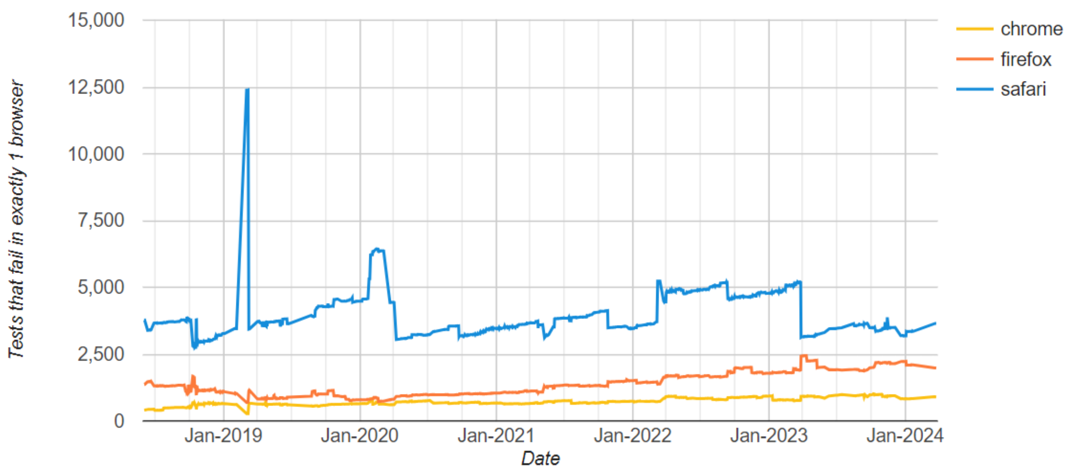
2.7 Many of the tests relate to web standards, including W3C, WHATWG and CSS Working Group specifications.[2] Some tests are for new features not yet widely adopted or formally recognised.

---

[1] Web Platform Tests Project (WPT), accessed on 21 October 2024.
[2] Web Platform Tests - test suite design, accessed on 21 October 2024.

2.8 For each browser, the line measures the number of tests that were failed by a given browser and passed by all other browsers each time the full set of test suites was run. This can broadly be interpreted as instances where the browser is not compatible while the other browsers are.

Figure 1.1: Browser-specific WPT failure scores for Chrome, Firefox, and Safari



Source: https://wpt.fyi/results/
Notes:
(1) Graph retrieved 20 March 2024.
(2) Graph shows WPT test results based on stable (rather than experimental) version of desktop browsers.

2.9 The blue Safari line (which represents any browser built on WebKit) is substantially and persistently higher than the yellow Chrome and red Firefox lines (representing browsers built on Blink and Gecko respectively). This indicates that WebKit has performed worse in terms of compatibility with these tests than Blink and Gecko over this period. For example, a result of 3,000 for Safari means that during that specific run of test suites, there were 3,000 tests which Safari failed, but all other browsers passed.

2.10 The drop in the Safari line in March 2023 may be attributed to the release of Safari 16.4,[3] which incorporated many updates, including over 100 additions and more than 270 fixes. By contrast, the previous update (16.3) had 1 addition and 23 fixes.

2.11 All the results reported in Figure 2.1 are for desktop versions of web browsers. There are no WPT tests run on iOS for any browser, including Safari.

2.12 On Android, test results are available for Chrome and Firefox. Table 2.1 shows that the percentage of tests passed were similar or identical for both browsers on desktop and mobile.

[3] Safari 16.4 Release notes, accessed on 21 October 2024.

Table 2.1: Chrome and Firefox WPT test results on desktop and mobile

Browser version	*Desktop OS (Linux)*	*Mobile OS (Android)*
Chrome 127	97%	96%
Firefox 128	96%	96%

Source: https://wpt.fyi/results/
Notes:
(1) Data was retrieved on 20 May 2024 based on Experimental results (Stable unavailable)
(2) WPT test date 20 May 2024, test ID 5e6793f

2.13 Apple's EU Web Browser Engine Entitlement requires browser apps that wish to use an alternative browser engine to pass a 'minimum of 90% of Web Platform Tests'.[4] The actual number of tests may vary – the requirement is further defined as 90% of the subtests that have been executed by any browser shown on the wpt.fyi front page. It is also a requirement that the app meets this pass percentage on an operating system that wpt.fyi supports, which does not include iOS. There are no tests currently run on wpt.fyi for browsers on iOS so it is not possible to see how an alternative browser might compare to Safari on iOS.

2.14 Failures observed for browsers in WPT may be caused by errors in the testing process itself.[5] Figure 2.1 shows trends over a five-year period which minimises the likelihood of test platform issues affecting the overall results.

The Interop Project

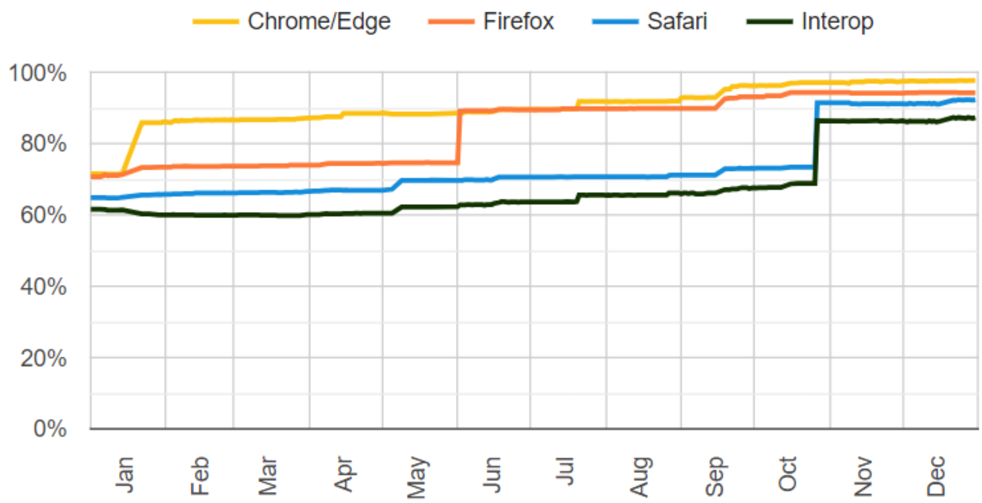
2.15 Another assessment provided by WPT is the Interop Project,[6] a collaboration between organisations that implement web technology in browser engines. It defines a metric based on a set of web technologies that it collectively believes to be important to improve interoperability. This metric publicly keeps track of that work by using automated tests to score how much progress each participating browser has made reaching the shared goals.

2.16 The Interop Project tracks key areas that represent the most painful compatibility bugs (i.e. a small subset of the features considered in Figure 2.1 above). These Web Compat Focus areas are agreed by consensus of participating organisations at the start of the project each year.

2.17 The scores represent how well browser engines are doing on the annual Compat Focus Areas (a higher score being better), with the black line representing the number of feature tests that pass in all browsers, to show overall interoperability.

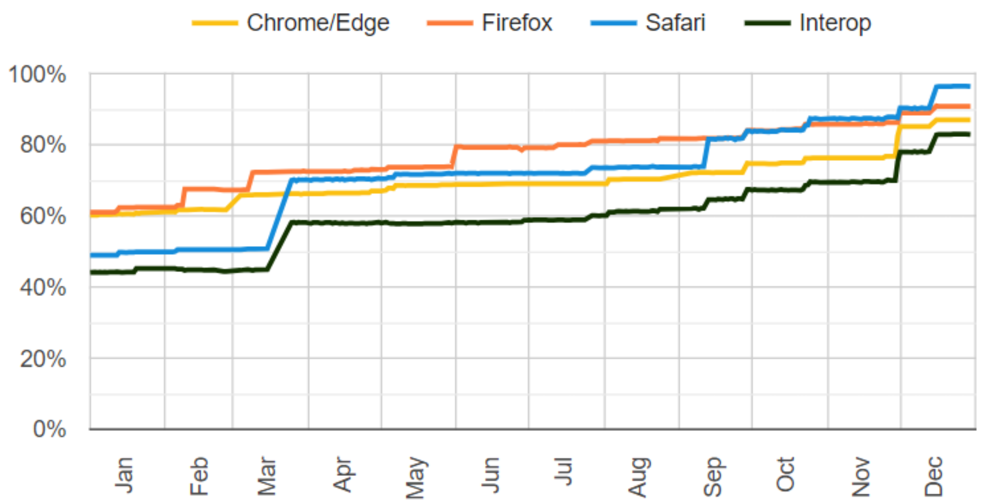
[4] Apple: Using alternative browser engines in the European Union (Web Browser Engine Entitlement), accessed on 21 October 2024.
[5] Examples of test completion errors in WPT, accessed on 21 October 2024.
[6] Interop Project, accessed on 21 October 2024.

Figure 1.2: Interop 2021 results



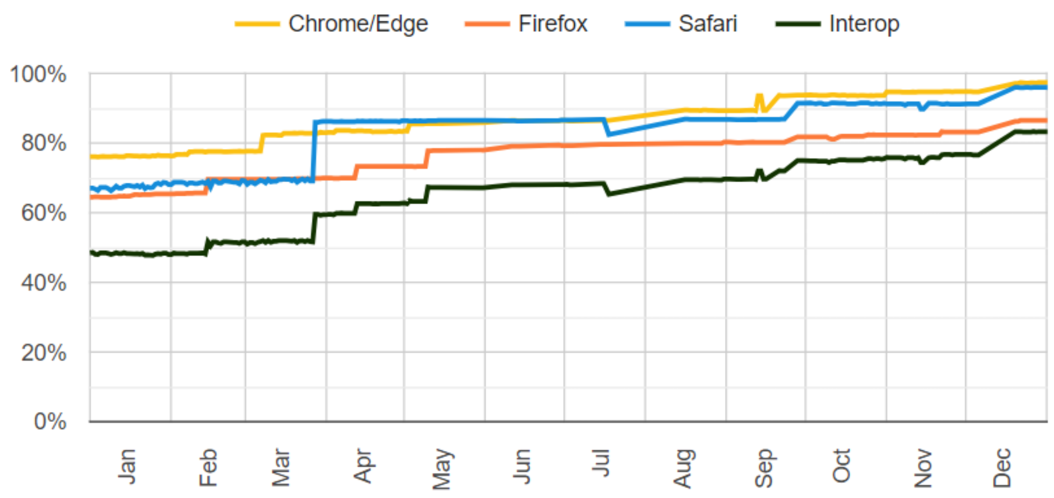
Source: https://wpt.fyi/interop-2021?stable, accessed on 21 October 2024.
Notes:
(1) Graph based on Stable results
(2) Safari updates in 2021 were released in April, September, and December

Figure 1.3: Interop 2022 results



Source: https://wpt.fyi/interop-2022?stable, accessed on 21 October 2024.
Notes:
(1) Graph based on Stable results
(2) Safari updates in 2022 were released in March, May, July, September, October, and December

Figure 1.4: Interop 2023 results



Source: https://wpt.fyi/interop-2023?stable, accessed on 21 October 2024.
Notes:
(1) Graph based on Stable results
(2) Safari updates in 2023 were released in Jan, March, May, July, September, October, and December
(3) The release of Safari 16.4 in March 2023 included over 100 additions to supported features and functionality, which increased Safari's score and improved overall interoperability

2.18 Over the past three years of Interop project work, all browsers demonstrate progress towards the annual compatibility targets during the year. The overall interoperability score has been just over 80% at the end of each year, indicating that whilst browsers have made improvements against the pre-determined list of features, they have not all pursued the same elements.

2.19 The Web Platform Test project, of which the Interop project is a subset of tests, does not run tests on iOS for any browser, including Safari. The need to add testing on mobile browsers was identified as one of the 'Focus Areas' in the 2023 Interop project but has not yet been implemented.[7]

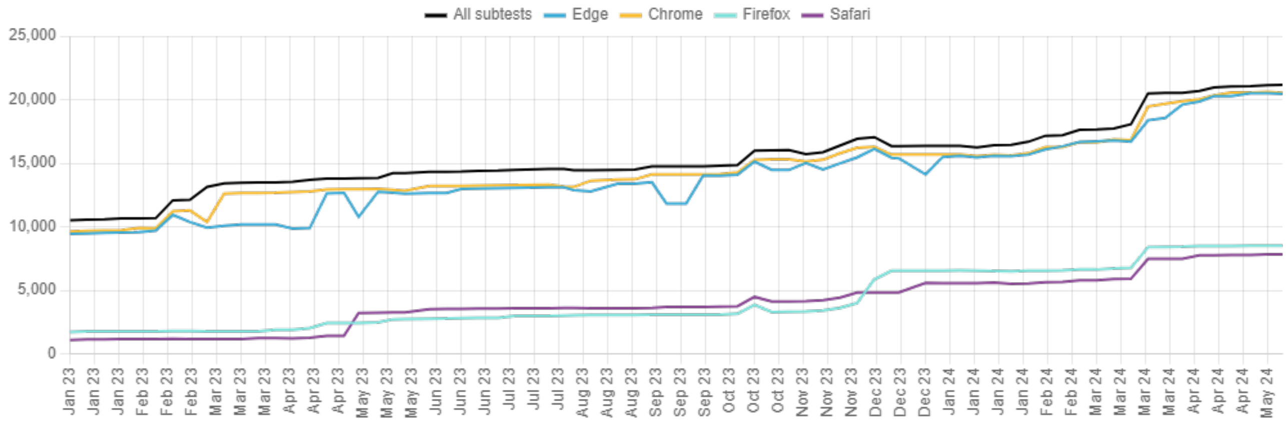
2.20 The Interop Project demonstrates that feature compatibility and support has continued to increase over the past few years, with all major browsers offering a substantial number of supported features as documented on the Can I Use website.

Microsoft Edge – 2024 web platform top developer needs

2.21 Similarly to the Interop project, the Microsoft Edge team have curated a set of wpt.fyi feature subtests representing top developer pain points and interoperability gaps, based on feedback received from web developers.[8]

[7] Mobile Testing Investigation in Interop, accessed on 21 October 2024.
[8] Microsoft Edge - 2024 web platform top developer needs, accessed on 22 October 2024.

Figure 1.5: Number of 'top developer needs' subtests passed for each browser

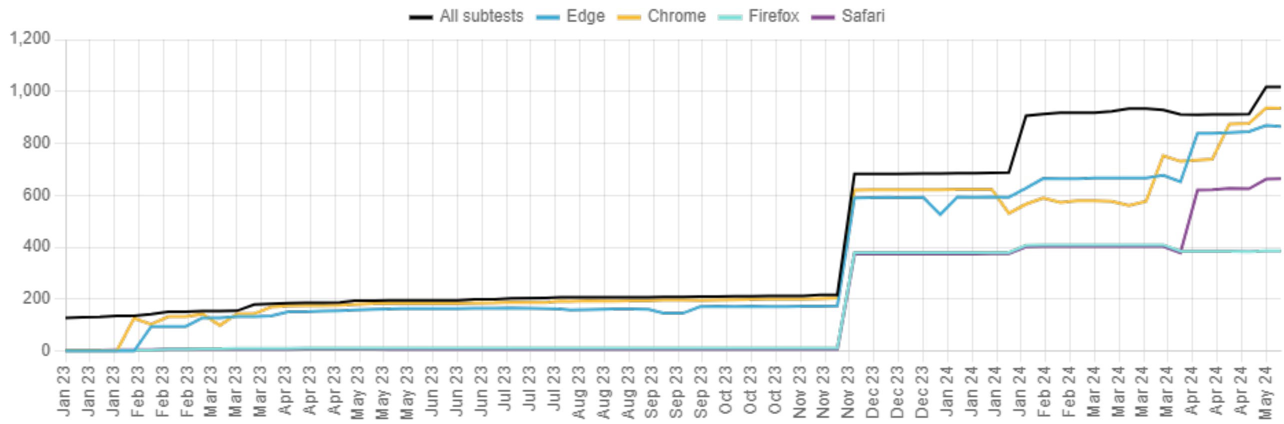


Source: https://microsoftedge.github.io/TopDeveloperNeeds/
Notes:
(1) The black line represents the total number of subtests performed for this feature
(2) Data retrieved 22 May 2024
In addition to an overall indicator of cross-browser progress towards support for these 'top developer needs' features shown in Figure 1.5, it is possible to view progress for individual features.

2.22 For example, the 'View Transitions' API allows creation of animated visual transitions between different states of a document, or between different documents. Figure 2.6 shows that support for this feature has been increasing in all browsers since November 2023.

2.23 Unlike other areas of this Appendix, Microsoft Edge browser has been included in this data as Microsoft is the curator of these wpt.fyi subtests. Edge and Chrome are both based on Blink which explains the comparable results.

Figure 1.6: Number of subtests passed by each browser for the View Transitions API



Source: https://microsoftedge.github.io/TopDeveloperNeeds/
Notes:
(1) The black line represents the total number of subtests performed for this feature
(2) Data retrieved 22 May 2024

Test 262 for JavaScript engines

2.24 The Test262 website[9] tests standards compliance of different JavaScript engines, as an example of a way to assess browser engines that goes beyond the Web Platform Tests project. JavaScript engines form part of browser engines and are responsible for interpreting and executing JavaScript code used within webpages displayed by browsers.

2.25 Test262 is the official conformance test suite for ECMAScript, the programming language behind JavaScript. It includes a collection of test cases maintained by the TC39 committee responsible for the evolution of ECMAScript. These tests are used by browser developers to verify their implementations correctly interpret and execute JavaScript code as defined in the standards.

2.26 Test262 is not intended to provide easily interpretable results for public use but does offer insight into the types of testing that web browser developers commonly undertake.

2.27 Browser developers use Test262 to identify and fix compatibility issues in their browsers. They can run the test suite to see if their browser passes all the tests, indicating proper ECMAScript compliance. In addition to experimental and unreleased JavaScript feature tests, Test262 checks against core specifications. The most recent core ECMAScript specification assessed by Test262 is ES2024.

2.28 Figure 2.7 shows the percentage of ECMAScript ES2024 tests passed by the JavaScript engines used in WebKit (JavaScriptCore), Blink (V8) and Gecko (SpiderMonkey).

Figure 1.7: Test262 results for 2024 ECMAScript

Source: https://test262.fyi/
Notes:
(1) Graph retrieved 18 October 2024

2.29 Apple's EU Web Browser Engine Entitlement[10] requires apps to pass a minimum of 80% of Test262 tests on an iOS device or Mac with Apple silicon. The Web

[9] Test262, accessed on 22 October 2024.
[10] Apple: Using alternative browser engines in the European Union (Web Browser Engine Entitlement), accessed on 22 October 2024.

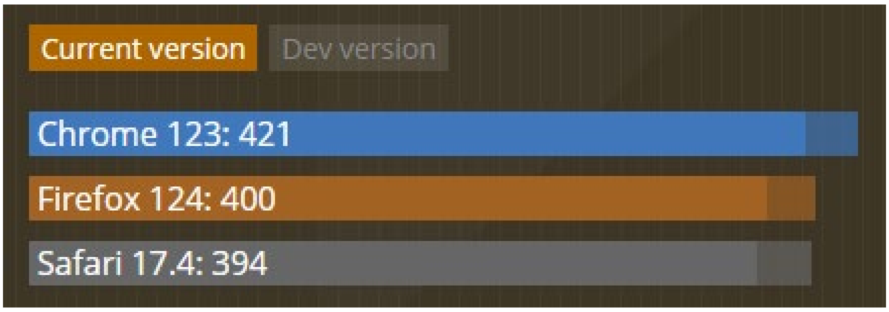
Browser Engine Entitlement does not specify which categories the 80% score must be obtained in.

Can I Use feature compatibility

2.30 The 'Can I Use' (caniuse) website[11] lists the status of support for different features in web browsers.

2.31 Figure 2.8 shows a summary of the main desktop browser scores based on all features tracked on 'caniuse', indicating that Safari supports fewer features than Chrome and Firefox. These scores represent tallies of features that caniuse tracks.

Figure 1.8: Summary of desktop browser feature support on caniuse on 14 March 2024



Source: https://caniuse.com/
Notes:
(1) Graph retrieved on 14 March 2024
(2) The fully opaque part represents supported features; the semi-transparent part represents partial support.

2.32 By comparison, in Appendix F of the Mobile Ecosystems Market Study,[12] the same summary of desktop browsers showed that Chrome (version 100) offered full support for 397 features, Firefox (version 98) 375 features, and Safari (version 15.4) 354 features. Whilst Safari has added more features (40 additions) than Chrome (24 features) and Firefox (25 features) since April 2022, it still has the lowest number of supported features overall.

2.33 When considering mobile browsers specifically, support for various categories of features varies, as shown in Table 2.2. Overall, only 66% of features available to web developers are fully supported in all mobile browsers. Features that are not fully supported may either be completely unsupported, or only supported for specific combinations of browser and device versions.

Table 2.2: caniuse overview of supported features across all mobile browsers, by category

Feature category

[11] Can I use… Support tables for HTML5, CSS3, etc, accessed on 22 October 2024.
[12] Mobile Ecosystems Market Study - Appendix F: browser engines, accessed on 22 October 2024.

	Total features	Fully supported	% of total features fully supported
All	543	357	66
CSS	190	136	72
HTML5	83	57	69
JavaScript	40	36	90
JavaScript APIs	140	75	54
Security	32	20	63
SVG	11	8	73
Other	136	81	60

Source: https://caniuse.com/
Notes:
(1) Data was retrieved on 14 March 2024
(2) Total features refers to the total number of features listed on https://caniuse.com

2.34 Table 2.3 shows the same list of categories, with percentage of support by mobile browser.

Table 2.3: caniuse percentage of supported features by category across mobile browsers

			%
Feature category			
	Chrome	*Safari*	*Firefox*
All	82	75	74
CSS	82	80	79
HTML5	81	78	77
JavaScript	90	90	93
JavaScript APIs	83	61	65
Security	81	75	66
SVG	82	82	82
Other	80	71	69

Source: https://caniuse.com/
Notes:
(1) Data was retrieved on 14 March 2024
(2) The browsers used for this comparison were Chrome 122 for Android (Blink), Safari and other browsers on iOS 17.4 (WebKit), Firefox 123 for Android (Gecko).

2.35 Table 2.3 shows that Chrome on Android has the highest percentage of supported features in all categories on caniuse. Safari on iOS comes second in most categories except JavaScript and JavaScript APIs, where Firefox on Android provides slightly more support than Safari on iOS.

2.36 While caniuse tracks a wide variety of features, it only covers a subset of all web technologies, so the scores are not 100% representative of any browser's capabilities.

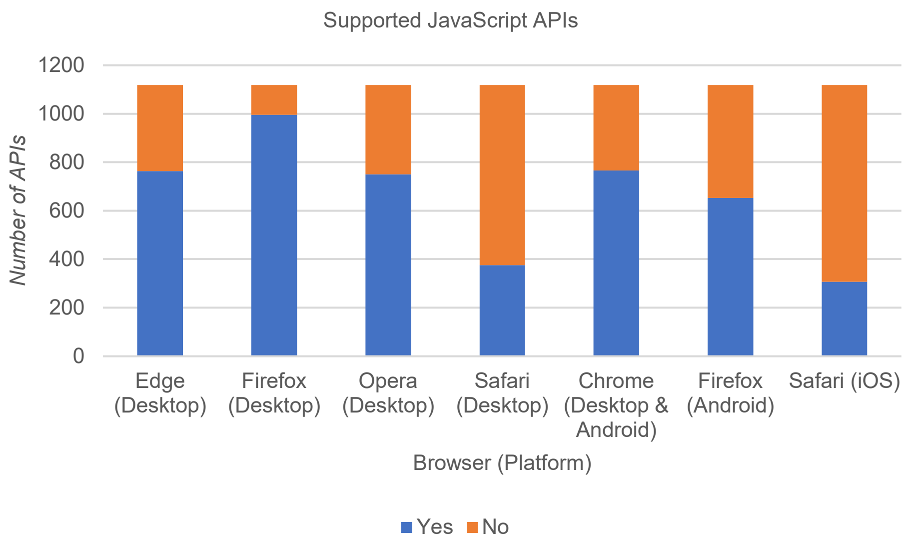
Browser support for Web Extension JavaScript APIs

2.37 An extension adds features and functions to a browser. It is created using familiar web-based technologies – HTML, CSS, and JavaScript. It can take advantage of the same web APIs as JavaScript on a web page, but an extension also has access to its own set of JavaScript APIs. This means that developers can do more in an extension than with code in a web page.

2.38 The Mozilla Developer Network (mdn) web docs site provides a detailed listing of current browser support for Web Extension JavaScript APIs.[13]

2.39 On 24 March 2024 there were 1,118 APIs and their associated parameters listed, with each item in the list given a Yes/No status to indicate whether it is supported. Support has been summarised in Figure 2.9 below.

Figure 1.9: Browser support for Web Extension JavaScript APIs



Source: Mozilla Development Network
Notes:
(1) Data retrieved on 24 March 2024
(2) Chrome includes Chrome desktop and Chrome on Android
(3) Safari on iOS is distinct from Safari on desktop (MacOS)
(4) Firefox for Android represents the Gecko engine

2.40 Edge, Opera and Chrome all have similar support which is expected as they are all based on the Blink engine. Firefox on desktop has much greater support than Firefox on Android even though they are both based on Gecko.

2.41 Safari has the lowest support for web extension JavaScript APIs of all the desktop browsers. Safari on iOS, representing all the WebKit browsers, has the lowest

[13] MDN Web Docs: Browser support for JavaScript APIs, accessed on 22 October 2024.

support of all the browsers in this comparison. Chrome and Firefox on iOS are not represented in this data as they currently do not support web extensions on iOS.

2.42 Differences in web extension API support across browsers and platforms illustrates challenges for web developers who want to develop browser extensions offering consistent behaviour and functionality for users wherever they choose to use them.

3. Quantitative comparisons of security and bugs between web browsers

3.1 Security of web browsers can be examined by identifying associated bugs and vulnerabilities, and identifying how vendors respond to these issues.

3.2 A vulnerability is a specific weakness in the browser's code that can be leveraged by malicious actors. These weaknesses can be exploited to gain unauthorized access to a user's system, steal sensitive data, or inject malicious code. A zero-day vulnerability is one that is discovered and potentially exploited before the vendor has become aware of it.

3.3 In this section we have included vulnerability information from Google's Project Zero and Common Vulnerabilities and Exposures (CVE) data, including data for exploited vulnerabilities and average resolution times for critical vulnerabilities.

3.4 A bug is a deviation from the intended behaviour of the web browser. This can manifest as unexpected rendering issues, crashes, or features malfunctioning. Bugs range in severity from minor inconveniences, like a misplaced button, to critical errors that prevent core functionalities. It is important to note that not all bugs translate into vulnerabilities. However, certain bugs can create exploitable openings for attackers.

3.5 In this section we have included bug information from publicly available bug trackers maintained by browser vendors. Not all bugs in these listings will relate specifically to security concerns. In each case we have identified a subset of the most serious bugs, as determined by vendor categorisation, to examine vendor responsiveness in more detail.

Google Project Zero

3.6 Project Zero is a team of security analysts employed by Google tasked with finding zero-day vulnerabilities, not only in Google software but any other software used by Google users.

3.7 In their own analysis, the Project Zero team have noted that their research on open-source browsers enables them to follow the timeline of a vulnerability from discovery to fix.[14]

3.8 Table 3.1 shows the number of vulnerabilities discovered by the Google Project Zero Team between 2019 and 2023 for the three major open-source browsers.

[14] Google Project Zero: A walk through Project Zero metrics, accessed on 22 October 2024.

The number in brackets is the mean time in days between the vulnerability being reported and marked as 'Fixed' in the Project Zero issue list.

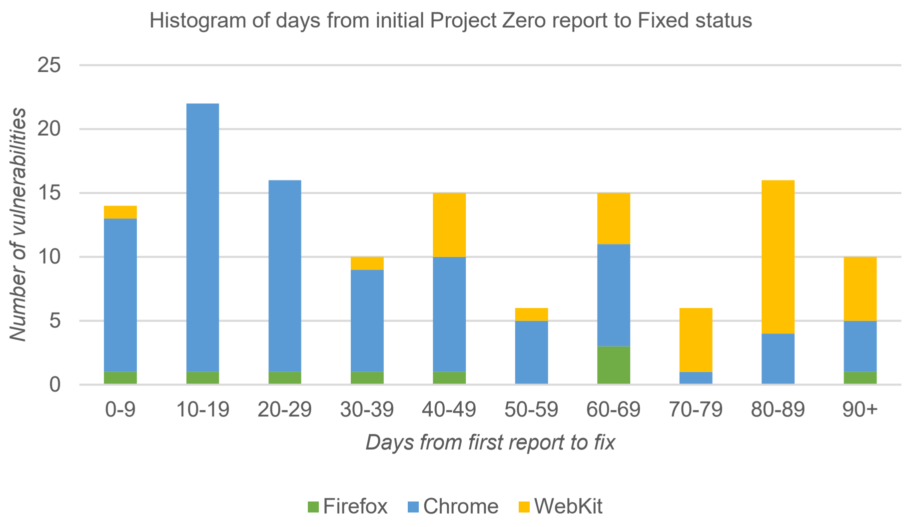
Table 3.1: Zero-day vulnerabilities and average fix time for vulnerabilities discovered by Google Project Zero

Browser			*Year*			*Total*
	2019	*2020*	*2021*	*2022*	*2023*	
Chrome	24 (39)	11 (14)	15 (51)	18 (32)	18 (32)	86 (35)
WebKit	25 (77)	2 (70)	6 (62)	2 (45)	0	35 (72)
Firefox	6 (33)	2 (54)	1 (92)	0	0	9 (44)

Source: https://bugs.chromium.org/p/project-zero/issues/list
Notes:
(1) Data retrieved on 19 March 2024
(2) Project Zero considers WebKit to represent all iOS browsers including Safari (https://googleprojectzero.blogspot.com/2022/02/a-walk-through-project-zero-metrics.html)

3.9 Whilst Chrome has the largest volume of vulnerabilities discovered, the average time for the vulnerability to be marked as 'Fixed' is 35 days, less than half the time taken for WebKit vulnerabilities (72). This is illustrated in Figure 3.1.

Figure 1.10: Histogram of days from vulnerability reported by Google Project Zero to status Fixed



Source: https://bugs.chromium.org/p/project-zero/issues/list
Notes:
(1) Data retrieved on 19 March 2024

(2) Date range January 2019 to December 2023
(3) 'Fixed' is defined by Project Zero as a patch being created and added into the source code. There may be additional time before this patch is cascaded to all users.

3.10 The Google Project Zero Team vulnerability data does not cover all zero-day vulnerabilities affecting software products, only the ones that their own team has identified.

3.11 No zero-day vulnerabilities were discovered by the Project Zero Team for Firefox after January 2021 and no zero-day vulnerabilities were discovered by the Project Zero Team for WebKit after February 2022, even though such vulnerabilities have been disclosed. One explanation for this might be that the other vendors were finding and fixing vulnerabilities before the Google Project Zero Team discovered them.

Common Vulnerabilities and Exposures (CVE)

3.12 The Common Vulnerabilities and Exposures (CVE) system provides a reference method for publicly known information-security vulnerabilities and exposures. Vulnerabilities can be submitted by individuals or organisations to one of several organisations who are authorised to assign CVE IDs to vulnerabilities, and this CVE ID subsequently enables the same issue to be referred to consistently across multiple reporting and recording systems.

3.13 Each vulnerability is assigned a numerical score out of ten which correlates to a category rating of Low, Medium, High or Critical. This Common Vulnerability Scoring System (CVSS) is a method used to supply a qualitative measure of severity.

3.14 The CVE security scorecard website tracks known vulnerabilities in software products.[15]

3.15 We retrieved data for the period 1 January 2022 – 31 December 2023, for Chrome,[16] Safari[17] and Firefox.[18]

3.16 Table 3.2 shows the number of published vulnerabilities by CVSS score for each browser during 2022 and 2023.

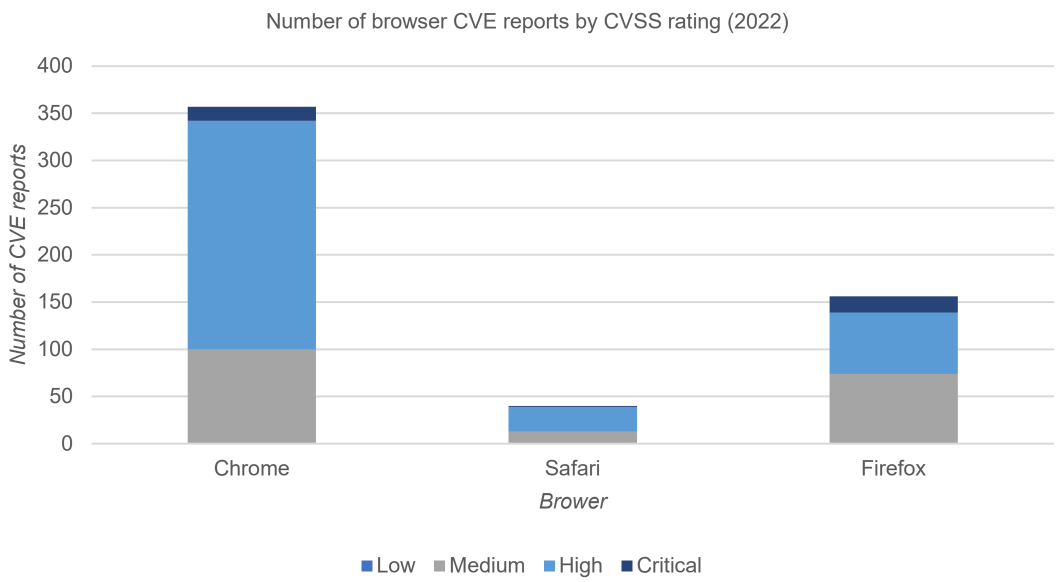
[15] CVE Vulnerability Database, accessed on 22 October 2024.
[16] CVE Vulnerability Database: Google Chrome product details, threats, and statistics, accessed on 22 October 2024.
[17] CVE Vulnerability Database: Apple Safari product details, threats, and statistics, accessed on 22 October 2024.
[18] CVE Vulnerability Database: Mozilla Firefox product details, threats, and statistics, accessed on 22 October 2024.

Table 3.2: Total vulnerabilities published on CVEdetails 2022 for each browser by CVSS Score

CVSS score	*Chrome*	*Safari*	*Firefox*
Low	0	0	0
Medium	100	13	74
High	242	26	65
Critical	15	1	17
Total	357	40	156

Source: https://www.cvedetails.com
Notes:
(1) Data retrieved on 21 March 2024

Figure 1.11: Number of CVE reports per browser and CVSS rating published during 2022



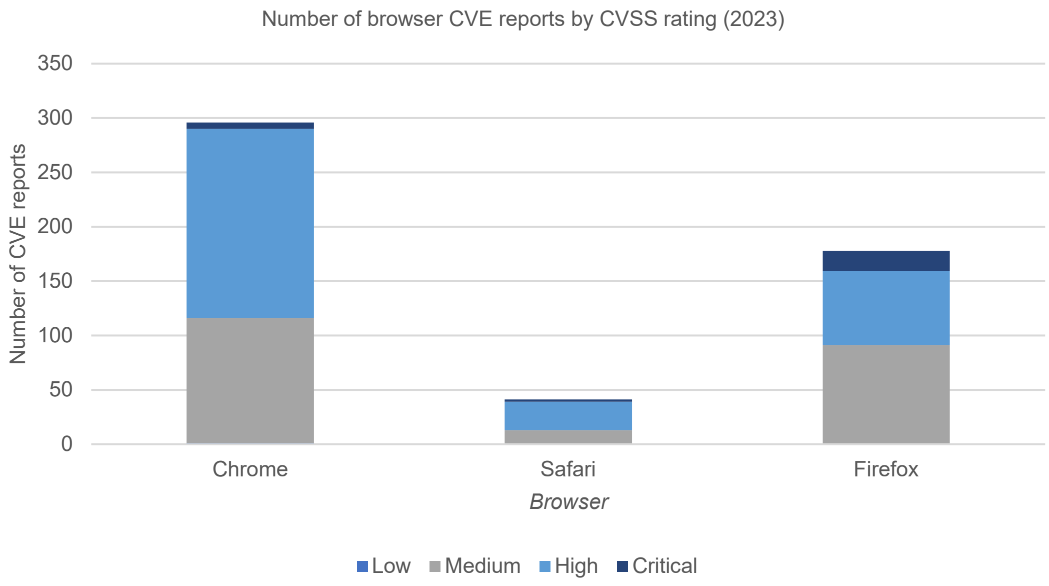
Source: https://www.cvedetails.com
Notes:
(1) Data retrieved on 21 March 2024

Table 3.3: Total vulnerabilities published on CVEdetails 2023 for each browser by CVSS score

CVSS score	*Chrome*	*Safari*	*Firefox*
Low	1	0	0
Medium	115	13	91
High	174	26	68
Critical	6	2	19
Total	296	41	178

Source: https://www.cvedetails.com
Notes:
(1) Data retrieved on 21 March 2024

Figure 1.12: Number of CVE reports per browser and CVSS rating published during 2023



Source: https://www.cvedetails.com
Notes:
(1) Data retrieved on 21 March 2024

Comparing public resolution times for Critical CVEs

3.17 During 2022 and 2023, there were 60 CVEs published with a CVSS score of Critical, the highest qualitative severity rating. Table 3.4 shows the average time in days between initial CVE report and release of a public update by the vendor, for each of the browsers reported in Tables 2.2 and 2.3.

Table 3.4: Average days between initial bug report and release of fix in product update

Browser	*Number of CVEs with critical CVSS score*	*Number of CVEs with public timelines*	*Average days from initial report to public update*
Chrome	21	19	84 days
Safari	3	0	Unknown*
Firefox	36	33	236 days

Source: https://www.cvedetails.com
** No bug details publicly available*
Notes:
(1) Data obtained 15 April 2024
(2) Each CVE was identified in the product's own issue tracker, to find the date the issue was first recorded for the product. This is usually earlier than the date the CVE information was released.
(3) The public update was recorded as the release date of the update containing the fix for the identified CVE.
(4) Some issues were not available for public viewing, so it was not possible to determine the date they were first reported. These were excluded from the average calculation.

Exploited vulnerabilities

3.18 The US Cybersecurity and Infrastructure Security Agency (CISA) maintains the authoritative source of vulnerabilities that have been exploited in the wild.

Exploitation refers to the use of malicious code by an individual to take advantage of a vulnerability.[19]

3.19 During 2022 and 2023 there were over 50 CVE vulnerabilities related to browsers that were known to have been exploited.

Table 3.5: Exploited vulnerabilities by browser and engine

Year	*Chrome only*	*All Chromium browsers*	*WebKit*	*Firefox*
2022	3	17	4	7
2023	0	7	11	1
Total	3	24	15	8

Source: https://www.cisa.gov/known-exploited-vulnerabilities-catalog
Notes:
(1) Data retrieved on 10 April 2024
(2) Exploit descriptions distinguish between the Chrome browser and Chromium engine, which impacts all Chromium browsers.

Limitations of vulnerabilities data

3.20 A single CVE record may include one or more vulnerabilities, so the total number of CVEs may not reflect the true number of vulnerabilities identified. Additionally, the list only includes vulnerabilities that are publicly disclosed.

3.21 CVE records do not generally differentiate between vulnerabilities on different devices and platforms, rather they are organised by Vendor and Product. As such, it is not possible to make direct comparisons between mobile browsers using CVE data.

Blink bug fixes

3.22 The Blink Issue tracker records all bugs that have been identified in the Blink engine. It is important to remember that bugs do not necessarily correlate to security issues, they represent any aspect of the product that is not behaving as expected. In Google issue trackers, 'fixed' means that the bug has been fixed in the source code.[20]

3.23 Over 5,000 bugs were marked as fixed during the period 1 January 2022 – 31 December 2023.

3.24 Bugs in the tracker are rated by priority and severity.

3.25 Priority refers to the urgency with which the bug needs to be fixed, with a priority rating from P0 (highest priority) to P4 (lowest priority). Google describes a P0 issue as one that 'needs to be addressed immediately and with as many resources

[19] Known Exploited Vulnerabilities Catalog | CISA, accessed on 22 October 2024.
[20] Report and track bugs on Android, accessed on 22 October 2024.

as is required. Such an issue causes a full outage or makes a critical function of the product to be unavailable for everyone, without any known workaround'.[21]

3.26 Table 3.6 shows that only a small number of Blink bugs were assigned the most serious priority level (P0).

Table 3.6: Fixed Blink bugs by priority

Bug priority	*Year*		*Total*
	2022	*2023*	
P0	6	7	13
P1	801	705	1506
P2	928	971	1899
P3	785	825	1610
P4	0	0	0
Total	2520	2508	5028

Source: https://issuetracker.google.com
Notes:
(1) Data retrieved on 21 March 2024
(2) Data filtered to include Blink and its sub-categories
(3) P0 = highest priority (e.g. product unusable), P4 = lowest priority

3.27 Severity refers to the impact of a bug on functionality or end-user experience, including security implications. It measures how severe the issue is and how critical it is to fix it. Table 3.7 shows that very few bugs were high severity.

Table 3.7: Resolved Blink bugs by severity

Bug severity	*Year*		*Total*
	2022	*2023*	
S0	0	0	0
S1	1	6	7
S2	0	1	1
S3	0	1	1
S4	2519	2500	5019
Total	2520	2508	5028

Source: https://issuetracker.google.com
Notes:
(1) Data retrieved on 21 March 2024
(2) Data filtered to include Blink and its sub-categories
(3) S0 = highest severity, s4 = lowest severity

3.28 A bug might be quite severe, but only affecting an older, little-used version of a product, so it is not assigned as high a priority as a less severe bug affecting a more widely used version. Therefore, priority may be considered a stronger indicator of bugs which had a significant impact on the product.

[21] How-to guide for Google Issue Tracker, issued 22 October 2022.

3.29 The most serious bugs will be (P0), usually blockers that render the product unusable. When these are assigned a high severity level, this indicates that they were a significant issue for Blink. Table 3.8 shows the average time in days between bug creation and marking as fixed, where the bug has been assigned a priority rating P0 and categorised S0 – S3 severity.

Table 3.8: Highest priority/severity Blink bugs with average days to fix

Resolved P0 / S0-S3 bugs	*Total*	*Average days to fix*
Bugs fixed in 2022	7	4
Bugs fixed in 2023	16	10
Total bugs 2022-23	23	7

Source: https://issuetracker.google.com
Notes:
(1) Data retrieved on 21 March 2024
(2) Data filtered to include Blink and its sub-categories

3.30 Some bug reports may not be publicly available, and thus excluded from the public data completely, for example Bink bug ID 40058035 discovered when researching CVE resolution timelines.

WebKit bug fixes

3.31 The WebKit bug tracker records all bugs reported for the WebKit engine. Examining the period 1 January 2022 – 31 December 2023, there were 650 bug reports created where the hardware was specified as iPhone/iPad.[22]

3.32 128 bugs of these were marked as fixed, with the remaining bugs flagged as duplicates or given another status which meant that a fix was not considered necessary. In the WebKit bug tracker, a status of ‘Fixed’ means that the fix has been added into the source code and tested.[23]

3.33 WebKit bugs are labelled with a priority grouping from P1 – P4[24] with P1 being the most serious priority to fix. Table 3.9 shows there were 128 bugs which were designated as high priority to fix in 2022 and 2023.

Table 3.9: Fixed WebKit bugs by priority

Priority	*Year*		*Total*
	2022	*2023*	
P1	3	0	3
P2	60	65	125

[22] WebKit Bugzilla bug tracker, accessed on 22 October 2024.
[23] WebKit Bugzilla bug status, accessed on 22 October 2024.
[24] WebKit Bugzilla bug prioritisation, accessed on 22 October 2024.

Total	63	65	128

Source: https://bugs.webkit.org/
Notes:
(1) Data retrieved on 18 March 2024
(2) P1 = highest priority (e.g. product unusable)
(3) No fixed bugs were identified from other Priority categories.

3.34 Bugs are assigned a severity rating according to how serious they are, with 'Blocker', 'Critical' and 'Major' describing those with the most significant impact.[25]

Table 3.10: Fixed WebKit bugs by severity

Severity	*Year*		*Total*
	2022	*2023*	
Blocker	3	6	9
Critical	5	7	12
Major	7	11	18
Minor	0	2	2
Normal	48	39	87
Total	63	65	128

Source: https://bugs.webkit.org/
Notes:
(1) Data retrieved on 18 March 2024
(2) Blocker = highest severity

3.35 The most serious bugs are assigned the status Blocker, Critical or Major. When these are assigned a high severity level, this indicates that they were a significant issue for WebKit. Table 3.11 shows the average time in days between bug creation and resolution, where the bug has been assigned a severity rating of status Blocker, Critical or Major.

Table 3.11: Blocker, Critical or Major WebKit bugs with average resolution time in days

	Total fixed bugs	*Average days to fix*
Fixed in 2022	15	74
Fixed in 2023	24	116
Total fixed 2022-23	39	95

Source: https://bugs.webkit.org/
Notes:
(1) Data retrieved on 18 March 2024

[25] WebKit bug severity, accessed on 22 October 2024.

3.36 Some bug reports may not be publicly available, and thus excluded from the public data completely, for example WebKit bug ID 261544 was discovered when researching CVE resolution timelines.

Firefox bug fixes

3.37 The Mozilla bug tracker Bugzilla[26] records all bugs reported for the Firefox browser. Examining the period 1 January 2022 – 31 December 2023, there were 7,870 bugs reported fixed for Firefox across all devices. There is no category to select for Gecko engine itself.

3.38 In Bugzilla, a status of 'Fixed' means that the fix has been added into the source code and tested.[27]

3.39 Firefox bugs are labelled with a priority grouping from P1 – P5[28] with P1 being the most urgent, stating that it should be fixed 'in the current release cycle'. Table 3.12 shows there were 1,833 bugs which were designated as P1 priority to fix in 2022 and 2023.

Table 3.12: Fixed Firefox bugs by priority

Bug priority	*Year*		*Total*
	2022	*2023*	
P1	943	890	1833
P2	397	522	919
P3	442	560	1002
P4	9	10	19
P5	185	180	365
No priority assigned	1667	2065	3732
Total bugs	3643	4227	7870

Source: https://bugzilla.mozilla.org/home
Notes:
(1) Data retrieved on 15 April 2024
(2) P1 = highest priority ('Fix in the current release cycle')

3.40 Bugs are also assigned a severity rating according to how serious they are, with S1 described as 'Catastrophic' and S2 as 'Serious'[29].

Table 3.13: Fixed Firefox bugs by severity

Bug severity	*Year*		*Total*
	2022	*2023*	
S1	13	5	18
S2	214	212	426

[26] Mozilla Bugzilla bug tracker, accessed on 22 October 2024.
[27] Mozilla Bugzilla bug statuses, accessed on 22 October 2024.
[28] Mozilla Bugzilla bug fields, accessed on 22 October 2024.
[29] Mozilla Bugzilla bug fields, accessed on 22 October 2024.

S3	663	752	1415
S4	408	434	842
No severity assigned	2345	2824	5169
Total	3643	4227	7870

Source: https://bugzilla.mozilla.org/home
Notes:
(1) Data retrieved on 15 April 2024
(2) S1 = highest priority ('Catastrophic')

3.41 Table 3.14 shows the average time in days between bug creation and fix, where the bug has been assigned a severity rating of status S1 'Catastrophic' and a Priority of P1.

Table 3.14: P1 + S1 Firefox bugs with average fix time in days

	Total fixed bugs	*Average days to fix*
Bugs fixed in 2022	13	5
Bugs fixed in 2023	5	11
Total bugs 2022-23	18	8

Source: https://bugzilla.mozilla.org/home
Notes:
(1) Data retrieved on 15 April 2024

3.42 Some bug reports may not be publicly available, and thus excluded from the public data completely, for example Firefox bug ID 1767205 discovered when researching CVE resolution timelines.

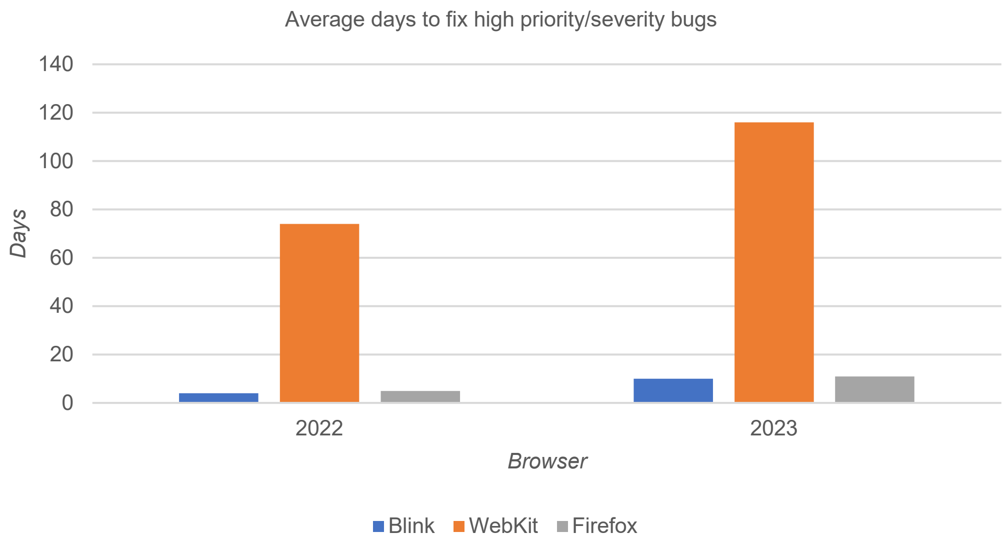
Comparing bug fix times for WebKit, Blink and Firefox

3.43 Volume of reported bugs is not a useful measure for comparison. In addition to the WebKit bug tracker, Apple operates a separate bug tracking system called 'Radar' for its own products. Other vendors also keep some bug information hidden from public view.

3.44 Average fix times for the most serious bugs recorded in public bug trackers have been determined for each of the three vendors and can be broadly compared.

3.45 Figure 3.4 shows that WebKit has a much greater average number of days between the most serious bugs being created and marked as fixed than the other vendors.

Figure 1.13: Average time in days to fix high priority/severity Blink, WebKit and Firefox bugs



Source: WebKit, Blink and Firefox bug data from Tables 2.8, 2.11 and 2,14

3.46 Having established a consistent definition of the term 'Fixed' across all three browser bug trackers, it is then possible to compare average days from bug creation to Fixed status as an indicator of how each vendor responds to the most serious issues affecting their product.

3.47 As the Mozilla tracker includes all Firefox-based products without enabling differentiation for the Gecko engine component, it is not possible to make direct comparisons for Gecko against the other engines. However, inclusion of the Firefox data is still indicative of how responsive Mozilla is to the most serious bugs affecting their browser product.

Days since last browser version

3.48 Browser updates are essential for mitigating security vulnerabilities, optimising performance, and improving feature support and compatibility. They may contain reactive fixes for exploits and bugs, and proactive developments to ensure defence against evolving threats.

3.49 More regular browser updates help ensure that end users have all these improvements available more quickly. In the case of bug fixes, the longer these are left unaddressed, the greater the risk to users.

3.50 Tables 3.15, 3.16 and 3.17 show the version numbers, release dates and days since previous release for Chrome, Firefox, and Safari.

Table 3.15: Chrome updates during 2022 to 2023

Version	*Release date*	*Days since previous release*
97	04/01/2022	49
98	01/02/2022	28
99	01/03/2022	28
100	29/03/2022	28
101	26/04/2022	28
102	26/05/2022	30
103	21/06/2022	26
104	02/08/2022	42
105	30/08/2022	28
106	27/09/2022	28
107	25/10/2022	28
108	29/11/2022	35
109	10/01/2023	42
110	01/02/2023	22
111	01/03/2023	28
112	29/03/2023	28
113	26/04/2023	28
114	24/05/2023	28
115	12/07/2023	49
116	09/08/2023	28
117	08/09/2023	30
118	04/10/2023	26
119	25/10/2023	21
120	29/11/2023	35
97	04/01/2022	49
98	01/02/2022	28

Source: https://chromereleases.googleblog.com/
Notes:
(1) Every four weeks a new version of Chrome is released across all platforms, see Google Chrome release cycle

Table 3.16: Firefox updates during 2022 to 2023

Version	*Release date*	*Days since previous release*
96	11/01/2022	35
97	08/02/2022	28
98	08/03/2022	28
99	05/04/2022	28
100	03/05/2022	28
101	31/05/2022	28
102	28/06/2022	28
103	26/07/2022	28
104	23/08/2022	28
105	20/09/2022	28
106	18/10/2022	28

107	15/11/2022	28
108	13/12/2022	28
109	17/01/2023	35
110	14/02/2023	28
111	14/03/2023	28
112	11/04/2023	28
113	09/05/2023	28
114	06/06/2023	28
115	04/07/2023	28
116	01/08/2023	28
117	29/08/2023	28
118	26/09/2023	28
119	24/10/2023	28
120	21/11/2023	28
121	19/12/2023	28

Source: https://wiki.mozilla.org/index.php?title=Release_Management/Calendar&redirect=no
Notes:
(1) Every four weeks a new version of Firefox is released across all platforms, see Mozilla Firefox release notes

Table 3.17: Safari updates during 2022 to 2023

Version	*Release date*	*Days since previous release*
15.4	14/03/2022	35
15.5	16/05/2022	63
15.6	20/07/2022	65
16	12/09/2022	38
16.1	24/10/2022	42
16.2	13/12/2022	50
16.3	23/01/2023	37
16.4	27/03/2023	63
16.5	18/05/2023	52
16.6	24/07/2023	36
17	18/09/2023	56
17.1	25/10/2023	37
17.2	11/12/2023	35

Source: Apple Safari release notes
Notes:
(1) Safari updates are bundled with iOS updates, see: https://support.apple.com/en-gb/102665

3.51 During this period Chrome released 24 versions with an average gap of 31 days between each release. Firefox released 26 versions with an average gap of 29 days between each release, whereas there were 13 versions of Safari during the same period with an average gap of 47 days between releases.

Internal data on security bugs and issues

3.52 We have received data on security bugs from Apple, Google and Mozilla. This section compares the total number of bugs on each platform, identified both internally and externally, as well as the time they took to resolve.

Volume of bugs and issues

3.53 Table 3.18 below shows the total number of security bugs and issues for each party. In each period, Google was the platform with the largest number of security bugs and issues, followed by Mozilla and then Apple.

Table 3.18: Total number of security bugs and issues [30]

Party	*1 November 2020 – 31 October 2021*	*1 November 2021 – 31 October 2022*	*1 November 2022 – 31 October 2023*	*1 November 2023 – 31 March 2024*
Apple	[✂]	[✂]	[✂]	[✂]
Google	[✂]	[✂]	[✂]	[✂]
Mozilla	[✂]	[✂]	[✂]	[✂]

3.54 For the period 1 November 2020 – 31 March 2024, Google had the largest number of bugs identified both internally and externally.

3.55 Bugs can either be identified by parties internally or via third parties:

(b) Between 1 November 2020 and 31 March 2024, Apple identified [✂]% of its bugs internally;[31]

(c) Between 1 November 2020 and 31 March 2024, Mozilla identified 57% of its bugs internally;[32]

(d) Between 1 November 2020 and 31 March 2024, Google identified [✂]% of its bugs internally.[33]

3.56 In addition to this, Apple submitted a breakdown of how many of these bugs were attributable to Safari and WebKit, for both the internally and externally identified bugs. In both cases, the majority of bugs were attributable to WebKit, rather than Safari.[34]

[30] CMA analysis of: Apple's response to the CMA's information request [✂]; Apple's response to the CMA's information request [✂]; Google's response to the CMA's information request [✂]; Google's response to the CMA's information request [✂]; Mozilla's response to the CMA's information request [✂]; Mozilla's response the CMA's information request [✂].
[31] CMA analysis of: Apple's response to the CMA's information request [✂]; Apple's response to the CMA's information request [✂].
[32] Mozilla's response to the CMA's information request [✂]; Mozilla's response the CMA's information request [✂].
[33] CMA analysis of: Google's response to the CMA's information request [✂]; Google's response to the CMA's information request [✂].
[34] Apple's response to the CMA's information request [✂].

3.57 For the bugs and issues identified internally:

(e) Of the [✂] security bugs and issues Apple internally identified from 1 November 2020 to 31 October 2022, it stated that [✂] were attributable to Safari and [✂] attributable to WebKit.[35]

(f) Of the [✂] security bugs and issues Apple internally identified from 1 November 2022 to 31 October 2023, it stated that [✂] were attributable to Safari and [✂] attributable to WebKit.[36]

(g) Of the [✂] security bugs and issues Apple internally identified from 1 November 2023 to 31 March 2024, it stated that [✂] were attributable to Safari and [✂] attributable to WebKit.[37]

3.58 For bugs and issues identified by third parties:

(h) Of the [✂] bugs and issues Apple stated third parties identified from 1 November 2020 to 31 October 2022, it stated that [✂] were attributable to Safari and [✂] (including all exploits) were attributable to WebKit.[38]

(i) Of the [✂] security bugs and issues Apple stated third parties identified from 1 November 2022 to 31 October 2023, Apple submitted that [✂] were attributable to Safari and [✂] attributable to WebKit.[39]

(j) Of the [✂] security bugs and issues Apple stated third parties identified from 1 November 2023 to 31 March 2024, Apple submitted that [✂] were attributable to Safari and [✂] attributable to WebKit.[40]

Resolving security bugs and issues

3.59 Table 3.19 shows the proportion of security bugs and issues that were resolved as of the date we received the data. Apple had resolved the largest proportion of reported bugs, with around [✂] of bugs already resolved in the three periods it submitted data for. This was then followed by Google and then Mozilla.

[35] Apple's response to the CMA's information request [✂].
[36] Apple's response to the CMA's information request [✂].
[37] Apple's response to the CMA's information request [✂].
[38] Apple's response to the CMA's information request [✂].
[39] Apple's response to the CMA's information request [✂].
[40] Apple's response to the CMA's information request [✂].

Table 3.19: Proportion of bugs and issues resolved as of submission date

Party	*1 November 2020 - 31 October 2021*	*1 November 2021 - 31 October 2022*	*1 November 2022 – 31 October 2023*	*1 November 2023 – 31 March 2024*
Apple	[✂]	[✂]	[✂]	[✂]
Google	[✂]	[✂]	[✂]	[✂]
Mozilla	[✂]	[✂]	[✂]	[✂]

3.60 Table 3.20 shows the average number of days between when a bug was first reported and when it was fixed in the central system, out of the bugs that were resolved. Of the bugs that were resolved, Google was the quickest at resolving them in all four periods. This is followed by Apple and then Mozilla.

Table 3.20: Average time to resolve bugs and issues (days)

Party	*1 November 2020 - 31 October 2021*	*1 November 2021 - 31 October 2022*	*1 November 2022 – 31 October 2023*	*1 November 2023 – 31 March 2024*
Apple	[✂]	[✂]	[✂]	[✂]
Google	[✂]	[✂]	[✂]	[✂]
Mozilla	56.4	32.3	48.7	25.8

3.61 Whilst Google resolved bugs faster than Apple, it had resolved a slightly lower proportion of the bugs on its platform, making it tricky to draw any conclusions as to which party is more efficient at resolving bugs. Mozilla, on the other hand, had resolved the lowest proportion of bugs in each period, and of the bugs resolved, was the slowest to report fixes.

Limitations of quantitative comparisons of security and bugs

3.62 There are limitations to quantitative comparisons of security across software or devices. For example:

(k) There is no way to effectively measure how many vulnerabilities software contains.

(l) A higher number of vulnerabilities may reflect more active efforts to find and fix security issues.

(m) Measures of attacks reveal more about attacker preferences than security.

(n) A higher number of updates may not reflect better security. For example, this could reflect imperfect fixes for old issues or the fact that a system has more features.

(o) Apple notes [✂].[41] [✂].

(p) According to the security consulting firm RET2, from which the CMA commissioned advice during the CMA's MEMS, there is no way to effectively measure how many vulnerabilities software contains. As not all fixed vulnerabilities are reported, the number of vulnerabilities found in a piece of software depends on the scrutiny that the software receives, which in turn may depend on resources allocated to this. A higher number of security fixes could therefore indicate that a piece of software has more vulnerabilities but also that more effort is being devoted to identifying and eliminating them.[42]

(q) [✂].[43]

(r) Mozilla submitted that the number of vulnerabilities could reflect different methods of measuring such vulnerabilities, such as whether internal security research is included alongside externally reported bugs. [44]

3.63 As a result of these limitations, we do not place significant weight on the above evidence.

[41] Apple's response to the CMA's information request [✂]; Apple's letter [✂]; Apple's response to the CMA's information request [✂].
[42] RET2's advice to the CMA [✂]. RET2 Systems Inc. is a computer security consulting firm that was commissioned by the CMA in 2022 to give expert technological advice to as part of the Mobile Ecosystems Market Study..
[43] [✂] submissions to the CMA [✂]
[44] Mozilla's response to Working Paper 2: The requirement for browsers operating on iOS devices to use Apple's WebKit browser engine dated 27 June 2024, page 6.

4. Browser stability analysis

4.1 This section considers crash and usage data we have received from browser vendors that operate on both iOS and Android. Comparing mobile browser that use different browser engines can indicate whether one is more stable. We have received data from Google, Mozilla, Opera, and Microsoft.

Google

Reported crashes

4.2 Google submitted that in its usual course of business, it assesses the stability of Chrome versions on an individual platform only looking at the change in the number of crashes from one version to the next. Google submitted that it [✂].[45]

4.3 Google submitted that data on crashes are reported only for users that have anonymous metrics reporting enabled. Of the users with this enabled, Google [✂]. Google provided various reasons why the [✂]:[46]

(s) Chrome on iOS and Android is used on different devices and [✂].

(t) [✂].

(u) [✂].

(v) [✂].

(w) [✂].

4.4 Google submitted that Chrome supports a number of different release channels and uses these channels to roll out updates to users on an incremental basis. [✂].[47]

4.5 Google submitted that it [✂]. We have made [✂] information requests for this data, meaning that in total we have six months of data:

(x) January 2024 – March 2024;[48] and

(y) November 2022 – January 2023.[49]

4.6 Google submitted that [✂].[50]

45 Google's response to the CMA's information request [✂].
46 Google's response to the CMA's information request [✂].
47 Google's response to the CMA's information request [✂].
48 Google's response to the CMA's information request [✂].
49 Google's response to the CMA's information request [✂].
50 Google's response to the CMA's information request [✂].

4.7 For these two time periods, Tables 4.1 and 4.2 and Tables 4.3 and 4.4 show Google Chrome's reported crashes, daily active users and time spent in Chrome on Android and iOS, respectively.

4.8 In absolute numbers, Google received [✂].

4.9 In terms of usage, Chrome had [✂] daily active users and time spent on Chrome [✂].

Table 4.1: Google Chrome reported crashes and usage on Android, November 2022 – January 2023[51]

	Reported crashes (m)	*DAU (b)*	*Time spent in Chrome (millions of days)*
November 2022	[✂]	[✂]	[✂]
December 2022	[✂]	[✂]	[%]
January 2023	[✂]	[✂]	[%]

Table 4.2: Google Chrome reported crashes and usage on Android, January – March 2024[52]

	Reported crashes (m)	*DAU (b)*	*Time spent in Chrome (millions of days)*
January 2024	[✂]	[✂]	[✂]
February 2024	[✂]	[✂]	[✂]
March 2024	[✂]	[✂]	[✂]

Table 4.3: Google Chrome reported crashes and usage on iOS, November 2022 – January 2023 [53]

	Reported crashes (m)	*DAU (b)*	*Time spent in Chrome (millions of days)*
November 2022	[✂]	[✂]	[✂]
December 2022	[✂]	[✂]	[✂]
January 2023	[✂]	[✂]	[✂]

Table 4.4: Google Chrome reported crashes and usage on iOS, January – March 2024[54]

	Reported crashes (m)	*DAU (b)*	*Time spent in Chrome (millions of days)*
January 2024	[✂]	[✂]	[✂]
February 2024	[✂]	[✂]	[✂]
March 2024	[✂]	[✂]	[✂]

4.10 Figure 4.1 and Figure 4.2 below uses the data Google provided (described above) and shows the number of reported crashes on Chrome relative to usage, on iOS and Android. Figure 4.1 shows the millions of reported crashes per billions of daily

[51] Google's response to the CMA's information request [✂].
[52] Google's response to the CMA's information request [✂].
[53] Google's response to the CMA's information request [✂].
[54] Google's response to the CMA's information request [✂].

active users and Figure 4.2 shows the millions of reported crashes per millions of days spent on Chrome. Both figures show that Chrome experiences [✂].

4.11 For both measures of relative crash reports, the ratio between Chrome crashes on Android and Chrome crashes on iOS [✂]:

(z) Chrome had just over [✂] times as many reported crashes per billions of daily active users on Android as it did on iOS between January 2024 and March 2024. This is compared to just over [✂] times as many between November 2022 and January 2023.

(aa) Chrome had [✂] times as many reported crashes per millions of days spent in Chrome on Android than it did on iOS between November 2022 and January 2023 and [✂] times as many between January 2024 and March 2024.

Figure 1.14: [✂]

[✂]

Figure 1.15: [✂]

[✂]

Google stability reports

4.12 Google submitted that it uses user feedback reports to assess Chrome's stability. It stated that users can provide feedback on stability under seven different titles.[55] The 'reported crashes' sub-section above considered data from a sample of crashes, whereas this sub-section considers data from user-initiated feedback reports.

4.13 Table 4.5 shows the number of user feedback reports submitted to Google between 31 October 2021 and 30 November 2022 on iOS and Android. This shows that Chrome experienced over [✂] as many stability reports on iOS as it did on Android, and over [✂] as many stability reports relative to daily active users.

4.14 Google submitted that feedback under the 'AwSnapError' title is the most relevant for comparing crashes across Blink and WebKit because 'AwSnapError' indicates that it is the renderer process specifically that has crashed. Google further submitted that [✂]% of the stability reports on Android were 'AwSnapError' reports, compared to [✂]% on iOS.

[55] (i) "AWSnapError"; (ii) "Crashing"; (iii) "BrowserNotLaunching", (iv) "BrowserNotClosing"; (v) "BrowserRunsinBackground"; (vi) "CantUninstallBrowser"; and (vii) "ProgramsNotinstalling"; Google's response to the CMA's information request [✂].

Table 4.5: Stability reports for Chrome on iOS and Android, 31 October 2021 – 30 November 2022 [56]

	Android	*iOS*
Stability reports	[✂]	[✂]
Stability reports per 10 million 1 day daily active users	[✂]	[✂]
Of which AwSnapErrors	[✂]	[✂]

Other browser vendors

4.15 Other smaller browser vendors' have provided crash data and that operate across iOS and Android. This shows that there is little difference between the relative stability of their mobile browser across platforms:

(bb) Mozilla submitted data showing that between November 2020 and July 2024, 99.72% of Firefox on Android app launches were crash free, compared to 99.83% of Firefox on iOS app launches.[57]

(cc) Opera stated that 99.8% of daily sessions on its Android mobile browser are crash free, compared to 99.7% on iOS.[58]

(dd) Microsoft stated that Edge for iOS has a crash rate of [✂]% and that Edge for Android has a crash rate of [✂]%. Microsoft submitted charts showing that the crash rate of Edge on [✂], whereas on Android it [✂].[59]

[56] Google's response to the CMA's information request [✂].
[57] CMA analysis of Mozilla's response to the CMA's information request [✂].
[58] Opera's response to the CMA's information request [✂].
[59] Microsoft's response to the CMA's information request [✂].

5. Browser privacy analysis

5.1 Privacy considerations for browsers include how advertising trackers and cookies are managed, whether tracking parameters are removed from URLs, handling HTTPS links or whether user search histories are stored or shared.

5.2 ‘PrivacyTests.org’[60] is an open-source initiative that tests the level of privacy of different browsers and publishes the results monthly. It performs 123 automated tests in multiple desktop and mobile browsers.

5.3 Table 5.1 shows the number of tests that are Passed, Failed or Unsupported for Chrome, Safari and Firefox on Android and iOS. On iOS, results indicate fewer test failures than on Android, but fewer supported features overall.

Table 5.1: PrivacyTests results for Chrome, Safari and Firefox on Android and iOS

Test results	*Android*		*iOS*		
	Chrome	*Firefox*	*Chrome*	Safari	Firefox
Passed	32	30	37	36	35
Failed	89	88	74	74	72
Unsupported	2	5	12	13	16

Source: https://privacytests.org/
Notes:
(1) Data retrieved on 17 September 2024

5.4 Table 5.2 compares the number of Passed tests for all the mobile browsers that are included in PrivacyTests.org and available on both Android and iOS. On iOS, there is a higher number of tests passed for each browser and a lower number of failures overall compared to Android. However, Table 5.3 shows a greater number of privacy features are noted as unsupported (not available) in iOS mobile browsers than for their Android equivalents.

Table 5.2: Passed tests for mobile browsers that are available on both Android and iOS

	Brave	*Chrome*	*DuckDuckGo*	*Edge*	*Firefox*	*Firefox Focus*	*Opera*	*Vivaldi*	*Yandex*
Android	88	32	36	32	30	77	33	32	30
iOS	78	37	52	36	35	60	41	36	47

Source: https://privacytests.org/
Notes:
(1) Data retrieved on 17 September 2024

60 ‘PrivacyTests.org’, accessed on 17 September 2024.

Table 5.3: Unsupported tests for mobile browsers that are available on both Android and iOS

	Brave	*Chrome*	*DuckDuckGo*	*Edge*	*Firefox*	*Firefox Focus*	*Opera*	*Vivaldi*	*Yandex*
Android	2	2	5	2	5	10	5	2	2
iOS	13	12	14	13	16	20	13	13	15

Source: https://privacytests.org/
Notes:
(1) Data retrieved on 17 September 2024