National College of Ireland

Security and Privacy Evaluation of Cloud-based Browsers

MSc Research Project
CyberSecurity

S Teja Pulleti Kurty
Student ID: 22120467

School of Computing
National College of Ireland

Supervisor: Dr Arghir-Nicolae Moldovan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	…S Teja Pulleti Kurty…………………………………………………………………………
Student ID:	…22120467………………………………………………………………………………….……
Programme:	……MSc Cybersecurity……………………… **Year:** ……..2023………..
Module:	……Master Thesis………………………………………………………………….………
Supervisor:	……Dr Arghir Nicolae Moldovan……………………………………………….………
Submission Due Date:	……………………………….14/12/2023…………………………………………….………
Project Title:	……Security and Privacy Evaluation of Cloud-based Browsers……….……
Word Count:	………5546……………………….. **Page Count**………21…………………………….……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: …………S Teja Pulleti Kurty…………………………………………………………

Date: ………………………14/12/2023…………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Security and Privacy Evaluation of Cloud-based Browsers

S Teja Pulleti Kurty
x22120467

Abstract

Cloud based browsers have emerged as a safer alternative to traditional native browsing in the recent years. An instance of a cloud browser runs in a secure virtualized container, which isolates the user from direct exposure to malicious web scripts and provides enhanced anonymity and security. Despite the multiple advantages associated with cloud browsing, the extent to which these cloud-based systems guarantee user-privacy is not fully understood. Browser fingerprinting is a potential tracking technique which can compromise the user-privacy even for cloud-based browsers, and even when safety measures like block cookies, VPN and incognito mode is used. The range of potential fingerprinting attributes is extensive, and the effectiveness of modern cloud browsers in limiting them is not entirely known. In the present work, nine potential fingerprinting attributes are selected, and the effectives of these browsers from modern cloud service providers in restricting browser fingerprinting is studied. In addition, the performance of the cloud browsers in terms of their speed and graphics rendering is examined and compared to native browser by Speedometer and MotionMark tests.

1 Introduction

Attackers have most frequently targeted traditional web browsers as their attack surface. Most users do not take cybersecurity risks into account for the browsers and applications they use and assume that they are safe, but this isn't the case. Various security flaws can be exploited by threat actors, such as users unknowingly downloading malicious attachments or files. Installation of unsafe browser plugins and moreover, sometimes browsers themselves contain vulnerable bugs which could be exploited, contribute to additional risks. Apart from these issues, users' activities are often tracked and shared with large organizations, leading to targeted ads and cookie tracking. The mobile landscape introduces further privacy issues, as other applications on mobile devices can track users' browser activities. In response to these challenges, cloud-based solutions have gained popularity among internet users due to their convenience, scalability, and cost-effectiveness. (Taivalsaari et al., 2013) argued that in future web browser is expected to predominantly shift to cloud-based platforms and this would enable users to access their browser sessions seamlessly from numerous computers and devices simultaneously. However, the security implications of moving a browser to cloud were not enough discussed.

Browser fingerprinting is a technique used to track and identify individual users based on the unique characteristics of their web browser and device. This method raises privacy concerns as it can be used to create a profile or history of a user's online activities without their knowledge or consent. In certain aspects, it poses a greater threat to privacy compared to tracking through cookies because users lack direct control over it (Al-Fannah et al., 2018).This research is aimed at finding out how effective are cloud-based browsers in limiting fingerprinting users and also testing against common security issues faced by browsers.

1.1 How cloud browsers work.
In Wang et al.'s 2010 study, they explained that a cloud browser operates as Software as a Service (SaaS). With this approach, users don't need to install or run applications on their local devices. Instead, the cloud browser leverages the cloud's capabilities, allowing users to access a browser instance in the cloud to interact with applications. When a user opens a webpage using a cloud browser, the browser sends a request to the target server before processing the response. Rather than displaying the actual code, the user sees a real-time streaming image of the requested page. Cloud browsers offer benefits like safeguarding users from potential security threats, providing anonymity, and protecting against malicious code. To ensure isolation, these browsers operate within virtualized containers.

1.2 Research Questions
How effective are cloud browsers in reducing browser fingerprints of users?
Are cloud browsers better alternatives to traditional browsers?

2 Related Work

2.1 Cloud Browser Performance: A fast and responsive speed is crucial for a browser's performance and user experience. The loading speed of a website or application plays a significant role in determining the overall user experience (UX) and can impact user engagement and usability.

Sivakumar et al. (2014) conducted an experiment on mobile cloud browsers to test for energy efficiency and download time. During the experiment, it was found that Cloud Browser (CB) did not provide clear benefits over device-based browsers in terms of energy or download time. In 38.87% of pages the CB download time was less compared to Direct Browsers but at the same time, it exceeded by 29.8s for other browsers. Furthermore, there was an efficiency gain in total energy consumption by 20.77J compared to the direct ones on nearly 52.7% tested pages but in other pages the consumption had increased up to 21.31J. Even though CB executes JavaScript in the cloud, it increases CPU and network energy for nearly 50% of the pages. In this research, it was not clear whether cloud browsers were suitable for everyday usage like light browsing.

Rempel et al. (2015) highlighted the importance of the impact of web browser performance on users, Mohamed and Ismail (2022) compared web browser performance of popular browsers in different experiments using benchmarking tools MotionMark and Speedometer. In their third experiment, they used a Task Manager tool for further analysis. These tests can be used in this research because these experiments are aimed at testing the browser's capacity to manage different volumes of interactions, responsiveness, and adaptability to complex scenarios requiring a lot of resources, as well as potential trade-offs associated with high performance (demand on the RAM, GPU and CPU). This will help us in determining the user-friendliness of cloud browsers in everyday usage.

2.2 Browser Security:

Hothersall-Thomas et al. (2015) developed a tool called BrowserAudit for evaluating browser security. It tests whether a deployed browser enforces the guarantees implied by standardised and experimental security mechanisms. It includes over 400 fully automated tests that exercise a broad range of security features. BrowserAudit helps web users, application developers, and security researchers to make informed security assessments of a deployed browser. It is validated by discovering both fresh and known security-related bugs in major browsers.

This test is essential for this research since provides results for security of cloud-browsers used in this paper for analysis in categories like Same-Origin Policy, Cross-Origin Resource Sharing, and Content Security Policy. The same-origin policy (SOP) effectively stops various cross-site scripting (XSS) attacks on users' web browsers and is a crucial element of web security (Hickson I.,2014). According to Barth et al. (2008) despite fully-compliant implementations of SOP and CORS mechanisms, access to resources like images, embedded objects, and web fonts can still be vulnerable to CSRF, clickjacking, framebusting, and CSS-based attacks. The Content Security Policy (CSP) standard allows for more precise control over how various resources are loaded on a web page, helping to address and lessen several related issues. (B. Sterne & A. Barth, 2012).

2.3 Browser Fingerprinting:

Browser fingerprinting can significantly impact user privacy due to its ability to create a unique identifier based on various attributes of a user's browser and device.

Eckersley (2010) investigated the extent to which modern web browsers are vulnerable to device fingerprinting, a technique that can be used to uniquely identify individual users. They implemented a fingerprinting algorithm and collected fingerprints from a large sample of browsers that visited their test site, panopticlick.eff.org. Their findings indicated that browser fingerprints contain a significant amount of entropy, meaning that they can be used to distinguish between a large number of users. Notably, browsers supporting Flash or Java demonstrated increased susceptibility, with an average of at least 18.8 bits of identifying information, and furthermore, 94.2% of such browsers exhibited unique fingerprints in the sample. Additionally, it was found that browser fingerprints change relatively rapidly over time, making it difficult to track users across multiple sessions. However, they also developed a heuristic approach that can accurately identify when a fingerprint is an upgraded version of a previously observed fingerprint, with 99.1% accuracy and a false positive rate of only 0.86%. These findings suggest that browser fingerprinting poses a significant privacy threat, and that current countermeasures are not always effective. In the present research it has been shown that even disabling JavaScript and Cookies, a user can still be fingerprinted from plugins, fonts, User-agent strings. This information is very helpful in testing browsers like Tor and Brave which have options to choose to strict safe browsing which disables JavaScript and Cookies.

The next challenge was to select appropriate fingerprinting attributes for this research. Narayanan and Shmatikov (2008) argued that the unique characteristics of web browsers (known as fingerprints) result from choices made by software developers and users, sometimes accidentally. The full range of possible fingerprint values is uncertain and quite extensive, posing privacy challenges. Although there is a limit to the options, the set is both large and scattered, raising concerns about privacy. Hence Eckersley (2010) chose 'User-Agent', 'HTTP Accept headers', 'Cookies enabled?', 'Screen resolution', 'Timezone', 'Browser Plugins and MIME types', 'System fonts', and 'Partial supercookie test' as attributes for their test site as these are the most common and prominent features for tracking a user via browser fingerprinting.

Fifield and Egelman (2015b) developed and proposed a web browser fingerprinting method that relies on measuring onscreen dimensions of font glyphs. This technique leverages the diverse factors influencing font rendering in browsers, such as version variations, installed fonts, and rendering settings, to create unique fingerprints for end-user systems. The study demonstrates the efficacy of even a basic approach, such as measuring glyph bounding boxes, as a significant privacy threat. Through a comprehensive user experiment involving more than 1,000 web browsers and an exhaustive survey of Unicode space, the research reveals that font metrics exhibit greater diversity compared to conventional User-Agent strings, uniquely identifying 34% of participants and reducing others into smaller anonymity sets. Overall, this study further stresses the importance of fonts as major fingerprinting attribute.
Acar et al., (2013) in their research used fonts one of the parameters while developing a fingerprinting algorithm called FPDetective, because fonts rank as a device's second most distinctive feature. In addition, fonts depend on the operating system, and allow multiple browsers to be linked on the same device.

Wadkar et al., (2014) is his research highlighted that browser specific information is one of the main reasons for information leakage. Among different vulnerability classes like CSRF, Content Spoofing, Cross-scripting and others the study showed that browser fingerprinting is responsible for 23% of the vulnerability. For the experimental analysis they used Browserleaks[1] to test various attributes like system CPU information, HTTP-request headers, User-Agent, Geolocation and others. For this research online tool Browserleaks will be used to test for selected attributes in the cloud browsers.

Mayer, (2009) conducted a study at the Princeton Center for Information Technology, where 1328 web clients visited scoop.princeton.edu using their preferred browser. A JavaScript snippet was executed to check for a cookie, and to concatenate and hash the contents of the navigator, screen, navigator.plugins, and navigator.mimeTypes objects, resulting in a unique 128-bit identifier. Over the course of the experiment, 96.23% of the visitors were be uniquely identified. This case study signifies the importance of including browser attributes

[1] http://www.browserleaks.com/

navigator.plugins and navigator.mimeTypes in this research to check the kind of information the cloud browsers reveal. This is done because the navigator object has details about the browser, like who made it and its version. It also includes info about plugins, MIME types it supports, and some basic details about the operating system and architecture where the browser is working (Acar et al., 2013).

Englehardt & Narayanan, (2016) discovered for the first time by analysing a large dataset that AudioContext can be used to create a fingerprint. AudioContext is a HTML5 element that is responsible for playing sounds in a browser. The audio signals are hashed and utilized as fingerprint. They developed an online tool called OpenWPM to test for AudioContext fingerprinting attribute. Audiofingerprint.openwpm[2] is a tool that was developed during this research and can be utilised for testing audio fingerprinting.

Olejnik et al., (2016) in his study investigated the privacy risks associated with the HTML5 Battery Status API, focusing particularly on its implementation in the Firefox browser. The research revealed that websites can exploit the high-precision readouts provided by Firefox on Linux to discover users' battery capacity, exposing a fingerprintable surface that enables efficient tracking in short time intervals. The risk is more noticeable for older or used batteries with reduced capacities, as battery capacity may serve as a tracking identifier. Hence battery attribute is also being taken into account for this research.

Table 1: Summary of research paper that evaluated browsers and fingerprint attributes

Paper	**GOAL**	**Browser**	**Metrics**	**Tools/ Benchmarks**	**HW/OS**	**Findings**
Mohamed and Ismail (2022)	Performance comparisons of internet browsers	MS Edge, Opera, Mozilla Firefox, Brave, google Chrome	Memory, CPU and GPU usage	Speedometer 2.0, MotionMark, Task Manager	16gb ram, I7. Intel CPU, SSD, Win-11 64bit	Chrome browser uses less GPU power but more CPU and RAM, meanwhile Edge does the opposite
Tendulkar (2012)	Execute large and parallel tasks in cloud browsers	Puffin, Amazon Silk,Opera Mini, Cloud Browse	CPU Cycles, Memory, Elapsed time	NA	Kindle Fire tablet, iPhone 3G , Samsung Galaxy Nexus	Cloud browsers were exploited for free large scale computing operations
Hothersall-Thomas, C., Maffeis, S., & Novakovic, C. (2015)	To test security features of deployed browser by running more than 400 tests.	NA	CORS, CSP, Header-Request, Header-Response	BrowserAudit	NA	The tool built in this research shows vulnerability of a browser across four categories (SOP,CSP,CORS, Headers)
Sivakumar, A., Gopalakrishnan, V., Lee, S., Rao, S., Sen, S., & Spatscheck, O. (2014).	Compare CB with Direct browsers in terms of download time and energy consumption.	Opera Mini, Amazon Silk , Sky Fire and Chrome beta	Energy Consumptio n, Page download time	open source ARO tool	Samsung galaxy S3	Cloud decreases energy consumption and download time by 52.7% and 38.8% respectively on tested web pages compared to native

[2] . https://audiofingerprint.openwpm.com

						browser. But sometimes it consumes more than native browser.
Fifield and Egelman (2015)	web browser fingerprinting technique centered on measuring onscreen font glyph dimensions.	Chrome, Safari, Firefox, Internet Explorer	On Screen Font Glyph Size, CSS, Unicode, Canavas	NA	Windows (XP,Vista,7,8) OS X 10.9 Android iOS	34% of the users that participated in the test were uniquely identified by gathering font information of their device.
Olejnik et al., (2016)	Track user browsing habits from battery information	FireFox	Battery Capacity, Charging Level, Discharging rate	Battery Status API	Linux Operating system, Windows, Mac OS, Android	Websites can track user's battery information that could be gathered given away by Firefox browser in linux based devices
Englehardt & Narayanan, (2016)	To gather fingerprint information collected by top 1-million websites	Deployed 20 browsers in Amazon EC2 instance	Canvas, WebGL, Cookies	OpenWPM, HTML5's AudioContext API	Amazon EC2(8 CPUs, 15 GiB Memory)	Combined input and output audio signals to generate a hash which can act a fingerprint.

3 Research Methodology and Specification

The main goal of this research was to carry out and evaluate performance, security and privacy aspects of cloud-based browser using a combination of performance testing, BrowserAudit tests and browser fingerprinting tests. In addition, privacy and security-focused browsers like Brave, Tor and DuckDuckgo were also taken into account for further comparison analysis.

3.1 Selection of browsers:

Firstly, commercially available and affordable cloud-based browsers were identified. Based on available CPU and RAM resource available Network Chuck Browser, KASM browser (which works both on PC and smartphone) and Puffin Cloud Browser (which is a mobile android application) were selected. Secondly from privacy point of view Log collection policy and compliance with law enforcement were also analysed. Silo Auhentic8 was not available as it is accessible only to enterprise users only and Amazon Silk browser was not tested due to the unavailability of an amazon device.

Table 2. List of Cloud-Browsers considered for research

	Network Chuck Browser	KASM	Puffin Cloud Browser	Silo Authentic8	Amazon Silk	Tor
Cost	$7/month	$5/month	$1/month	Enterprise Only	Free	Free
Hardware/ VM	1 cores, 2Gb RAM	2 Core/ 2GB RAM	Dependent on Device	N/A	Amazon Kindle, Alexa, Firestrick, EC2	Dependent on Device

					Instance	
Browsers Provided	Brave, Chrome, Chromium, Edge, Firefox, Vivaldi	Brave Chrome Chromium Edge Firefox Vivaldi Tor Browser	Puffin Browser Android App	Chromium	Amazon Silk App	Tor
Cloud Provider OS	Linux	Linux	Linux	N/A	N/A	N/A
How to run?	Runs inside any browser	Runs inside any browser	Can be installed from Playstore or manually from apk	N/A	Built in Amazon fire devices	Can be installed from Playstore or manually from apk
Log Collection	Application logs, Browser logs deleted when session ends	Application logs, Browser logs deleted when session ends	Yes.180 days retention policy	All logs are collected	All logs, user history, shared with third parties	Doesn't keep any logs that could identify a particular user.
Compliance with Law Enforcement	Yes	Yes	Yes	N/A	Yes	No

3.2 Selection of Tools:

Speedometer[3] is a test that checks how well browsers handle over 200 website searches. It assesses the browsing engine's responsiveness by visiting various websites and measuring the time it takes to load each page. The browser's responsiveness relies on how the CPU handles multi-threaded processes (Mohamed & Ismail, 2022). The higher the score better the result.

MotionMark is a benchmark test that looks at how well web browsers handle graphics. It displays many elements on the screen using HTML, CSS, and JavaScript. MotionMark runs various animations on the browser to see how smoothly they can handle them at 60 frames per second (the test lasts for approximately five minutes). The better the score, the better is the browser performance. This test relies on graphics because it checks the browser's capability to show and animate objects simultaneously at 60fps and measures how much it utilizes the GPU (Mohamed & Ismail, 2022). The higher the score better the result.

[3] https://browserbench.org/Speedometer2.0/

BrowserAudit[4] is an automated online tool that conducts more than 400 tests for security of browser in different categories like Same-Origin Policy, Content Security Policy, Cross-Origin Resource Sharing, Cookies, Request Headers, Response Headers. User can also customize the selection of above mention categories whether to include or not during tests. The results from this test are categorized into ‘Passed’, ’Warning’, ’Critical’ and ‘Skipped’.

Table 3. BrowserAudit Testing Process

Security Test	**Testing Process**
SOP	The scripts written thoroughly test how a web browser implements the SOP as well as checks any unauthorized DOM access.
CORS	This test evaluates a browser's compliance with the CORS standard, which allows controlled sharing of resources across different origins. BrowserAudit assesses compliance by sending cross-origin XMLHttpRequest requests and checking if the browser exhibits CORS-compliant behavior based on the server's response headers. A browser is considered compliant if it correctly handles CORS-compliant requests and identifies and rejects CORS-violating requests.
CSP	A script is written that loads around 280 iframes with different scenarios. The browser is expected to either allow or block access to a given resource according to the CSP policy. A CSP-compliant browser correctly implements the standard by allowing permitted requests and blocking restricted ones. BrowserAudit tracks Failure if the browser attempts to access a resource with a restrictive policy.
Cookies	Verifies whether a cookie’s security attributes ‘HttpOnly’ and ‘Secure’ are correctly implemented or not by checking that it is not vulnerable to XSS exploit and should not be accessible by JavaScript either from client-side or server-side scripts.
Request and Response Headers	In this, a web page is loaded over HTTPS and contains an image file that loads over HTTP. BrowserAudit checks whether HSTS(HTTP Strict Transport Security) has been implemented in browser or not (i.e. checking whether the browser uses only use secure HTTPS or not).

BrowserLeaks provides a collection of tools designed to assess the security and privacy of your web browser. The tests it offers are aimed at detecting potential vulnerabilities such as the exposure of your actual IP address, the gathering of device-related information, and the execution of browser fingerprinting by websites. Nine attributes were selected for this research.

Table 4. Justification of attributes

Attributes	**Remarks**
Screen	The screen attribute reveals the size of the user's screen, the pixel ratio,

[4] https://browseraudit.com

	and the color depth. This information can be used to track the user's device and to serve them ads that are tailored to their screen size.
Navigator.plugins	Reveals a list of the plugins that are installed on the user's browser.
Navigator.mime	Contains information about the browser's capabilities, such as the types of files that it can open and the codecs that it supports. This information can be unique to a specific browser instance, even if the browser is running on the same operating system and with the same settings.
Font Metrics	The Font Metrics attribute reveals a list of the fonts that are installed on the user's browser and their sizes.
User Agent	The User Agent attribute reveals the type of browser, the version of the browser, and the operating system that the user is using. This string contains information about the user's browser, operating system, and device and is often used to distinguish one user from another
Canvas	Used for drawing graphics on web page and can be used to create a unique user ID as it information about graphic card, installed OS, browser as well as device fonts
WebGL	It is JavaScript API for 3D rendering on browsers. It utilizes device's graphic card/GPU whose details can be gathered to make unique fingerprint of a user.
Battery	Through HTML5 battery status API, the battery capacity details and charge remaining information can be used to create a unique identifier.
AudioContext	It is HTML5 audio attribute to play sounds. Through a complex process the input and output signals can be combined to generate a hash which in turn can be used a fingerprint.

OpenWPM[5] uses the Canvas and AudioContext APIs to test audio context browser fingerprinting.

3.3 Device Specifications:

Device Type: Smartphone
Resolution:1080 x 2400 pixels
Chipset: Qualcomm SM7325-AE Snapdragon 778G+ 5G (6 nm)
CPU: Octa-core (1x2.5 GHz Cortex-A78 & 3x2.4 GHz Cortex-A78 & 4x1.9 GHz Cortex-A55)
GPU: Adreno 642L
RAM: 8GB RAM

3.4 General Cloud Browser Architecture

For cloud browser the architecture is divided into two environments, consisting of a server and a user environment. The server is where the browser is safely deployed and user environment is the user interface through which the user interacts with the browser instance running in server. The user interface could be a browser (browser within browser) or an application inside which the browser is running. The user inputs are safely transmitted to the

[5] audiofingerprint.openwpm.com

instance and the server processes and renders output of the destination website as images/stream to the user (Palanques et al., 2012).

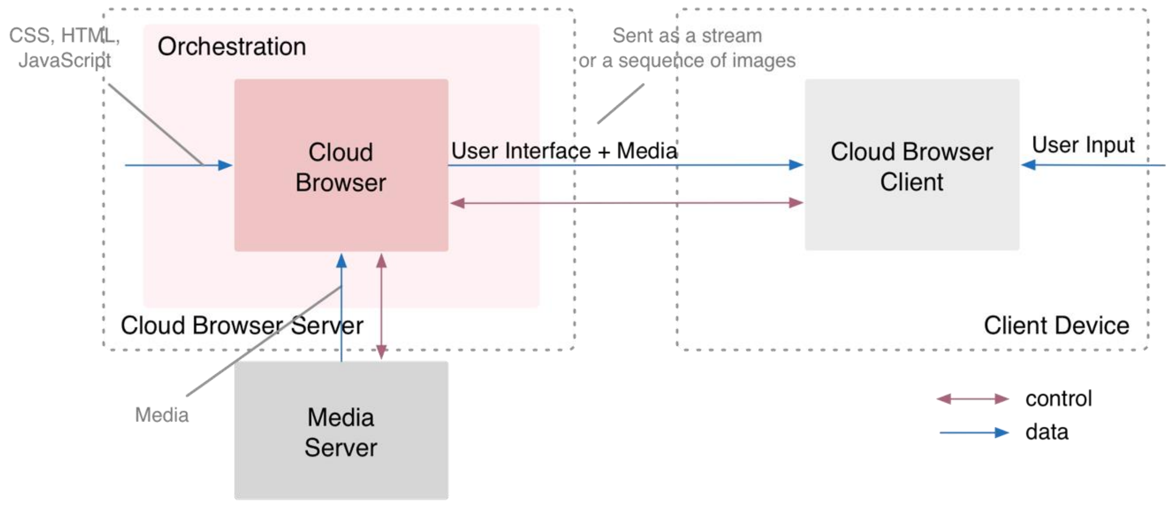

Fig 1. Cloud Browser Architecture (Source: Cloud Browser Architecture, 2017)
For testing all the browser following Android Phone was used:

3.4 MotionMark Specifications

Tests several drawing operations with methods like CSS, SVG, and Canvas:

Table 5: MotionMark testing parameters (Source: (About MotionMark 1.2, n.d.))

Score Parameters	**Descriptions**
Multiply	CSS border radius, transforms, opacity
Arcs and Fills	Canvas path fills and arcs
Leaves	CSS-transformed elements, opacity
Paths	Canvas line, quadratic, and Bezier paths
Lines	Canvas line segments
Focus	CSS blur filter, opacity
Image	Canvas get ImageData () and putImageData
Design	HTML text rendering
Suits	SVG clip paths, gradients and transforms

4 Implementation

In performance tests for both Speedometer and MotionMark, five runs were performed for all the browsers. Before running the test, it was also made sure that the memory usage was cleared for consistent results since the phone would otherwise throttle on continuous stress on CPU and GPU. In browsers like Tor, where level of security can be changed, the tests were repeated with different modes across native mobile and other two cloud instances of Tor. After five runs for each test, mean and standard deviation were was calculated.

For Browseraudit, the test was only run once, as the running multiple time would result incorrect results as the data is cached. The results were noted after running across all browsers.
For the final fingerprinting test, the selected attributes were checked in Browserleaks and were compared against original values for all browsers. Here it was checked whether the values remained same or changed or were spoofed/obfuscated by browser compared to supposed values. Moreover, audiofingerprint.openwpm.com was used for testing audiocontext attribute.

5 Evaluation

5.1 Performance Study

In both Speedometer and Motionmark tests it can be observed that in Tor browser, stricter security and privacy settings (standard to safer browsing mode) lead to decrease in the performance indicating that the browser is restricting some the features and functionalities for more security. The Puffin Cloud browser scores the lowest in motionmark test implying that it is not leveraging hardware resources of original device on which it is being run but, instead those of the cloud instance on which it is running. The DuckDuckgo mobile scores highest in both tests indicating that it leverages the resources of the device on which it is running showing that it is least secure in terms of privacy.

Similar patterns for Tor browsers can be observed. The performance decreases as the security settings are made stricter. Overall, the KASM browsers perform better compared to Network Chuck Browsers since the KASM has better CPU and GPU cores allocation in the cloud instance.

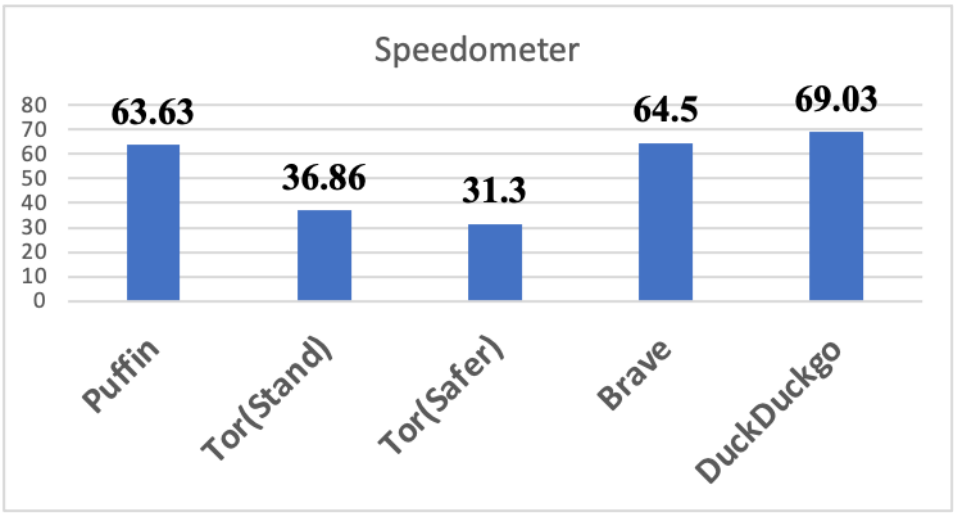

Fig 2. Speedometer results of Native Android Browser Apps

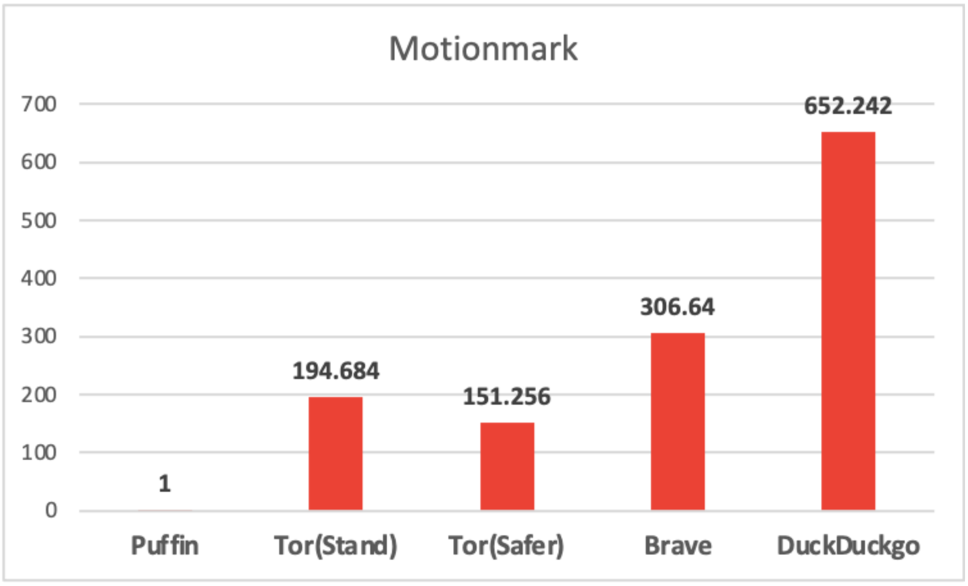

Fig 3. MotionMark results of Native Android Browser Apps

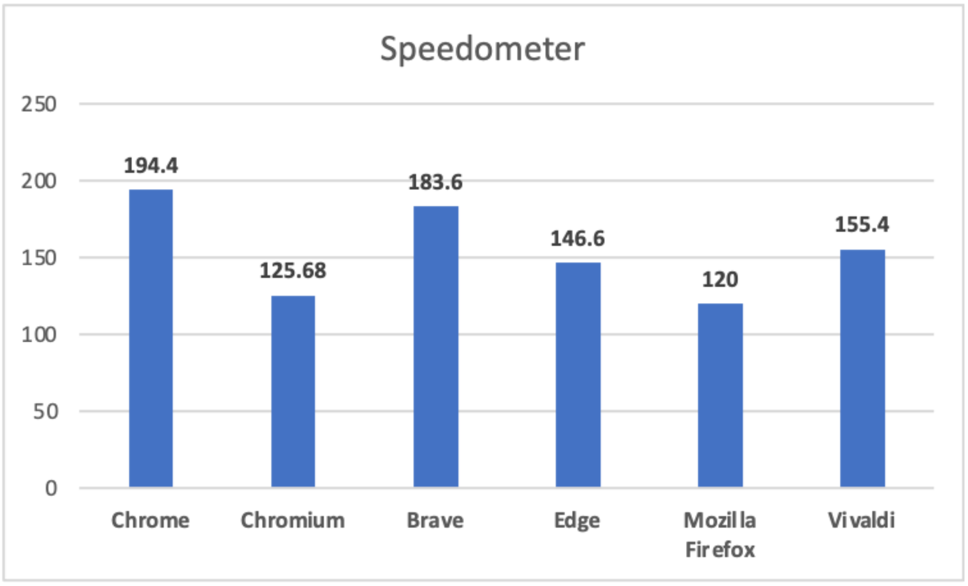

Fig 4. Speedometer Results of Browsers in Network Chuck Cloud

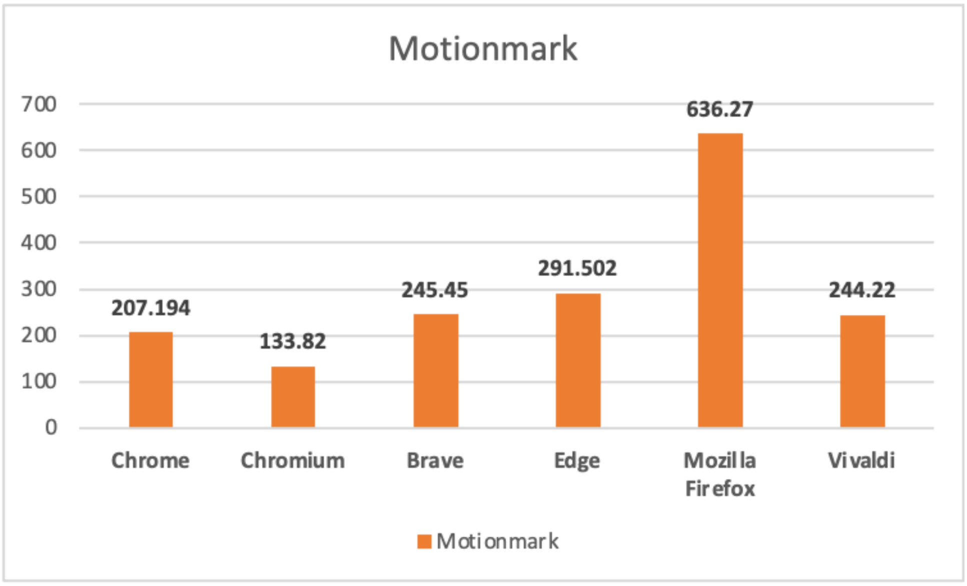

Fig 5. MotionMark Results of Browsers in Network Chuck Cloud

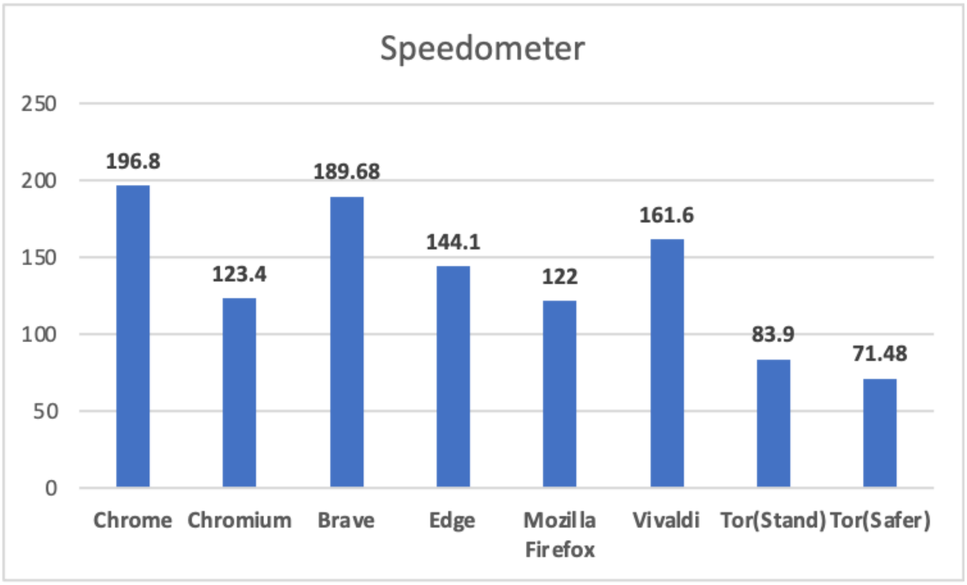

Fig 6. Speedometer results of browsers in KASM Cloud

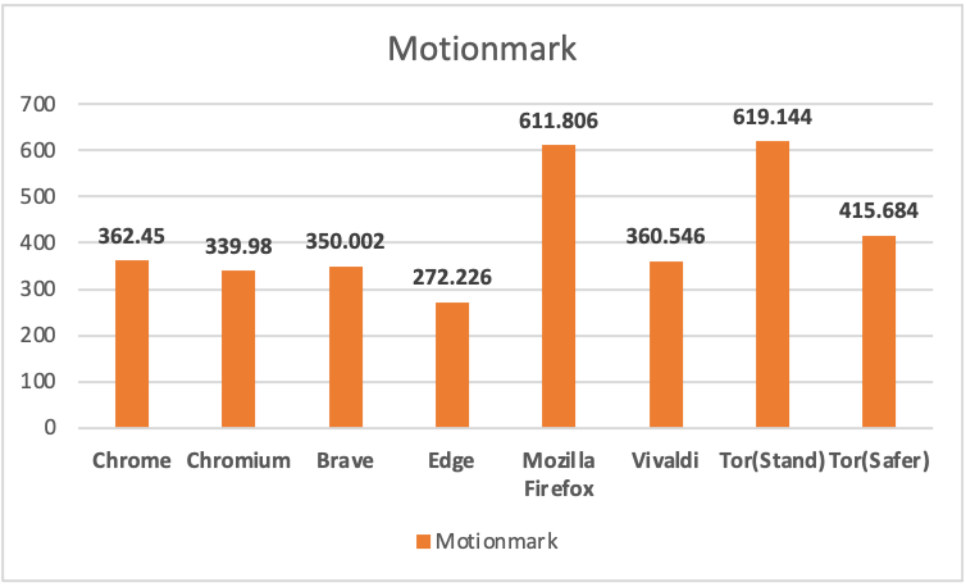

Fig 7. Speedometer results of browsers in KASM Cloud

5.2 Browser Audit Test

The results indicated that there zero critical security issues in the all browsers and all browsers have passed almost of the test. Under the ‘warnings’ category of Browseraudit.com (CSP, CORS, Request and Response Headers fall under warnings category) there were few alerts. Since JavaScript is completed disabled in Tor’s safest security mode the tool did not run since it requires JavaScript for execution.

Table 6. Browseraudit.com Scores for all browsers

Browser	PASSED	Content Security Policy	CORS	Request Headers	Response headers	Critical	Skipped
Tor (Standard)	353	21	4	6	3	0	44
Tor (Safer)	346	29	4	6	3	0	43
Tor (Safest)	Test did not run						
Puffin	395	28	4	4	0	0	0
DuckDuckGo	395	18	4	2	3	0	9
Brave	394	18	4	3	3	0	9
KASM (inside chrome android app)							
Firefox	404	14	4	6	0	0	1
Brave	401	20	4	3	3	0	0
Chrome	401	20	4	3	3	0	0
Chromium	401	20	4	3	3	0	0
Edge	401	20	4	3	3	0	0
Vivaldi	401	20	4	3	3	0	0
Tor	395	20	4	6	3	0	3
Tor (Safer)	383	32	4	6	3	0	3
Tor (Safest)	Test did not run						

Network Chuck (inside chrome android app)							
Firefox	403	15	4	6	0	0	3
Brave	402	19	4	3	3	0	0
Chrome	401	20	4	3	3	0	0
Chromium	401	20	4	3	3	0	0
Edge	401	20	4	3	3	0	0
Vivaldi	401	20	4	3	3	0	0

5.3 Fingerprint Test

Table 7. Fingerprint results for native mobile browsers

Attribute	Puffin	Tor (Standard)	Safer	Safest	Brave (Standard)	Brave (Strict)	DuckDuckgo
Screen	Original	Changed	Changed	Blocked	Original	Original	Original
Navigator. plugin	Changed	Changed	Changed	Blocked	Obfuscated	Obfuscated	Blocked
Navigator. mimeTypes	Changed	Changed	Changed	Blocked	Blocked	Blocked	Blocked
Font Metrics	Changed	Changed	Changed/ Obfuscated	Blocked	**Original**	**Original**	Changed
Canvas	Original	Obfuscated	Changed	Blocked	Obfuscated	Obfuscated	Changed
User Agent	Original	Obfuscated	Obfuscated	Blocked	Changed	Changed	Changed
WebGL	Changed	Changed	Blocked	Blocked	Original	Original	Original
Battery	Blocked	Blocked	Blocked	Blocked	Blocked	Blocked	Spoofed
Audio	Spoofed	Blocked	Blocked	Blocked	Spoofed	Spoofed	Original

Table 8. Fingerprint results for KASM cloud browsers

Attribute	KASM Brave	Brave (Strict)	Chrome	Edge	Chromium	Tor (Standard)	Tor(safer)	Tor (safest)	FireFox	Vivaldi
Screen	Changed	Changed	Changed	Changed	Changed	Changed	Changed	Blocked	Changed	Changed
Navigator .plugin	Changed	**Obfuscated**	Original	Original	Original	Original	Original	Blocked	Original	Original
Navigator .mimeTypes	Original	Original	Original	Original	Original	Original	Original	Blocked	Original	Original
Font Metrics	Changed	Changed	Changed	Changed	Changed	Changed	Changed	Blocked	Changed	Changed
Canvas	Changed	Changed	Changed	Changed	Changed	Changed	Changed	Blocked	**Not unique**	Changed
User Agent	Changed	Changed	Changed	Changed	Changed	Changed	Changed	Blocked	Changed	Changed
WebGL	Changed	Changed	Changed	Changed	Changed	Changed	Changed	Blocked	Changed	Changed
Battery	Spoofed	Spoofed	Spoofed	Spoofed	Spoofed	Blocked	Blocked	Blocked	Blocked	Spoofed
Audio	Changed	Changed	Original*	Original*	Changed	Blocked	Blocked	Blocked	Changed	Original*
*close to real value										

Table 9. Fingerprint results for Network Chuck cloud browsers

Attribute	NetworkChuck Brave	Brave (Strict)	Chrome	Edge	Chromium	FireFox	Vivaldi
Screen	Changed	Changed	Changed	Changed	Changed	Changed	Changed
Navigator .plugin	Changed	**Obfuscated**	Original	Original	Original	Original	Original
Navigator .mimeTypes	Original	Original	Original	Original	Original	Original	Original
Font Metrics	Changed	Changed	Changed	Changed	Changed	Changed	Changed
Canvas	Changed	Changed	Changed	Changed	Changed	**Not unique**	Changed
User Agent	Changed	Changed	Changed	Changed	Changed	Changed	Changed
WebGL	Changed	Changed	Changed	Changed	Changed	Changed	Changed
Battery	Spoofed	Spoofed	Spoofed	Spoofed	Spoofed	Blocked	Spoofed
Audio	Spoofed	Spoofed	Original*	Original*	Changed	Changed	Original*
*close to real value							

Table 10.Audio Fingerprint Scores in Native Android Browsers

Browser	**Fingerprint**
Chrome native android app*	124.08072766105033
Puffin	35.12035473063588
Brave	1.2882823928033254e-22
DuckDuckGo	124.08072766105033
Tor	Test was Blocked

Table 11. KASM Cloud Browsers' Audio Finger Print Score

KASM Browsers	Fingerprint	Difference (%)
Chrome	124.04347527516074	≈0.03004%
Chromium	124.04347527516074	≈0.03004%
Brave	3.5032461608120427e-41	100%
Edge	124.04347527516074	≈0.03004%
FireFox	35.749968223273754	≈71.25%
Vivaldi	124.04347527516074	≈0.03004%
Tor	Test Blocked	NA

Table 12. KASM Cloud Browsers' Audio Finger Print Score

Network Chuck Browsers	Fingerprint	Difference from Real Value (%)
Chrome	124.04347527516074	≈0.03004%
Chromium	124.04347527516074	≈0.03004%

Edge	124.04347527516074	≈0.03004%
FireFox	35.749968223273754	≈71.25%
Vivaldi	124.04347527516074	≈0.03004%

5.4 Discussion

From the performance tests done via Speedometer and Motionmark tests it can be observed that the cloud service provider which allocates higher CPU and GPU resources, performs better and at the same time the cost is also higher. If users want better performance while using cloud browsers, they should opt for expensive subscription. Hosting applications on the cloud is generally expensive and service providers usually limit resources allocated to browser which can be seen in the case of Motionmark test of Puffin cloud browser which scored result only 1 in tests because it's the cheapest subscription compared to the rest and the cloud provider might have heavily restricted the GPU allocation for it.

From all the attributes tested for fingerprinting, the cloud browsers did not reveal any original value or data (i.e the real values of the device or browser information except for audio fingerprinting attribute). While testing for battery attribute all browsers either blocked or spoofed the Battery API data from the website accessing it. The browsers which spoofed the data showed the battery level always as 100% and charging status always as 'charged' even if the device's battery level was different and it was not being charged.

Except Brave, Puffin, Firefox and Tor other browsers showed close to real value (the difference between them was close to 0.03004%) of the audio fingerprinting score and randomness of the value is very negligible indicating that these browsers' AudioContext data can serve as a potential fingerprint to uniquely identify a device or a user by an adversary. Brave showed different score because it employs a technique called farbling to protect user privacy from potential threats such as fingerprinting and tracking. Farbling introduces randomness to signal outputs, making it difficult for websites to detect a user's fingerprint but not breaking benign, user-serving websites (Frola, 2022).

Coming to answer the initial research question speculated in the beginning, it is safe to say yes that cloud browsers are quite safe alternative for traditional web browsers in terms of safety and anonymity they provide, although their performance is slight subpar. Moreover, the browsers did not fail or anything was flagged as critical during the BrowserAudit tests. From all the tests conducted across the browsers it is found that the Brave and Tor instances of cloud browsers are the most secure in terms of fingerprinting privacy, at same time, Brave is better in terms of performance compared to Tor. The cloud browsers performed well against the fingerprinting attributes chosen for this research except Audio Finger Printing.

6 Conclusion and Future Work

During the performance tests conducted in the present work, it was found that by implementing stricter security measures in native browsers, in general the performance scores calculated by MotionMark and Speedometer decreased, implying that these browsers limit the access to and amount of hardware resources that can be utilized to enhance security.
Testing the performance in the cloud browsers produced varied result, since the performance scores depend on the resources allocated, traffic, and other cloud server limitations of the

cloud browser instance. While evaluating the fingerprinting attributes used in the present work, it was observed that the cloud browsers changed or spoofed eight out of the nine attributes when compared to real supposed values of the device which were referenced during native application tests. However, the audio attribute's fingerprinted value of all the cloud browsers except Brave and Tor were close to the device's value. This shows that audio attribute is a potential browser attribute in cloud browsers that can be used for building a unique fingerprint profile of a user.

In future more attributes can be tested for their vulnerability towards fingerprinting in cloud browsers. Moreover, a more extensive test needs to be done to verify the audio attribute as potential parameter for fingerprinting.

References

Acar, G., Juarez, M., Nikiforakis, N., Diaz, C., Gürses, S., Piessens, F., & Preneel, B. (2013, November 4). FPDetective: Dusting the web for fingerprinters. Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security - CCS '13, 1129–1140. https://doi.org/10.1145/2508859.2516674

Al-Fannah, N., Li, W., & Mitchell, C. (2018). Beyond Cookie Monster Amnesia: Real World Persistent Online Tracking (pp. 481–501). https://doi.org/10.1007/978-3-319-99136-8_26

Barth, A., Jackson, C., and Mitchell, J. C. (2008). Robust Defenses for Cross-site Request Forgery. In Proceedings of CCS'08, pages 75–88, 2008.

Cloud Browser Architecture. (2017, June 8). https://www.w3.org/TR/cloud-browser-arch/

Eckersley, P. (2010). How unique is your web browser? In Lecture Notes in Computer Science (pp. 1–18). https://doi.org/10.1007/978-3-642-14527-8_1

Fifield, D., & Egelman, S. (2015). Fingerprinting Web Users Through Font Metrics. In R. Böhme & T. Okamoto (Eds.), Financial Cryptography and Data Security (pp. 107–124). Springer. https://doi.org/10.1007/978-3-662-47854-7_7

Hothersall-Thomas, C., Maffeis, S., & Novakovic, C. (2015). BrowserAudit: Automated testing of browser security features. Proceedings of the 2015 International Symposium on Software Testing and Analysis, 37–47. https://doi.org/10.1145/2771783.2771789

Hickson, I., & Hyatt, D. (2014). HTML5: A vocabulary and associated APIs for HTML and XHTML. W3C Candidate Recommendation CR-HTML5-20140429, Apr. 2014.

Mayer, J. R. (2009). "Any person... a pamphleteer": Internet Anonymity in the Age of Web 2.0. Undergraduate Senior Thesis, Princeton University, 85. https://dataspace.princeton.edu/handle/88435/dsp01nc580n467

Mohamed, A. H., & Ismail, I. (2022). A performance comparative on most popular internet web browsers. Procedia Computer Science, 215, 589–597. https://doi.org/10.1016/j.procs.2022.12.061

MotionMark 1.2. (n.d.). Retrieved December 14, 2023, from https://browserbench.org/MotionMark1.2/about.html

Narayanan, A., & Shmatikov, V. (2008). Robust De-anonymization of Large Sparse Datasets. 2008 IEEE Symposium on Security and Privacy (sp 2008), Oakland, CA, USA, 2008, pp. 111-125. doi: 10.1109/SP.2008.33.

Palanques, M., Dipietro, R., Ojo, C. del, Malet, M., Mariño, M., & Felguera, T. (2012). Secure Cloud Browser: Model and Architecture to Support Secure WEB Navigation. 2012 IEEE 31st Symposium on Reliable Distributed Systems, 402–403. https://doi.org/10.1109/SRDS.2012.64

Sivakumar, A., Gopalakrishnan, V., Lee, S., Rao, S., Sen, S., & Spatscheck, O. (2014). Cloud is not a silver bullet: A case study of cloud-based mobile browsing. Proceedings of the 15th Workshop on Mobile Computing Systems and Applications, 1–6. https://doi.org/10.1145/2565585.2565601

Sterne, B., & Barth, A. (2012). Content Security Policy 1.0. Nov. 2012. W3C Candidate Recommendation CR-CSP-20121115.

Tendulkar, V., Snyder, R., Pletcher, J., Butler, K., Shashidharan, A., & Enck, W. (2012). Abusing cloud-based browsers for fun and profit. Proceedings of the 28th Annual Computer Security Applications Conference, 219–228. https://doi.org/10.1145/2420950.2420984

Wang, Lizhe & von Laszewski, Gregor & Younge, Andrew & He, Xi & Kunze, Marcel & Tao, Jie & Fu, Cheng. (2010). Cloud Computing: a Perspective Study. New Generation Comput. 28. 137-146. 10.1007/s00354-008-0081-5.

Wadkar, H., Mishra, A., & Dixit, A. (2014). Prevention of information leakages in a web browser by monitoring system calls. 2014 IEEE International Advance Computing Conference (IACC), 199–204. https://doi.org/10.1109/IAdCC.2014.6779320

Taivalsaari, A., Mikkonen, T., & Systä, K. (2013). Cloud Browser: Enhancing the Web Browser with Cloud Sessions and Downloadable User Interface. In J. J. (Jong H. Park, H. R. Arabnia, C. Kim, W. Shi, & J.-M. Gil (Eds.), Grid and Pervasive Computing (pp. 224–233). Springer. https://doi.org/10.1007/978-3-642-38027-3_24