Investigando a Efetividade das Contramedidas em Navegadores Modernos contra o Web Tracking

Giovana Lins Cavalcanti[1], **Eduardo Luzeiro Feitosa**[1]

[1]Instituto de Computação – Universidade Federal do Amazonas
Av. Rodrigo Otávio, nº 6200, Coroado I – CEP 69080-900 – Manaus – AM

{giovana.cavalcanti,efeitosa}@icomp.ufam.edu.br

Abstract. *Modern browsers implement countermeasures against web tracking that vary in effectiveness. This study evaluates six popular desktop browsers (Chrome, Edge, Safari, Firefox, Opera, Brave) regarding stateful (e.g., cookie syncing) and stateless (e.g., fingerprinting) tracking techniques. Using automated tests across 90 websites and fingerprinting analysis tools, results show that all browsers handle traditional stateful tracking well, but differ significantly against fingerprinting. Chrome lacks native fingerprinting protections, while Safari, Firefox, and Brave implement advanced measures. Due to automation detection, intra-browser comparisons were prioritized. The analysis supports informed browser choice and future privacy solutions.*

Resumo. *Navegadores modernos implementam contramedidas contra rastreamento web com eficácia variada. Este artigo avaliou seis navegadores desktop populares (Chrome, Edge, Safari, Firefox, Opera, Brave) quanto às técnicas stateful (ex.: cookie syncing) e stateless (ex.: fingerprinting). Por meio de testes automatizados em 90 sites e ferramentas de análise de fingerprinting, os resultados mostram que todos lidam bem com rastreamento stateful, mas diferem significativamente contra fingerprinting. O Chrome não oferece proteções nativas a essa técnica, enquanto Safari, Firefox e Brave possuem medidas avançadas. Devido à detecção de automação, as comparações foram prioritariamente qualitativas. A análise auxilia na escolha consciente de navegadores.*

1. Introdução

As técnicas de rastreamento na web (conhecidas como *web tracking*) são empregadas por diversos serviços para coletar dados dos usuários. Esses dados podem incluir desde informações técnicas como o navegador sendo utilizado, o sistema operacional, o endereço IP e detalhes do hardware, até informações mais sensíveis, como localização geográfica, preferências de navegação e histórico de acesso [Bujlow et al. 2017]. A princípio, o uso de *web tracking* pode trazer benefícios. Para empresas, por exemplo, permite a exibição de anúncios direcionados aos interesses dos usuários. Já para usuários finais, pode auxiliar na autenticação de contas, permitindo logins persistentes em serviços online [Alaca and van Oorschot 2016].

O problema é que, via de regras, serviços que usam *web tracking* não pedem o consentimento dos usuários para recolher tais dados, gerando preocupações de privacidade e segurança digital. É normal que dados pessoais coletados sejam frequentemente

compartilhados com terceiros, tipicamente rastreadores escondidos como redes de anunciantes incorporadas na maioria das páginas da web [Englehardt and Narayanan 2016], tornando a atividade de rastreio uma violação de privacidade. Ademais, devido à natureza sensível dos dados coletados, técnicas de *web tracking* podem ser utilizadas para prejudicar os usuários caso exploradas por agentes maliciosos.

Diante dessas preocupações, diferentes medidas contra o rastreamento online foram tomadas. No âmbito legislativo, múltiplos países promulgaram leis que definem as hipóteses específicas em que dados pessoais podem ser utilizados por terceiros dentro de seus territórios, a exemplo da Lei Geral de Proteção de Dados (LGPD) no Brasil [Brasil 2018] e do Regulamento Geral sobre a Proteção de Dados (GDPR) na União Europeia [Parlamento europeu e Conselho da União Europeia 2016]. Já no âmbito digital, diversos navegadores passaram a dispor de mecanismos para bloquear o rastreamento de usuários, tornando a privacidade um diferencial.

As técnicas de *web tracking* podem ser divididas em duas categorias: *stateful*, que armazena informações no dispositivo do usuário, e *stateless*, que não armazena informações no dispositivo do usuário [Mayer and Mitchell 2012]. Dentre as técnicas *stateful*, a mais conhecida é o uso dos cookies, identificadores únicos do usuário que são recuperados por sites acessados para reconhecê-lo [Bujlow et al. 2017], fáceis e simples de combater, uma vez que o próprio usuário pode, em seu navegador, apagá-los ou bloqueá-los [Ximenes et al. 2016]. Já as técnicas *stateless* tem o *fingerprinting*, técnica que coleta informações para construir uma "impressão digital" (*fingerprint*) do usuário, tipicamente por meio de chamadas para APIs [Laperdrix et al. 2020], como principal representante.

Embora existam diversos trabalhos avaliando contramedidas contra *web tracking*, eles frequentemente focam apenas em *fingerprinting* ou não se aprofundam em comparações diretas e abrangentes entre as implementações de diferentes navegadores modernos, deixando uma lacuna na compreensão das divergências de eficácia e limitações de cada um. Diante desse problema, o objetivo do artigo é avaliar e comparar os mecanismos empregados pelos navegadores mais populares (Chrome, Edge, Safari, Firefox, Opera e Brave) para impedir o rastreamento de dados na web, investigando as diferenças de implementação contra técnicas *stateful* (cookies) e *stateless* (*fingerprinting*) e o impacto de diferentes sistemas operacionais (Windows, Ubuntu e macOS) na eficácia dessas contramedidas, a fim de determinar suas limitações e lacunas de proteção.

2. Fundamentação Teórica

2.1. Cookies e Fingerprinting

Os cookies são pequenos arquivos de texto criados por sites e armazenados no dispositivo do usuário para armazenar informações como dados de login, preferências de navegação, histórico de atividades e outros dados úteis para melhorar a experiência do usuário [Barth 2011]. Por sua vez, os supercookies (ou *evercookies*) objetivam recriar cookies deletados ao armazenar seu valor em locais incomuns, como o cache do navegador ou o armazenamento localStorage [Bujlow et al. 2017]. Já o *cookie syncing* [Acar et al. 2014] é a troca de cookies entre domínios de terceiros, onde um envia cookies de identificação para o outro, permitindo que rastreadores distintos colaborem para identificar quais usuários acessaram o site onde estão presentes, juntando os cookies do

usuário. Por fim, outro método de rastreamento via cookies é o *bounce tracking*, onde um site redireciona o usuário a outro domínio para que um cookie primário de rastreamento seja armazenado, burlando o bloqueio de cookies de terceiros [MDN Web Docs 2026].

Por outro lado, o *fingerprinting* é uma técnica mais avançada que não depende do armazenamento de identificadores no lado do usuário. Em vez disso, ele coleta informações sobre as características e configurações do dispositivo do usuário, como tipos de fontes instaladas, resolução de tela, suporte a APIs gráficas e características do hardware para identificá-lo de maneira única, sem a necessidade de cookies [Laperdrix et al. 2020]. Algumas das principais APIs exploradas são a Canvas, WebGL, WebAudio e Battery Status [Cassel et al. 2022].

2.2. Mecanismos de Bloqueio dos Navegadores

O método mais tradicional e direto é bloqueio baseado em listas (*Blocklists*). O navegador utiliza listas públicas e constantemente atualizadas como as listas EasyList ou Disconnect. De modo geral, sempre que uma página web tenta carregar um script, uma imagem invisível (pixel de rastreamento) ou um anúncio, o navegador checa a origem desse arquivo. Se o domínio estiver na lista, o navegador bloqueia o download antes mesmo que ele aconteça. Isso economiza dados e acelera o carregamento da página.

Com o passar do tempo, os navegadores mais populares aderiram às iniciativas *Cookies Having Independent Partitioned State* (CHIPS) e *Storage Partitioning*, o que significa que cada site acessado possui seu próprio armazenamento de cookies e de outros dados por padrão [Privacy Sandbox 2023, Privacy Sandbox 2025]. Com essas implementações, o uso de *cookie syncing* e supercookies torna-se menos eficaz, já que impede que um cookies incorporado no site A seja lido ao visitar o site B que também o incorpora. O Safari passou a aderir à CHIPS muito depois dos outros navegadores, mas já possuía uma política restritiva de cookies de terceiros, a qual permite que eles armazenem novos cookies apenas se seu domínio fora acessado pelo usuário anteriormente, desde a versão 1.0 [WebKit 2026].

Na proteção contra *fingerprinting*, os navegadores tipicamente fazem a "mistura" dos dados, fornecendo informações genéricas ou levemente alteradas e fazendo que o *hash* gerado seja igual a milhões de outros já existentes ou diferente para cada sessão do usuário. O Safari, Firefox e Brave são exemplos de navegadores que aplicam tal contramedida [Mozilla Support 2025a, Brave Software 2020, Wilander et al. 2024]. Por fim, alguns navegadores, como Safari e Brave, também fazem o *URL Stripping* Com isso, endereços de URL que contém queries como $?utm\_source = facebook\&fbclid = XYZ123?$ são identificados como sufixos de rastreamento conhecidos e removidos antes do carregamento da página [Brave Software 2021, Wilander et al. 2024].

3. Trabalhos Relacionados

Trabalhos anteriores analisaram técnicas de *web tracking* e contramedidas possíveis e aplicadas em navegadores sob diferentes contextos. [Englehardt and Narayanan 2016] desenvolveram a ferramenta OpenWVM para automatizar a busca por mecanismos de *web tracking* nos top 1 milhão de sites da mesma base de dados. Eles também testaram configurações e extensões anti-rastreamento para o navegador Firefox da época, observando que eram eficazes em reduzir a quantidade de cookies de terceiros, porém falharam em detectar a maioria de scripts de *fingerprinting* nos sites.

[Al-Fannah and Li 2017] compararam como os navegadores mais populares da época retornavam atributos de *fingerprinting*, acessando uma página web de coleta de *fingerprint* e analisando os resultados. Na pesquisa, navegadores que não mascaram os atributos foram classificados de alta vulnerabilidade ao *fingerprinting*; aqueles que mascaravam os atributos a cada sessão de navegação, de média vulnerabilidade; e aqueles que mascaravam os atributos a cada visita à página, de baixa vulnerabilidade. Dentre os 5 navegadores para desktop testados (Chrome, Internet Explorer, Firefox, Edge e Safari), o Firefox e o Safari foram consistentemente os menos vulneráveis ao *fingerprinting*.

[Cassel et al. 2022] apresentaram um estudo sobre rastreamento web em navegadores desktop, a fim de comparar o rastreamento entre navegadores emulados e o Selenium, entre os navegadores tradicionais (Chrome e Firefox) e entre navegadores focados em privacidade (Brave, Tor, Firefox Focus, DuckDuckGo, Ghostery). Os resultados mostram que: (i) navegadores emulados e o Selenium introduziram distorções significativas nos dados, tornando-os menos confiáveis; (ii) os navegadores focados em privacidade reduziram substancialmente o rastreamento (especialmente Brave), mas foram menos eficazes contra entidades menos prevalentes e variam muito entre si em recursos e eficácia; (iii) a localização geográfica influencia a quantidade de rastreamento (maior na América do Norte).

[Pan and Ruiz-Martínez 2023] analisaram se as melhorias de privacidade implementadas nas versões 83, 86 e 90 do navegador Chrome foram efetivas na proteção dos usuários contra rastreamento web. Os resultados mostraram que, entre as versões 83 e 90, houve uma redução modesta de 7,55% no número total de rastreadores e de 4,76% nos elementos do Google Analytics, graças à introdução dos mecanismos *SameSite header*, *Referrer-Policy* e c*ache partitioning*. No entanto, nenhuma proteção contra técnicas de *fingerprinting* (Canvas, WebGL, WebRTC e objetos informativos) foi implementada nas versões analisadas, permanecendo inalterada.

É neste contexto que o presente trabalho busca expandir a análise dos mecanismos contra *web tracking* empregados por navegadores modernos, concentrada na implementação e eficácia das contramedidas, destacando quais técnicas de rastreamento elas combatem.

4. Metodologia

Neste estudo, apenas navegadores para desktop foram considerados, pois os navegadores para dispositivos móveis podem apresentar peculiaridades que dificultariam a comparação. A metodologia deste trabalho foi dividida nas seguintes etapas: (1) definição dos navegadores e o ambiente onde serão investigados, (2) definição dos testes práticos e da avaliação das contramedidas, (3) execução e comparação do desempenho dos navegadores na etapa anterior.

4.1. Definição dos Navegadores e Ambientes de Testes

A escolha dos navegadores foi feita a partir dos dados públicos de tráfego da Fundação Wikimedia[1], da rede de distribuição de conteúdo Cloudfare[2] e dos serviços de Web analy-

[1]https://analytics.wikimedia.org/dashboards/browsers/#all-sites-by-browser/browser-family-timeseries
[2]https://radar.cloudflare.com/explorer

tics Statcounter[3] e W3Counter[4]. Comparando os dados de agosto de 2025 dessas quatro fontes, foi constatado que os navegadores para desktop mais utilizados na web são Chrome, Edge, Safari, Firefox, Opera e Brave. Nota-se que três desses — Edge, Opera e Brave — são baseados no Chromium, projeto código-aberto do Chrome. Uma vez que o Safari está disponível apenas para dispositivos fabricados pela Apple, optamos por não criar uma máquina virtual, mas sim utilizar um dispositivo real para conduzir os testes. Também optamos por não testar os outros navegadores nesse dispositivo Mac, já que aumentaria drasticamente a superfície de rastreamento de um dispositivo para uso diário.

Detalhes dos ambientes de testes encontram-se na Tabela 1, incluindo a resolução da tela, o idioma e o fuso horário do sistema, atributos comumente utilizados para gerar um *fingerprint*.

Tabela 1. Configurações dos ambientes utilizados para testes.

SO	Tipo	Resolução	Idioma	Fuso	Navegadores
Ubuntu 24.04	Máquina virtual	1280x800	pt-br	UTC-4	Chrome 147.0.7727.137 Edge 147.0.3912.98 Firefox 150.0.1 Opera 130.0.5.5847.41 Brave 1.89.145
Windows 11	Máquina virtual	1920x1080	en-us	UTC-8	Chrome 148.0.7789.97 Edge 148.0.3967.54 Firefox 150.0.2 Opera 131.0.5.5877.24 Brave 1.90.121
macOS 12.7.6	Local	1440x900	pt-br	UTC-4	Safari 17.6

4.2. Definição dos Testes e Avaliação das Contramedidas

As sessões de testes foram definidas em duas fases: (1) uso das configurações padrão do navegador e (2) uso das configurações de proteção contra *web tracking*. A implementação e a lista completa dos sites analisados encontram-se no repositório `https://github.com/bolachamole/testes-tracking`.

A avaliação das medidas contra rastreamento *stateful* foi feita com base na análise das implementações de cada navegador e das seguintes métricas: o número de instâncias de *cookie syncing* e a diferença do número de requisições a rastreadores entre as configurações testadas. Como a principal medida adotada contra cookies e supercookies é particionar o armazenamento, considerou-se que utilizá-los como métricas traria uma visão incompleta das contramedidas dos navegadores. Para coletar as métricas, foram selecionados os top 50 sites de notícia e de e-commerce mais acessados de acordo com a plataforma Similarweb[5], pois possuem alta quantidade de cookies de terceiros em comparação a outras categorias, o que possibilita melhor observação dos mecanismos contra essa técnica de rastreamento [Englehardt and Narayanan 2016, Singh et al. 2025].

Inicialmente, cada execução a cada um dos sites durou 90 segundos, mas percebemos que a frequência de requisições feitas decresce após 60 segundos e que não havia

[3]https://gs.statcounter.com/browser-market-share/desktop/worldwide
[4]https://w3counter.com
[5]https://similarweb.com/top-websites

requisições de rastreamento detectadas em alguns sites. Assim, removemos esses sites da seleção e ajustamos o tempo para 60 segundos, poupando tempo e memória. A condução dos acessos foi feita por meio do Selenium WebDriver[6], utilizando a ferramenta mitmproxy[7] para capturar as requisições HTTP e, assim, calcular as métricas.

Com base em trabalhos anteriores, os cookies foram contados por meio do cabeçalho Set-Cookie e identificados como rastreadores se um ou mais de seus valores são reconhecidos pela expressão regular *"[a-zA-Z0-9_=-]+"* [Chen et al. 2021, Munir et al. 2023]. Os cookies de rastreamento detectados foram, então, utilizados para detectar *cookie syncing* ao verificar se o valor de um deles se encontra na URL da requisição e se o domínio do cookie e o do domínio da requisição não pertencem ao mesmo dono. Já as URLs de rastreamento foram identificadas por meio das listas públicas *EasyPrivacy*, utilizada pelos bloqueadores de rastreamento dos navegadores Safari, Opera e Brave, e *Disconnect*, utilizada pelo Edge e Firefox.

Vale destacar que o Selenium WebDriver pode ser detectado pelas propriedades de JavaScript que ele altera, o que pode mudar o comportamento de alguns sites. Para mitigar os efeitos da detecção de automação, [Cassel et al. 2022] realizou um teste preliminar em uma amostra de 10 sites aleatórios, verificando a presença de scripts anti-bot conhecidos (ex.: Cloudflare). Observou que o Chrome, conduzido pelo Selenium, recebia 20% de requisições de terceiros a mais do que o Chrome sem o mesmo, enquanto o Firefox recebia 84% de requisições de terceiros a menos. Assim, os resultados comparativos entre navegadores devem ser interpretados com cautela, priorizando-se as análises intra-navegador. Por isso, as métricas obtidas nos testes foram utilizadas apenas para averiguar a efetividade das contramedidas de um navegador comparada ao mesmo antes de ativá-las, enquanto a comparação entre navegadores diferentes foi feita por meio da análise das implementações.

Para avaliar a eficácia das contramedidas dos navegadores contra códigos de *fingerprinting*, são observados *fingerprints* coletados pelo serviço AmIUnique[8], referência utilizada por múltiplos estudos [Laperdrix et al. 2020, Lin et al. 2023], e pela ferramenta CreepJS[9], projeto de código aberto que objetiva explorar fraquezas de tecnologias anti-*fingerprinting*. Ambos apresentam quais atributos foram utilizados para gerar o *fingerprint*, permitindo visualizar quais APIs o navegador foi capaz de bloquear ou modificar e quais valores foram alterados para impedir o *fingerprinting* preciso do usuário.

Para cada ambiente de testes, os experimentos foram repetidos duas vezes para cada fase. A avaliação das métricas foi feita por meio da média dos resultados de cada execução. Já a comparação dos *fingerprints* envolveu verificar mudanças no *hash* antes e após ativar as proteções e comparar os valores reportados entre configurações e navegadores diferentes.

5. Resultados

Esta seção apresenta os resultados da (1) detecção de rastreadores, (2) das medidas contra o rastreamento *stateful* e (3) das medidas contra o rastreamento *stateless*.

[6]https://selenium.dev
[7]https://mitmproxy.org
[8]https://amiunique.org
[9]https://github.com/ abrahamjuliot/creepjs

5.1. Detecção de Rastreadores

A Figura 1 ilustra os resultados obtidos da intercepção do tráfego de rede feita em cada navegador. Ela sumariza a capacidade dos navegadores de bloquear domínios conhecidos por utilizar técnicas de *web tracking*.

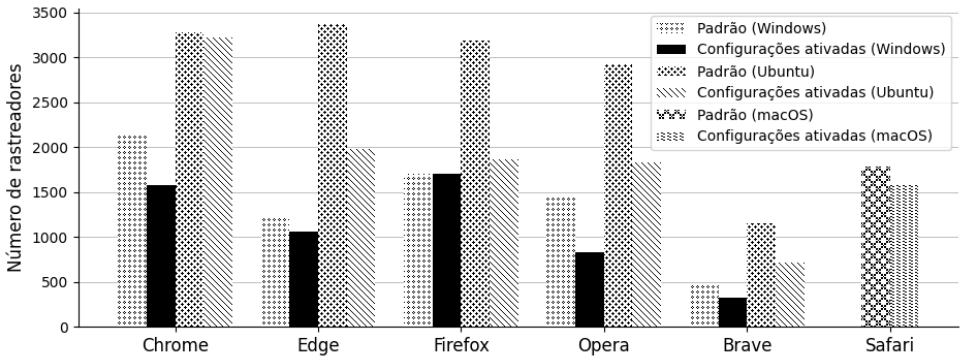

Figura 1. Quantidade de rastreadores observada em cada ambiente de testes.

Observamos que, em geral, os navegadores foram mais eficazes em detectar rastreadores no Ubuntu, enquanto, no Windows, apesar da diferença antes e depois de ativar as configurações de privacidade manter-se pequena, foram detectadas menos requisições feitas. Uma hipótese é que a presença do sistema operacional "Linux"no cabeçalho *User-Agent* das requisições alerta as proteções anti-bot do domínio acessado e, assim, são enviadas mais requisições de rastreadores para identificar o dispositivo ou confirmar essa suposição. No contexto das contramedidas únicas de cada navegador, notamos que os navegadores Opera e Brave aplicam os filtros independentemente da categoria do site e bloqueiam conteúdo de anunciantes, enquanto os outros navegadores abrem exceções no modo padrão e, mesmo com as configurações avançadas ativadas, permitem anúncios.

Uma peculiaridade vista no Edge é que, por padrão, seu bloqueador leva em conta o quão frequente é o engajamento com o site acessado e seus subdomínios, ignorando as imposições dos filtros (exceto para *fingerprinters* e criptomineradores) para sites com base nessa medida [Microsoft Edge Team 2019]. Isso garante que os sites mais acessados por um usuário não quebrem por conta das contramedidas, mas pode se tornar problemático se eles forem propensos a abusar de *Web Tracking*. Ele permite mudar o modo de proteção para "estrito", desativando esse comportamento e ativando o bloqueio de requisições a todas as categorias de rastreadores da lista do serviço *Disconnect*.

O Chrome difere dos outros navegadores por não possuir um mecanismo de bloqueio de rastreamento. Ao invés disso, o navegador oferece dois modos de "navegação segura" pensados para proteger o usuário de *malware* e *phishing*, onde a proteção padrão utiliza de uma tabela *hash* para verificar se estão registrados como tal, enquanto a proteção reforçada envia as URLs visitadas e uma amostra do seu conteúdo aos servidores da Google para verificar os riscos em tempo real [Jin et al. 2025]. Como alguns usos de *web tracking* podem se encaixar como maliciosos, era esperado que essa configuração traria resultados. De fato, foi constatado que a ativação desse mecanismo pode reduzir o número de requisições a rastreadores no Windows, visível na Figura 1.

5.2. Medidas contra o Rastreamento *Stateful*

A Figura 2 apresenta uma comparação visual das quantidades de instâncias de *cookie syncing* registradas nos ambientes de testes, permitindo avaliar o impacto das configurações padrão em relação às configurações de proteção ativadas no Windows, Ubuntu e macOS.

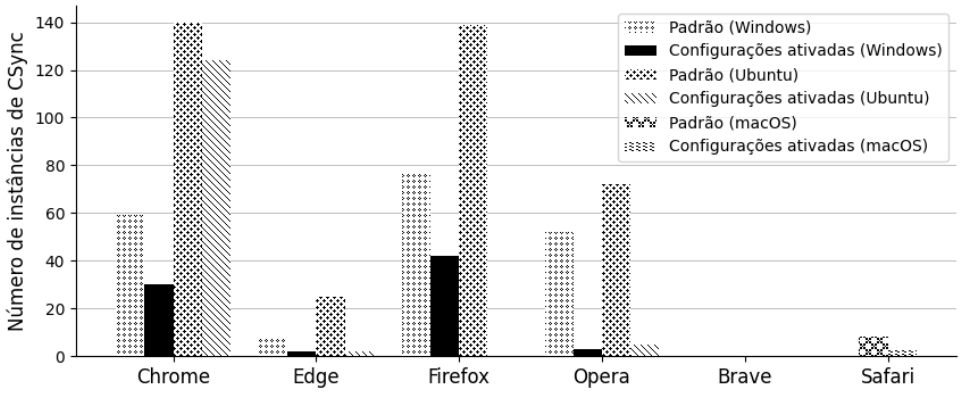

Figura 2. Cookie Syncing observados em cada ambiente de testes.

Os resultados ilustrados revelam que o Chrome apresenta um número elevado de ocorrências dessa técnica e, ao contrário dos outros, foi incapaz de reduzí-las consideravelmente após ativar as configurações de privacidade. Em contraste, o Edge e o Opera não aparentam ter implementado mitigações diretamente contra o *web tracking stateful* além daquelas presentes no Chromium, mas possuem bloqueadores de rastreadores integrados que diminuíram sua propensão a receber requisições de *cookie syncing*.

Dentre as contramedidas únicas de cada navegador, destacam-se as oferecidas pelo Safari, Firefox, e Brave. O Safari possui uma política restritiva de cookies de terceiros, a qual permite que eles armazenem novos cookies apenas se seu domínio fora acessado pelo usuário anteriormente, e força os sites a terem a permissão de usar a *API Storage Access* para armazená-los [WebKit 2026]. Para impedir o *cookie syncing* e outros compartilhamentos de dados entre sites, o navegador remove conteúdo de rastreamento de URLs redirecionadores ao usar a navegação privativa [Wilander et al. 2024].

O Firefox, por sua vez, apresenta duas modalidades de proteção contra rastreamento nas suas configurações de privacidade: normal e rigoroso. Ambos os modos bloqueiam cookies de rastreamento entre sites e o modo rigoroso também deleta os vetores de armazenamento associados a *bounce trackers* [Mozilla Support 2026]. Já o Brave possui um conjunto mecanismos anti-*web tracking* ativado por padrão, o qual pode ser alterado entre o modo padrão e agressivo. No primeiro, há mitigações contra o *cookie syncing* e *bounce tracking*, removendo parâmetros de rastreamento das URLs. Já no segundo, são adicionadas mais proteções contra o *bounce tracking*, redirecionando o usuário à página de destino verdadeira ao detectar que a URL visitada utiliza essa técnica, por meio de uma lista mantida pelos desenvolvedores [Brave Software 2021].

5.3. Medidas contra o Rastreamento *Stateless*

Dentre os seis navegadores avaliados neste trabalho, apenas três oferecem configurações explicitamente contra o *fingerprinting*: Safari, Firefox e Brave. A síntese da análise feita

encontra-se na Tabela 2.

Tabela 2. Comparativo entre os Navegadores

Atributo	Chrome	Edge	Safari	Firefox	Firefox (RFP)	Opera	Brave
Battery API	X	X	-	-	-	X	M+
Canvas API	X	X	M	M	M	X	M
Fontes	X	X	O	O	O	X	O
Fuso horário	X	X	X	X	M	X	X
Idiomas	X	X	O	X	M	X	O
Tela	X	X	M+	M+	M+	X	M+
User-Agent	X	X	X	X	M	X	X
WebAudio API	X	X	M	M	M	X	M
WebGL API	X	X	M	M	M	X	M

Note: X indica que nada foi feito. M indica que o valor foi modificado. M+ indica que o valor foi modificado para um valor fixo. O indica que o valor real foi ocultado e - indica que a API não é suportada pelo navegador. RFP significa *Resist Fingerprinting*.

Os três empregam uma abordagem similar: bloquear scripts de *fingerprinting* já conhecidos na web e dificultar o rastreamento entre sites ao deixar atributos únicos para cada site, reduzindo a precisão do *fingerprint*, ou o mesmo para todos, reduzindo a entropia [Wilander et al. 2024, Mozilla Support 2025a, Brave Software 2020]. Esse comportamento foi observado principalmente nas APIs Canvas e WebGL, conhecidas por sua alta entropia [Laperdrix et al. 2020].

Enquanto o Brave tem suas contramedidas ativadas por padrão, o Safari e Firefox disponibilizam suas proteções apenas em janelas privadas inicialmente, mas permitem que sejam ativadas para todas as janelas nas configurações. O Firefox ainda possui uma configuração escondida na página avançada de preferências $about : config$, ativada ao mudar o valor da variável $privacy.resistFingerprinting$ para "true". Ela é chamada de RFP (*Resist Fingerprinting*). Essa configuração combate o *fingerprinting* de forma mais rigorosa, alterando múltiplos atributos como o fuso horário do sistema e injetando grande quantidade de ruído nas APIs, razão pela qual recomenda-se que seja usada apenas por usuários experientes, já que pode impactar negativamente na experiência do usuário [Mozilla Support 2025b].

Em contrapartida, ao analisar a documentação e as *fingerprints* obtidas dos navegadores restantes, percebeu-se que suas proteções contra o *web tracking stateless* são inexistentes ou escassas. Nenhum dos três oferecem medidas para mitigar o *fingerprinting* de atributos, apresentando *hash* idêntico nas duas fases de teste em ambos ambientes, o que não foi o caso dos outros navegadores. Porém, os bloqueadores de rastreamento do Edge e Opera bloqueiam scripts de *fingerprinting* conhecidos por padrão, ao contrário do Chrome, o qual não oferece nenhuma proteção contra o *fingerprinting*. Isso é uma grande lacuna desse navegador em comparação aos outros.

6. Conclusão

Este artigo visou avaliar e comparar a efetividade das medidas contra *web tracking* dos navegadores modernos. Foi observado que, em geral, os navegadores possuem medidas

sólidas para o combate aos métodos *stateful*, mas alguns deixam a desejar no combate ao *fingerprinting*. Tal conjuntura é especialmente preocupante no caso do Chrome, o qual é o navegador mais popular entre usuários da web, superando o tráfego de outros em mais de 50% ou até 60% em certas análises. Por outro lado, nota-se os esforços do Safari, Firefox e Brave para combater essa técnica.

Assim, a análise apresentada poderá servir como base para futuras investigações sobre a eficácia de medidas de proteção e para o desenvolvimento de novas tecnologias de preservação da privacidade, sobretudo na web. A principal limitação do estudo foi a dificuldade de comparação justa de métricas devido a detecção de automatização dos sites visitados, logo, considera-se que o desenvolvimento e aprimoramento de ferramentas para testes automatizados com navegadores seja crucial para este tipo de análise comparativa.

Como a pesquisa vinculada a este trabalho encontra-se em desenvolvimento, os próximos passos serão os seguintes:

- Expansão dos testes utilizando *frameworks* de automação alternativos e técnicas de evasão de detecção (anti-bot *fingerprinting*) para mitigar as distorções causadas pelo Selenium WebDriver.
- Aprofundamento da investigação sobre os motivos analíticos que levam os rastreadores a alterarem seu comportamento agressivo em dependência direta do sistema operacional.
- Uma vez que a heterogeneidade geográfica dos sites analisados pode ser vista como uma limitação do estudo e pode introduzir viés nos resultados (a intensidade de rastreamento varia conforme a localização do usuário e do domínio), pretende-se testar amostras balanceadas por região.
- Investigação das diferenças no comportamento de rastreadores conforme o sistema operacional por meio da manipulação de strings de *User-Agent* e da análise de estratégias de evasão adaptativas, utilizando ferramentas de emulação de SO no nível de kernel para isolar essas variáveis.

Agradecimentos

Os autores agradecem ao Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas (PPGI/UFAM) e ao Instituto Federal do Amazonas (IFAM). Esta pesquisa foi parcialmente financiada, conforme previsto nos Arts. 21 e 22 do Decreto No. 10.521/2020, nos termos da Lei Federal No. 8.387/1991, através do convênio No. 015/2025, firmado entre ICOMP/UFAM, Digiboard da Amazônia Ltda e Lenovo Ltda.

Referências

Acar, G., Eubank, C., Englehardt, S., Juarez, M., Narayanan, A., and Diaz, C. (2014). The web never forgets: Persistent tracking mechanisms in the wild. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, page 674–689, New York, NY, USA. ACM.

Al-Fannah, N. M. and Li, W. (2017). Not all browsers are created equal: Comparing web browser fingerprintability. In Obana, Satoshi Chida, K., editor, *Advances in Information and Computer Security*, pages 105–120. Springer International Publishing.

Alaca, F. and van Oorschot, P. C. (2016). Device fingerprinting for augmenting web authentication: classification and analysis of methods. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, ACSAC '16, page 289–301, New York, NY, USA. ACM.

Barth, A. (2011). HTTP State Management Mechanism. RFC 6265.

Brasil (2018). Lei nº 13.709, de 14 de agosto de 2018. lei geral de proteção de dados pessoais (LGPD). Disponível em `https://planalto.gov.br/ccivil_03/_Ato2015-2018/2018/Lei/L13709.htm`. Acesso em: 4 de fevereiro de 2026.

Brave Software (2020). Fingerprinting protections. Disponível em `https://github.com/brave/brave-browser/wiki/Fingerprinting-Protections`. Acesso em: 7 de maio de 2026.

Brave Software (2021). Improving privacy by improving web compatibility. Disponível em `https://brave.com/privacy-updates/9-web-compat-blocking/`. Acesso em: 15 de maio de 2026.

Bujlow, T., Carela-Español, V., Solé-Pareta, J., and Barlet-Ros, P. (2017). A survey on web tracking: Mechanisms, implications, and defenses. *Proceedings of the IEEE*, 105(8):1476–1510.

Cassel, D., Lin, S.-C., Buraggina, A., Wang, W., Zhang, A., Bauer, L., Hsiao, H.-C., Jia, L., and Libert, T. (2022). Omnicrawl: Comprehensive measurement of web tracking with real desktop and mobile browsers. *Proceedings on Privacy Enhancing Technologies*, 2022(2):227–252.

Chen, Q., Ilia, P., Polychronakis, M., and Kapravelos, A. (2021). Cookie swap party: Abusing first-party cookies for web tracking. In *Proceedings of the Web Conference 2021*, WWW '21, page 2117–2129, New York, NY, USA. ACM.

Englehardt, S. and Narayanan, A. (2016). Online tracking: A 1-million-site measurement and analysis. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 1388–1401, New York, NY, USA. ACM.

Jin, B., Li, H., and Zou, Y. (2025). Impact of extensions on browser performance: An empirical study on google chrome. *Empirical Software Engineering*, 30(4):103.

Laperdrix, P., Bielova, N., Baudry, B., and Avoine, G. (2020). Browser fingerprinting: A survey. *ACM Trans. Web*, 14(2).

Lin, X., Araujo, F., Taylor, T., Jang, J., and Polakis, J. (2023). Fashion faux pas: Implicit stylistic fingerprints for bypassing browsers' anti-fingerprinting defenses. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 987–1004.

Mayer, J. R. and Mitchell, J. C. (2012). Third-party web tracking: Policy and technology. In *2012 IEEE Symposium on Security and Privacy*, pages 413–427.

MDN Web Docs (2026). Bounce tracking mitigations. Disponível em `https://developer.mozilla.org/en-US/docs/Web/Privacy/Guides/Bounce_tracking_mitigations`. Acesso em: 3 de maio de 2026.

Microsoft Edge Team (2019). Improving tracking prevention in microsoft edge. Disponível em `https://blogs.windows.com/msedgedev/2019/12/03/`

improving-tracking-prevention-microsoft-edge-79. Acesso em: 15 de maio de 2026.

Mozilla Support (2025a). Firefox's protection against fingerprinting. Disponível em https://support.mozilla.org/en-US/kb/firefox-protection-against-fingerprinting. Acesso em: 7 de maio de 2026.

Mozilla Support (2025b). Resist fingerprinting. Disponível em https://support.mozilla.org/en-US/kb/resist-fingerprinting. Acesso em: 29 de janeiro de 2026.

Mozilla Support (2026). Enhanced tracking protection in firefox for desktop. Disponível em https://support.mozilla.org/en-US/kb/enhanced-tracking-protection-firefox-desktop. Acesso em: 3 de maio de 2026.

Munir, S., Siby, S., Iqbal, U., Englehardt, S., Shafiq, Z., and Troncoso, C. (2023). Cookiegraph: Understanding and detecting first-party tracking cookies. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, CCS '23, page 3490–3504, New York, NY, USA. ACM.

Pan, R. and Ruiz-Martínez, A. (2023). Evolution of web tracking protection in chrome. *Journal of Information Security and Applications*, 79:103608.

Parlamento europeu e Conselho da União Europeia (2016). Regulamento (UE) 2016/679 do parlamento europeu e do conselho. Disponível em https://eur-lex.europa.eu/legal-content/PT/TXT/?uri=CELEX:32016R0679. Acesso em: 27 de fevereiro de 2025.

Privacy Sandbox (2023). Cookies having independent partitioned state (chips). Disponível em https://privacysandbox.google.com/cookies/chips. Acesso em: 15 de maio de 2026.

Privacy Sandbox (2025). Storage partitioning. Disponível em https://privacysandbox.google.com/cookies/storage-partitioning. Acesso em: 15 de maio de 2026.

Singh, N., Do, Y., Yu, Y., Fouad, I., Kim, J., and Kim, H. (2025). Crumbled cookies: Exploring e-commerce websites' cookie policies with data protection regulations. *ACM Trans. Web*, 19(1).

WebKit (2026). Tracking prevention in webkit. Disponível em https://webkit.org/tracking-prevention/. Acesso em: 12 de maio de 2026.

Wilander, J., Wolfe, C., Finkel, M., Hsieh, W., and Holleman, K. (2024). Private browsing 2.0. Disponível em https://webkit.org/blog/15697/private-browsing-2-0/. Acesso em: 7 de maio de 2026.

Ximenes, P., Correia, M., Mello, P., Carvalho, F., Franklin, M., and Andrade, R. (2016). Tarp fingerprinting: Um mecanismo de browser fingerprinting baseado em html5 resistente a contramedidas. In *Anais do XVI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pages 100–113, Porto Alegre, RS, Brasil. SBC.