University of Ljubljana
Faculty of computer and information science

Jordan Lazov

Web Tracking Without Cookies: Methods, Privacy Threats and Countermeasures

bachelor thesis

professional study program
first cycle
computer and information science

Mentor: viš. pred. dr. David Jelenc

Ljubljana, 2025

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Jordan Lazov

Spletno sledenje brez piškotkov: Metode, grožnje zasebnosti in protiukrepi

diplomsko delo

visokošolski strokovni študijski program
prve stopnje
računalništvo in informatika

Mentor: viš. pred. dr. David Jelenc

Ljubljana, 2025

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Candidate: Jordan Lazov

Title: Web Tracking Without Cookies: Methods, Privacy Threats and Countermeasures

Task type: Bachelor thesis for the professional study programme at the Faculty of Computer and Information Science

Mentor: viš. pred. dr. David Jelenc

Description:

Investigate modern web tracking methods that do not rely on cookies, with a focus on their privacy implications. Analyze existing countermeasures implemented in browsers, extensions, and search engines, and evaluate their effectiveness. Develop an automated testing framework to empirically assess these privacy tools. Combine theoretical analysis with practical experimentation, and conclude with recommendations for improving user privacy.

Kandidat: Jordan Lazov

Naslov: Spletno sledenje brez piškotkov: Metode, grožnje zasebnosti in protiukrepi

Vrsta naloge: Diplomska naloga na visokošolskem programu prve stopnje Računalništvo in informatika

Mentor: viš. pred. dr. David Jelenc

Opis:

Raziščite sodobne metode spletnega sledenja, ki ne temeljijo na piškotkih, s poudarkom na njihovem vplivu na zasebnost uporabnikov. Analizirajte protiukrepe, ki jih ponujajo spletni brskalniki, brskalniški vtičniki in spletni iskalniki ter ocenite njihovo učinkovitost. Razvijte avtomatizirano ogrodje za vrednotenje, s katerim boste empirično preverili uspešnost teh orodij. Združite teoretične ugotovitve in praktične rezultate ter podajte priporočila za varovanje zasebnosti na spletu.

I would like to express my sincere gratitude to viš. pred. dr. David Jelenc for his exceptional mentorship, unwavering support, and prompt feedback throughout the writing of this thesis—his guidance was instrumental in helping me achieve my best work. I am deeply grateful to my family for their constant encouragement and understanding during the many hours dedicated to this project, and to my girlfriend for her patience and belief in me throughout this journey. Finally, I want to acknowledge everyone who supported me during my four years of study; your presence through both the challenging and rewarding moments has been invaluable in making this achievement possible.

Contents

List of used abbreviations

Abbreviation	English	Slovenian
HTTP	Hypertext Transfer Protocol	Protokol Za Prenos Hipertek-sta
GDPR	General Data Protection Regulation	Splošna uredba o varstvu podatkov
CCPA	California Consumer Privacy Act	Zakon o zasebnosti potrošnikov v Kaliforniji
HTML	Hypertext Markup Language	Jezik za označevanje hipertek-sta
CSS	Cascading Style Sheets	Kaskadne slogovne predloge
IP	Internet Protocol	Internetni protokol
FQDN	Fully Qualified Domain Name	Popolnoma kvalificirano domensko ime
URL	Uniform Resource Locator	Enotni lokator vira
TCP	Transimission Control Protocol	Protokol za nadzor prenosa
TOR	The Onion Router	Usmerjevalnik TOR
CLI	Command Line Interface	Vmesnik ukazne vrstice
AES-NI	Advanced Encryption Standard New Instructions	Nova navodila za napredni šifrirni standard
API	Application Programming Interface	Programski vmesnik aplikacije
GPU	Graphical Processing Unit	Grafična procesna enota
UUID	Universally unique identifier	Univerzalno edinstven identifikator
DNS	Domain Name Service	Storitev domenskih imen
MAC	Media Access Control	Nadzor dostopa do medijev
DOM	Document Object Model	Objektni model dokumenta

Abstract

Title: Web Tracking Without Cookies: Methods, Privacy Threats and Countermeasures

Author: Jordan Lazov

This thesis examines modern web tracking techniques without the use of cookies and their impact on online privacy. Through comprehensive analysis of fingerprinting, bounce tracking, and link decorating methods, it evaluates the effectiveness of current privacy protection measures across different browsers, search engines, and privacy-enhancing extensions. The research combines theoretical investigation with practical testing using an automated tool to assess various privacy solutions. Results demonstrate that browser engines alone provide insufficient protection, and that optimal privacy requires a layered approach combining trusted browsers with appropriate extensions and different search engines for navigating the web based on the individual user's needs. The study concludes with recommendations for privacy-conscious browsing practices suitable for both technical and non-technical users.

Keywords: privacy, cookies, tracking, web fingerprinting, link decorating, bounce tracking, browsers, extensions, search engines.

Povzetek

Naslov: Spletno sledenje brez piškotkov: Metode, grožnje zasebnosti in protiukrepi

Avtor: Jordan Lazov

Diplomska naloga preučuje sodobne tehnike sledenja brez piškotkov in njihov vpliv na spletno zasebnost. Skozi celovito analizo metod prstnih odtisov, preusmerjenega sledenja in dekoriranja povezav ocenjuje učinkovitost trenutnih ukrepov za zaščito zasebnosti v različnih brskalnikih, iskalnikih in razširitvah za izboljšanje zasebnosti. Raziskava združuje teoretično preiskavo s praktičnim testiranjem z uporabo avtomatiziranega orodja za oceno različnih rešitev za zasebnost. Rezultati kažejo, da brskalniški pogoni sami ne zagotavljajo zadostne zaščite in da optimalna zasebnost zahteva večplasten pristop, ki združuje zaupanja vredne brskalnike z ustreznimi razširitvami in različnimi iskalniki za navigacijo po spletu glede na potrebe posameznega uporabnika. Študija se zaključi s priporočili za prakse brskanja, ki upoštevajo zasebnost in so primerne tako za tehnične kot netehnične uporabnike.

Ključne besede: zasebnost, piškotki, sledenje, spletno prstno odtisovanje, okrasitev povezav, preusmeritev sledenja, brskalniki, razširitve, iskalniki.

Razširjeni Povzetek

V sodobnem digitalnem okolju, kjer uporabniki vsakodnevno uporabljajo spletne storitve za delo, komunikacijo, zabavo in izobraževanje, postaja varovanje spletne zasebnosti ključno vprašanje. Razmah množično zbiranje podatkov in razvoj novih analitičnih orodij so povzročili premik v načinu, kako spletna podjetja sledijo uporabnikom. Čeprav zakonodajni ukrepi, kot je Splošna uredba o varstvu podatkov (GDPR), omejujejo uporabo piškotkov za sledenje, se industrija vse bolj zanaša na metode, ki ne zahtevajo neposredne privolitve uporabnika — tako imenovane tehnike sledenja brez piškotkov. Namen te diplomske naloge je raziskati te napredne metode, oceniti njihove vplive na spletno zasebnost in razviti rešitve za obrambo pred njimi, ki so dostopne tako tehničnim kot netehničnim uporabnikom.

Osrednji poudarek dela je na treh najpomembnejših oblikah sledenja brez piškotkov: prstnem odtisovanju (angl. fingerprinting), preusmeritvenom sledenju (angl. bounce tracking) in dekoriranju povezav (angl. link decoration). Prstno odtisovanje vključuje zbiranje različnih lastnosti naprave in brskalnika (kot so ločljivost zaslona, pisave, vtičniki, jezikovne nastavitve, omrežni parametri) ter njihovo združevanje v unikaten identifikator. Ta pristop omogoča sledenje brez potrebe po nameščanju lokalnih podatkov, kar bistveno otežuje njegovo zaznavanje in blokiranje. Preusmeritveno sledenje temelji na začasnem preusmerjanju uporabnikov prek sledilnih domen, ki omogočajo beleženje obiskov. Dekoriranje povezav pa vključuje vnašanje identifikatorjev neposredno v naslove URL, kar omogoča sledenje uporabnika pri vsakem kliku, tudi če uporabljajo zaščitne ukrepe proti piškotkom.

Teoretični del naloge poleg podrobne razčlembe posameznih metod vsebuje tudi pregled drugih, manj znanih tehnik, kot so sledenje prek polja ETag v zaglavju spletnega zahtevka. Poudarek je tudi na preučitvi obstoječih protiukrepov, ki jih nudijo sodobni spletni brskalniki, vgrajene funkcije za zaščito pred sledenjem, odprtokodnih razširitev za brskalnike in na zasebnost usmerjeni spletni iskalniki. Analiza vključuje brskalnike Chrome, Edge, Firefox, Safari, Brave kot tudi manj znane, kjer se pokažejo bistvene razlike v privzeti zaščiti in podpori naprednim funkcijam, kot so izolacija konteksta, zaščita pred prstnim odtisovanjem in avtomatsko odstranjevanje parametrov iz naslove URL. Prav tako je ocenjena učinkovitost razširitev, kot so uBlock Origin, Privacy Badger, Ghostery, ClearURLs in Canvas Blocker, ter zmogljivosti iskalnikov, kot so Google Search, Microsoft Bing, Yahoo, DuckDuckGo, Brave Search, Startpage, Mojeek, Qwant kot tudi manj znanih, ki obljubljajo boljšo zasebnost pri iskanju informacij.

Praktični del naloge dopolnjuje teoretične ugotovitve z razvojem lastnega orodja za avtomatizirano testiranje, ki omogoča ponovljivo, primerljivo in sistematično vrednotenje zaščitnih ukrepov. Orodje, zasnovano v programskem jeziku TypeScript s pomočjo ogrodja Playwright, omogoča simulacijo dejanskih uporabniških sej v različnih kombinacijah brskalnikov, razširitev in iskalnikov. Med testiranjem se izvaja nabor spletnih interakcij na dejanskih spletnih mestih, pri čemer orodje beleži vedenje brskalnika, prisotnost identifikatorjev, sledenje prek preusmeritev in druge signale. Poleg tega omogoča zajem in analizo vsebine spletnih zahtevkov in odgovorov (vključno z izvajanjem programske kode JavaScript), kar omogoča natančnejše odkrivanje prisotnih tehnik sledenja. S tem je zagotovljena visoka stopnja objektivnosti in ponovljivosti rezultatov, kar povečuje raziskovalno vrednost naloge.

Ugotovitve pokažejo, da noben posamezen brskalnik ne nudi popolne zaščite. Celo brskalniki z vgrajenimi mehanizmi za zaščito pred sledenjem lahko podležejo naprednim oblikam prstnega odtisovanja ali preusmeritvnega sledenja. Uporaba razširitev bistveno izboljša zaščito, vendar ne zagotavlja popolne varnosti brez dodatnih ukrepov. Iskalniki se med seboj

močno razlikujejo v pristopih — nekateri med njimi (kot je Google Search) aktivno sodelujejo pri sledenju, medtem ko drugi (kot Startpage) dejansko zagotavljajo anonimizacijo poizvedb. Rezultati tako potrjujejo potrebo po večplastnem pristopu, kjer se kombinirajo tehnične rešitve, premišljena izbira orodij in ozaveščenost uporabnika.

Na podlagi analiz naloga priporoča uporabo brskalnika Brave v kombinaciji z razširitvijo uBlock Origin, saj se je ta konfiguracija izkazala za najbolj učinkovito pri blokiranju različnih vrst sledenja, ne da bi pri tem bistveno zmanjšala uporabniško izkušnjo. Vzporedno se priporoča uporaba iskalnikov, ki ne shranjujejo osebnih podatkov uporabnika in ne posredujejo informacij tretjim osebam, pri čemer je izbor konkretnega iskalnika odvisen tudi od preferenc uporabnika glede uporabnosti in relevantnosti rezultatov.

Diplomska naloga prispeva k razumevanju sodobnih izzivov na področju sledenja brez piškotkov ter ponuja konkretne, izvedljive in utemeljene smernice za zaščito spletne zasebnosti. Delo je namenjeno tako strokovni javnosti, ki se ukvarja z informacijsko varnostjo, kot tudi običajnim uporabnikom, ki želijo prevzeti večji nadzor nad svojo spletno identiteto. Poleg tega služi kot osnova za nadaljnje raziskave na področju pasivnega sledenja in etičnih implikacij digitalnega nadzora ter spodbuja razvoj orodij in pristopov, ki krepijo pravico do zasebnosti v digitalni dobi.

Chapter 1

Introduction

The digital landscape has undergone a profound transformation over the past three decades, with HTTP cookies serving as the backbone of web personalization and user tracking. What began as a simple mechanism for websites to remember user preferences has evolved into a complex ecosystem of data collection that touches nearly every aspect of our online experience. As privacy awareness has grown and regulatory frameworks like GDPR [1] and CCPA [2] have emerged, the traditional cookie-based tracking model faces unprecedented challenges. This shift has catalyzed the development of alternative tracking methods that operate beyond the reach of conventional privacy controls, creating new concerns about user consent, data transparency, and regulatory compliance. Understanding this evolution from cookie-based to cookieless tracking is essential for comprehending the current state of digital privacy and the ongoing struggle between technological innovation and user protection.

1.1 What cookies are and how they function

HTTP cookies are small blocks of data that websites store on a user's device via the web browser [3]. They are designed to hold a modest amount of data specific to a particular client and website, allowing servers to deliver tailored

experiences to users. When a user visits a website, the server generates a cookie and sends it to the user's browser, which stores it locally. Upon subsequent visits to the same website, the browser includes the cookie in its requests, enabling the server to recall previous interactions. This mechanism facilitates essential web functionalities such as session management, personalization, and tracking. For instance, cookies can retain user preferences, maintain login sessions, and monitor browsing activities to enhance user experience [3]. However, it's important to note that while cookies play a crucial role in the modern web, they also raise privacy and security considerations, as they can be used to track user behavior across multiple sites [4].

1.2 Cookie-based tracking and privacy regulations

Alternative or cookieless tracking methods are relatively unknown to the public. As many people have by now seen a cookie banner, have had articles about cookies or cookie-based tracking show up on their social media feeds or have read articles about how cookies work. Cookies have been around for 30 years now and most of the people have at least heard of them or have a rough idea of what they do. Even though first-party cookies are for the most part essential for websites to work, third-party cookies are mostly used for tracking, data gathering purposes, and these third-party cookies are what the general public thinks of when they hear the word "cookies".

However, growing privacy concerns, stricter regulations like the GDPR [1] and CCPA [2], and advancements in browser privacy settings are making cookie-based tracking increasingly ineffective. Regulations now require explicit user consent for tracking cookies, leading to widespread adoption of cookie banners and opt-out mechanisms [5]. At the same time, major browsers like Safari and Firefox have implemented tracking prevention mechanisms such as the Enhanced Tracking Protection in the Firefox Browser [6] and Intelligent Tracking Prevention in the Safari Browser [7]. As a result, adver-

tisers and data-driven businesses face significant challenges in maintaining their tracking capabilities, pushing the industry toward alternative, cookieless tracking methods.

1.3 Alternatives to cookie-based tracking and motives

The industry's shift toward cookieless tracking methods is driven by growing privacy concerns and stricter data protection laws. Techniques such as fingerprinting and server-side tracking are increasingly employed to gather user data without relying on traditional cookies. However, these methods often bypass user consent mechanisms, raising significant ethical and legal issues [8].

This transition to cookieless tracking resembles a continuous "cat and mouse" game. As browsers enhance security and implement measures to resist tracking methods, companies and developers often find new workarounds to achieve their goals. For instance, the United Kingdom Information Commissioner's Office criticized Google for allowing advertisers to track users' digital "fingerprints," a practice that complicates efforts by privacy-conscious users to block online monitoring [9].

This shift has also led many companies to adopt lesser-known tracking methods that operate in legal gray areas where laws are less specific. While cookieless tracking reduces reliance on user-stored data, making it harder for users to block tracking entirely, it comes with significant drawbacks:

- It is often unclear how companies use the data collected through cookieless methods, making transparency an issue.
- Many of these methods may not comply with privacy regulations like GDPR [1] and CCPA [2], leaving users vulnerable.
- Laws designed to address traditional cookie-based tracking have inadvertently driven the adoption of less-regulated alternatives.

In this thesis, we explore and explain the most common cookieless tracking methods, describe how they work, outline their potential ethical and legal implications, and recommend what users can do to protect their privacy in these situations.

Chapter 2

Cookieless tracking methods

Cookieless tracking methods represent a significant shift in how user behavior is monitored online, especially as traditional cookie-based tracking faces increasing regulatory and technical restrictions. This chapter examines various cookieless tracking techniques, detailing their mechanisms, implications, and potential countermeasures. From sophisticated fingerprinting approaches that extract device characteristics, to indirect methods like bounce tracking, link decoration, and repurposed technologies such as ETags and TCP/IP fingerprinting, these practices enable persistent and often opaque surveillance of users across the web. By understanding the inner workings of these methods, we can better assess their privacy risks and the effectiveness of current mitigation strategies.

2.1 Fingerprinting

Fingerprinting is a method used to uniquely identify an entity—such as a device, user, or data set—by analyzing its distinct characteristics or patterns. For example, in device fingerprinting, information like browser type, screen resolution, and installed plugins is combined to create a unique identifier for tracking or fraud prevention. In data fingerprinting, algorithms like hashes or checksums generate a unique representation of data to verify integrity or

detect duplicates. Fingerprinting is widely used for online tracking, authentication, and data validation but raises privacy concerns due to its ability to persistently identify users without consent [10]. There are many different fingerprinting methods that are used to create a fingerprint and track users without them ever knowing.

2.1.1 Measuring fingerprinting

In browser and device fingerprinting, entropy measures the unpredictability or information content of an attribute's distribution: higher entropy means the attribute varies more across the population and therefore contributes more to uniquely identifying a browser or a device. Each characteristic of your device or browser—such as installed fonts, screen resolution, browser plugins, and system settings—contributes to your overall fingerprint. The extent of such contribution is defined by its prevalence: if you are using a very common browser, revealing that information may not be sufficient for identification. But if you use a very rare GPU - that may be enough. The formula for entropy in information theory is the following:

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x) \tag{2.1}$$

Where:

- $H(X)$ is the entropy measured in bits;
- $p(x)$ is the probability of a specific attribute value occurring;
- $\sum$ represents the sum over all possible values of the attribute; and
- $\log_2$ denotes the logarithm with base 2. Base 2 gives the unit of bits.

A feature where there are many different possible values (such as the list of fonts installed) can contribute many bits to the total, whereby something without much distinguishing power (such as which operating system you're using) may only add a few [11].

2.1.2 Device fingerprinting

A device fingerprint or machine fingerprint is information collected about the software and hardware of a remote computing device for the purpose of identification [12]. The information is usually assimilated into a brief identifier using a fingerprinting algorithm.

Applications that are locally installed on a device can gather a large amount of information about the software and the hardware of the device, often including unique identifiers such as the MAC address and serial numbers assigned to the machine hardware. Diverse and stable information can also be collected below the application layer, by leveraging the protocols that are used to transmit data. Even if they are not designed to gather and share identifying information, local applications may unwittingly expose identifying information to the remote parties with which they interact. The most prominent example is that of web browsers, which have been proven to expose diverse and stable information in such an amount to allow remote identification, this is called the browser fingerprint [12].

2.1.3 Browser fingerprinting

A browser fingerprint is information collected through interaction with the web browser of the device. Device fingerprints can be used to fully or partially identify individual devices even when persistent cookies (and zombie cookies, which is a tracking cookie that automatically regenerates itself after deletion using persistent storage mechanisms [13]) cannot be read or stored in the browser, when the client IP address is hidden, or when one switches to another browser on the same device [12].

Many different attributes can be used to create a fingerprint:

- **Browser version:** Browsers provide their name and version, together with some compatibility information, in the User-Agent request header. Being a statement freely given by the client, it should not be trusted when assessing its identity. Instead, the type and version of the browser

can be inferred from the observation of quirks in its behavior: for example, the order and number of HTTP header fields is unique to each browser family and, most importantly, each browser family and version differs in its implementation of HTML, CSS and JavaScript.

- **Browser extensions:** A combination of extensions or plugins unique to a browser can be added to a fingerprint directly. Extensions may also modify how any other browser attributes behave, adding additional complexity to the user's fingerprint.

- **Hardware properties:** User agents may provide system hardware information, such as phone model, in the HTTP header. Properties about the user's operating system, screen size, screen orientation, and display aspect ratio can also be retrieved using JavaScript to observe the result of CSS media queries.

- **Browsing history:** The fingerprinter could determine which sites the browser had previously visited within a list it provided, by querying the list using JavaScript with the CSS selector `:visited`. Typically, a list of 50 popular websites was sufficient to generate a unique user history profile, as well as provide information about the user's interests [14]. However, browsers have since then mitigated this risk [15].

- **Font metrics:** The letter bounding boxes differ between browsers based on anti-aliasing and font hinting configuration and can be measured by JavaScript.

- **Canvas fingerprinting:** One of a number of browser fingerprinting techniques for tracking online users that allow websites to identify and track visitors using the HTML5 canvas element instead of browser cookies or other similar means. This tracking technique is explained more in detail in subsection 2.1.4.

- **Hardware benchmarking:** Benchmark tests can be used to determine whether a user's CPU utilizes AES-NI or Intel Turbo Boost by

comparing the CPU time used to execute various computational or cryptographic algorithms. Specialized APIs can also be used, such as the Battery API, which constructs a short-term fingerprint based on the actual battery state of the device, or OscillatorNode, which can be invoked to produce a waveform based on user entropy. A device's hardware ID, which is a cryptographic hash function specified by the device's vendor, can also be queried to construct a fingerprint.

Possible mitigations:

- **Spoofed fingerprints:** Spoofing some of the information exposed to the fingerprinter (e.g. the user agent) may create a reduction in diversity, but the contrary could also be achieved if the spoofed information differentiates the user more than the real browser information. Spoofing the information differently at each website visit, for example by perturbing the sound and canvas rendering with a small amount of random noise, allows a reduction of stability.
- **Blocking scripts:** Blindly blocking client-side scripts served from third-party domains, and possibly also first-party domains (e.g. by disabling JavaScript or using NoScript) can sometimes render websites unusable. The preferred approach is to block only third-party domains that seem to track people, either because they are found on a blacklist of tracking domains (the approach followed by most ad blockers) or because the intention of tracking is inferred by past observations (the approach followed by Privacy Badger).

2.1.4 Canvas fingerprinting

The canvas element is part of HTML5 and allows for dynamic, scriptable rendering of 2D shapes and bitmap images. It is a low level, procedural model that updates a bitmap [16].

Canvas fingerprinting is one of several browser fingerprinting techniques for tracking online users that allow websites to identify and track visitors

using the HTML5 canvas element instead of browser cookies or other similar means [17]. The HTML5 Canvas is the single largest fingerprinting threat browsers face today [18].

It works by exploiting the HTML5 canvas element:

1. When a user visits a page the fingerprinting script first draws text with the font and size of its choice and adds background colors.

2. Next, the script calls Canvas API's ToDataURL method to get the canvas pixel data in dataURL format, which is basically a Base64 encoded representation of the binary pixel data.

3. Finally, the script takes the hash of the text-encoded pixel data, which serves as the fingerprint.

The technique is effectively fingerprinting the GPU. Variations in which the GPU or graphics driver is installed may cause fingerprint variation. While not sufficient to identify individual users by itself, this fingerprint could be combined with other entropy sources to provide a unique identifier.

There are multiple possible ways to mitigate and lower the risk of being tracked by canvas fingerprinting:

- Browser add-ons like Privacy Badger, DoNotTrackMe, or Adblock Plus manually enhanced with EasyPrivacy list are able to block third-party ad network trackers and can be configured to block canvas fingerprinting, provided that the tracker is served by a third party server (as opposed to being implemented by the visited website itself). Canvas Defender, a browser add-on, spoofs Canvas fingerprinting.

- The LibreWolf browser project includes technology to block access to the HTML5 canvas by default, only allowing it in specific instances green-lit by the user.

- Tor Browser notifies the user of canvas read attempts and provides the option to return blank image data to prevent fingerprinting. However,

Tor Browser is currently unable to distinguish between legitimate uses of the canvas element and fingerprinting efforts, so its warning cannot be taken as proof of a website's intent to identify and track its visitors.

2.2 Redirect/Bounce tracking

Bounce tracking is a technique used by web trackers. It involves inserting an intermediary link between you and the website you want to visit, allowing a tracker to know you and your interests, and thus use this data to sell more targeted ads. This technique is also sometimes known as "redirect tracking" [19].

2.2.1 How bounce tracking works

When you click a bounce-tracked link, your browser first goes to a tracking website, which quickly redirects you to the real website. This redirect happens so quickly that you likely won't even notice. You generally cannot tell if a link will take you to a bounce tracker, even if you hover over the link before clicking it and look at the URL your browser shows. That URL might be for the real website the link is meant to go to, but it may be swapped out for a bounce-tracked link just as you click on it. Bounce tracking works even in browsers that block third-party cookies, which more and more browsers are doing. While your browser is on the bounce tracker's site—however briefly—the tracker can set first-party cookies, which are much less likely to be blocked. This ensures that the bounce tracker can reliably identify your browser any time it passes through, and thus build a more complete profile of your browsing activity. The technique works by injecting additional sites between a website you're visiting and the website to which you intend to navigate, as displayed in Figure 2.1. These intermediate sites over time learn what sites you've visited, and so can perform the same kinds of tracking sites used to use third-party cookies for. During the brief "bounce" from the tracking website to the real website you intended to visit, the tracking

website has a chance to set first-party cookies, to see which URL you came from, and to see which URL you're going to. That information can be useful for advertising purposes, such as measuring the effectiveness of sponsored links [19].

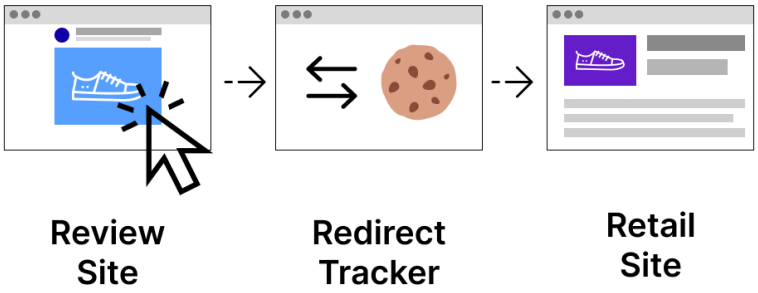

Figure 2.1: Illustration of bounce tracking mechanism

2.2.2 Bounce tracking mitigation

There are some browser extensions that can do debouncing. Debouncing refers to delaying or limiting the redirection or execution of tracking scripts to reduce or block the effectiveness of bounce tracking techniques. It aims to prevent users from being instantly redirected through tracking domains by adding a delay or requiring user interaction [20].

Some browsers try to mitigate the risks of extensions by limiting what they're allowed to do. But that creates a different problem: With limited capabilities, extensions might not be able to fully block bounce tracking. There's a fundamental tradeoff between safe extensions and powerful extensions. Some browsers, like Brave, can recognize when you're about to visit a bounce tracking website, and instead take you straight to the real URL, also known as debouncing. Debouncing that's built into the browser avoids this tradeoff entirely [20]. Or other browsers, like Firefox, periodically clear cookies and website data set by known trackers.

2.3 Link decorating

Link decorating is the process of appending additional information, such as tracking parameters or metadata, to a URL. This is often used to attribute traffic sources, enable analytics, or facilitate cross-domain tracking by carrying context (e.g., user IDs, campaign data) across links [21].

2.3.1 How link decorating works

A URL, such as `https://a.site.example/YYY/ZZZ/pixel.jpg?ISBN=XXX&UID=ABC123#xyz`,
consists of several parts [21]:

- **Base URL**: `https://a.site.example` (combined scheme and FQDN).
- **Resource Path**: `/YYY/ZZZ` identifies a server-side resource, with `pixel.jpg` being the resource file.
- **Query Parameters**: `ISBN=XXX&UID=ABC123` are key-value pairs after the `?` delimiter, separated by `&`.
- **Fragments**: `xyz` follows the `#` delimiter and may act similarly to query parameters.

2.3.2 Mitigation

To address the abuse of link decorations, several countermeasures have been developed, as outlined below [21]:

- **Filter List Approaches**: Browsers like Brave, Firefox, and Safari use manually curated lists to strip known tracking parameters from URLs. These lists, while effective for some cases, cannot scale with the rapid evolution of tracking techniques.

- **Machine Learning-Based Solutions**: PURL (pronounced purel-l) uses a supervised classifier to detect and sanitize tracking link decorations by analyzing webpage execution across multiple layers (HTML, JavaScript, network requests, and storage). It achieves 98.74% accuracy while minimizing website breakage compared to traditional methods.

- **Browser Extensions**: Tools such as uBlock Origin and AdGuard offer advanced parameter removal capabilities, using rules and regular expressions to sanitize URLs.

- **Emerging Standards**: Privacy-focused measures, like blocking third-party cookies, indirectly mitigate link decoration abuse by reducing trackers' reliance on decorated URLs.

These approaches highlight the importance of balancing functionality with privacy, as overly aggressive sanitization may break legitimate website features.

2.4 Other tracking methods

While we covered the most known, used and advanced methods of cookieless tracking, there are also other ways to utilize already existing software to achieve the purpose of tracking without cookies.

2.4.1 ETags

The ETag or entity tag is part of HTTP, the protocol for the World Wide Web. It is one of several mechanisms that HTTP provides for web cache validation, which allows a client to make conditional requests. This mechanism allows caches to be more efficient and saves bandwidth, as a web server does not need to send a full response if the content has not changed [22].

Although not used for tracking by default, ETags can be configured for both caching and tracking purposes. When a browser requests a resource, the server assigns a unique ETag to identify the resource version. The browser caches the resource with the ETag. On subsequent requests, the browser sends the ETag back in the `If-None-Match` header, allowing the server to recognize returning users by matching ETags, even if cookies are disabled. This is used to save bandwidth if content doesn't change on the website but also can enable persistent user tracking via browser cache without relying on traditional client-side storage [23].

2.4.2 TCP/IP stack fingerprinting

TCP/IP stack fingerprinting is the remote detection of characteristics of a TCP/IP stack implementation. The combination of parameters may then be used to infer the remote machine's operating system (aka, OS fingerprinting), or incorporated into a device fingerprint. Certain parameters within the TCP protocol definition are left up to the implementation. Different operating systems, and different versions of the same operating system, set different defaults for these values. By collecting and examining these values, one may differentiate among various operating systems and implementations of TCP/IP [24].

Chapter 3

Counteractive measures

Counteractive measures can vary from mild counteractive measures like downloading extensions to extreme like using the TOR Browser via the onion network, each with their pros and cons. Usually, proper functionality with good user experience and best possible privacy are on the opposite sides of the spectrum as seen in Figure 3.1. In this section we look at counteractive measures, what they mean, and which ones best apply to different needs and purposes.

Figure 3.1: Illustration of the privacy spectrum

3.1 Browsers

For the theoretical part of this thesis, we do not examine the underlying engines of these browsers, since even though the Google Chrome browser and Brave browser rely on the same engine - Chromium, they are both very different privacy-wise. We look at the final browser products to form conclusions, since that's what the majority of users are using in their everyday life.

By examining the following browsers with their percentage of market shares: Google Chrome (66.85%), Safari (17.19%), Microsoft Edge (5.21%), Mozilla Firefox (2.39%), we cover approximately 90% of the total browser usage share worldwide. Both mobile and desktop included [25].

In Subsection 3.1.6 we examine globally lesser used browsers and the benefits and drawbacks of using those browsers.

The protection a web browser can provide is important because more than 75% of tracking activities are made previously to the cookie consent banner or when user rejects all of them [8].

3.1.1 Google Chrome

Google began gradually improving its users' privacy protection starting in 2019 through a proposal called Privacy Sandbox. Privacy Sandbox is Google's new initiative to develop a set of open standards to improve internet privacy on the web. Currently, this initiative is still in development and looking for new ideas, but some measures have already been implemented. The main measures implemented in the latest versions of Chrome are the SameSite header, the Referrer-Policy header, and cache partitioning [26].

In addition to these measures, Google has introduced the **Privacy Budget**, a concept designed to limit fingerprinting techniques by restricting the amount of entropy available to websites. Moreover, the **User-Agent Reduction** initiative gradually reduces the information exposed in the User-Agent string, further mitigating passive fingerprinting risks. These efforts

align with broader industry trends, such as Mozilla's Total Cookie Protection and Apple's Intelligent Tracking Prevention (ITP), reinforcing privacy as a core web standard.

As of 2025, despite Google's Privacy Budget and Privacy Sandbox initiatives, Chrome still offers minimal built-in privacy protection by default. It does not obscure unique device attributes, making fingerprinting easy, nor does it remove tracking parameters from URLs. Chrome also fails to block major tracking scripts, pixels, and cookies, and does not prevent cross-session tracking by third-party trackers. As a result, users remain highly exposed to various tracking techniques unless they manually adjust settings or use extensions [26].

3.1.2 Safari

Apple Safari uses some privacy protection mechanisms to improve users' online privacy. These include Intelligent Tracking Prevention (ITP), which blocks cross-site tracking cookies and provides insight into tracking attempts via a privacy report. Fingerprinting prevention reduces the data accessible to websites for unique identification, while Enhanced ITP further limits cookie usage and storage. Cross-site tracking prevention and sandboxing provide isolation and security between websites. In addition, Apple's App Tracking Transparency (ATT) obtains user consent for tracking, and enhanced privacy settings allow for customization. Intelligent Tracking Prevention (ITP) is enabled in Safari by default, so there's no need to configure anything to get Safari's enhanced protection mechanism.

In recent updates, Safari has introduced the **Distraction Control** feature, allowing users to hide distracting elements such as cookie preference pop-ups while browsing. This feature enhances the browsing experience by reducing interruptions and clutter on web pages. Additionally, Safari's **Link Tracking Protection** detects and removes tracking parameters from link URLs, preventing third-party sites from tracking users' navigation behavior. This feature is automatically enabled in Mail, Messages, and when browsing

with Safari in Private Mode, further strengthening user privacy.

Despite these privacy-focused features, Safari performs poorly on many benchmarks, often lagging behind browsers like Brave and Firefox in areas such as tracker blocking, fingerprinting resistance, and network privacy protections [27]. This is primarily because Safari relies on Apple's proprietary privacy mechanisms rather than incorporating more aggressive third-party tracking protection lists or advanced anti-fingerprinting techniques.

3.1.3 Microsoft Edge

Microsoft Edge offers a suite of privacy features designed to enhance user online security. The browser includes built-in tracking prevention that blocks trackers and cookies, with customizable levels—Basic, Balanced, and Strict—to tailor the degree of blocking. **Microsoft Defender SmartScreen** protects against malicious websites and phishing attempts by checking visited URLs against a dynamic list of reported threats. **InPrivate Browsing** ensures that browsing history, cookies, and website data are not saved after the session ends. Additionally, Edge provides enhanced password protection, alerting users if their saved passwords are found in online breaches, and offers safeguards against downloading malware. Family safety settings allow for content filtering and activity reporting, while the Do Not Track setting enables users to request that websites do not track their browsing activity. The Tracking Prevention dashboard offers insights and customization options, and features like Secure DNS and the ability to delete browser data further bolster privacy [28].

Despite these robust features, Microsoft Edge has faced criticism regarding user privacy. Research indicates that Edge sends hardware identifiers, such as the device's unique hardware UUID, to Microsoft servers, creating persistent identifiers that can be challenging to change or delete [29]. This practice raises concerns about potential tracking across different installations and applications. Additionally, Edge has been observed importing data from other browsers without explicit user consent, leading to allegations of unau-

thorized data collection. The browser's integration with Windows 10 and 11 has also been scrutinized for redirecting certain web links to Edge, disregarding users' default browser preferences [29]. Furthermore, features like the "follow creators" function were criticized for privacy issues, as they involved sending users' browsing data to Microsoft. In response to these concerns, Microsoft has made adjustments, such as removing the "follow creators" feature and providing options to disable certain data collection practices [29].

Microsoft Edge, like Google Chrome, fails many privacy tests, particularly the open-source tests of web browser privacy [27], where it allows third-party trackers and lacks strong fingerprinting protection. Despite offering security features like SmartScreen and tracking prevention, its deep integration with Microsoft services makes it a weaker choice for privacy-conscious users compared to Firefox or Brave [27].

3.1.4 Mozilla Firefox

Mozilla Firefox employs several privacy protection mechanisms to strengthen users' online privacy. **Enhanced Tracking Protection (ETP)** blocks third-party trackers and cookies, preventing advertisers from monitoring users' browsing habits. **Total Cookie Protection** isolates cookies in separate containers, ensuring that each website stores cookies independently, thereby preventing cross-site tracking. **Fingerprinting Protection** minimizes the device information accessible to websites, reducing the ability to uniquely identify and track users based on their hardware and software configurations. Additionally, **DNS over HTTPS (DoH)** encrypts DNS traffic, preventing intermediaries from monitoring the websites users visit. The browser's **Privacy Protections Dashboard** provides insights and customization options, allowing users to manage their privacy settings effectively [6].

Despite these robust features, Firefox has faced criticism regarding certain privacy practices. In September 2024, the advocacy group NOYB filed a complaint alleging that Mozilla's **Privacy Preserving Attribution** feature tracks user behavior without explicit consent, potentially violating EU

privacy laws. Mozilla defended the feature, stating it aims to help websites understand ad performance without collecting individual user data [30]. Additionally, concerns have been raised about Firefox's telemetry data collection, which gathers information about browser performance and usage. While Mozilla asserts that this data is anonymized and used to improve user experience, some users are concerned about the potential for data to be traced back to individual users. Furthermore, certain privacy features in Firefox are not enabled by default, requiring users to manually adjust settings to achieve optimal privacy protection. This reliance on user initiative may result in less tech-savvy individuals not fully benefiting from the browser's privacy capabilities [31].

Additionally, Private Internet Access highlights concerns about telemetry data collection [32], though Mozilla states this is anonymized and used for improvements. Some privacy features, such as strict fingerprinting protection, require manual activation, making Firefox less private out of the box compared to browsers like Brave. Adding to this, Mozilla also made their privacy regulations even weaker in their newest update on their terms of use.

3.1.5 Brave

Brave positions itself as a privacy-centric browser, offering multiple layers of protection to enhance users' online privacy.

1. **The first layer: Brave Shields**, blocks trackers, cross-site cookies, fingerprinting attempts, and more. Users can view and manage these protections by clicking the Shields icon in the address bar on any page they visit.

2. **The second layer:** The second layer comprises **advanced privacy protections**, including reduced network server calls, partitioning, and blocking bounce tracking. These features are built directly into the browser to minimize data exposure.

3. **The third layer:** Involves Brave's **policies and practices**, emphasizing data minimization by not collecting user data in the first place. Brave aims to adhere to—and exceed—government data protections like GDPR [1] and CCPA [2], supporting and contributing to the online privacy community.

For extra security - Tor can be enabled in the Brave browser for even further enhanced private browsing [33].

Despite these robust features, Brave has faced criticism regarding certain aspects of its privacy practices. Some users have expressed concerns about the browser's integration of cryptocurrency services, such as the Brave Rewards program and the built-in crypto wallet, viewing them as unnecessary additions that could introduce potential security risks. However, these features are optional and can be disabled in the settings.

Additionally, Brave's use of the Chromium engine has raised questions about its reliance on Google's infrastructure, which some argue could pose privacy risks despite Brave's efforts to mitigate them through various customizations. Furthermore, while Brave's aggressive ad and tracker blocking enhances privacy, it has led to compatibility issues on certain websites, requiring users to adjust Shields settings to access content properly.

These concerns highlight the importance of user awareness and the need to customize settings according to individual privacy preferences.

3.1.6 Other browsers

Several alternative browsers prioritize privacy more aggressively than mainstream options. Tools like LibreWolf, Mullvad Browser, and Tor offer strong protections by removing telemetry and blocking tracking, though often at the cost of usability or compatibility. Each browser takes a different approach to privacy, highlighting the trade-offs users face when seeking greater anonymity online.

LibreWolf: A community-driven fork of Firefox, LibreWolf focuses on enhancing privacy and security by removing telemetry, disabling features

like Pocket, and blocking sponsored shortcuts. By default, it deletes cookies and history upon closing, though this can be adjusted [34]. However, users may experience compatibility issues with certain websites, and the absence of auto-updating necessitates manual updates to maintain security. It has one of the highest scores on privacy focused benchmarks [27].

Mullvad Browser: Developed collaboratively by Mullvad VPN and the Tor Project, this browser aims to minimize tracking and fingerprinting without routing traffic through the Tor network. It's designed to be used with a VPN, offering privacy benefits similar to the Tor Browser but with potentially improved performance [35]. Users should be aware that, like other privacy-focused browsers, some websites may not function optimally due to strict privacy protections. It has one of the highest scores on privacy focused, but coming with drawbacks of many sites not functioning properly [27].

Opera: A feature-rich browser that includes a built-in VPN, ad blocker, and tracker blocker to enhance user privacy. However, Opera has faced criticism due to its ownership by a Chinese consortium, raising concerns about data privacy and security. Additionally, being closed-source means its code isn't publicly auditable, which may be a drawback for users seeking transparency [36]. It also performs poorly on privacy focused benchmarks [27].

Ungoogled Chromium: This browser offers a Chromium experience stripped of Google services integration, aiming to improve privacy. While it removes dependencies on Google, it lacks built-in protections against fingerprinting and may require manual configuration to enhance privacy. The absence of automatic updates can also pose security risks if users do not update regularly [37]. The only difference on privacy focused benchmarks is that Ungoogled Chromium blocks known tracking cookies [27].

Vivaldi: Created by former Opera developers, Vivaldi is a highly customizable browser offering features like tab stacking, page tiling, and built-in notes. It includes an ad and tracker blocker, but as a closed-source application, its privacy claims cannot be independently verified [38]. Users who prioritize open-source software may view this as a limitation. It also performs

very poorly on privacy focused benchmarks [27].

Tor Browser: Developed by the Tor Project, Tor Browser is designed to provide users with enhanced online anonymity by routing internet traffic through the Tor network's encrypted, multi-layered relay system. This process conceals users' IP addresses and encrypts their data, making it challenging for third parties to monitor online activities. The browser includes security settings that allow users to disable certain web features that can compromise privacy, such as JavaScript and cookies. However, users should be aware that while Tor Browser offers significant privacy protections, it is not entirely foolproof. Engaging in activities like installing additional plugins or opening documents downloaded through Tor while online can potentially expose one's IP address. Additionally, some websites may not function properly due to the browser's strict security measures. Users are advised to follow best practices, such as avoiding the use of browser plugins and ensuring that sensitive activities are conducted over secure connections, to maintain anonymity [39]. Tor also performs excellently on privacy focused benchmarks [27].

3.1.7 Privacy focused benchmarks table comparison

According to the open-source tests of web browser privacy [27], Table 3.1 showcases the broad overview of how various browsers perform in terms of privacy and security:

Browser	State Part.	Fing. Resist.	IP Protect.	Tracker Block.	HTTPS
Brave (1.75)	Yes	Yes	Yes	Yes	Yes
Chrome (133.0)	Partial	No	No	No	Yes
Edge (133.0)	Partial	No	No	No	Yes
Firefox (135.0)	Yes	Partial	No	Yes	Yes
LibreWolf (135.0)	Yes	Yes	Yes	Yes	Yes
Mullvad (14.0)	Yes	Yes	Yes	Yes	Yes
Opera (117.0)	Partial	No	No	No	Yes
Safari (18.3)	Partial	No	No	No	Yes
Tor Browser (14.0)	Yes	Yes	Yes	Yes	Yes
Ungoogled (133.0)	Partial	No	No	No	Yes
Vivaldi (7.1)	Partial	No	No	No	Yes

Table 3.1: Open-source test results of web browser privacy for multiple known browsers and their respective versions.

- **State Partitioning:** Indicates whether the browser isolates data (like cookies and cache) between websites to prevent tracking.
- **Fingerprinting Resistance:** Refers to the browser's ability to prevent websites from uniquely identifying users based on device and browser characteristics.
- **IP Address Protection:** Denotes whether the browser has features to mask or protect the user's IP address, such as built-in VPNs or integration with Tor.
- **Tracker Blocking:** Shows if the browser blocks third-party trackers by default.
- **HTTPS Enforcement:** Indicates if the browser enforces secure connections (HTTPS) by default.

3.2 Extensions

Browser extensions have become essential tools for enhancing user privacy, particularly against emerging tracking techniques like fingerprinting and cookieless tracking. These extensions block or modify data exchanged between users and websites, preventing unauthorized tracking without compromising user experience. As cookieless tracking advances, browser extensions remain crucial in protecting user privacy. This section examines their role, effectiveness, and the innovations driving their development in the fight against modern tracking methods.

3.2.1 uBlock Origin

uBlock Origin is a CPU and memory-efficient wide-spectrum content blocker for Chromium and Firefox. It blocks ads, trackers, coin miners, popups, annoying anti-blockers, malware sites, and more by default using EasyList, EasyPrivacy, Peter Lowe's Blocklist, Online Malicious URL Blocklist, and uBO filter lists. There are many other lists available to block even more. Hosts files are also supported. uBlock Origin uses the EasyList filter syntax and extends the syntax to work with custom rules and filters [40].

While uBlock Origin is highly effective in blocking unwanted content, it relies on predefined filter lists, which may require regular updates to address new tracking methods. Additionally, some users may find the advanced features and customization options complex to navigate. There is also a potential for website breakage, necessitating manual adjustments to the extension's settings.

3.2.2 Privacy Badger

Privacy Badger is a browser extension that automatically learns to block invisible trackers. It sends the Global Privacy Control signal to opt you out of data sharing and selling, and the Do Not Track signal to tell companies not to track you. If trackers ignore these signals, Privacy Badger will learn

to block them. Besides automatic tracker blocking, Privacy Badger comes with privacy features like click-to-activate replacements for potentially useful trackers (video players, comments widgets, etc.), and link cleaning on Facebook and Google [41].

Although, turning on the Do Not Track signal does more harm than good in this case, since the Do Not Track HTTP header is deprecated. Meaning that by sending this signal, it does nothing since many companies did not respect it because of lack of legislation and as of 2025, Apple Safari and Mozilla Firefox have removed it all together. While the DNT HTTP header does not provide privacy, it also adds an extra bit for fingerprinting [42].

For what Privacy Badger is trying to do, uBlock Origin is a better option in this use case scenario. Although both can work together on the same browser well, with Privacy Badger included, we are providing extra information for fingerprinting.

3.2.3 Canvas Blocker

CanvasBlocker is an add-on that allows users to prevent websites from using certain JavaScript APIs to fingerprint them. Users can choose to block the APIs entirely on some or all websites (which may break some websites) or just block or fake their fingerprinting-susceptible readout API.

While effective at preventing canvas fingerprinting, CanvasBlocker may cause compatibility issues with websites that rely on the blocked APIs for legitimate functionality. Users may need to whitelist certain sites or adjust settings to balance privacy and usability [43].

3.2.4 ClearURLs

ClearURLs is an add-on based on the new WebExtensions technology and is optimized for Firefox and some Chrome based browsers. This extension will automatically remove tracking elements from URLs to help protect your privacy when browsing the Internet, which is regularly updated.

However, some users have reported that ClearURLs can cause issues with certain websites, such as breaking functionality or preventing access. Additionally, while it effectively removes known tracking elements, it may not catch all tracking methods, especially as new techniques emerge. Users should be aware of these limitations and consider using ClearURLs in conjunction with other privacy tools for comprehensive protection [44].

3.2.5 Ghostery

Ghostery enables its users to detect and control JavaScript tags and trackers to remove JavaScript bugs and beacons that are embedded in many web pages which allow for the collection of a user's browsing habits via HTTP cookies, as well as participating in more sophisticated forms of tracking such as canvas fingerprinting [45].

Ghostery offers extensive functionality, for example:

- Block all ads on websites, including YouTube and Facebook.
- Stop trackers from collecting your personal data.
- Automatically remove intrusive cookie pop-ups and express dissent to online tracking.
- Get detailed tracker information on any website you visit, including the number of trackers, their type, and the company operating them.
- Preview tracker information on search engine result pages to make informed choices.
- Inspect the largest database of trackers, updated fast and reliably for all users.

In a study done by Carnegie, Mellon University, where they compared three different privacy focused browser extensions (Ghostery, DNTMe, Disconnect and a placebo tool), they found that Ghostery detected the most trackers out of the four (232 unique trackers, 3.81 average per website) [46].

And, in another study done by The University of Lahore [47], they discovered that by default, Ghostery does not provide protection; if appropriately set enabled and configured, it offers the best protection from third-party trackers (93.99% of third-party trackers blocked), is faster than the baseline, with page loading 32.37% acceleration and provides the best trade-off among web page quality and protection. This study concludes that, out of all the extensions reviewed (Ghostery, Privacy Badger, Disconnect), when properly configured, Ghostery shows the best results on all benchmarks - protection, least bandwidth usage and fastest page loading acceleration.

The main drawback here with Ghostery is that users have to create an account and configure Ghostery themselves.

3.3 Search engines

Search engines are the primary means of accessing information on the internet, but they often track users for personalized results and targeted advertising. While cookies have traditionally been used for tracking, search engines are increasingly adopting cookieless tracking methods, such as fingerprinting and browser history analysis, to gather data without relying on cookies. This section explores how search engines contribute to cookieless tracking and the privacy implications of these practices [48].

3.3.1 Google Search

Google Search collects and utilizes extensive user data to enhance its services and deliver personalized experiences. This data collection includes search queries, location information, and browsing history, which are used to refine search results and target advertising. While these practices aim to improve user experience, they have raised significant privacy concerns. An entire Wikipedia page is dedicated to the privacy concerns of Google Search [49].

Critics argue that Google's data retention policies and the integration of user information across various services can lead to potential overreach and

misuse of personal data. For instance, in 2012, Google consolidated its privacy policies, allowing for the sharing of user data across multiple platforms such as YouTube, Gmail, and Maps. This move was widely criticized for creating an environment that discourages internet innovation by making users more fearful online [49].

Furthermore, Google's advertising practices have been scrutinized for privacy implications. The company has been known to place long-term cookies on users' devices to store preferences, enabling the tracking of search terms and retention of data for extended periods [50].

In response to these concerns, Google has implemented tools to give users more control over their data. Features like the Privacy Checkup and My Activity allow users to manage, export, and delete their information. However, the effectiveness of these measures is often debated, as the default settings still favor data collection, and navigating these tools can be complex for the average user.

3.3.2 DuckDuckGo

DuckDuckGo is a search engine that prioritizes user privacy by not tracking or storing personal information. Unlike traditional search engines, it does not save your search history, IP address, or any other personal data. This approach ensures that your searches remain anonymous and are not used to create user profiles or target advertisements. DuckDuckGo offers a privacy-centric alternative to traditional search engines by not collecting or sharing personal data, providing tools to enhance online privacy, and offering additional services for comprehensive protection [51].

In addition to its search engine, DuckDuckGo offers a privacy-focused web browser that includes features such as tracker blocking, automatic HTTPS upgrading, and a "Fire Button" that allows users to clear all tabs and data with one tap. These tools are designed to provide a safer and more private browsing experience [51].

While DuckDuckGo provides robust privacy protections, it's important

to note that it cannot prevent tracking by external websites or internet service providers once you navigate away from its platform. To achieve more comprehensive privacy, combining the use of DuckDuckGo with additional tools like a reputable VPN is recommended.

3.3.3 Bing

Both Google Chrome and Bing prioritize data collection for personalization and advertising rather than privacy. They track searches, browsing activity, and location data, integrating this information across their respective ecosystems (Google services for Chrome, Microsoft services for Bing). While both offer privacy controls, they require manual configuration and do not default to privacy-friendly settings. Chrome collects more extensive data, including full web activity, whereas Bing is limited to searches and Microsoft services [52].

Bing, Microsoft's search engine, collects user data such as search queries, IP addresses, and location information to enhance its services and deliver personalized experiences. This data collection enables Bing to refine search results and target advertising. Microsoft asserts that it does not sell or rent personal information to third parties [52].

However, Bing has faced criticism regarding its privacy practices. In December 2022, France's data privacy watchdog, the Commission Nationale de l'Informatique et des Libertés (CNIL), fined Microsoft €60 million for imposing advertising cookies on users without obtaining their explicit consent. The CNIL highlighted that Bing had not set up a simple system allowing users to refuse cookies as easily as accepting them, thereby violating data protection laws [53].

Additionally, concerns have been raised about Bing's data collection methods. Microsoft's terms state that Bing can collect and process data in various forms, including text that has been inked or typed, voice data, and images. When users are signed in, some products may display a user's name or username and their profile photo as part of their use of Microsoft products,

including in communications, social interactions, and public posts. This extensive data collection has led to privacy concerns among users [54].

Furthermore, Bing has been criticized for its collaboration with the National Security Agency (NSA) on internet surveillance. Leaked NSA documents revealed that Microsoft was the first company to participate in the PRISM surveillance program, which authorizes the government to secretly access data of non-US citizens hosted by American companies without a warrant. Microsoft has denied participation in such a program [55].

In response to these concerns, Microsoft has implemented tools to give users more control over their data. Features like the Microsoft privacy dashboard allow users to manage, export, and delete their information. However, the effectiveness of these measures is often debated, as the default settings still favor data collection, and navigating these tools can be complex for the average user [54].

In summary, while Bing offers personalized search experiences by collecting user data, it has faced significant criticism and legal challenges regarding its privacy practices. Users concerned about their privacy may need to explore alternative search engines that prioritize data protection and offer more transparent data handling practices.

3.3.4 Brave Search

Brave Search is a privacy-focused search engine developed by Brave Software, Inc., designed to prioritize user privacy and security. Unlike traditional search engines, Brave Search does not collect personal information such as IP addresses or search history, ensuring that user searches remain anonymous. This approach prevents the creation of user profiles and eliminates the possibility of targeted advertisements based on search behavior [56].

A distinguishing feature of Brave Search is its independence from major tech companies. It operates on its own independent index, reducing reliance on third-party data sources and minimizing potential privacy risks associated with external data providers. This independence allows Brave Search to

maintain greater control over its data collection practices and uphold its commitment to user privacy [57].

In addition to its search engine, Brave offers a web browser that integrates with Brave Search, providing a comprehensive solution for users seeking a secure and private browsing experience. The Brave browser includes features such as tracker blocking, automatic HTTPS upgrading, and a built-in ad blocker, further enhancing user privacy and security [33].

However, some users have raised concerns regarding Brave's privacy practices. Discussions in the Brave Community have highlighted issues related to integrations with multiple companies, AI features, and sync settings, suggesting that these tools, while intended to improve user experience, may introduce potential risks for personal data [58].

Additionally, critiques have been made about Brave's approach to privacy, with some arguing that the company uses privacy as a marketing tool while potentially engaging in tracking practices [59].

In summary, Brave Search, in conjunction with the Brave browser, offers a robust solution for users prioritizing privacy and data protection. Its commitment to not collecting personal information and its independence from major tech companies make it a compelling choice for those seeking to maintain anonymity online.

3.3.5 Startpage

Startpage is a privacy-focused search engine that emphasizes user anonymity and data protection. It provides search results from Google without tracking or storing personal information, ensuring that your search history remains private. Startpage does not record your IP address or search queries, maintaining your anonymity. The 'Anonymous View' feature allows you to visit websites from your search results through a proxy, preventing the sites from tracking your IP address and protecting your browsing activity. Startpage does not use cookies to track your activity or create user profiles, ensuring that your searches are not used for targeted advertising. All searches and

browsing are conducted over secure HTTPS connections, safeguarding your data from potential interception [60].

However, in October 2019, Privacy One Group, owned by the adtech company System1, acquired a majority stake in Startpage. An initial lack of transparency surrounding the deal caused some concern among privacy researchers, leading to its removal from the PrivacyTools review website. After responding to questions from PrivacyTools team members, Startpage was able to clarify that the acquisition would not impact their privacy-focused mission, and its recommendation was ultimately restored. According to the company, its "founders may unilaterally reject any potential technical change that could negatively affect user privacy". By maintaining its headquarters and operations in the Netherlands, Startpage continues to be protected by Dutch and European Union (EU) privacy laws [61].

In summary, while Startpage offers robust privacy features, users should remain informed about its ownership structure and consider how it aligns with their personal privacy preferences.

3.3.6 Other search engines

Other search engines also vary widely in how they handle user data. While many—such as Yandex, Naver, and Baidu—collect search queries and location information to personalize results and target ads, others like Qwant and Mojeek prioritize user privacy by minimizing or avoiding data collection. These regional differences reflect varying attitudes toward privacy and regulation, influencing how search services operate across different countries.

Yandex (Russia): Yandex collects user data, including search queries and location information, to personalize search results and advertisements [62].

Naver (South Korea): Naver gathers user data such as search history and location to enhance search functionalities and deliver targeted advertising [63].

Baidu (China): Baidu collects user data, including search queries and location information, to personalize search results and advertisements [64].

Yahoo! Japan: Yahoo! Japan collects user data, including search queries and location information, to personalize search results and advertisements [65].

Yahoo! Taiwan: Yahoo! Taiwan collects user data, including search queries and location information, to personalize search results and advertisements [66].

Qwant (France): Qwant emphasizes user privacy by not tracking search history or personal information, aiming to provide an unbiased search experience [67].

Seznam (Czech Republic): Seznam collects user data, including search queries and location information, to personalize search results and advertisements [68].

Mojeek (United Kingdom): Mojeek is a UK-based search engine known for its focus on privacy and independence from other major search indexes [69].

3.4 Theoretical conclusion

Based on the theoretical analysis and comparative evaluation of privacy tools presented in this research, this chapter synthesizes the findings to provide clear, actionable recommendations for users seeking to enhance their online privacy. The conclusions drawn here balance the competing demands of privacy protection, usability, and functionality, recognizing that different users have varying technical expertise and privacy requirements. Through systematic evaluation of browsers, extensions, and search engines, this analysis identifies optimal configurations that maximize privacy benefits while maintaining practical usability for everyday web browsing. The recommendations presented offer both comprehensive solutions for privacy-conscious users and accessible options for those with limited technical knowledge, ultimately providing a framework for making informed decisions about digital privacy tools.

3.4.1 Browsers

Out of the most popular browsers, only Firefox, Brave and Safari are reasonable options for privacy. The problems here are that Firefox doesn't offer the privacy features out of the box, which means that non-tech-savvy individuals will have harder time configuring the browser to get the best privacy features out of it. Firefox is available on Windows, macOS and Linux, while Safari is only available on macOS. While this does give it an advantage since it's only within the Apple Ecosystem using the privacy features we mentioned in subsection 3.1.2 this thesis, it has a harder accessibility and not everyone will have access to this browser, as Windows has a global market share of 70% [70].

Out of the lesser known browsers, Mullvad, Librewolf and Tor Browser perform excellently on privacy based benchmarks, and are widely known as secure within the tech community of this niche, they are less known and usually require a higher tech knowledge of using these browsers, as because of their intense privacy settings, many sites appear broken.

Out of all of the browsers that we covered, one particularly mixes both privacy and functionality the best, has good privacy settings out of the box so that non-techsavvy individuals can use it and also offers even further privacy security for using it in private browsing - that browser is **the Brave browser**. It's good for regular users as the Brave browser is based on the Chromium engine. While this is deemed as it's main setback for privacy, it offers the regular user to get the same experience as the most popular browser, Google Chrome [25], while being secure online. It is also good for advanced users as they can also use Tor in private browsing for further and extended privacy on the internet.

3.4.2 Extensions

To begin with, skipping Privacy Badger is a good idea, as it offers less functionality than uBlock Origin while adding the Do Not Track HTTP header,

which is deprecated and can decrease fingerprinting entropy.

CanvasBlocker takes a randomized approach by generating a different canvas fingerprint every time you visit a page, which, although seemingly counterintuitive, can actually make you more identifiable. If fingerprinting is a major concern, a more robust solution would be using the Brave browser.

The behavior provided by ClearURLs is already integrated into recent versions of uBlock Origin, so installing ClearURLs alongside uBlock Origin is unnecessary and can increase your fingerprintability.

This results in with a comparison between Ghostery and uBlock Origin. Both are open-source and block trackers, but they differ in focus and ease of use. Ghostery emphasizes tracker detection and user awareness, offering detailed insights and anti-fingerprinting features, but it requires an account and manual configuration to enable privacy-focused settings. uBlock Origin, on the other hand, is a straightforward solution that works effectively right out of the box, with robust script blocking, extensive filter lists, and low resource usage. In conclusion, while Ghostery can be useful for understanding trackers, uBlock Origin is the more efficient, privacy-preserving choice for most users, offering better protection against fingerprinting with minimal setup.

3.4.3 Search engines

Privacy-conscious users should approach mainstream search engines like **Google Search** and **Bing** with caution, as both heavily rely on extensive data collection to fuel personalization and advertising. Despite offering privacy controls, their default configurations favor data retention and behavioral tracking. These practices often involve cookieless tracking techniques such as fingerprinting, IP logging, and search query profiling, which pose significant privacy concerns.

Qwant and **Mojeek** offer fully privacy-focused experiences. **Qwant** is based in Europe and emphasizes neutrality, while **Mojeek** is one of the few search engines with its own independent index. However, both can deliver

less accurate or exhaustive search results compared to mainstream engines.

DuckDuckGo is a well-rounded option, combining privacy-friendly policies with generally reliable search results. It is widely trusted within the privacy community. However, it still relies partially on Bing's search index, which means it is not fully independent.

Startpage is ideal for users who prefer Google's search results without the associated tracking. It acts as a privacy wrapper around Google, but concerns persist due to its acquisition by adtech company **System1**. Nevertheless, Startpage continues to operate under strict European privacy laws, offering robust anonymity.

Brave Search stands out for its independence, operating on its own search index rather than relying on Google or Bing. This enhances privacy but can result in less comprehensive search coverage, especially for niche queries.

With search engines, the answer is slightly more complex as seen in Table 3.2; there is not one concrete solution, but rather multiple excellent solutions, tailored to what the user's priorities are.

User Priority	**Search Engine**
Maximum privacy above all else	Mojeek and Qwant
Good privacy, independent search result and generally trusted	Brave Search
Good privacy, generally trusted and used by the public with solid search results	DuckDuckGo
Good privacy, slightly less trusted but offering excellent search results similar to Google Search	Startpage

Table 3.2: Search engine recommendations based on user priorities

These search engines will cover most if not all of the needs of the general user and users can easily switch between search engines based on what they need at the moment, regardless of their browser.

3.4.4 Recommended privacy solution

Based on the research conducted, the most effective solution for browsing the web privately—suitable for both everyday users and those seeking higher privacy—combines high accuracy, a good user experience, and robust privacy and security. This solution is to use the **Brave browser** with **uBlock Origin** installed, alongside one of the privacy-focused search engines discussed in the previous subsection 3.4.3, depending on the user's specific priorities.

Additionally, **Brave** offers an intuitive feature that allows users to easily switch between search engines by simply typing `:` followed by a shortcut (e.g., `:br` for Brave Search, `:g` for Google, `:q` for Qwant, `:d` for DuckDuckGo, etc.) directly in the search bar, further enhancing convenience and flexibility.

Chapter 4

Automated analysis tool for web tracking without cookies

The practical part of this bachelor's thesis focuses on evaluating the effectiveness of various privacy measures implemented by different web browsers, search engines, and extensions. Through automated testing, this project assesses how well these measures protect users from common tracking techniques such as fingerprinting, bounce tracking, and link decorating. The automated nature of these tests ensures systematic and repeatable execution, yielding consistent results that can be reliably analyzed and compared across different privacy protection configurations.

The goal of the project is to develop a tool that allows for running these tests on different combinations of browsers, search engines, and extensions, enabling the analysis of privacy protection effectiveness across various web environments. Using the browser emulation tool - Playwright, different tests are executed on real websites, and the results are analyzed to better understand how different privacy measures affect user tracking.

The testing involves various browser engines, such as Chromium, Firefox, and WebKit, along with different search engines and privacy-protecting extensions. To facilitate testing, the project provides a CLI script that allows users to specify various combinations of browsers, search engines, and exten-

sions they wish to test. This enables end users to easily evaluate the privacy protections offered by their preferred combination of tools and configurations, rather than relying on predefined test scenarios.

This part of the thesis focuses on implementing the testing environment, running the tests, and analyzing the results to contribute to a better understanding of current privacy protection methods on the web.

4.1 Open source repository

This project is open source and can be found on GitHub at `https://github.com/jaylzv/wtwc-bsc-thesis-fri-2025`. The repository contains the code for the testing tool, as well as instructions on how to set up and run the tests. The project is licensed under the GPL-3.0 license.

The code structure in the project is created with scalability in mind, so that every reader and possible future contributor can easily implement new methods, variables or end-to-end testing scenarios.

By making this testing tool open source, the goal is to benefit researchers, developers and privacy advocates who use it, modify it, or build upon it for their own privacy research and development needs.

4.1.1 Used software

The project is built using the following software:

- **Node.js:** Node.js is a cross-platform, open-source JavaScript runtime environment that can run on Windows, Linux, Unix, macOS, and more. Node.js runs on the V8 JavaScript engine and executes JavaScript code outside a web browser [71].

- **Node Package Manager**: Node package manager (npm) is a package manager for the JavaScript programming language maintained by npm, Inc., a subsidiary of GitHub. npm is the default package manager

for the JavaScript runtime environment Node.js, and is included as a recommended feature in the Node.js installer [72].

- **Playwright:** Playwright is an open-source automation library for browser testing and web scraping, which has since become popular among programmers and web developers. Playwright provides the ability to automate browser tasks in Chromium, Firefox and WebKit with a single API. This allows developers to create reliable end-to-end tests that are capable of running in non-headless mode, as well as in headless mode for automation [73].

- **TypeScript:** TypeScript (abbreviated as TS) is a free and open-source high-level programming language developed by Microsoft that adds static typing with optional type annotations to JavaScript. It is designed for the development of large applications and transpiles to JavaScript [74].

- **Git:** A distributed version control system that tracks versions of files. It is often used to control source code by programmers who are developing software collaboratively [75].

- **Docker:** Docker is a set of platform as a service products that use OS-level virtualization to deliver software in packages called containers. The service has both free and premium tiers. The software that hosts the containers is called Docker Engine [76].

4.1.2 Limitations

The testing environment faces several key limitations that should be considered when interpreting results:

- **Browser Engine Limitations:** Playwright only provides access to the core open source browser engines (Chromium, Firefox, WebKit) rather than complete closed source browser implementations. This means:

  - Tests don't reflect the exact behavior of browsers like Google Chrome or Microsoft Edge.
  - Browser-specific features and modifications are not captured.
  - Users with specific browser configurations may need to conduct manual testing.

- **DOM Dependency:** The testing framework relies heavily on the DOM structure to:
  - Navigate websites.
  - Extract data.
  - Interact with elements.

  This dependency means tests may break if websites change their DOM structure.

- **Extension Support:** Not all privacy extensions are compatible with automated testing, and some may behave differently in a test environment compared to real-world usage.

- **Dynamic Content:** Modern websites often use dynamic content loading and complex JavaScript interactions that can be challenging to test consistently.

- **Network Conditions:** Tests are run under ideal network conditions, which may not reflect real-world scenarios with varying connection speeds and reliability.

To address these limitations, the project:

- Provides detailed documentation of test procedures.
- Uses explicit test methods that are easy to reproduce manually.
- Includes checkpoint scripts to record test conditions and results.
- Maintains a modular code structure for easy updates as websites change.

4.1.3 CLI Scripts

The project contains the main CLI script, which is used to run the tests and provides various arguments to adjust what to include in the browser emulation, such as which search engines to visit through which browser with which extensions.

The script is designed to be run from the command line and accepts various arguments to customize the testing process. The arguments are:

- `-a`, `--all`: Run all tests.
- `-t`, `--tests`: Specify tests to run (comma-separated).
  - *Available tests*: `link_decorating`, `fingerprinting`, `bounce_tracking`.
- `-b`, `--browsers`: Specify browsers to use (comma-separated).
  - *Available browsers*: `chromium`, `firefox`, `webkit`.
- `-s`, `--search-engines`: Specify search engines to use (comma-separated).
  - *Available search engines*: `google`, `bing`, `startpage`, `duckduckgo`, `yahoo`, `search.brave`, `mojeek`, `qwant`.
- `-e`, `--extensions`: Specify extensions to use (comma-separated).
  - *Available extensions*: `ublockorigin`, `privacybadger`, `ghostery`, `canvasblocker`, `clearurls` or `empty` if you wish to test without extensions.
- `-h`, `--headed`: Enable headed mode. (Enables the graphical user interface. Worse performance).

As well as a separate script for writing down the results, so that we know what the state of the technology was at the time of running the tests.

The main script in the `main.ts` file, uses the following code approach as in Listing 4.1 to parse arguments and check which tests should run:

```
const shouldRunTest = (test: TestType, args: ArgumentsType): boolean => {
  return args.tests.includes(test);
};

const main = async (): Promise<void> => {
  const args = parseArgs();

  shouldRunTest(TestEnum.FINGERPRINTING, args) &&
    (await testAllScenarios(TestEnum.FINGERPRINTING, args));
  shouldRunTest(TestEnum.BOUNCE_TRACKING, args) &&
    (await testAllScenarios(TestEnum.BOUNCE_TRACKING, args));
  shouldRunTest(TestEnum.LINK_DECORATING, args) &&
    (await testAllScenarios(TestEnum.LINK_DECORATING, args));
};
```

Listing 4.1: Main function in repository for handling arguments and running automated tests.

And the function for parsing the arguments is as in the following Listing 4.2:

```
const parseArgs = (): ArgumentsType => {
  const scriptArgs = process.argv.slice(2);

  const args: ArgumentsType = {
    tests: Object.values(TestEnum),
    browsers: BROWSERS,
    searchEngines: SEARCH_ENGINES,
    extensions: EXTENSIONS,
    headed: false,
    websites: FINGERPRINTING_WEBSITE_CLI_ARGS,
  };

  if (scriptArgs.length === 0) {
    logCLIHelp();
    process.exit(0);
  } else if (!scriptArgs.includes("-a") && !scriptArgs.includes("--all"))
      {
```

```
for (const cliArg of POSSIBLE_CLI_ARGS) {
  if (scriptArgs.includes(cliArg)) {
    const index = scriptArgs.indexOf(cliArg);
    const value = scriptArgs[index + 1];

    if (
      value === undefined &&
      cliArg !== "-d" &&
      cliArg !== "-h" &&
      cliArg !== "--headed"
    ) {
      console.error(`No value provided for ${cliArg}.`);
      process.exit(1);
    }

    switch (cliArg) {
      case "-t":
      case "--tests":
        args.tests = value.split(",") as TestType[];
        break;
      case "-b":
      case "--browsers":
        args.browsers = value.split(",") as BrowsersType[];
        break;
      case "-s":
      case "--search-engines":
        args.searchEngines = value.split(",") as SearchEngineType[];
        break;
      case "-e":
      case "--extensions":
        args.extensions = value.includes("empty") ? [] : value.split(",
            ");
        break;
      case "-h":
      case "--headed":
        args.headed = true;
        break;
      case "-w":
```

```
          case "--websites":
            args.websites = value.split(",");
            break;
          default:
            console.error(`Invalid argument: ${cliArg}`);
            process.exit(1);
        }
      }
    }
  }

  return args;
};
```

Listing 4.2: Function for parsing CLI arguments to determine which variables should be tested.

4.2 Practical results

Inside the repository under the `checkpoints/` directory, there is an already existing, large text file with the results of the tests run at that given time. In this text file, the script was run with all possible combinations of browsers, search engines and extensions that the repository provides.

The tests are run for every possible combination of browser engines, search engines and extensions.

To calculate the total number of possible combinations, we must consider that extensions only work with the Chromium browser. For Chromium, we have 1 browser $\times$ 8 search engines $\times$ 2^5 extension combinations (each of the 5 extensions can be included or excluded) $= 1 \times 8 \times 32 = 256$ combinations. For the remaining browsers (Firefox and WebKit), we have 2 browsers $\times$ 8 search engines $\times$ 1 extension option (no extensions available) $= 2 \times 8 \times 1 = 16$ combinations. Therefore, the total number of possible combinations is $256 + 16 = 272$.

Firefox and webkit do not support the same extensions as Chromium, so the combinations of browsers and extensions are not the same.

The following browser engines are used:

- Chromium
- Firefox
- WebKit

The following search engines are used:

- Google
- Bing
- Startpage
- DuckDuckGo
- Yahoo
- Search Brave
- Mojeek
- Qwant

The following extensions are used:

- uBlock Origin
- Privacy Badger
- Ghostery
- Canvas Blocker
- ClearURLs

And the result of all possible combinations of browsers, search engines and extensions is in the `checkpoints/` directory.

The results are divided into three categories: **fingerprinting**, **bounce tracking** and **link decorating**.

Readers are encouraged to visit the websites, or use the link decorating methods mentioned and check out the results for themselves and their browser and compare.

4.2.1 Fingerprinting results

Fingerprinting tests are run on the BrowserScan platform [77], which is a website that tests the fingerprinting capabilities of the browser.

The main function in code which is the same as in the repository is as in the following Listing 4.3:

```
const testFingerprinting = async (
  testOptions: TestOptionsType
): Promise<void> => {
  console.log("\nTesting␣fingerprinting...");

  const { page, currentArgs } = testOptions;
  const { searchEngine } = currentArgs;
  let fingerprintData: Map<string, FingerprintDataType> = new Map();

  for (const site of FINGERPRINTING_SITES_URLS) {
    if (
      currentArgs.websites.some((website) => site.includes(website)) ||
      currentArgs.websites.some((_) => site.includes("all"))
    ) {
      const fingerprint: FingerprintDataType = await
          retrieveFingerprintData(
        page,
        searchEngine,
        site
      );
      fingerprintData.set(site, fingerprint);
```

```
    }
  }

  displayFingerprintData(fingerprintData, currentArgs);
};
```

Listing 4.3: Main function for testing fingerprinting.

The fingerprinting results from scraping the BrowserScan platform [77] for the three browser engines for each possible combination of search engines and extensions are as in Table 4.1:

Browser Engine	**Search Engine**	**Extension**	**Fingerprinting Results**
Chromium	All search engines	All extensions	Browserscan recognized everything correctly
Firefox	All search engines	All extensions	Browserscan recognized everything correctly
WebKit	All search engines	All extensions	Browserscan incorrectly identified the OS as MacOS while tests are running on Linux. All other attributes were correctly identified.

Table 4.1: Fingerprinting test results

The results show that the BrowserScan platform [77] accurately identified browser characteristics in all cases except for the WebKit engine, which incorrectly reported MacOS as the operating system when tests are conducted on Linux. This indicates that browser engines provide minimal fingerprinting protection, even with privacy extensions enabled.

4.2.2 Bounce tracking results

Bounce tracking tests are run on the `https://bounce-tracking-demo.glitch.me/` website, which is a website that uses the bounce tracking method to place trackers as cookies or trackers on the browser's local storage.

The main function in code which is the same as in the repository is as in the following Listing 4.4:

```
const testBounceTracking = async (
  testOptions: TestOptionsType
): Promise<void> => {
  console.log("Testing␣bounce␣tracking...");

  const { page, currentArgs } = testOptions;
  const { searchEngine } = currentArgs;
  const mainWebsiteURL = "https://bounce-tracking-demo.glitch.me/";

  console.log('Navigating␣to␣"Bounce␣Tracking"␣demo␣website...');
  await navigateToWebsiteThroughSearchEngine(
    page,
    searchEngine,
    mainWebsiteURL
  );
  await waitForBounceTrackingPageToLoad(page);

  await resetStorageAndCookies(page);

  const { initialCookies, initialLocalStorage } = await
      initializeInitialValues(
    page
  );

  console.log("Performing␣redirects...");
  console.log("Redirecting␣to␣server␣with␣cookies...");
  await redirect(page, `//a[@href="${AnchorHrefEnum.SERVER_WITH_COOKIES
      }"]`);
  console.log("Redirecting␣to␣server␣with␣local␣storage...");
  await redirect(
```

```
    page,
    `//a[@href="${AnchorHrefEnum.CLIENT_WITH_LOCAL_STORAGE}"]`
  );

  const finalStorage = await page.context().storageState();
  const finalCookies: CookiesType = finalStorage.cookies;
  const finalLocalStorage: LocalStorageType =
    finalStorage.origins[0].localStorage;

  const results: DisplayResultsType = {
    currentArgs,
    initialCookies,
    initialLocalStorage,
    finalCookies,
    finalLocalStorage,
  };

  displayResults(results);
};
```

Listing 4.4: Main function for bounce tracking.

The bounce tracking results from `https://bounce-tracking-demo.glitch.me/` for the three browser engines for each possible combination of search engines and extensions are as in Table 4.2:

Browser Engine	Search Engine	Extension	Bounce Tracking Results
Chromium	All search engines	All extensions	Cookie trackers placed successfully, while local storage remained the same
Firefox	All search engines	All extensions	Cookie trackers placed successfully, while local storage remained the same
WebKit	All search engines	All extensions	Cookie trackers placed successfully, while local storage remained the same

Table 4.2: Bounce tracking test results

Even though local storage and cookies were reset for all browsers before each test - cookie trackers were placed successfully while local storage remained exactly the same as before.

4.2.3 Link decorating results

Link decorating tests were constructed by manually appending known trackers and visiting the `https://example.com/`. By doing this, we can directly compare how many trackers were removed after visitation.

The results are displayed as a *"cleanliness"* score - the more link decoration trackers were removed, the lower and better the score.

The main function in code which is the same as in the repository is as in the following Listing 4.5:

```
const testLinkDecorating = async (
  testOptions: TestOptionsType
): Promise<void> => {
  console.log("\nTesting␣link␣decorating...");
```

```
  const { page, currentArgs } = testOptions;
  const { searchEngine } = currentArgs;

  const templateURL: string = "https://example.com";
  const decoratedURL: string = decorateLink(templateURL, LINK_DECORATORS);

  await navigateToWebsiteThroughSearchEngine(page, searchEngine,
      decoratedURL);

  displayResults(page, currentArgs);
};
```

Listing 4.5: Main function for link decorating.

The link decorating results from manually appending known trackers to `https://example.com/` for the three browser engines for each possible combination of search engines and extensions are as in Table 4.3:

Browser Engine	**Search Engine**	**Extension**	**Link Decorating Results**
Chromium	All search engines	All extensions	No trackers were removed
Firefox	All search engines	All extensions	No trackers were removed
WebKit	All search engines	All extensions	No trackers were removed for dockerized tests run on Linux, while all trackers were removed for dockerized tests run on MacOS

Table 4.3: Link decorating test results

The link decorating results confirm the theoretical conclusion from the

Safari subsection 3.1.2 where it's explained that the Safari browser and software from Apple rely on Apple's built in privacy mechanisms rather than the software itself.

Chapter 5

Conclusion

This thesis has investigated modern cookieless tracking techniques and evaluated the effectiveness of various privacy protection measures. Through both theoretical analysis and practical testing, several key findings have emerged.

5.1 Key findings

The comprehensive analysis of cookieless tracking techniques and privacy protection measures has revealed several critical insights that challenge conventional assumptions about web privacy. These findings, derived from both theoretical examination and practical testing across multiple browser engines, extensions, and search platforms, demonstrate the complex landscape of modern digital tracking and the limitations of current privacy solutions. The results highlight significant gaps in protection capabilities and underscore the necessity for multi-layered approaches to achieving meaningful privacy online.

- **Browser Engine Limitations:** The practical tests reveal that browser engines (Chromium, Firefox, WebKit) alone provide insufficient protection against tracking techniques. Even when combined with privacy-enhancing extensions, they failed to prevent fingerprinting, bounce tracking, and link decoration tracking effectively.

- **Extension Effectiveness:** While privacy extensions offer some protection, they are not comprehensive solutions. Our testing showed that even popular extensions like uBlock Origin and Ghostery could not fully prevent cookieless tracking methods when used in isolation.
- **Search Engine Privacy:** The analysis of various search engines reveal a clear divide between privacy-focused alternatives (like DuckDuckGo, Brave Search, and Startpage) and mainstream options (Google, Bing). However, each privacy-focused search engine solution presents different trade-offs between privacy and functionality.
- **Layered Protection Approach:** The research indicates that effective privacy protection requires a multi-layered approach combining:
  - A privacy-focused browser with built-in protections.
  - Carefully selected privacy extensions.
  - Privacy-respecting search engines.

5.2 Practical recommendations

Drawing from the research findings, this section provides actionable guidance tailored to different user profiles and privacy requirements. These recommendations recognize that privacy needs vary significantly among users, from casual browsers seeking basic protection to privacy-conscious individuals requiring comprehensive safeguards. The suggestions presented here offer practical pathways for implementing effective privacy measures while maintaining usable browsing experiences across different technical skill levels and use cases.

- **For General Users:** The Brave browser represents the best balance between privacy and usability, especially for users transitioning from Chrome. It offers robust out-of-the-box privacy protection while maintaining compatibility with Chrome's ecosystem.

- **For Privacy-Conscious Users:**
  - **Primary Browser:** Brave or Firefox with enhanced privacy settings.
  - **Essential Extension:** uBlock Origin.
  - **Search Engine:** A combination of privacy-focused search engines based on specific needs:
    * Brave Search for independent results.
    * DuckDuckGo for general privacy-conscious searching.
    * Startpage for Google-quality results with enhanced privacy.
- **For Maximum Privacy:** Consider using specialized browsers like Tor Browser or Mullvad Browser, understanding that this may impact website functionality and user experience.

5.3 Future considerations

The rapidly evolving landscape of web tracking technologies and privacy protection measures necessitates forward-looking analysis of emerging challenges and opportunities. Understanding these evolving dynamics is crucial for anticipating new threats and developing more effective privacy strategies that can adapt to changing technological and regulatory environments.

- **Regulatory Framework:** The need for stronger privacy regulations that specifically address cookieless tracking methods.
- **Technical Innovation:** Development of more effective countermeasures against emerging tracking techniques.
- **User Education:** Improved awareness and understanding of privacy risks and protection measures.

5.4 Final thoughts

This research demonstrates that while complete protection against cookieless tracking remains challenging, users can significantly enhance their privacy through informed choices about their browsing tools and habits. The key is finding the right balance between privacy protection and practical usability based on individual needs and risk tolerance.

The future of web privacy will likely require continued evolution of both technical solutions and regulatory frameworks. As tracking methods continue to become more sophisticated, the importance of comprehensive, user-friendly privacy solutions will only increase.

Bibliography

[1] *General Data Protection Regulation*. URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng (visited on 06/02/2025).

[2] *California Consumer Privacy Act*. URL: https://oag.ca.gov/privacy/ccpa (visited on 06/02/2025).

[3] *What are cookies? What are the differences between them (session vs. persistent)?* URL: https://www.cisco.com/c/en/us/support/docs/security/web-security-appliance/117925-technote-csc-00.html (visited on 06/03/2025).

[4] Anthony D. Miyazaki. "Online Privacy and the Disclosure of Cookie Use: Effects on Consumer Trust and Anticipated Patronage". In: *Journal of Public Policy & Marketing* 27.1 (2008), pp. 19–33. DOI: 10.1509/jppm.27.1.19. eprint: https://doi.org/10.1509/jppm.27.1.19. URL: https://doi.org/10.1509/jppm.27.1.19.

[5] Ognjen Pantelic, Kristina Jovic, and Stefan Krstovic. "Cookies Implementation Analysis and the Impact on User Privacy Regarding GDPR and CCPA Regulations". In: *Sustainability* 14.9 (2022). ISSN: 2071-1050. DOI: 10.3390/su14095015. URL: https://www.mdpi.com/2071-1050/14/9/5015.

[6] *Enhanced Tracking Protection in Firefox for desktop*. URL: https://support.mozilla.org/en-US/kb/enhanced-tracking-protection-firefox-desktop (visited on 06/03/2025).

[7] *What Is Intelligent Tracking Prevention and How Does It Work? [versions 1.0 - 2.3]*. URL: https://clearcode.cc/blog/intelligent-tracking-prevention/ (visited on 06/03/2025).

[8] Emmanouil Papadogiannakis et al. "User Tracking in the Post-cookie Era: How Websites Bypass GDPR Consent to Track Users". In: *Proceedings of the Web Conference 2021*. WWW '21. ACM, Apr. 2021. DOI: 10.1145/3442381.3450056. URL: http://dx.doi.org/10.1145/3442381.3450056.

[9] *UK data regulator criticises Google for 'irresponsible' ad tracking change*. URL: https://www.theguardian.com/technology/2024/dec/19/google-advertisers-digital-fingerprints-ico-uk-data-regulator (visited on 06/03/2025).

[10] Pierre Laperdrix et al. "Browser Fingerprinting: A Survey". In: *ACM Trans. Web* 14.2 (Apr. 2020). ISSN: 1559-1131. DOI: 10.1145/3386040. URL: https://doi.org/10.1145/3386040.

[11] Paul D. Beale. *Statistical Mechanics*. Academic Press, 2011. ISBN: 9780123821898. URL: https://books.google.si/books?id=KdbJJAXQ-RsC.

[12] Peter Eckersley. "How Unique Is Your Web Browser?" In: *Privacy Enhancing Technologies*. Ed. by Mikhail J. Atallah and Nicholas J. Hopper. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 1–18. ISBN: 978-3-642-14527-8.

[13] Ove Sörensen. "Zombie-cookies: Case studies and mitigation". In: *8th International Conference for Internet Technology and Secured*

Transactions (ICITST-2013). 2013, pp. 321–326. DOI: 10.1109/ICITST.2013.6750214.

[14] Lukasz Olejnik, Claude Castelluccia, and Artur Janc. "Why Johnny Can't Browse in Peace: On the Uniqueness of Web Browsing History Patterns". In: *5th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2012)*. Vigo, Spain, July 2012. URL: https://inria.hal.science/hal-00747841.

[15] *Privacy and the :visited selector*. URL: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_selectors/Privacy_and_the_visited_selector (visited on 06/02/2025).

[16] Wikipedia contributors. *Canvas element — Wikipedia, The Free Encyclopedia*. URL: https://en.wikipedia.org/wiki/Canvas_element (visited on 06/03/2025).

[17] Muath A Obidat et al. *Canvas Deceiver-A New Defense Mechanism Against Canvas Fingerprinting*. 2021. (Visited on 06/03/2025).

[18] *The Design and Implementation of the Tor Browser*. URL: https://2019.www.torproject.org/projects/torbrowser/design/#fingerprinting-linkability (visited on 06/03/2025).

[19] Martin Koop, Erik Tews, and Stefan Katzenbeisser. "In-depth evaluation of redirect tracking and link usage". In: *Proceedings on Privacy Enhancing Technologies* (2020).

[20] *Debouncing*. URL: https://brave.com/privacy-updates/11-debouncing/ (visited on 06/03/2025).

[21] Shaoor Munir et al. "PURL: Safe and Effective Sanitization of Link Decoration". In: *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA: USENIX Association, Aug. 2024,

pp. 4103–4120. ISBN: 978-1-939133-44-1. URL: https://www.usenix.org/conference/usenixsecurity24/presentation/munir.

[22] *ETag Header - HTTP*. URL: https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/ETag (visited on 06/03/2025).

[23] *Cookieless Cookies*. URL: https://lucb1e.com/randomprojects/cookielesscookies/ (visited on 06/03/2025).

[24] Cyrus Peikari and Anton Chuvakin. *Security Warrior: Know Your Enemy*. " O'Reilly Media, Inc.", 2004.

[25] *Browser Market Share Worldwide*. URL: https://gs.statcounter.com/browser-market-share (visited on 06/02/2025).

[26] Ronghao Pan and Antonio Ruiz-Martínez. "Evolution of web tracking protection in Chrome". In: *Journal of Information Security and Applications* 79 (2023), p. 103643. ISSN: 2214-2126. DOI: 10.1016/j.jisa.2023.103643. URL: https://www.sciencedirect.com/science/article/pii/S2214212623002272.

[27] *Privacy Tests*. URL: https://privacytests.org/ (visited on 06/03/2025).

[28] *Microsoft Edge Privacy Whitepaper*. URL: https://learn.microsoft.com/en-us/legal/microsoft-edge/privacy (visited on 06/03/2025).

[29] Douglas J Leith. "Web browser privacy: What do browsers say when they phone home?" In: *IEEE Access* 9 (2021), pp. 41615–41627.

[30] *Mozilla hit with privacy complaint over Firefox user tracking*. URL: https://www.reuters.com/technology/mozilla-hit-with-privacy-complaint-over-firefox-user-tracking-2024-09-25/ (visited on 06/03/2025).

[31] *The Firefox Browser is a privacy nightmare on desktop and mobile.* URL: https://www.privateinternetaccess.com/blog/the-firefox-browser-is-a-privacy-nightmare-on-desktop-and-mobile/ (visited on 06/03/2025).

[32] *An update on our Terms of Use.* URL: https://blog.mozilla.org/en/firefox/update-on-terms-of-use/ (visited on 02/28/2025).

[33] *Advanced privacy. A long list of Brave's behind-the-scenes protections and commitments.* URL: https://brave.com/privacy-features/ (visited on 06/02/2025).

[34] Wikipedia contributors. *LibreWolf — Wikipedia, The Free Encyclopedia.* URL: https://librewolf.net/ (visited on 06/03/2025).

[35] *Free the internet with the Mullvad Browser.* URL: https://mullvad.net/en/browser (visited on 06/02/2025).

[36] Wikipedia contributors. *Opera (web browser) — Wikipedia, The Free Encyclopedia.* URL: https://www.opera.com/ (visited on 06/03/2025).

[37] ungoogled-software. *ungoogled-chromium.* URL: https://github.com/ungoogled-software/ungoogled-chromium (visited on 06/03/2025).

[38] Wikipedia contributors. *Vivaldi (web browser) — Wikipedia, The Free Encyclopedia.* URL: https://vivaldi.com/ (visited on 06/03/2025).

[39] *Am I totally anonymous if I use Tor?* URL: https://support.torproject.org/faq/staying-anonymous/ (visited on 06/02/2025).

[40] gorhill. *uBlock Origin.* URL: https://github.com/gorhill/uBlock (visited on 06/03/2025).

[41] EFForg. *Privacy Badger*. URL: https://github.com/EFForg/privacybadger (visited on 06/03/2025).

[42] *DNT*. URL: https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/DNT (visited on 06/02/2025).

[43] kkapsner. *CanvasBlocker*. URL: https://github.com/kkapsner/CanvasBlocker (visited on 06/03/2025).

[44] Clean URLs. *ClearURLs*. URL: https://github.com/ClearURLs/Addon (visited on 06/03/2025).

[45] Ghostery. *Ghostery*. URL: https://github.com/ghostery/ghostery-extension (visited on 06/03/2025).

[46] Florian Schaub et al. "Watching them watching me: Browser extensions' impact on user privacy awareness and concern". In: *NDSS workshop on usable security*. Vol. 10. 2016.

[47] Muhammad Muzamil et al. "Analysis of tracker-blockers performance". In: *Pakistan Journal of Engineering and Technology* 4.1 (2021), pp. 184–190.

[48] Wikipedia contributors. *Search Engines Privacy — Wikipedia, The Free Encyclopedia*. URL: https://en.wikipedia.org/wiki/Search_engine_privacy (visited on 06/03/2025).

[49] Wikipedia contributors. *Privacy concerns with Google — Wikipedia, The Free Encyclopedia*. URL: https://en.wikipedia.org/wiki/Privacy_concerns_with_Google (visited on 06/03/2025).

[50] Becca Caddy. *How to delete your Google search history and stop tracking*. URL: https://web.archive.org/web/20170324030045/https://www.wired.co.uk/article/google-history-search-tracking-data-how-to-delete (visited on 06/03/2025).

[51] Wikipedia contributors. *DuckDuckGo — Wikipedia, The Free Encyclopedia*. URL: https://en.wikipedia.org/wiki/DuckDuckGo (visited on 06/03/2025).

[52] *Microsoft Edge browsing activity for personalized advertising and experiences*. URL: https://support.microsoft.com/en-us/microsoft-edge/microsoft-edge-browsing-activity-for-personalized-advertising-and-experiences-37aa831e-6372-238e-f33f-7cd3f0e53679 (visited on 06/03/2025).

[53] *France fines Microsoft €60 million for imposing ad cookies on users*. URL: https://www.france24.com/en/technology/20221222-france-fines-microsoft-60-million-euros-over-imposing-ad-cookies-on-users (visited on 06/04/2025).

[54] *Privacy Evaluation for Microsoft Bing*. URL: https://privacy.commonsense.org/evaluation/Microsoft-Bing (visited on 06/04/2025).

[55] Peter Baker Charlie Savage Edward Wyatt. *U.S. Confirms That It Gathers Online Data Overseas*. URL: https://web.archive.org/web/20170216072437/http://www.nytimes.com/2013/06/07/us/nsa-verizon-calls.html (visited on 06/03/2025).

[56] *Brave Search privacy notice*. URL: https://search.brave.com/help/privacy-policy (visited on 04/21/2025).

[57] *How to use Brave Search: an overview*. URL: https://search.brave.com/help/index (visited on 04/21/2025).

[58] *Concerned Feedback: Brave Browser's Privacy and Security Issues.* URL: https://community.brave.com/t/concerned-feedback-brave-browsers-privacy-and-security-issues/575988 (visited on 06/03/2025).

[59] *Are There Any Valid Privacy or Security Concerns in This Discussion?* URL: https://community.brave.com/t/are-there-any-valid-privacy-or-security-concerns-in-this-discussion/507472 (visited on 06/03/2025).

[60] *Our Privacy Policy.* URL: https://www.startpage.com/en/privacy-policy (visited on 06/02/2025).

[61] *Startpage Merger and Recent History.* URL: https://en.wikipedia.org/wiki/Startpage.com#Merger_and_recent_history (visited on 06/03/2025).

[62] Wikipedia contributors. *Yandex — Wikipedia, The Free Encyclopedia.* URL: https://en.wikipedia.org/wiki/Yandex (visited on 06/03/2025).

[63] Wikipedia contributors. *Naver — Wikipedia, The Free Encyclopedia.* URL: https://en.wikipedia.org/wiki/Naver (visited on 06/03/2025).

[64] Wikipedia contributors. *Baidu — Wikipedia, The Free Encyclopedia.* URL: https://en.wikipedia.org/wiki/Baidu (visited on 06/03/2025).

[65] Wikipedia contributors. *Yahoo! Japan — Wikipedia, The Free Encyclopedia.* URL: https://en.wikipedia.org/wiki/Yahoo_Japan (visited on 06/03/2025).

[66] Wikipedia contributors. *Yahoo! Kimo — Wikipedia, The Free Encyclopedia.* URL: https://en.wikipedia.org/wiki/Yahoo_Kimo (visited on 06/03/2025).

[67] Wikipedia contributors. *Qwant — Wikipedia, The Free Encyclopedia*. URL: https://en.wikipedia.org/wiki/Qwant (visited on 06/03/2025).

[68] Wikipedia contributors. *Seznam.cz — Wikipedia, The Free Encyclopedia*. URL: https://en.wikipedia.org/wiki/Seznam.cz (visited on 06/03/2025).

[69] Wikipedia contributors. *Mojeek — Wikipedia, The Free Encyclopedia*. URL: https://en.wikipedia.org/wiki/Mojeek (visited on 06/03/2025).

[70] *Desktop Operating System Market Share Worldwide*. URL: https://gs.statcounter.com/os-market-share/desktop/worldwide (visited on 06/02/2025).

[71] *Node.js*. URL: https://nodejs.org/ (visited on 06/03/2025).

[72] *npm*. URL: https://www.npmjs.com/ (visited on 06/03/2025).

[73] *Playwright*. URL: https://playwright.dev/ (visited on 06/03/2025).

[74] *TypeScript*. URL: https://www.typescriptlang.org/ (visited on 06/03/2025).

[75] *Git*. URL: https://git-scm.com/ (visited on 06/03/2025).

[76] *Docker*. URL: https://www.docker.com/ (visited on 06/03/2025).

[77] *BrowserScan*. URL: https://www.browserscan.net/ (visited on 06/02/2025).