The Fragile Privacy of Encrypted Client Hello: Quantifying Systemic Gaps in a Centralized Ecosystem

Jannis Hajda, Bengin Oezdil, John Bauer, Felix Hoffmann, and Vaibhav Bajpai
Hasso Plattner Institute, University of Potsdam, Germany
{jannis.hajda, bengin.oezdil, john.bauer, felix.hoffmann}@student.hpi.de,
vaibhav.bajpai@hpi.de

***Abstract*—Encrypted Client Hello (ECH) aims to close the last major metadata leak in the Transport Layer Security (TLS) handshake. In this paper, we present the first comprehensive, end-to-end analysis of the ECH ecosystem, combining a longitudinal scan of over 13 million domains, global Firefox telemetry, and a novel active measurement campaign using a custom browser extension.**

Our results reveal a fragile ecosystem defined by extreme centralization and potential deployment gaps. We identify 1.1 million active ECH deployments, 99.99 % of which are controlled by Cloudflare. Furthermore, while roughly one in five Firefox clients initiate ECH handshakes daily, we identify a substantial gap between client-side readiness and server-side support. Most critically, our end-to-end audit demonstrates that successful ECH negotiation during the initial handshake does not guarantee hostname privacy across the entire page load. We detected 5207 unique cases out of 150 778 analyzed domains where privacy was compromised by persistent ECH negotiation failures on common subdomains like `www`, `cdn`, and `api`. We conclude that without broader provider adoption and holistic configuration management, ECH currently offers a false sense of security.

I. Introduction

The universal adoption of Hypertext Transfer Protocol Secure (HTTPS) has successfully encrypted the content of web traffic, protecting sensitive user data from passive surveillance. However, the metadata surrounding these connections remains a critical privacy vulnerability. Specifically, the Server Name Indicator (SNI) extension in the TLS handshake transmits the target hostname in cleartext, allowing on-path network observers to monitor browsing habits and block access to specific services.

To address this leakage, the Internet Engineering Task Force (IETF) recently standardized ECH as RFC 9849 [1]. Unlike previous attempts such as Encrypted Server Name Indication (ESNI), ECH encrypts the entire `ClientHello` message (see Section II), theoretically rendering the target hostname indistinguishable from the public-facing service provider. Deployment has rapidly accelerated with default support in major browsers like Chrome and Firefox, alongside Cloudflare's rollout to millions of free-tier customers.

Despite this momentum, the mere presence of ECH records in the Domain Name System (DNS) does not guarantee privacy in practice. The complexity of modern web infrastructure creates a fragile ecosystem where a single request can compromise an entire session. While prior work (Section III) has characterized the server-side availability of ECH, the community lacks a comprehensive, client-centric view. Existing studies did not quantify the gap between theoretical availability and the effective privacy users experience during full, complex page loads.

To address this gap, we structure our research around three core questions:

RQ1 Server-Side Deployment Landscape: What is the current state of server-side ECH adoption, and does the centralization of providers introduce new security risks?

RQ2 Client-Side Readiness: How does the availability of ECH in client browsers compare to server-side support, and what is the actual usage rate in the wild?

RQ3 Effective Privacy: How effective is ECH in protecting the target hostname during complex, multi-request page loads, and are there systematic configuration gaps that weaken these guarantees?

By combining longitudinal server scans, global Mozilla telemetry, and our custom `doech` browser extension (detailed in Section IV), we address these questions and make the following contributions:

- **Uncovering the Server-Side Monoculture (Section V):** Our scan of over 13 million domains identifies 1.1 million active ECH deployments, revealing a landscape dominated by a single provider (99.99 % Cloudflare). We find that independent deployments often fail to form meaningful anonymity sets, and that integrity protection is virtually non-existent, with only 15 records signed with Domain Name System Security Extensions (DNSSec).
- **Quantifying Client-Side Readiness (Section VI):** We quantify the gap between client capability and actual deployment, identifying a discrepancy of two orders of magnitude between potential (GREASE) and genuine ECH handshakes over TCP. Additionally, we find that approximately one in five Firefox users utilizes ECH on a daily basis.
- **Exposing End-to-End Privacy Leaks (Section VII):** Using our `doech` Firefox extension to analyze 150 778

ISBN 978-3-903176-82-9

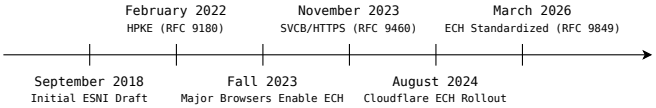

Fig. 1: Timeline of key milestones in the standardization and deployment of ECH.

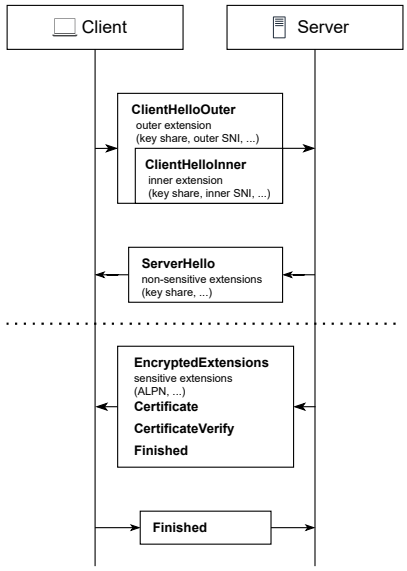

Fig. 2: Overview of the ECH handshake mechanism. The client encrypts and encapsulates the `ClientHelloInner` containing sensitive parameters inside a benign `ClientHelloOuter`.

domains, we identify systematic leakage in 5207 cases where the primary domain supported ECH while specific subdomains consistently failed ECH negotiation. Our results demonstrate that administrative aliases (e.g., `www`), static assets (e.g., `cdn`, `images`), and third-party trackers (e.g., `gtm`, `sst`) are frequently responsible for this leakage, pointing to apparent deployment gaps that undermine the privacy of the entire page load.

To support future research and transparency, we publicly release our Go-based scanning tool, the `doech` extension source code, and our own aggregated datasets alongside the analysis code. [1].

II. Background

While TLS 1.3 encrypts application data, the initial handshake remains unencrypted [1], [2]. A critical vulnerability is the leakage of connection metadata; specifically, the SNI extension transmits the target hostname in cleartext to facilitate virtual hosting. This exposes the destination to on-path observers, enabling traffic analysis and censorship despite subsequent payload encryption.

[1]https://github.com/JannisHajda/fragile-privacy-of-ech/

The initial countermeasure, ESNI, distributed public keys via DNS `TXT` records to encrypt the SNI [3], but proved insufficient. Its encryption was limited to the SNI, leaving other extensions in cleartext, and it lacked robust error-handling. Consequently, outdated keys from caching or tampering caused immediate connection failures without a mechanism for server-side correction.

ECH (RFC 9849) supersedes ESNI by encrypting the entire `ClientHello` [1], with key milestones summarized in Figure 1. Clients bootstrap this by retrieving the server's public key via DNS `HTTPS` or `SVCB` records [4]. As shown in Figure 2, the client encapsulates sensitive parameters in an encrypted `ClientHelloInner`, wrapped within a `ClientHelloOuter` carrying a benign outer SNI. This configuration creates an *anonymity set* where hostnames are indistinguishable from other services on the same infrastructure. To ensure reliability, servers can provide fresh keys via `RetryConfig` upon decryption failure, preventing the connection drops common in ESNI.

To prevent protocol ossification by middleboxes, ECH employs the `GREASE` mechanism [5]. Even when not performing an encrypted handshake, supporting clients insert a syntactically valid but randomized ECH extension into the `ClientHello`. This ensures that the extension is present in the vast majority of connections, not just those using ECH. Consequently, any middlebox attempting to block ECH by filtering this extension would inadvertently block standard traffic, raising the collateral cost of censorship.

Cryptographically, ECH relies on Hybrid Public Key Encryption (HPKE) to secure the handshake. HPKE combines a Key Encapsulation Mechanism (KEM) with Key Derivation Functions (KDFs) to establish a shared secret, which is then utilized by an Authenticated Encryption with Associated Data (AEAD) algorithm to encrypt the `ClientHelloInner`. This modular design provides cryptographic agility, allowing individual components to be upgraded independently, a critical feature for future migration to post-quantum algorithms [6], [7].

However, the retrieval of the `ECHConfig` via DNS creates a potential privacy leak. If the DNS records are retrieved via cleartext DNS (Do53), the query itself reveals the target domain, negating the hostname privacy gains of the encrypted handshake. Consequently, ECH provides meaningful hostname privacy only when paired with encrypted DNS transports like DNS over HTTPS (DoH) or DNS over TLS (DoT) [8], [9]. Furthermore, to ensure the ECH keys have not been spoofed or stripped by an active adversary, DNSSec is required to cryptographically verify the integrity and authenticity of the retrieved records [10].

While the protocol provides robust privacy guarantees, its effectiveness relies on widespread deployment. Although most major browsers have adopted ECH, server-side availability remains fragmented depending on the underlying library (see Table I), with `OpenSSL` notably lacking support as of early 2026.

TABLE I: Overview of client-side (top) and server-side ecosystem (bottom) support for ECH.

Client	Platform	ECH	Notes
Firefox	*Desktop, Android*	Yes	Since v119 [11]
Chrome	*Desktop, Android*	Yes	Since v105 [12], [13]
Edge	*Desktop, Android*	Yes	Chromium-based
Brave	*Desktop, Android*	Yes	Chromium-based
Opera	*Desktop, Android*	Yes	Chromium-based
Safari	*iOS, macOS*	Exp.	Disabled by default [14]

Project	ECH	Notes
SSL/TLS Libraries		
WolfSSL	Yes	[15]
BoringSSL	Yes	[16]
GoLang Crypto	Yes	Client & Server [17]–[19]
RustTLS	Yes	Client only [20], [21]
OpenSSL	No	Pre-Refinement phase since Aug 2025 [22], [23]
mbedTLS	No	
Conscrypt	No	In dev. [24]
Netty	No	[25]
Vert.x Core	No	Req. Netty [26]
Quarkus	No	Req. Vert.x Core [27]
Web Servers / Proxies		
Apache	No	Req. OpenSSL
NGINX	Yes	Collab with OpenSSL; supported by DEfO [28]
HAProxy	Yes	[29]
LightHTTPD	(Yes)	Depends on OpenSSL; can be compiled with ECH lib [30]
Hosting & Tools		
Cloudflare	Yes	Enabled by default on Free zones [31]
Wireshark	Yes	
SSLyze	No	

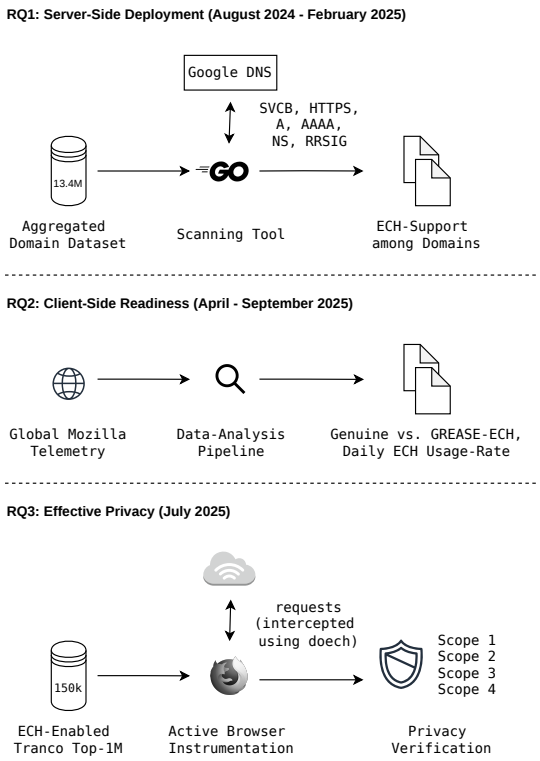

Fig. 3: Overview of our multi-perspective measurement approach.

III. Related Work

Early measurements of encrypted handshake evolution focused primarily on ESNI. In 2022, Tsiatsikas et al. [32] found that while ESNI had achieved moderate visibility (18.2% of the Tranco Top-1M), ECH support was virtually non-existent. This scarcity was further confirmed in 2023 by Zirngibl et al. [33], whose scan of 400 million domains identified only 20 valid ECH configurations, highlighting the slow initial transition from draft specifications to deployment.

The landscape shifted dramatically in late 2023. A longitudinal analysis by Dong et al. [34] captured the rapid, centralized rollout of ECH among the Tranco Top-1M list, identifying Cloudflare as the singular driver of adoption. Their data showed that Cloudflare-managed domains accounted for over 99.9% of all ECH support, a fragility underscored when adoption dropped to near-zero during a temporary provider-side disablement. This ongoing centralization was confirmed by a passive network analysis at a university campus in 2025 [35], which observed that while client-side readiness is high (driven by *GREASE* signals), genuine server-side support remains rare and largely confined to Cloudflare's authoritative infrastructure.

As ECH enters the mainstream, it has also become a target for state-level censorship. Recent investigations [36] reveal that the Russian Federation explicitly blocks `ClientHello` messages containing the public name `cloudflare-ech.com`. In contrast, China and Iran appear to target the underlying encrypted DNS channels rather than the extension itself, likely to minimize collateral damage to standard traffic.

Our work extends this body of research by bridging the gap between theoretical availability and practical privacy. While previous studies [34], [35] effectively quantified server-side adoption via DNS, they lacked visibility into the application-layer implications of partial deployment. By integrating a longitudinal server scan, global Mozilla telemetry, and active client-side instrumentation via our custom Firefox extension, we provide a holistic view of the ecosystem and the first empirical analysis of hostname leakage during full page loads. This allows us to move beyond binary "supported/unsupported" metrics and quantify the effective privacy provided to end-users in a complex web ecosystem.

IV. Methodology

Our methodology triangulates server-side adoption, client-side readiness, and end-to-end privacy (Figure 3) to evaluate the global ECH ecosystem.

A. Longitudinal Server-Side Adoption

We characterize server-side deployment trends of ECH by actively scanning `HTTPS` and `SVCB` DNS records over a six-month period, from August 2024 to February 2025.

a) Measurement Platform: The scanning infrastructure relies on a custom Go architecture built atop the `miekg/dns` library [37]. By resolving all queries via Google Public

TABLE II: Composition of the aggregated domain dataset used for longitudinal server-side analysis.

Dataset	Entries
OpenPageRank [39]	10 000 000
Tranco (QGN64, PNLVJ, YXLZG) [40]	3 000 000
Majestic (Million & Root) [41]	2 000 000
Cisco Umbrella [42]	1 000 000
Cloudflare Radar [43]	1 000 000
Crawlson [44]	1 000 000
Crux [45]	1 000 000
CitizenLab (Censorship, & Official Italy) [46]	41 787
ECH Test Domains (Handselected and [47])	13
Total Unique Domains	13 428 012

DNS, we ensured a consistent vantage point throughout the study. Beyond the essential `HTTPS` and `SVCB` records, our scanner concurrently fetches `A`, `AAAA`, `NS`, and `RRSIG` records to contextualize adoption patterns, mapping the resulting IP addresses to Autonomous Systems (ASes) via the MaxMind GeoLite2 database [38]. Scanning frequency increased from bi-weekly to daily over the observation period; we excluded incomplete runs between September and October 2024 to maintain data integrity following datacenter outages and rate-limiting events.

b) Domain Set: To ensure broad coverage, we aggregated a superset of 13 428 012 unique domains from major ranking lists (detailed in Table II). The normalization pipeline converted all entries to lowercase and removed malformed domains, but crucially retained `www` prefixes. This preservation is vital, as stripping prefixes might trigger resolution failures or redirects traffic to non-CDN replicas that lack ECH support.

B. Global Client Scale

Aggregated Firefox telemetry provided by Mozilla complements our active server-side scanning with a passive client-side perspective [48]. This analysis covers daily records from a stabilized window between April and September 2025 across 237 named countries and territories. Strict privacy standards govern this dataset: raw telemetry is aggregated by Mozilla into daily summaries indexed by date, geography, and specific metrics, including ECH outcomes (`GREASE` vs. real) and a privacy bitmask that groups sessions by the privacy features negotiated during the handshake. Crucially, each record reports two distinct counters: the total volume of TLS handshakes and the number of unique clients contributing to that specific bucket.

a) Client-Side Readiness vs. Server-Side Availability: Quantifying the gap between client readiness and server support requires analyzing daily handshake volumes. The explicit distinction in the telemetry between genuine ECH connections and `GREASE` signals allows us to measure the volume of TCP traffic supporting the protocol versus the fraction of connections actually using it.

b) Estimating Global Daily Usage Rate: We derive user estimates from the telemetry privacy bitmask buckets. Because individual clients frequently contribute to multiple buckets, summing these counts introduces duplicate counting and inflates totals. Instead, we establish a strict lower bound by defining the total daily user population as the maximum client count observed in any single bucket. Similarly, taking the maximum client count among only those buckets explicitly indicating active ECH negotiation yields a conservative, duplicate-free lower bound for daily ECH users.

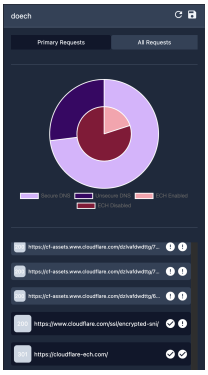

Fig. 4: The `doech` extension sidepanel interface. In real-time, it visualizes per-request ECH and DoH usage, exposing potential hostname leakage during a full page load.

C. End-to-End Privacy Verification

Neither server-side support nor client-side readiness guarantees effective privacy. True hostname protection requires the simultaneous usage of ECH and DoH for every request in a session, not just the initial handshake. Actual page loads typically involve fetching resources from distinct subdomains sharing the same registered domain (e.g., `api.example.com`, `cdn.example.com`). Deployment inconsistencies where the registered domain is configured to use ECH while a specific subdomain is not result in hostname leakage. To detect these gaps, we developed `doech`, a custom Firefox extension that audits the cryptographic consistency of every request in a browsing session.

a) Extension Setup: To observe network activity and extract relevant metadata, we leverage the `webRequest` API [49], which allows us to register listeners for specific events in the request lifecycle. We attach an `onHeadersReceived` listener [50] to capture metadata immediately after the response headers become available. Crucially, Firefox is currently the only major browser that exposes the `SecurityInfo` object [51], [52], granting us access to granular TLS handshake properties including ECH negotiation and private DNS usage. While Chromium-based browsers expose ECH status via the Chrome DevTools Protocol, they critically lack visibility into the usage of private DNS. Furthermore, choosing Firefox ensures methodological consistency with our Mozilla telemetry analysis and allows us to package our auditing tool as an accessible browser extension for end users, rather than a specialized debugging script [53]. We store each {`request, securityInfo`}

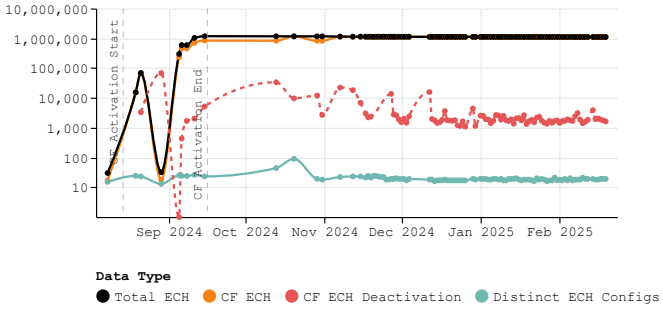

Fig. 5: Overall ECH adoption rate, the daily deactivation rate for Cloudflare-terminated domains, and the number of distinct ECH configurations observed over time, identifying Cloudflare as the primary driver of adoption.

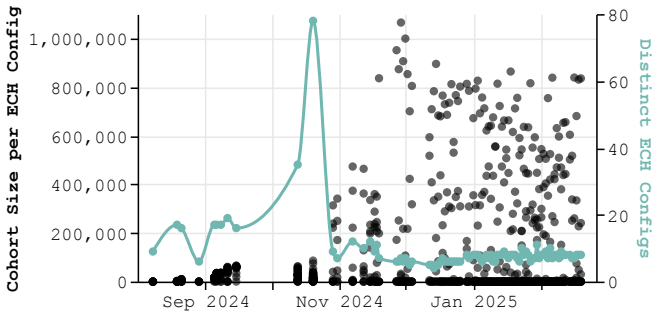

Fig. 6: Anonymity set sizes and number of distinct ECH configurations for public name `cloudflare-ech.com`.

pair in memory and update the extension sidepanel in real-time, providing immediate feedback to the user as depicted in Figure 4. Finally, `doech` allows exporting collected session data to facilitate offline analysis.

b) Evaluation Platform: We employ Selenium [54] to automate `doech`, evaluating the end-to-end privacy gains of domains on the Tranco Top-1M list [40] that advertise ECH support. Given the high computational overhead of full browser instrumentation compared to lightweight DNS probing, we restrict this analysis to this top-tier selection rather than the full 13 million dataset described in Section IV-A. We configured Firefox in "Mode 2" (DoH-preferred) using Cloudflare as the Trusted Recursive Resolver (TRR) [55], matching the browser's default provider selection.

V. Server-Side ECH Landscape

a) Overall ECH Adoption: Results from our longitudinal study confirm that the global ECH ecosystem is effectively a Cloudflare monoculture. The overall adoption trend, depicted in Figure 5, closely mirrors the trajectory of domains terminated by Cloudflare (identified by the public name `cloudflare-ech.com`). Cloudflare enabled ECH by default for free-tier users during a rollout period between August and September 2024 (marked as *CF Activation* in Figure 5). The sharp drop in late August corresponds to a temporary deactivation event, likely due to infrastructure maintenance. Following this rollout, the ecosystem stabilized; the daily rate of users disabling the feature was punctuated only by episodic spikes, which is to be expected with an ecosystem in active rollout and tuning. By 2025-02-19, out of the 13 428 012 unique domains monitored, we observed 1 106 666 active ECH deployments (8.2 %). Of these, Cloudflare accounted for 99.99 %. Crucially, only 15 of these 1.1 million records were cryptographically signed with DNSSec. This virtually non-existent integrity protection leaves the vast majority of ECH deployments vulnerable to DNS spoofing and downgrade attacks.

b) The ECH Configuration Monoculture: Cloudflare's dominance results in a massive discrepancy in anonymity set sizes, as shown in Figure 6. Individual configurations under `cloudflare-ech.com` frequently encompass over 1 million domains sharing an outer SNI. In contrast, non-Cloudflare deployments exhibit significantly smaller sets, including instances of size 1 where the outer SNI matches the domain, rendering hostname privacy benefits obsolete.

By mapping resolved `A` and `AAAA` responses to IP addresses using MaxMind GeoLite2 [38], we also identified multiple cases where domains utilized misleading public names potentially to obfuscate their destinations. For example, `google.com` served as the public name for eight configurations hosted on Amazon infrastructure in Tokyo and three on Hetzner in Helsinki. Similarly, domains associated with illicit streaming and gambling (such as `kinotam.org` and `spaces.im`) initially used `yandex.video` before switching to `world79.spcs.bio` as their shared public name.

While the massive anonymity set of Cloudflare-terminated domains prevents on-path observers from distinguishing specific services, it enables targeted censorship; by filtering traffic based on a specific outer SNI, censors can block the entire provider and therefore effectively the entire ECH infrastructure, as observed in Russia [36].

c) Cryptographic Homogenization: Beyond identity concentration, we observed a near-total homogenization of cryptographic parameters. As shown by the distinct configuration count in Figure 6, we identified only 10 to 30 distinct active ECH configurations throughout most of the observation period. The vast majority of domains rely exclusively on `HKDF-SHA-256` with `AES-GCM-128` for symmetric encryption. Alternative cipher suites, such as `ChaCha20-Poly1305`, appeared only in limited test deployments (e.g., `defo.ie`). Furthermore, `DHKEM(X25519, HKDF-SHA256)` was the sole KEM in use across all observed deployments.

While standardization simplifies deployment, it risks protocol ossification, potentially hindering future transitions to Post-Quantum Cryptography (PQC). Furthermore, this monoculture creates a systemic single point of failure: a compromise of Cloudflare's private key or a vulnerability in `X25519` or `AES-GCM` would simultaneously compromise the entire ECH ecosystem.

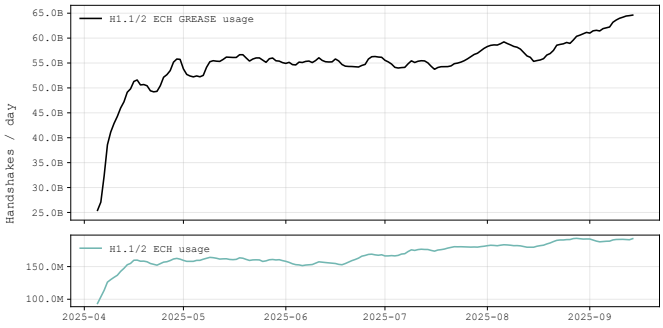

Fig. 7: Daily TLS handshake volume comparing client capability (GREASE) against genuine ECH handshakes. The two orders of magnitude difference quantifies the substantial gap between client-side readiness and server-side support (7-day moving average).

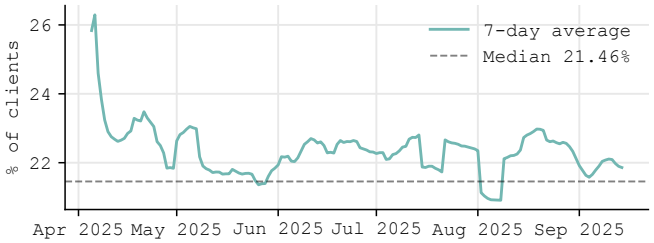

Fig. 8: Daily global share of unique Firefox clients utilizing ECH at least once. On average, approximately one in five users actively use the protocol daily (7-day moving average).

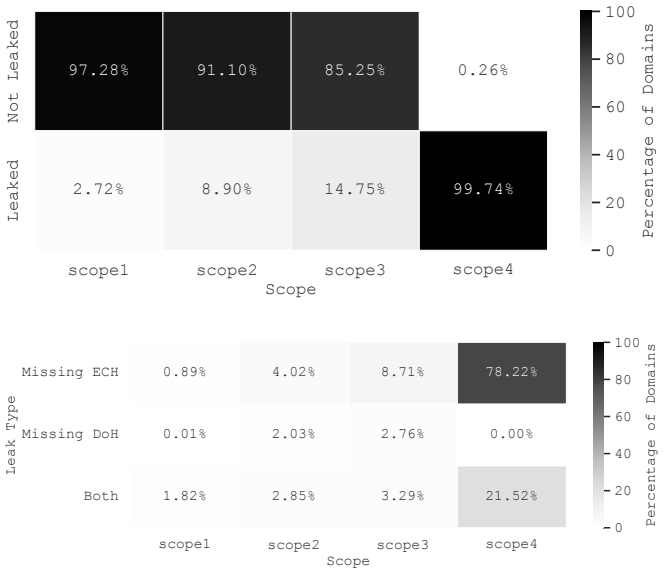

Fig. 9: Leakage among domains advertising ECH. **Top:** Proportion of domains indicating leakage across four cumulative scopes: (1) Initial Connection; (2) Same Registered Domain; (3) Redirect Chain; (4) Full Page Load. **Bottom:** Distribution of leakage causes within each scope, distinguishing missing ECH, missing DoH, or the absence of both.

VI. Client-Side Readiness

The privacy guarantees of ECH rely equally on the client's ability and willingness to negotiate encrypted handshakes. We extend our analysis to real-world client traffic using Mozilla telemetry to complement the server-side landscape.

a) Global Traffic Volumes: The daily volume of ECH and GREASE-ECH handshakes, depicted in Figure 7, reveals a massive gap between client readiness and server-side support. After an initial stabilization phase, the number of daily handshakes containing GREASE ECH signals rose until 2025-04-15, eventually reaching a consistent plateau of over 60 billion daily handshakes. Comparing this to the roughly 180 million handshakes that were performed using genuine ECH, we observe a discrepancy of more than two orders of magnitude. This profound misalignment between client-side demand and server-side availability aligns with previous passive measurement results [35].

b) Daily ECH Usage: Based on the methodology described in Section IV-B, we determine the number of daily clients attempting ECH negotiation, as shown in Figure 8. Following the stabilization in April, daily usage plateaus at a median of 21.46 % with negligible volatility (median daily change of 0.73 percentage points). These estimates rely on a robust sample of 45.13 million daily clients, accounting for nearly a quarter of the estimated 190–200 million monthly active users during this period [56]. In short, approximately one in five Firefox users utilizes ECH on a daily basis, effectively seeking to benefit from its promised privacy gains.

VII. End-to-End Privacy Analysis

Even if the client and server successfully negotiate ECH for the initial connection setup, this does not guarantee effective hostname privacy in real-world environments. We utilize our custom Firefox extension, `doech`, to perform an end-to-end privacy analysis from a client-side perspective.

a) Baseline of ECH Advertisement: Because our custom measurement pipeline relies on active browser instrumentation (described in Section IV-C), we performed this analysis on a fresh scan of the Tranco Top-1M list on 2025-07-14. We identified 154 981 domains (15.6 % of all responsive domains) publishing ECH keys via DNS. From this set, we successfully collected complete `doech` audit logs for 150 778 domains, which form the basis for the subsequent analysis. Consistent with our longitudinal findings in Section V, this ecosystem is effectively a monoculture: Cloudflare accounts for 99.99 % of these ECH-enabled domains.

b) Definition of Privacy Scopes: To quantify leakage throughout the entire page load, we define four cumulative scopes. Within each scope, we consider the privacy of the hostname compromised if a single request failing within that boundary does not use both ECH and DoH.

1) **Initial Connection:** The handshake of the very first request to the target domain.

TABLE III: Classification of leakage levels within Scope 2. The vast majority of leakages (80.05 %) occur on subdomains of registered domains supporting ECH (L2).

Leakage Level	Domains	Share
L1: Registered domain	1298	19.95 %
L2: Solely subdomains	5207	80.05 %
Total potential configuration gaps	6505	100.00 %

2) **Same Registered Domain:** All requests sharing the same registered domain as the original target.
3) **Redirect Chain:** All requests sharing the registered domain of any hop encountered during the redirect chain.
4) **Full Page Load:** Every request in the session, including third-party scripts and trackers.

The proportion of ECH-advertising domains maintaining privacy across these scopes, along with the specific causes of leakage (missing ECH, missing DoH, or both), is summarized in Figure 9.

c) Validation of Advertisement (Scope 1): The results of Scope 1 show that within our dataset, domains advertising ECH largely fulfill their promise for the initial connection. In 97.28 % of cases, the target hostname was successfully protected during the initial handshake. Deconstructing the failure causes reveals that roughly 2.71 % of the total dataset failed due to missing ECH. This indicates that "false advertising", where a domain publishes keys via DNS without actual server-side support, was rarely observed.

d) Configuration Gaps (Scope 2): Looking beyond the initial handshake reveals a distinct increase in hostname leakage. Overall, 8.90 % of domains advertising ECH support were compromised by requests to the same registered domain. Noticeably, 6.87 % of all analyzed domains leaked due to missing ECH usage, with 4.02 % leaking solely due to missing ECH rather than DoH issues.

To determine whether missing ECH usage stemmed from transient network errors or indicates configuration gaps, we performed a forensic analysis of the specific domains responsible for the hostname leakage. We consider a domain as unconfigured if none of the requests to it during the page load used ECH, while DoH was used at least once, confirming the client could resolve the domain and the leakage was not due to a sole resolution error. We then aggregated these potential configuration gaps by their registered domain to distinguish between issues at the registered domain level (L1) and those isolated solely to specific subdomains (L2).

The results reveal a stark imbalance, as shown in Table III. In 19.95 % (1298 cases) of these potential gaps, the registered domain entirely lacked ECH usage (L1). However, in the vast majority of cases (80.05 %, 5207 unique cases), the leakage occurred specifically on subdomains (L2). In these L2 scenarios, at least one request to the registered domain successfully used ECH, indicating that the registered domain is correctly configured. In contrast, specific subdomains consistently failed to establish an encrypted connection, pointing to a likely missing ECH configuration for those specific endpoints.

TABLE IV: Top 10 subdomains of registered domains supporting ECH responsible for L2 leakage. The list is dominated by likely internally managed subdomains (`www`, `api`) and third-party infrastructure accessed via `CNAME` records (`icdn05`, `sst`, `gtm`).

Subdomain Label	Incidents
`www`	1677
`cdn`	392
`api`	221
`icdn05`	115
`static`	107
`cdnstatic`	78
`images`	70
`img`	69
`sst`	67
`gtm`	63

To verify that these failures stem from deployment gaps rather than transient errors, we analyzed the specific subdomains responsible for L2-leakage. As detailed in Table IV, the results reveal a divide between internal deployment gaps and external infrastructure bottlenecks. Leaks on likely internally managed infrastructure (e.g. `www`, `api`) suggest administrators fail to secure their entire stack pervasively. Conversely, outsourced infrastructure creates systemic blind spots; the presence of shared infrastructure `CNAME`s like `icdn05` and aliases likely used for `CNAME` cloaking (`sst`, `gtm`) demonstrates that operators are constrained by third-party support. Securing these endpoints requires terminating infrastructure to actively support ECH and publish compatible keys. Consequently, reliance on third-party providers can expose user requests to on-path observers, a vulnerability first-party administrators cannot independently remediate.

e) Leakage across the Redirect Chain (Scope 3): Expanding the analysis to include every registered domain along the redirect chain reveals further privacy degradation. We observe a general leakage rate of 14.75 %, with missing ECH usage alone accounting for 8.71 % of the total dataset. In this scope, the privacy impact depends on the semantic relationship between the different hops. For instance, we observed that accessing `cloudflare-ech.com` uses ECH for the initial handshake but immediately redirects to `cloudflare.com`, which lacks ECH support. In such cases, this can allow an observer to retroactively infer the originally visited site based on the plaintext SNI of the redirect target.

f) The Third-Party Reality (Scope 4): Finally, Scope 4 reveals that 99.74 % of page loads leaked at least one hostname. This near-total leakage does not imply ECH is ineffective, as its primary goal is protecting the identity of the visited site; rather, it reflects a heavy reliance on third-party resources lacking ECH support. While these leaks do not directly expose the primary hostname, they significantly expand the metadata available to on-path observers.

Furthermore, leakage involving missing DoH spiked to

21.52 % in Scope 4. This likely reflects our resolver's fallback mechanism when third-party trackers are either unreachable via DoH, intentionally blocked by the provider, or fail to resolve within performance timeouts.

VIII. Discussion and Conclusion

This study characterizes the global ECH ecosystem, demonstrating that reliance on single-vantage measurements risks creating a false sense of security. By triangulating server-side availability via DNS scanning, client-side readiness via Mozilla telemetry, and actual end-to-end privacy gains via active browser instrumentation, our analysis captures the full operational reality of the protocol.

A. Centralization and Fragility

Longitudinal results between August 2024 and February 2025 from over 13 million DNS records identify Cloudflare as the dominant driver of adoption, accounting for 99.99 % of the 1.1 million domains indicating ECH support. This results in a highly centralized infrastructure and a distinct asymmetry: while Cloudflare-terminated endpoints benefit from massive anonymity sets, independent deployments remain fragmented and small. To realize the full potential of ECH, other major hosting providers must follow suit; diversifying the ecosystem is essential to reduce reliance on a single entity, increase cryptographic variance, and prevent targeted blocking. Furthermore, the widespread absence of DNSSec signing among ECH records introduces a critical vulnerability, leaving clients susceptible to DNS poisoning and downgrade attacks.

B. The Supply-Demand Gap

While server-side adoption is growing, our analysis of Mozilla Firefox telemetry reveals a substantial lag behind client-side readiness. We observe a discrepancy of two orders of magnitude between potential usage (GREASE signals) and genuine ECH handshakes. Although our methodology for estimating daily user counts acts as a conservative lower bound, we observed that approximately one in five Firefox users negotiated ECH at least once daily between April and September 2025. This metric serves as a strong signal of unmet demand: the user base is actively attempting to utilize ECH, but the server ecosystem has yet to catch up.

C. Implementation Realities vs. Privacy Promises

Our active client-side audit demonstrates that successful ECH negotiation during the initial handshakes does not guarantee effective hostname privacy, which is an insight invisible to sole DNS scans. While Scope 1 success was high, 8.9 % of the 150 778 analyzed domains advertising ECH via DNS leaked the original hostname during the initial page load in Scope 2. This figure is likely to rise as extended sessions trigger more sub-resource requests. Forensic analysis suggests persistent configuration gaps rather than transient errors; we identified 5207 domains where subdomains successfully resolved via DoH yet consistently failed ECH negotiation. These ranged from administrative endpoints (`www`, `api`) to potential `CNAME` cloaking targets (`sst`, `gtm`). Without secondary DNS lookups, we cannot pinpoint the exact failure mechanism, such as missing records or mismatched keys, nor confirm if internal subdomains are aliased to third parties. Nevertheless, these consistent failures highlight a systemic challenge: true privacy requires pervasive ECH deployment and rigorous configuration management across all interconnected infrastructure by all involved parties.

We acknowledge that our perspectives are methodologically bounded: our active audit relies on a single client configuration and domain subset, our longitudinal scans depend on a single DNS resolver, and both datasets span different time windows within a heavily Cloudflare-dominated ecosystem. To address these constraints, future work must integrate our large-scale DNS and active `doech` pipelines into a synchronized, multi-vantage framework. This combined pipeline will not only pinpoint exact failure mechanisms at a global scale but must also evaluate the inferential privacy risks of third-party dependencies. Specifically, investigating whether the extensive metadata leakage observed across redirect chains (Scope 3) and full page loads (Scope 4) allows observers to deanonymize the primary hostname via SNI-based fingerprinting remains a critical open question. Nevertheless, even within these bounds, our current findings demonstrate that protocol adoption alone is insufficient.

Ultimately, the extreme centralization around Cloudflare and the prevalence of misconfigured subdomains reveal that achieving actual end-to-end privacy requires broader deployment by hosting providers alongside rigorous, holistic configuration management by domain administrators.

IX. Ethical Considerations

We adhered to standard ethical practices by maintaining strict rate limits for server-side scanning and limiting browser instrumentation to initial page loads. Additionally, our telemetry analysis relied strictly on aggregated, anonymized datasets to guarantee user privacy.

X. Acknowledgements

The authors thank Mozilla for providing the Firefox telemetry data used for parts of this study. The authors also acknowledge Vasilis Ververis for shepherding the students during the early stages of this research.

References

[1] E. Rescorla, K. Oku, N. Sullivan, and C. A. Wood, "TLS Encrypted Client Hello," RFC 9849, Mar. 2026.

[2] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, Aug. 2018.

[3] E. Rescorla, K. Oku, N. Sullivan, and C. A. Wood, "Encrypted Server Name Indication for TLS 1.3," Internet Engineering Task Force, Internet-Draft draft-ietf-tls-esni-01, accessed: 2025-12-16. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-tls-esni/01/

[4] B. M. Schwartz, M. Bishop, and E. Nygren, "Service Binding and Parameter Specification via the DNS (SVCB and HTTPS Resource Records)," RFC 9460, Nov. 2023.

[5] D. Benjamin, "Applying Generate Random Extensions And Sustain Extensibility (GREASE) to TLS Extensibility," RFC 8701, Jan. 2020.

[6] R. Barnes, K. Bhargavan, B. Lipp, and C. A. Wood, "Hybrid Public Key Encryption," RFC 9180, Feb. 2022.

[7] B. Westerbaan and C. D. Rubin, "Defending against future threats: Cloudflare goes post-quantum," Oct. 2022. [Online]. Available: https://blog.cloudflare.com/post-quantum-for-all

[8] P. E. Hoffman and P. McManus, "DNS Queries over HTTPS (DoH)," RFC 8484, Oct. 2018.

[9] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. E. Hoffman, "Specification for DNS over Transport Layer Security (TLS)," RFC 7858, May 2016.

[10] S. Rose, M. Larson, D. Massey, R. Austein, and R. Arends, "DNS Security Introduction and Requirements," RFC 4033, Mar. 2005.

[11] Mozilla, "Encrypted Client Hello (ECH) - Frequently asked questions," Aug. 2024, accessed: 2026-02-07. [Online]. Available: https://support.mozilla.org/en-US/kb/faq-encrypted-client-hello

[12] D. Benjamin, "Feature: TLS Encrypted Client Hello (ECH)," Dec. 2023, accessed: 2026-02-07. [Online]. Available: https://chromestatus.com/feature/6196703843581952

[13] ——, "Post-launch ECH cleanup (5100554) · Gerrit Code Review," Dec. 2023, accessed: 2026-02-07. [Online]. Available: https://chromium-review.googlesource.com/c/chromium/src/+/5100554

[14] Apple Inc., "sec_protocol_options_set_enable_encrypted_client_hello — Apple Developer Documentation," 2024, accessed: 2026-02-07. [Online]. Available: https://developer.apple.com/documentation/security/sec_protocol_options_set_enable_encrypted_client_hello

[15] wolfSSL, "Encrypted Client Hello (ECH) now supported in wolfSSL - wolfSSL," Dec. 2022, accessed: 2024-12-06. [Online]. Available: https://www.wolfssl.com/encrypted-client-hello-ech-now-supported-wolfssl/

[16] BoringSSL Contributors, "ssl/encrypted_client_hello.cc," 2024, accessed: 2026-02-07. [Online]. Available: https://boringssl.googlesource.com/boringssl/+/master/ssl/encrypted_client_hello.cc

[17] Golang.org, "Go 1.23 Release Notes - The Go Programming Language," Aug. 2024, accessed: 2024-12-06. [Online]. Available: https://go.dev/doc/go1.23

[18] ——, "Go 1.24 Release Notes - The Go Programming Language," 2024, accessed: 2024-12-06. [Online]. Available: https://go.dev/doc/go1.24

[19] ——, "Go 1.24 is released! - The Go Programming Language," Feb. 2025, accessed: 2025-03-01. [Online]. Available: https://go.dev/blog/go1.24

[20] D. McCarney, "Support Encrypted ClientHellos (ECH, formerly ESNI) · Issue #199 · rustls/rustls," Jul. 2024, accessed: 2026-02-07. [Online]. Available: https://github.com/rustls/rustls/issues/199

[21] ——, "Server-side Encrypted Client Hello (ECH) support · Issue #1980 · rustls/rustls," May 2024, accessed: 2026-02-07. [Online]. Available: https://github.com/rustls/rustls/issues/1980

[22] A. Craig, "Support Encrypted Client Hello (formerly known as ESNI) · Issue #7482 · openssl/openssl," Oct. 2018, accessed: 2026-02-07. [Online]. Available: https://github.com/openssl/openssl/issues/7482

[23] N. Horman, "Community Assistance: Encrypted Client Hello," Oct. 2018, accessed: 2026-02-07. [Online]. Available: https://github.com/openssl/project/issues/892

[24] H.-C. Steiner, "Basic TLS Encrypted ClientHello (ECH) support by eighthave · Pull Request #1044 · google/conscrypt," Nov. 2021, accessed: 2026-02-07. [Online]. Available: https://github.com/google/conscrypt/pull/1044

[25] Glennos, "TLS Encrypted Client Hello (ECH) · Issue #12155 · netty/netty," Mar. 2022, accessed: 2026-02-07. [Online]. Available: https://github.com/netty/netty/issues/12155

[26] F. Almalki, "Add support for TLS Encrypted Client Hello (ECH) · Issue #5018 · eclipse-vertx/vert.x," Dec. 2023, accessed: 2026-02-07. [Online]. Available: https://github.com/eclipse-vertx/vert.x/issues/5018

[27] ——, "Add support for TLS Encrypted Client Hello (ECH) · Issue #37512 · quarkusio/quarkus," Dec. 2023, accessed: 2026-02-07. [Online]. Available: https://github.com/quarkusio/quarkus/issues/37512

[28] sftcd, "OpenSSL ECH integration," accessed: 2026-02-07. [Online]. Available: https://github.com/nginx/nginx/pull/840

[29] Tristan971, "ECH (Encrypted client hello) support · Issue #1924 · haproxy/haproxy," Nov. 2022, accessed: 2026-02-07. [Online]. Available: https://github.com/haproxy/haproxy/issues/1924

[30] LightHTTPD, "TLS ECH - Lighttpd - lighty labs," Feb. 2025, accessed: 2026-02-07. [Online]. Available: https://redmine.lighttpd.net/projects/lighttpd/wiki/TLS_ECH

[31] Cloudflare, "ECH Protocol," 2024, accessed: 2026-02-07. [Online]. Available: https://developers.cloudflare.com/ssl/edge-certificates/ech/

[32] Z. Tsiatsikas, G. Karopoulos, and G. Kambourakis, "Measuring the Adoption of TLS Encrypted Client Hello Extension and Its Forebear in the Wild," in *Computer Security. ESORICS 2022 International Workshops*, 2023, pp. 177–190.

[33] J. Zirngibl, P. Sattler, and G. Carle, "A First Look at SVCB and HTTPS DNS Resource Records in the Wild," in *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, Jul. 2023, pp. 470–474.

[34] H. Dong, Y. Zhang, H. Lee, S. Huque, and Y. Sun, "Exploring the Ecosystem of DNS HTTPS Resource Records: An End-to-End Perspective," in *Proceedings of the 2024 ACM on Internet Measurement Conference*. ACM, Nov. 2024, pp. 423–440.

[35] G. Merlach, M. Trevisan, and D. Giordano, "Encrypted Client Hello Is Coming: A View from Passive Measurements," *Network*, vol. 5, no. 3, 2025.

[36] N. Niere, F. Lange, N. Heitmann, and J. Somorovsky, "Encrypted Client Hello (ECH) in Censorship Circumvention," 2025, accessed: 2025-12-16. [Online]. Available: https://www.petsymposium.org/foci/2025/foci-2025-0016.pdf

[37] M. Gieben, "miekg/dns," Nov. 2024, accessed: 2024-11-21. [Online]. Available: https://github.com/miekg/dns

[38] MaxMind, "GeoLite Databases and Web Services," Aug. 2025, accessed: 2026-02-05. [Online]. Available: https://dev.maxmind.com/geoip/geolite2-free-geolocation-data

[39] DomCop, "Top 10 million websites based on Open data from Common Crawl & Common Search," 2024, accessed: 2024-11-21. [Online]. Available: https://www.domcop.com/top-10-million-websites

[40] Tranco, "A research-oriented top sites ranking hardened against manipulation," 2024, accessed: 2024-11-21. [Online]. Available: https://tranco-list.eu/

[41] Majestic, "The Majestic Million," 2024, accessed: 2024-11-21. [Online]. Available: https://majestic.com/reports/majestic-million?domain=&majesticMillionType=0&tld=&oq=&canUseDefault=

[42] Cisco, "Umbrella Popularity List," 2016, accessed: 2024-11-21. [Online]. Available: https://umbrella-static.s3-us-west-1.amazonaws.com/index.html

[43] Cloudflare, "Domain Rankings Worldwide," Nov. 2024, accessed: 2024-05-31. [Online]. Available: https://radar.cloudflare.com/domains

[44] Crawlson, "Crawlson Dataset," 2024, accessed: 2024-11-21. [Online]. Available: https://www.crawlson.com/

[45] Z. Durumeric, "zakird/crux-top-lists," Nov. 2024, accessed: 2024-07-06. [Online]. Available: https://github.com/zakird/crux-top-lists

[46] Citizen Lab, "citizenlab/test-lists," Nov. 2024, accessed: 2024-11-21. [Online]. Available: https://github.com/citizenlab/test-lists

[47] Tolerant Networks Ltd, "Developing ECH for OpenSSL (DEfO)," Jan. 2023, accessed: 2024-11-21. [Online]. Available: https://defo.ie/

[48] Mozilla, "Firefox telemetry: ECH adoption rate dataset," Google Cloud BigQuery, mozilla-public-data.telemetry_derived.ech_adoption_rate_v1, 2025.

[49] Mozilla, "webRequest," accessed: 2026-02-07. [Online]. Available: https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/webRequest

[50] ——, "webRequest.onHeadersReceived - Mozilla | MDN," Jul. 2025, accessed: 2026-02-05. [Online]. Available: https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/webRequest/onHeadersReceived

[51] ——, "webRequest.getSecurityInfo() - Mozilla | MDN," Jul. 2025, accessed: 2026-02-05. [Online]. Available: https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/webRequest/getSecurityInfo

[52] ——, "webRequest.SecurityInfo - Mozilla | MDN," Jul. 2025, accessed: 2026-02-05. [Online]. Available: https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/webRequest/SecurityInfo

[53] Google, "Chrome DevTools Protocol - Network.SecurityDetails," 2025, accessed: 2026-04-23. [Online]. Available: https://chromedevtools.github.io/devtools-protocol/tot/Network/#type-SecurityDetails

[54] Software Freedom Conservancy, "Selenium," Aug. 2025, accessed: 2026-02-05. [Online]. Available: https://www.selenium.dev

[55] Mozilla, "Trusted Recursive Resolver," 2025, accessed: 2026-02-05. [Online]. Available: https://wiki.mozilla.org/Trusted_Recursive_Resolver

[56] ——, "User Activity | Firefox Public Data Report," 2025, accessed: 2026-02-05. [Online]. Available: https://data.firefox.com/dashboard/user-activity