

SOEN 321: Information Systems Security

SECURITY ANALYSIS OF SOCIAL MEDIA APPLICATIONS

Christelle Charles 40249246

Zixin Deng 40047744

Imen Khezzar 40246836

Hanseok Kim 40281850

Sonali Patel 40176580

Vanisha Patel 40242554

Concordia University

April 22, 2025

Abstract

This report explores the privacy and security risks present in various social media applications and their websites through the hybrid evaluation model, which incorporates both static and dynamic analysis. The report identifies critical vulnerabilities such as Flash cross-domain policy misconfigurations, insecure CORS implementations, and injection-based flaws including XPath, Python code, OS command, and SQL injection. The testing tools involved with this study include Burp Suite (with a custom Python plugin), Mobile Security Framework (MobSF), Wireshark for packet inspection, and behavioural analysis of background data transmission. The research demonstrates that sometimes, even well-known platforms may be subjected to legacy misconfigurations or overlooked flaws that expose users to risks such as session hijacking, data leakage, unauthorized access, and server-side exploitation. Through combining automated and manual penetration testing methods based on OWASP WSTG and PTES standards, this report aims to deliver insights into improving application and web resilience.

1 Introduction

With the ever-increasing use of social media platforms in daily communication, content sharing, and community engagement, ensuring the security and privacy of these platforms has become a critical concern. Users put their trust in social media websites with a vast amount of personal data, including identities, conversations, preferences, and connections. A breach or exploitation in these platforms can lead to significant consequences such as identity theft, privacy violations, platform-level misinformation, and account hijacking. This project analyzes the security risks of 43 social media websites, examining how these platforms handle authentication, session control, cross-domain requests, and input validation. The scope of the analysis stems from mobile apps to websites, allowing the detection of both client and server-side flaws using a hybrid testing method using tools such as Burp Suite Professional (with the addition of a custom-developed Python plugin adapted using jython), MobSF, Wireshark, and in-browser network logging to simulate real-world attacks. The use of both static and dynamic evaluation allowed for a much deeper analysis to dig up hidden vulnerabilities that standard scanning may overlook. Lastly, this research aims to not only evaluate technical flaws but also raise awareness about various persistent risks in social media ecosystems and support for better security-by-design practices. By following the OWASP Web Security Testing Guide and the Penetration Testing Execution Standard (PTES), this report contributes to a more secure and privacy-conscious web environment.

2 Findings

2.1 Wireshark Background Activity Analysis

- **Facebook:** While the app was idle, query responses from endpoints such as `graph.facebook.com` and `b-graph.facebook.com` were detected.
- **Youtube:** Multiple instances of background activity initiated by the YouTube app and or Google were detected, such as frequent QUIC packets containing PING and CRYPTO frames and TLS 1.3 Client Hello message targeting `play.googleapis.com`.

- **Discord:** A TLS 1.3 connection was initiated to ‘cdn.discordapp.com’ (IP: 162.159.133.233), indicating that the app was actively communicating with Discord servers even when not in use.
- **Pinterest:** TLS connections to `www.pinterest.com` were observed while the app was idle, the app is likely maintaining a connection or sending telemetry in the background. It could be preloading data, syncing analytics or checking for new data.
- **Instagram:** The device initiated a DNS query for `gateway.instagram.com`, which resolved to 31.13.80.6 via the Meta-owned domain `dgw.c10r.facebook.com`. Shortly afterward, the device initiated a TLSv1.3 handshake with that server, explicitly identifying `gateway.instagram.com` in the SNI.
- **QQ:** Query responses from `astrategy.beacon.qq.com` endpoint were detected while the app was idle. These are often linked to behavioral tracking.
- **LinkedIn:** The `px.ads.linkedin.com` endpoint was accessed during idle time which can indicate analytics upload, session management or background session tracking.

2.2 Burp Suite Dynamic Analysis

- **Yiyo.io:** QR login and password reset endpoints lack CSRF protection, enabling unauthorized logins and user enumeration.
- **Tagged.com:** Exposes a permissive Flash policy (`crossdomain.xml`) and allows cross-origin requests with credentials, posing a session theft risk.
- **Mengchenghui.com:** Utilizes a vulnerable version of jQuery (v1.8.3), enabling cross-site scripting (XSS) via the `.load()` method.
- **Rumble.com (1):** Vulnerable to XPath injection due to unsanitized input in the URL path.
- **Rumble.com (2):** Sends user email and password in plaintext within both the request and response, exposing sensitive credentials.
- **Mingle2.com:** Accepts session cookies over unencrypted HTTP and lacks the Secure flag, making them susceptible to interception.
- **Plurk.com:** Allows universal Flash access through `crossdomain.xml`, introducing cross-origin security vulnerabilities.
- **Mastodon:** Transmits user email and password in plaintext, compromising sensitive user information.

3 Methodology

3.1 Approach

We conducted a hybrid application penetration test that combined manual testing and automated static and dynamic analysis. Our methodology is based on recognized industry standards, including the OWASP Web Security Testing Guide (WSTG) and the Penetration Testing Execution Standard (PTES). The objective was to identify security vulnerabilities related to authentication, authorization, session management, GraphQL APIs, WebSocket communications, and transport security.

3.2 Security Analysis Framework

The security issues considered in this assessment are grounded in previous research on the security of social media platforms [1, 2, 3] and refined to address a broad range of vulnerabilities relevant to modern web and mobile applications. To support the design of the security analysis framework, we systematically reviewed the academic literature on API security [4, 5], mobile application security [6], and recent industry threat reports [7]. The framework itself is aligned with established industry standards, including the OWASP Web Security Testing Guide (WSTG) [8], the Penetration Testing Execution Standard (PTES) [9], the OWASP API Security Top 10 [5], and the OWASP Mobile Security Testing Guide (MSTG) [10]. Our focus was on vulnerabilities that can be exploited by remote attackers, particularly those affecting authentication, authorization, session handling, API and WebSocket communication, input validation, and transport layer security across both mobile and web interfaces.

Overview: For the social media applications under evaluation, we conducted a comprehensive hybrid application penetration test. The process began with automated static analysis, followed by a dynamic assessment phase. We created dummy user accounts for each application and initiated an authenticated scan using Burp Suite. To enhance coverage beyond Burp’s default capabilities, we developed custom scripts to test additional areas, including authentication flaws, session management issues, transport layer security weaknesses, GraphQL vulnerabilities, and WebSocket exposure. For every medium- and high-severity finding we received from the scanner, we leveraged Burp Suite’s AI-powered analysis to perform deeper investigations by replaying and validating the corresponding requests. We then transitioned to manual dynamic analysis, where we used passive network monitoring to assess encryption strength, detect potential data leakage, and observe domain communication. This included reviewing POST/PUT traffic, unencrypted DNS queries, TLS handshake details, and background request patterns to uncover any silent or insecure transmissions.

3.2.1 Broken Authentication and Access Control

To detect broken authentication and access control vulnerabilities, we analyze how the application handles user identity tokens during requests to protected endpoints. Specifically, we send requests with no token, an invalid token, or a token belonging to a user with lower privileges. By comparing the server’s response to those of valid requests—looking at status codes, headers, and content—we determine whether access is improperly granted. If a profile or admin endpoint is accessible without proper verification, this indicates

a failure in enforcing access control policies. This technique is effective in detecting unauthorized access to user data, administrative dashboards, and account modification functions, which are commonly misconfigured in web and mobile applications.

3.2.2 Cross-Site Request Forgery and CORS Misconfiguration

We assess CSRF vulnerabilities by examining whether state-changing operations—such as updating user settings or triggering password resets—are protected against unauthorized cross-origin requests. We specifically look for requests that use form-encoded data, do not require custom headers, and lack anti-CSRF tokens. If these conditions are met, an attacker could trick a logged-in user into executing unintended actions. For CORS misconfiguration, we spoof the `Origin` header with a malicious domain (e.g., `http://evil.com`) and observe if the server responds with `Access-Control-Allow-Origin: *` and `Access-Control-Allow-Credentials: true`. Such configurations may allow third-party sites to access sensitive data using the victim’s authenticated session, compromising the browser’s same-origin policy.

3.2.3 Injection Attacks (XSS, XPath, SQL)

Injection vulnerabilities are identified by inserting crafted payloads into URL paths or input fields to test the robustness of client-side and server-side data handling. For cross-site scripting (XSS), we use payloads with various HTML and JavaScript encodings to determine if user inputs are reflected or stored without sanitization. For XPath injection, we analyze error messages and JavaScript logic that relies on functions like `XPathEvaluator()`, indicating client-side XML processing. SQL injection is tested by injecting expressions such as `' OR 1=1` or `sleep(5)` and observing timing delays or database error responses. These tests reveal the application’s failure to properly validate and sanitize user inputs, potentially enabling data theft or session hijacking.

3.2.4 Session Management and Sensitive Data Exposure

We evaluate session management by observing how authentication tokens and session cookies are handled during client-server communication. Requests are sent over unencrypted HTTP as well as HTTPS to determine if cookies are transmitted securely. We also inspect the `Set-Cookie` headers to confirm whether flags like `Secure` and `HttpOnly` are properly set. Failure to enforce secure transmission or isolate session data exposes the application to session hijacking through man-in-the-middle attacks. Additionally, we examine POST request payloads and API responses to identify any transmission of sensitive data such as plaintext passwords, email addresses, or authentication tokens. These issues often surface during manual inspection of HTTP traffic using tools like Burp Suite or Wireshark.

3.2.5 Background Activity and Telemetry Tracking

To identify hidden background activity, we use Wireshark to monitor network traffic while the app is idle. Filters such as `dns`, `http.request.method == POST`, and `tls.handshake.extensions_server_name` are applied to detect DNS queries, data uploads, and TLS handshakes to analytics or telemetry services. We analyze whether the

app establishes persistent connections, sends telemetry, or uploads logs without user interaction. Repeated pings or large POST requests when the app is not actively used may indicate aggressive background tracking, which raises concerns about privacy and excessive data usage, particularly on mobile networks.

3.2.6 Legacy Flash Cross-Domain Policies

Although modern applications rarely rely on Flash, some legacy platforms may still include outdated Flash support files. We check for the presence of permissive Flash cross-domain policies by requesting `/crossdomain.xml` and inspecting its contents. If the file includes `<allow-access-from domain="*">`, it means any external domain can interact with the app's Flash components, which violates the same-origin policy. Attackers can exploit this to run Flash-based scripts that steal data or impersonate users. While this is a legacy issue, its presence indicates neglected security hygiene and may point to other overlooked vulnerabilities in older systems.

3.3 Tools

We used Burp Suite Professional [11] as the primary tool for dynamic testing, including authenticated scans, request interception, and vulnerability validation. To enhance our testing of authentication, session management, GraphQL, and WebSocket security, we incorporated custom scripts and the AuthMatrix extension. In particular, we developed a custom Burp Suite plugin using Python to automate the detection of several classes of vulnerabilities during live traffic inspection. This plugin inspected both HTTP requests and responses to identify security issues such as credentials submitted over insecure channels, potential IDOR (Insecure Direct Object Reference) patterns, missing HTTP security headers, and insecure cookie configurations. It also flagged dangerous behaviors such as GraphQL introspection being enabled and insecure WebSocket upgrades. Each detection routine was designed around known attack surfaces: for instance, authentication flaws were flagged when credentials were submitted over HTTP; authorization flaws were inferred from exposed user identifiers; and insecure session cookies were logged when they lacked Secure or HttpOnly attributes. This automation allowed for consistent detection and logging of issues across all interactions without manual intervention. Wireshark[12] was used for passive network monitoring, enabling the analysis of DNS queries, TLS handshakes, and potential data leakage. For injection testing, we utilized Burp Suite Intruder to perform targeted payload-based attacks and identify vulnerabilities such as cross-site scripting (XSS) and SQL injection. For static analysis, we are using mobSF [13] This focused toolset enabled a comprehensive assessment across both mobile and web interfaces.

4 Results

4.1 Background Activity Analysis Results

The background activity analysis revealed that several popular mobile applications, including Facebook, YouTube, Discord, Pinterest, Instagram and Q,Q engage in background network activity when they appear idle to the user. For Facebook, DNS queries to endpoints such as `graph.facebook.com`, `b-graph.facebook` and `edge-mqtt.facebook`.

com suggest ongoing GraphQL communications and persistent MQTT connections, likely used for real-time features like Messenger presence, feed updates and telemetry reporting. Similarly, YouTube maintains persistent QUIC connections and initiates TLS handshakes with Google servers (e.g. `play.googleapis.com`), indicating background data preloading and session management aimed at providing a seamless video streaming experience. Discord’s idle time TLS connections to `cdn.discordapp.com` suggest synchronization of messages and assets, while Pinterest shows periodic TLS communications with its main domain, potentially for data synching and telemetry. Instagram also demonstrates active background communication through DNS and TLS handshakes with `gateway.instagram.com`, likely for preloading content and delivering notifications. The QQ app exhibits a similar pattern, accessing domains such as `strategy.beacon.qq.com`, indicating behavioural analytics and telemetry syncing. Notably, `px.ads.linkedin.com` was also accessed during idle time, pointing to background analytics or ad-related data exchange. While much of this behaviour is standard for modern apps aiming to enhance responsiveness and user experience, it may raise concerns for users focused on privacy.

4.2 A01: Broken Access Control

We identified critical broken access control vulnerabilities in multiple applications, which could allow attackers to bypass authorization mechanisms, escalate privileges, and access protected resources. These issues were validated through authenticated scans and manual testing using Burp Suite’s AI-powered analysis.

Trust Exploitation via Flash Policy Misconfiguration. On Tagged.com, we discovered a permissive `crossdomain.xml` policy that allowed any subdomain of `tagstat.com` to perform cross-origin requests with arbitrary headers. This overly broad trust model poses a significant risk, as a compromised or malicious subdomain could exploit this trust to act on behalf of authenticated users. Similar policy files were found across multiple paths (e.g., `/cdn/`, `/flash/`), increasing the platform’s attack surface.

Session Enforcement Failures. On Yiyo.io, the QR code login flow was found to be vulnerable to CSRF due to the `/auth/qrcode_check` endpoint accepting requests without validating a CSRF token. By submitting only a valid verification code (e.g., 873037) without the expected token, the server still returned a successful response, enabling unauthorized login attempts. Additionally, the password reset functionality lacked CSRF protection and leaked user account existence through varying response messages, allowing for user enumeration.

4.3 A02: Cryptographic Failures

The dynamic analysis revealed major issues with exposing sensitive data in the requests and responses. On `rumble.com`, the sign up process exposes confidential data such as email and password in plaintext in the request. In addition, the credentials are echoed back after a successful sign up in the response. Similarly, `mastodon.social` exposes email and password in the login process. This is a security risk because an attacker can monitor network traffic and intercept the credentials. The attacker can have access to the account. Many users reuse the same credentials for other services, therefore their other

accounts can also be compromised. It is important to encrypt sensitive encrypted data and to never echo back the data in the response.

4.4 A03: Injection

Vulnerable JQuery Library: In the security assessment of `mengchenghui.com`, we discovered that the webpage includes a vulnerable and outdated version of jQuery v.1.8.3. This allows cross-site scripting attacks using the `.load()` method, which fails to recognize and remove the `<script>` HTML tags that contain a whitespace character. During testing, many XSS payloads with whitespace variations were sent to target the `jQuery.load()` vulnerability. The response returned 200, meaning that jQuery failed to handle whitespace characters and the request did not get blocked. If the vulnerability is exploited, the attacker can inject JavaScript into the page leading to unauthorized actions performed on the webpage.

4.5 A05: Security Misconfiguration

Insecure Flash Cross-Domain Policy: On `Plurk.com`, we discovered a permissive `crossdomain.xml` file that allowed any domain to perform cross-origin requests with arbitrary headers. This overly broad trust model poses a significant risk, as a malicious website could exploit this trust to act on behalf of authenticated users. The file was also publicly cached, making it easily discoverable and exploitable.

CORS misconfiguration: This was discovered on `www.tagged.com` where the server accepts requests from arbitrary origins, including `http://evil.com`, and responds with `Access-Control-Allow-Credentials: true`. This behavior was confirmed on both the main page and the authentication endpoint (`/secure_login.html`), which also set session cookies in the response. This insecure configuration allows attackers to potentially steal sensitive user data or session credentials via cross-origin requests from a malicious website.

Insecure Session Cookie Transmission. On `Mingle2.com`, session cookies were transmitted over unencrypted HTTP connections and lacked the `Secure` flag. This misconfiguration exposes session cookies to interception via man-in-the-middle attacks, allowing attackers to hijack user sessions and access sensitive information.

5 Discussion

5.1 Practical Implications of Background Activity Findings

The background network activity observed across apps like Facebook, YouTube, Discord, Instagram, and QQ highlights a trade-off between user convenience and resource consumption. These apps maintain persistent connections to support real-time features such as instant messaging, content preloading and push notifications. While this enhances responsiveness and user experience, it can lead to increased mobile data usage, battery drain, and reduced device performance.

From a privacy standpoint, many of the background communications involve telemetry, analytics, and behavioural tracking. Endpoints like `strategy.beacon.qq.com`

and `px.ads.linkedin.com` suggest passive data collection, raising concerns about transparency and informed consent.

These findings emphasize the need for greater transparency, user control, and informed consent around background data operations. Platform providers should disclose telemetry practices clearly and give users fine-grained control over which background services are enabled.

5.2 Limitations

1. Scope of Evaluation: We assessed only publicly available or free-tier social media applications. Platforms requiring organizational identity verification or paid access were not included, which may limit the generalization of our results.

2. Ethical Boundaries: We did not attempt automated mass enumeration or brute-force attacks to avoid violating terms of service or ethical guidelines.

3. Tool reliance and Manual Work: While Burp Suite’s AI modules aided in injection and CSRF detection, certain logic flaws, such as broken object-level authorization, still required manual investigation.

4. Observation Window: Our passive monitoring setup captured background activity over a limited observation window; extended or long-term data collection might reveal additional patterns of concern.

5. Content-Related Risk Area: Due to the scope of this study, we did not analyze content moderation systems, recommendation algorithms, or abuse reporting mechanisms, which may also carry security or privacy implications.

5.3 Lessons Learned

From our study, several recurring and noteworthy observations emerged:

1. Background Data Collection Is Ubiquitous: Many applications maintain persistent connections for telemetry, tracking, and content preloading, even when idle. Users have minimal visibility or control over this behaviour, which can impact privacy and device resources.

2. Broken Access Control Is Still Prevalent: Vulnerabilities such as when CSRF protections are missing from sensitive endpoints, or when authorization tokens are not properly validated. Vulnerabilities like those found in Yiyo.io and Tagged.com highlight how common these flaws still are.

3. Legacy and Deprecated Configurations Persist: Permissive Flash crossdomain policies were still present in platforms like Plurk, demonstrating that deprecated features can remain exploitable years after their official obsolescence.

4. Input Validation and Injection Risks Are Ongoing: Weak input validation, including XPath injection and the use of outdated client-side libraries (e.g., jQuery 1.8.3), remain prevalent. These could allow attackers to manipulate DOM behavior or backend logic if not remediated.

5. Third-Party Domains Are Widely Used Without Transparency: Apps routinely communicate with third-party domains (e.g., ad networks and telemetry services) in the background. These background communications are not always clearly disclosed and may raise regulatory compliance issues under frameworks like GDPR and CCPA.

6. Sensitive Data Often Transmitted Insecurely: Sensitive data may not be handled securely, and an account can be compromised through network traffic analysis. It is important to have different passwords for each service.

5.4 Recommendation

1. Enhance User Control Over Background Activity: To address concerns around background activity and user control, developers and platform providers should give the users control over background activity. This can be achieved by offering various options to limit background activity, and platform-level tools should allow the users to manage how and when apps connect in the background. Additionally, privacy and resource management should be enhanced. This can be achieved by auditing app permissions and implementing restrictions on background data usage, particularly for users who are privacy-conscious.

2. Encrypt and Minimize Sensitive Data Exposure: For insecure data handling, developers need to ensure that credentials and sensitive data are never included in plaintext in an API request and response. The data needs to be securely transmitted.

3. Update Legacy Components and Libraries: To address the injection vulnerability caused by using an outdated jQuery library, developers and upgrade jQuery to the latest stable version. In addition, content security policy (CSP) can be used to help prevent XSS attacks by controlling which code is allowed to execute in the page. The CSP header stops the browser from running harmful scripts when they are injected [14].

References

- [1] E. S. Alashwali and K. B. Rasmussen, “Security of social media applications,” in *IEEE Symposium on Security and Privacy Workshops (SPW)*, 2018.
- [2] OWASP Foundation, “Owasp top 10: Web application security risks,” <https://owasp.org/www-project-top-ten/>, 2021.

- [3] Verizon, “2023 data breach investigations report,” <https://www.verizon.com/business/resources/reports/dbir/>, 2023.
- [4] OWASP Foundation, “Owasp api security top 10,” <https://owasp.org/www-project-api-security/>, 2023.
- [5] Salt Security, “Api security: A guide to api threats and solutions,” <https://salt.security/>, 2022, whitepaper.
- [6] ENISA, “Enisa threat landscape for mobile,” <https://www.enisa.europa.eu/publications/enisa-threat-landscape-for-mobile-devices>, 2021.
- [7] PortSwigger Research, “Exploiting graphql apis: An overview of techniques and attacks,” <https://portswigger.net/research/exploiting-graphql-apis>, 2022.
- [8] OWASP Foundation, “Owasp web security testing guide (wstg),” <https://owasp.org/www-project-web-security-testing-guide/>, 2021.
- [9] Penetration Testing Execution Standard (PTES), “Technical guidelines,” http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines, 2023.
- [10] OWASP Foundation, “Mobile security testing guide (mstg),” <https://owasp.org/www-project-mobile-security-testing-guide/>, 2022.
- [11] PortSwigger Ltd., “Burp suite professional,” <https://portswigger.net/burp/pro>, 2025.
- [12] The Wireshark Foundation, “Wireshark: The world’s foremost network protocol analyzer,” <https://www.wireshark.org/>, 2025.
- [13] MobSF Developers, “Mobile security framework (mobsf),” <https://github.com/MobSF/Mobile-Security-Framework-MobSF>, 2025.
- [14] MDN Web Docs, “Content security policy (csp),” n.d., accessed: 2025-04-22. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/CSP>

A List of Applications analyzed

Table 1: List of Applications Analyzed

Application	Application	Application	Application
Badoo	Bluesky	Bumble	Clubhouse
Discord	Facebook	imo	Instagram
Likee	Line	LinkedIn	Mastodon
Messenger	MeWe	mengchenghui	mingle2.com
Parler	Pinterest	Plurk	QQ
Quora	Reddit	Rumble	Skype
Snapchat	Steemit	Tango	Tapatalk
tagged	Teams	Telegram	Threads
TikTok	Tinder	Truth Social	Tumblr
Twitch	WhatsApp	X	Yiyo
YouTube	Yubo	Zello	

B Dynamic analysis (Wireshark)

DNS	137	Standard query response 0x524e AAAA b-graph.facebook.com CNAME star-mini.c10r.facebo...
DNS	125	Standard query response 0x56e8 A b-graph.facebook.com CNAME star-mini.c10r.facebook...
DNS	82	Standard query 0xf8bd AAAA edge-mqtt.facebook.com
DNS	82	Standard query 0xd37c A edge-mqtt.facebook.com
DNS	134	Standard query response 0xf8bd AAAA edge-mqtt.facebook.com CNAME mqtt.c10r.facebook...
DNS	122	Standard query response 0xd37c A edge-mqtt.facebook.com CNAME mqtt.c10r.facebook.com...
DNS	78	Standard query 0xe864 AAAA graph.facebook.com
DNS	78	Standard query 0xa2b9 A graph.facebook.com
DNS	130	Standard query response 0xe864 AAAA graph.facebook.com CNAME star.c10r.facebook.com ...
DNS	118	Standard query response 0xa2b9 A graph.facebook.com CNAME star.c10r.facebook.com A 3...
DNS	78	Standard query 0xa958 AAAA graph.facebook.com
DNS	78	Standard query 0xd1c4 A graph.facebook.com
DNS	130	Standard query response 0xa958 AAAA graph.facebook.com CNAME star.c10r.facebook.com ...
DNS	118	Standard query response 0xd1c4 A graph.facebook.com CNAME star.c10r.facebook.com A 3...

Figure 1: Facebook query responses received while idle

Protocol	Length	Info
QUIC	1292	Initial, DCID=928f5c32944e4d68, PKT: 3, CRYPTO, PADDING, CRYPTO, PADDING, PING, PADDING, CRYPTO, CRYPTO, PING, PADDING, PING, CRYPTO, PING, PING, PING, PADDING, PING, PADDING, PING, PAD...
QUIC	1292	Initial, DCID=31597b6c5e38728e, PKT: 2, PADDING, PING, CRYPTO, PING, PING, CRYPTO, CRYPTO, CRYPTO, PING, CRYPTO, PING, CRYPTO, PING, CRYPTO, PING, PING
TLSv1.3	370	Client Hello (SNI=proactlivebackend-pa.googleapis.com)
QUIC	1292	Initial, DCID=cc385268af6cd074, PKT: 2, CRYPTO, PING, PING, CRYPTO, PING, PADDING, CRYPTO, CRYPTO, PADDING, PING, CRYPTO, CRYPTO, PADDING
QUIC	1292	Initial, DCID=cf48a2b20f977762, PKT: 2, CRYPTO, PADDING, PING, CRYPTO, PADDING, CRYPTO, PADDING, PING, CRYPTO, CRYPTO, PING, PING, CRYPTO, PING, PING, PING, PING, PING, CRYPTO
QUIC	1292	Initial, DCID=38684d467c96633c, PKT: 2, PING, PING, PADDING, CRYPTO, CRYPTO, PADDING, CRYPTO, PADDING
QUIC	1292	Initial, DCID=ee3e3b389344b3ac, PKT: 2, PADDING, PING, PING, CRYPTO, CRYPTO, CRYPTO, PING, CRYPTO, PING, CRYPTO, PING
QUIC	1292	Initial, DCID=91a8c4d31530994, PKT: 2, CRYPTO, PING, PADDING, CRYPTO, PADDING, CRYPTO, PING, PING, PADDING, PING, PADDING
QUIC	1292	Initial, DCID=081609c34f6a3ac, PKT: 3, PING, PING, CRYPTO, CRYPTO, PING, PING, PING, PADDING, CRYPTO, PING, PING, PADDING, PING, PING
QUIC	1292	Initial, DCID=467d8dc151c0b7, PKT: 2, PING, CRYPTO, PING
QUIC	1292	Initial, DCID=7c5971a63993a3ai, PKT: 2, CRYPTO, CRYPTO, CRYPTO, CRYPTO, CRYPTO, PING, PADDING, PING, CRYPTO, PING, PADDING, CRYPTO, CRYPTO, CRYPTO, PADDING
QUIC	1292	Initial, DCID=467d8dc151c0b7, PKT: 2, PING, PADDING, CRYPTO, PING, PING, PING, CRYPTO, CRYPTO, PADDING, PING, PADDING, PING, PING, PING
QUIC	1292	Initial, DCID=4f6e4d4952b18dc, PKT: 2, CRYPTO, PING, PING, CRYPTO, PING, PING, PING, CRYPTO, CRYPTO, PADDING, CRYPTO, PADDING, PING, PING, PING, PING, PING, PADDING, CRYPTO, PADDING
TLSv1.3	571	Client Hello (SNI=play.googleapis.com)
QUIC	1292	Initial, DCID=361851fcc077426, PKT: 2, PING, CRYPTO, PING, PADDING, PING, CRYPTO, PING, PING, PING, CRYPTO, CRYPTO, CRYPTO, PING, PING, CRYPTO, PING, PING
QUIC	1292	Initial, DCID=09f7a20364dc0ff, PKT: 2, PADDING, CRYPTO, CRYPTO, PADDING, PING, PING, PING, CRYPTO, PING, PING, PING, PING, PING, PING
QUIC	1292	Initial, DCID=79f1b086eff7372, PKT: 2, CRYPTO, CRYPTO, PING, CRYPTO, PING
QUIC	1292	Initial, DCID=f2058ae76f72803, PKT: 2, PING, PING, PING, CRYPTO, CRYPTO, CRYPTO, PING, CRYPTO, PING, CRYPTO, CRYPTO, CRYPTO, CRYPTO
QUIC	1292	Initial, DCID=88342967080769d6, PKT: 2, PING, CRYPTO, CRYPTO, CRYPTO, PING, PING, PING
QUIC	1292	Initial, DCID=907f421ca7e11b07, PKT: 2, PING, CRYPTO, PING, CRYPTO, CRYPTO, CRYPTO, PING, CRYPTO

Figure 2: Frequent QUIC packets containing PING and CRYPTO frames received while YouTube was idle

```

DNS      83 Standard query 0x6815 AAAA astrategy.beacon.qq.com
DNS      83 Standard query 0xe436 A astrategy.beacon.qq.com
DNS     163 Standard query response 0xe436 A astrategy.beacon.qq.com CNAME ins-x9e4tvue.ias.tenc...
DNS     186 Standard query response 0x6815 AAAA astrategy.beacon.qq.com CNAME ins-x9e4tvue.ias.t...
DNS      83 Standard query 0x93c3 AAAA aeventlog.beacon.qq.com
DNS      83 Standard query 0xfb8c A aeventlog.beacon.qq.com
DNS     163 Standard query response 0xfb8c A aeventlog.beacon.qq.com CNAME ins-dv111tc4.ias.tenc...
DNS     186 Standard query response 0x93c3 AAAA aeventlog.beacon.qq.com CNAME ins-dv111tc4.ias.t...
DNS      83 Standard query 0x3167 A aeventlog.beacon.qq.com
DNS     163 Standard query response 0x3167 A aeventlog.beacon.qq.com CNAME ins-dv111tc4.ias.tenc...

```

Figure 3: Query responses received from QQ while the app was idle

```

79 Standard query 0x03fa AAAA px.ads.linkedin.com
79 Standard query 0x2fd9 A px.ads.linkedin.com
200 Standard query response 0x03fa AAAA px.ads.linkedin.com CNAME afd-lnkd.www.linkedin....
188 Standard query response 0x2fd9 A px.ads.linkedin.com CNAME afd-lnkd.www.linkedin.com...

```

Figure 4: The px.ads.linkedin.com domain being accessed while LinkedIn app was idle

C Dynamic analysis (Burpsuite)

```

1 GET /auth/login HTTP/1.1
2 Host: yiyo.io
3 Accept-Encoding: gzip, deflate, br
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US;q=0.9,en;q=0.8
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
7 Connection: close
8 Cookie: lang=en
9
10
11 HTTP/2 200 OK
12 Date: Fri, 11 Apr 2025 20:18:48 GMT
13 Content-Type: text/html; charset=UTF-8
14 Vary: Accept-Encoding
15 Strict-Transport-Security: max-age=31536000
16 Cf-Cache-Status: DYNAMIC
17 Report-To:
  {"endpoints":[{"url":"https://a.nel.cloudflare.com/report/v4?s=Sn0cqZw5F09fp72DrPq5qCk6jf4un4V3kYeGrEKSJggkpsZ0%2FI0BQYu0%2FcNHIZZVjHSsfLHmaz7eiEe7PkrNRu1pDHmuAXrTYI8GLJQ2y732R%2BGDTh9ANS8%3D"}],"group":"cf-nel","max_age":604800}
18 Nel:
  {"success_fraction":0,"report_to":"cf-nel","max_age":604800}
19 Server: cloudflare
20 Cf-Ray: 92ed2cdab900a2a2-YUL
21 Alt-Svc: h3=":443"; ma=86400
22 Server-Timing:
  cfl4;desc="?proto=TCP&rtt=10789&min_rtt=3747&rtt_var=7540&sent=96&recv=13&lost=0&retrans=0&sent_bytes=3697&recv_bytes=1092&delivery_rate=65538&cwnd=247&unsent_bytes=0&cid=281f93080164f342&ts=876x=0"
23
24 <!DOCTYPE html>

```

Figure 5: Request to /auth/login successfully returned the login page, confirming the authentication interface is accessible

<pre> 1 GET /auth/qrcode_check HTTP/1.1 2 Host: yiyo.io 3 Accept-Encoding: gzip, deflate, br 4 Accept: text/html,application/xhtml+xml,application/x ml;q=0.9,image/avif,image/webp,*/*;q=0.8 5 Accept-Language: en-US;q=0.9,en;q=0.8 6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36 7 Connection: close 8 Cookie: lang=en 9 10 </pre>	<pre> 1 HTTP/2 200 OK 2 Date: Fri, 11 Apr 2025 20:18:59 GMT 3 Content-Type: text/html; charset=UTF-8 4 Vary: Accept-Encoding 5 Strict-Transport-Security: max-age=31536000 6 Cf-Cache-Status: DYNAMIC 7 Report-To: {"endpoints":[{"url":"https://a.nel.cloudflare. com/report/v4?s=jfZxNyFdCs6Z4nNYCuMVVo EQa4K%2F%2B2tPXvYn6dDDW1pMQ6KGbi0V7LFDQdNG0or 9ynhhwZItDvvtq8PNctqITIGp0mMeqivKJmwt3%2B97VE jBHzuFU8fHsM%3D"}],"group":"cf-nel","max_age" :604800} 8 Nel: {"success_fraction":0,"report_to":"cf-nel","m ax_age":604800} 9 Server: cloudflare 10 Cf-Ray: 92ed2d1f3ce4a2a2-YUL 11 Alt-Svc: h3=":443"; ma=86400 12 Server-Timing: cfl4;desc="?proto=TCP&rtt=290566min_rtt=3747& rtt_var=32573&sent=23&recv=23&lost=0&retrans= 2&sent_bytes=9013&recv_bytes=1324&delivery_ra te=655538&cwnd=252&unsent_bytes=0&cid=281f930 80164f342&ts=110976;x=0" 13 14 <!DOCTYPE html> </pre>
---	--

Figure 6: Request to /auth/qrcode_check without parameters also returned HTTP 200, indicating the endpoint is reachable but inactive without input

<pre> 1 POST /auth/qrcode_check HTTP/1.1 2 Host: yiyo.io 3 Accept-Encoding: gzip, deflate, br 4 Accept: application/json, text/javascript, */*; q=0.01 5 Accept-Language: en-US;q=0.9,en;q=0.8 6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36 7 Connection: close 8 Cookie: lang=en 9 Content-Type: application/x-www-form-urlencoded 10 Content-Length: 13 11 12 number=873037 </pre>	<pre> 1 HTTP/2 200 OK 2 Date: Fri, 11 Apr 2025 20:19:09 GMT 3 Content-Type: text/html; charset=UTF-8 4 Strict-Transport-Security: max-age=31536000 5 Cf-Cache-Status: DYNAMIC 6 Report-To: {"endpoints":[{"url":"https://a.nel.cloudflare. com/report/v4?s=%2B%2BtQBmNtEhCny%2Bqvu LLld74UrKAKN75VfS6UVU132mzGKDKyNHl0%2F0udZFP1 TtWeNn%2FphxadwyYzlwneuoKqDjrzILDdNoMU%2F4N7 0UpDLU5MinsSKBS8js%3D"}],"group":"cf-nel","ma x_age":604800} 7 Nel: {"success_fraction":0,"report_to":"cf-nel","m ax_age":604800} 8 Server: cloudflare 9 Cf-Ray: 92ed2d612e72a2a2-YUL 10 Alt-Svc: h3=":443"; ma=86400 11 Server-Timing: cfl4;desc="?proto=TCP&rtt=229916min_rtt=3747& rtt_var=23514&sent=31&recv=29&lost=0&retrans= 2&sent_bytes=10562&recv_bytes=1552&delivery_r ate=655538&cwnd=252&unsent_bytes=0&cid=281f93 080164f342&ts=215766;x=0" 12 13 {"ret":0} </pre>
---	--

Figure 7: Request to /auth/qrcode_check with only a valid code (873037) and no token returned a success response, confirming that the endpoint does not validate CSRF tokens

```

1 POST /password/reset HTTP/1.1
2 Host: yiyo.io
3 Accept: application/json, text/javascript, */*; q=0.01
4 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like
  Gecko) Chrome/135.0.0.0 Safari/537.36
5 Cookie: lang=zh
6 Origin: https://yiyo.io
7 X-Requested-With: XMLHttpRequest
8 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
9 Content-Length: 15
10
11 email=$payload$

```

Figure 8: Request sent with various email address that we brute forced

Request	Payload	Status code	Length	Content type
0	admin@yiyo.io	200		text/html; charset...
1	administrator@yiyo.io	200		text/html; charset...
2	support@yiyo.io	200		text/html; charset...
3	info@yiyo.io	200		text/html; charset...
4	contact@yiyo.io	200		text/html; charset...
5	help@yiyo.io	200		text/html; charset...
6	service@yiyo.io	200		text/html; charset...
7	test@yiyo.io	200		text/html; charset...
8	user@yiyo.io	200		text/html; charset...
9	webmaster@yiyo.io	200		text/html; charset...

Figure 9: Successfully find valid emails in database

Advisory	Request	Response	Path to issue
Pretty	Raw	Hex	
1	GET /crossdomain.xml	HTTP/2	
2	Host: www.plurk.com		
3	Accept-Encoding: gzip, deflate, br		
4	Accept: */*		
5	Accept-Language: en-US;q=0.9,en;q=0.8		
6	User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36		
7	Connection: close		
8	Cache-Control: max-age=0		
9			
10			
Advisory	Request	Response	Path to issue
Pretty	Raw	Hex	Render
1	HTTP/2 200 OK		
2	Date: Sun, 20 Apr 2025 02:56:15 GMT		
3	Content-Type: application/xml; charset=utf-8		
4	Content-Disposition: inline; filename=crossdomain.xml		
5	Last-Modified: Fri, 18 Apr 2025 05:56:59 GMT		
6	Cache-Control: public, max-age=2678400		
7	Etag: W/"1744955819.4733067-203-1026692365"		
8	Cf-Cache-Status: MISS		
9	Expires: Wed, 21 May 2025 02:56:15 GMT		
10	Vary: Accept-Encoding		
11	Server: cloudflare		
12	Cf-Ray: 93315e117fd9ab33-YYZ		
13	Alt-Svc: h3=":443"; ma=86400		
14			
15	<?xml version="1.0"?>		
16	<!DOCTYPE cross-domain-policy SYSTEM "http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">		
17	<cross-domain-policy>		
18	<allow-access-from domain="*" />		
19	</cross-domain-policy>		
20			

Figure 10: The GET /crossdomain.xml request asks for the Flash security policy file. HTTP/2 200 OK response confirms that the file exists and allows all domains

Cross-origin resource sharing: arbitrary origin trusted

Advisory	Request	Response	Path to issue
Pretty	Raw	Hex	
1	POST /cdn-cgi/rum? HTTP/2		
2	Host: www.plurk.com		
3	Accept-Encoding: gzip, deflate, br		
4	Accept: */*		
5	Accept-Language: en-US;q=0.9,en;q=0.8		
6	User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36		
7	Connection: close		
8	Cache-Control: max-age=0		
9	Cookie: _ga=GA1.1.1573372922.1745118044; plurkcookiea=		
10	+dLKeFu0de4AFdwrBAGGPMV2aHX5wSqH7ZXmjV0CB10=?login_return_url=IIi=6&tid=bnVsbA==&user_id=bnVsbA==; _ga_15X0DL5VWS=GS1.1.1745118044.1.1.1745118216.60.0.0		
11	Origin: https://nkeyojlwovsm.com		
12	Referer: https://www.plurk.com/portal/		
13	Content-Type: application/json		
14	Sec-Ch-Ua: "Google Chrome";v="135", "Not=A?Brand";v="8", "Chromium";v="135"		
15	Sec-Ch-Ua-Platform: "macOS"		
16	Sec-Ch-Ua-Mobile: ?0		
17	Content-Length: 1060		
18	{		
19	"referrer": "",		
20	"eventType": 3,		
21	"versions": {		
22	"js": "2024.6.1",		
23	"fl": "2025.4.0-i-g37f21b1"		
24	},		
25	"pageLoadId": "91b10ac6-0035-4fda-9444-50f7616307bc",		
26	"location": "https://www.plurk.com/portal/",		
27	"landingPath": "/portal/",		
28	"startTime": 174511790525.3,		
29	"nt": "navigate",		
30	"serverTimings": {		
31	{		
32	"name": "cfCacheStatus",		
33	"dur": 0,		
34	"desc": "DYNAMIC"		
35	}		
36	},		
37	"siteToken": "f0f3f0d7075642cd877ac61c08650bb3",		
38	"lcp": {		
39	"value": 20628,		
40	}		
41	}		
42			
43			
44			
45			
46			
47			
48			
49			
50			
51			
52			
53			
54			
55			
56			
57			
58			
59			
60			
61			
62			
63			
64			
65			
66			
67			
68			
69			
70			
71			
72			
73			
74			
75			
76			
77			
78			
79			
80			
81			
82			
83			
84			
85			
86			
87			
88			
89			
90			
91			
92			
93			
94			
95			
96			
97			
98			
99			
100			
101			
102			
103			
104			
105			
106			
107			
108			
109			
110			
111			
112			
113			
114			
115			
116			
117			
118			
119			
120			
121			
122			
123			
124			
125			
126			
127			
128			
129			
130			
131			
132			
133			
134			
135			
136			
137			
138			
139			
140			
141			
142			
143			
144			
145			
146			
147			
148			
149			
150			
151			
152			
153			
154			
155			
156			
157			
158			
159			
160			
161			
162			
163			
164			
165			
166			
167			
168			
169			
170			
171			
172			
173			
174			
175			
176			
177			
178			
179			
180			
181			
182			
183			
184			
185			
186			
187			
188			
189			
190			
191			
192			
193			
194			
195			
196			
197			
198			
199			
200			
201			
202			
203			
204			
205			
206			
207			
208			
209			
210			
211			
212			
213			
214			
215			
216			
217			
218			
219			
220			
221			
222			
223			
224			
225			
226			
227			
228			
229			
230			
231			
232			
233			
234			
235			
236			
237			
238			
239			
240			
241			
242			
243			
244			
245			
246			
247			
248			
249			
250			
251			
252			
253			
254			
255			
256			
257			
258			
259			
260			
261			
262			
263			
264			
265			
266			
267			
268			
269			
270			
271			
272			
273			
274			
275			
276			
277			
278			
279			
280			
281			
282			
283			
284			
285			
286			
287			
288			
289			
290			
291			
292			
293			
294			
295			
296			
297			
298			
299			
300			
301			
302			
303			
304			
305			
306			
307			
308			
309			
310			
311			
312			
313			
314			
315			
316			
317			
318			
319			
320			
321			
322			
323			
324			
325			
326			
327			
328			
329			
330			
331			
332			
333			
334			
335			
336			
337			
338			
339			
340			
341			
342			
343			
344			
345			
346			
347			
348			
349			
350			
351			
352			
353			
354			
355			
356			
357			
358			
359			
360			
361			
362			
363			
364			
365			
366			
367			
368			
369			
370			
371			
372			
373			
374			
375			
376			
377			
378			
379			
380			
381			
382			
383			
384			
385			
386			
387			
388			
389			
390			
391			
392			
393			
394			
395			
396			
397			
398			
399			
400			
401			
402			
403			
404			
405			
406			
407			
408			
409			
410			
411			
412			
413			
414			
415			
416			
417			
418			
419			
420			
421			
422			
423			
424			
425			
426			
427			
428			
429			
430			
431			
432			
433			
434			
435			
436			
437			
438			
439			
440			
441			
442			
443			
444			
445			
446			
447			
448			
449			
450			
451			
452			


```

1 GET / HTTP/1.1
2 Host: www.tagged.com
3 Accept-Encoding: gzip, deflate, br
4 Accept:
  text/html,application/xhtml+xml,application/x
  ml;q=0.9,image/avif,image/webp,image/apng,*/
  ;q=0.8,application/signed-exchange;v=b3;q=0.7
5 Accept-Language: en-US;q=0.9,en;q=0.8
6 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac
  OS X 10_15_7) AppleWebKit/537.36 (KHTML,
  like Gecko) Chrome/135.0.0.0 Safari/537.36
7 Connection: close
8 Cache-Control: max-age=0
9 Upgrade-Insecure-Requests: 1
10 Sec-CH-UA: "Google Chrome";v="135",
  "Not=A7Brand";v="8", "Chromium";v="135"
11 Sec-CH-UA-Platform: "macOS"
12 Sec-CH-UA-Mobile: ?0
13 Origin: http://evil.com
14
15
1 HTTP/1.1 200 OK
2 Date: Wed, 16 Apr 2025 13:28:28 GMT
3 Server: Apache
4 Set-Cookie: S=pu5tv5ukcs5lkegl8u467cu1sd;
  path=/; domain=.tagged.com; secure; HttpOnly
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache,
  must-revalidate
7 Pragma: no-cache
8 Set-Cookie: B=b-B9F4E129F93B7025;
  expires=Sat, 14-Apr-2035 13:28:28 GMT;
  Max-Age=315360000; path=/;
  domain=.tagged.com; secure; HttpOnly
9 Vary: Accept-Encoding
10 Access-Control-Allow-Credentials: true
11 Access-Control-Allow-Headers: *
12 Content-Length: 42272
13 Content-Type: text/html; charset=UTF-8
14 Connection: close
15
16 <!DOCTYPE html>
17 <html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:fb="http://www.facebook.com/2008/fbml"
  >
18 <head id="html_head">

```

Figure 15: Request with Origin `http://evil.com` Response with HTTP 200 OK with cookies and CORS headers (`Access-Control-Allow-Credentials: true`), confirming CORS misconfiguration.

```

GET /secure_login.html HTTP/1.1
Host: www.tagged.com
Accept-Encoding: gzip, deflate, br
Accept:
  text/html,application/xhtml+xml,application/x
  ml;q=0.9,image/avif,image/webp,image/apng,*/
  ;q=0.8,application/signed-exchange;v=b3;q=0.7
5 Accept-Language: en-US;q=0.9,en;q=0.8
6 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac
  OS X 10_15_7) AppleWebKit/537.36 (KHTML,
  like Gecko) Chrome/135.0.0.0 Safari/537.36
7 Connection: close
8 Cache-Control: max-age=0
9 Upgrade-Insecure-Requests: 1
10 Sec-CH-UA: "Google Chrome";v="135",
  "Not=A7Brand";v="8", "Chromium";v="135"
11 Sec-CH-UA-Platform: "macOS"
12 Sec-CH-UA-Mobile: ?0
13 Origin: http://evil.com
14
15
1 HTTP/1.1 200 OK
2 Date: Wed, 16 Apr 2025 13:28:43 GMT
3 Server: Apache
4 Set-Cookie: S=3eq0gsv39mvsslj4raq1p5fg96;
  path=/; domain=.tagged.com; secure; HttpOnly
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache,
  must-revalidate
7 Pragma: no-cache
8 Set-Cookie: B=b-FE83A1E6AFE9AD39;
  expires=Sat, 14-Apr-2035 13:28:43 GMT;
  Max-Age=315360000; path=/;
  domain=.tagged.com; secure; HttpOnly
9 Vary: Accept-Encoding
10 Access-Control-Allow-Credentials: true
11 Access-Control-Allow-Headers: *
12 Content-Length: 2081
13 Content-Type: text/html; charset=UTF-8
14 Connection: close
15
16 <!DOCTYPE HTML>
17 <html xmlns="http://www.w3.org/1999/xhtml">
18 <head>

```

Figure 16: Request with Origin `http://evil.com` received a 200 OK response with auth cookies and insecure CORS headers, confirming a CORS vulnerability on the login end-point.

```

1 GET /online-dating/neuchatel/chat HTTP/1.1
2 Host: mingle2.com
3 Accept-Encoding: gzip, deflate, br
4 Accept:
  text/html,application/xhtml+xml,application/x
  ml;q=0.9,image/avif,image/webp,image/apng,*/*
  ;q=0.8,application/signed-exchange;v=b3;q=0.7
5 Accept-Language: en-US;q=0.9,en;q=0.8
6 User-Agent: Mozilla/5.0 (Windows NT 10.0;
  Win64; x64) AppleWebKit/537.36 (KHTML, like
  Gecko) Chrome/120.0.0.0 Safari/537.36
7 Connection: close
8 Cookie: _session_id=
  68a55f90ede06cec02b1557408951da9
9
10
1 HTTP/2 200 OK
2 Date: Sun, 20 Apr 2025 21:16:22 GMT
3 Content-Type: text/html; charset=utf-8
4 Vary: Accept-Encoding
5 X-Frame-Options: SAMEORIGIN
6 X-Frame-Options: DENY
7 X-Xss-Protection: 1; mode=block
8 X-Xss-Protection: 1; mode=block
9 X-Content-Type-Options: nosniff
10 X-Content-Type-Options: nosniff
11 Cache-Control: max-age=0, private,
  must-revalidate
12 Set-Cookie: tracker=
  id%3D%3E%7Cuser_id%3D%3E%7Ccp%3D%3E%7Cs1%3D%
  3E%7Cs2%3D%3E%7Ccr%3D%3E%7Clp%3D%3Ehttp%3A%2
  F%2Fmingle2.com%2Fonline-dating%2Fneuchatel%
  2Fchat%7Creferring_url%3D%3E%7Cinitial_click
  _at%3D%3E2025-04-20+14%3A16%3A22+-0700%7Csub
  scribed_at%3D%3E%7Cinternal_source%3D%3E%7Ck
  w%3D%3E%7Cmt%3D%3E%7Cactual_kw%3D%3E%7Csite%
  3D%3E%7Csearch_engine%3D%3E%7Csource_domain%
  3D%3E%7Clp_category%3D%3E%7Cgeog%7Csubcategor
  y%3D%3E%7Cregistration_site_id%3D%3E%7C
  7Cdevice%3D%3E; path=/; expires=Mon, 20 Apr
  2026 21:16:22 -0000
13 Set-Cookie: cleared_gta_version_1=true;
  path=/

```

Figure 17: session cookies were accepted over unencrypted HTTP.

Request

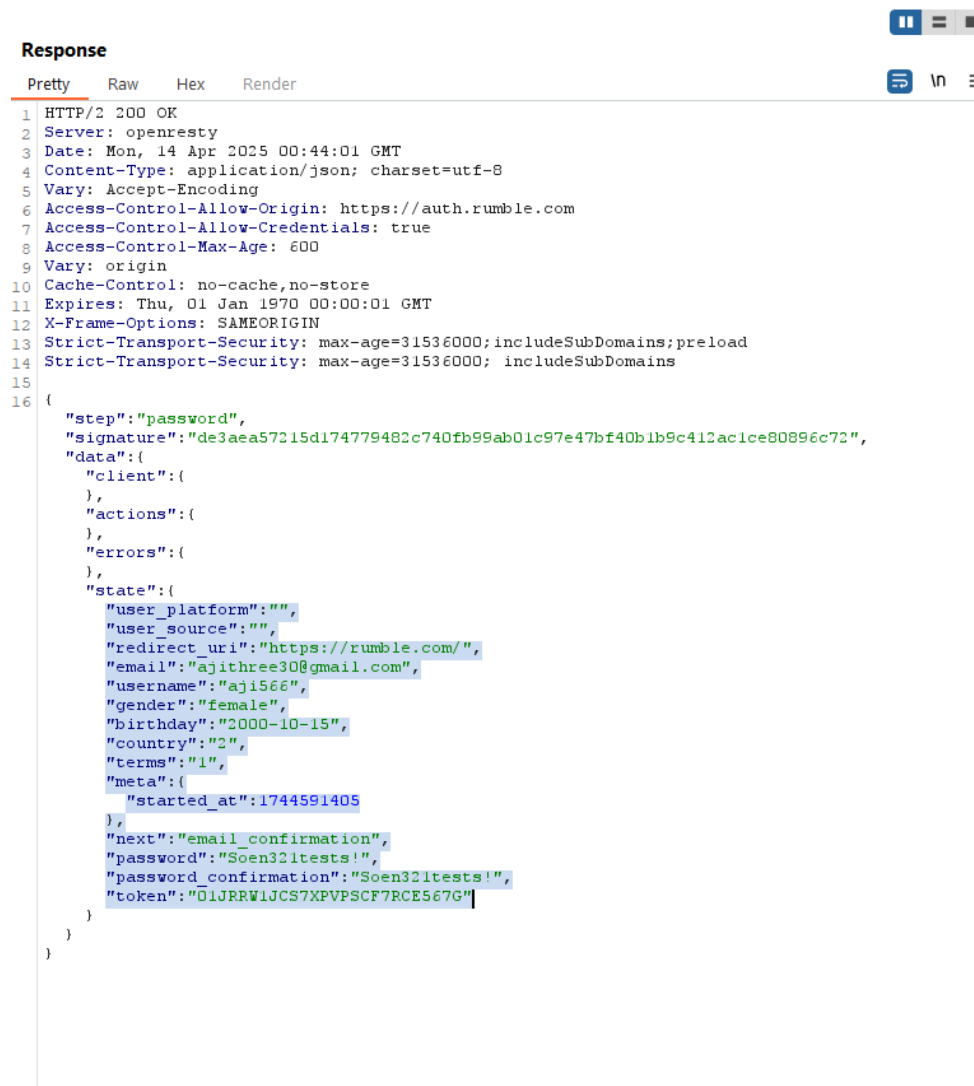
Pretty Raw Hex

```

1 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
  Gecko) Chrome/135.0.0.0 Safari/537.36
2 Accept: */*
3 Origin: https://auth.rumble.com
4 Sec-Fetch-Site: same-site
5 Sec-Fetch-Mode: cors
6 Sec-Fetch-Dest: empty
7 Referer: https://auth.rumble.com/
8 Accept-Encoding: gzip, deflate, br
9 Priority: u=1, i
10
11 {
12   "step": "password",
13   "signature": "88a6efefdef65c004b1ef289d27dc3b880dc725d9b2ab9e1184ebd4271edef94",
14   "data": {
15     "client": {
16       "email": "ajithree30@gmail.com",
17       "username": "aj1566",
18       "password": "Soen321tests!",
19       "password_confirmation": "Soen321tests!",
20       "turnstile_token":
21         "0.5MXJ9H0v7rQ1WgSiMjFg1WLi25Vny2UB5pwRFUlpG_Bctpv6MHNASZKhFQvhl7Pj0eSwwOEQY1hSoyOg
22         d1NyiQeEfArbRPa16tTAQx37DBSsv19pGshZN-HICROb9K3YJ82LHxef3A1zcdUYJNoQE6b22tHxRuofVn
23         XxV3Meo0I5-xbFpK57VckLtzdUSBFKnokh3L7U82IarTgdFoFbTGDGc6B1rbw6pw4ssClu3Ms105ajK_sYPM
24         ebrw6W6HcyOGPAZfrXoF9vCgu2ya_GloiqLzd7XkLCY76ZvaWd3pJb_qwmEcDnvd3HGFQ6uG8nAnRiaTaE
25         IXuIfGR902wY2_hflyZQHU7wd5t50-PzjSjzTAJ2_GLUu2aLNG2681FNLStfGvzeWjY2X9eU2t0OSI4nCeuh
26         QFI7lpVagvDD3nLbrJ-4Dxmhecln7d2fNc7n0x0Eyw33xdL_nR-T-8Y_SXfIn9pXC-fDDORP22BqGQC128
27         RgKYhaTwUEB-1ehKVu5utMWMhb_ehUuQRvpTR6IQXqL1hRA76sacgFc5S1l1ZH8pQTYxigD0vTqTY_2pKpoV
28         TCxORFqhzlyYNoKYoOEutPRUdqzQAiN3g-6FELgCptQK2Gz3Sgcn58FvDStU6J6Kxfn59gkmmcMyHe8Y
29         KFNZAXTEUs2IOqn0dCd5NE303MDvT2sMeQHAm1R6cWtK51DcoMA2K503N1g74CFDIRyDzBf8XzL3Fsa1AE
30         NVy_xHORZWWGecpFDINg2kvaKgGKmqomAd19e1PlDg6W5MINagBsergo82hzjGVVAdqdgSWiD22nWGJ4wTfP
31         oKk8iFEnyzsaEgGbl1lbaaEuzKA.PS9qt23xmx1zadSg9_JLbw.a5f62ebf57b0b1fe6cfcd3ad562b568fc
32         fd62793dcb8ae5946adf864af4f339b"
33     },
34     "actions": {
35     },
36     "errors": {
37     },
38     "state": {
39       "user_platform": "",
40       "user_source": "",
41       "redirect_uri": "https://rumble.com/",
42       "email": "ajithree30@gmail.com",
43       "username": "aj1566",
44       "gender": "female",
45       "birthday": "2000-10-15",
46       "country": "2",
47       "terms": "1",
48       "meta": {
49         "started_at": 1744591405
50       }
51     },
52     "next": "password"
53   }
54 }

```

Figure 18: Sending Sign Up Request on rumble.com



Response

Pretty Raw Hex Render

```
1 HTTP/2 200 OK
2 Server: openresty
3 Date: Mon, 14 Apr 2025 00:44:01 GMT
4 Content-Type: application/json; charset=utf-8
5 Vary: Accept-Encoding
6 Access-Control-Allow-Origin: https://auth.rumble.com
7 Access-Control-Allow-Credentials: true
8 Access-Control-Max-Age: 600
9 Vary: origin
10 Cache-Control: no-cache,no-store
11 Expires: Thu, 01 Jan 1970 00:00:01 GMT
12 X-Frame-Options: SAMEORIGIN
13 Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
14 Strict-Transport-Security: max-age=31536000; includeSubDomains
15
16 {
  "step": "password",
  "signature": "de3aea57215d174779482c740fb99ab01c97e47bf40b1b9c412ac1ce80896c72",
  "data": {
    "client": {
    },
    "actions": {
    },
    "errors": {
    },
    "state": {
      "user_platform": "",
      "user_source": "",
      "redirect_uri": "https://rumble.com/",
      "email": "ajithree30@gmail.com",
      "username": "aji566",
      "gender": "female",
      "birthday": "2000-10-15",
      "country": "2",
      "terms": "1",
      "meta": {
        "started_at": 1744591405
      }
    },
    "next": "email_confirmation",
    "password": "Soen321tests!",
    "password_confirmation": "Soen321tests!",
    "token": "01JRRW1JCS7XPVPSCF7RCE567G"
  }
}
```

Figure 19: Response with email and password echoed on rumble.com

```
Request
Pretty Raw Hex
1 POST /auth/sign_in HTTP/2
2 Host: mastodon.social
3 Cookie: _mastodon_session=
v8dgRHb6TOq6k%2F42oK3XqfbxpUSs%2F17tsMS5xQAkXD2fk9Pvp1MX%2Fu5VU1JyG4RLmmYPVBMEbj%2BW3MmKmZq
Fvndnk1f7gVMSfLoak9Bs7Bj%2Bct3YLb6SSmFjNG5wKIGqQOPS84PNJN1M%2BEyCTzXCM2vP1BCDhhzU1VFIEh9unz
kMenRxeaOhT5DDbrvSLScqG%2FSjpLdfDcPa7QBH47aVoqG3jqggOerxAoqBe9CKqXB3yDBknvQF9WRx7OmYS9sT4Pc
AP4gBpLPLw7Kk%2FOg79ggIEHk2CDraFu07K1N%2FW3CAJ2mxs17o324X3KBItgIFJBnl11A8mwjzLMjbCQ%3D%3D--
BQEp8GuLxoTOuGSf--BvrMnuqGYxqb2ozHpbQeTg%3D%3D
4 Content-Length: 183
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Chromium";v="135", "Not-A.Brand";v="8"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "Windows"
9 Accept-Language: en-GB,en;q=0.9
10 Origin: https://mastodon.social
11 Content-Type: application/x-www-form-urlencoded
12 Upgrade-Insecure-Requests: 1
13 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/135.0.0.0 Safari/537.36
14 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;
q=0.8,application/signed-exchange;v=b3;q=0.7
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-Mode: navigate
17 Sec-Fetch-User: ?1
18 Sec-Fetch-Dest: document
19 Referer: https://mastodon.social/auth/sign_in
20 Accept-Encoding: gzip, deflate, br
21 Priority: u=0, i
22
23 authenticity_token=
d714oHMj8IA7_4YOOKSRQW3L99zsJQBrixHkjV-ZBUbco-Dd8yzX3_P3h2Qx4hkgaPC9LjBX1FapOajKu_Uvg&
user%5Bemail%5D=a%40gmail.com&user%5Bpassword%5D=Soen321test&button=
```

Figure 20: Login Request in mastodon.social

```

Response
Pretty Raw Hex Render
1 HTTP/2 302 Found
2 X-Frame-Options: DENY
3 Accept-Ranges: bytes
4 X-Xss-Protection: 0
5 Cache-Control: private, no-store
6 Content-Type: text/html; charset=utf-8
7 Content-Security-Policy: base-uri 'none'; default-src 'none'; frame-ancestors 'none';
font-src 'self' https://mastodon.social; img-src 'self' data: blob: https://mastodon.social
https://files.mastodon.social; style-src 'self' https://mastodon.social
'nonce-GTlbFxtah/nJPT6c2iF8uw=='; media-src 'self' data: https://mastodon.social
https://files.mastodon.social; manifest-src 'self' https://mastodon.social; form-action
'self'; child-src 'self' blob: https://mastodon.social; worker-src 'self' blob:
https://mastodon.social; connect-src 'self' data: blob: https://mastodon.social
https://files.mastodon.social wss://streaming.mastodon.social; script-src 'self'
https://mastodon.social 'wasm-unsafe-eval'; frame-src 'self' https:
8 Referrer-Policy: same-origin
9 X-Content-Type-Options: nosniff
10 X-Runtime: 0.181949
11 X-Request-Id: f62bcb188ddf263cb832be4f2828c6aa6
12 Via: 1.1 varnish, 1.1 varnish, 1.1 varnish
13 Location: https://mastodon.social/
14 Set-Cookie: _session_id=
eyJfcjFmFpbHMlOmsibWVzc2FnZSI6IktTm1PROV6TnpVeU5UTXhaalU1TldJek4yTTJORiUyTldOa0lqUmpOVUySWE
9PSIsImV4CjE6IjIwMjYtMDQ0Lm1jFUMjI6MTg6NTMuMjc0WiIsImBldiI6ImNvb2tpZS5fc2Vzc2lwb19pZCJ9fQ43D4
3D--7226e224d8a647f846b587a29b049b1c6210bce6; path=/; expires=Tue, 21 Apr 2026 22:18:53
GMT; HttpOnly; SameSite=Lax; secure
15 Set-Cookie: _mastodon_session=
3BJsgJWKKQSDelgZyifAGQUNi0zVx42F42BWEUC42Bmid42B1bDXF93LcRqUBTh0CcY40Dp3wokBjeLBCFII42FRK7r
GyLHv8WqOc1P0JRv3MMHQ23MyOrIckAT98EdWgHyu7ym9H7JazQBR42FjXMvBX1dQGCC2X2xxTr6G9qvCTwODSELOL9
42F442BGeDOSa3izPgMgJUcp1KFFk6yQC6UbrBrCv42Fp6Fv1kXGNLoU42BrPvsc42BaYsrOwoFvmQXC42Fed01YBbA
iDzvFgR47Qnycz2KATHzfwSQ91DkzuT9jYR8BLmQkZA0bV42Byg7R--dPgKZ77WNPrWEUC9--w7ATz4zmwqHQtWUNWn
BRdA43D43D; path=/; HttpOnly; SameSite=Lax; secure
16 Date: Mon, 21 Apr 2025 22:18:53 GMT
17 X-Served-By: cache-fra-etou8220107-FRA, cache-fra-etou8220050-FRA, cache-yul1970056-YUL
18 X-Cache: MISS, MISS, MISS
19 X-Cache-Hits: 0, 0, 0
20 X-Timer: S1745273933.061757,V50,V295
21 Strict-Transport-Security: max-age=31557600
22 Alt-Svc: h3=":443";ma=86400,h3-29=":443";ma=86400,h3-27=":443";ma=86400
23 Content-Length: 0
24
25

```

Figure 21: Login Response in mastodon.social

request:

```

1 GET /forum.php?id=$placeholder$ HTTP/1.1
2 Host: mengchenghui.com
3 Accept-Encoding: gzip, deflate, br
4 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
5 Accept-Language: en-US;q=0.9,en;q=0.8
6 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
7 Connection: close
8 Cookie: security_session_verify=64fe42185e34312eb2e39a1bc98ff861
9 Referer: https://mengchenghui.com/
10
11

```

response:

0	["<script>alert(1)</script>"]	200	text/html; charset=utf-8
1	**	200	text/html; charset=utf-8
2	*<script>alert(1)</script>*	200	text/html; charset=utf-8
3	*<script>v20>alert(1)</script>*	200	text/html; charset=utf-8
4	*<script>src=datz*	200	text/html; charset=utf-8
5	alert(1);*	200	text/html; charset=utf-8
6	**	200	text/html; charset=utf-8

Figure 22: Testing XSS payloads exploiting jQuery.load() in version 1.8.3 using script tags with whitespace, revealing sanitization failure (200 OK response).

D Python plugin

```
[www.instagram.com] [Authorization] Potential user-specific identifier found in URL/body: https://www.instagram.com/443/api/graphql
[www.google.com] + Review this request for IDOR vulnerability potential.
[www.google.com] [Authorization] Potential user-specific identifier found in URL/body: https://www.google.com/443/ccm/collect?en-page_view&dl=https%3A%2F%2Fbadoo.com%2F&srcsrc=www.googletagmanager.com&frm=0&rnd=15071169023.1744616975&st=b
[www.google.com] + Review this request for IDOR vulnerability potential.
[www.reddit.com] [Session Management] Insecure cookie detected: Set-Cookie: loid=000000001na739ume2.1744616976545;20F8QJF8Qm5F0R8UXJpck1l@haeflyawta18tUXKFHINAVU1T0UWmmtvWFZ0dTFrdJESUJ1Tuhza112R1QK2M3Sh0MkMkKzcfpkckUhpCLU5UW
[www.reddit.com] [Session Management] Insecure cookie detected: Set-Cookie: session_tracker=mg3jpmad11nahrbor.0.1744616976548.20F8QJF8Qm5F0R8UXJpck1l@haeflyawta18tUXKFHINAVU1T0UWmmtvWFZ0dTFrdJESUJ1Tuhza112R1QK2M3Sh0MkMkKzcfpkckUhpCLU5UW
[www.reddit.com] [Session Management] Insecure cookie detected: Set-Cookie: csrf_token=Sa383f3574c8e6c8d487608ca24d16; path=/; domain=.reddit.com; samesite=strict; secure
[www.reddit.com] [Session Management] Insecure cookie detected: Set-Cookie: theme=; path=/; expires=Thu, 01 Jan 1970 00:00:00 GMT; domain=.reddit.com; secure
[www.reddit.com] [Session Management] Insecure cookie detected: Set-Cookie: cqv=2; Max-Age=3072000; Domain=.reddit.com; Path=/; Secure; SameSite=None
[o1118521.ingest.us.sentry.io] [Authorization] Potential user-specific identifier found in URL/body: https://o1118521.ingest.us.sentry.io/4507/775599378432/envelope/?sentry_version=7&sentry_key=bd5559e88480ec92a1769a01943cd8a&en
[o1118521.ingest.us.sentry.io] + Review this request for IDOR vulnerability potential.
[www.google.com] [Transport Security] Missing HSTS header
[www.instagram.com] [Session Management] Insecure cookie detected: Set-Cookie: _js_dtr=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=-1744616976; path=/; domain=.instagram.com; httponly
[www.instagram.com] [Session Management] Insecure cookie detected: Set-Cookie: _js_ig_did=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=-1744616976; path=/; domain=.instagram.com; httponly
[ca.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: JSESSIONID=ajax:092842415819932255; Domain=.ca.linkedin.com; Path=/; Secure; SameSite=None
[ca.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: csrf_token=978c156a02d180f84982c21503e366; path=/; expires=Tue, 14 Apr 2026 07:49:37 GMT; samesite=lax; secure
[ca.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: bcookie=vv2834d4d486-f729-d802-80a2-6591143b79b0; Domain=.linkedin.com; Expires=Tue, 14-Apr-2026 07:49:37 GMT; Path=/; Secure; SameSite=None
[ca.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: li_gc=HtSw0E3HQDQ2M7Y5MzY7Mj5uYjZT5z5AurN64P6YnpG8IxdM8Qbx+94dXureQXxu9Aa; Domain=.linkedin.com; Expires=Sat, 11 Oct 2025 07:49:36 GMT; Path=/; Secure; Sam
[ca.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: lldc=b0G5T05:s=0:r=0:a=0:p=0:g=3358:u=1:x=1:1-1744616977:t=1744703377:v=2:sig=AQE0z0kcn-HQ2TEv0ck7qQ0E51CR7n; Expires=Tue, 15 Apr 2025 07:49:37 GMT; domain
[badoo.com] [Transport Security] Missing HSTS header
[badoo.com] [Session Management] Insecure cookie detected: Set-Cookie: HDR-X-User-Id; path=/; domain=.badoo.com; Secure;
[badoo.com] [Session Management] Insecure cookie detected: Set-Cookie: session_cookie_name=session; path=/; domain=.badoo.com; Secure; Expires=Tue, 14-Apr-26 07:49:37 GMT; Max-Age=31536000;
[ca.pinterest.com] [Session Management] Insecure cookie detected: Set-Cookie: _routing_id=70c575f2e-b6ab-4707-beb8-c8b5b08f657; Max-Age=86400; Path=/; HttpOnly
[ca.pinterest.com] [Session Management] Insecure cookie detected: Set-Cookie: _js_dtr=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=-1744616976; path=/; domain=.instagram.com; httponly
[www.instagram.com] [Session Management] Insecure cookie detected: Set-Cookie: _js_ig_did=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=-1744616976; path=/; domain=.instagram.com; httponly
[www.instagram.com] [Session Management] Insecure cookie detected: Set-Cookie: _js_dtr=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=-1744616976; path=/; domain=.instagram.com; httponly
[www.instagram.com] [Session Management] Insecure cookie detected: Set-Cookie: _js_ig_did=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=-1744616976; path=/; domain=.instagram.com; httponly
[www.instagram.com] [Session Management] Insecure cookie detected: Set-Cookie: _js_dtr=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=-1744616976; path=/; domain=.instagram.com; httponly
[www.instagram.com] [Session Management] Insecure cookie detected: Set-Cookie: _js_ig_did=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=-1744616976; path=/; domain=.instagram.com; httponly
[www.instagram.com] [Session Management] Insecure cookie detected: Set-Cookie: _js_dtr=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=-1744616976; path=/; domain=.instagram.com; httponly
[badoo.com] [Session Management] Insecure cookie detected: Set-Cookie: device_id=bcl59788-9788-88fc-rcfs-f5401cbf5381; expires=Wed, 21 Mar 2125 07:49:37 GMT; Max-Age=3153600000; path=/; domain=.badoo.com
```

Figure 23: Output from the Burp Suite custom plugin identifying missing HSTS headers, insecure cookies, and potential IDOR vectors during dynamic analysis of multiple web-sites.

```
[www.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: lang=28lang=en-us; SameSite=None; Path=/; Domain=linkedin.com; Secure
[www.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: lldc=b0G5T05:s=0:r=0:a=0:p=0:g=3358:u=1:x=1:1-1744616975:t=1744703375:v=2:sig=AQHmKfErs-nd4qGHE4X3wPZQ01GuH20; Expires=Tue, 15 Apr 2025 07:49:35 GMT; domai
[www.facebook.com] [Authorization] Potential user-specific identifier found in URL/body: https://www.facebook.com/443/x/oauth/status?client_id=2742660671648&input_token=origin=1&redirect_uri=https%3A%2F%2Fca.pinterest.com%2F%23featured-bo
[www.facebook.com] + Review this request for IDOR vulnerability potential.
[www.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: lang=28lang=en-us; SameSite=None; Path=/; Domain=linkedin.com; Secure
[www.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: lldc=b0G5T05:s=0:r=0:a=0:p=0:g=3358:u=1:x=1:1-1744616975:t=1744703375:v=2:sig=AQHmKfErs-nd4qGHE4X3wPZQ01GuH20; Expires=Tue, 15 Apr 2025 07:49:35 GMT; domai
[www.facebook.com] [Authorization] Potential user-specific identifier found in URL/body: https://www.facebook.com/443/x/oauth/status?client_id=2742660671648&input_token=origin=1&redirect_uri=https%3A%2F%2Fca.pinterest.com%2F%23featured-bo
[www.facebook.com] + Review this request for IDOR vulnerability potential.
[www.instagram.com] [Session Management] Insecure cookie detected: Set-Cookie: csrf_token=x1VCBRVQunklw6-bpa7cy; expires=Mon, 13-Apr-2026 07:49:35 GMT; Max-Age=31449600; path=/; domain=.instagram.com; secure; SameSite=None
[www.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: lang=28lang=en-us; SameSite=None; Path=/; Domain=linkedin.com; Secure
[www.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: lldc=b0G5T05:s=0:r=0:a=0:p=0:g=3358:u=1:x=1:1-1744616975:t=1744703375:v=2:sig=AQHmKfErs-nd4qGHE4X3wPZQ01GuH20; Expires=Tue, 15 Apr 2025 07:49:35 GMT; domai
[www.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: lang=28lang=en-us; SameSite=None; Path=/; Domain=linkedin.com; Secure
[www.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: lldc=b0G5T05:s=0:r=0:a=0:p=0:g=3358:u=1:x=1:1-1744616975:t=1744703375:v=2:sig=AQHmKfErs-nd4qGHE4X3wPZQ01GuH20; Expires=Tue, 15 Apr 2025 07:49:35 GMT; domai
[www.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: lang=28lang=en-us; SameSite=None; Path=/; Domain=linkedin.com; Secure
[www.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: lldc=b0G5T05:s=0:r=0:a=0:p=0:g=3358:u=1:x=1:1-1744616975:t=1744703375:v=2:sig=AQHmKfErs-nd4qGHE4X3wPZQ01GuH20; Expires=Tue, 15 Apr 2025 07:49:35 GMT; domai
[google.com] [Transport Security] Missing HSTS header
[www.reddit.com] [Transport Security] Missing HSTS header
[accounts.google.com] [Transport Security] Missing HSTS header
[www.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: lang=28lang=en-us; SameSite=None; Path=/; Domain=linkedin.com; Secure
[www.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: lldc=b0G5T05:s=0:r=0:a=0:p=0:g=3358:u=1:x=1:1-1744616975:t=1744703375:v=2:sig=AQHmKfErs-nd4qGHE4X3wPZQ01GuH20; Expires=Tue, 15 Apr 2025 07:49:35 GMT; domai
[badoo.com] [Session Management] Insecure cookie detected: Set-Cookie: device_id=427e09c9-09c9-c958-582b-2b4811915337; expires=Wed, 21 Mar 2125 07:49:35 GMT; Max-Age=3153600000; path=/; domain=.badoo.com
[www.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: lang=28lang=en-us; SameSite=None; Path=/; Domain=linkedin.com; Secure
[www.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: lldc=b0G5T05:s=0:r=0:a=0:p=0:g=3358:u=1:x=1:1-1744616975:t=1744703375:v=2:sig=AQHmKfErs-nd4qGHE4X3wPZQ01GuH20; Expires=Tue, 15 Apr 2025 07:49:35 GMT; domai
[badoo.com] [Session Management] Insecure cookie detected: Set-Cookie: device_id=f02a732f-732f-2f4a-4a31-314408901531; expires=Wed, 21 Mar 2125 07:49:36 GMT; Max-Age=3153600000; path=/; domain=.badoo.com
[www.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: lang=28lang=en-us; SameSite=None; Path=/; Domain=linkedin.com; Secure
[www.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: lldc=b0G5T05:s=0:r=0:a=0:p=0:g=3358:u=1:x=1:1-1744616975:t=1744703375:v=2:sig=AQFy10507YehICr3sPu06jBFcRjDQ004; Expires=Tue, 15 Apr 2025 07:49:36 GMT; domai
[eul.badoo.com] [Transport Security] Missing HSTS header
[www.google.com] [Transport Security] Missing HSTS header
[www.reddit.com] [Session Management] Insecure cookie detected: Set-Cookie: rdt=4e326c448a141391a4a611925081e4; path=/; domain=.reddit.com; samesite=none; secure;
[www.reddit.com] [Session Management] Insecure cookie detected: Set-Cookie: edgebucket=Mo8Wvee3jHfDfaw9; Domain=.reddit.com; Max-Age=63071999; Path=/; secure
[ca.linkedin.com] [Transport Security] Missing HSTS header
[ca.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: bcookie=vv28d3b866b8-749-4c00-80a0-91a4c4d07b2f; Domain=.linkedin.com; Expires=Tue, 14-Apr-2026 07:49:36 GMT; Path=/; Secure; SameSite=None
[ca.linkedin.com] [Session Management] Insecure cookie detected: Set-Cookie: li_gc=HtSw0E3HQDQ2M7Y5MzY7Mj5uYjZT5z5AurN64P6YnpG8IxdM8Qbx+94dXureQXxu9Aa; Domain=.linkedin.com; Expires=Sat, 11 Oct 2025 07:49:36 GMT; Path=/; Secure; Sam
[www.instagram.com] [Authorization] Potential user-specific identifier found in URL/body: https://www.instagram.com/443/ajax/qm?_a=18_user=0&_comet_req=78jazoest=2936
[www.instagram.com] + Review this request for IDOR vulnerability potential.
[consent.badoo.com] [Authorization] Potential user-specific identifier found in URL/body: https://consent.badoo.com/443/mms/v2/get_site_data?hasCsp=true&href=https%3A%2F%2Fbadoo.com%2F&account_id=1789
[consent.badoo.com] + Review this request for IDOR vulnerability potential.
[ca.pinterest.com] [Session Management] Insecure cookie detected: Set-Cookie: csrf_token=b48ec40473f21682arfcb315a613df376; path=/; expires=Tue, 14 Apr 2026 07:49:36 GMT; samesite=lax; secure
[ca.pinterest.com] [Session Management] Insecure cookie detected: Set-Cookie: _routing_id=97ccaae-4709-46a4-0952-0830708c22bc; Max-Age=86400; Path=/; HttpOnly
[consent.badoo.com] [Authorization] Potential user-specific identifier found in URL/body: https://consent.badoo.com/443/wrapper/v2/meta_data?hasCsp=true&account_id=1789&env=prod&meta_data=578X822gdp9K2X3AX8X22grouphIdX2K3AS89610K70X2KX2
[consent.badoo.com] + Review this request for IDOR vulnerability potential.
```

Figure 24: Consolidated plugin output highlighting session management flaws, transport security issues, and authorization weaknesses across various endpoints.