Prevention of
Passive Browser Fingerprinting

Bc. Jozef Hruška

Supervisor: Ing. Libor Polčák, Ph.D.
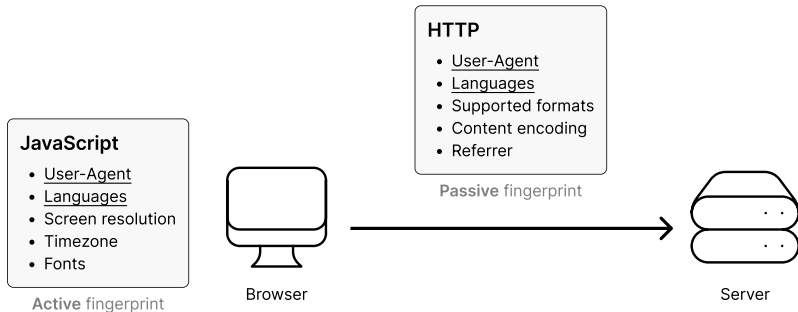
BRNO FACULTY
UNIVERSITY OF INFORMATION
OF TECHNOLOGY TECHNOLOGY

January 17, 2023

- Why should I care about fingerprinting?
- How does it affect me?
- What can I do to prevent it?

Buy a new Tesla Model 3 with 30% discount.

Ad www.example.com/tesla/discount

Praesent metus lectus, dapibus at diam sed, viverra maximus neque. Donec laoreet aliquet nibh a tempor.

Fingerprinting and inconsistencies

JavaScript
- User-Agent
- Languages
- Screen resolution
- Timezone
- Fonts

Active fingerprint

HTTP
- User-Agent
- Languages
- Supported formats
- Content encoding
- Referrer

Passive fingerprint

Browser

Server

- The fingerprinting application (Server) collects data from the device (Browser) actively and passively.
- Countermeasures alter data to confuse the fingerprinter.
- When changed improperly, the fingerprinter can detect inconsistencies.

Active fingerprinting protection (JShelter)

- JShelter is an anti-malware browser extension.
- Includes active fingerprinting countermeasures.
- Lacks passive fingerprinting countermeasures.

My master thesis aims to design and implement passive fingerprinting protection to complement the already existing protections of JShelter.

Minimizing user side effects

Description:

- Every modification made to HTTP headers could result in a different response from the server.
- The goal is to minimize these side effects, which could negatively affect the user experience.

Example:

- English is a prevalent language on the web.
- Let's change the preferred language to English to create a „homogenous" fingerprint.
- The site language changes to English, but the user doesn't understand English.

HTTP Header: User-Agent

> Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/**51.0.2704.103** Safari/537.36

> Mozilla/5.0 (iPhone; CPU iPhone OS **13_5_1** like Mac OS X) AppleWebKit/**605.1.15** (KHTML, like Gecko) Version/**13.1.1** Mobile/15E148 Safari/604.1

Transformation:

- Downgrade of version numbers in the User-Agent string.
- Two version numbers that differ in the last segment (PATCH) should have the same functionality (Semantic Versioning).
- Example: `AWK/605.1.15 → AWK/605.1.14`

Consistency:

- An identical 1:1 copy stored in `Navigator.userAgent`.

Accept: text/html, application/xhtml+xml, application/xml;q=0.9, **image/webp**, */*;q=0.8

Transformation:

- Replace newer formats with older alternatives.
- Would result in a performance hit, but not break the page.
- Example: `image/webp → image/png`

Consistency:

- It is not easy to detect inconsistencies as there is no interface listing all supported formats.
- The `Navigator.mimeTypes` interface that partially allowed this is deprecated.

HTTP Header: Accept-Language

> Accept-Language: **fr-CH**, fr;q=0.9, en;q=0.8, de;q=0.7, *;q=0.5

Transformation:

- Add, modify or remove language regions (subtags).
- Example:
    - `en → en-GB`
    - `fr-CH → fr-CA`
    - `de-AT → de`

Consistency:

- Preferred languages are also available through the `Navigator.languages` interface.
- The first language is accessible through the `Navigator.language` interface.

List of HTTP headers

Accept, Accept-Encoding, Accept-Language, Connection, Cookie, Host, User-Agent

Transformation:

- Add new headers to the HTTP request.
- Custom headers pose a low risk of a breaking event.

Consistency:

- Not available through the browser APIs.

Thank you for your attention.

Jozef Hruška