Renforcer la sécurité sur le web en disséquant les empreintes de navigateurs

Antoine Vastel - Université de Lille / INRIA

Qui suis-je ?

Doctorant en 3eme année (équipe SPIRALS INRIA Lille)

Empreintes de navigateurs

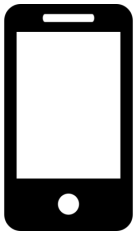
Vie privée et sécurité

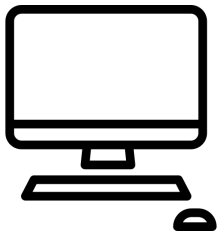
La diversité, la cause du browser fingerprinting

Diversité des appareils

Grand nombre d’appareils pouvant naviguer sur le web :

- Ordinateurs fixes
- Ordinateurs portables
- Smartphones
- Télévisions connectées





Diversité des configurations

Affichage de la barre des favoris

Niveau du zoom par défaut

Langues utilisées

Gestion de la diversité

Promesse des sites web → fonctionner sur tous les appareils ayant un navigateur respectant les standards

Différents écrans, langues, moyens d'interaction

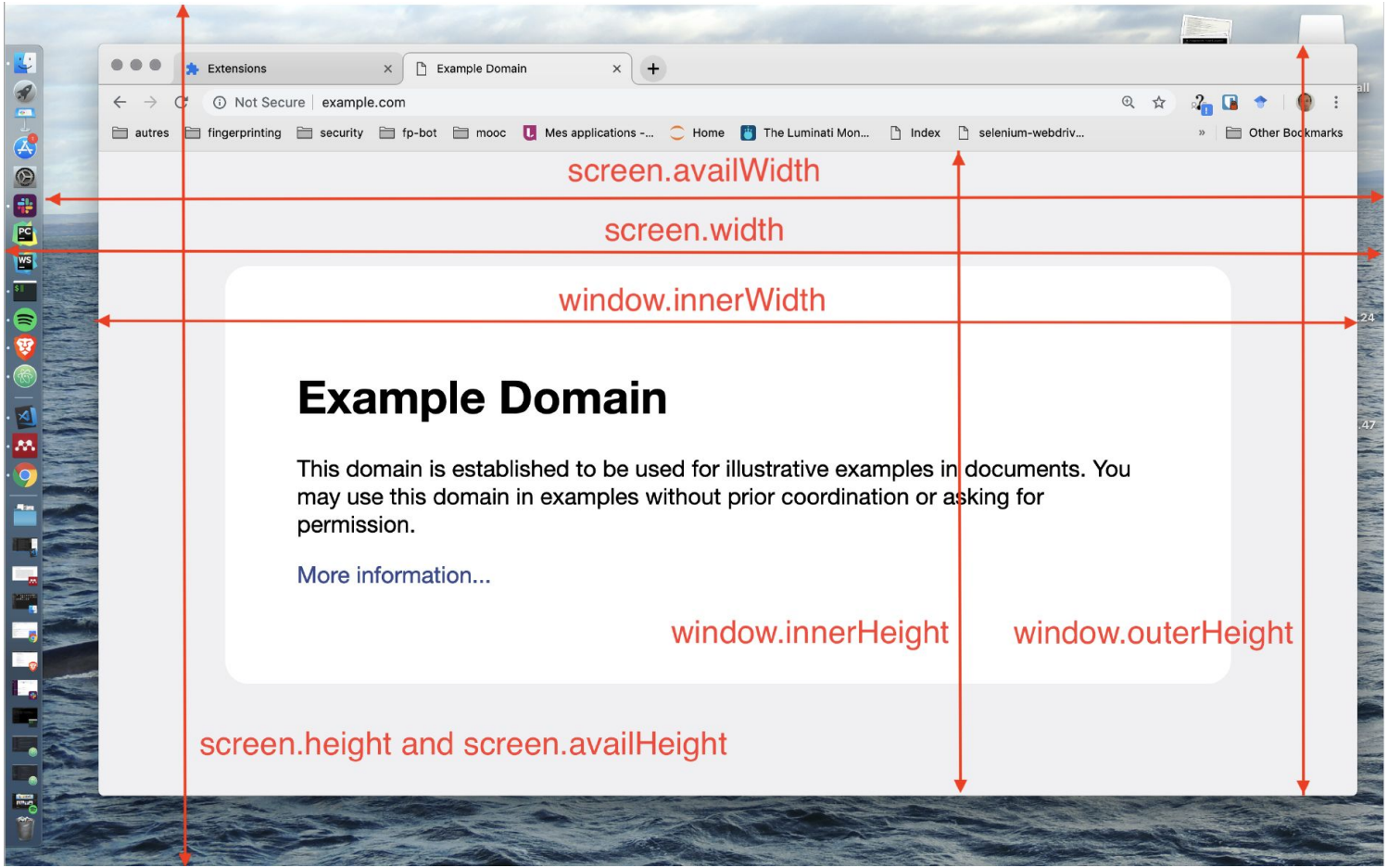
- Media queries (CSS)
- APIs JavaScript
- Headers HTTP

Qu’est ce qu’une empreinte de navigateur ?

Ensemble d’attributs fournissant des informations sur l’appareil / navigateur et sa configuration

Attributs accessibles sans autorisation

Depuis le navigateur (JavaScript, CSS, headers HTTP)


Extensions
Example Domain
Not Secure
example.com
autres
fingerprinting
security
fp-bot
mooc
Mes applications -...
Home
The Luminati Mon...
Index
selenium-webdriv...
Other Bookmarks
screen.availWidth
screen.width
window.innerWidth
Example Domain
This domain is established to be used for illustrative examples in documents. You may use this domain in examples without prior coordination or asking for permission.
More information...
window.innerHeight
window.outerHeight
screen.height and screen.availHeight

Exemple d’attributs

User Agent : Mozilla/5.0 (Macintosh; Intel **Mac OS X** 10_14_4) AppleWebKit/537.36 (KHTML, like Gecko) **Chrome/74**.0.3729.108 Safari/537.36

Langues : fr,en-GB,en-US,en

Mémoire : 8 Gb

Fuseau horaire : America/Tijuana

Exemple de canvas

!H71JCaj)]# 1@#

Hel$&?6%){mZ+#@

Cwm fjordbank glyphs vext quiz, 😀

Impact sur la vie privée

Empreintes **uniques et stables** (entre 40 % et 90%)

Traçage marketing

Utilisées en complément des cookies

Utilisation pour la sécurité

Pas stockée sur la machine → plus résilient qu'un cookie

Cas d'utilisation :

1. Éviter les votes multiples
2. Paywall pour les journaux
3. **Renforcer l'authentification**
4. **Détection de bots** (crawlers/fraude à la pub)

Empreintes de navigateurs pour renforcer l'authentification

Objectifs

1. Sécuriser un compte, même si nom d'utilisateur / mot de passe volés
2. Protéger contre les vols de session

Modèle de menace

Un attaquant connaissant **nom d'utilisateur et mot de passe**

Attaquant capable de **voler empreinte de la victime** (phishing)

Challenges :

- Empreinte **collectée côté client**
- Peut être **forgée/rejouée**

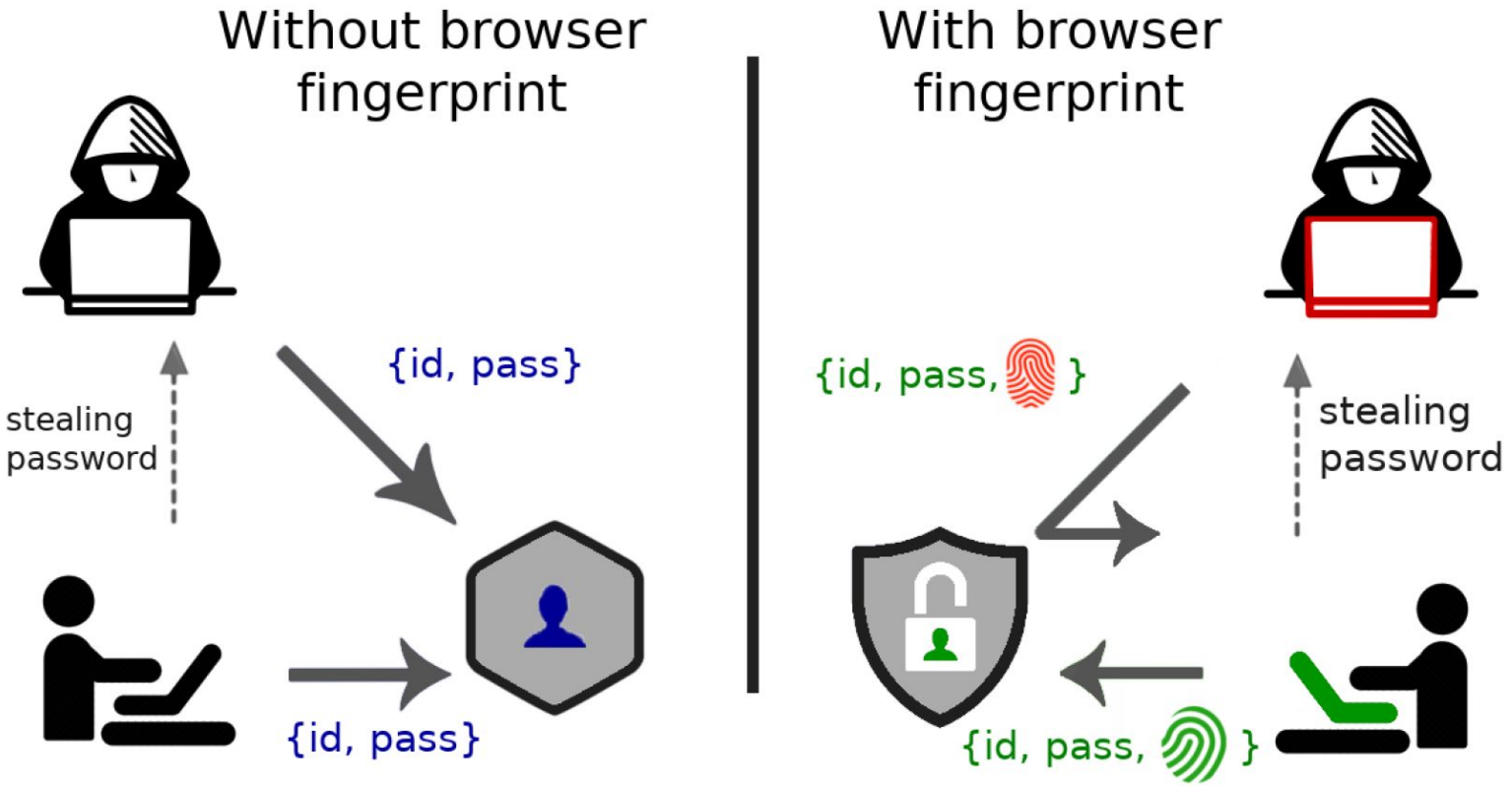


Schéma réalisé par Antonin Durey

Que mettre dans l'empreinte ?

Attributs :

- **Uniques** → évite les collisions (faux positifs)
- **Stables** → facilite la vérification (faux négatifs)
- **Difficiles à forger** → bloque attaques par rejeu (faux positifs)

Attributs dynamiques

Utilisation du **canvas** (*Bursztein 2016*, *Laperdrix 2019*) :

- Rendu imprévisible
- Rendu stable dans le temps

Générer des **challenges dynamiques** basés sur le rendu de canvas

- **Courbes** (Bézier, cubique, quadratique)
- **Texte** (couleur, emojis, ombre, gradients)

Bursztein, Elie, et al. "Picasso: Lightweight device class fingerprinting for web clients." Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices. ACM, 2016.

Laperdrix, Pierre, et al. "Morellian Analysis for Browsers: Making Web Authentication Stronger With Canvas Fingerprinting" International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, 2019.

Évaluation

Canvas collectés sur > 2M utilisateurs

Pas de collisions

Relativement stable (pour authentification) > 30 jours

Empreintes de navigateurs pour la détection de bots

Qu'est qu'un crawler ?

Programme informatique

Visualiser les pubs, vol de contenu, vol de compte

Clients HTTP : urllib (python), HTTP (NodeJS)

Vrais navigateurs + Selenium / Puppeteer / Chrome Devtools Protocol

Navigateurs headless : Chrome headless / Firefox headless

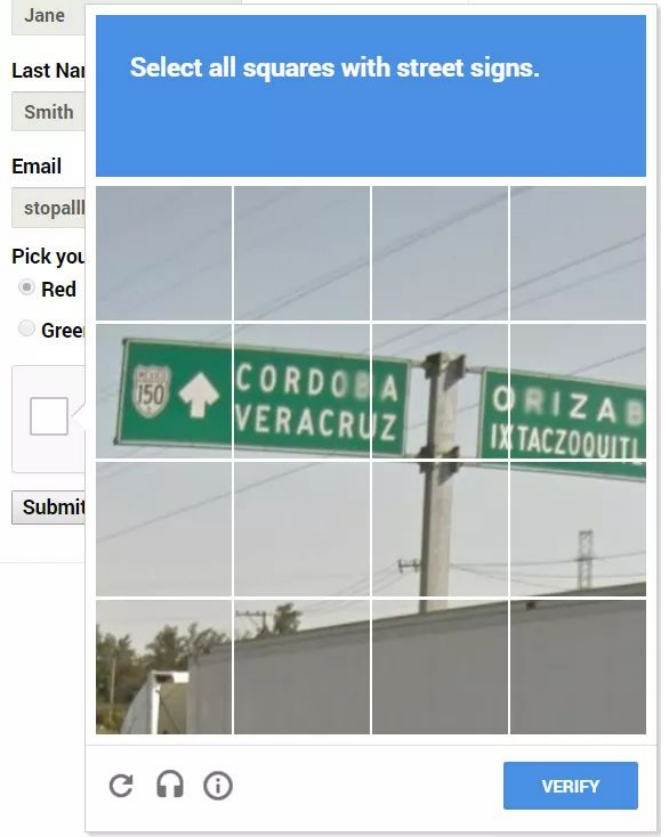
Approches existantes

CAPTCHAs (test de Turing) :

- Censé être **facile pour les humains**
- Censé être **difficile pour les robots**

Détection anomalies / analyse séries temporelles :

- Extrait attributs (nombre pages vues, taux erreurs, etc)
- Modélise navigation en série temporelle



Popularité sur le web

Utilisation sur le web

Crawl top **Alexa 10,000**

291 sites bloquent les crawlers (oracle pour l'évaluation)

93 (31.96%) des sites qui bloquent les crawlers utilisent du fingerprinting

Techniques de detection

1. Détection **d'attributs ajoutés par les crawlers**
2. Détection **d'incohérences**
3. Détection **empreintes headless**

Attributs ajoutés par les crawlers

window._selenium

window._phantom

navigator.webdriver

window.geb (Groovy)

Peu robuste mais simple, **aucun faux positifs**

Detection d'incohérences

Le développeur du crawlers **ment sur la nature du navigateur**

Exemple : Prétend être **Chrome version 72** plutôt que **Chrome Headless version 70**

Problème : incohérence entre le navigateur présent dans le user agent et le vrai navigateur

Detection d'incohérences : exemple

Présence/absence de fonctionnalités : certaines fonctionnalités sont liées à certains navigateurs (**msRequestAnimationFrame**, **webkitRequestAnimationFrame**)

Erreurs différentes (**toSource** seulement sur Firefox)

Signature des fonctions différentes : **eval.toString().length** :

- 37 (Safari, Firefox)
- 39 (Internet Explorer)
- 33 (Chrome)

Détection empreintes headless

Navigateurs headless similaires mais différents des non headless

Exemple : **gestion des notifications** dans Chrome headless

```
navigator.permissions.query({name:'notifications'})
    .then(function(permissionStatus) {
        if(Notification.permission === 'denied' &&
            permissionStatus.state === 'prompt') {
            console.log('This is Chrome headless')
        } else {
            console.log('This is not Chrome headless
                ')
        }
    });
```

Absence de header **HTTP Accept-Language**, **requête favicon**

Évaluation

7 crawlers graduellement plus difficiles à détecter (modification de leurs empreintes)

Visite de sites qui bloquent les crawlers

- 20 avec fingerprinting, 20 sans fingerprinting

Résultats

Sites avec fingerprinting détectent **plus de crawlers et plus rapidement**

Attention cependant :

- Les crawlers les plus avancés ne sont **pas détectés**
- Requiert **peu de lignes de code** (< 300 loc)

Conclusion

Utilisation pour le **tracking** / **renforcer la sécurité sur le web**

Authentification / détection de bots

Challenge car **empreinte collectée côté client**

Doit être **utilisé en plus d'autres techniques**