EVERYONE IS DIFFERENT:
Client-side Diversification for Defending Against Extension Fingerprinting

Erik Trickel, Oleksii Starov, Alexandros Kapravelos, Nick Nikiforakis, and Adam Doupé
Browser Extensions
Browser Extensions
Customization at a Cost

- Extension granted more privileges
- Cookieless identification across browsing sessions
- Inferences based on installed extensions
Customization at a Cost

https://github.com/prophittcorey/nefarious-linkedinst
How to prevent this?
CloakX

- Extension cloaking tool
- Static and dynamic analysis
- Client-side modification
  - Without modification to browser
  - Without requiring extension developers to modify their code
Extension Fingerprinting

- Extension fingerprinting is not intentionally supported but side-channels exist
- Web Accessible Resources (WARs) Fingerprinting
  - ACM CODASPY 2017
- DOM Fingerprinting (XHound)
  - Oakland 2017
WAR Fingerprinting

- WARs are uniquely identifiable resources that extensions deliberately expose to webpages
- WAR Fingerprints
  - 16,479 extensions
  - 50% of the top 1,000 extensions
DOM Fingerprinting

- XHound
  - Exercises extensions
  - Track DOM modifications to create fingerprint

- 5,323 extensions create a DOM fingerprint
Detection
Detection—Anchorprints

- An anchor is a unique identifier used by the extension and accessible to webpages
- WARs, IDs, class names, and custom attributes
- Save to Pocket adds

<svg class="pocketIconStroke_1zNwYwpH"...>
Detection—Structureprints

- Uses the structure of the changes an extension makes to a webpage
- Google calendar extension injects an `<a>` and an `<img>` each with specific attributes that no other extension adds
Detection—Behaviorprints

- Target an extension’s behavior
- Grammarly injects a green image into a textarea
Webpage Environment

- **Slide 15**

---

- **Title:** The Laboratory of Security Engineering for Future Computing
Extensions in Chrome

- **Extension Bundle**
  - **Manifest file**
    - `content_scripts:
      - js:
        - content_a.js
        - content_b.js
    - background:
      - scripts:
        - backgrnd_1.js
        - backgrnd_2.js
    - `web_accessible_resources:
      - common.js
      - img/*

- **JS Execution Env.**
  - `<script>`
  - DOM
  - HTML
Extensions in Chrome

![Diagram of Extension Bundle]

- **Manifest file**
  - content_scripts:
    - content_a.js
    - content_b.js
  - background:
    - scripts:
      - backgrnd_1.js
      - backgrnd_2.js
  - web_accessible_resources:
    - common.js
    - img/*

---

```html
<script src='chrome-extension://abcdefghijklmnopnrlkjihgfedcba/common.js'>
```
Extensions in Chrome

- **Extension Bundle**
  - Manifest file:
    - content_scripts:
      - content_a.js
      - content_b.js
    - background:
      - scripts:
        - backgrnd_1.js
        - backgrnd_2.js
    - web_accessible_resources:
      - common.js
      - img/*

- **Background Page**
  - JS Execution Env.
    - backgrnd_1.js
    - backgrnd_2.js

- **DOM**
  - HTML

- **JS Execution Env.**
  - <script>
    - Droplet
    - common.js
    - <style>

- `<script src='chrome-extension://abcdefghijklmnopmnlkjihgfedcba/common.js'>`
Extensions in Chrome

- **Manifest file**
  - `content_scripts: js: content_a.js content_b.js`
  - `background: scripts: backgrnd_1.js backgrnd_2.js`
  - `web_accessible_resources: common.js img/*`

- **Background Page**
  - JS Execution Env.
    - backgrnd_1.js
    - backgrnd_2.js

- **Content Scripts**
  - JS Execution Env.
    - content_a.js
    - content_b.js
    - Dynamic JS

- **DOM**
  - HTML

- **JS Execution Environments**
  - `<script>`
  - `<style>`
  - `common.js`

- `<script src='chrome-extension://abcdefghijklmnopmnlkjihgfedcba/common.js'>`
Extensions in Chrome

How it Works!

Extension Bundle

Manifest file
content_scripts:
- content_a.js
- content_b.js

background:
- scripts:
  - backgrnd_1.js
  - backgrnd_2.js

web_accessible_resources:
- common.js
- img/*

Background Page

JS Execution Env.
- backgrnd_1.js
- backgrnd_2.js

DOM

HTML

Content Scripts

JS Execution Env.
- content_a.js
- content_b.js
- Dynamic JS

DOM

HTML

<script src='chrome-extension://abcdefghijklmnopmnlkjihgfedcba/common.js'>
CloakX
Cloaking Extensions

- Renaming
  - WARs
  - IDs
  - Class names

- Random Insertion
  - Tags
  - IDs and custom attributes
Cloaking Process
Cloaking Process
Cloaking Process
Droplets

How it Works!

Content Scripts

JS Execution Env.

Droxy

content_a.js
content_b.js
Dynamic JS

DOM

HTML

<script>
Droplet
zpptvar.js
<style>

How it Works!
Cloaking Process

- **TAJS** – Type Analysis for JavaScript
  - Added taint analysis
  - Limiting changes to the use of ID and class names that interacted with DOM
- Rewrite IDs and class names inside droplets
Cloaking Process
Cloaking Extensions
Cloaking Extensions
Evaluation

- Functionality Experiments
  - Low Fidelity
  - High Fidelity

- Detectability Experiments
  - Anchorprints
  - Structureprints
  - Behaviorprints
Low Fidelity

- 18,937 fingerprintable extensions tested
- WAR Fingerprintable 99.0% passed
- DOM Fingerprintable 98.7% passed
- WAR & DOM Fingerprintable 97.9% passed
High Fidelity

- 150 tested
- WAR Fingerprintable 50 passed
- DOM Fingerprintable 48 passed
- WAR & DOM Fingerprintable 47 passed
Evaluation - Errors

- Remote code loading
- Hardcoded values that Droxy alters
- Droxy limitations
Detection-Anchorprints

- 17,678 extensions tested
- Cloaked extensions were undetectable
- But 96 of the cloaked extensions did not maintain equivalent functionality
Detection-Structureprints

- 5,311 extensions tested with fuzzy matching
- Tags, Attributes, Text, 4.2% detected
- Tags and Attributes, 1.8% detected
- Tags 1.7% detected
Detection-Behaviorprints

- Ten of the most popular extensions
  - Seven detectable
- Randomly selected ten extensions
  - Five detectable
Summary

- Extension fingerprinting is a real problem
- Successfully performed late-stage customizations on browser extensions to break extension fingerprints
- Cloaked extensions:
  - 99.9% undetectable using anchorprints
  - 98.3% undetectable using structureprints
Thank you

EVERYBODY’S DIFFERENT
Client-side Diversification for Defending Against Extension Fingerprinting

Erik Trickel, Oleksii Starov, Alexandros Kapravelos, Nick Nikiforakis, and Adam Doupé