Carnus: Exploring the Privacy Threats of Browser Extension Fingerprinting

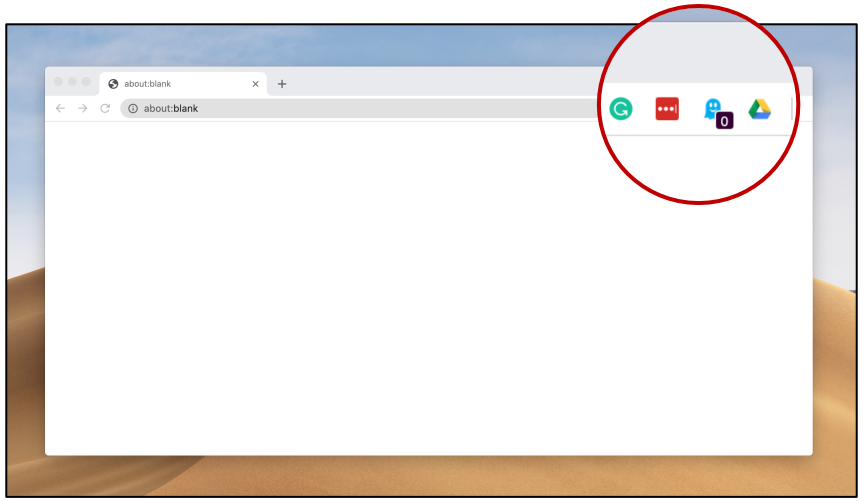
Soroush Karami, Panagiotis Ilia, Konstantinos Solomos, Jason Polakis
University of Illinois at Chicago, USA

skaram5@uic.edu

February 24, 2020

Browser extensions

- Extend functionality of the browser
  - “Adblock Plus” with 10,000,000+ users
  - “Tampermonkey” with 10,000,000+ users
  - “LastPass” with 10,000,000+ users



- Security threats of extensions have been studied
  - (e.g., Kapravelos et al; USENIX Security 2014)
- We focus on the privacy aspect of browser extensions
  - First, we build and evaluate the most comprehensive extension-fingerprinting system to date

Installed extensions might reveal user's interests, preferences, browsing habits, and demographic information

WebFilter FREE: Parental Control & Anti-Porn

Young Users

Ya'Muslim

Religion

Don't Pay Trump

Politics

中国空气质量指数

Ethnicity

3asyR

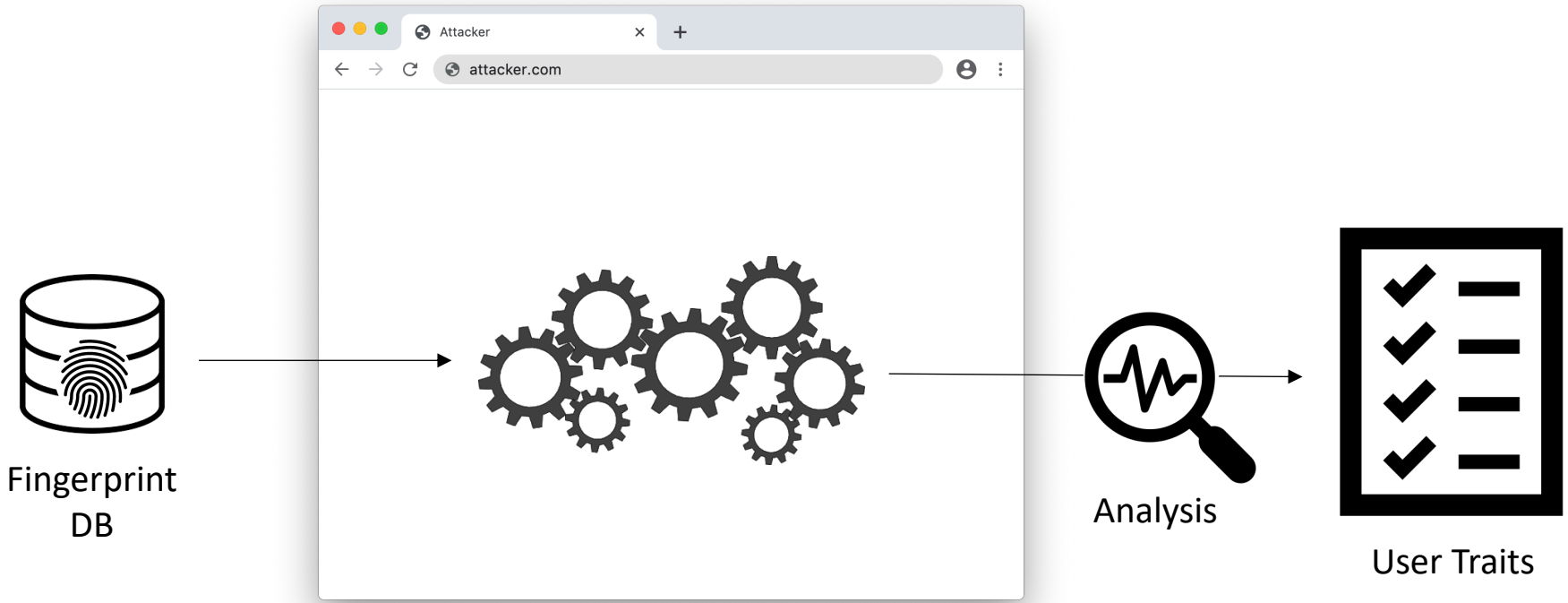
Health

LGBT Pride

Gender/ sexuality

Threat model

User visits attacker's website, which attempts to detect installed extensions



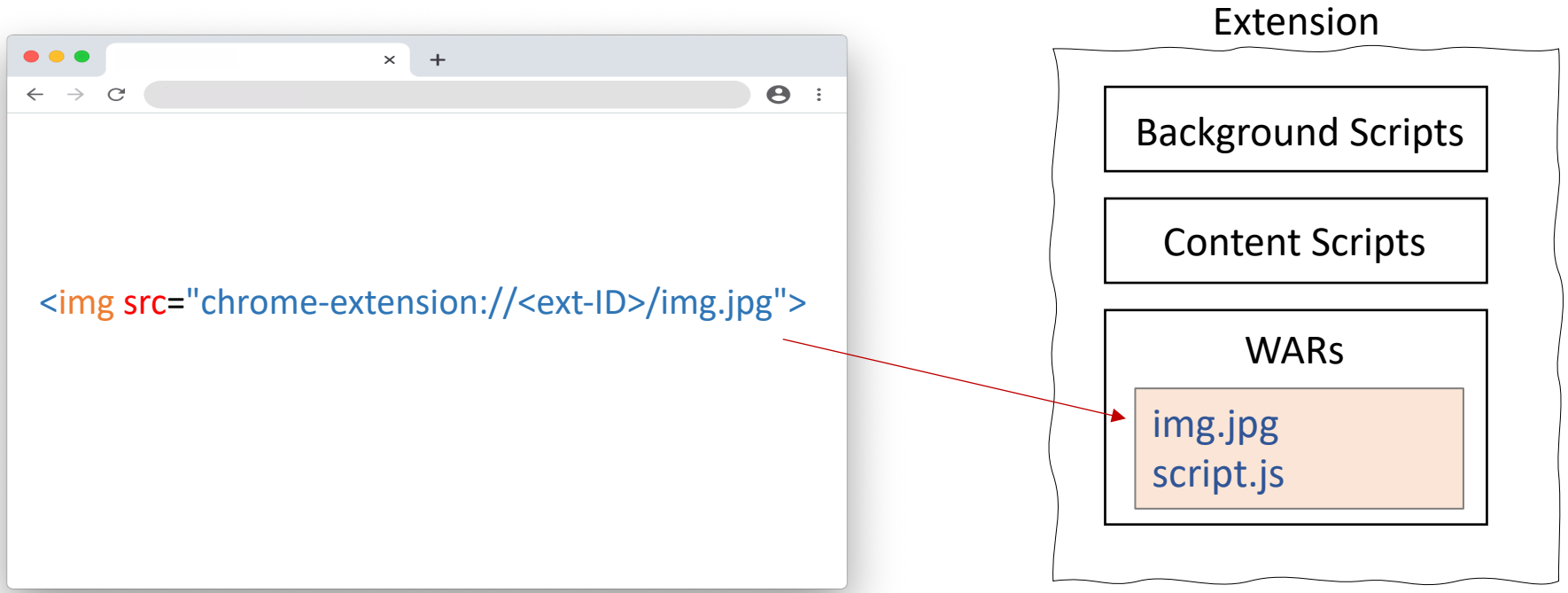
Fingerprinting techniques

For the purpose of detection, we generate a **Fingerprint** for each extension

1. WARs (web accessible resources)
2. Behavior-based
3. Intra-communication-based
4. Inter-communication-based

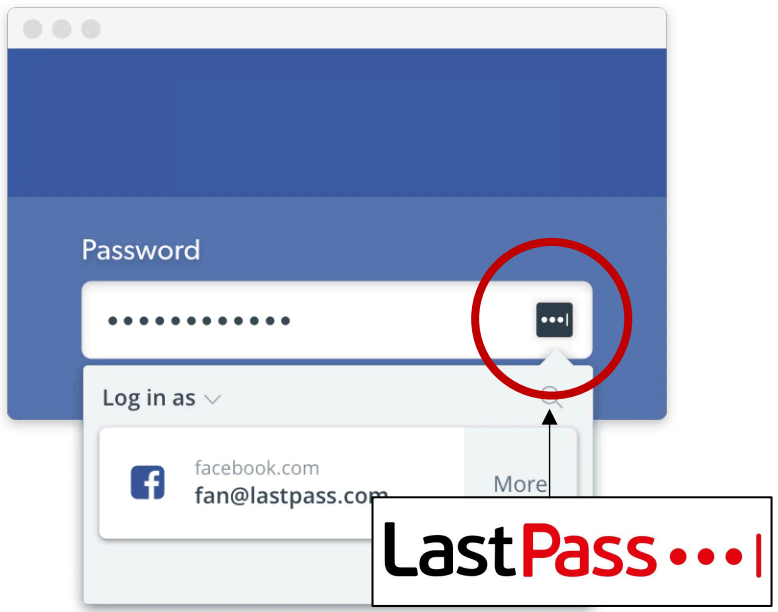
1. WAR-Based Fingerprints

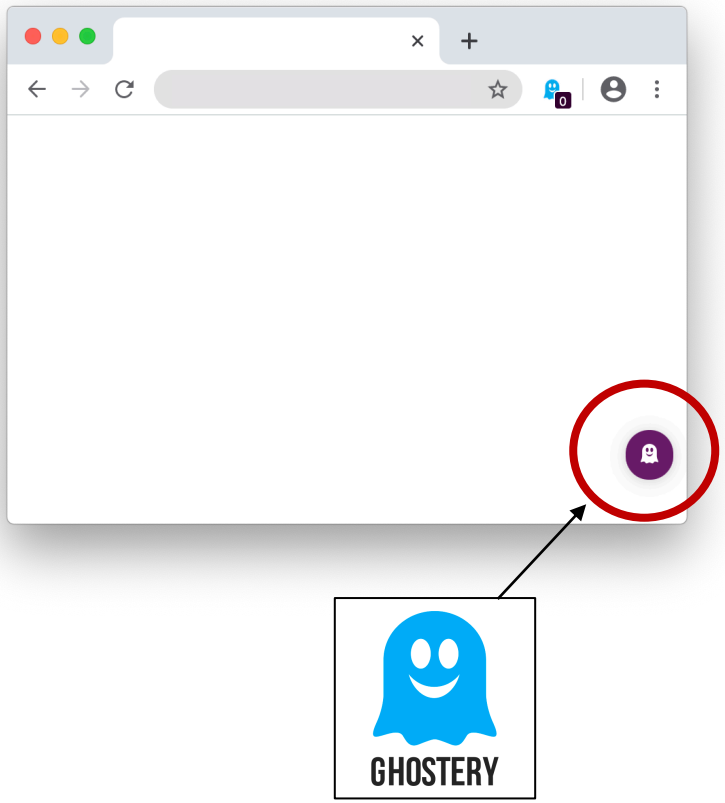
- Extensions may have some resources that are accessible from the DOM
- Websites can probe WARs to detect which extensions are installed in the user's browser
- Well-known approach for detecting extensions
  - Maximizes the coverage of our attack, enabling extensive exploration of privacy implications

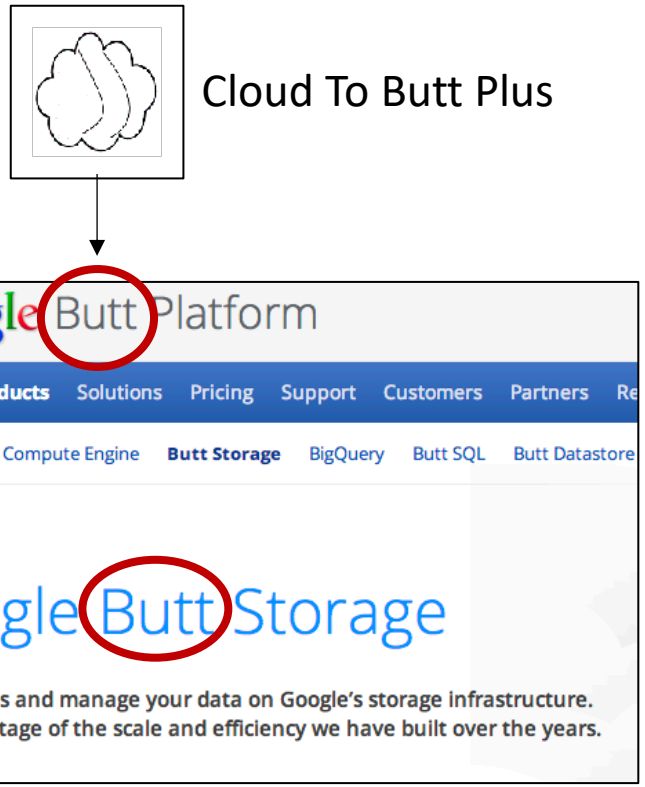


2. Behavior-Based Fingerprints

Extensions might add/remove images, buttons, code, or text to the web page

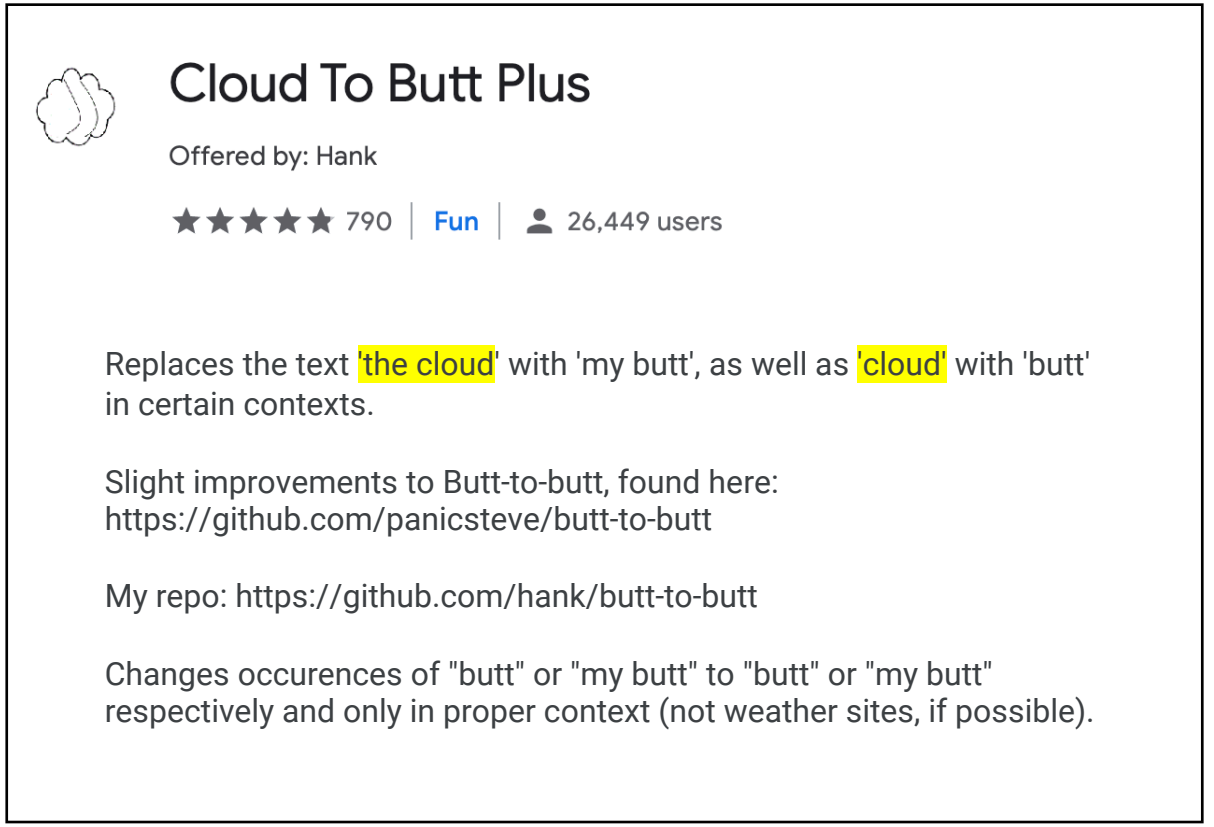






2. Behavior-Based Fingerprints

- Created a honeypage to trigger as many extensions as possible
  - Includes HTML, JS, CSS, text, etc
- Detecting content-based triggering is challenging
- **Observation**: use the extension's description to trigger such behavior



2. Behavior-Based Fingerprints

```
<form action="/action_page.php">
  <label for="uname"> Username </label>
  <input type="text" name="uname" autocomplete="on">
  <label for="psw"> Password </label>
  <input type="password" name="psw" autocomplete="on">
  <button type="submit"> Login </button>
</form>
```

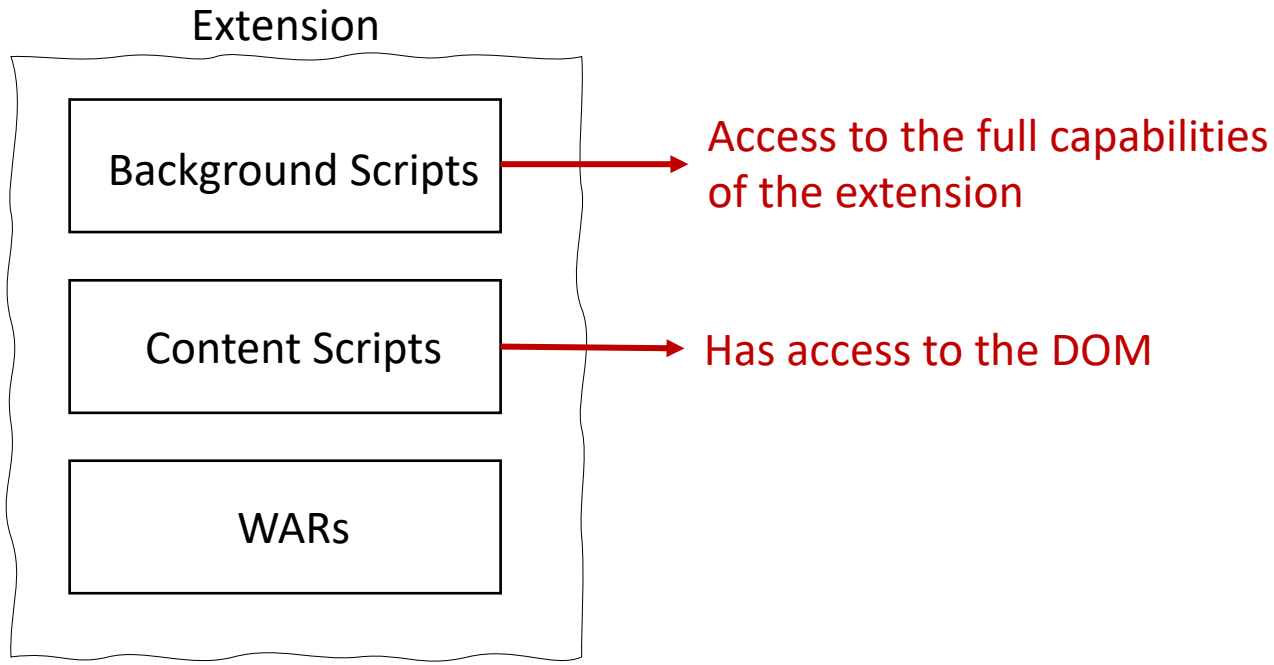
added

modified

```
<form action="/action_page.php">
  <label for="uname"> Username </label>
  <input type="text" name="uname" autocomplete="off"
style="background-image: url('...');">
  <label for="psw"> Password </label>
  <input type="password" name="psw" autocomplete="off"
style="background-image: url('...');">
  <button type="submit"> Login </button>
</form>
```

Added: {style="background-image: url('...');", autocomplete="off"}
Removed: {autocomplete="on"}

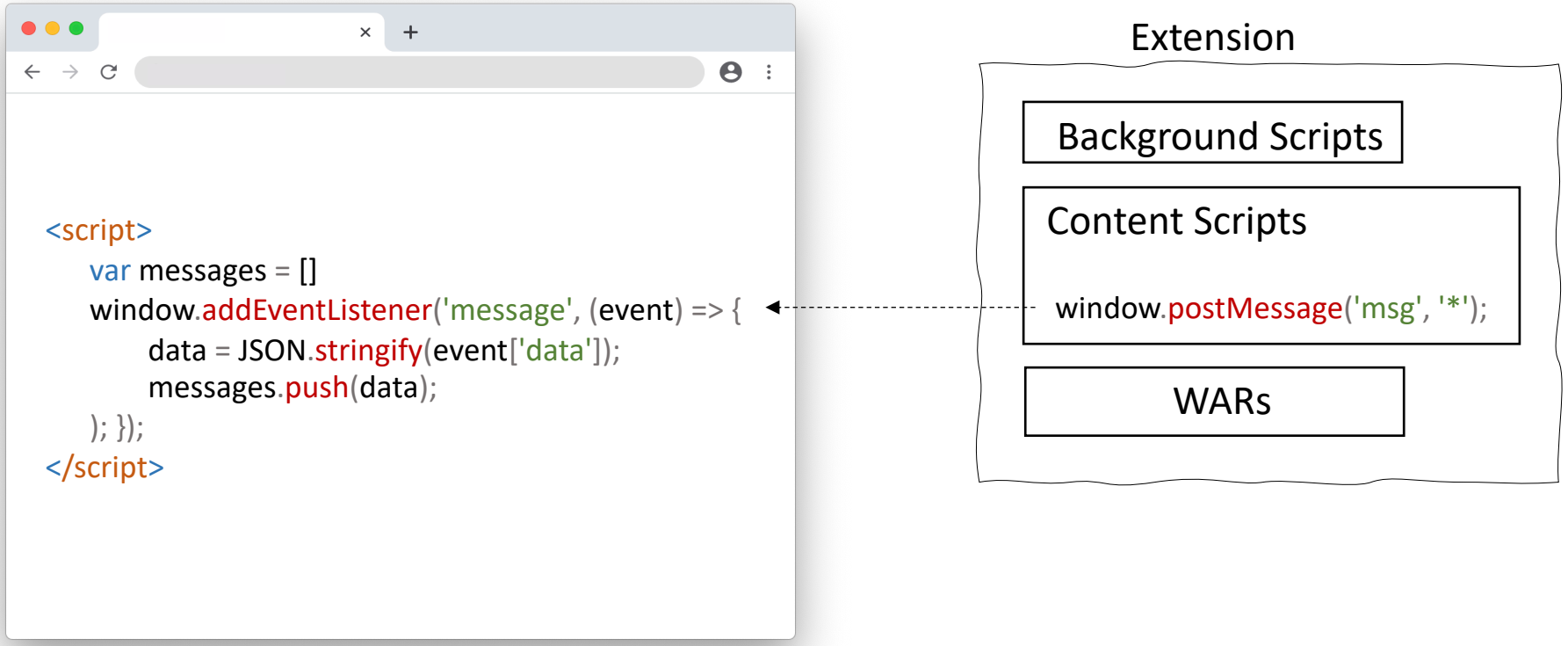
3. Intra-communication Based Fingerprints



We use the messages that are sent by content scripts to detect extensions.

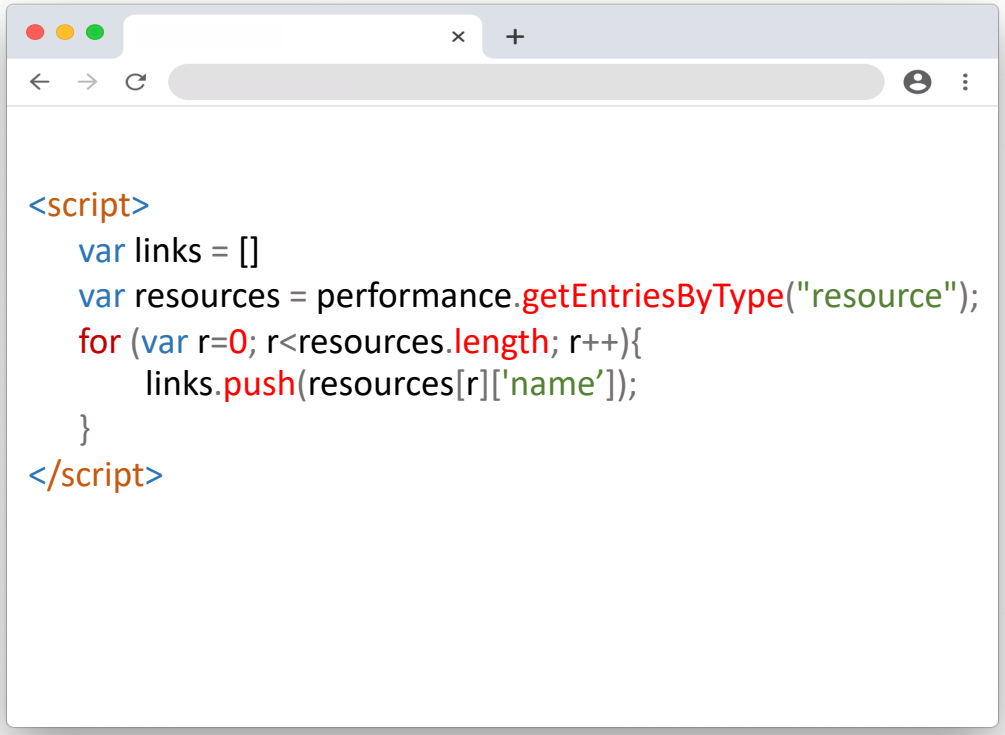
3. Intra-communication Based Fingerprints

We use the messages sent by content scripts to detect extensions.

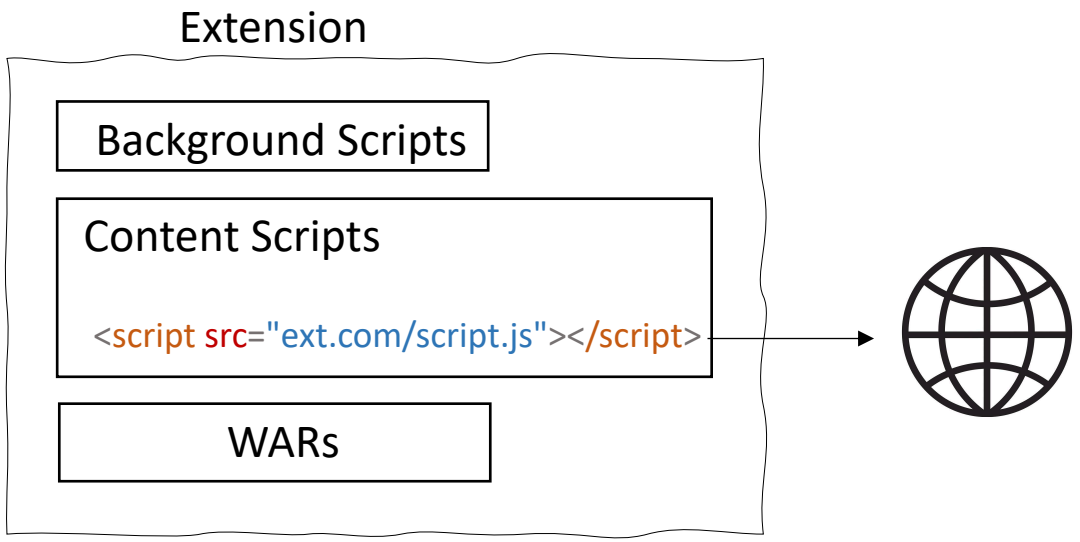


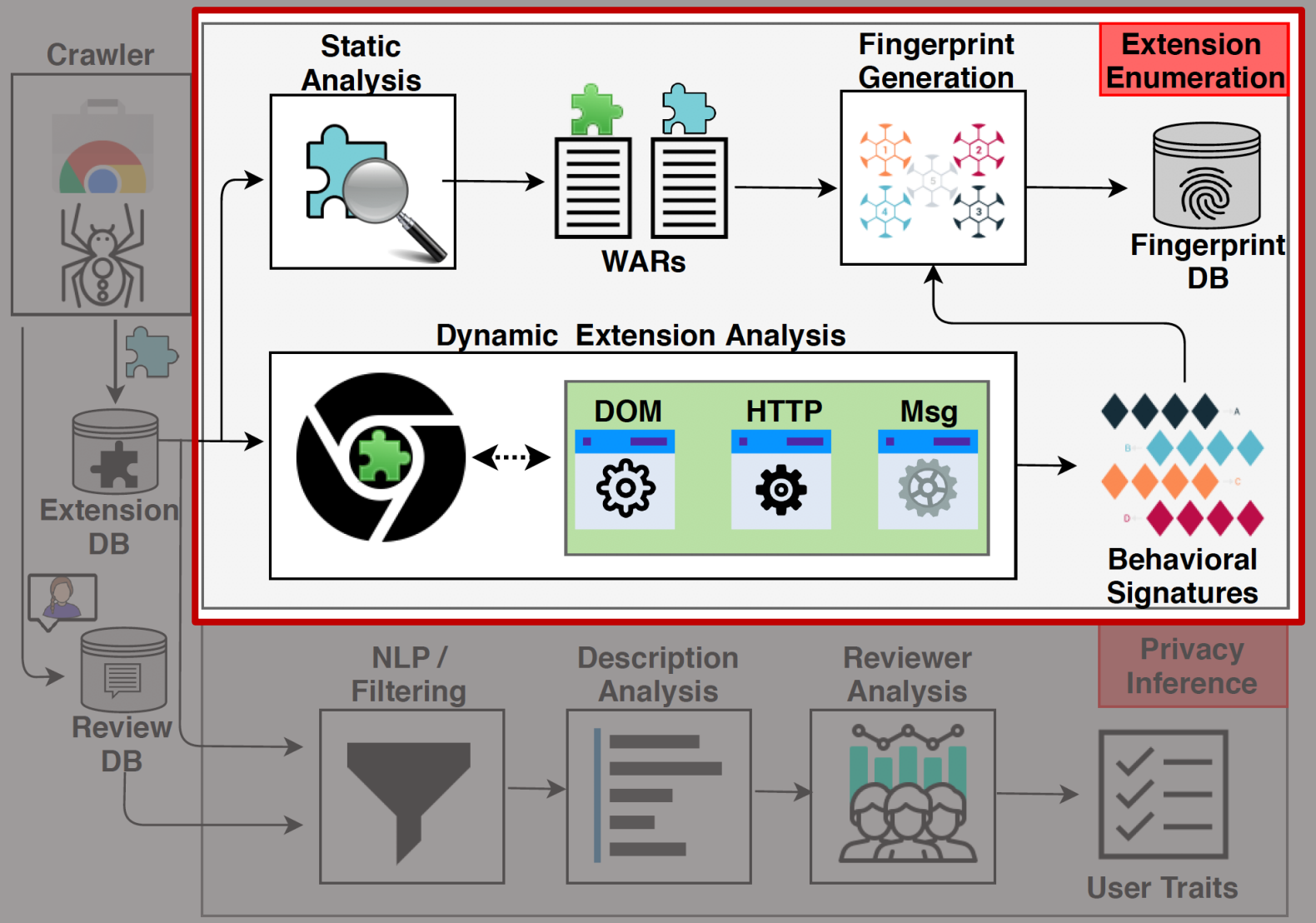
4. Inter-communication Based Fingerprints

- Content scripts may fetch resources from the network
- Attackers can use Performance API to obtain list of fetched resources



```
<script>
    var links = []
    var resources = performance.getEntriesByType("resource");
    for (var r=0; r<resources.length; r++){
        links.push(resources[r]['name’]);
    }
</script>
```




Crawler
Static Analysis
WARs
Fingerprint Generation
Extension Enumeration
Fingerprint DB
Dynamic Extension Analysis
DOM
HTTP
Msg
Behavioral Signatures
Extension DB
Review DB
NLP / Filtering
Description Analysis
Reviewer Analysis
Privacy Inference
User Traits

Extension Enumeration Phases

This phase is repeated three times.

Reason:

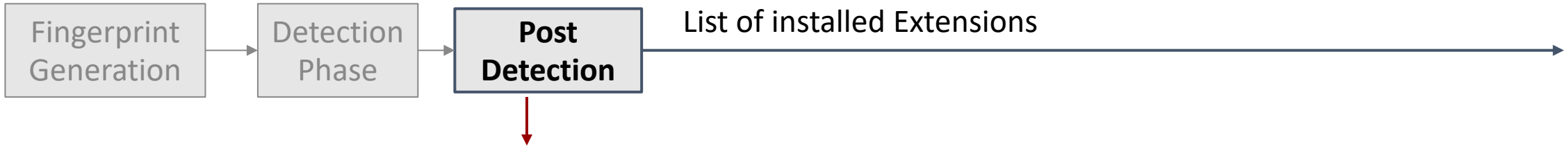
1. Different behaviors of an extension.
   - 1st behavior: {"image-1.jpg"}
   - 2nd behavior: {"image-2.jpg"}
2. Dynamic components
   - {…, timestamp="123"}
   - {…, timestamp="456"}
   - {…, timestamp="789"}

Extension Enumeration Phases

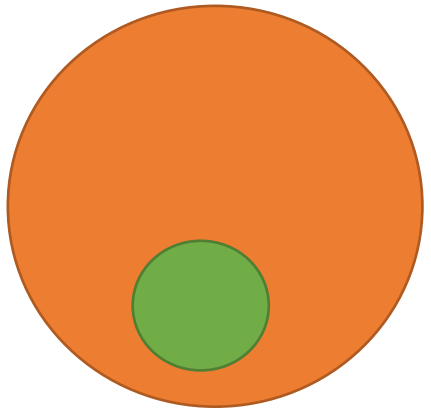
We allow a certain number of components to mismatch

Reason: for removing false negatives.

Extension Enumeration Phases

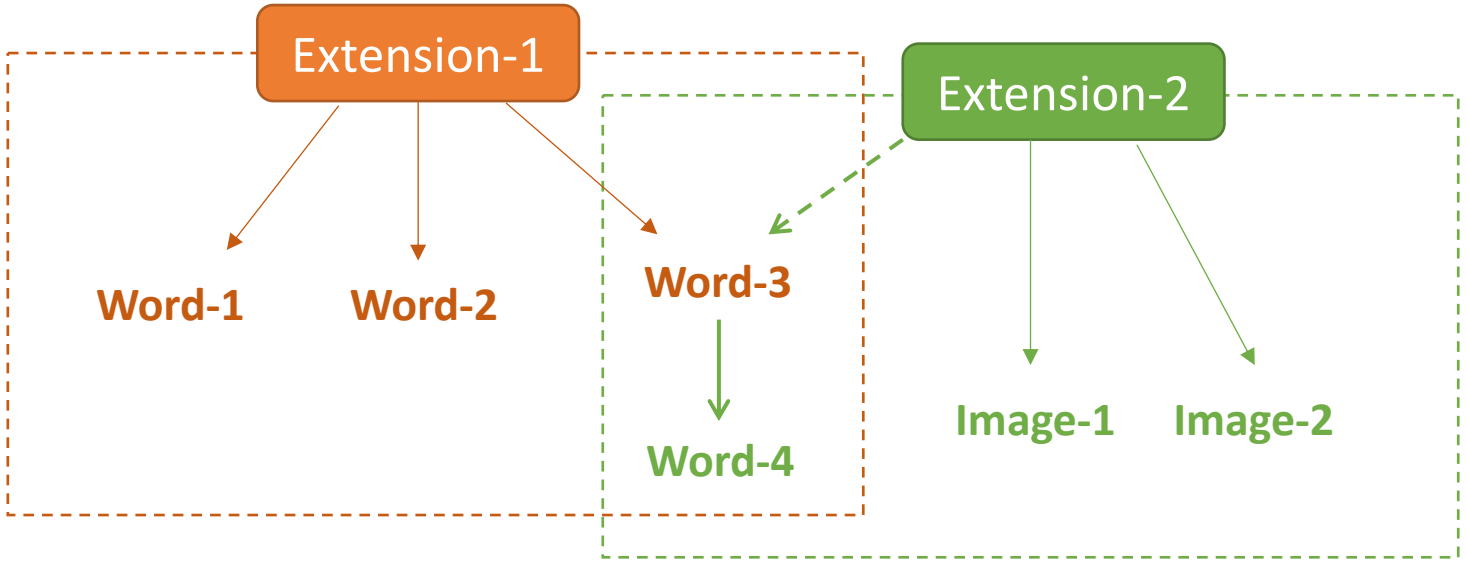


- From the list of detected extensions
  - if one extension's fingerprint is a subset of another one
    - remove this extension from the list of detected extensions



Practical Challenges: co-interference

Modifications of one extension can affect the modifications of the other



Experimental Evaluation

Attack Accuracy

- Randomly install a set of extensions (N=2..10), run detection
- Repeat this process 100 times
- Our system always correctly identifies more than 97% of installed extensions
  - Average false positive rate: 4.77%
  - Average false negative rate: 1.93%

Attack Duration

- Optimize attack by offloading most computation to server
- Average client-side attack: 8.77 seconds
- Average server-side computation: 3.62 seconds
- (Off-the-shelf desktop: Quad Core Intel i7-7700 and 32GB of RAM)

Comparison to previous studies

Paper	Attack	Platform	Extensions	Detectable
[Starov et al., S&P '17]	Behavior-based	Chrome	10,000	920
[Sjosten et al., CODASPY '17]	WAR-based	Chrome Firefox	43,429 14,896	12,154 1,003
[Gulyas et al., WPES '18]	WAR-based	Chrome	13,000	5,107
[Sanchez-Rola et al., USENIX '17]	WAR Side-channel	Chrome Firefox	10,620 10,620	10,620 10,620
[Sjosten et al., NDSS '19]	WAR Revelation	Chrome Firefox	10,459 8,646	1,932 1,379
Ours	**Multi-class**	**Chrome**	**102,482**	**29,536**

Countermeasure effects

- [Trickel et al,. USENIX '19] is a defense against extension fingerprinting
  - Randomizes the values of **ID** and **class** attributes
  - Injects random **tags** and **attributes** into each page
  - Randomizes the **path** of the WARs

- During the fingerprint generation phase, we can identify and remove the unstable components from fingerprints

Countermeasure effects: example

1. CloakX doesn't affect this fingerprint

Before {font-size:10px, color:white, initial, text-align:left, justify-content:center, line-height:4px, id="dv_masterkey_banner", flex-grow:0, rgb(160,160,160), class="dv_masterkey_message", access, id="____ok_icom_in___", position:absolute, Arial, display:flex, font-size:14px, class="dv_masterkey_banner", id="dv_launch_onepassui", style="color:orange", center, z-index}

After {font-size:10px, color:white, initial, text-align:left, justify-content:center, flex-grow:0, rgb(160,160,160), access, position:absolute, Arial, display:flex, style="color:orange", line-height:4px, center, z-index, font-size:14px}

2. CloakX renders this fingerprint useless

Before {style="display:none;", class="hashmenu01"}

After {style="display:none;"} → Too generic

Countermeasure effects: example

1. CloakX doesn't affect

Before {font-size:10p… …ight:4px,
id="dv_mast… …y_message",
access, id="_… …:14px,
class="dv_ma… …', center, z-index}

After {font-size:10p… …ow:0,
rgb(160,160,… …nge", line-
height:4px, c…

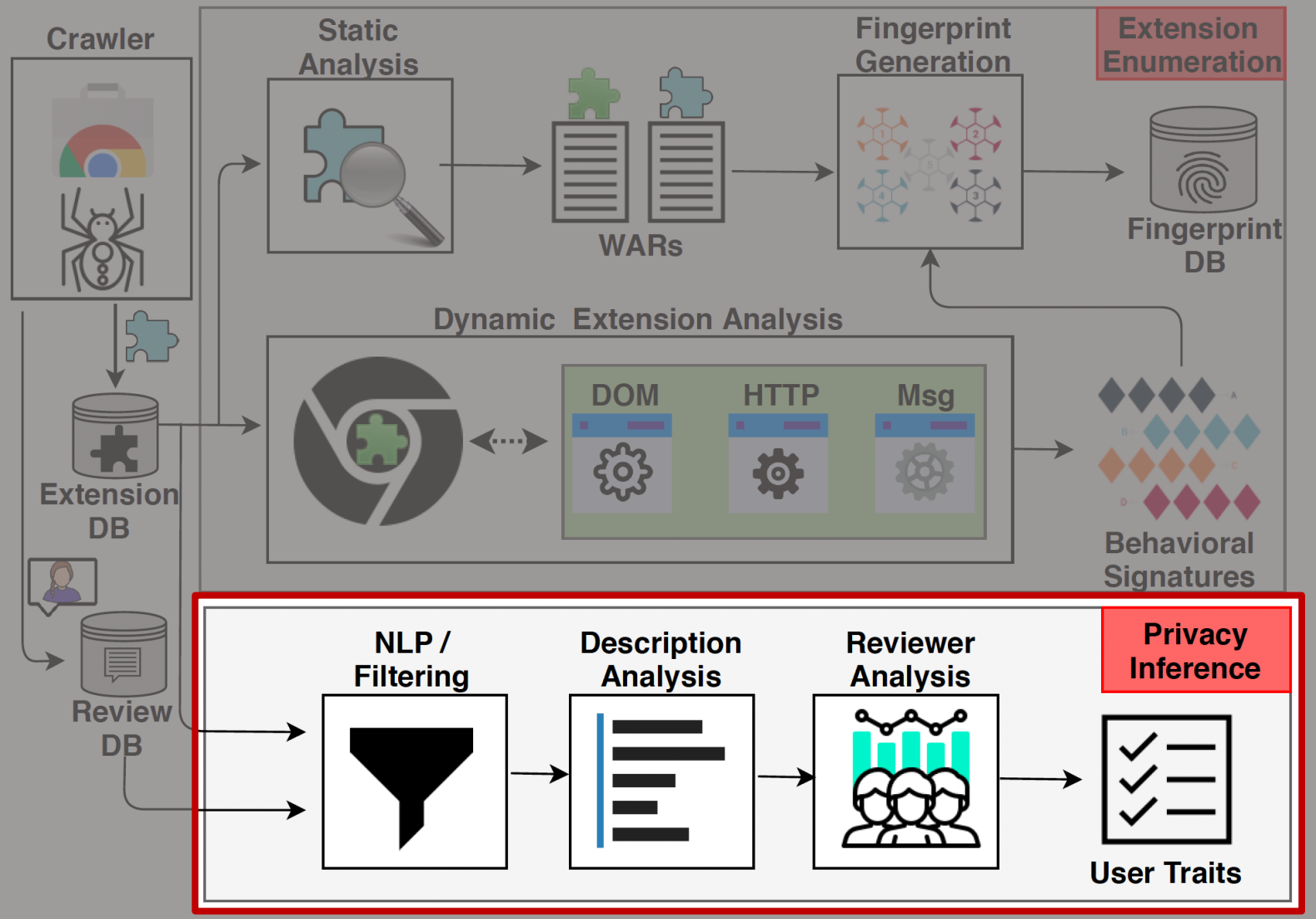
2. CloakX renders this fi…

Before {style="display:none;", class="hashmenu01"}

After {style="display:none;"} → Too generic

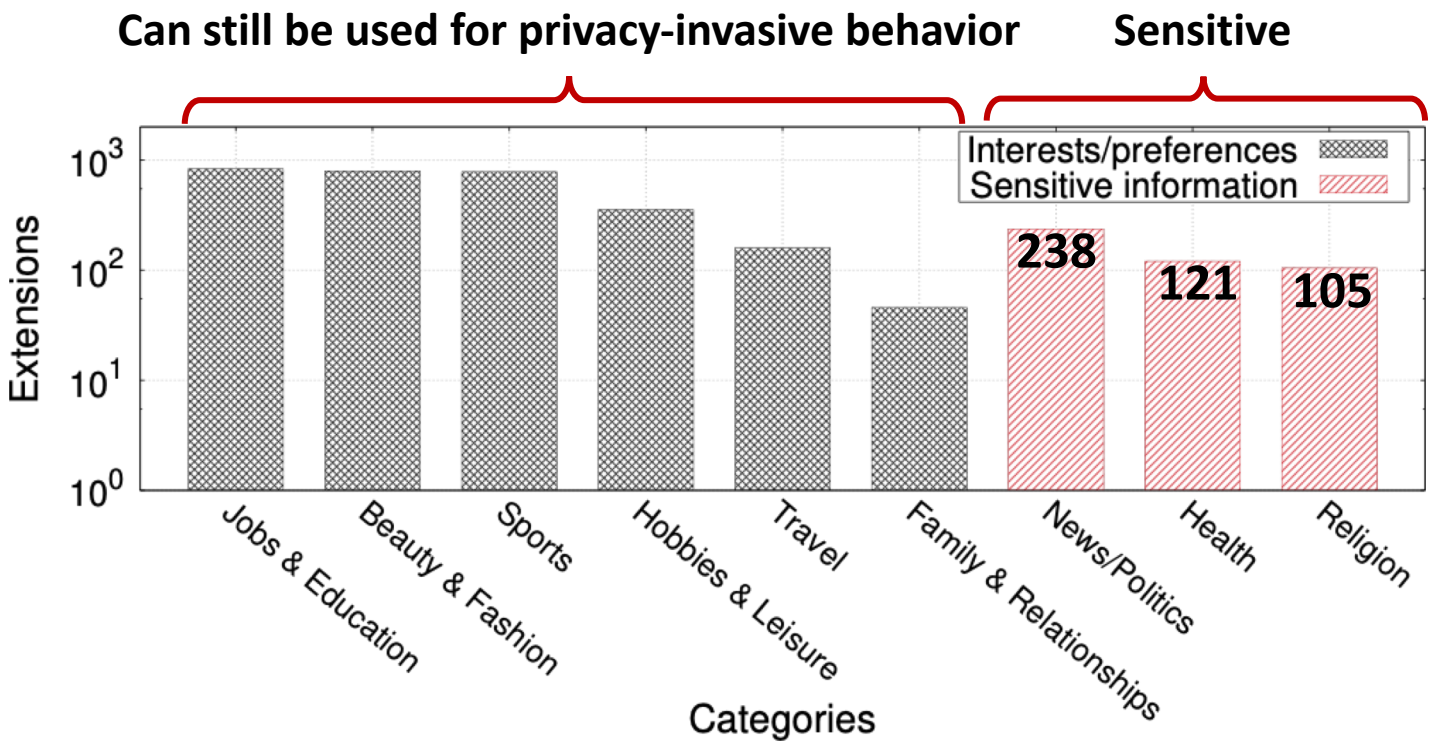
At least **83.6%** of our behavior-based fingerprints remain effective.

Still, this defense is an important step in the **right direction.** We hope that our work incentivizes more research.


Crawler
Static Analysis
WARs
Fingerprint Generation
Extension Enumeration
Fingerprint DB
Dynamic Extension Analysis
DOM
HTTP
Msg
Behavioral Signatures
Extension DB
Review DB
NLP / Filtering
Description Analysis
Reviewer Analysis
Privacy Inference
User Traits

1. Inference Attacks: Topic Classification

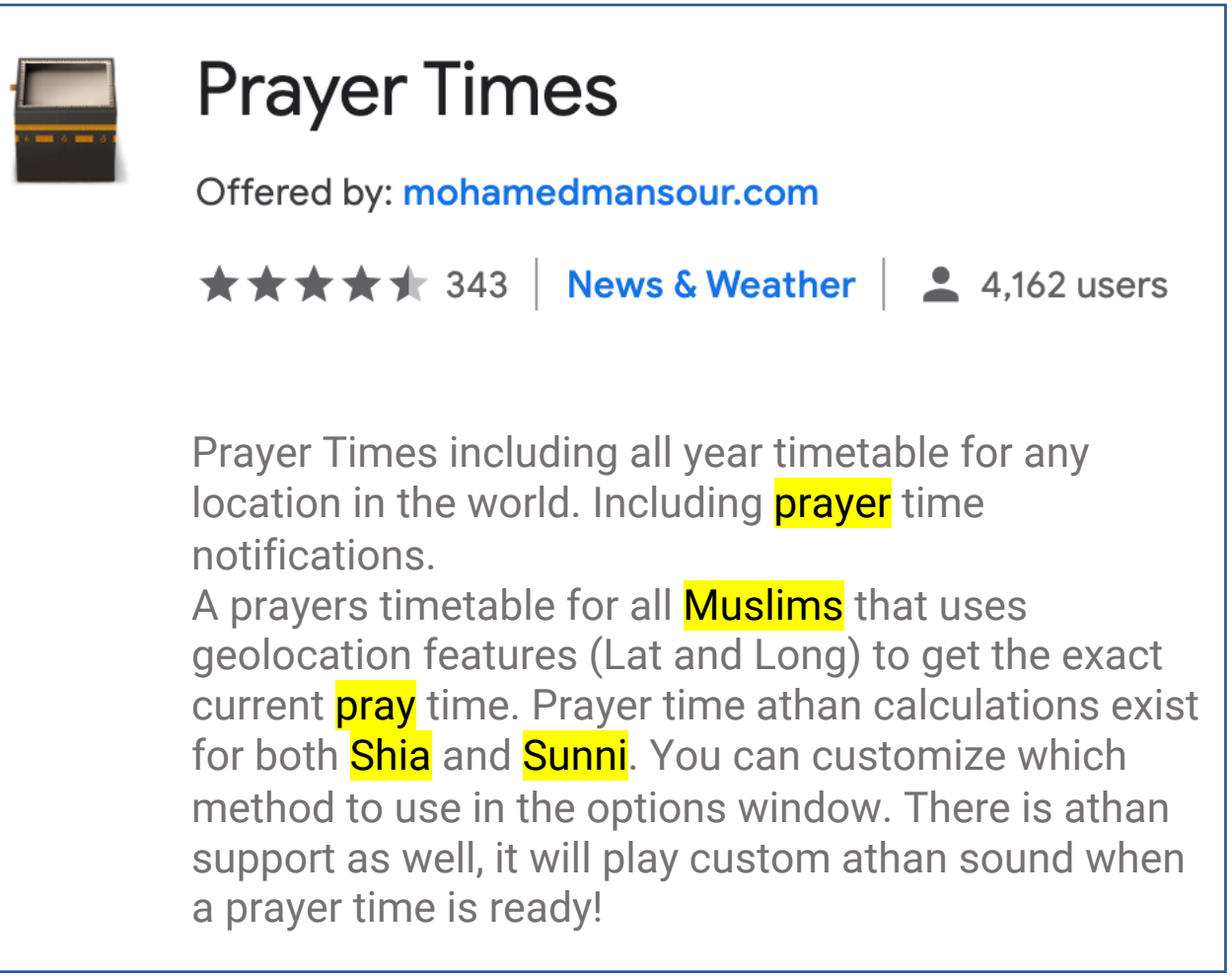
- Use extensions' description text from Chrome Web Store
- Contains a lot of irrelevant text → **Pre-process**, **translate** and **clean** descriptions
- Google's Natural Language API



2. Inference Attacks: Description-based

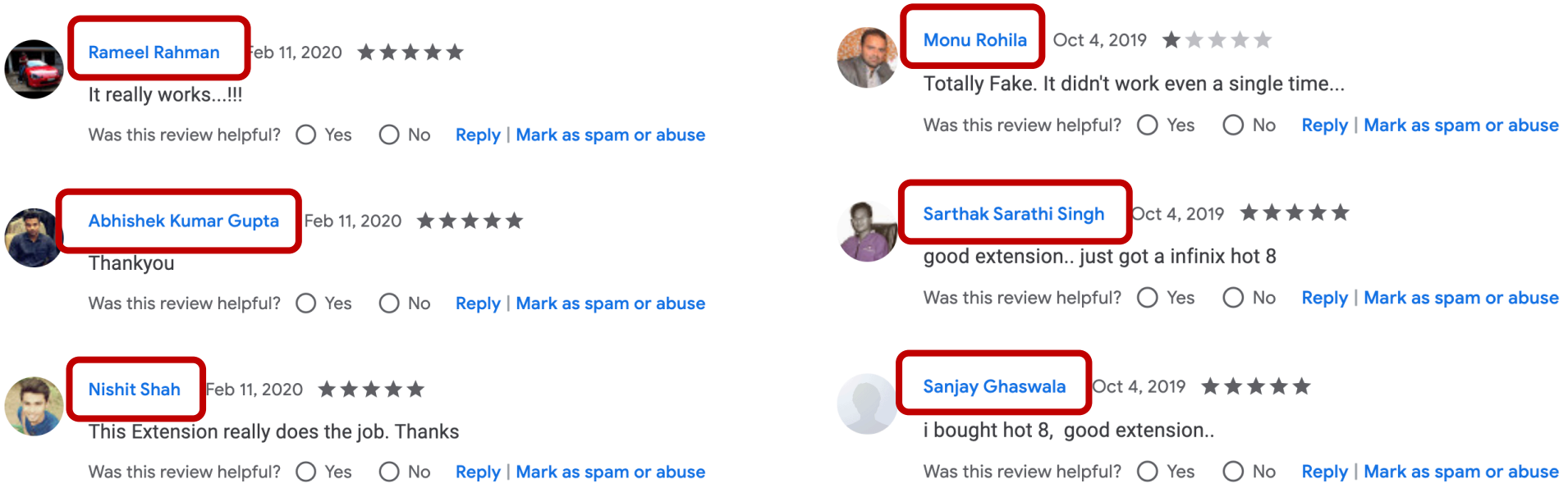
- **spaCy's** Named Entity Recognition
  - E.g., locations, people, etc.

- Using different wordlists
  - Religious terms
  - Medical terms
  - Political terms



3. Inference Attacks: Reviewer-based Inference

- Extract name of extensions' reviewers → map names to **ethnicities** and **sex**
  - Use Shannon-Wiener index to identify predominant ethnicity/sex
- Example: "FlipShope- Flash sale autobuy" is mainly reviewed by users with Indian names



Contributions

- Demonstrated the *first* automated creation and detection of behavior-based fingerprints for identifying browser extensions.
- Introduced two novel fingerprinting techniques, that are robust against all existing countermeasures.
- Presented the largest extension fingerprinting study, and evaluated a state-of-the-art countermeasure.
- Presented the first empirical analysis on the privacy inference attacks enabled by browser extensions.
- Conduct the largest extension-unicity analysis and explore the use of user reviews as a novel deanonymization vector (see paper).

Questions?

Feel free to contact me:
skaram5@uic.edu