Latex Gloves:
Protecting Browser Extensions from Probing and Revelation Attacks

Alexander Sjösten, Steven Van Acker, Pablo Picazo-Sanchez, Andrei Sabelfeld

CHALMERS
UNIVERSITY OF TECHNOLOGY

Browser extensions

- Allows users to modify browser behaviour
  - Block advertisement & tracking scripts
  - Password managers

GHOSTERY

- Written in a combination of JavaScript, HTML and CSS
  - Content scripts
  - Background scripts

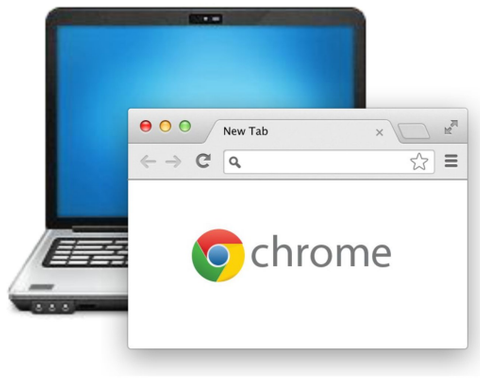
- User grants permissions

- Can inject content
  - One way through “web accessible resources”
  - `chrome-extension://` and `moz-extension://`

Google Cast example

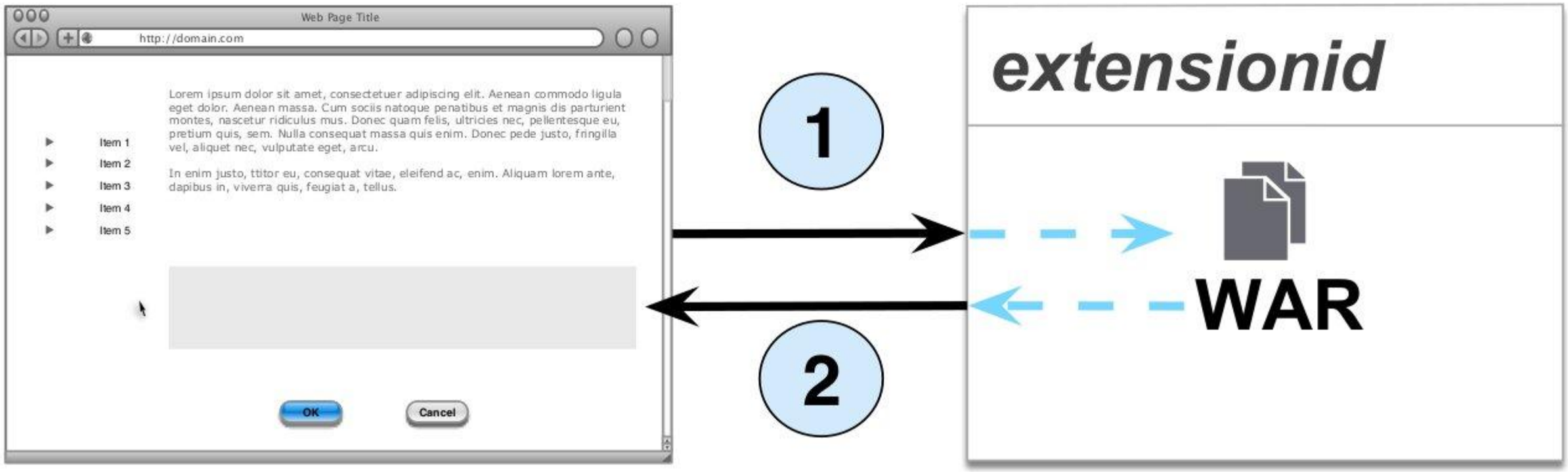
Detect google cast extension

`chrome-extension://boadgeojelhgndaghljhdicfkmllpafd/cast_sender.js`



Discover Chromecast on the network

Probing attack

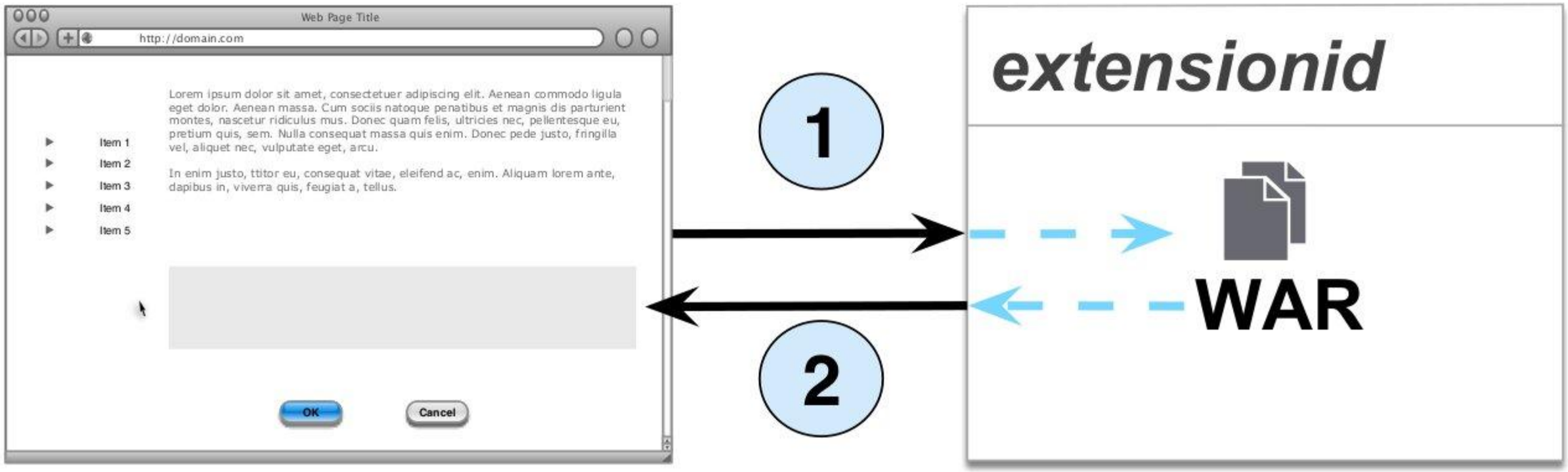


1) Web page makes request to

```
chrome-extension://boadgeojelhgndaghljhdicfkmllpafd/cast_sender.js
```

Sjösten et al., CODASPY 2017
Gulyás et al., WPES 2018 (demo web page: https://extensions.inrialpes.fr/)
Sanchez-Rola et al., USENIX 2017

Probing attack



1) Web page makes request to

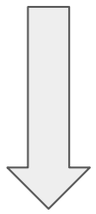
```
chrome-extension://boadgeojelhgndaghljhdicfkmllpafd/cast_sender.js
```

2) If extension is installed, resource is returned.

Sjösten et al., CODASPY 2017
Gulyás et al., WPES 2018 (demo web page: https://extensions.inrialpes.fr/)
Sanchez-Rola et al., USENIX 2017

Mozilla’s solution

moz-extension://actual-extension-id/resource.js

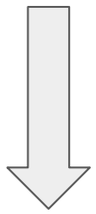


Randomized

moz-extension://30bb95e6-4208-4633-ab7b-5623c0b09483/resource.js

Mozilla’s solution

moz-extension://actual-extension-id/resource.js



moz-extension://30bb95e6-4208-4633-ab7b-5623c0b09483/resource.js

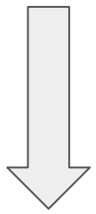
“*It is randomly generated for every browser instance. This prevents websites from fingerprinting a browser by examining the extensions it has installed.*”

- Mozilla documentation

  https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/manifest.json/web_accessible_resources

Mozilla’s solution

moz-extension://actual-extension-id/resource.js



moz-extension://30bb95e6-4208-4633-ab7b-5623c0b09483/resource.js

“*It is randomly generated for every browser instance. This prevents websites from fingerprinting a browser by examining the extensions it has installed.*”

- Mozilla documentation

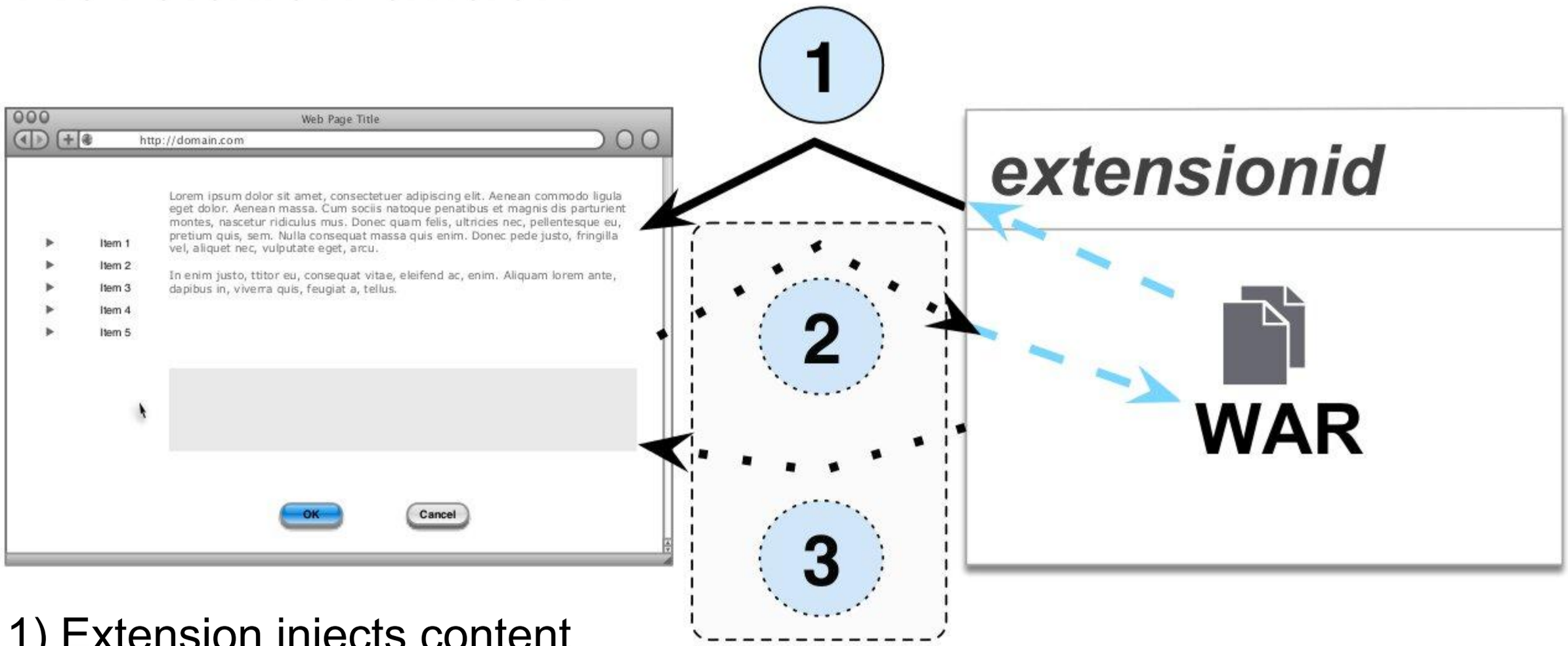
https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/manifest.json/web_accessible_resources

“*This is something we'd like to do when we have the opportunity to make a breaking change.*”

- Chrome developer forum

https://bugs.chromium.org/p/chromium/issues/detail?id=611420#c19

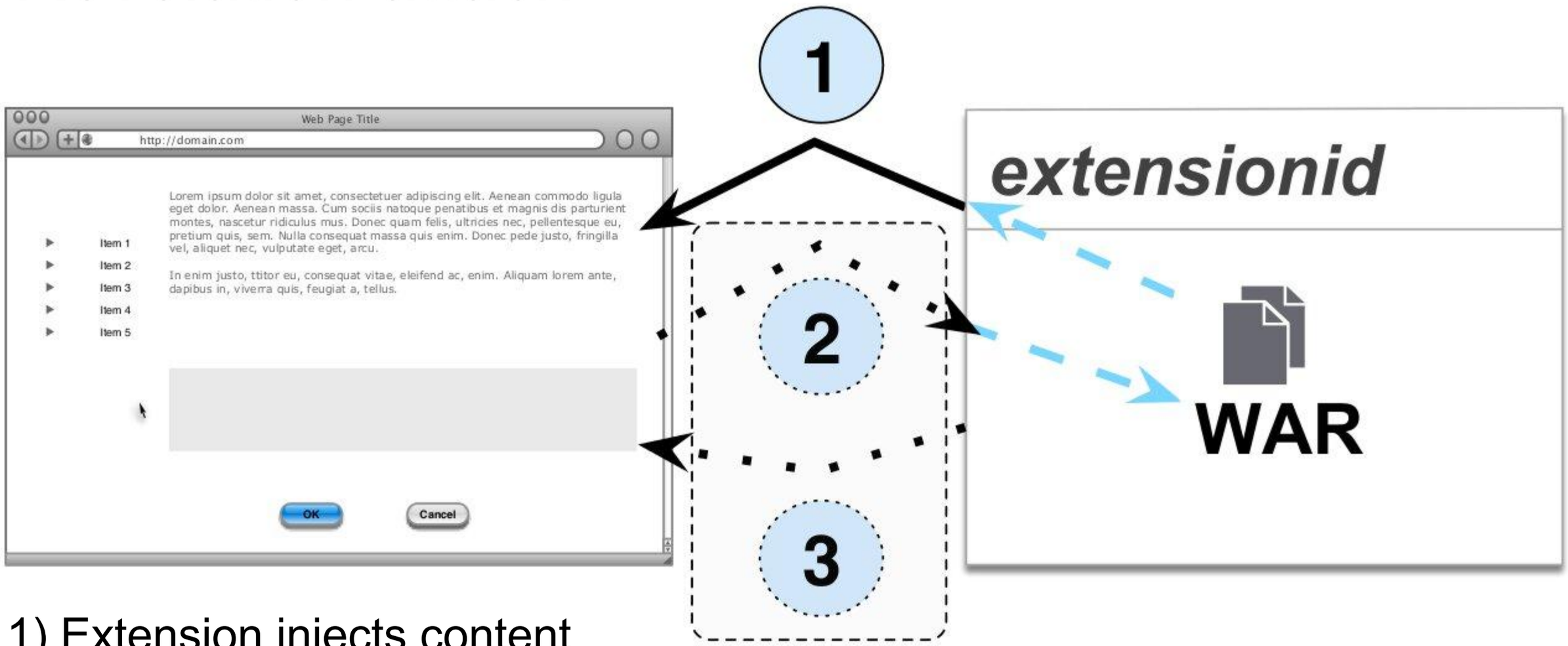
Revelation attack



1) Extension injects content

```
moz-extension://30bb95e6-4208-4633-ab7b-5623c0b09483/resource.js
```

Revelation attack

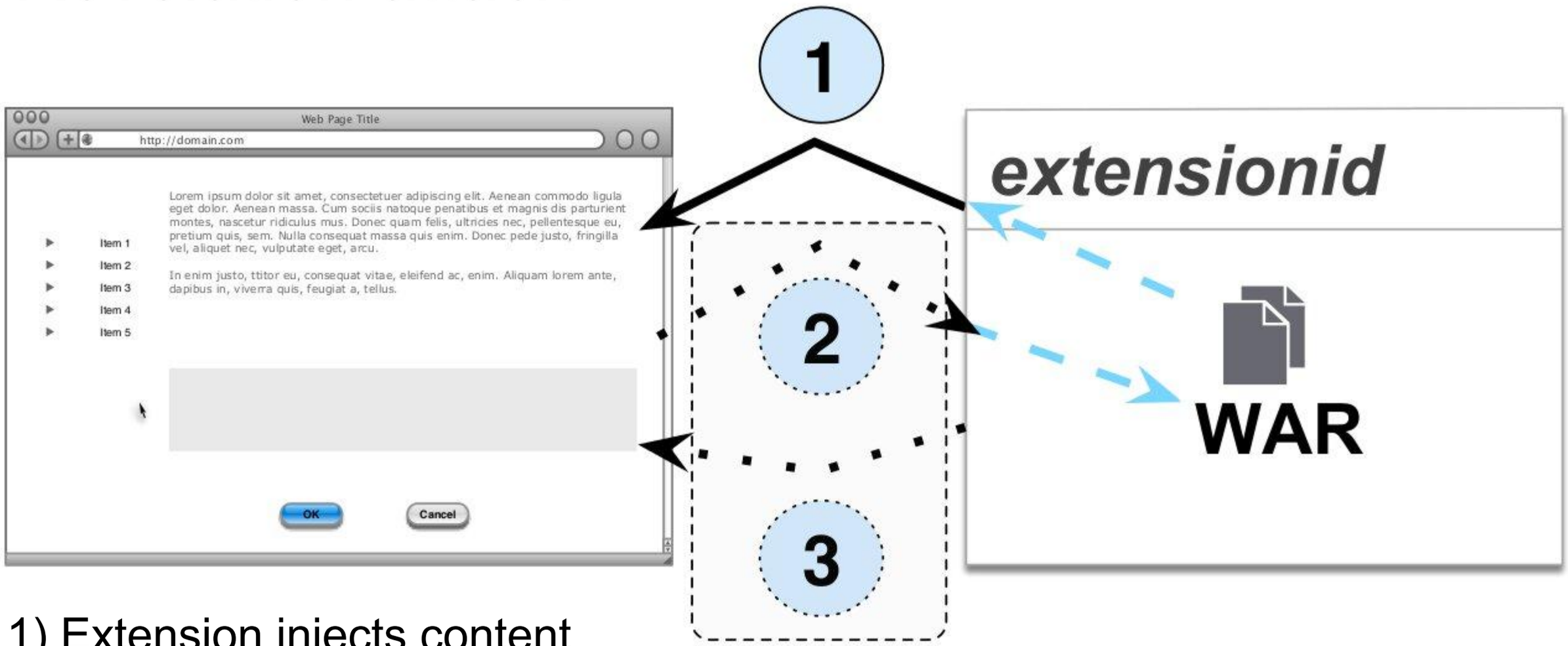


1) Extension injects content

`moz-extension://30bb95e6-4208-4633-ab7b-5623c0b09483/resource.js`

2) Use the recently acquired random ID to probe for a unique resource in an extension.

Revelation attack



1) Extension injects content

   `moz-extension://30bb95e6-4208-4633-ab7b-5623c0b09483/resource.js`

2) Use the recently acquired random ID to probe for a unique resource in an extension.
3) If extension is installed, resource is returned.

Extensions susceptible to revelation attack

- Filter extensions which might inject content
- Check if they have (at least) one unique path to a resource
- Check if they have (at least) one resource with unique content

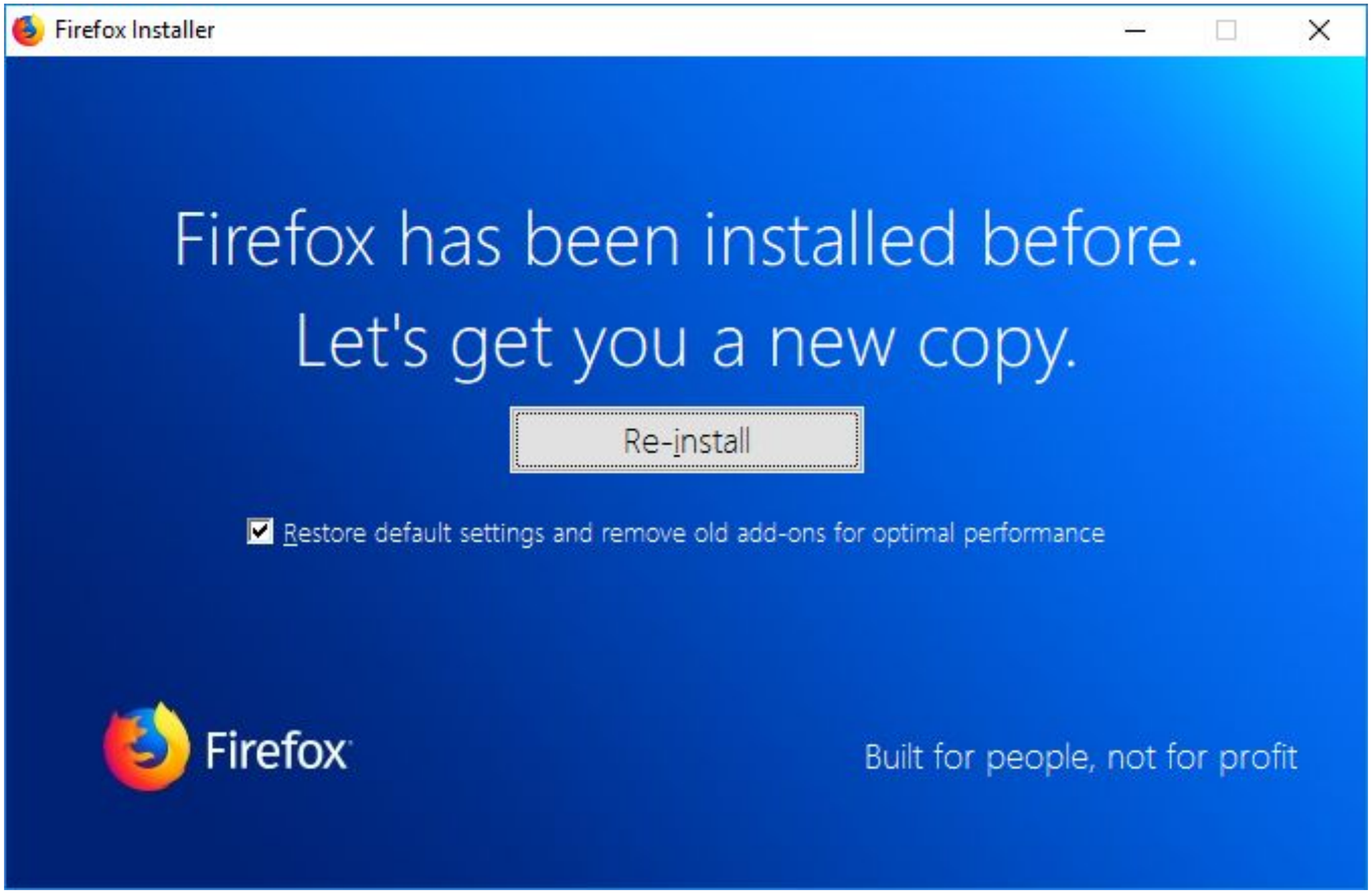
Extensions susceptible to revelation attack

- Filter extensions which might inject content
- Check if they have (at least) one unique path to a resource
- Check if they have (at least) one resource with unique content

	Extensions total	Susceptible
Firefox	1,378	1,301 (94.41%)
Chrome	11,633	10,459 (89.91%)
Total	13,011	11,760 (90.39%)

How can one reset the random UUID?

		Linux	Mac OSX	Windows
Restarting browser		No		
Updating browser		No		
Re-installing browser		No		Yes
Updating extension		No		
Re-installing extension	w/ browser restart	Yes		
	w/o browser restart	No		
Incognito mode		No		
Clearing cache and cookies		No		
Clearing the profile		Yes		


Firefox Installer
Firefox has been installed before.
Let's get you a new copy.
Re-install
Restore default settings and remove old add-ons for optimal performance
Firefox
Built for people, not for profit

How can one reset the random UUID?

		Linux	Mac OSX	Windows
Restarting browser		No		
Updating browser		No		
Re-installing browser		No		Yes
Updating extension		No		
Re-installing extension	w/ browser restart	Yes		
	w/o browser restart	No		
Incognito mode		No		
Clearing cache and cookies		No		
Clearing the profile		Yes		

How many extensions reveal themselves?

```
"exclude_matches": [ "*://*/_/chrome/newtab*" ],
"js": [ "dist/content_script_bundle.js" ],
"matches": [ "http://*/*", "https://*/*" ],
```

How many extensions reveal themselves?

- 3 sets of URLs
  - “real” URLs: derived from the `matches` attribute
  - “attackerhost” URLs: replace hostname with `attacker.invalid`
    `http://www.example.com/abc` ⇒ `http://www.attacker.invalid/abc`
  - “buydns” URLs: for more fine-grained regexps, e.g. `http://*.com/abc`
    `http://www.example.com/abc` ⇒ `http://www.attacker.com/abc`

How many extensions reveal themselves?

- 3 sets of URLs
  - “real” URLs: derived from the `matches` attribute
  - “attackerhost” URLs: replace hostname with `attacker.invalid`
    `http://www.example.com/abc` ⇒ `http://www.attacker.invalid/abc`
  - “buydns” URLs: for more fine-grained regexps, e.g. `http://*.com/abc`
    `http://www.example.com/abc` ⇒ `http://www.attacker.com/abc`

- Extract the regular expressions
  - 24,398 unique regular expressions

How many extensions reveal themselves?

- 3 sets of URLs
  - “real” URLs: derived from the `matches` attribute
  - “attackerhost” URLs: replace hostname with `attacker.invalid`
    `http://www.example.com/abc` ⇒ `http://www.attacker.invalid/abc`
  - “buydns” URLs: for more fine-grained regexps, e.g. `http://*.com/abc`
    `http://www.example.com/abc` ⇒ `http://www.attacker.com/abc`

- Extract the regular expressions
  - 24,398 unique regular expressions

- Performed crawling using CommonCrawl database
  - Contains ~4.57 billion URLs
  - For each regular expression: consider only first 100 matching URLs
  - For each extension: take random set of max 1000 URLs

How many extensions reveal themselves?

	Content-dependent		Any content		Total	
Chromium	508	(5,908,381)	2,176	(31,903,741)	2,684	(37,812,122)
Firefox	68	(115,720)	154	(676,318)	222	(792,038)
Either browser	576	(6,024,101)	2,330	(32,580,059)	2,906	(38,604,160)

How many extensions reveal themselves?

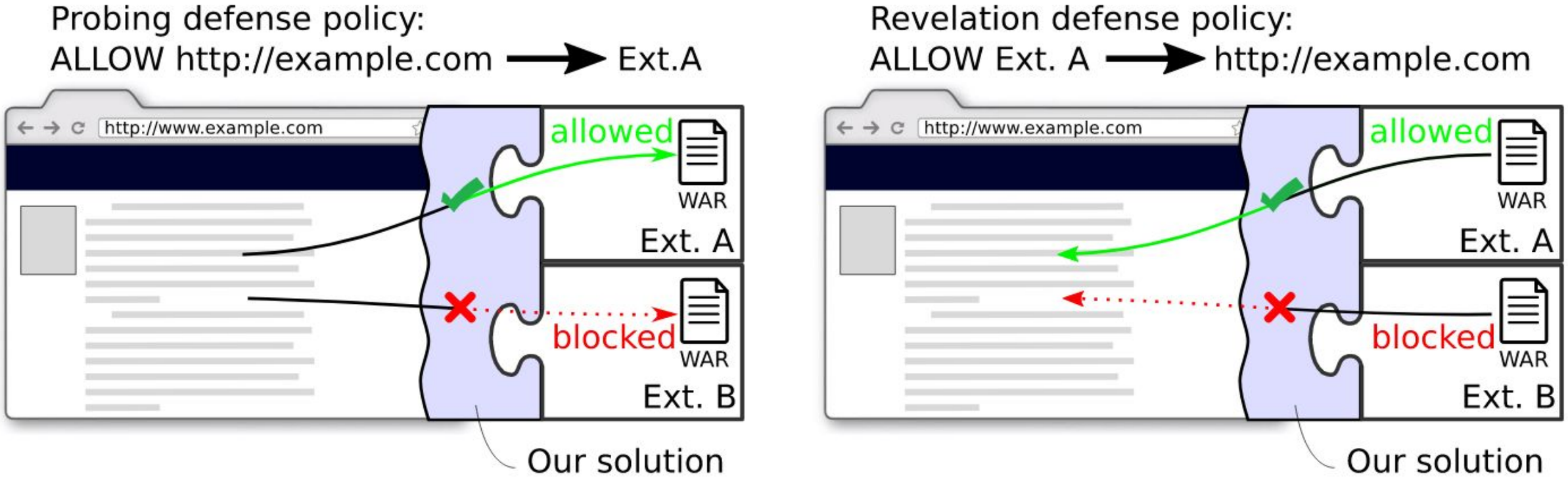
	Content-dependent		Any content		Total	
Chromium	508	(5,908,381)	2,176	(31,903,741)	2,684	(37,812,122)
Firefox	68	(115,720)	154	(676,318)	222	(792,038)
Either browser	576	(6,024,101)	2,330	(32,580,059)	2,906	(38,604,160)

Each of the 792,038 Firefox users are uniquely identifiable

Revealed and susceptible to revelation attack?

	Revealed	Susceptible
Chromium	2,684	2,606 (97.09%)
Firefox	222	216 (97.30%)
Total	2,906	2,822 (97.11%)

Measures: Latex Gloves



(a) Probing defense

(b) Revelation defense

Measures: Latex Gloves

- Blacklists from browser vendors

Measures: Latex Gloves

- Blacklists from browser vendors
- Allow web pages to specify whitelists

Measures: Latex Gloves

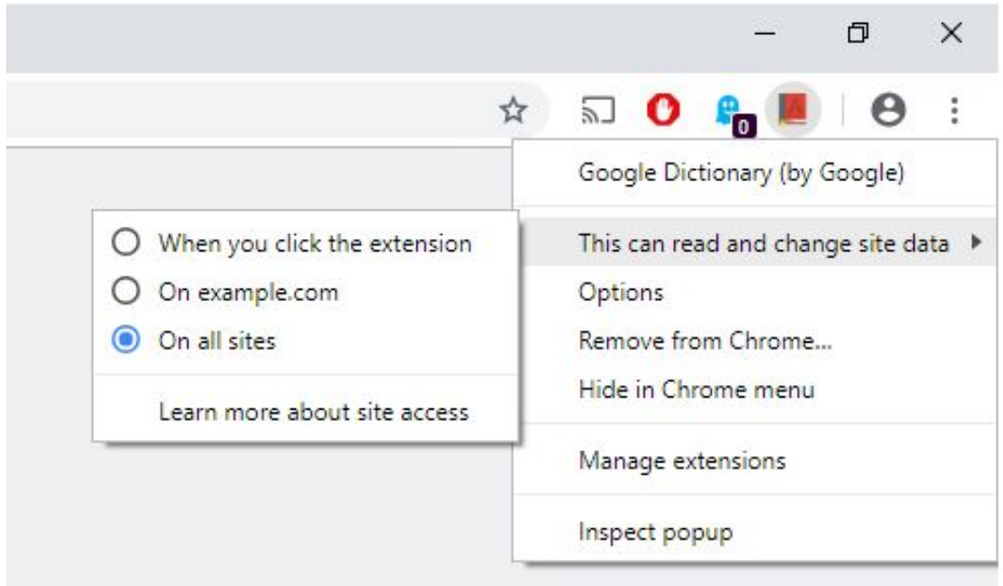
- Blacklists from browser vendors
- Allow web pages to specify whitelists
- Users classify web pages

Sensitive

Insensitive

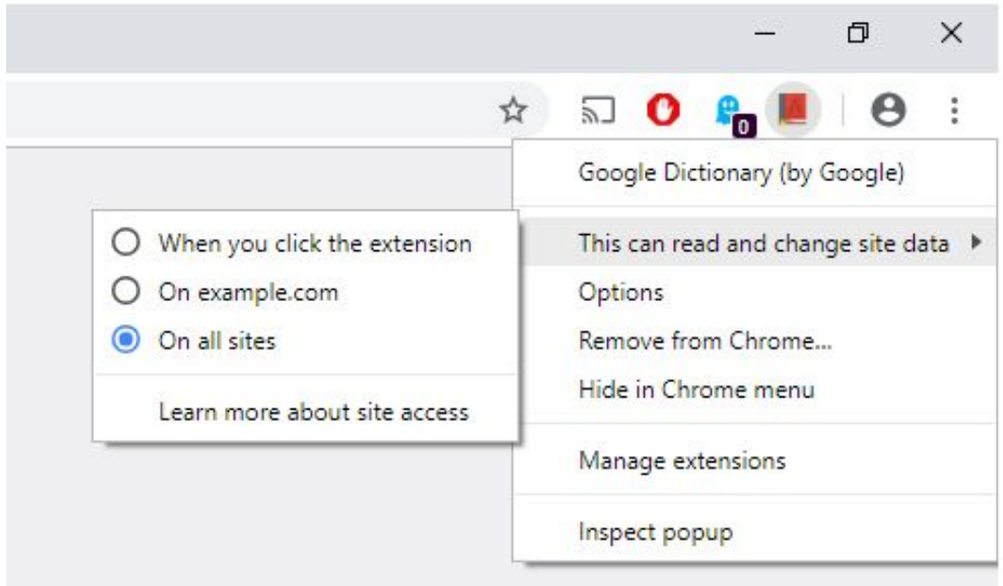
Countermeasures

- Long term
  - Latex Gloves



Countermeasures

- Long term
  - Latex Gloves
- Short term
  - Re-generate the random UUID more often
    - When starting the browser
  - Re-generate the random UUID when entering private browsing mode
  - Randomize the full URL, including the path
    - Helps, but is not perfect...
  - Use data URIs



Thank you!

Questions?