

BROWSER EXTENSION AND LOGIN-LEAK EXPERIMENT

APVP 2017, Vienna
Joint work with Nataliia Bielova, Claude
Castelluccia

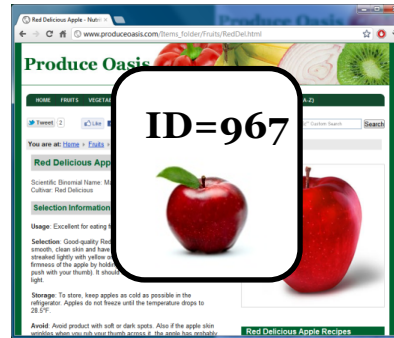
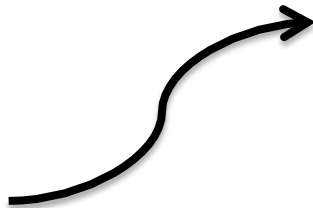
Gábor György Gulyás
Privatics Team, INRIA
<http://gulyas.info> // @GulyasGG



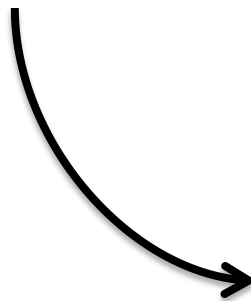
USER TRACKING ON THE WEB

The „de-facto” business model of the web

User



Advertiser

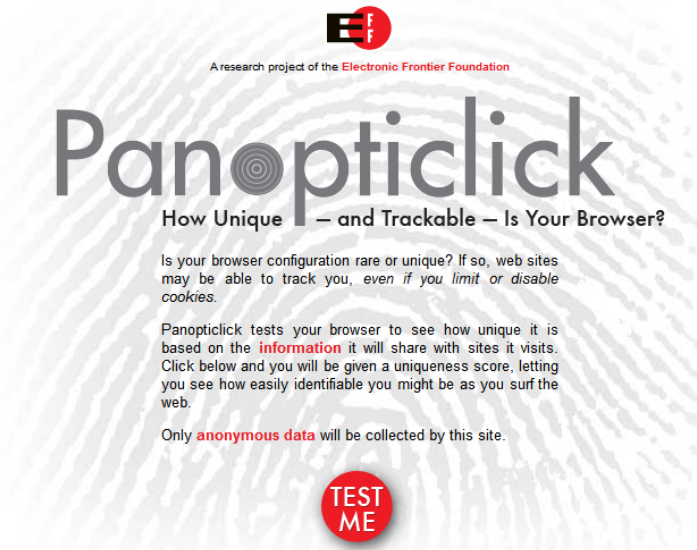


Storing the identifier on the client side

- Cookies
 - Flash
 - HTML5
- Caching in files of
 - JavaScript
 - CSS
 - Images (pixel-level)
- E-tags
- Last-mod timestamps
- HTTP authentication
- HTTP 301 redirect
- HSTS caches
- ...

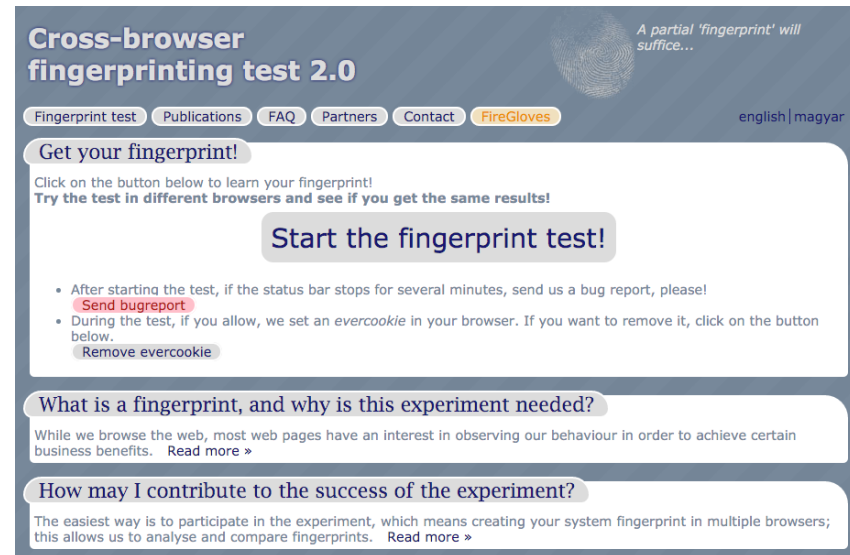


Browser fingerprinting appears (2010-2012) [3]



<http://panopticlick.eff.org>

- Browser fingerprint
 - Flash/Java required (for 95% uniqueness)
 - Browser dependent



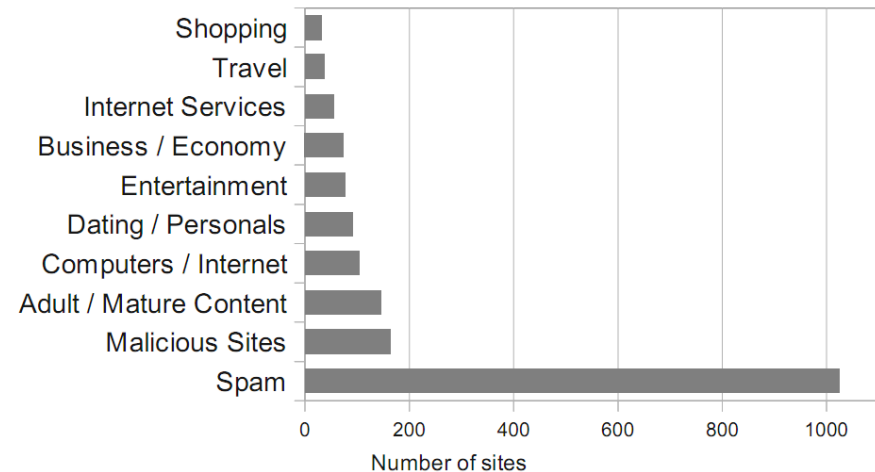
<https://fingerprint.pet-portal.eu>

- Cross-browser fingp.
 - Device fingerprint
 - No plugins, just JS
 - Concept appears later in the wild

Fingerprinting penetration (2013-2016)

2013: Alexa TOP 10k.

- 20 pages deep
- 0,4% adoption (40 sites)
- Skype.com, porn and dating
- 3 804 less popular sites are tracked



Nickiforakis et al.: Cookieless monster: Exploring the ecosystem of web-based device fingerprinting (2013)

2016: Alexa TOP 1M.



Rank Interval	% of First-parties		
	Canvas	Canvas Font	WebRTC
[0,1K)	5.10%	2.50%	0.60%
[1K,10K)	3.91%	1.98%	0.42%
[10K,100K)	2.45%	0.86%	0.19%
[100K,1M)	1.31%	0.25%	0.06%

S. Englehardt, A. Narayanan: Online tracking: A 1-illion-site measurement and analysis (2016)

Behavioral fingerprinting

You are what you install to you computer?

Fonts are good indicators of what is installed.



Boda et al.: User Tracking on the Web via Cross-Browser Fingerprinting (2011)

Google.com Youtube.com
Facebook.com Baidu.com
Yahoo.com Wikipedia.org
Google.co.in Qq.com Sohu.com
Google.co.jp Taobao.com
Tmall.com Live.com Amazon.com
Vk.com Twitter.com
Instagram.com 360.cn

The list of the sites you have visited also describe you well.

Can be used to de-anonymize you as a natural person.

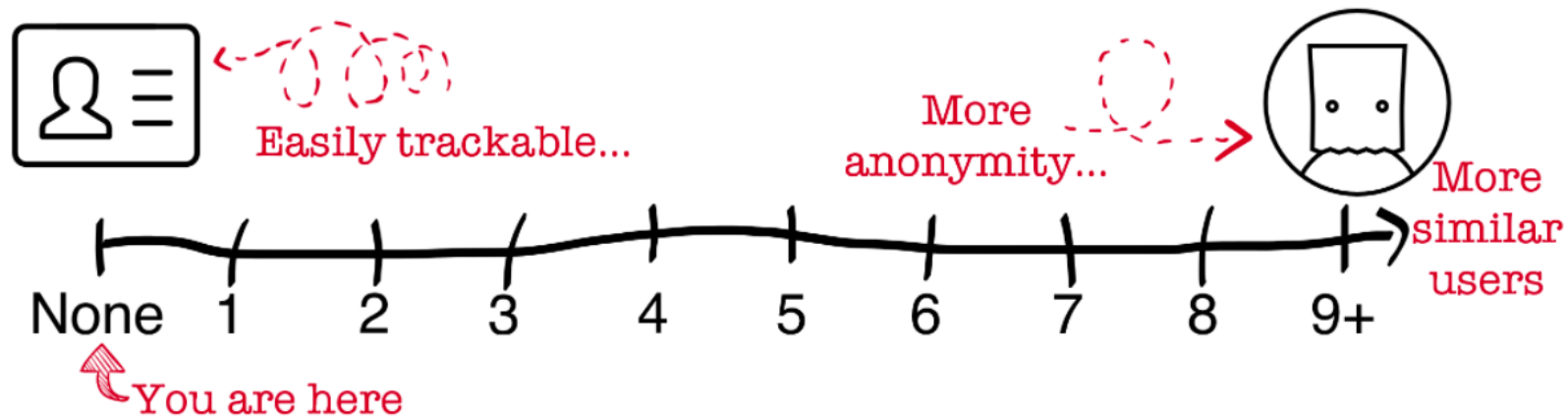
Su et al.: De-anonymizing Web Browsing Data with Social Networks (2017)



Browser Extension and Login-Leak Experiment

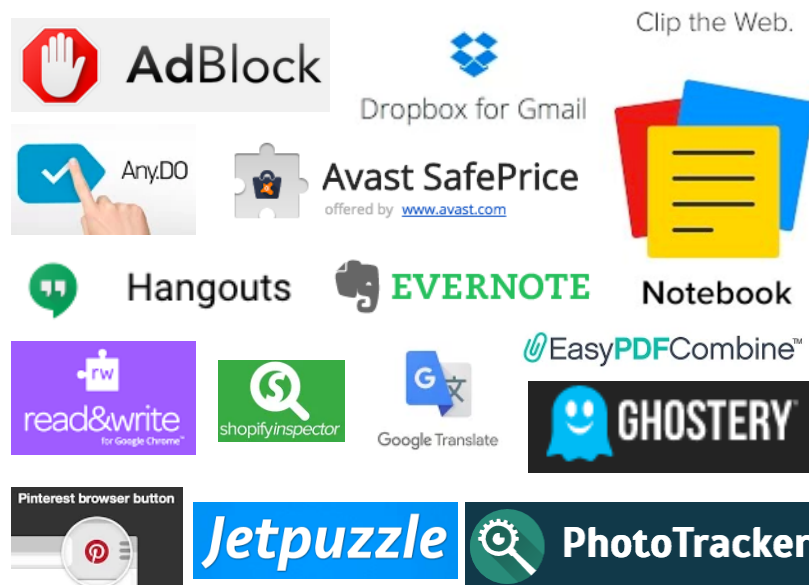
Are you identifiable?

Yes, you are identifiable, as there are no other users who looks like you among the 18498 users we tested so far:



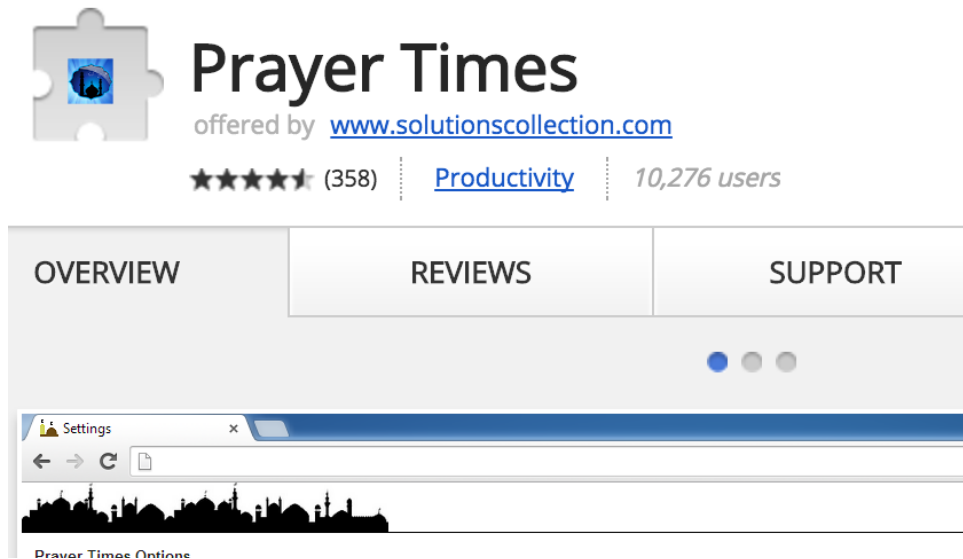
Browser Extension and Login-Leak Experiment

- Extension detection
 - Detecting extension resources
- Detecting web logins
 - Redirection URL hijacking
 - Misusing CSP violation



Why is this a problem?

Extensions can **leak private information!**



The more privacy extensions you install,
the **more identifiable you are!**

History of browser extension detection

ha.ckers

« [Google Redirection Hole Used For Phishing](#)
[CSS History Stealer](#)

Detecting FireFox Extentions

FRIDAY, FEBRUARY 17, 2012

Intro to Chrome addons

tldr; *Webpages can sometimes inject scripts later. Meanwhile, a warmup - trick*

While all of us are used to http / https schemes including:

- javascript: URIs [bypass](#)
- data: URIs that is a common
- view-source: that may be
- file: that reads your local



[BeEF Project Home](#) - [GitHub](#) - [Source Control](#) - [Bug Reporting](#) - [Wiki](#) - [Twitter](#) - [YouTube](#) - [LinkedIn](#) - [Mailing List](#)

Thursday, April 4, 2013

The Evolution of Chrome Extensions Detection

Today's blog post is by guest blogger Giovanni Cattani.

The Old Technique

Detecting which Chrome extensions are installed has always been a trivial matter of finding a specific extension ID and trying to load the *manifest.json* file, which is always located in the same spot:

```
chrome-extension://abcdefghijklmnopqrstuvwxyz012345/manifest.json
```

you should see a list of detected extensions; if any, again, I'm not capturing this data, just redisplaying it.

History of browser extension detection (2)

Discovering Browser Extensions via Web Accessible Resources

Alexander Sjösten
Chalmers University of
Technology
Gothenburg, Sweden
sjosten@chalmers.se

Steven Van Acker
Chalmers University of
Technology
Gothenburg, Sweden
svanacke@chalmers.se

Andrei Sabelfeld
Chalmers University of
Technology
Gothenburg, Sweden
sabelfeld@chalmers.se

ABSTRACT

Browser extensions provide a powerful platform to enhance the browsing experience. At the same time, they raise important security questions. From the point of view of a website, some browser extensions are invasive, removing intended features and adding unintended ones, e.g. extensions that block Facebook likes. Conversely, from the point of view of extensions, some websites are invasive, e.g. websites that pass ad blockers. Motivated by security goals at clash, this paper explores browser extension discovery, through a non-behavioral technique, based on detecting extensions' web accessible resources. We report on an empirical study of free Chrome and Firefox extensions, being able to detect over 50% of the top 1,000 free Chrome extensions, including popular security- and privacy-critical extensions such as Adblock, LastPass, Avast Online Security, and Ghostery. We also conduct an empirical study of non-behavioral extension detection on the Alexa top 100,000 websites. We present dual measures of making extension detection easier in the interest of websites and making extension detection more difficult in the interest of extensions. Finally, we discuss browser architecture that allows a user to take control over the conflicting security goals.

Non-behavioral extension detector

This web application attempts to detect which browser extensions you have installed.

Similar extension detectors traditionally use an indirect behavioral technique, attempting to detect an extension by observing its behavior. For instance, Adblock can be detected by injecting a fake advertisement and then detecting whether it was removed from the webpage.

This detector relies on a **non-behavioral technique** to directly reveal the existence of browser extensions, by querying browser extensions' **web accessible resources**. For instance, the Adblock extension in Chrome has a web accessible resource at <chrome-extension://gighmmpioibklfepjocnamgkkgbiglidom/icons/icon24.png>, which this detector probes for. If this web accessible resource is present, the extension is installed.

Extension signatures data was last updated on Dec 8, 2016 2:47:39 PM (3 months ago).

✓ Accept

Disclaimer

This webpage will **probe for several thousands of web accessible resources** in your browser. If you press the "Accept" button to the right, you give us **permission** to do this.

The results of this scan will not be shared with anyone, **we do not store any of the results**.

Because web extensions are updated frequently, their set of exposed web accessible resources may change over time. To keep up with these changes, **we update this webpage regularly**. If your extension is not detected, it may simply be because this webpage has not caught up with the latest version of the extension. Please try again later.

Press the "Accept" button on the right to start the scan.

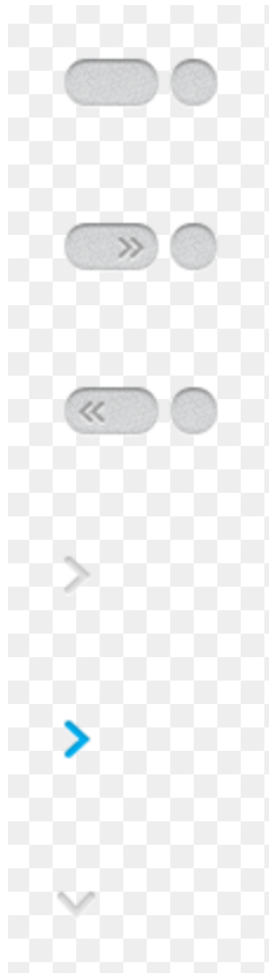
Scan thoroughness

How does it work?

chrome-extension://mlomiejdffkolichcflejclcbmpeanii/app/images/apps_pages/tracker.png

Extension ID
(Ghostery)

Local filepath



- Try yourself: <http://tinyurl.com/chrome-ghostery>
- High precision & coverage:
 - Large fraction of extensions covered ~28% (13 892)
 - No false-positives (uninstalled extensions not reported)
- Robustness (multiple resources can be checked)



Other browsers?

- Firefox



- Smaller impact: ~7% (direct possibility to manipulate UI)
- WebExtensions → same vulnerability as Chrome (but ~5.5%)
- Resources leak more information

- Opera



Browser extension details

Tested extensions:		5/12154
Detected extension		Detected resource
🔗 ghostery_opera		chrome-extension://bbkekonodcdmedgffkkgmnnekbainbg/app/images/panel/ghostery-i

- Brave



- Comes with detectable built-in extensions
- Test it here: <https://extensions.inrialpes.fr/brave/>

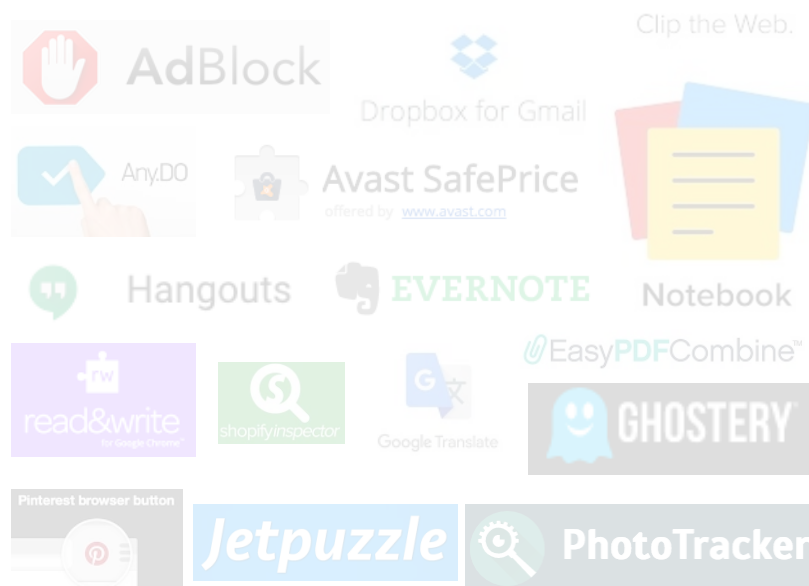
- Edge



- It is possible [<http://tinyurl.com/edge-ext>]
- Low number of extensions are available

Browser Extension and Login-Leak Experiment

- Extension detection
 - Detecting extension resources
- Detecting web logins
 - Redirection URL hijacking
 - Misusing CSP violation



Why is this a problem?

Allows very **precise profiling**.



Leaks **sensitive info (security!)**.

Tells about **where you work**.



Allow **behavioral tracking**.

Currently detected sites (60)

Social & Fun

- Battle.net
- Facebook
- Flickr
- Foursquare
- Gmail
- Google Plus
- Instagram
- LinkedIn
- Meetup
- Pinterest
- Skype
- Spotify
- Tumblr
- Twitter
- VK
- Youtube

Shopping

- 500px
- Alibaba.com, Aliexpress.com
- Airbnb
- Amazon.{co.uk, com, de, fr, it}
- eBay.{co.uk, com, de, fr, it}
- Expedia
- Paypal
- Photobucket
- shutterstock
- Steam
- Square

Gray zone

- Youporn
- Dating sites

News & Blogging

- Forbes
- Hackernews
- LeMonde.fr
- LiveJournal
- Medium
- Reddit
- Spiegel.de
- Yahoo

Work & Education

- Academia.edu
- BitBucket
- Carbonmade
- Dropbox
- EdX
- Evernote
- Github
- Indeed
- Inria
- Khan Academy
- PluralSight
- Scribd
- Slack
- SugarSync
- Viadeo

Techniques used

Your Social Media Fingerprint

Without your consent most major web platforms leak whether you are logged in. This allows any website to detect on which platforms you're signed up. Since there are lots of platforms with specific demographics an attacker could reason about your personality, too.

This project is an open source contribution of [RobinLinus](#) - Security, Privacy & Blockchain Consulting.

Demonstration

You are logged in to:



Redirection URL hijacking
by [@robin_linus](#)

Abusing Content
Security Policy
by [@homakov](#)

Monday, January 13, 2014

Using Content-Security-Policy for Evil

TL;DR How can we use technique created to protect websites for Evil? (We used [XSS Auditor](#) for Evil before) There's a neat way: taking advantage of CSP we can detect whether URL1 does redirect to URL2 and even bruteforce /path of URL2/path. This is a conceptual vulnerability in CSP design (violation == detection), and there's no obvious way to fix it.

Demo & playground: <http://homakov.github.io/csp.html>

How do they work?

Redirection URL hijacking

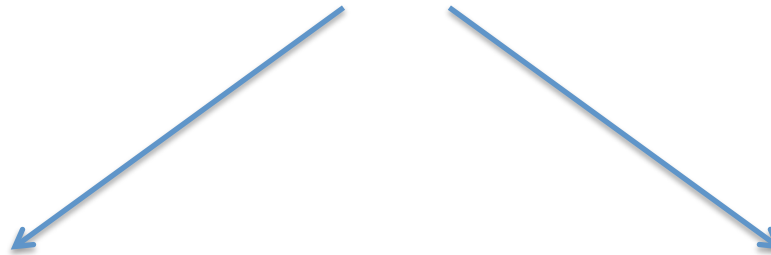
```
https://inria.fr/login?return=CALENDAR
```

How do they work? [2]

Redirection URL hijacking

`<img />`

`https://inria.fr/login?return=logo_INRIA.png`



Not logged in
(login page)



Logged in
(silent & unchecked
redirection to image)



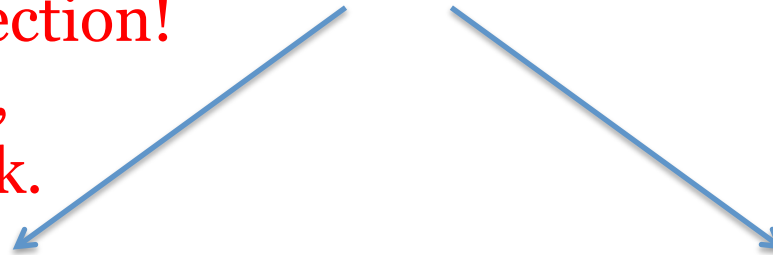
How do they work? [3]

Abusing CSP

`<img />`

`http://my.ebay.com`

Not allowed redirection!
Raises error,
reports it back.



Not logged in
(`http://www.ebay.com`)

Logged in
(`http://my.ebay.com`)



<https://extensions.inrialpes.fr>



Browser Extension and Login-Leak Experiment

When you browse the web, **small beacons** (trackers) are spying on your online activities. Even though such trackers are invisible, they collect information about you such as which pages you visit, which buttons clicked, and what text you typed. This information is often used to show you **targeted advertisements** and **may require you to pay a higher price during online shopping** depending on the collected information.

Did you know websites can track you by your browser extensions and web logins?

Recent studies show that you can be tracked **based on your web browser properties**. In this experiment, we demonstrate that you can also be tracked by

- your browser extensions (such as AdBlock, Pinterest, or Ghostery), and
- the websites you have logged in (such as Facebook, Gmail, or Twitter).

You can learn more here about how these detection techniques work.

In the experiment, we will collect your browser fingerprint, together with the browser extensions installed and a list of websites you have logged in. We only collect anonymous data during the experiment (see our **Privacy Policy**), we will securely store the data on an Inria server, use it only for research purpose and not share it with anyone outside of Inria. You can also read **the frequently asked questions here**.

☒ Test which websites I am logged into. Your browser will silently visit **these sites**.

NEW What is your relation to computers? (we would like to see whether our dataset is biased)

☐ Computer scientist or geek. ☐ Regular computer user. ☒ I don't want to declare.

☒ I agree, test my browser!

What could we do (for now)?

Extension detection

- Chrome, Opera, Brave: not much.
- Safari: not evaluated.
- Firefox: vulnerable.
But: few extensions, and good for privacy.



History

Firefox will: Use custom settings for history ▼

☐ Always use private browsing mode

☒ Remember my browsing and download history

☒ Remember search and form history

☒ Accept cookies from sites

Accept third-party cookies: Never ▼

Keep until: they expire ▼

Web login detection

- Best advice is to turn off third-party cookies.
- Or use an extension that blocks
 - access to third-part cookies,
 - tracking, or
 - JavaScript (noscript).

Research issues

- Assess privacy risks posed by these leaks
 - Uniqueness / tracking
 - De-anonymization attacks
 - Security risks
 - ...
- Evaluation of extension use and login habits
 - On a wider scale, and also
 - On the personal level (returning visitors)
- Longitudinal study of extensions

Research issues (2)

XHOUND: Quantifying the Fingerprintability of Browser Extensions



Similar
roots!

Ilia Starov
Stony Brook University
istarov@stonybrook.edu

Nick Nikiforakis
Stony Brook University
nick@cs.stonybrook.edu

Abstract—In recent years, researchers have shown that unwanted web tracking is on the rise, as advertisers are trying to capitalize on users' online activity, using increasingly intrusive and sophisticated techniques. Among these, browser fingerprinting has received the most attention since it allows trackers to uniquely identify users despite the clearing of cookies and the use of a browser's private mode.

In this paper, we investigate and quantify the fingerprintability of browser extensions, such as, AdBlock and Ghostery. We show that an extension's organic activity in a page's DOM can be used to infer its presence, and develop XHOUND, the first fully automated system for fingerprinting browser extensions. By applying XHOUND to the 10,000 most popular Google Chrome extensions, we find that a significant fraction of popular browser extensions are fingerprintable and could thus be used to sup-

browser plugins remain one of the most revealing features of desktop and laptop browsers, the entropy that they offer has considerably decreased since Eckersley's study. This is due to the increasing popularity of HTML5, providing powerful functionality to developers, and the decreasing trust in proprietary browser plugins which have, in the past, caused many performance and security issues. For instance, Google Chrome has stopped supporting plugins utilizing the NPAPI architecture [3] and Mozilla Firefox announced a plan to do the same [5], while already limiting the ability of scripts to enumerate plugins [1]. As the popularity and fingerprinting power of plugins is decreasing, we argue that trackers will look for new techniques to fingerprint users. Predicting these

Thank you for your attention!

ANY QUESTIONS?

Gábor György Gulyás

Privatics Team, INRIA

<http://gulyas.info> // [@GulyasGG](#)

