Detecting Malicious or Fingerprintable Browser Extensions

Dr.-Ing. Aurore Fass

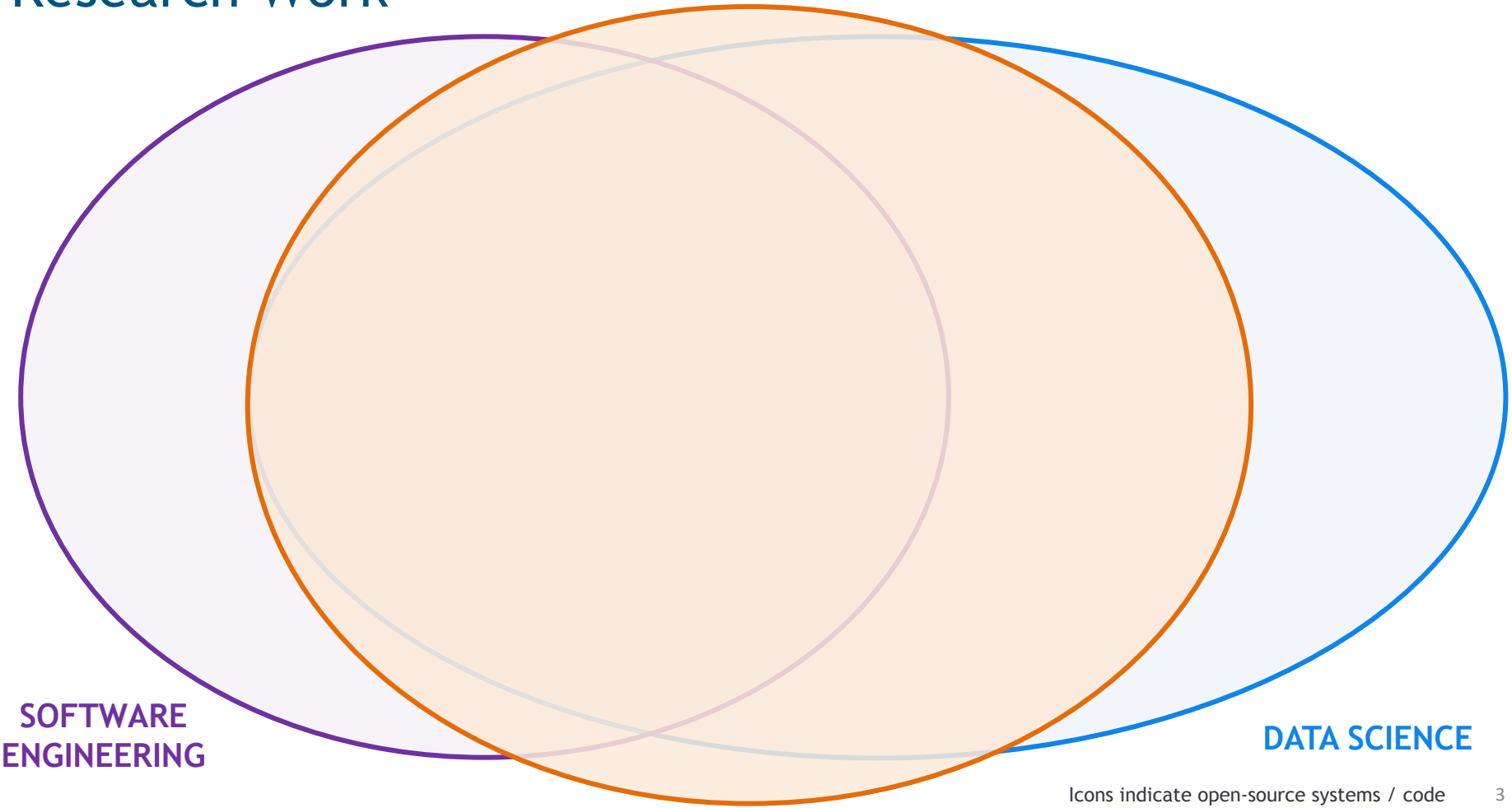
Tenured Researcher — Inria Centre at Université Côte d'Azur (since Dec 1, 2025)

Privaski Seminar, March 11th 2026

Dr.-Ing. Aurore (/ɔʀɔʀ/) FASS

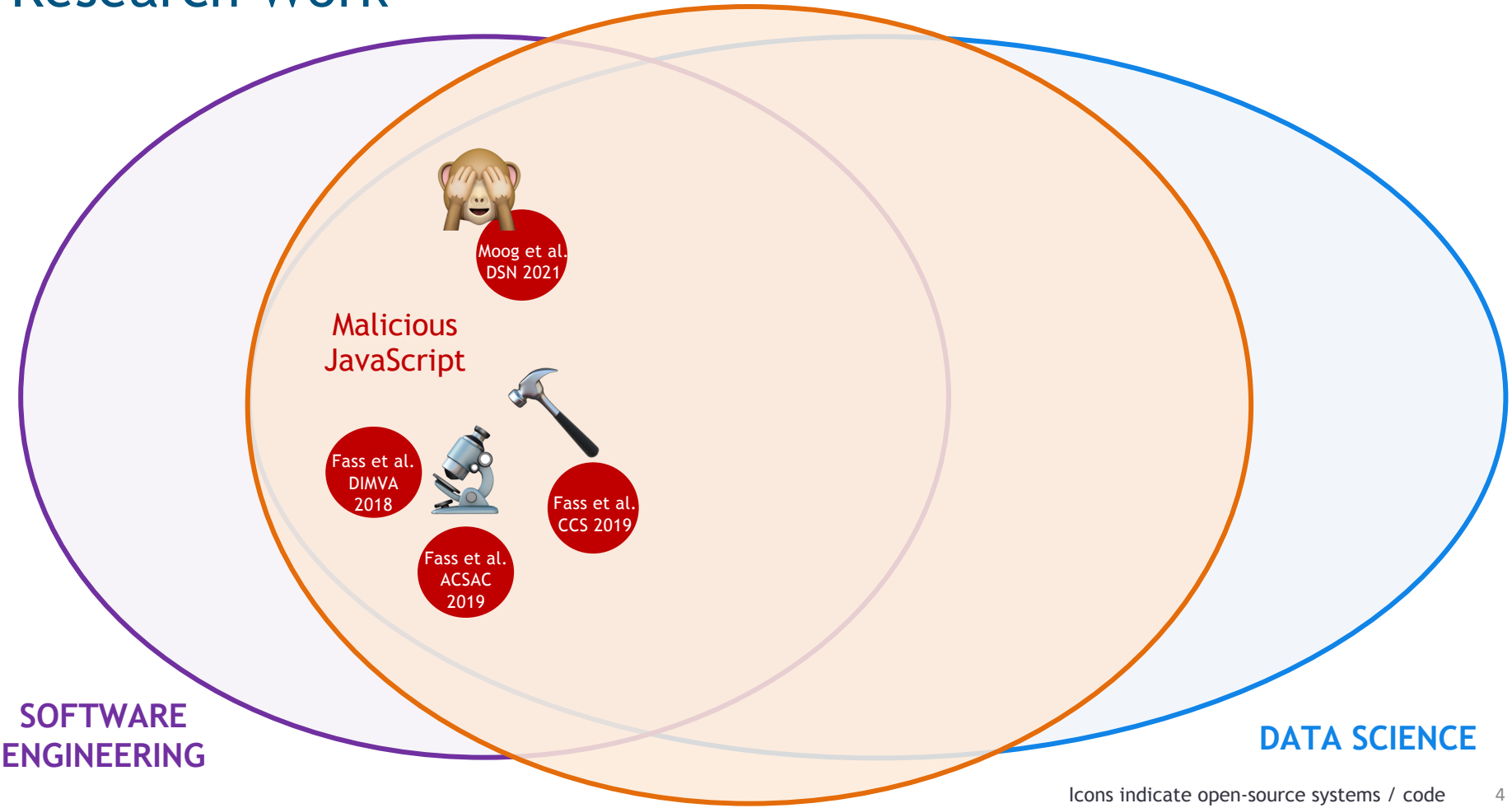
- 🇫🇷 2017: Graduated from TELECOM Nancy (**FR**)
- 🇩🇪 2017—21: PhD Student + Postdoc at CISPA (**DE**)
- 🇺🇸 2021—23: Visiting Assistant Professor at Stanford (**US**)
- 🇩🇪 2023—25: Tenure-Track Faculty W2 at CISPA (**DE**)
- 🇫🇷 2025—: Tenured Researcher at Inria (**FR**)

Research Work



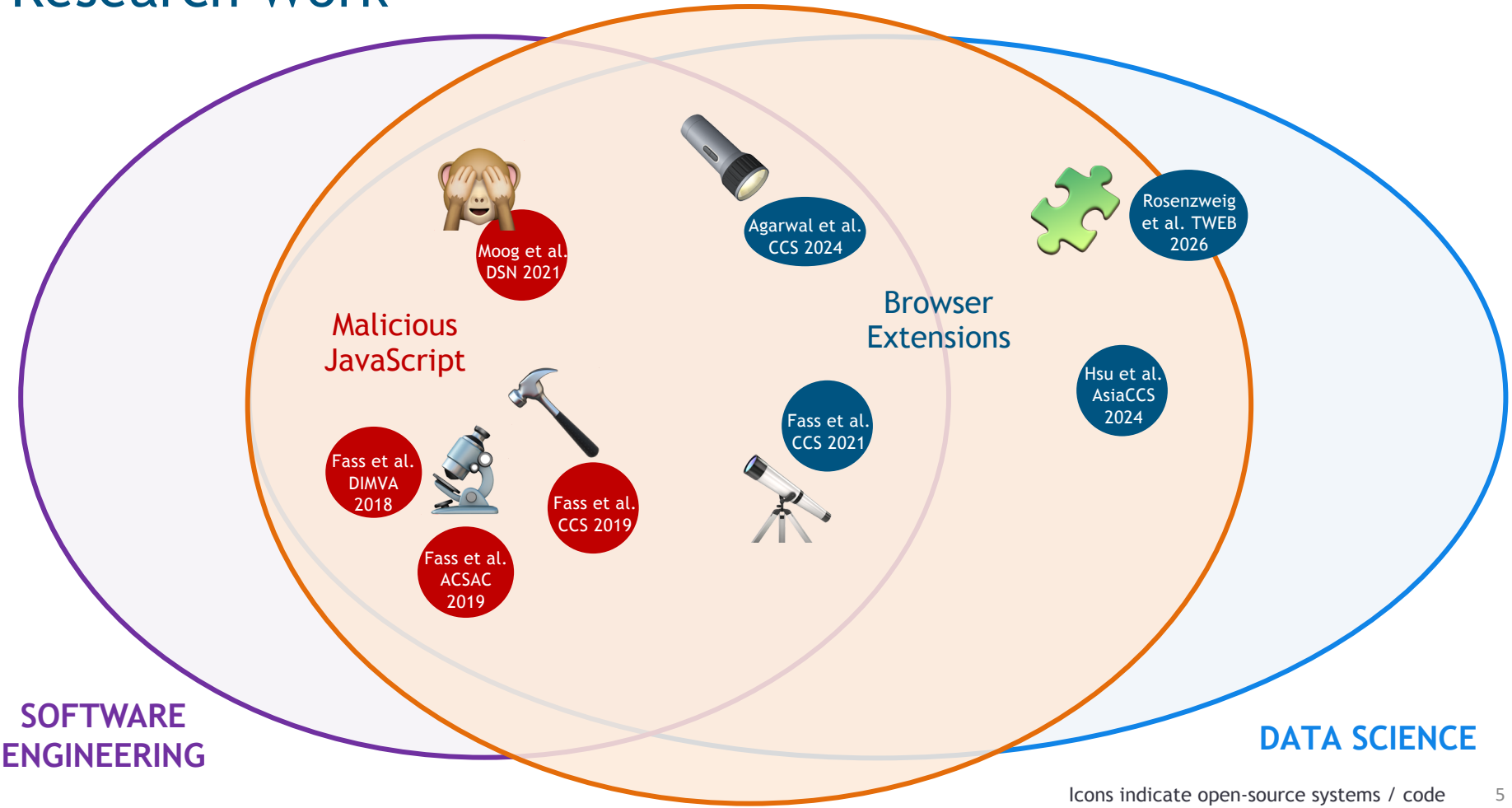
Icons indicate open-source systems / code

Research Work



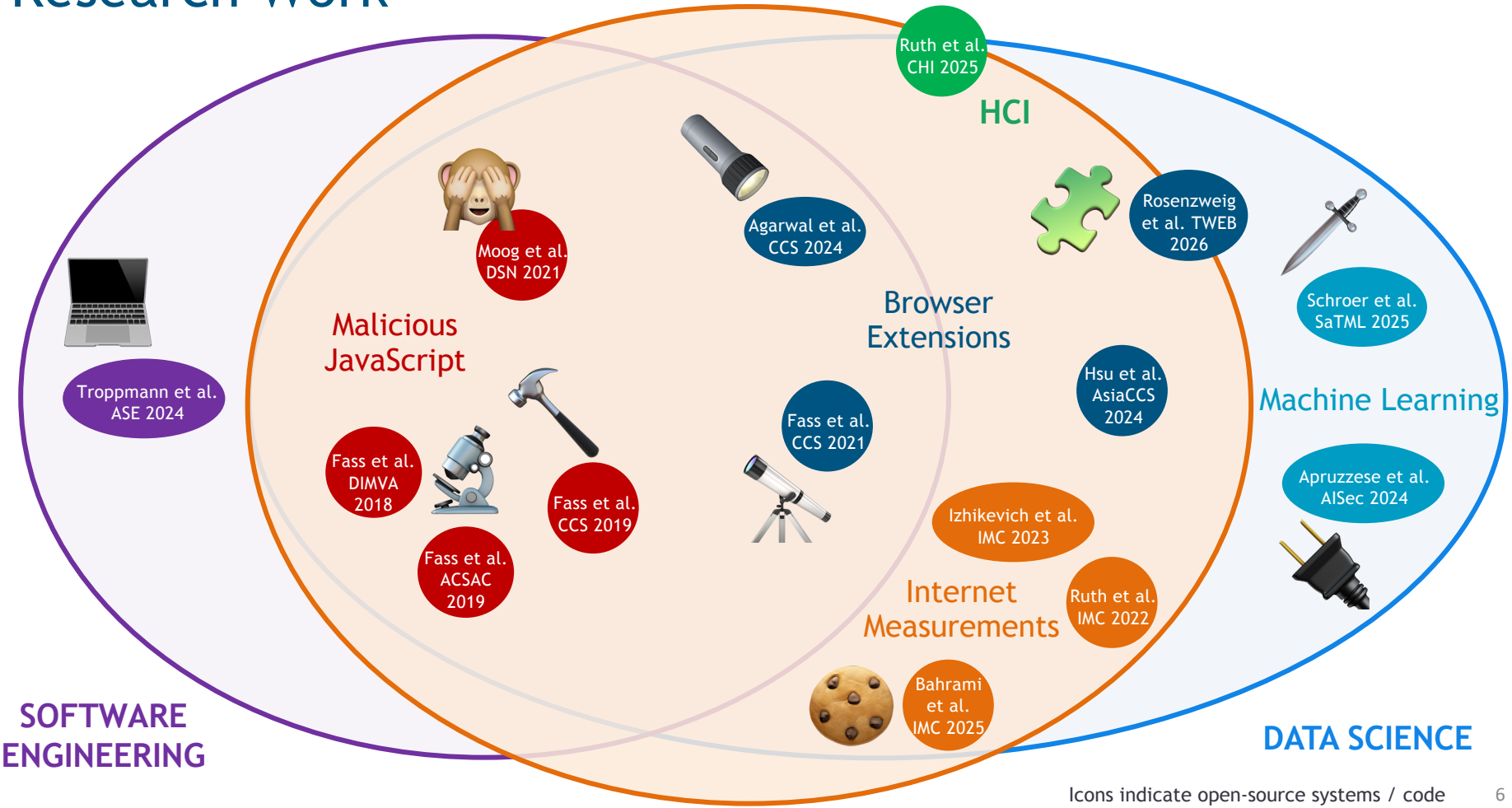
Icons indicate open-source systems / code

Research Work



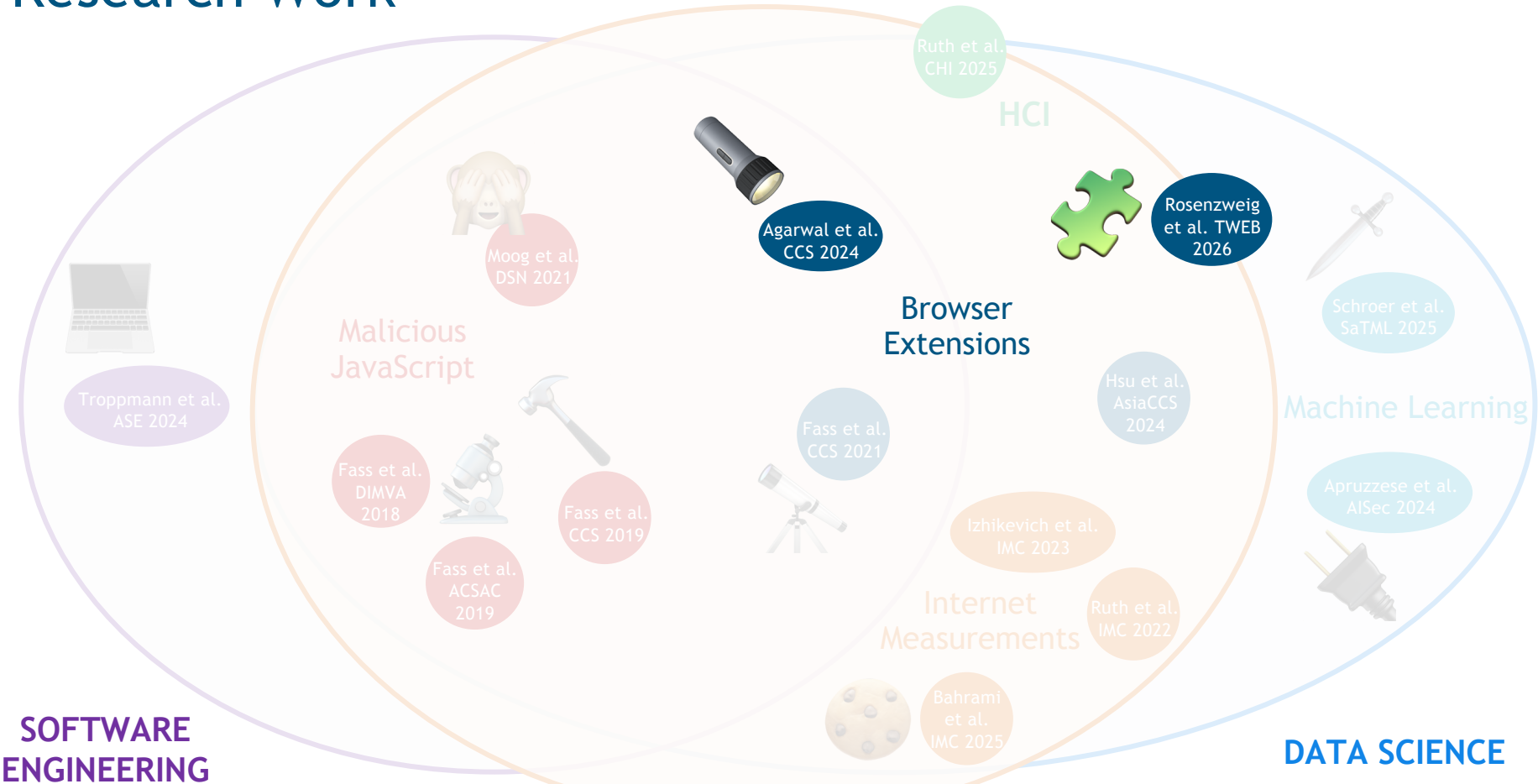
Icons indicate open-source systems / code

Research Work



Icons indicate open-source systems / code

Research Work



Icons indicate open-source systems / code

What are Browser Extensions?

- Third-party programs to **improve user browsing experience**

Adblock Plus - free ad blocker
Offered by: adblockplus.org

Skype
Offered by: www.skype.com

Adobe Acrobat
Offered by: Adobe Inc.

Grammarly for Chrome
Offered by: grammarly.com

Honey
Offered by: https://www.joinhoney.com

Google Translate
Offered by: translate.google.com

LastPass: Free Password Manager
Offered by: LastPass

Cisco Webex Extension
Offered by: webex.com

- **~200k** Chrome extensions totaling almost **2B** active users
  - ➢ Necessitates a thorough **security and privacy assessment**

How Secure are Browser Extensions?

- Browser extensions provide additional functionality…
- … so browser extensions need additional & elevated privileges compared to web pages
  - e.g., ad-blockers need to modify web page content or intercept network requests

➢ Browser extensions are an attractive target for attackers 😈

Dangerous Browser Extensions

→ Extensions can put their users' security & privacy at risk:

- **Contain malware**: designed by malicious actors to harm victims
  - E.g., propagate malware, steal users' credentials, track users [1, 3]
- **Contain vulnerabilities**: designed by well-intentioned developers… but buggy
  - E.g., can lead to user-sensitive data exfiltration [1, 2]
- **Violate the Chrome Web Store policies**
  - E.g., deceive users, promote unlawful activities, lack a privacy policy [1]
- **Be fingerprintable**: can be recognized and uniquely identified
  - E.g., can lead to user tracking or inferring of user personal information [4]

[1] Hsu, Tran, and Fass; *AsiaCCS* 2024 | [2] Fass, Somé, Backes, and Stock; *CCS* 2021 | [3] Rosenzweig, Dalla Valle, Apruzzese, and Fass; *TWEB* 2026 | [4] Agarwal, Fass, and Stock; *CCS* 2024

Media Coverage

Forbes

FORBES > INNOVATION > CYBERSECURITY

280 Million Google Chrome Users Installed Dangerous Extensions, Study Says

Davey Winder Senior Contributor

Davey Winder is a veteran cybersecurity writer, hacker and analyst.

Follow

Jun 24, 2024, 06:57am EDT

How safe are Google Chrome extensions? SOPA IMAGES/LIGHTROCKET VIA GETTY IMAGES

The Register

Risk of installing dodgy extensions from Chrome store way worse than Google's letting on, study suggests

All depends on how you count it – Chocolate Factory claims 1% fail rate

Thomas Claburn

Sun 23 Jun 2024 // 10:36 UTC

ADGUARD

Subscribe to news

Search blog

AdGuard > Blog > Google is failing miserably at weeding out bad extensions, new research indicates

Google is failing miserably at weeding out bad extensions, new research indicates

July 5, 2024 · 7 min read

TECHSPOT

TRENDING FEATURES REVIEWS THE BEST DOWNLOADS PRODUCT FINDER FORUMS

SECURITY THE WEB MALWARE CHROME

Researchers say 280 million people have installed malware-infected Chrome extensions in the last 3 years

Google claims less than 1% of all installs include malware

By Rob Thubron June 24, 2024 at 11:39 AM

[1] Hsu, Tran, and Fass; *AsiaCCS* 2024

Dangerous Browser Extensions

→ Extensions can put their users' security & privacy at risk:

- **Contain malware**: designed by malicious actors to harm victims
  - E.g., propagate malware, steal users' credentials, track users [1, 3]
- **Contain vulnerabilities**: designed by well-intentioned developers… but buggy
  - E.g., can lead to user-sensitive data exfiltration [1, 2]
- **Violate the Chrome Web Store policies**
  - E.g., deceive users, promote unlawful activities, lack a privacy policy [1]
- **Be fingerprintable**: can be recognized and uniquely identified
  - E.g., can lead to user tracking or inferring of user personal information [4]

[1] Hsu, Tran, and Fass; *AsiaCCS* 2024 | [2] Fass, Somé, Backes, and Stock; *CCS* 2021 | [3] Rosenzweig, Dalla Valle, Apruzzese, and Fass; *TWEB* 2026 | [4] Agarwal, Fass, and Stock; *CCS* 2024

Dangerous Browser Extensions

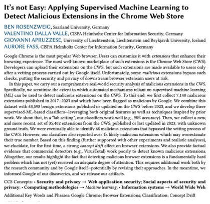
→ Extensions can put their users' security & privacy at risk:

- **Contain malware**: designed by malicious actors to harm victims
  - E.g., propagate malware, steal users' credentials, track users [1, 3]
- **Contain vulnerabilities**: designed by well-intentioned developers… but buggy
  - E.g., can lead to user-sensitive data exfiltration [1, 2]
- **Violate the Chrome Web Store policies**
  - E.g., deceive users, promote unlawful activities, lack a privacy policy [1]
- **Be fingerprintable**: can be recognized and uniquely identified
  - E.g., can lead to user tracking or inferring of user personal information [4]

[1] Hsu, Tran, and Fass; *AsiaCCS* 2024 | [2] Fass, Somé, Backes, and Stock; *CCS* 2021 | [3] Rosenzweig, Dalla Valle, Apruzzese, and Fass; *TWEB* 2026 | [4] Agarwal, Fass, and Stock; *CCS* 2024

Detecting Malicious Extensions — Lab Setting

Rosenzweig et al. TWEB 2026



It's not Easy: Applying Supervised Machine Learning to Detect Malicious Extensions in the Chrome Web Store

BEN ROSENZWEIG, Saarland University, Germany
VALENTINO DALLA VALLE, CISPA Helmholtz Center for Information Security, Germany
GIOVANNI APRUZZESE, University of Liechtenstein, Liechtenstein and Reykjavik University, Iceland
AURORE FASS, CISPA Helmholtz Center for Information Security, Germany

Google Chrome is the most popular Web browser. Users can customize it with *extensions* that enhance their browsing experience. The most well-known marketplace of such extensions is the Chrome Web Store (CWS). Developers can upload their extensions on the CWS, but such extensions are made available to users only after a vetting process carried out by Google itself. Unfortunately, some *malicious* extensions bypass such checks, putting the security and privacy of downstream browser extension users at risk.

In this paper, we carry out a comprehensive real-world security analysis of malicious extensions in the CWS. Specifically, we scrutinize the extent to which automated mechanisms reliant on supervised machine learning (ML) can be used to detect malicious extensions on the CWS. To this end, we first collect 7,140 malicious extensions published in 2017–2023 and which have been flagged as malicious by Google. We combine this dataset with 63,598 benign extensions published or updated on the CWS before 2023, and we develop three supervised-ML-based classifiers—leveraging both original features as well as techniques inspired by prior work. We show that, in a "lab setting", our classifiers work well (e.g., 98% accuracy). Then, we collect a new, and more recent, set of 35,462 extensions from the CWS, published or last updated in 2023, with unknown ground truth. We were eventually able to identify 68 malicious extensions that bypassed the vetting process of the CWS. However, our classifiers also reported over 1k likely malicious extensions which may overestimate their true number. Based on this finding (further supported with other experiments and realistic analyses), we elucidate, for the first time, a strong *concept drift* effect on browser extensions. We also provide factual evidence that commercial detectors (e.g., VirusTotal) work poorly to detect known malicious extensions. Altogether, our results highlight the fact that detecting malicious browser extensions is a fundamentally hard problem which has not (yet) received an adequate degree of attention. This requires additional work both by the research community and by Google itself—potentially by revising their approaches. In the meantime, we informed Google of our discoveries, and we release our artifacts.

CCS Concepts: • **Security and privacy → Web application security; Social aspects of security and privacy**; • **Computing methodologies →** *Machine learning*; • **Information systems → World Wide Web**.

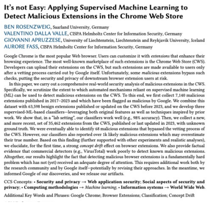
Additional Key Words and Phrases: Google Chrome; Browser Extensions; Classification; Concept Drift

> It's not Easy: Applying Supervised ML to Detect Malicious Extensions in the CWS

In *ACM TWEB* 2026. Ben Rosenzweig, Valentino Dalla Valle, Giovanni Apruzzese, and <u>Aurore Fass</u>

Detecting Malicious Extensions — Lab Setting

Rosenzweig et al. TWEB 2026


It's not Easy: Applying Supervised Machine Learning to Detect Malicious Extensions in the Chrome Web Store

BEN ROSENZWEIG, Saarland University, Germany
VALENTINO DALLA VALLE, CISPA Helmholtz Center for Information Security, Germany
GIOVANNI APRUZZESE, University of Liechtenstein, Liechtenstein and Reykjavik University, Iceland
AURORE FASS, CISPA Helmholtz Center for Information Security, Germany

Google Chrome is the most popular Web browser. Users can customize it with *extensions* that enhance their browsing experience. The most well-known marketplace of such extensions is the Chrome Web Store (CWS). Developers can upload their extensions on the CWS, but such extensions are made available to users only after a vetting process carried out by Google itself. Unfortunately, some *malicious* extensions bypass such checks, putting the security and privacy of downstream browser extension users at risk.

In this paper, we carry out a comprehensive real-world security analysis of malicious extensions in the CWS. Specifically, we scrutinize the extent to which automated mechanisms reliant on supervised machine learning (ML) can be used to detect malicious extensions on the CWS. To this end, we first collect 7,140 malicious extensions published in 2017–2023 and which have been flagged as malicious by Google. We combine this dataset with 63,598 benign extensions published or updated on the CWS before 2023, and we develop three supervised-ML-based classifiers—leveraging both original features as well as techniques inspired by prior work. We show that, in a "lab setting", our classifiers work well (e.g., 98% accuracy). Then, we collect a new, and more recent, set of 35,462 extensions from the CWS, published or last updated in 2023, with unknown ground truth. We were eventually able to identify 68 malicious extensions that bypassed the vetting process of the CWS. However, our classifiers also reported over 1k likely malicious extensions which may overestimate their true number. Based on this finding (further supported with other experiments and realistic analyses), we elucidate, for the first time, a strong *concept drift* effect on browser extensions. We also provide factual evidence that commercial detectors (e.g., VirusTotal) work poorly to detect known malicious extensions. Altogether, our results highlight the fact that detecting malicious browser extensions is a fundamentally hard problem which has not (yet) received an adequate degree of attention. This requires additional work both by the research community and by Google itself—potentially by revising their approaches. In the meantime, we informed Google of our discoveries, and we release our artifacts.

CCS Concepts: • **Security and privacy** → **Web application security**; **Social aspects of security and privacy**; • **Computing methodologies** → *Machine learning*; • **Information systems** → **World Wide Web**.

Additional Key Words and Phrases: Google Chrome; Browser Extensions; Classification; Concept Drift

> It's not Easy: Applying Supervised ML to Detect Malicious Extensions in the CWS

In *ACM TWEB* 2026. Ben Rosenzweig, Valentino Dalla Valle, Giovanni Apruzzese, and Aurore Fass

DATASET

7,140 malicious extensions

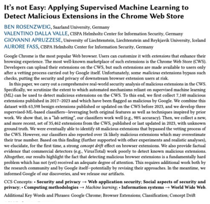
Last update 2017—22

63,598 benign extensions

Last update 2012—22

Detecting Malicious Extensions — Lab Setting

Rosenzweig et al. TWEB 2026


It's not Easy: Applying Supervised Machine Learning to Detect Malicious Extensions in the Chrome Web Store

BEN ROSENZWEIG, Saarland University, Germany
VALENTINO DALLA VALLE, CISPA Helmholtz Center for Information Security, Germany
GIOVANNI APRUZZESE, University of Liechtenstein, Liechtenstein and Reykjavik University, Iceland
AURORE FASS, CISPA Helmholtz Center for Information Security, Germany

Google Chrome is the most popular Web browser. Users can customize it with *extensions* that enhance their browsing experience. The most well-known marketplace of such extensions is the Chrome Web Store (CWS). Developers can upload their extensions on the CWS, but such extensions are made available to users only after a vetting process carried out by Google itself. Unfortunately, some *malicious* extensions bypass such checks, putting the security and privacy of downstream browser extension users at risk.

In this paper, we carry out a comprehensive real-world security analysis of malicious extensions in the CWS. Specifically, we scrutinize the extent to which automated mechanisms reliant on supervised machine learning (ML) can be used to detect malicious extensions on the CWS. To this end, we first collect 7,140 malicious extensions published in 2017–2023 and which have been flagged as malicious by Google. We combine this dataset with 63,598 benign extensions published or updated on the CWS before 2023, and we develop three supervised-ML-based classifiers—leveraging both original features as well as techniques inspired by prior work. We show that, in a "lab setting", our classifiers work well (e.g., 98% accuracy). Then, we collect a new, and more recent, set of 35,462 extensions from the CWS, published or last updated in 2023, with unknown ground truth. We were eventually able to identify 68 malicious extensions that bypassed the vetting process of the CWS. However, our classifiers also reported over 1k likely malicious extensions which may overestimate their true number. Based on this finding (further supported with other experiments and realistic analyses), we elucidate, for the first time, a strong *concept drift* effect on browser extensions. We also provide factual evidence that commercial detectors (e.g., VirusTotal) work poorly to detect known malicious extensions. Altogether, our results highlight the fact that detecting malicious browser extensions is a fundamentally hard problem which has not (yet) received an adequate degree of attention. This requires additional work both by the research community and by Google itself—potentially by revising their approaches. In the meantime, we informed Google of our discoveries, and we release our artifacts.

CCS Concepts: • **Security and privacy** → **Web application security**; **Social aspects of security and privacy**; • **Computing methodologies** → *Machine learning*; • **Information systems** → **World Wide Web**.

Additional Key Words and Phrases: Google Chrome; Browser Extensions; Classification; Concept Drift

> It's not Easy: Applying Supervised ML to Detect Malicious Extensions in the CWS

In *ACM TWEB* 2026. Ben Rosenzweig, Valentino Dalla Valle, Giovanni Apruzzese, and Aurore Fass

DATASET

7,140 malicious extensions

Last update 2017—22

63,598 benign extensions

Last update 2012—22

FEATURES

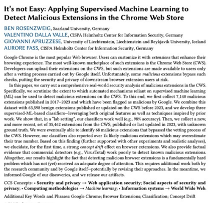
Source code [A]

Metadata

[A] Fass et al.; *ACSAC* 2018

Detecting Malicious Extensions — Lab Setting

Rosenzweig et al. TWEB 2026



It's not Easy: Applying Supervised Machine Learning to Detect Malicious Extensions in the Chrome Web Store

BEN ROSENZWEIG, Saarland University, Germany
VALENTINO DALLA VALLE, CISPA Helmholtz Center for Information Security, Germany
GIOVANNI APRUZZESE, University of Liechtenstein, Liechtenstein and Reykjavik University, Iceland
AURORE FASS, CISPA Helmholtz Center for Information Security, Germany

Google Chrome is the most popular Web browser. Users can customize it with *extensions* that enhance their browsing experience. The most well-known marketplace of such extensions is the Chrome Web Store (CWS). Developers can upload their extensions on the CWS, but such extensions are made available to users only after a vetting process carried out by Google itself. Unfortunately, some *malicious* extensions bypass such checks, putting the security and privacy of downstream browser extension users at risk.

In this paper, we carry out a comprehensive real-world security analysis of malicious extensions in the CWS. Specifically, we scrutinize the extent to which automated mechanisms reliant on supervised machine learning (ML) can be used to detect malicious extensions on the CWS. To this end, we first collect 7,140 malicious extensions published in 2017–2023 and which have been flagged as malicious by Google. We combine this dataset with 63,598 benign extensions published or updated on the CWS before 2023, and we develop three supervised-ML-based classifiers—leveraging both original features as well as techniques inspired by prior work. We show that, in a "lab setting", our classifiers work well (e.g., 98% accuracy). Then, we collect a new, and more recent, set of 35,462 extensions from the CWS, published or last updated in 2023, with unknown ground truth. We were eventually able to identify 68 malicious extensions that bypassed the vetting process of the CWS. However, our classifiers also reported over 1k likely malicious extensions which may overestimate their true number. Based on this finding (further supported with other experiments and realistic analyses), we elucidate, for the first time, a strong *concept drift* effect on browser extensions. We also provide factual evidence that commercial detectors (e.g., VirusTotal) work poorly to detect known malicious extensions. Altogether, our results highlight the fact that detecting malicious browser extensions is a fundamentally hard problem which has not (yet) received an adequate degree of attention. This requires additional work both by the research community and by Google itself—potentially by revising their approaches. In the meantime, we informed Google of our discoveries, and we release our artifacts.

CCS Concepts: • **Security and privacy → Web application security; Social aspects of security and privacy**; • **Computing methodologies →** *Machine learning*; • **Information systems → World Wide Web**.

Additional Key Words and Phrases: Google Chrome; Browser Extensions; Classification; Concept Drift

> It's not Easy: Applying Supervised ML to Detect Malicious Extensions in the CWS

In *ACM TWEB* 2026. Ben Rosenzweig, Valentino Dalla Valle, Giovanni Apruzzese, and Aurore Fass

DATASET

7,140 malicious extensions

Last update 2017—22

63,598 benign extensions

Last update 2012—22

FEATURES

Source code [A]

Metadata

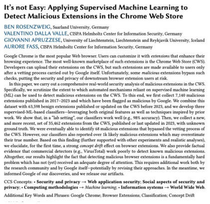
[A] Fass et al.; *ACSAC* 2018

DETECTORS

80% train | 20% test

Detecting Malicious Extensions — Lab Setting

Rosenzweig et al. TWEB 2026


It's not Easy: Applying Supervised Machine Learning to Detect Malicious Extensions in the Chrome Web Store

BEN ROSENZWEIG, Saarland University, Germany
VALENTINO DALLA VALLE, CISPA Helmholtz Center for Information Security, Germany
GIOVANNI APRUZZESE, University of Liechtenstein, Liechtenstein and Reykjavik University, Iceland
AURORE FASS, CISPA Helmholtz Center for Information Security, Germany

Google Chrome is the most popular Web browser. Users can customize it with *extensions* that enhance their browsing experience. The most well-known marketplace of such extensions is the Chrome Web Store (CWS). Developers can upload their extensions on the CWS, but such extensions are made available to users only after a vetting process carried out by Google itself. Unfortunately, some *malicious* extensions bypass such checks, putting the security and privacy of downstream browser extension users at risk.

In this paper, we carry out a comprehensive real-world security analysis of malicious extensions in the CWS. Specifically, we scrutinize the extent to which automated mechanisms reliant on supervised machine learning (ML) can be used to detect malicious extensions on the CWS. To this end, we first collect 7,140 malicious extensions published in 2017–2023 and which have been flagged as malicious by Google. We combine this dataset with 63,598 benign extensions published or updated on the CWS before 2023, and we develop three supervised-ML-based classifiers—leveraging both original features as well as techniques inspired by prior work. We show that, in a "lab setting", our classifiers work well (e.g., 98% accuracy). Then, we collect a new, and more recent, set of 35,462 extensions from the CWS, published or last updated in 2023, with unknown ground truth. We were eventually able to identify 68 malicious extensions that bypassed the vetting process of the CWS. However, our classifiers also reported over 1k likely malicious extensions which may overestimate their true number. Based on this finding (further supported with other experiments and realistic analyses), we elucidate, for the first time, a strong *concept drift* effect on browser extensions. We also provide factual evidence that commercial detectors (e.g., VirusTotal) work poorly to detect known malicious extensions. Altogether, our results highlight the fact that detecting malicious browser extensions is a fundamentally hard problem which has not (yet) received an adequate degree of attention. This requires additional work both by the research community and by Google itself—potentially by revising their approaches. In the meantime, we informed Google of our discoveries, and we release our artifacts.

CCS Concepts: • **Security and privacy → Web application security; Social aspects of security and privacy;** • **Computing methodologies →** *Machine learning;* • **Information systems → World Wide Web.**

Additional Key Words and Phrases: Google Chrome; Browser Extensions; Classification; Concept Drift

\> It's not Easy: Applying Supervised ML to Detect Malicious Extensions in the CWS

In *ACM TWEB* 2026. Ben Rosenzweig, Valentino Dalla Valle, Giovanni Apruzzese, and Aurore Fass

DATASET

7,140 malicious extensions

Last update 2017—22

63,598 benign extensions

Last update 2012—22

FEATURES

Source code [A]

Metadata

[A] Fass et al.; *ACSAC* 2018

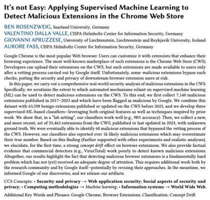
DETECTORS

Source code classifier

80% train | 20% test

Detecting Malicious Extensions — Lab Setting

Rosenzweig et al. TWEB 2026


It's not Easy: Applying Supervised Machine Learning to Detect Malicious Extensions in the Chrome Web Store

BEN ROSENZWEIG, Saarland University, Germany
VALENTINO DALLA VALLE, CISPA Helmholtz Center for Information Security, Germany
GIOVANNI APRUZZESE, University of Liechtenstein, Liechtenstein and Reykjavik University, Iceland
AURORE FASS, CISPA Helmholtz Center for Information Security, Germany

Google Chrome is the most popular Web browser. Users can customize it with *extensions* that enhance their browsing experience. The most well-known marketplace of such extensions is the Chrome Web Store (CWS). Developers can upload their extensions on the CWS, but such extensions are made available to users only after a vetting process carried out by Google itself. Unfortunately, some *malicious* extensions bypass such checks, putting the security and privacy of downstream browser extension users at risk.

In this paper, we carry out a comprehensive real-world security analysis of malicious extensions in the CWS. Specifically, we scrutinize the extent to which automated mechanisms reliant on supervised machine learning (ML) can be used to detect malicious extensions on the CWS. To this end, we first collect 7,140 malicious extensions published in 2017–2023 and which have been flagged as malicious by Google. We combine this dataset with 63,598 benign extensions published or updated on the CWS before 2023, and we develop three supervised-ML-based classifiers—leveraging both original features as well as techniques inspired by prior work. We show that, in a "lab setting", our classifiers work well (e.g., 98% accuracy). Then, we collect a new, and more recent, set of 35,462 extensions from the CWS, published or last updated in 2023, with unknown ground truth. We were eventually able to identify 68 malicious extensions that bypassed the vetting process of the CWS. However, our classifiers also reported over 1k likely malicious extensions which may overestimate their true number. Based on this finding (further supported with other experiments and realistic analyses), we elucidate, for the first time, a strong *concept drift* effect on browser extensions. We also provide factual evidence that commercial detectors (e.g., VirusTotal) work poorly to detect known malicious extensions. Altogether, our results highlight the fact that detecting malicious browser extensions is a fundamentally hard problem which has not (yet) received an adequate degree of attention. This requires additional work both by the research community and by Google itself—potentially by revising their approaches. In the meantime, we informed Google of our discoveries, and we release our artifacts.

CCS Concepts: • **Security and privacy → Web application security; Social aspects of security and privacy;** • **Computing methodologies →** *Machine learning;* • **Information systems → World Wide Web.**

Additional Key Words and Phrases: Google Chrome; Browser Extensions; Classification; Concept Drift

> It's not Easy: Applying Supervised ML to Detect Malicious Extensions in the CWS

In *ACM TWEB* 2026. Ben Rosenzweig, Valentino Dalla Valle, Giovanni Apruzzese, and Aurore Fass

DATASET

7,140 malicious extensions

Last update 2017—22

63,598 benign extensions

Last update 2012—22

FEATURES

Source code [A] → Source code classifier

Metadata → Metadata classifier

[A] Fass et al.; *ACSAC* 2018

DETECTORS

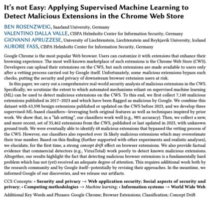
Source code classifier

Metadata classifier

80% train | 20% test

Detecting Malicious Extensions — Lab Setting

Rosenzweig et al. TWEB 2026


It's not Easy: Applying Supervised Machine Learning to Detect Malicious Extensions in the Chrome Web Store

BEN ROSENZWEIG, Saarland University, Germany
VALENTINO DALLA VALLE, CISPA Helmholtz Center for Information Security, Germany
GIOVANNI APRUZZESE, University of Liechtenstein, Liechtenstein and Reykjavik University, Iceland
AURORE FASS, CISPA Helmholtz Center for Information Security, Germany

Google Chrome is the most popular Web browser. Users can customize it with *extensions* that enhance their browsing experience. The most well-known marketplace of such extensions is the Chrome Web Store (CWS). Developers can upload their extensions on the CWS, but such extensions are made available to users only after a vetting process carried out by Google itself. Unfortunately, some *malicious* extensions bypass such checks, putting the security and privacy of downstream browser extension users at risk.

In this paper, we carry out a comprehensive real-world security analysis of malicious extensions in the CWS. Specifically, we scrutinize the extent to which automated mechanisms reliant on supervised machine learning (ML) can be used to detect malicious extensions on the CWS. To this end, we first collect 7,140 malicious extensions published in 2017–2023 and which have been flagged as malicious by Google. We combine this dataset with 63,598 benign extensions published or updated on the CWS before 2023, and we develop three supervised-ML-based classifiers—leveraging both original features as well as techniques inspired by prior work. We show that, in a "lab setting", our classifiers work well (e.g., 98% accuracy). Then, we collect a new, and more recent, set of 35,462 extensions from the CWS, published or last updated in 2023, with unknown ground truth. We were eventually able to identify 68 malicious extensions that bypassed the vetting process of the CWS. However, our classifiers also reported over 1k likely malicious extensions which may overestimate their true number. Based on this finding (further supported with other experiments and realistic analyses), we elucidate, for the first time, a strong *concept drift* effect on browser extensions. We also provide factual evidence that commercial detectors (e.g., VirusTotal) work poorly to detect known malicious extensions. Altogether, our results highlight the fact that detecting malicious browser extensions is a fundamentally hard problem which has not (yet) received an adequate degree of attention. This requires additional work both by the research community and by Google itself—potentially by revising their approaches. In the meantime, we informed Google of our discoveries, and we release our artifacts.

CCS Concepts: • **Security and privacy** → **Web application security**; **Social aspects of security and privacy**; • **Computing methodologies** → *Machine learning*; • **Information systems** → **World Wide Web**.

Additional Key Words and Phrases: Google Chrome; Browser Extensions; Classification; Concept Drift

> It's not Easy: Applying Supervised ML to Detect Malicious Extensions in the CWS

In *ACM TWEB* 2026. Ben Rosenzweig, Valentino Dalla Valle, Giovanni Apruzzese, and Aurore Fass

DATASET

7,140 malicious extensions

Last update 2017—22

63,598 benign extensions

Last update 2012—22

FEATURES

Source code [A]

Metadata

[A] Fass et al.; *ACSAC* 2018

DETECTORS

Source code classifier

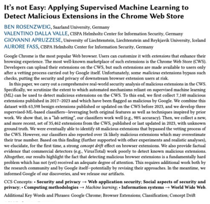
Metadata classifier

Combined classifier

80% train | 20% test

Detecting Malicious Extensions — Lab Setting

Rosenzweig et al. TWEB 2026


It's not Easy: Applying Supervised Machine Learning to Detect Malicious Extensions in the Chrome Web Store

BEN ROSENZWEIG, Saarland University, Germany
VALENTINO DALLA VALLE, CISPA Helmholtz Center for Information Security, Germany
GIOVANNI APRUZZESE, University of Liechtenstein, Liechtenstein and Reykjavik University, Iceland
AURORE FASS, CISPA Helmholtz Center for Information Security, Germany

Google Chrome is the most popular Web browser. Users can customize it with *extensions* that enhance their browsing experience. The most well-known marketplace of such extensions is the Chrome Web Store (CWS). Developers can upload their extensions on the CWS, but such extensions are made available to users only after a vetting process carried out by Google itself. Unfortunately, some *malicious* extensions bypass such checks, putting the security and privacy of downstream browser extension users at risk.

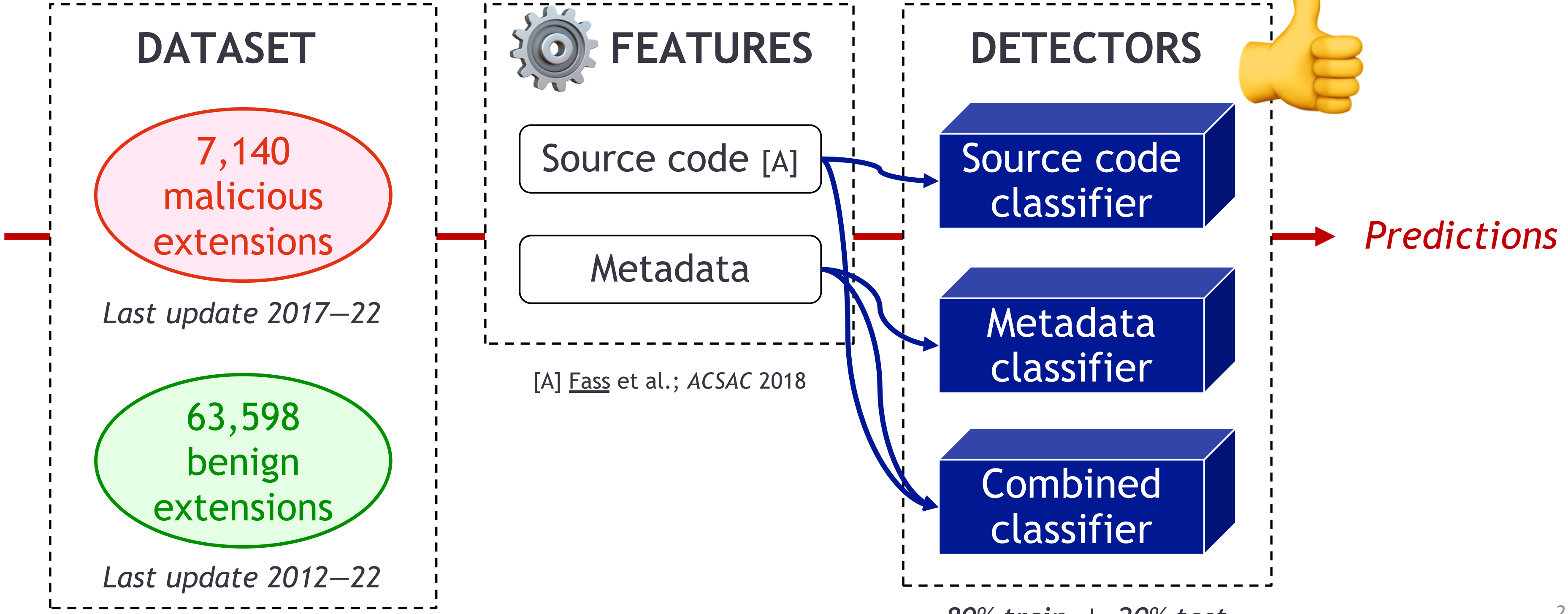
In this paper, we carry out a comprehensive real-world security analysis of malicious extensions in the CWS. Specifically, we scrutinize the extent to which automated mechanisms reliant on supervised machine learning (ML) can be used to detect malicious extensions on the CWS. To this end, we first collect 7,140 malicious extensions published in 2017–2023 and which have been flagged as malicious by Google. We combine this dataset with 63,598 benign extensions published or updated on the CWS before 2023, and we develop three supervised-ML-based classifiers—leveraging both original features as well as techniques inspired by prior work. We show that, in a "lab setting", our classifiers work well (e.g., 98% accuracy). Then, we collect a new, and more recent, set of 35,462 extensions from the CWS, published or last updated in 2023, with unknown ground truth. We were eventually able to identify 68 malicious extensions that bypassed the vetting process of the CWS. However, our classifiers also reported over 1k likely malicious extensions which may overestimate their true number. Based on this finding (further supported with other experiments and realistic analyses), we elucidate, for the first time, a strong *concept drift* effect on browser extensions. We also provide factual evidence that commercial detectors (e.g., VirusTotal) work poorly to detect known malicious extensions. Altogether, our results highlight the fact that detecting malicious browser extensions is a fundamentally hard problem which has not (yet) received an adequate degree of attention. This requires additional work both by the research community and by Google itself—potentially by revising their approaches. In the meantime, we informed Google of our discoveries, and we release our artifacts.

CCS Concepts: • **Security and privacy** → **Web application security**; **Social aspects of security and privacy**; • **Computing methodologies** → *Machine learning*; • **Information systems** → **World Wide Web**.

Additional Key Words and Phrases: Google Chrome; Browser Extensions; Classification; Concept Drift

> It's not Easy: Applying Supervised ML to Detect Malicious Extensions in the CWS

In *ACM TWEB* 2026. Ben Rosenzweig, Valentino Dalla Valle, Giovanni Apruzzese, and Aurore Fass

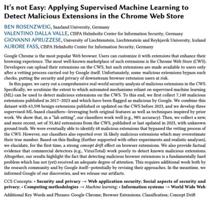


[A] Fass et al.; *ACSAC* 2018

80% train | 20% test

Detecting Malicious Extensions — Lab Setting

Rosenzweig et al. TWEB 2026


It's not Easy: Applying Supervised Machine Learning to Detect Malicious Extensions in the Chrome Web Store

BEN ROSENZWEIG, Saarland University, Germany
VALENTINO DALLA VALLE, CISPA Helmholtz Center for Information Security, Germany
GIOVANNI APRUZZESE, University of Liechtenstein, Liechtenstein and Reykjavik University, Iceland
AURORE FASS, CISPA Helmholtz Center for Information Security, Germany

Google Chrome is the most popular Web browser. Users can customize it with *extensions* that enhance their browsing experience. The most well-known marketplace of such extensions is the Chrome Web Store (CWS). Developers can upload their extensions on the CWS, but such extensions are made available to users only after a vetting process carried out by Google itself. Unfortunately, some *malicious* extensions bypass such checks, putting the security and privacy of downstream browser extension users at risk.

In this paper, we carry out a comprehensive real-world security analysis of malicious extensions in the CWS. Specifically, we scrutinize the extent to which automated mechanisms reliant on supervised machine learning (ML) can be used to detect malicious extensions on the CWS. To this end, we first collect 7,140 malicious extensions published in 2017–2023 and which have been flagged as malicious by Google. We combine this dataset with 63,598 benign extensions published or updated on the CWS before 2023, and we develop three supervised-ML-based classifiers—leveraging both original features as well as techniques inspired by prior work. We show that, in a "lab setting", our classifiers work well (e.g., 98% accuracy). Then, we collect a new, and more recent, set of 35,462 extensions from the CWS, published or last updated in 2023, with unknown ground truth. We were eventually able to identify 68 malicious extensions that bypassed the vetting process of the CWS. However, our classifiers also reported over 1k likely malicious extensions which may overestimate their true number. Based on this finding (further supported with other experiments and realistic analyses), we elucidate, for the first time, a strong *concept drift* effect on browser extensions. We also provide factual evidence that commercial detectors (e.g., VirusTotal) work poorly to detect known malicious extensions. Altogether, our results highlight the fact that detecting malicious browser extensions is a fundamentally hard problem which has not (yet) received an adequate degree of attention. This requires additional work both by the research community and by Google itself—potentially by revising their approaches. In the meantime, we informed Google of our discoveries, and we release our artifacts.

CCS Concepts: • **Security and privacy** → **Web application security**; **Social aspects of security and privacy**; • **Computing methodologies** → *Machine learning*; • **Information systems** → **World Wide Web**.

Additional Key Words and Phrases: Google Chrome; Browser Extensions; Classification; Concept Drift

> It's not Easy: Applying Supervised ML to Detect Malicious Extensions in the CWS

In *ACM TWEB* 2026. Ben Rosenzweig, Valentino Dalla Valle, Giovanni Apruzzese, and Aurore Fass

DATASET

7,140 malicious extensions

Last update 2017—22

63,598 benign extensions

Last update 2012—22

FEATURES

Source code [A]

Metadata

[A] Fass et al.; *ACSAC* 2018

DETECTORS

Source code classifier

Metadata classifier

Combined classifier

80% train | 20% test

Predictions

Acc: 98.37%

TNR: 98.71%

TPR: 95.39%

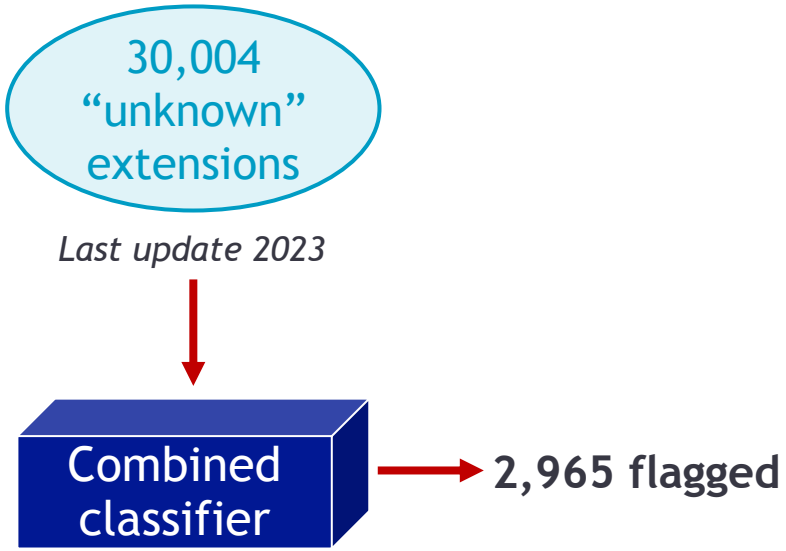
Detecting Malicious Extensions — Real World

30,004 “unknown” extensions

Last update 2023

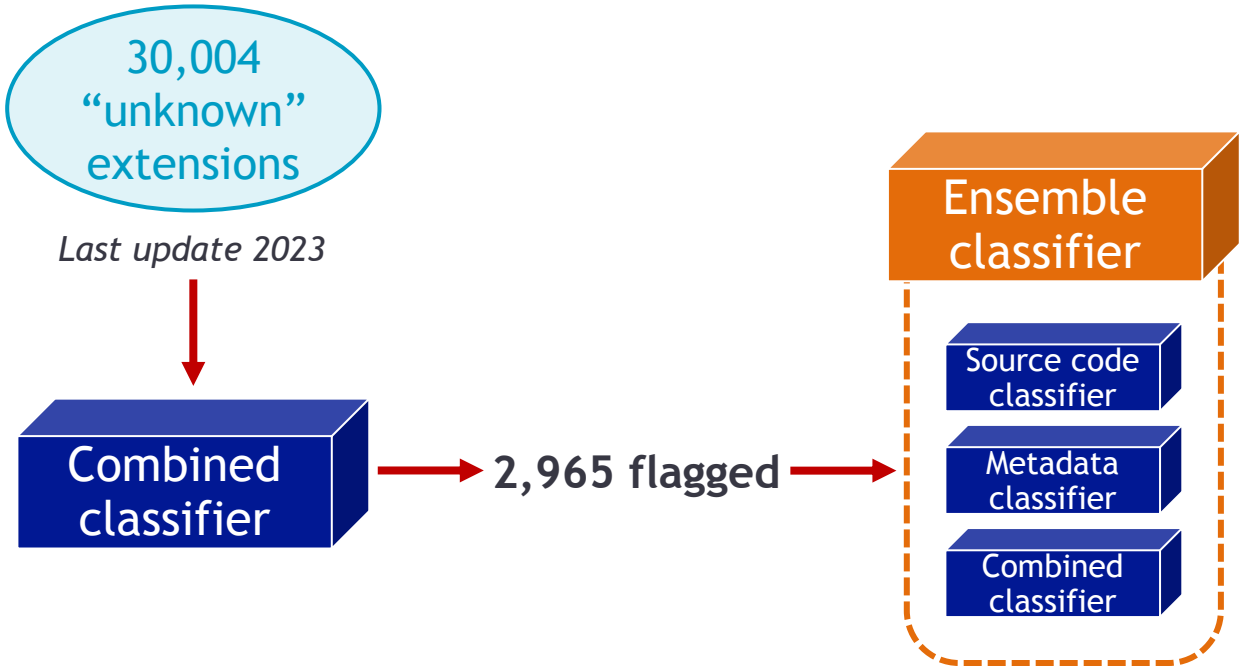
[3] Rosenzweig et al.; *TWEB* 2026

Detecting Malicious Extensions — Real World



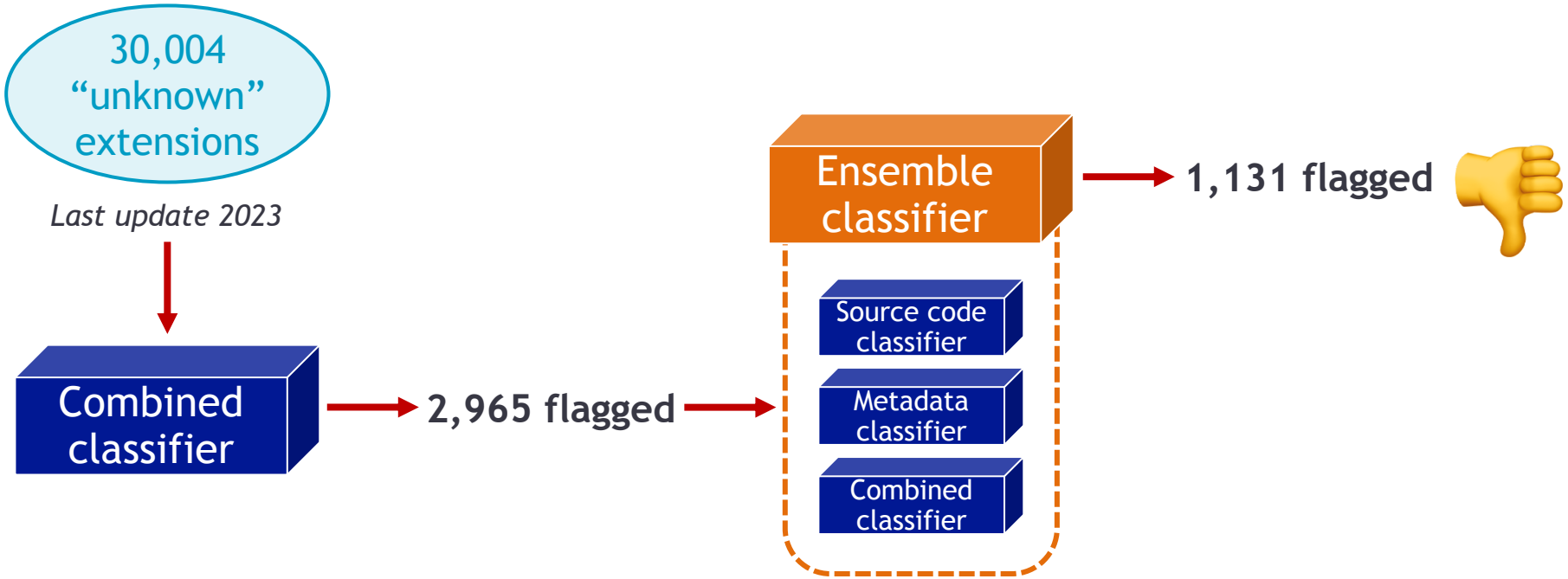
[3] Rosenzweig et al.; *TWEB* 2026

Detecting Malicious Extensions — Real World



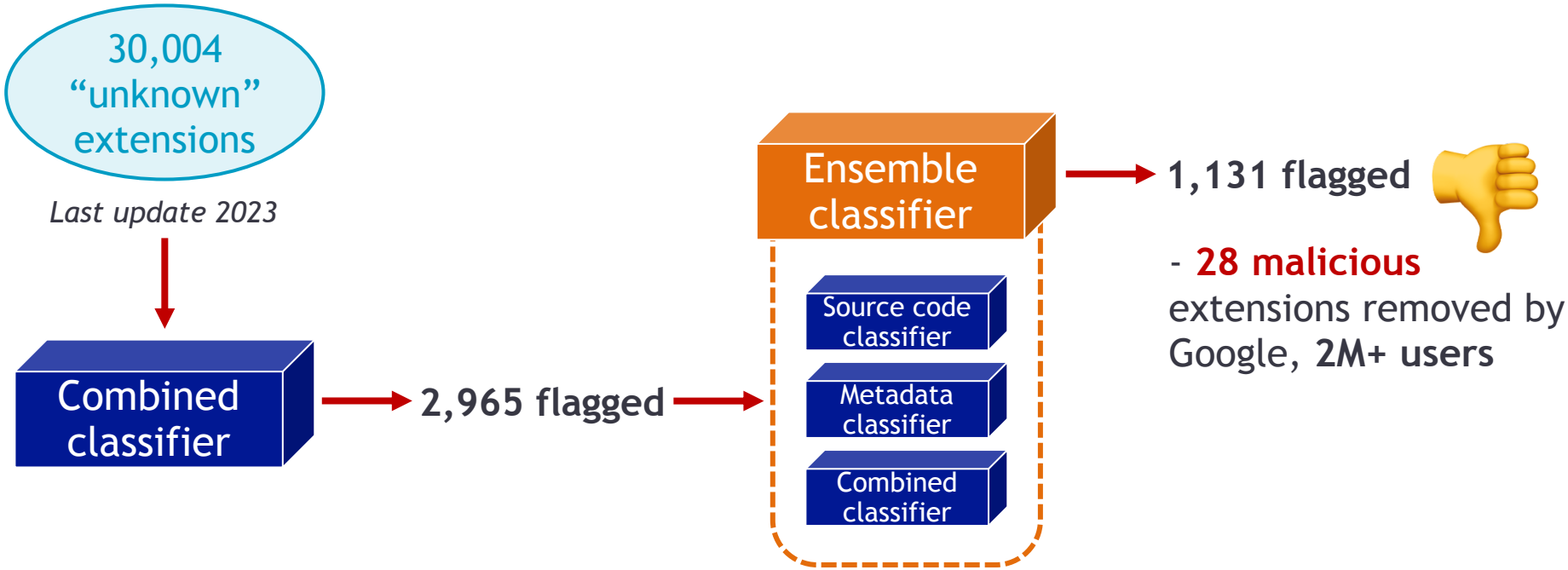
[3] Rosenzweig et al.; *TWEB* 2026

Detecting Malicious Extensions — Real World



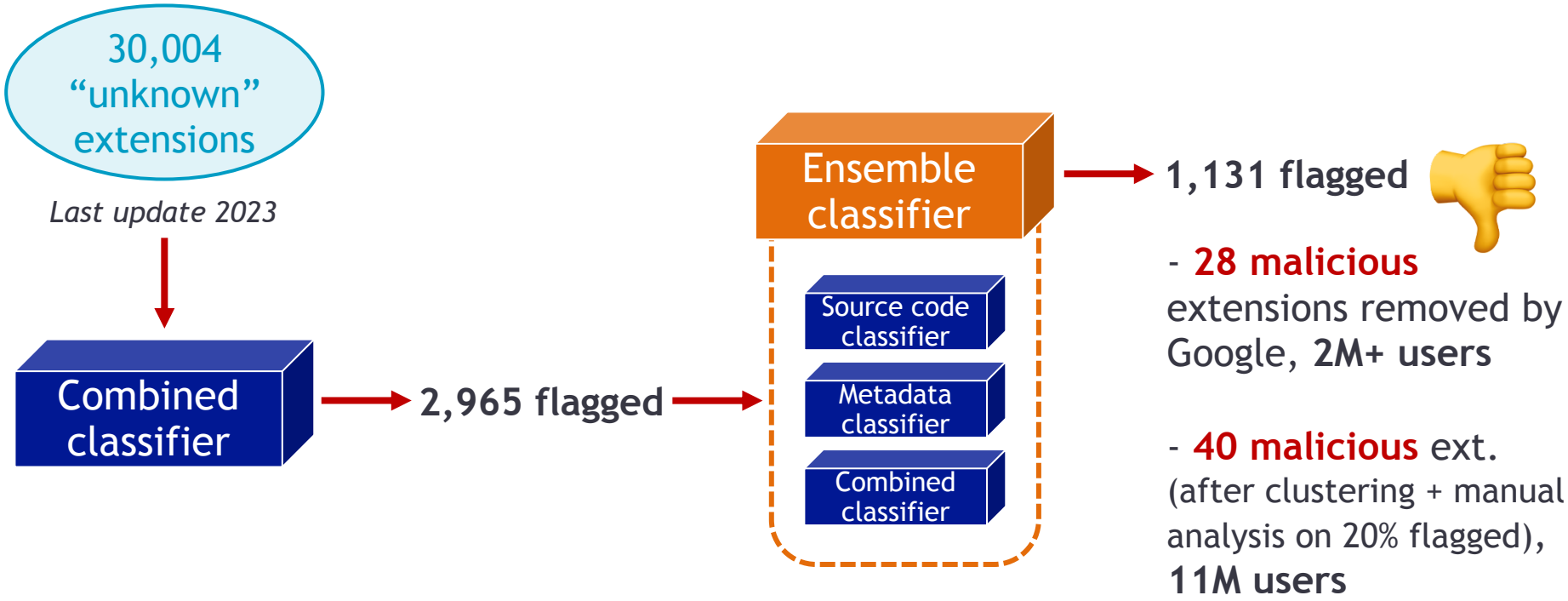
[3] Rosenzweig et al.; *TWEB* 2026

Detecting Malicious Extensions — Real World



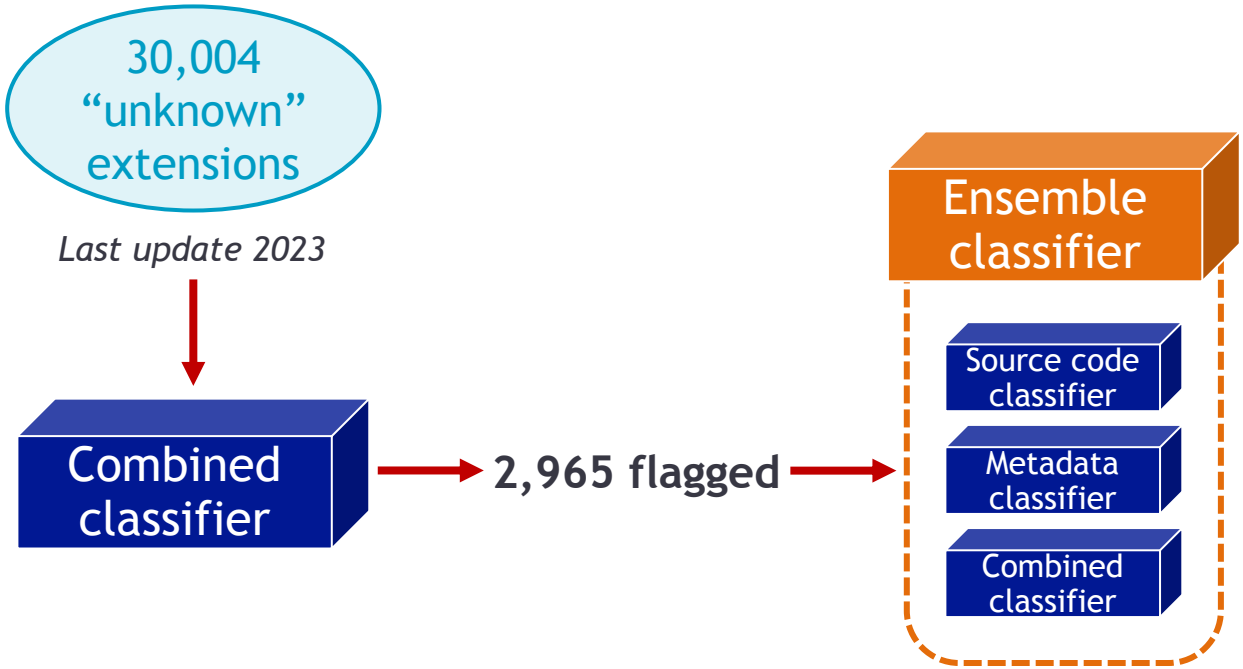
[3] Rosenzweig et al.; *TWEB* 2026

Detecting Malicious Extensions — Real World



[3] Rosenzweig et al.; *TWEB* 2026

Detecting Malicious Extensions — Real World



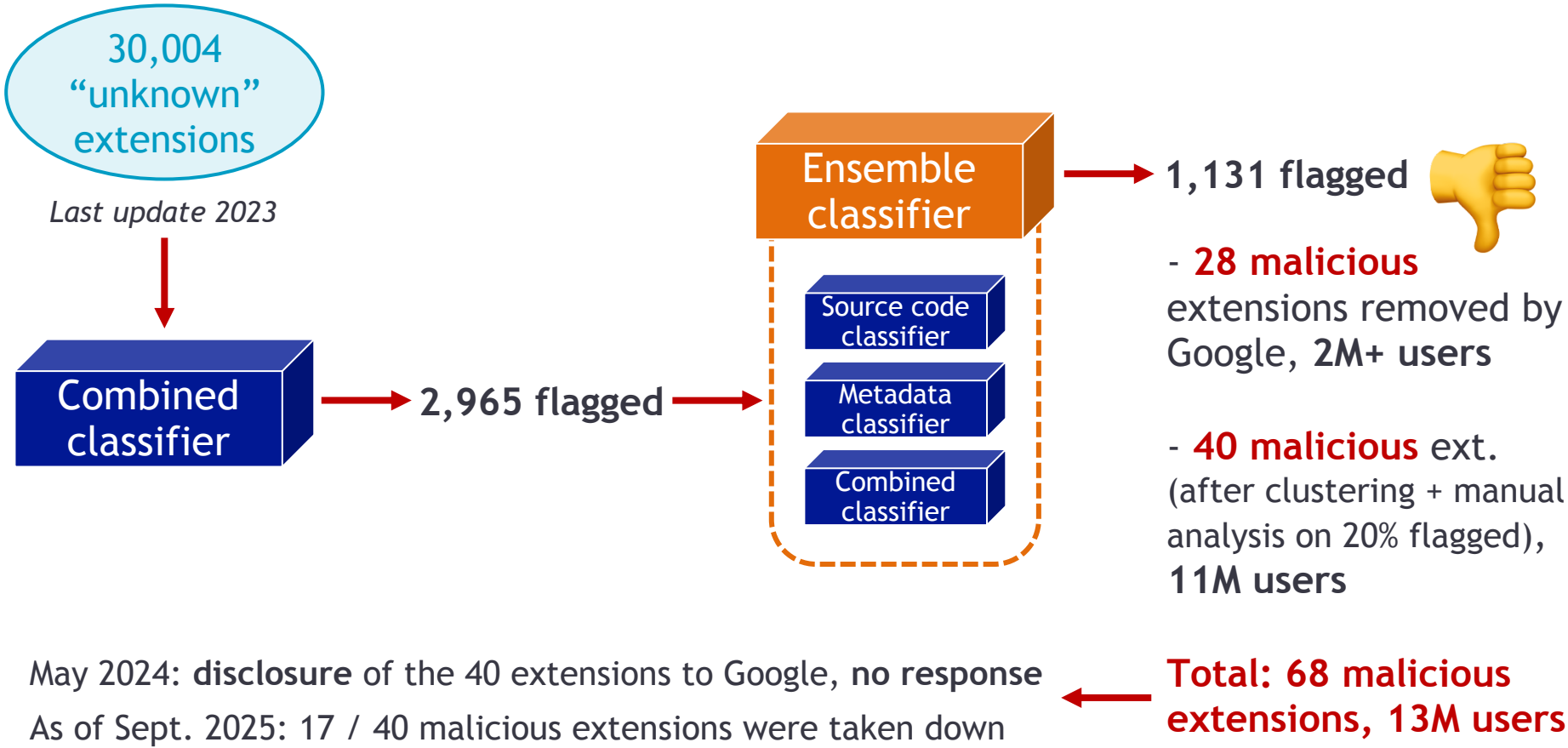
- **28 malicious** extensions removed by Google, **2M+ users**

- **40 malicious** ext. (after clustering + manual analysis on 20% flagged), **11M users**

Total: 68 malicious extensions, 13M users

[3] Rosenzweig et al.; *TWEB* 2026

Detecting Malicious Extensions — Real World



[3] Rosenzweig et al.; *TWEB* 2026

Detecting Malicious Extensions — Lab Setting vs. Real World

- **Lab setting**: 98.37% accuracy
- **Real world**: 1,131 / 30,004 extensions flagged | 68 verified malicious extensions

???

[3] Rosenzweig et al.; *TWEB* 2026

Detecting Malicious Extensions — Lab Setting vs. Real World

- **Lab setting**: 98.37% accuracy
- **Real world**: 1,131 / 30,004 extensions flagged | 68 verified malicious extensions

???

- **Concept drift**
  - **Intrinsic evolution** of extensions, hard for supervised ML tools to keep a (high) accuracy over time
  - **How** to validate concept drift? → see experiments in the paper
  - **Why** are extensions affected? → see paper (study feature importance)

[3] Rosenzweig et al.; *TWEB* 2026

Detecting Malicious Extensions — Lab Setting vs. Real World

- **Lab setting**: 98.37% accuracy
- **Real world**: 1,131 / 30,004 extensions flagged | 68 verified malicious extensions

???

- **Concept drift**
  - **Intrinsic evolution** of extensions, hard for supervised ML tools to keep a (high) accuracy over time
  - **How** to validate concept drift? → see experiments in the paper
  - **Why** are extensions affected? → see paper (study feature importance)

➢ ML evaluations can be **misleading**: detectors performing well in a lab setting are **no guarantee of practical applicability** in the real world

[3] Rosenzweig et al.; *TWEB* 2026

Dangerous Browser Extensions

→ Extensions can put their users' security & privacy at risk:

- **Contain malware**: designed by malicious actors to harm victims
  - E.g., propagate malware, steal users' credentials, track users [1, 3]
- **Contain vulnerabilities**: designed by well-intentioned developers… but buggy
  - E.g., can lead to user-sensitive data exfiltration [1, 2]
- **Violate the Chrome Web Store policies**
  - E.g., deceive users, promote unlawful activities, lack a privacy policy [1]
- **Be fingerprintable**: can be recognized and uniquely identified
  - E.g., can lead to user tracking or inferring of user personal information [4]

[1] Hsu, Tran, and Fass; *AsiaCCS* 2024 | [2] Fass, Somé, Backes, and Stock; *CCS* 2021 | [3] Rosenzweig, Dalla Valle, Apruzzese, and Fass; *TWEB* 2026 | [4] Agarwal, Fass, and Stock; *CCS* 2024

Browser Extension Fingerprinting

Browser extensions can **interact with web pages…**

[4] Agarwal, Fass, and Stock; *CCS* 2024

Browser Extension Fingerprinting

Browser extensions can **interact with web pages…**

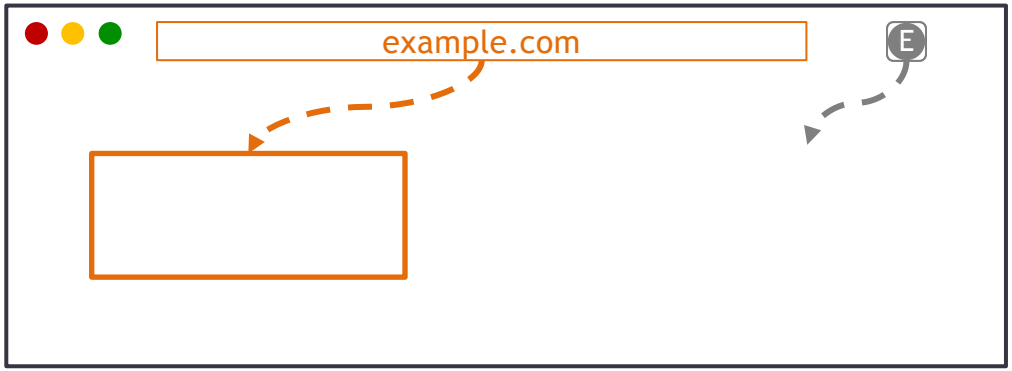
- 1) Extensions send **PostMessage** to web pages

[4] Agarwal, Fass, and Stock; *CCS* 2024

Browser Extension Fingerprinting

Browser extensions can **interact with web pages…**

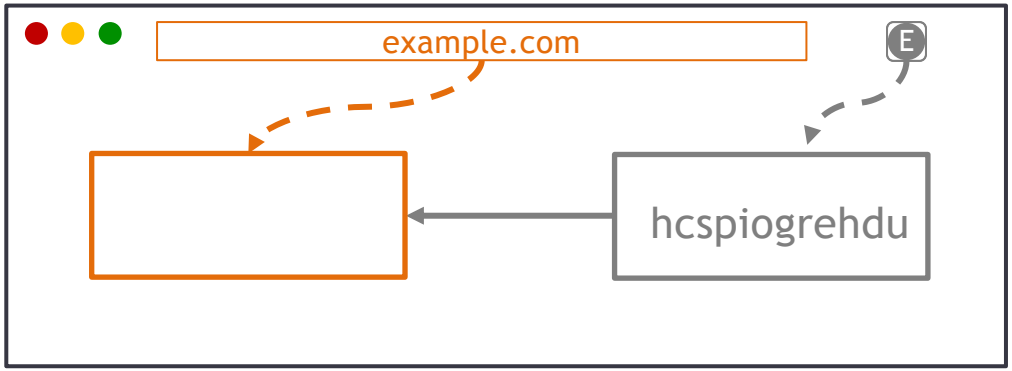
- 1) Extensions send **PostMessage** to web pages



Browser Extension Fingerprinting

Browser extensions can **interact with web pages…**

- 1) Extensions send **PostMessage** to web pages



[4] Agarwal, Fass, and Stock; *CCS* 2024

Browser Extension Fingerprinting

Browser extensions can **interact with web pages…**

- 1) Extensions send **PostMessage** to web pages
- 2) Extensions store data on the client side through **storage APIs**

  (e.g., cookies, local/session storage, IndexedDB)

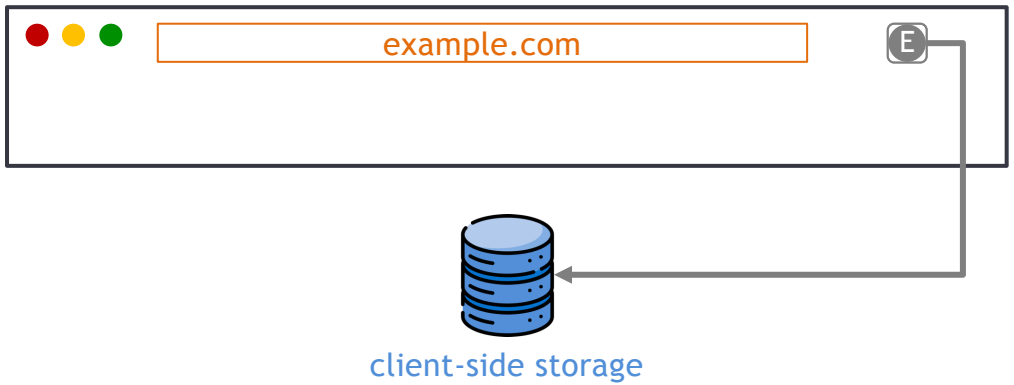
[4] Agarwal, Fass, and Stock; *CCS* 2024

Browser Extension Fingerprinting

Browser extensions can **interact with web pages…**

- 1) Extensions send **PostMessage** to web pages
- 2) Extensions store data on the client side through **storage APIs**

  (e.g., cookies, local/session storage, IndexedDB)

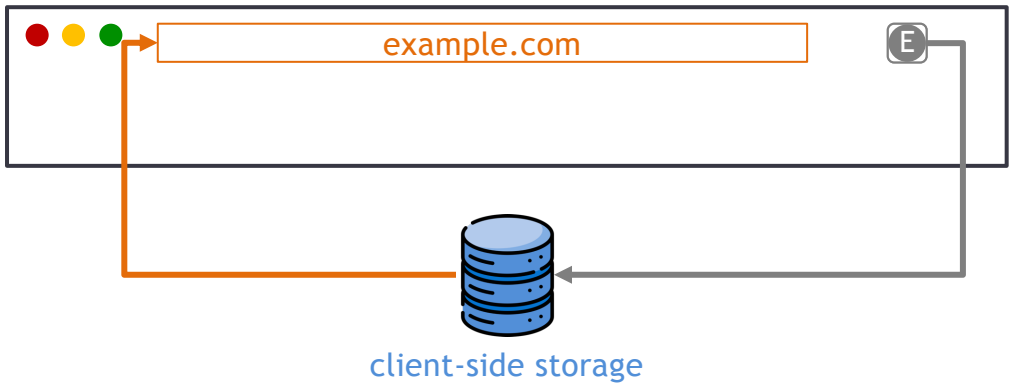


Browser Extension Fingerprinting

Browser extensions can **interact with web pages…**

- 1) Extensions send **PostMessage** to web pages
- 2) Extensions store data on the client side through **storage APIs**

  (e.g., cookies, local/session storage, IndexedDB)

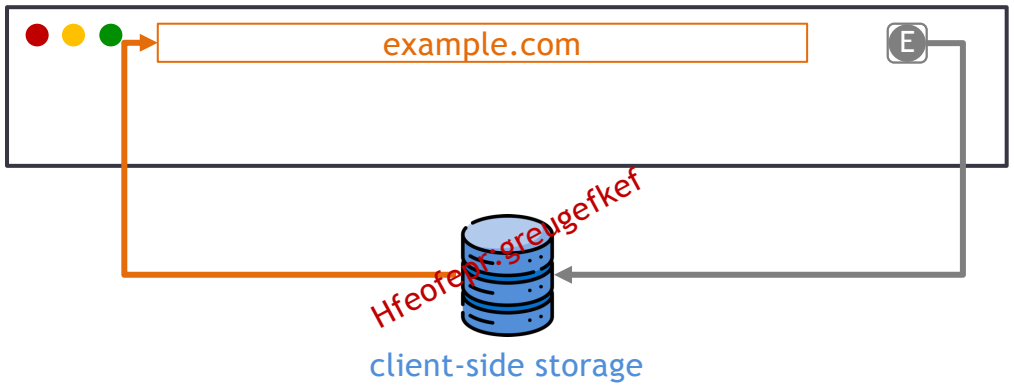


Browser Extension Fingerprinting

Browser extensions can **interact with web pages…**

- 1) Extensions send **PostMessage** to web pages
- 2) Extensions store data on the client side through **storage APIs**

  (e.g., cookies, local/session storage, IndexedDB)



Browser Extension Fingerprinting

Browser extensions can **interact with web pages…**

- 1) Extensions send **PostMessage** to web pages
- 2) Extensions store data on the client side through **storage APIs**

  (e.g., cookies, local/session storage, IndexedDB)
- 3) Extensions **inject JavaScript** code directly into web pages

[4] Agarwal, Fass, and Stock; *CCS* 2024

Browser Extension Fingerprinting

Browser extensions can **interact with web pages…**

- 1) Extensions send **PostMessage** to web pages
- 2) Extensions store data on the client side through **storage APIs**

  (e.g., cookies, local/session storage, IndexedDB)
- 3) Extensions **inject JavaScript** code directly into web pages

… which leaves traces

[4] Agarwal, Fass, and Stock; *CCS* 2024

Browser Extension Fingerprinting

Browser extensions can **interact with web pages…**

- 1) Extensions send **PostMessage** to web pages
- 2) Extensions store data on the client side through **storage APIs**

  (e.g., cookies, local/session storage, IndexedDB)
- 3) Extensions **inject JavaScript** code directly into web pages

  → registering global variables

  → invocation of global APIs and properties

… which leaves traces

[4] Agarwal, Fass, and Stock; *CCS* 2024

Browser Extension Fingerprinting

Browser extensions can **interact with web pages…**

- 1) Extensions send **PostMessage** to web pages
- 2) Extensions store data on the client side through **storage APIs**

  (e.g., cookies, local/session storage, IndexedDB)
- 3) Extensions **inject JavaScript** code directly into web pages

  → registering global variables

  → invocation of global APIs and properties

… which leaves traces → **observable side effects**, can be seen by a “malicious” site

[4] Agarwal, Fass, and Stock; *CCS* 2024

Browser Extension Fingerprinting

Browser extensions can interact with web pages…

- 1) Extensions send **PostMessage** to web pages
- 2) Extensions store data on the client side through **storage APIs**
  (e.g., cookies, local/session storage, indexDB)
- 3) Extensions **inject JavaScript** code directly into web pages
  → registering global variables
  → invocation of global APIs and properties

… which leaves traces → **observable side effects**, can be seen by a "malicious" site

Why is this bad?

[4] Agarwal, Fass, and Stock; *CCS* 2024

Browser Extension Fingerprinting

Browser extensions can interact with web pages…

- 1) Extensions send **PostMessage** to web pages
- 2) Extensions store data on the client side through **storage APIs**
  (e.g., cookies, local/session storage, indexDB)
  - "Malicious" websites can **track users** by fingerprinting their extensions
- 3) Extensions **inject JavaScript** code directly into web pages
  → registering global variables
  → invocation of global APIs and properties

Why is this bad?

… which leaves traces → **observable side effects**, can be seen by a "malicious" site

[4] Agarwal, Fass, and Stock; *CCS* 2024

Browser Extension Fingerprinting

Browser extensions can **interact with web pages**…

Why is this bad?

- 1) Extensions send **PostMessage** to web pages
- 2) Extensions store data on the client side through **storage APIs** (e.g., cookies, local/session storage, indexDB)
- “Malicious” websites can **track users** by fingerprinting their extensions
- 3) Extensions **inject JavaScript** code directly into web pages
  → registering global variables
  → invocation of global APIs and properties
- Extensions can **reveal personal user information**, e.g., geolocation, ethnicity, social/personal interests, medical issues, religion, etc. [Karami; NDSS’19]

… which leaves traces → **observable side effects**, can be seen by a “malicious” site

[4] Agarwal, Fass, and Stock; *CCS* 2024

Detecting Fingerprintable Extensions

- 1) Extensions send **PostMessage** to web pages
- 2) Extensions store data on the client side through **storage APIs**
- 3) Extensions **inject JavaScript** code directly into web pages

Detecting Fingerprintable Extensions

- 1) Extensions send **PostMessage** to web pages
- 2) Extensions store data on the client side through **storage APIs**
- 3) Extensions **inject JavaScript** code directly into web pages

*How many extensions can be **uniquely fingerprinted** through these observable side effects?*

Detecting Fingerprintable Extensions

- 1) Extensions send **PostMessage** to web pages
- 2) Extensions store data on the client side through **storage APIs**
- 3) Extensions **inject JavaScript** code directly into web pages

*How many extensions can be **uniquely fingerprinted** through these observable side effects?*

Peeking through the `window`: Fingerprinting Browser Extensions through Page-Visible Execution Traces and Interactions

Shubham Agarwal, CISPA Helmholtz Center for Information Security, shubham.agarwal@cispa.de
Aurore Fass, CISPA Helmholtz Center for Information Security, fass@cispa.de
Ben Stock, CISPA Helmholtz Center for Information Security, stock@cispa.de

ABSTRACT

Browser extensions are third-party add-ons that provide myriads of features to their users while browsing on the Web. Extensions often interact with the websites a user visits and perform various operations such as DOM-based manipulation, script injections, and so on. However, this also enables nefarious websites to track their visitors by fingerprinting extensions. Researchers in the past have shown that extensions are susceptible to fingerprinting based on the resources they include, the styles they deploy, or the DOM-based modifications they perform. Fortunately, the current extension ecosystem contains safeguards against many such known issues through appropriate defense mechanisms.

We present the first study to investigate the fingerprinting characteristics of extension-injected code in pages' JavaScript namespace and through other observable side-effects like changed cookies. Doing so, we find that many extensions inject JavaScript that pollutes the applications' global namespace by registering variables. It also enables the attacker application to monitor the execution of the injected code by overwriting the JavaScript APIs and capturing execution traces through the *stacktrace*, the set of APIs invoked, etc. Further, extensions also store data on the client side and perform event-driven functionalities that aid in attribution. Through our tests, we find 2,747 Chrome and 572 Firefox extensions to be susceptible to fingerprinting. Unfortunately, none of the existing defense mechanisms prevent extensions from being fingerprinted through our proposed vectors. Therefore, we also suggest potential measures for developers and browser vendors to safeguard the extension ecosystem against such fingerprinting attempts.

CCS CONCEPTS

• **Security and privacy → Web application security**.

KEYWORDS

Client-side Security, Extension Fingerprinting, Browser Extensions

ACM Reference Format:

Shubham Agarwal, Aurore Fass, and Ben Stock. 2024. Peeking through the window: Fingerprinting Browser Extensions through Page-Visible Execution Traces and Interactions. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS '24), October 14–18, 2024, Salt Lake City, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CCS '24, October 14–18, 2024, Salt Lake City, USA
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

With the rising trend of cookie-less tracking, online trackers are in an arms race with Web Privacy advocates. They continuously compete with various anti-tracking and fingerprinting measures to uniquely identify users on the Web. Recent studies on browser fingerprinting techniques have shown many cookie-less vectors, such as Canvas APIs [5], WebGL APIs [6], and other side channels [1, 29, 32, 44, 51, 58]. These techniques allow a malicious website to effectively harvest client-side information specific to individual users on the Web and further track their activity across websites.

Browser extensions have emerged as one of the most interesting fingerprinting vectors for these trackers in recent times, owing to their unique position in the overall Web ecosystem. Extensions can perform privileged operations implemented in their background script or even execute code in the Web applications' execution context through the content scripts. Tailored to provide a specific set of features to their respective users, browser extensions may also inadvertently reveal personal information about their users, such as their geolocation, background, ethnicity, or social and personal interests [26]. These extensions perform a specific and often highly privileged set of operations, such as DOM-based modifications, changes to cookies, or script injections, to implement their intended functionality at runtime. These operations, however, could also expose them to being uniquely identified by online trackers.

Prior work has shown different ways of uniquely identifying browser extensions. Some of these include Web-accessible resources-based (WAR-based) fingerprints [48], the side effect of code bloating [53], behavior-based fingerprints [26, 50, 54], user-induced side effects [49], and stylesheet-based fingerprints [30]. On the other hand, several mitigation techniques also intend to thwart any fingerprinting attempts based on the above techniques [27, 47, 60]. The studies have led to a cat-and-mouse game of newly introduced fingerprinting vectors and subsequent defense mechanisms. A recent example is the optional usage of randomized runtime identifiers for extensions [41] to thwart WAR-based fingerprinting.

However, ultimately, every extension has some intended functionality, much of which is to interact with the loaded page. This also means that the extension needs to interact with the page through various APIs from the content script. However, when not done carefully, this can leave traces of the extension's execution in the JavaScript namespace of the document, i.e., within the reach of a fingerprinting attacker. These range from client-side storage (e.g., *localStorage* or *cookies*) through the invocation of global APIs

> Peeking through the `window`: Fingerprinting Browser Extensions through Page-Visible Execution Traces and Interactions

In *ACM CCS* 2024. Shubham Agarwal, Aurore Fass, and Ben Stock

Agarwal et al.
CCS 2024

Detecting Fingerprintable Extensions with Raider

Analyzed 38k Chrome extensions from 2024 with Raider

- **2,747 fingerprintable Chrome extensions** (lower bound)
- Impacting **169M users**

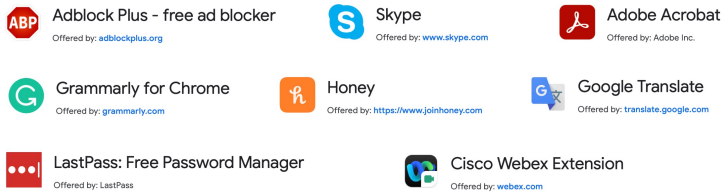
- **Notified 1,967 developers** about their fingerprintable extension(s)
  - Only 30 (!) replied
  - Of those, only 16 positively acknowledged the issues
    - But: they heavily rely on our fingerprinting vectors (e.g., script injection or data storage) for their extensions' functionality

- Raider PoC is **available online** Raider-ext/raider

[4] Agarwal, Fass, and Stock; *CCS* 2024

Summary

Browser Extensions are Popular



- ~200k Chrome extensions totaling almost 2B active users
  - ➢ Necessitates a thorough **security and privacy assessment**

Dangerous Browser Extensions

→ **Extensions can put their users' security & privacy at risk:**

- **Contain malware:** designed by malicious actors to harm victims
  - E.g., propagate malware, steal users' credentials, track users
- **Contain vulnerabilities:** designed by well-intentioned developers… but buggy
  - E.g., can lead to user-sensitive data exfiltration
- **Violate the Chrome Web Store policies**
  - E.g., deceive users, promote unlawful activities, lack a privacy policy
- **Be fingerprintable:** can be recognized and uniquely identified
  - E.g., can lead to user tracking or inferring of user personal information

Detecting Malicious Extensions – Lab Setting vs. Real World

- **Lab setting**: 98.37% accuracy
- **Real world**: 1,131 / 30,004 extensions flagged | 68 verified malicious extensions

Rosenzweig et al. TWEB 2026

its-not-easy/tweb25

- **Concept drift**
  - **Intrinsic evolution** of extensions, hard for supervised ML tools to keep a (high) accuracy over time

➢ ML evaluations can be **misleading**: detectors performing well in a lab setting are **no guarantee of practical applicability** in the real world

Detecting Fingerprintable Extensions with Raider

Agarwal et al. CCS 2024

- 1) Extensions send **PostMessage** to web pages
- 2) Extensions store data on the client side through **storage APIs**
  (e.g., cookies, local/session storage, IndexedDB)
- 3) Extensions **inject JavaScript** code directly into web pages
  - → registering global variables
  - → invocation of global APIs and properties

Raider-ext/raider

➢ Raider detects **2,747** fingerprintable extensions | **169M users**

Corresponding Publications

- What is in the Chrome Web Store?

Sheryl Hsu, Manda Tran, and Aurore Fass. In *ACM AsiaCCS* 2024

- DoubleX: Statically Detecting Vulnerable Data Flows in Browser Extensions at Scale

Aurore Fass, Dolière Francis Somé, Michael Backes, and Ben Stock. In *ACM CCS* 2021

- It's not Easy: Applying Supervised Machine Learning to Detect Malicious Extensions in the Chrome Web Store

Ben Rosenzweig, Valentino Dalla Valle, Giovanni Apruzzese, and Aurore Fass. In *ACM TWEB* 2026

- Peeking through the window: Fingerprinting Browser Extensions through Page-Visible Execution Traces and Interactions

Shubham Agarwal, Aurore Fass, and Ben Stock. In *ACM CCS* 2024