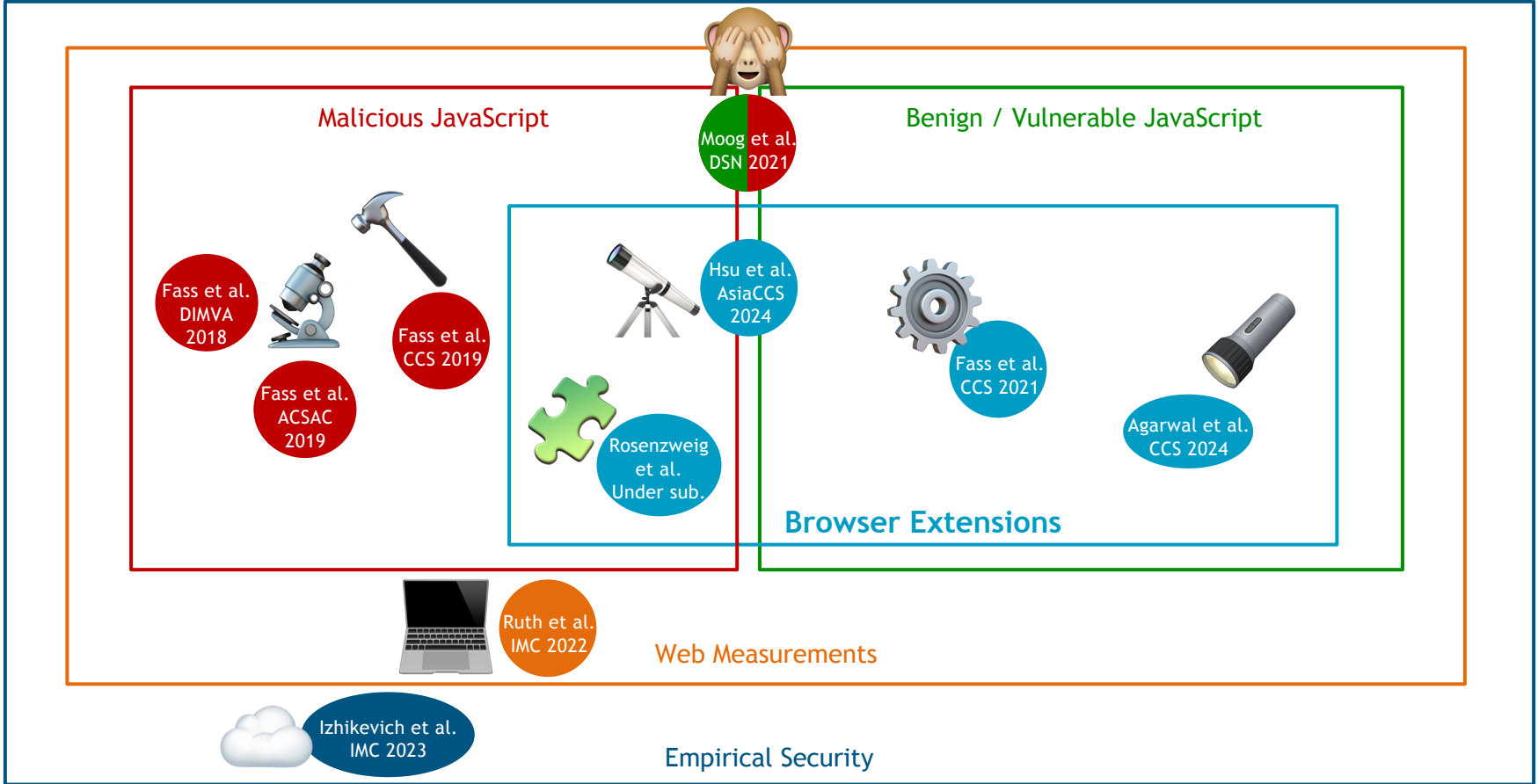
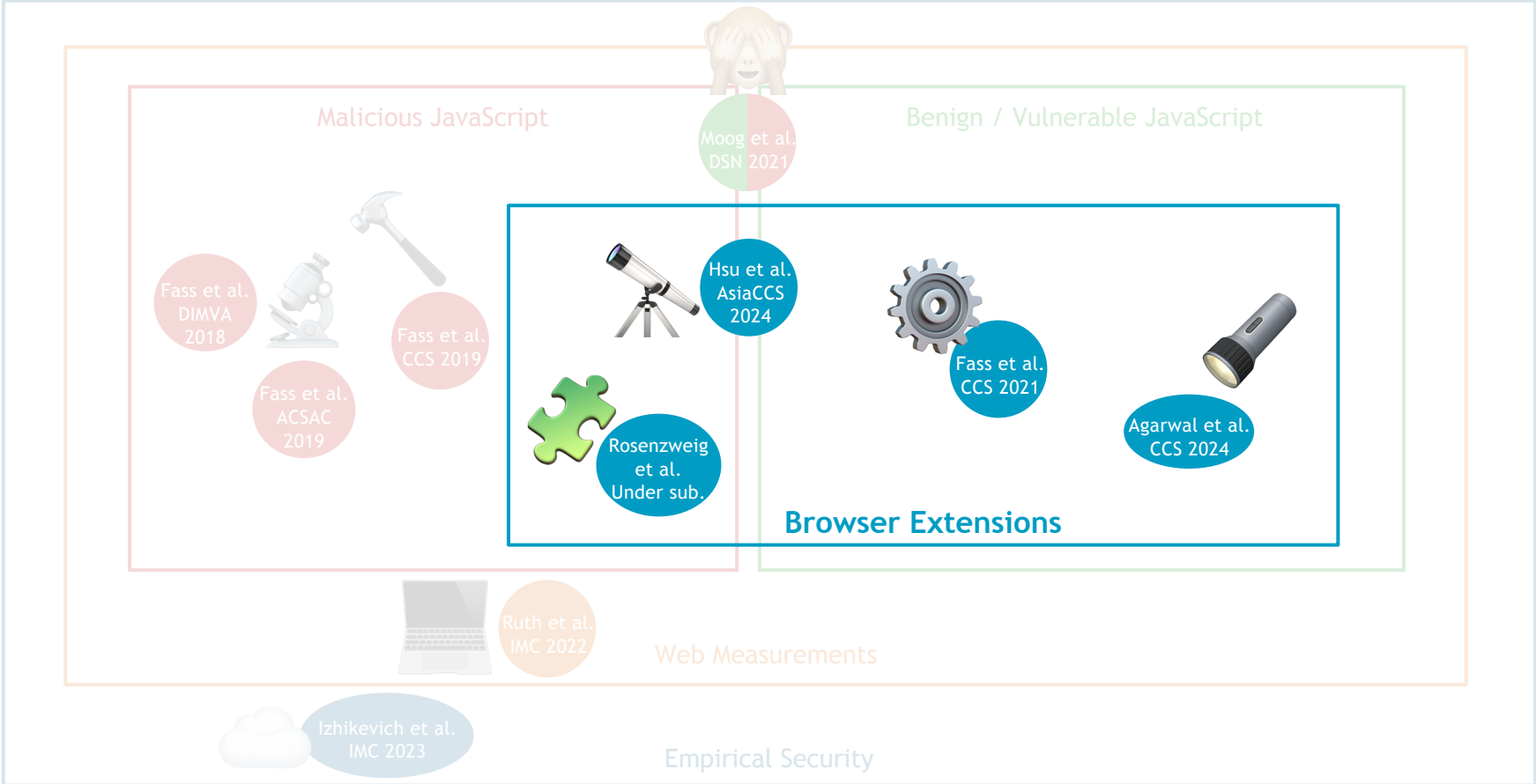


# Browser Extension (In)Security

Aurore Fass

Tenure-Track Faculty at CISPA – Helmholtz Center for Information Security





# What are Browser Extensions?

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



**Adblock** — best ad blocker

Offered by: [getadblock.com](https://getadblock.com)



**Adblock Plus** - free ad blocker

Offered by: [adblockplus.org](https://adblockplus.org)



**Adobe Acrobat**

Offered by: Adobe Inc.



**Avast Online Security**

Offered by: <https://www.avast.com>



**Cisco Webex Extension**

Offered by: [webex.com](https://webex.com)



**Google Translate**

Offered by: [translate.google.com](https://translate.google.com)



**Grammarly for Chrome**

Offered by: [grammarly.com](https://grammarly.com)



**Honey**

Offered by: <https://www.joinhoney.com>



**Pinterest Save Button**

Offered by: [pinterest.com](https://pinterest.com)



**Skype**

Offered by: [www.skype.com](https://www.skype.com)



**uBlock Origin**

Offered by: Raymond Hill (gorhill)




**LastPass: Free Password Manager**

Offered by: LastPass

- 125k** Chrome extensions totaling over **1.6B** active users

# How Safe are Browser Extensions?

- Browser extensions provide **additional functionality**...
- ... so browser extensions need **additional & elevated privileges** compared to web pages
- **Browser extensions are an attractive target for attackers** 

→ 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

- **Violate the Chrome Web Store policies**

- E.g., deceive users, promote unlawful activities, lack a privacy policy

- Contain **vulnerabilities**

- Designed by well-intentioned developers... but contain some vulnerabilities
- E.g., can lead to user-sensitive data exfiltration

# Did you know that...

- **350M users** installed **Security-Noteworthy Extensions** in the last 3 years?
- These **dangerous extensions** stay in the Chrome Web Store *for years*?
- **60%** of extensions have **never received a single update**?

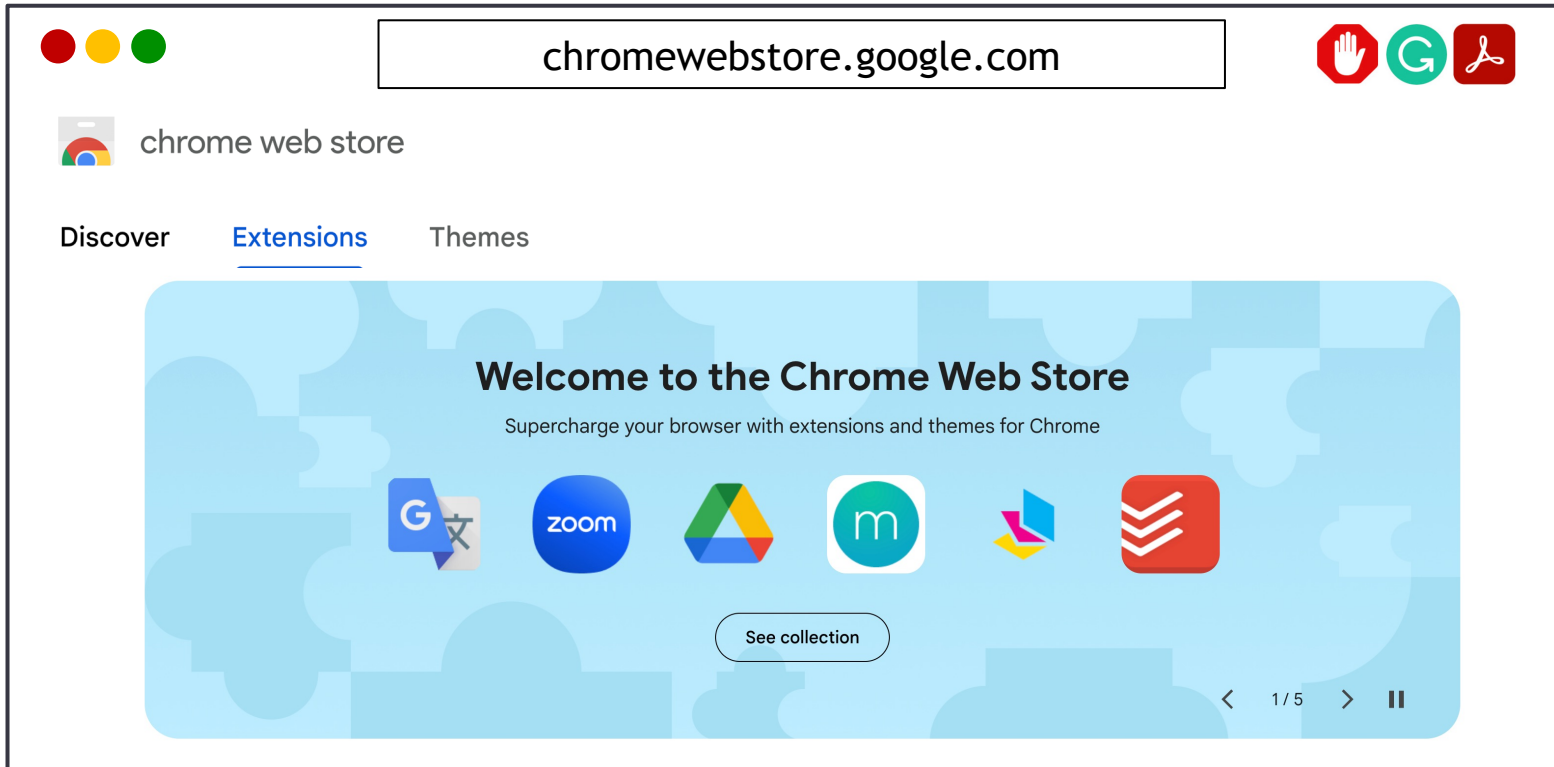


> What is in the Chrome Web Store?



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

# How to Install Extensions or SNE?





# How to Install Extensions or SNE?

The image is a screenshot of a web browser window displaying the Chrome Web Store. The address bar shows 'chromewebstore.google.com'. The page title is 'chrome web store'. The navigation menu includes 'Discover', 'Extensions', and 'Themes', with 'Extensions' being the active tab. A large blue banner is centered on the page, featuring a white text box that reads '>26k SNE (in the last 3 years)'. Below this text is a button labeled 'See collection'. The banner also includes a 'Welcome to the Chrome Web Store' message and a 'Subscribe to our newsletter' link. The background of the banner is a light blue pattern of abstract shapes. In the bottom right corner of the banner, there are navigation controls: a left arrow, '1/5', a right arrow, and a pause icon. The browser window also shows standard OS window controls (red, yellow, green buttons) and three icons in the top right corner: a red hand icon, a green 'G' icon, and a red Adobe PDF icon.

# Browser Extension Collection: Chrome-Stats

chrome-stats.com

Compare and analyze Chrome extensions  
All-in-one platform for competitor research, risk analysis, and growth tracking

Search extensions

127862 Extensions      27638 Themes

Chrome Web Store stats

Number of extension

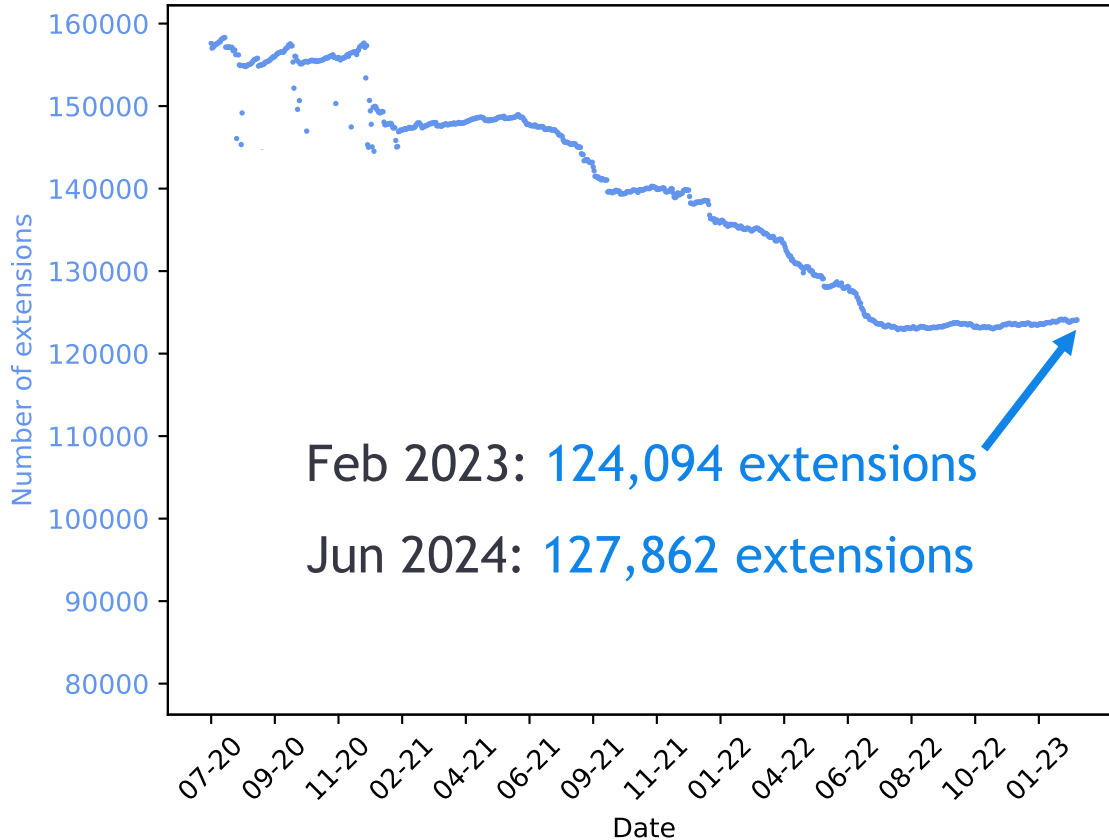
Extensions Themes

03-07 04-10 05-16 06-18 07-20 08-28 09-31 10-03 11-05 12-08 01-10 02-12 03-14 04-16 05-18 06-21 07-23 08-25 09-27 10-29 11-31 12-31 01-15 02-18 03-21 04-23 05-25 06-27 07-29 08-31 09-02

Explore more Chrome extension statistics

Chrome-Stats makes Chrome extension metrics more accessible to everyone, enable competitive analysis, identify bad actors, and help support the growth of good Chrome extensions.

# Number of Extensions in the Chrome Web Store

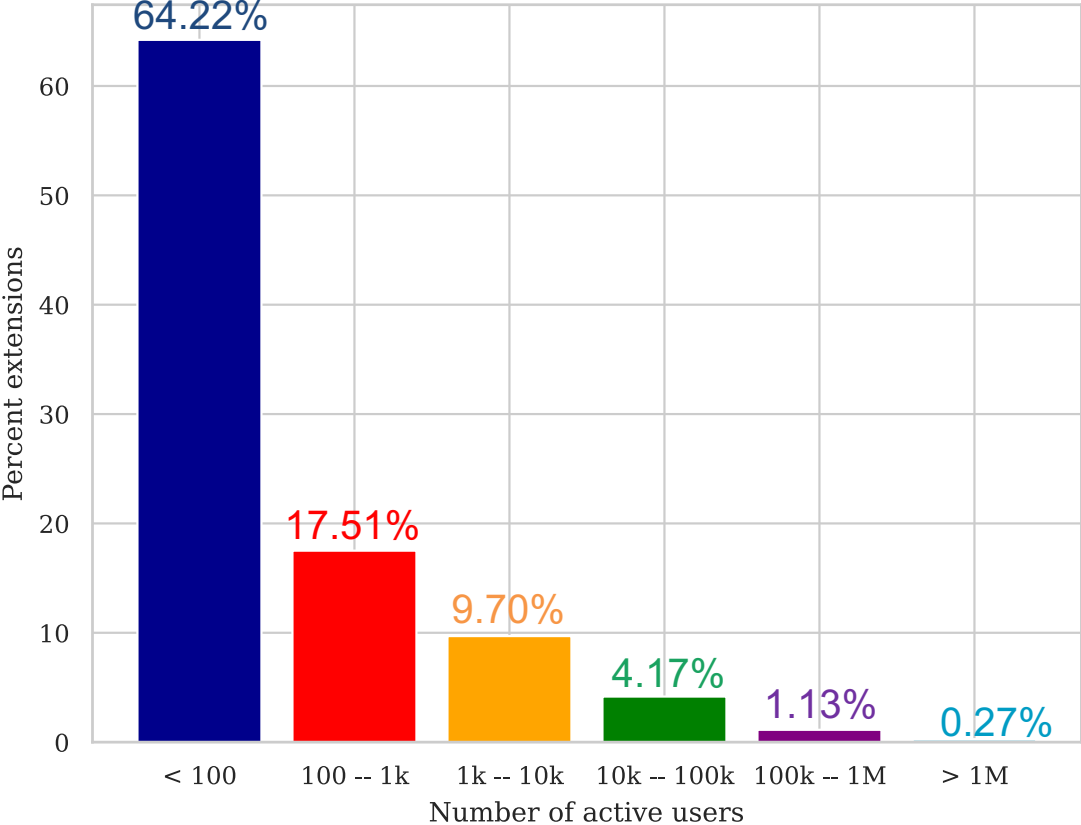


Every month:

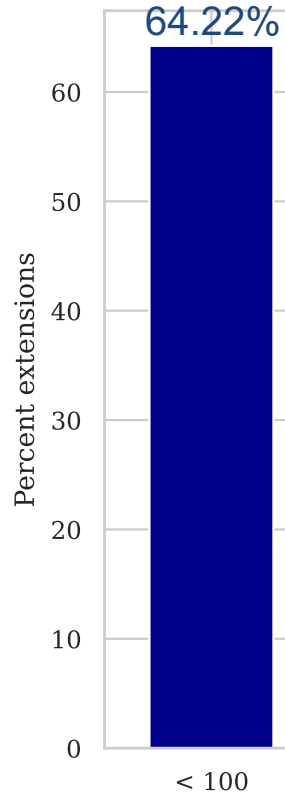
- -3,775 extensions removed
- +2,687 extensions added

➤ **Analyses** on the CWS should be **run regularly**

# Breakdown of Extension Users



# Breakdown of Extension Users



- > 1M users: 500 extensions
- > 10M users: 34 extensions

Number of active users

The “**number of users**” on the CWS for a given extension corresponds to:

*“the number of Chromes with the extension installed that are active and checking in to [their] update servers over the previous seven days only, not for all time. It is not equal to the sum of historic installs minus the sum of historic uninstalls”*

~ Chrome Web Store Developer Support

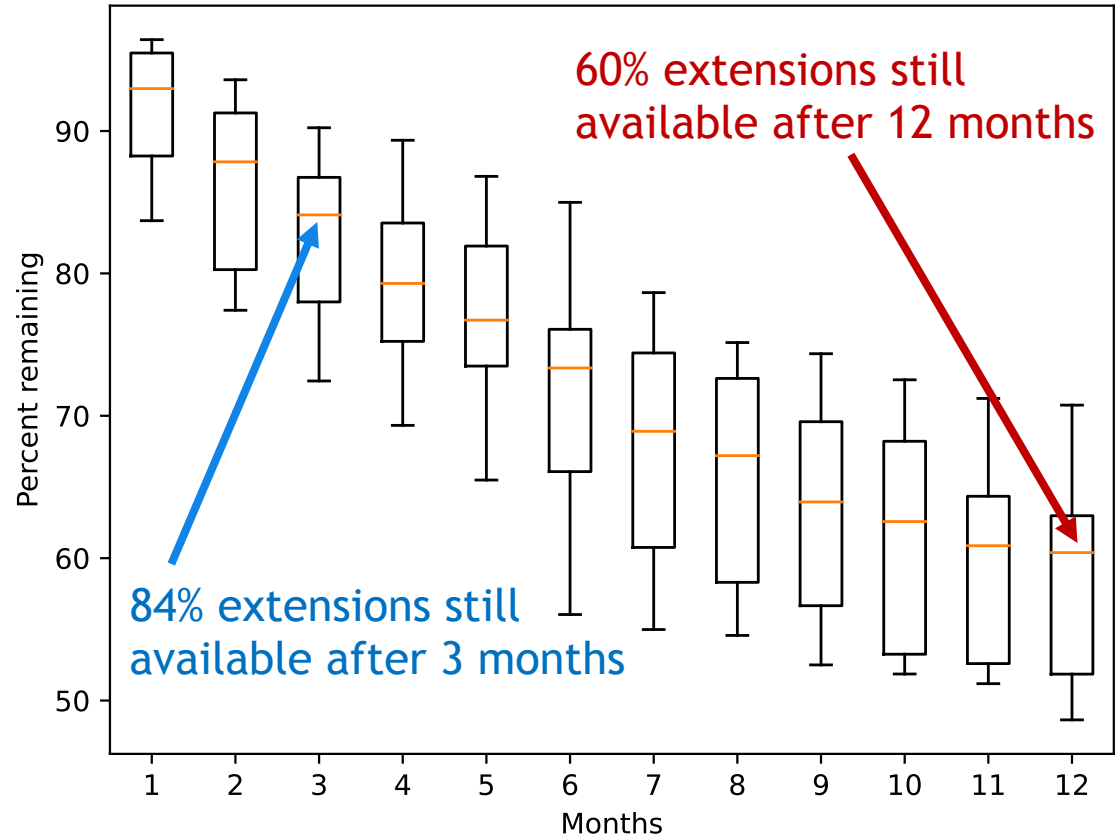
# Life Cycle of Extensions

## Methodology:

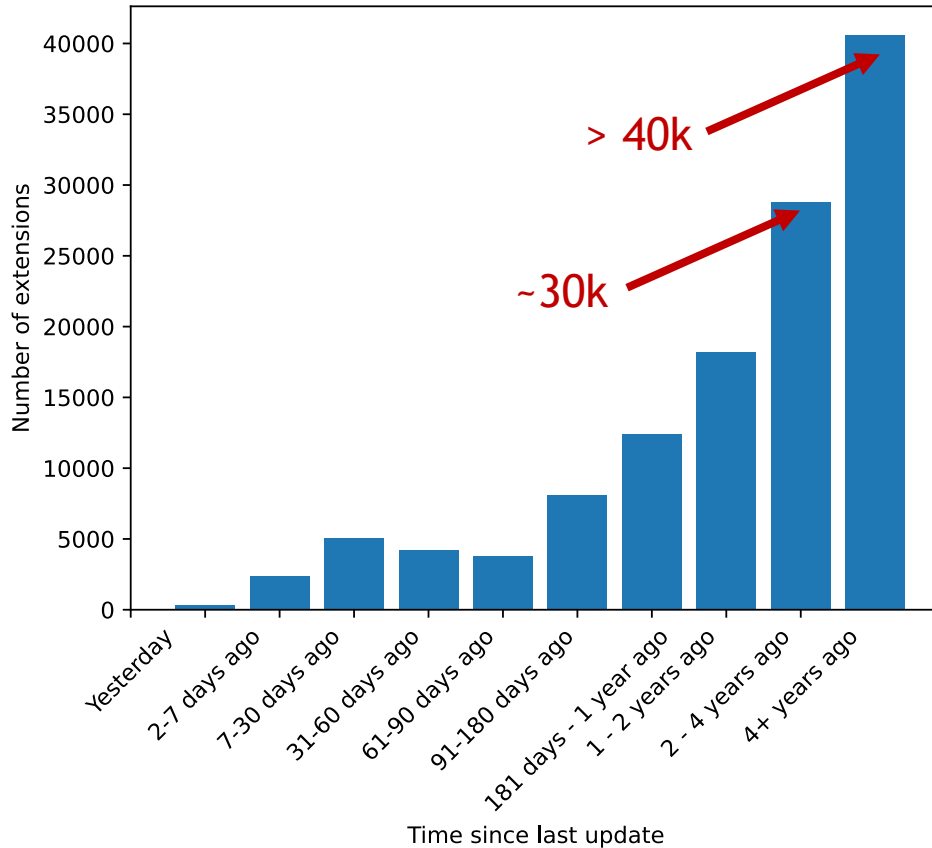
- Collected extensions added to the CWS in Jan–Dec 2021
- Computed the percentage of those extensions still in the CWS 1, 2, ..., 12 months later

➤ Extensions have a very short life cycle

➤ Analyses on the CWS should be run regularly



# Extension Maintenance and Security



- Critical **lack of maintenance** in the CWS
- **60%** of the extensions have **never been updated**
- **Security & privacy implications**



# Malicious Extension Collection: Chrome-Stats

The screenshot shows the 'chrome-stats.com' website with an 'Advanced search' section. A search filter is applied to 'obsoleteReason' with the value 'malware'. The search results table lists various extensions, including 'Video downloader for Instagram™', 'Voice Aloud Reader for pc,windows and mac (Free Use)', 'YTBlock - Adblock para Youtube', 'OVO Official', 'Snake', and 'Settings for Chrome'. The 'obsoleteReason' column for all listed items is 'malware'.

chrome-stats.com

Advanced search

Search extensions Search reviews

obsoleteReason = malware

Search

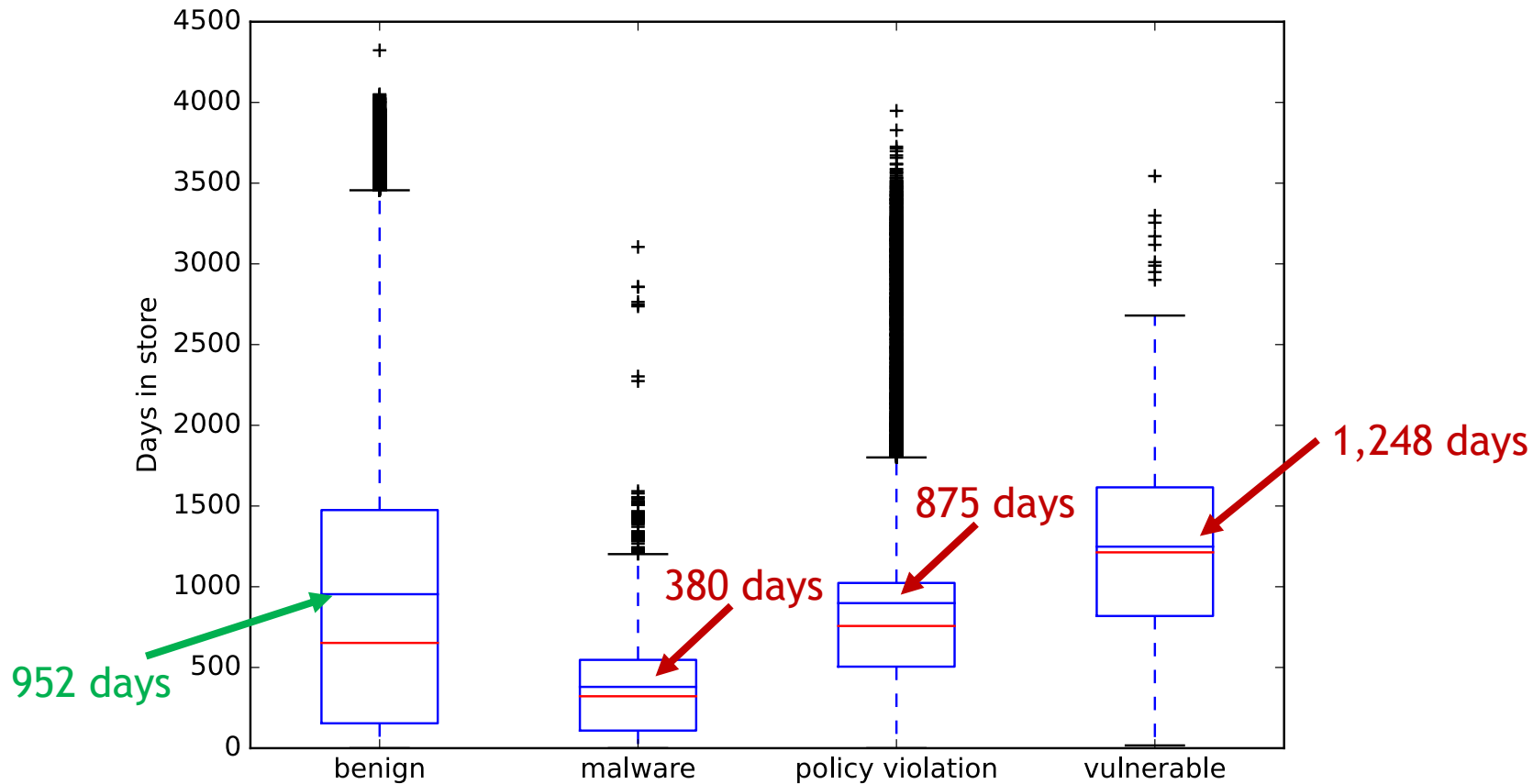
Export Saved query Visible columns + Add condition

10944 results. Page 1 of 438.

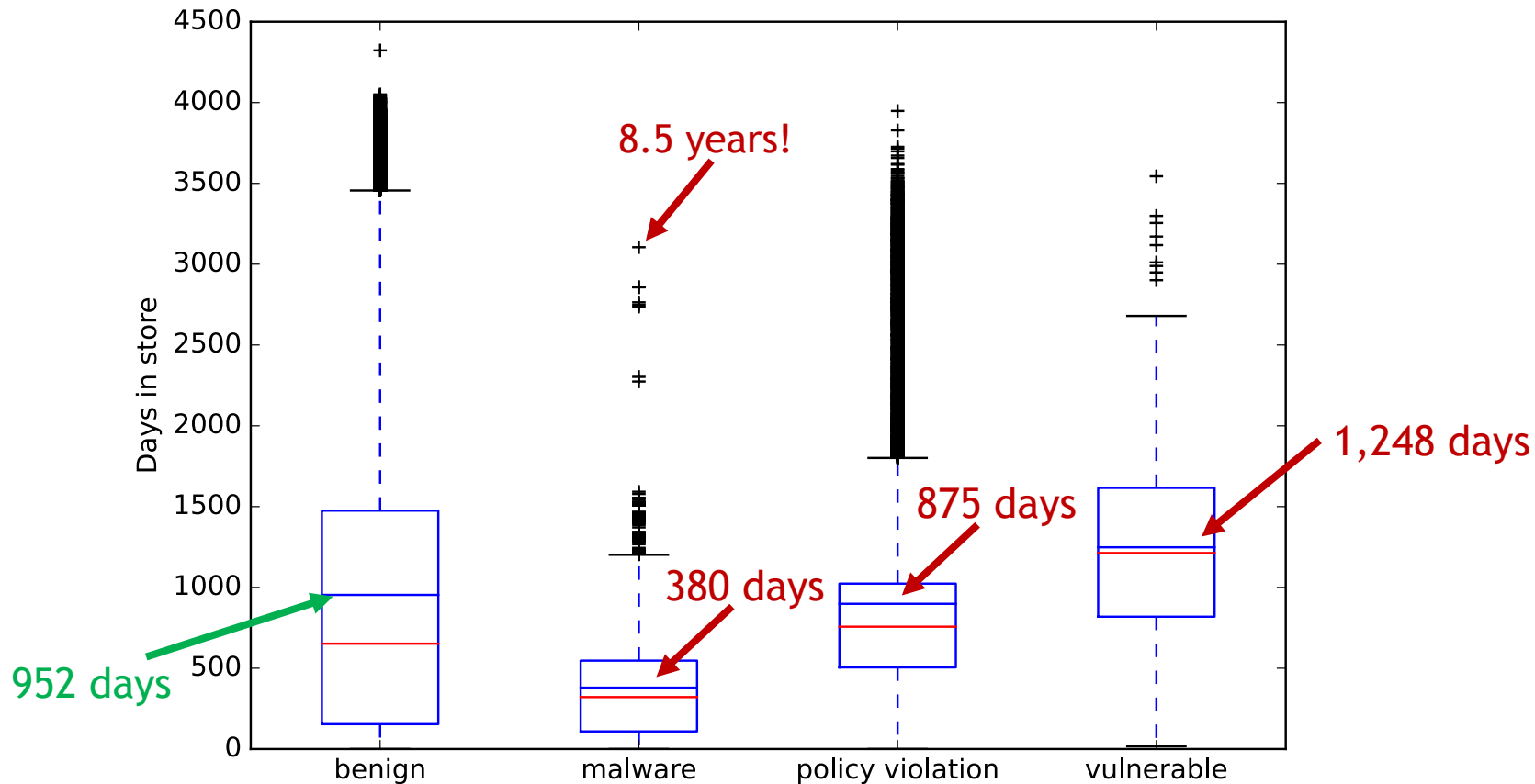
logo	name	userCount	author	ratingValue	ratingCount	obsoleteReason	lastUpdate	creationDate
	Video downloader for Instagram™	100000	<a href="https://instagram-downloader.instvid.site">https://instagram-downloader.instvid.site</a>	4.27	30	malware	2024-03-07	2022-11-15
	Voice Aloud Reader for pc,windows and mac (Free Use)	11	<a href="https://voicealoudreaderforpc.blogspot.com">https://voicealoudreaderforpc.blogspot.com</a>	0.00	0	malware	2024-03-06	2024-03-06
	YTBlock - Adblock para Youtube	9000	YTAdblock	4.91	57	malware	2024-03-01	2024-02-09
	OVO Official	30	<a href="https://ovogame.pro">https://ovogame.pro</a>	0.00	0	malware	2024-02-28	2024-02-28
	Snake	50000	<a href="https://snake.9834722.xyz">https://snake.9834722.xyz</a>	4.19	52	malware	2024-02-27	2021-10-04
	Settings for Chrome	600000	Chrome Settings	3.75	4	malware	2024-02-27	2022-06-24

Category	#Extensions Metadata collected	#Extensions Code collected	When collected
SNE	26,014	16,377	Before May 1, 2023
- Malware-containing	10,426	6,587	Before May 1, 2023
- Policy-violating	15,404	9,638	Before May 1, 2023
- Vulnerable [1]	184	152	March 16, 2021
Benign extensions	226,762	92,482	Before May 1, 2023

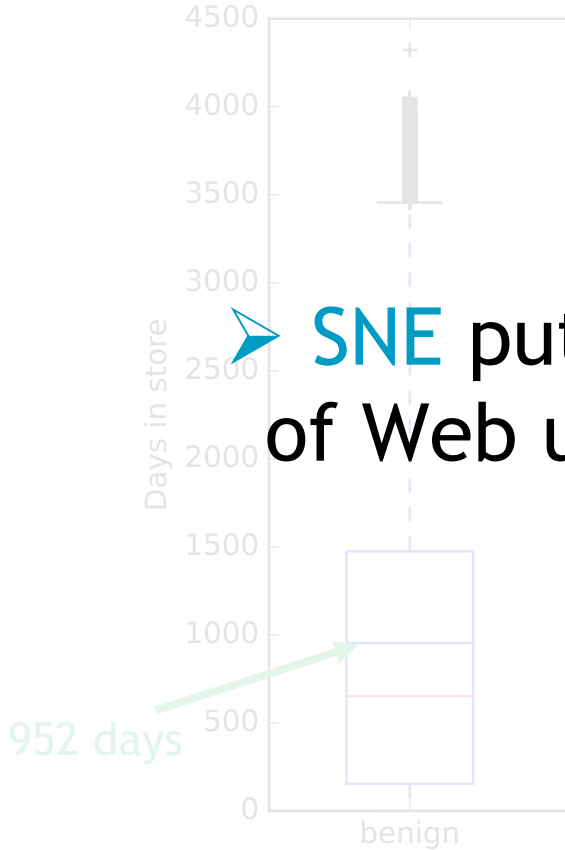
# Number of Days in the CWS



# Number of Days in the CWS

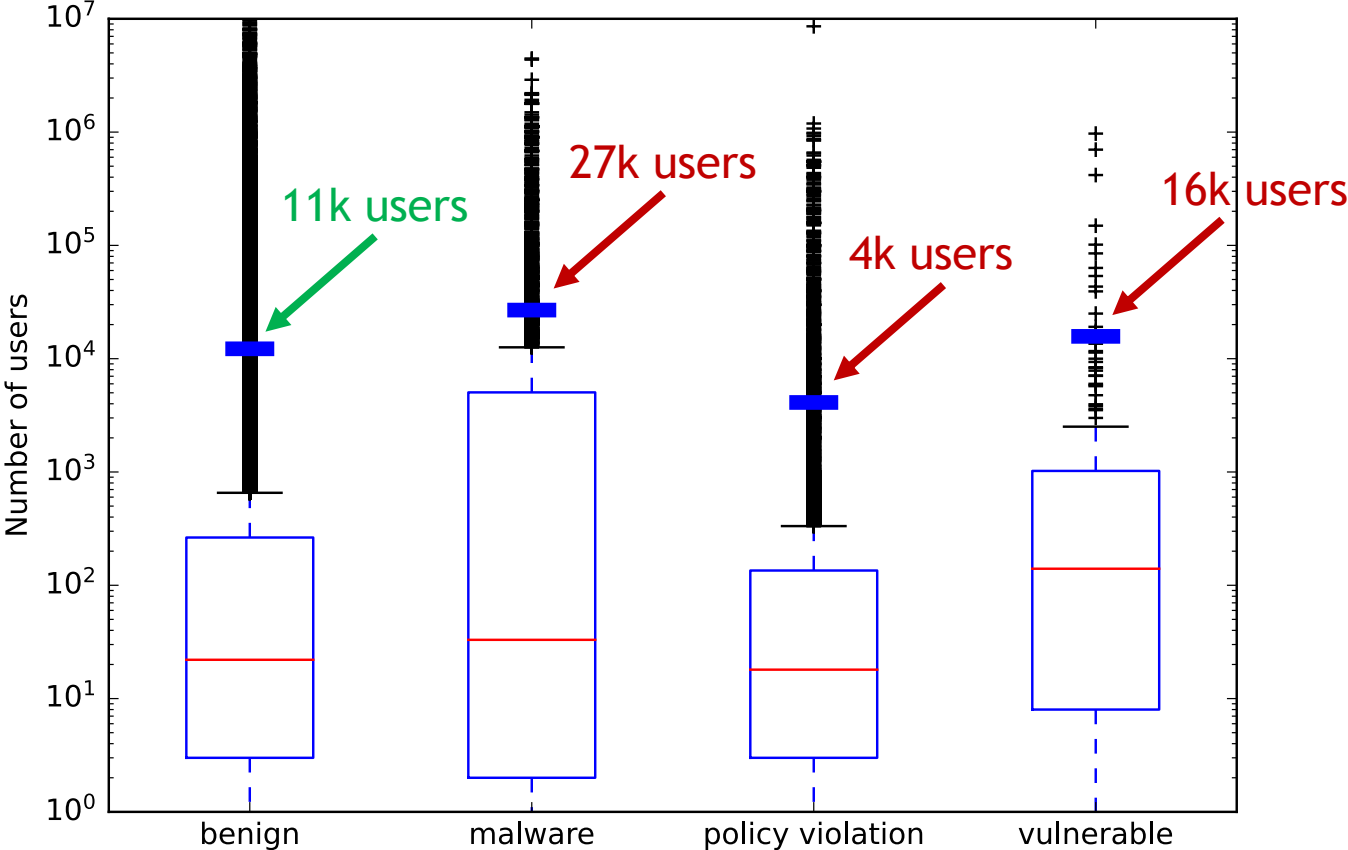


# Number of Days in the CWS

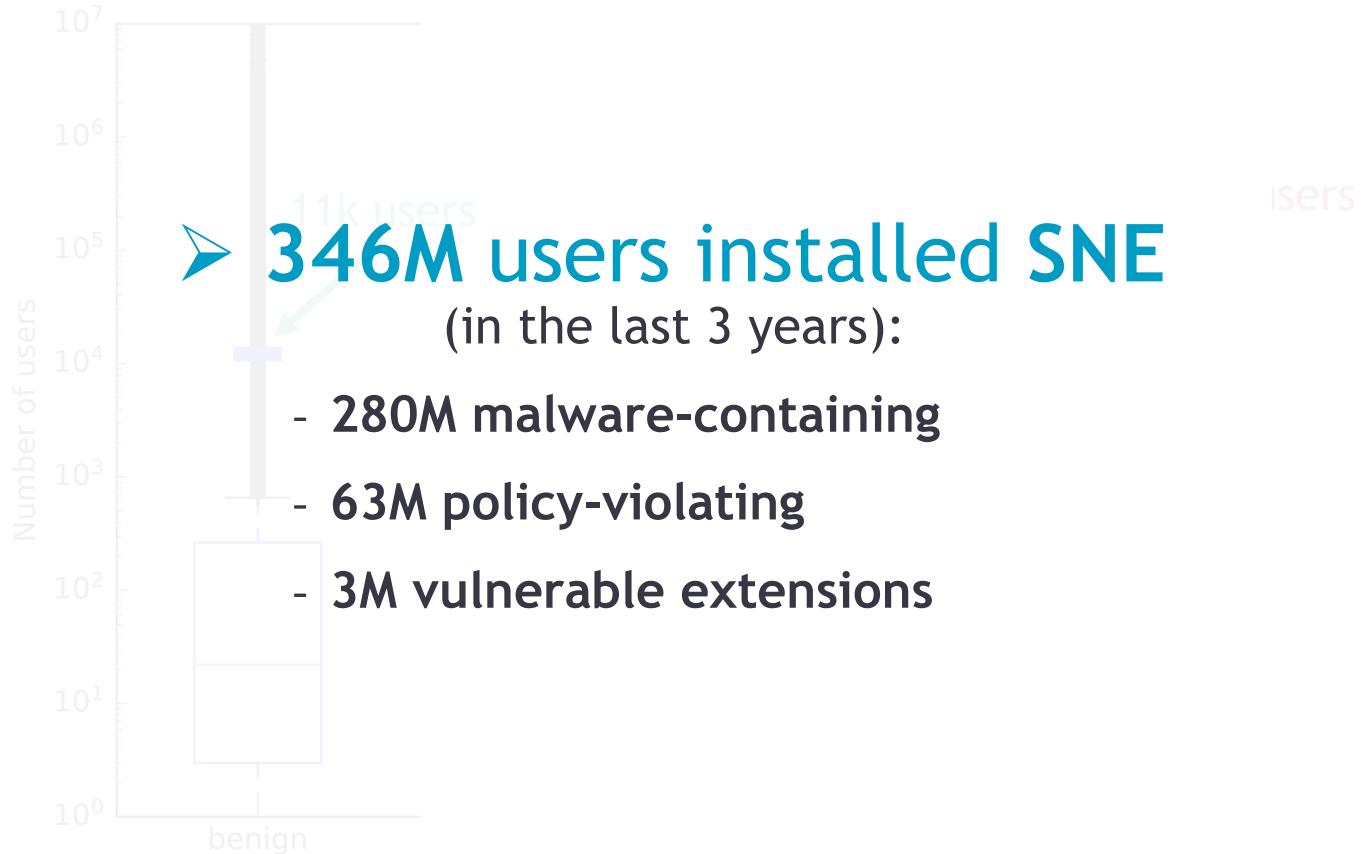


➤ SNE put the security & privacy of Web users at risk *for years*

# Number of Users



# Number of Users



- Source-code comparison across extensions (*ssdeep* fuzzy hash)
- Clustering similar extensions together (i.e., 100% *ssdeep* overlap)
- 3,270 clusters with [2; 1,397] extensions (20,822 extensions clustered)



- 3,270 clusters:
  - 2,296 clusters contain just benign extensions
  - 321 clusters only SNE
    - 14 clusters with > 100 SNE and 2 with > 863 SNE each
    - Analyzing extensions for similarities could enable to detect SNE
  - 653 clusters of benign (5,552 extensions) and SNE (5,126)
    - Extensions in a cluster with SNE should be flagged for more analyses

# How to Detect Security-Noteworthy Extensions?

- Contain **malware**

- Designed by malicious actors to harm victims
- E.g., steal user-sensitive data, track users, propagate malware

- **Violate the Chrome Web Store policies**

- E.g., deceive users, promote unlawful activities, lacking a privacy policy

- Contain **vulnerabilities**



Fass et al.  
CCS 2021

- Designed by well-intentioned developers... but contain some vulnerabilities
- E.g., can lead to user-sensitive data exfiltration

# How to Detect Security-Noteworthy Extensions?

- Contain malware

- Designed by malicious actors to harm victims

- E.g., steal user sensitive data, track users, propagate malware

- Violate the Chrome Web Store policies

- E.g., deceive users, promote unlawful activities, lacking a privacy policy

- Contain **vulnerabilities**



Fass et al.  
CCS 2021

- Designed by well-intentioned developers... but contain some vulnerabilities

- E.g., can lead to user-sensitive data exfiltration

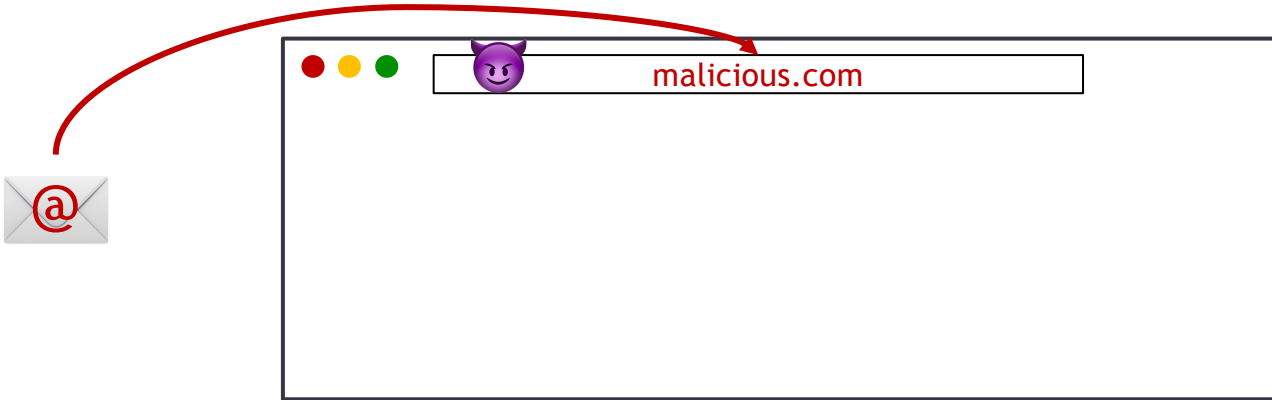
# Analysis of Vulnerable Extensions

Challenging to detect due to their inherently benign intent (*benign-but-buggy*)



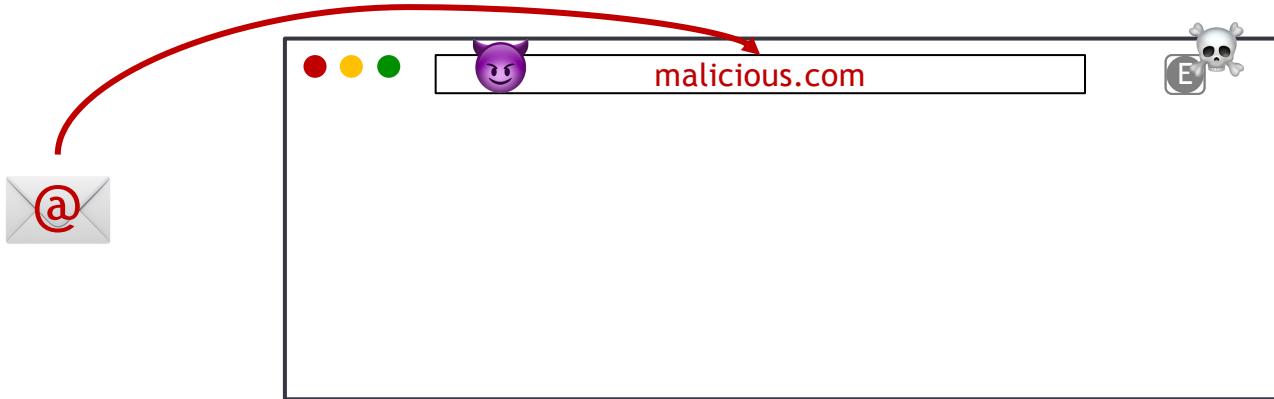
# Analysis of Vulnerable Extensions

Challenging to detect due to their inherently benign intent (*benign-but-buggy*)



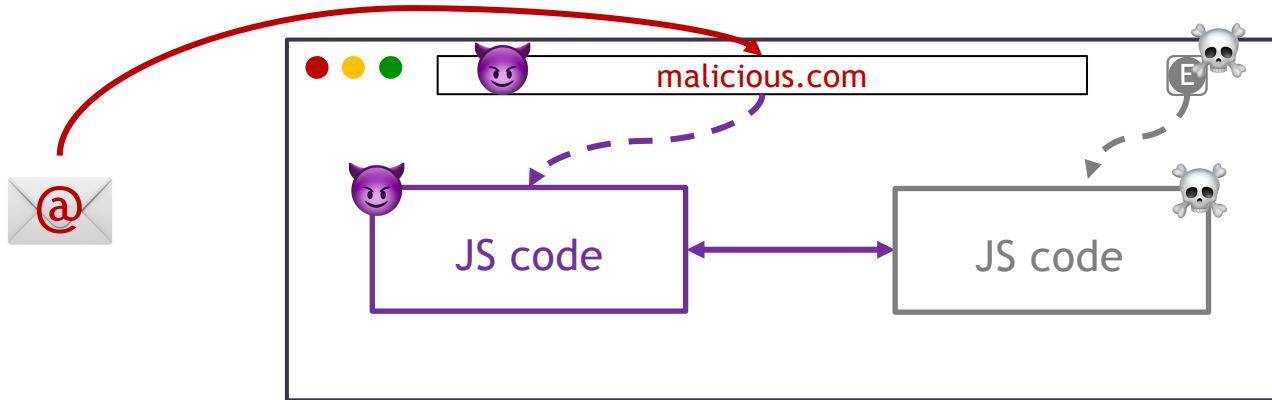
# Analysis of Vulnerable Extensions

Challenging to detect due to their inherently benign intent (*benign-but-buggy*)



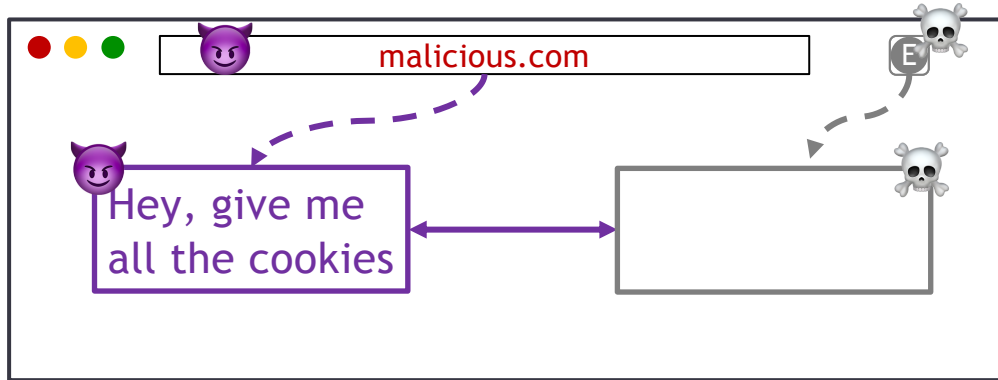
# Analysis of Vulnerable Extensions

Challenging to detect due to their inherently benign intent (*benign-but-buggy*)



# Analysis of Vulnerable Extensions

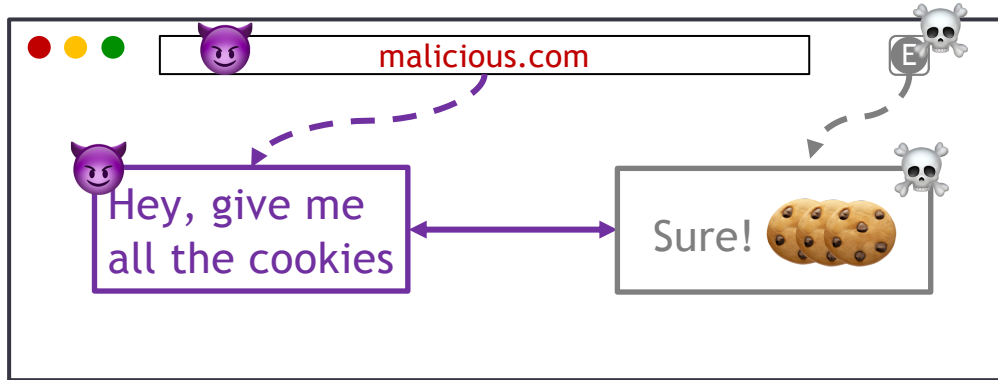
Challenging to detect due to their inherently benign intent (*benign-but-buggy*)





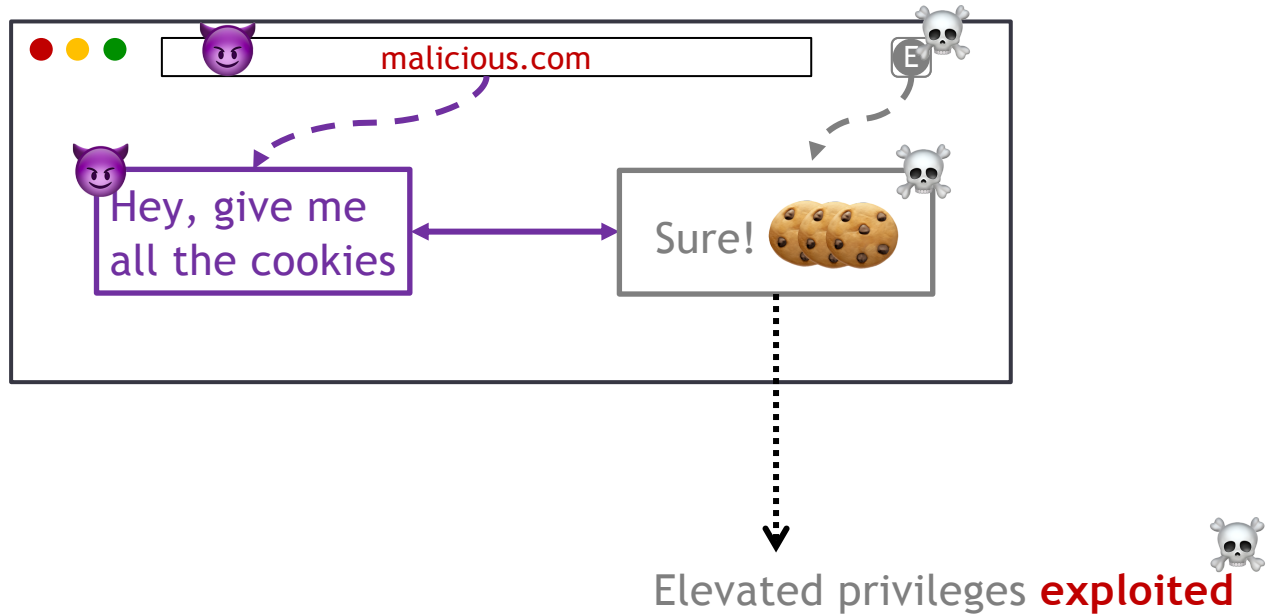
# Analysis of Vulnerable Extensions

Challenging to detect due to their inherently benign intent (*benign-but-buggy*)



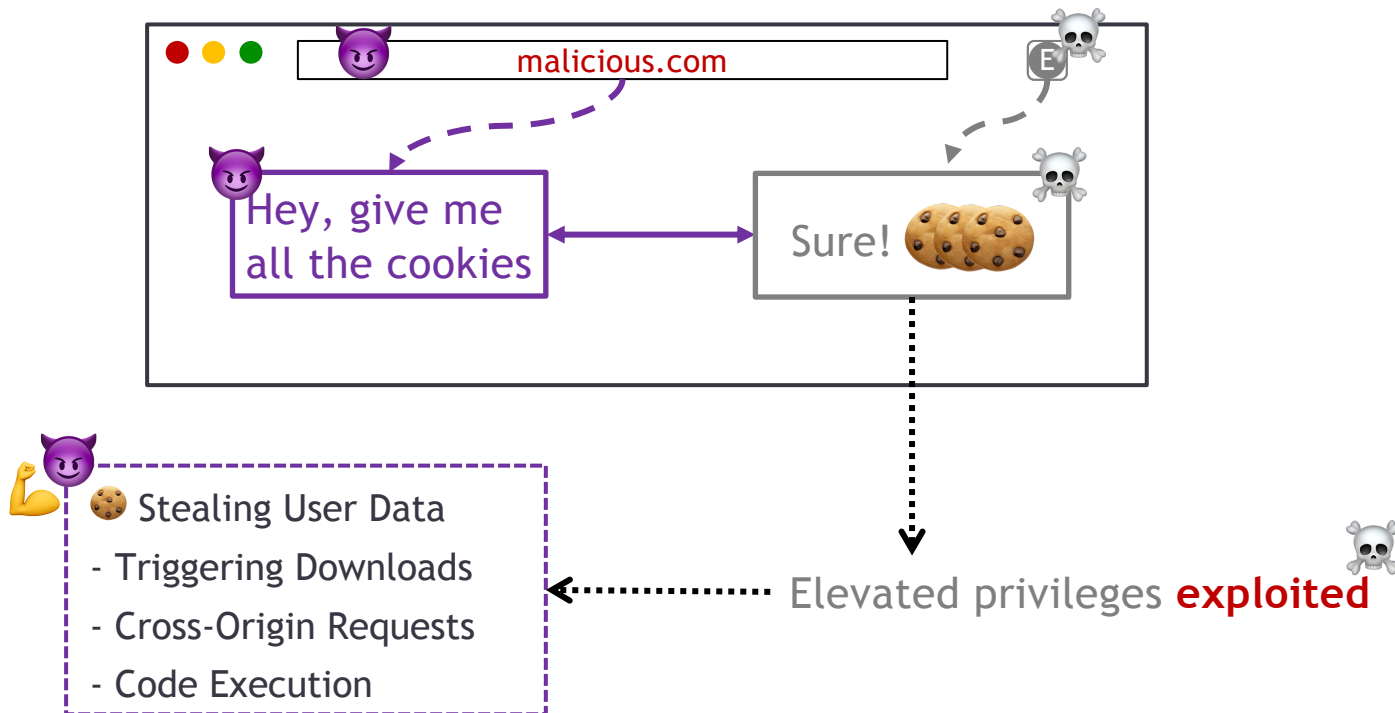
# Analysis of Vulnerable Extensions

Challenging to detect due to their inherently benign intent (*benign-but-buggy*)



# Analysis of Vulnerable Extensions

Challenging to detect due to their inherently benign intent (*benign-but-buggy*)



# Detecting Vulnerable Extensions



Fass et al.  
CCS 2021

**DOUBLEX: Statically Detecting Vulnerable Data Flows in Browser Extensions at Scale**  
Aurora Fass, Dolie Francis Somé, Michael Backes, and Ben Stock  
1 Introduction

> **DOUBLEX: Statically Detecting Vulnerable Data Flows in Browser Extensions**  
In ACM CCS 2021. Aurora Fass, Dolie Francis Somé, Michael Backes, and Ben Stock

# Detecting Vulnerable Extensions



Fass et al.  
CCS 2021

**DOUBLEX: Statically Detecting Vulnerable Data Flows in Browser Extensions at Scale**  
Aurore Fass, Dolière Francis Somé, Michael Backes, and Ben Stock

**Abstract**  
Browser extensions are a popular way to customize a browser's behavior. However, they are also a common source of security vulnerabilities. In this paper, we present DOUBLEX, a static analysis tool that detects vulnerable data flows in browser extensions. DOUBLEX is designed to be efficient and accurate, and is able to analyze a large number of extensions. We evaluate DOUBLEX on a dataset of 10,000 browser extensions and show that it is able to detect a high number of vulnerabilities. We also discuss the challenges of static analysis for browser extensions and the implications of our findings for browser security.

**1 Introduction**  
Browser extensions are a popular way to customize a browser's behavior. However, they are also a common source of security vulnerabilities. In this paper, we present DOUBLEX, a static analysis tool that detects vulnerable data flows in browser extensions. DOUBLEX is designed to be efficient and accurate, and is able to analyze a large number of extensions. We evaluate DOUBLEX on a dataset of 10,000 browser extensions and show that it is able to detect a high number of vulnerabilities. We also discuss the challenges of static analysis for browser extensions and the implications of our findings for browser security.

**CCS Concepts**  
Security and privacy → Software security protection; Software security protection → Software security protection; Software security protection → Software security protection.

**Keywords**  
Browser extensions; Static analysis; Vulnerability detection; Data flows; Security.

> **DOUBLEX: Statically Detecting Vulnerable Data Flows in Browser Extensions**  
In ACM CCS 2021. Aurore Fass, Dolière Francis Somé, Michael Backes, and Ben Stock



# Detecting Vulnerable Extensions



Fass et al.  
CCS 2021

**DOUBLEX: Statically Detecting Vulnerable Data Flows in Browser Extensions at Scale**  
Aurore Fass, Dolière Francis Somé, Michael Backes, and Ben Stock  
1 Introduction

> **DOUBLEX: Statically Detecting Vulnerable Data Flows in Browser Extensions**

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



Malicious web page

Content script

Background page

Vulnerable extension

# Detecting Vulnerable Extensions



Fass et al.  
CCS 2021

**DOUBLEX: Statically Detecting Vulnerable Data Flows in Browser Extensions at Scale**  
Aurore Fass, Dolière Francis Somé, Michael Backes, and Ben Stock  
CCS 2021

**Abstract**  
Browser extensions are a popular way to customize web browsing. However, they are also a common source of security vulnerabilities. In this paper, we present DOUBLEX, a static analysis tool that detects vulnerable data flows in browser extensions. DOUBLEX is the first tool to detect vulnerable data flows in browser extensions at scale. It is able to detect vulnerable data flows in over 100,000 browser extensions. DOUBLEX is able to detect vulnerable data flows in browser extensions by analyzing the control flow and data flow of the code. DOUBLEX is able to detect vulnerable data flows in browser extensions by analyzing the control flow and data flow of the code. DOUBLEX is able to detect vulnerable data flows in browser extensions by analyzing the control flow and data flow of the code.

**Introduction**  
Browser extensions are a popular way to customize web browsing. However, they are also a common source of security vulnerabilities. In this paper, we present DOUBLEX, a static analysis tool that detects vulnerable data flows in browser extensions. DOUBLEX is the first tool to detect vulnerable data flows in browser extensions at scale. It is able to detect vulnerable data flows in over 100,000 browser extensions. DOUBLEX is able to detect vulnerable data flows in browser extensions by analyzing the control flow and data flow of the code. DOUBLEX is able to detect vulnerable data flows in browser extensions by analyzing the control flow and data flow of the code. DOUBLEX is able to detect vulnerable data flows in browser extensions by analyzing the control flow and data flow of the code.

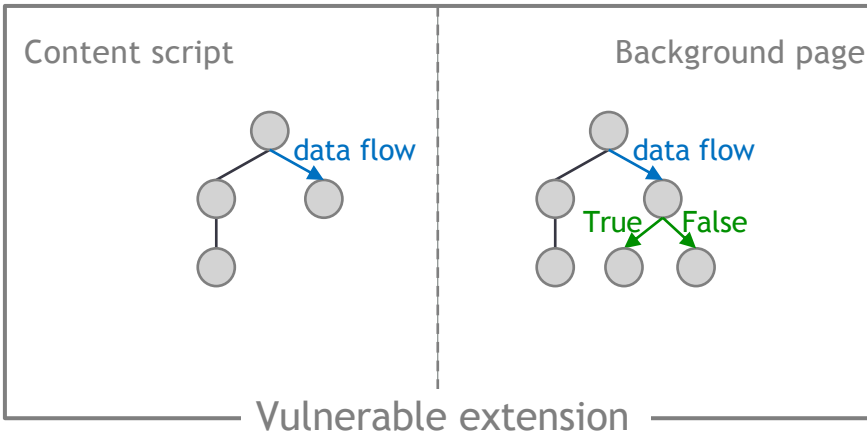
**CCS keywords**  
Static analysis, Browser extensions, Security, Vulnerability detection

## > DOUBLEX: Statically Detecting Vulnerable Data Flows in Browser Extensions

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



Malicious web page



### Per-component JS code abstraction

- AST (Abstract Syntax Tree)
- Control flow
- Data flow
- Pointer analysis





# Detecting Vulnerable Extensions



Fass et al.  
CCS 2021

**DOUBLEX: Statically Detecting Vulnerable Data Flows in Browser Extensions at Scale**  
Aurore Fass, Dolière Francis Somé, Michael Backes, and Ben Stock  
ACM CCS 2021

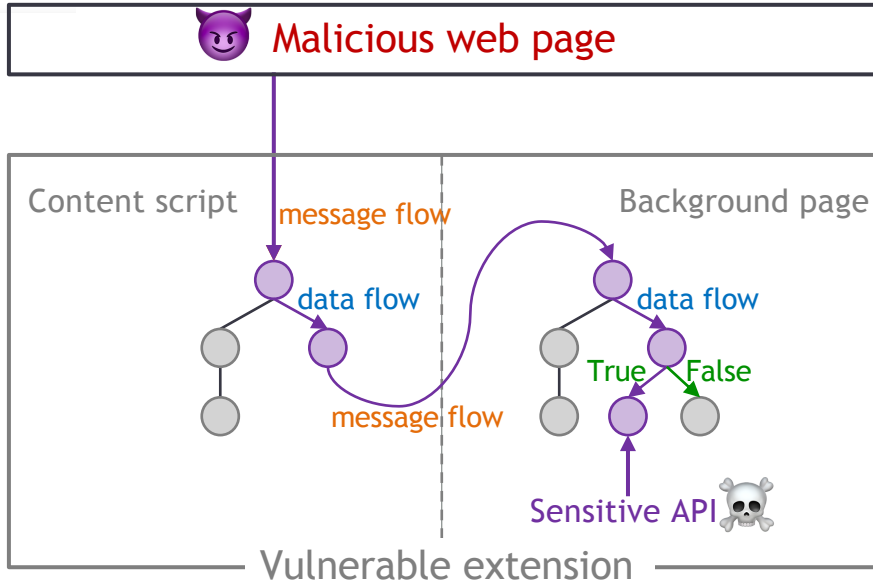
**Abstract**  
Browser extensions are a popular way to customize the user experience of web browsers. However, they are also a common source of security vulnerabilities. In this paper, we present DOUBLEX, a static analysis tool that detects vulnerable data flows in browser extensions. DOUBLEX is based on a novel abstraction of JavaScript code that allows for the detection of data flows that are vulnerable to attacks. We evaluate DOUBLEX on a large dataset of browser extensions and show that it can detect a significant number of vulnerabilities that were previously undetected.

**1 Introduction**  
Browser extensions are a popular way to customize the user experience of web browsers. However, they are also a common source of security vulnerabilities. In this paper, we present DOUBLEX, a static analysis tool that detects vulnerable data flows in browser extensions. DOUBLEX is based on a novel abstraction of JavaScript code that allows for the detection of data flows that are vulnerable to attacks. We evaluate DOUBLEX on a large dataset of browser extensions and show that it can detect a significant number of vulnerabilities that were previously undetected.

**CCS keywords**  
Security and privacy, Web applications, Browser extensions, Static analysis, Vulnerability detection

## > DOUBLEX: Statically Detecting Vulnerable Data Flows in Browser Extensions

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



### Per-component JS code abstraction

- AST (Abstract Syntax Tree)
- Control flow
- Data flow
- Pointer analysis

### Extension Dependence Graph (EDG)

- Message interactions

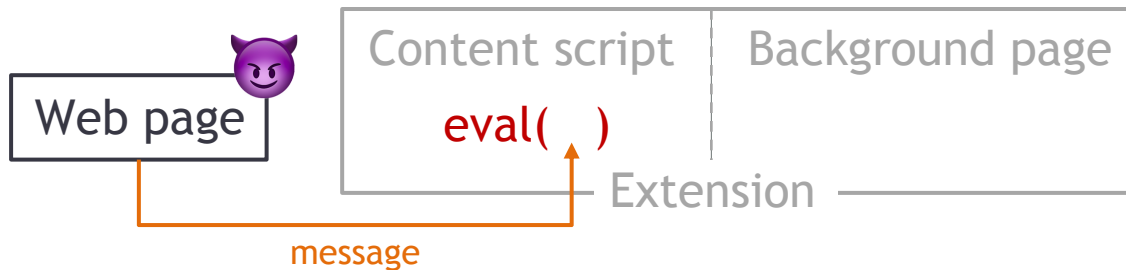
### Suspicious data flow tracking

- Detects any path between an attacker & sensitive APIs



# Simplified Example of a Vulnerability

```
// Content script code  
window.addEventListener("message", function(event) {  
  
    eval(event.data);  
  
})
```



# Detecting Vulnerable Extensions with DOUBLEX

Analyzed 155k Chrome extensions from 2021 with DOUBLEX

- **184 vulnerable Chrome extensions**
- **Impacting 3M users**

# Life Cycle of Vulnerable Chrome Extensions

- Analyzed 165k extensions from 2020 with DOUBLEX
    - 193 vulnerable extensions (184 in 2021)
    - vulnerability disclosure for 35 extensions (48 extensions when including 2021)
  - Comparison of vulnerable extensions in 2020 vs. 2021
    - not in the CWS anymore: 30 / 193
    - vulnerability fixed: 3 / 193
    - turned vulnerable: 5 / 184
    - new vulnerable: 19 / 184
- **still vulnerable: 160 (83%)**    ➤ **Need to prevent vulnerable extensions from entering the CWS → DOUBLEX**

# Detecting Vulnerable Extensions with DOUBLEX

Analyzed 155k Chrome extensions from 2021 with DOUBLEX

- **184 vulnerable Chrome extensions**
- Impacting **3M users**
  
- **Precision: 89%** of the flagged extensions are vulnerable
- **Recall: 93%** of known vulnerabilities [2] are detected
  
- **Integration** in the **vetting process** conducted by Google
- **Available online**, for developers  
(even in other fields!)



 Aurore54F/DoubleX

# Security-Noteworthy Extensions

- Contain **malware**

- Designed by malicious actors to harm victims
- E.g., steal user-sensitive data, track users, propagate malware

- **Violate** the Chrome Web Store **policies**

- E.g., deceive users, promote unlawful activities, lacking a privacy policy

- Contain **vulnerabilities**

- Designed by well-intentioned developers but contain some vulnerabilities
- E.g., can lead to user-sensitive data exfiltration



Fass et al.  
CCS 2021

- Can be **fingerprinted**

- Designed by well-intentioned developers...
- ... but can lead to, e.g., tracking users across sites, inferring sensitive user information

# Security-Noteworthy Extensions

- Contain **malware**

- Designed by malicious actors to harm victims
- E.g., steal user-sensitive data, track users, propagate malware

- **Violate** the Chrome Web Store **policies**

- E.g., deceive users, promote unlawful activities, lacking a privacy policy

- Contain **vulnerabilities**

- Designed by well-intentioned developers but contain some vulnerabilities
- E.g., can lead to user-sensitive data exfiltration



Fass et al.  
CCS 2021

- Can be **fingerprinted**

- Designed by well-intentioned developers...
- ... but can lead to, e.g., tracking users across sites, inferring sensitive user information

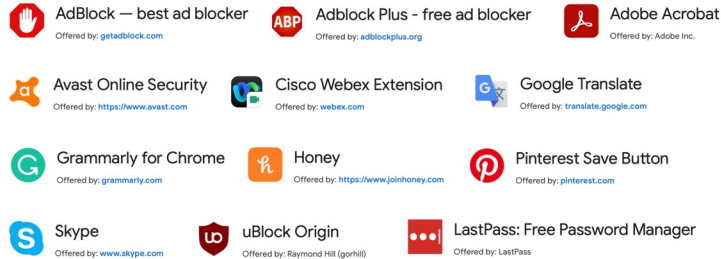


Agarwal et al.  
CCS 2024



# Takeaways – Browser Extension (In)Security

## Browser Extensions are Popular



- 125k Chrome extensions totaling over 1.6B active users

## Security-Noteworthy Extensions (SNE)

- Contain **malware**
- Violate the Chrome Web Store **policies**
- Contain **vulnerabilities**
- Can be **fingerprinted**



Thank you

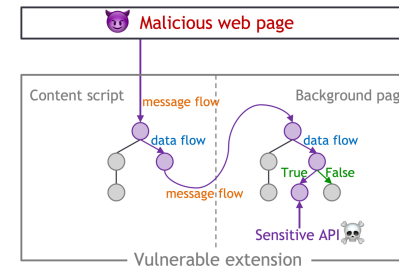
## What is in the Chrome Web Store?

- 350M users** installed SNE in the last 3 years
- These SNE stay in the Chrome Web Store *for years*
- Extensions have a **short life cycle** in the CWS (60% stay 1 year)
- Critical **lack of maintenance** in the CWS (60% received no update)



Hsu et al. AsiaCCS 2024

## Detecting Vulnerable Extensions with DOUBLEX



Fass et al. CCS 2021

Aurore54F/DoubleX

- DOUBLEX **detects suspicious data flows** in browser extensions  
**184 vulnerable extensions** | **Precision: 89%** | **Recall: 93%**

- [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*

- [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*

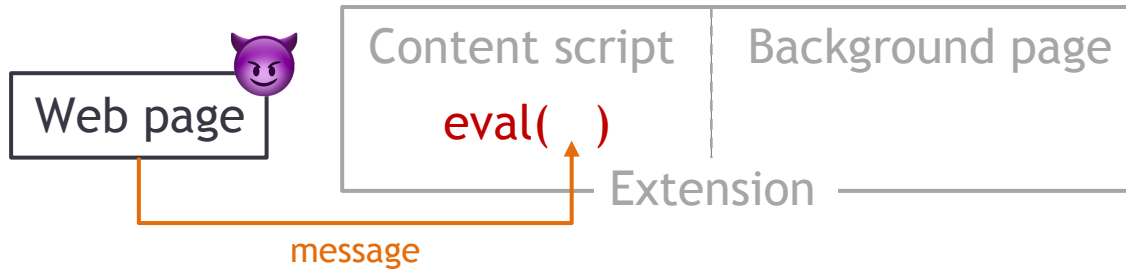
Additional slides about some questions asked

# Per-Component JavaScript Code Abstraction

```
// Content script code
window.addEventListener("message", function(event) {

    eval(event.data);

})
```



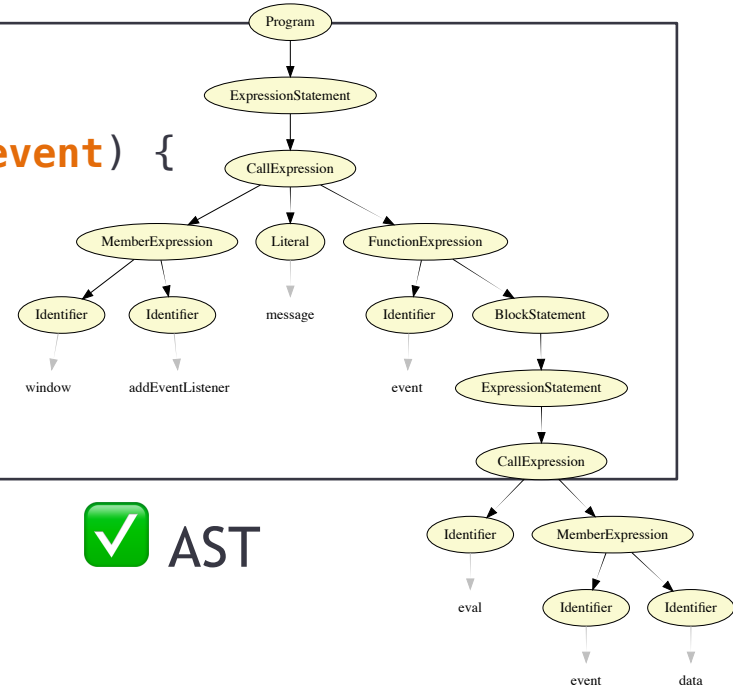
# Per-Component JavaScript Code Abstraction

```
// Content script code
```

```
window.addEventListener("message", function(event) {
```

```
    eval(event.data);
```

```
});
```



Abstract code representation



AST

```
// Content script code
window.addEventListener("message", function(event) {
    eval(event.data);
})
```



Abstract code representation



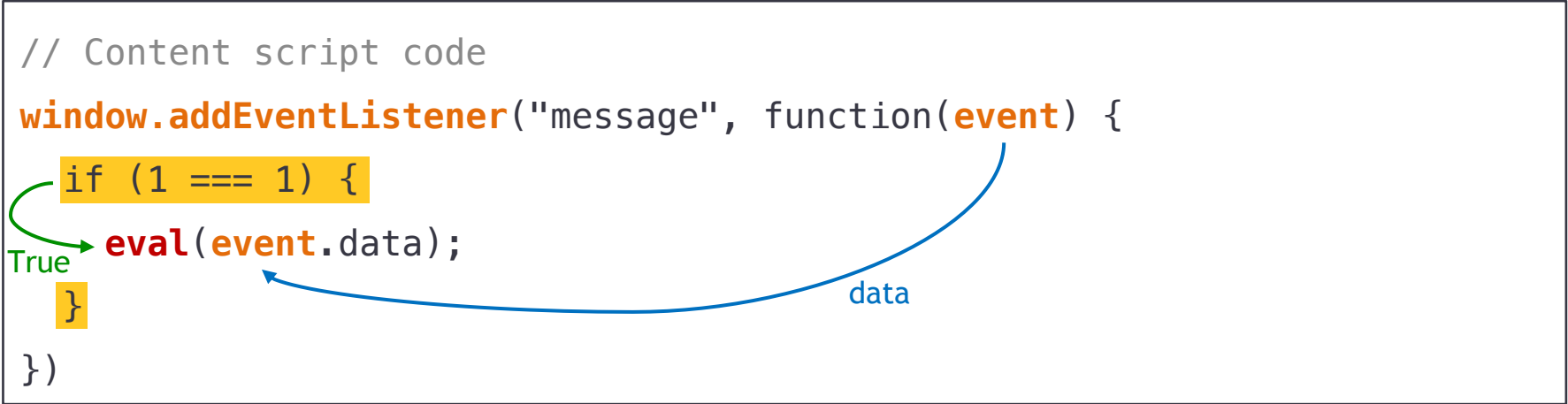
AST

– variable dependencies



data flow

```
// Content script code
window.addEventListener("message", function(event) {
  if (1 === 1) {
    eval(event.data);
  }
})
```



Abstract code representation



✓ AST

– conditions



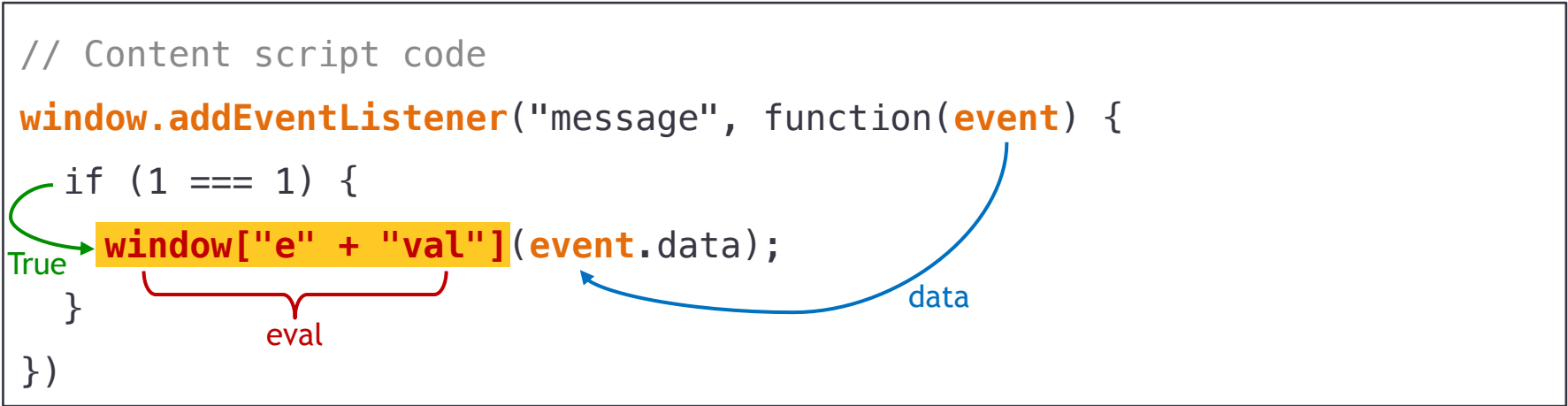
✓ control flow

– variable dependencies



✓ data flow

```
// Content script code
window.addEventListener("message", function(event) {
  if (1 === 1) {
    window["e" + "val"](event.data);
  }
})
```



Abstract code representation

– conditions

– variable dependencies

– variable values



✓ AST



✓ control flow



✓ data flow

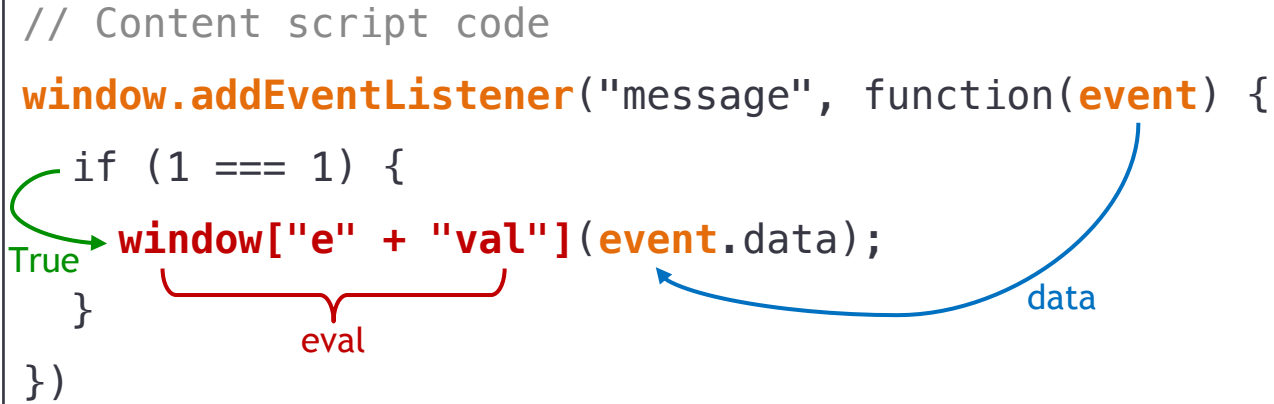


✓ pointer analysis



# Extension Dependence Graph

```
// Content script code
window.addEventListener("message", function(event) {
  if (1 === 1) {
    window["e" + "val"](event.data);
  }
})
```




- external messages
- internal messages

# Extension Dependence Graph

```
// Content script code
window.addEventListener("message", function(event) {
  if (1 === 1) {
    window["e" + "val"](event.data);
  }
})
```

Diagram annotations:  
- A purple devil emoji is placed above the parameter `event`.  
- A blue arrow labeled `data` points from `event` to `event.data`.  
- A red bracket labeled `eval` spans the expression `"e" + "val"`.  
- A green arrow labeled `True` points from the `if` condition `(1 === 1)` to the function call.

- external messages 
- internal messages

# Extension Dependence Graph

```
// Content script code  
chrome.runtime.sendMessage({toBP: mess});
```

```
// Background page code  
chrome.runtime.onMessage.addListener(function(request) {  
  
})
```

– external messages



– internal messages

# Extension Dependence Graph

```
// Content script code  
chrome.runtime.sendMessage({toBP: mess});
```

message

```
// Background page code  
chrome.runtime.onMessage.addListener(function(request) {  
})
```

– external messages



– internal messages





# Extension Dependence Graph

```
// Content script code  
chrome.runtime.sendMessage({toBP: mess});
```

message

```
// Background page code  
chrome.runtime.onMessage.addListener(function(request) {  
})
```

- external messages 
  - internal messages 
- Models message interaction within and outside of an extension

# Suspicious Data Flow Tracking

```
1 // Content script code
2 window.addEventListener("message", function(event) {
3     if (1 === 1) {
4         window["e" + "val"](event.data);
5     }
6 })
```

Diagram annotations:

- A green arrow labeled "True" points from the condition `(1 === 1)` on line 3 to the `eval` call on line 4.
- A red bracket underlines the string `"e" + "val"` in the `eval` call on line 4, with the label `eval` centered below it.

# Suspicious Data Flow Tracking

```
1 // Content script code
2 window.addEventListener("message", function(event) {
3     if (1 === 1) {
4         window["e" + "val"](event.data);
5     }
6 })
```



# Suspicious Data Flow Tracking

```
1 // Content script code
2 window.addEventListener("message", function(event) {
3     if (1 === 1) {
4         window["e" + "val"](event.data);
5     }
6 })
```

Diagram illustrating suspicious data flow tracking in the provided code:

- The variable `event` is highlighted in yellow and accompanied by a purple devil emoji.
- A blue arrow labeled `data` points from the `event` parameter to the `event.data` property access in the `eval` call.
- A red bracket labeled `eval` highlights the string concatenation `"e" + "val"` in the `eval` function call.
- A green arrow labeled `True` points to the `if (1 === 1)` condition, indicating it is always true.



# Suspicious Data Flow Tracking

```
1 // Content script code
2 window.addEventListener("message", function(event) {
3     if (1 === 1) {
4         window["e" + "val"](event.data);
5     }
6 })
```

Diagram annotations:  
- A purple devil emoji is placed above the `event` parameter in line 2.  
- A blue arrow labeled `data` points from the `event` parameter to the `event.data` property access in line 4.  
- A red bracket labeled `eval` spans the string `"e" + "val"` in line 4.  
- A green arrow labeled `True` points to the `if (1 === 1)` condition in line 3.



# Suspicious Data Flow Tracking

```
1 // Content script code
2 window.addEventListener("message", function(event) {
3   if (1 === 1) {
4     window["e" + "val"](event.data);
5   }
6 })
```

The diagram shows the code from the previous block with several annotations. A purple devil emoji is placed above the `event` parameter in line 2. A blue arrow labeled `data` points from `event` to `event.data` in line 4. A red bracket labeled `eval` spans the expression `["e" + "val"]` in line 4. A green arrow labeled `True` points to the `if (1 === 1)` condition in line 3.



```
// Data flow report
{"direct-danger1": "eval",
 "value": "eval(event.data)",
 "line": "4 - 4",
 "dataflow": true,
 "param1": {
   "received": "event",
   "line": "2 - 2"}}}
```

# Suspicious Data Flow Tracking

```
1 // Content script code
2 window.addEventListener("message", function(event) {
3   if (1 === 1) {
4     window["e" + "val"](event.data);
5   }
6   event = {"data": 42};
7   eval(event.data);
8 }
9 })
10
```

```
// Data flow report
{"direct-danger1": "eval",
 "value": "eval(event.data)",
 "line": "4 - 4",
 "dataflow": true,
 "param1": {
   "received": "event",
   "line": "2 - 2"}},

{"direct-danger2": "eval",
 "value": "eval(42)",
 "line": "8 - 8",
 "dataflow": false}
```

Flaw category	All components	High-privilege components
Code Execution	<code>eval, setInterval, setTimeout</code>	<code>tabs.executeScript</code>
Triggering Downloads		<code>downloads.download</code>
Cross-Origin Requests	<code>\$.ajax, jQuery.ajax, fetch, \$.get, jQuery.get, \$http.get, \$.post, \$http.post, XMLHttpRequest().open, jQuery.post, XMLHttpRequest.open</code>	
Data Exfiltration		<code>bookmarks.getTree, cookies.getAll, history.search, topSites.get</code>

# Large-Scale Analysis of Chrome Extensions

Attacker capabilities	#Exploitable
Code Execution	63
Triggering Downloads	21
Cross-Origin Requests	49
Data Exfiltration	76
Sum	209