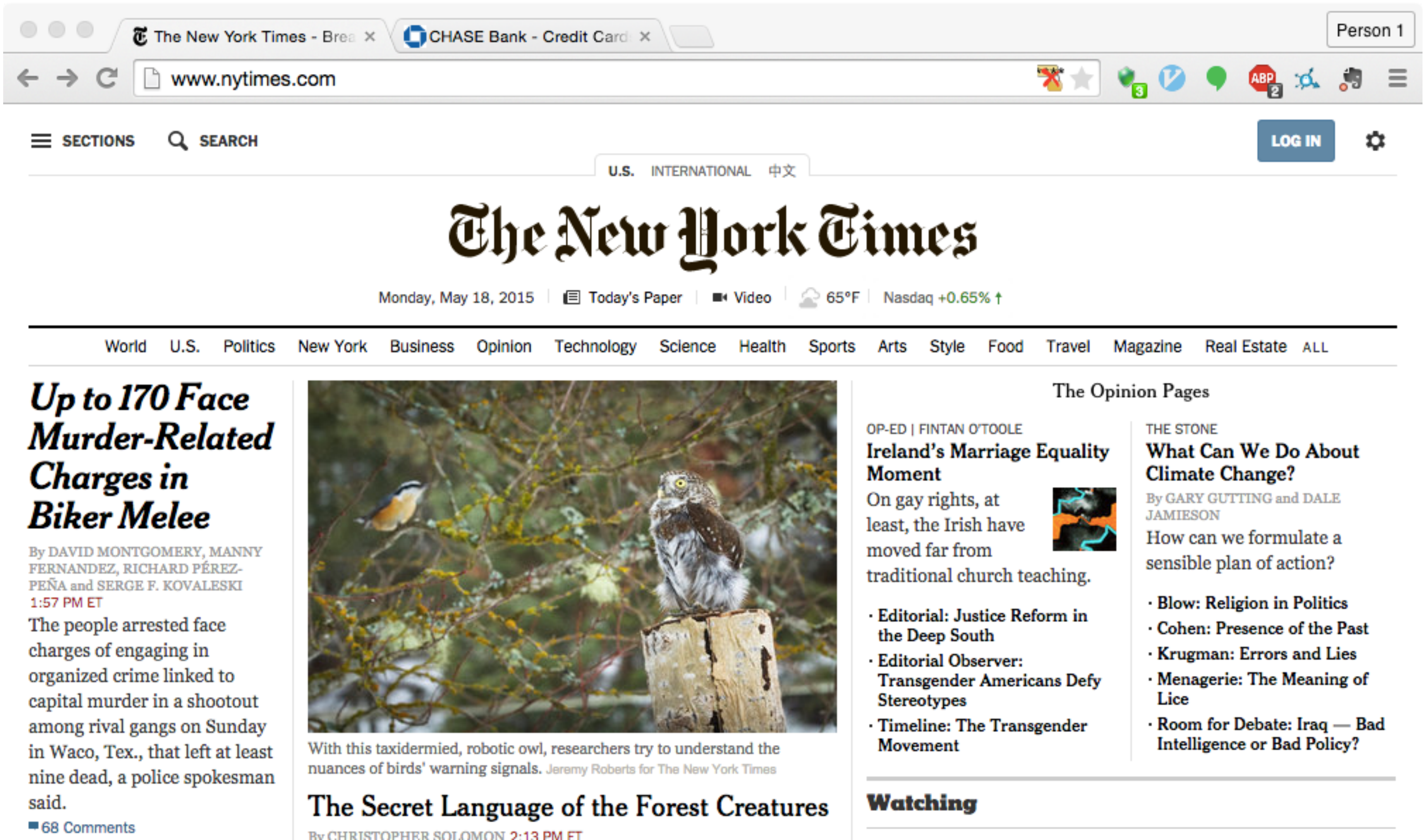
The Most Dangerous Code in the Browser

Stefan Heule, Devon Rifkin, Alejandro Russo, Deian Stefan

STANFORD UNIVERSITY

CHALMERS

Modern web experience



The New York Times - Brea ×
CHASE Bank - Credit Card ×
Person 1
www.nytimes.com

SECTIONS SEARCH LOG IN

U.S. INTERNATIONAL 中文

The New York Times

Monday, May 18, 2015 | Today's Paper | Video | 65°F | Nasdaq +0.65% ↑

World U.S. Politics New York Business Opinion Technology Science Health Sports Arts Style Food Travel Magazine Real Estate ALL

Up to 170 Face Murder-Related Charges in Biker Melee

By DAVID MONTGOMERY, MANNY FERNANDEZ, RICHARD PÉREZ-PEÑA and SERGE F. KOVALESKI
1:57 PM ET

The people arrested face charges of engaging in organized crime linked to capital murder in a shootout among rival gangs on Sunday in Waco, Tex., that left at least nine dead, a police spokesman said.

68 Comments

With this taxidermied, robotic owl, researchers try to understand the nuances of birds' warning signals. Jeremy Roberts for The New York Times

The Secret Language of the Forest Creatures

By CHRISTOPHER SOLOMON 2:13 PM ET

The Opinion Pages

OP-ED | FINTAN O'TOOLE

Ireland's Marriage Equality Moment

On gay rights, at least, the Irish have moved far from traditional church teaching.

- **Editorial: Justice Reform in the Deep South**
- **Editorial Observer: Transgender Americans Defy Stereotypes**
- **Timeline: The Transgender Movement**

THE STONE

What Can We Do About Climate Change?

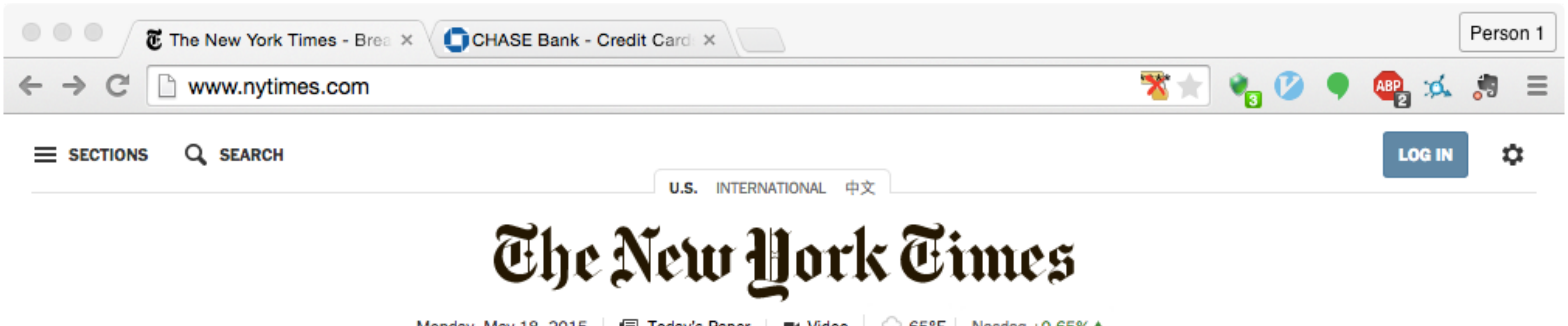
By GARY GUTTING and DALE JAMIESON

How can we formulate a sensible plan of action?

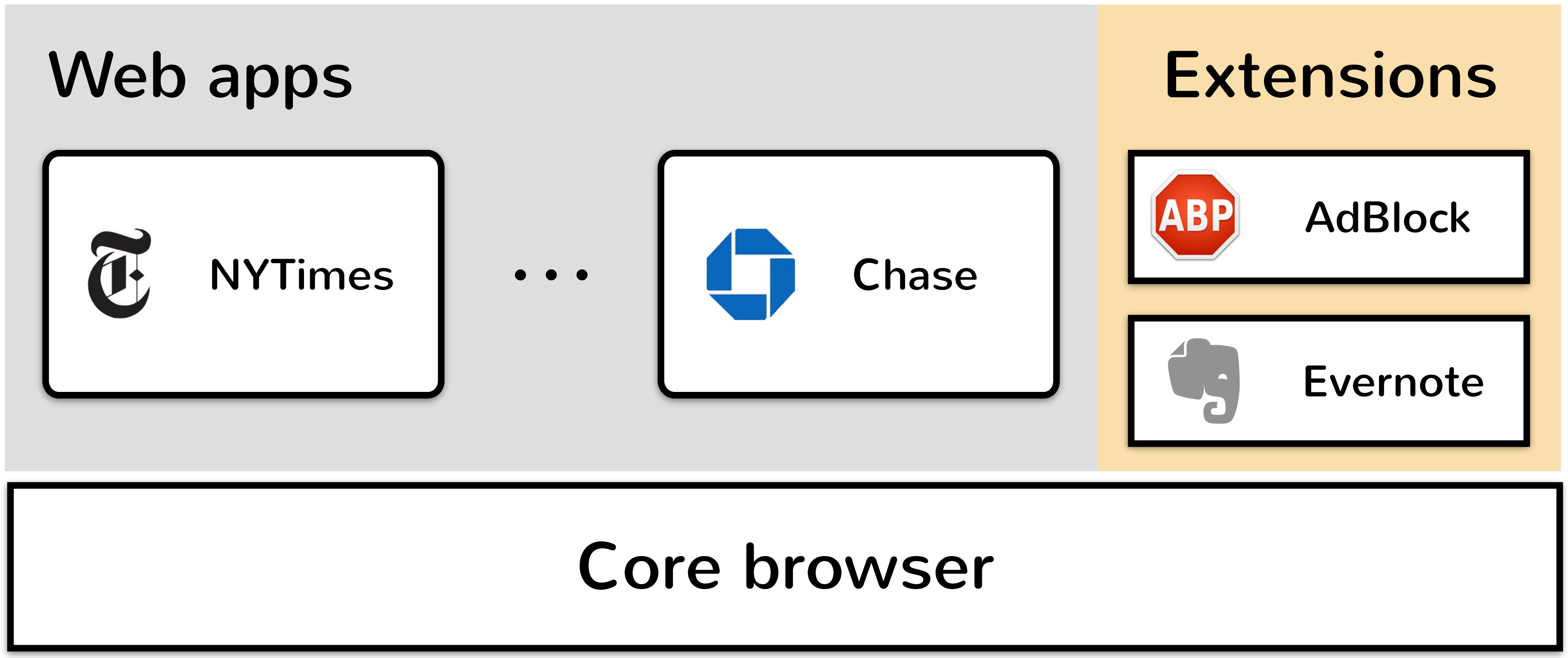
- **Blow: Religion in Politics**
- **Cohen: Presence of the Past**
- **Krugman: Errors and Lies**
- **Menagerie: The Meaning of Lice**
- **Room for Debate: Iraq — Bad Intelligence or Bad Policy?**

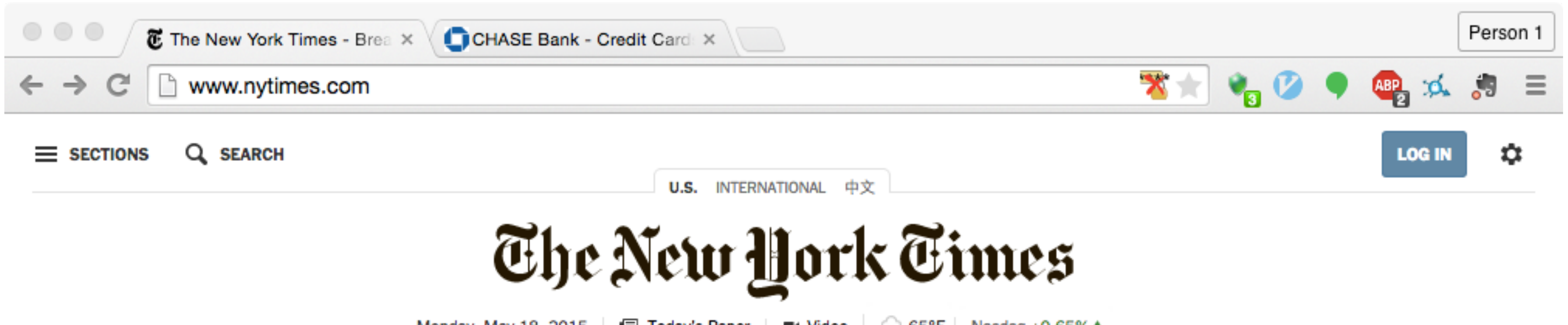
Watching

Modern web experience



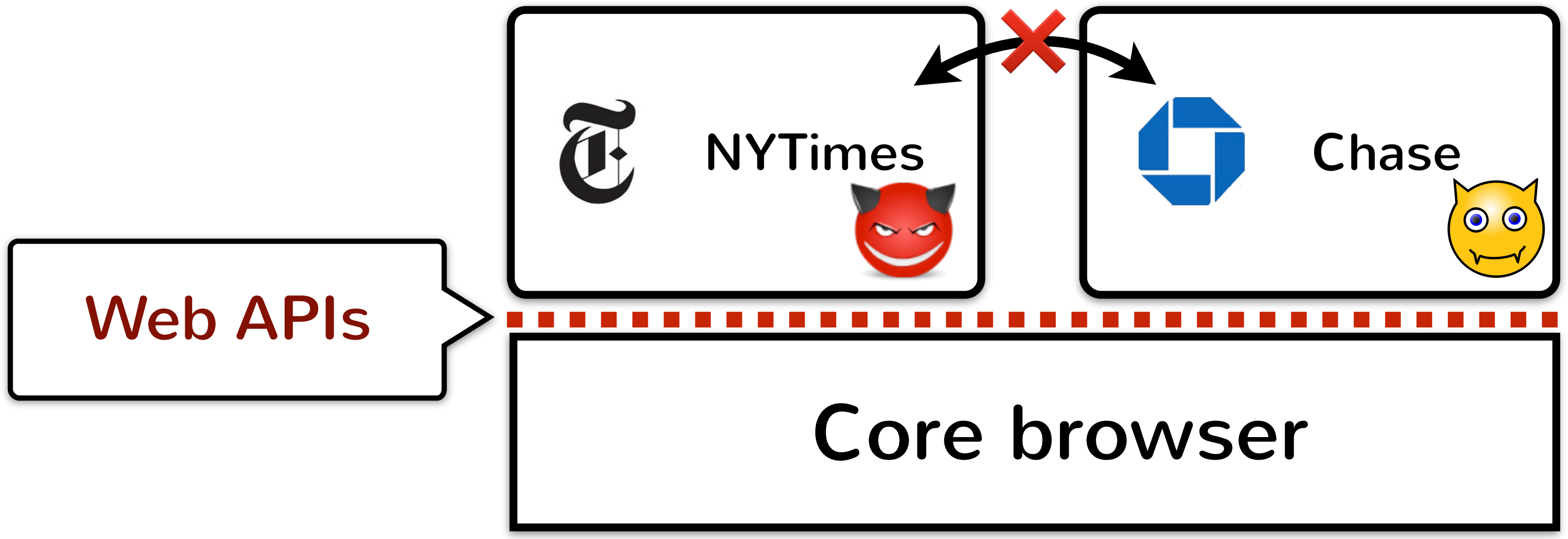
Modern web experience





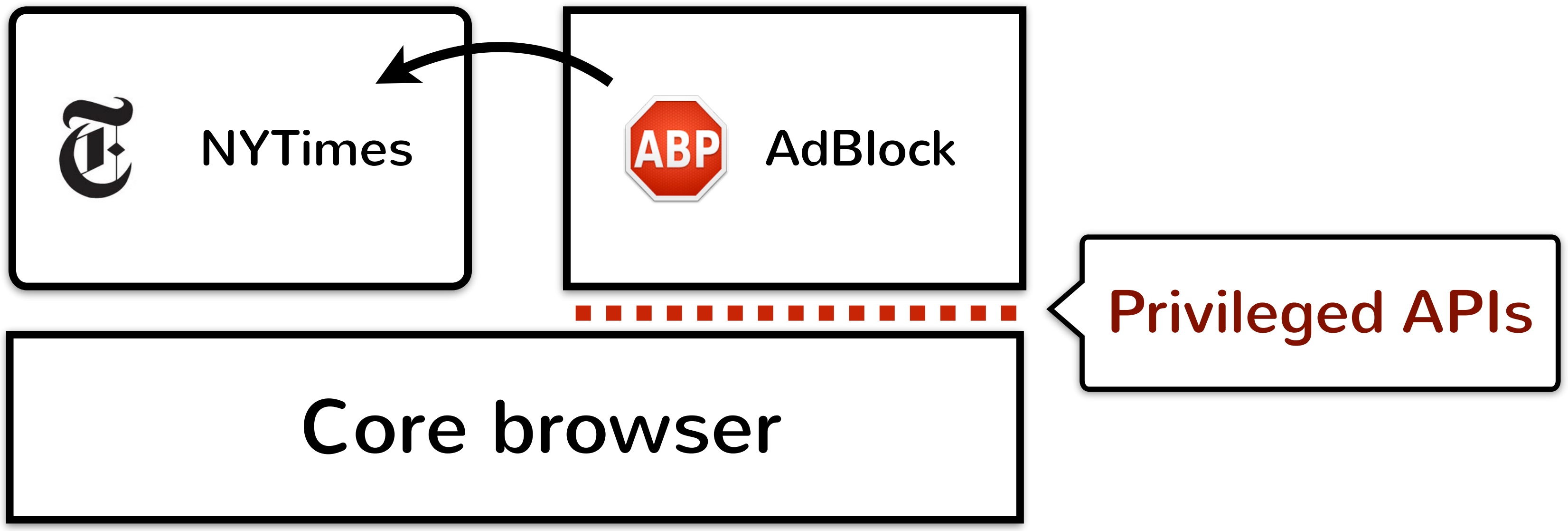
Web app security

- **Trust model:** malicious code



- Apps are isolated according to same-origin policy
- Apps are constrained to Web APIs (e.g., DOM)
  - ➤ They cannot access arbitrary files, devices, etc.

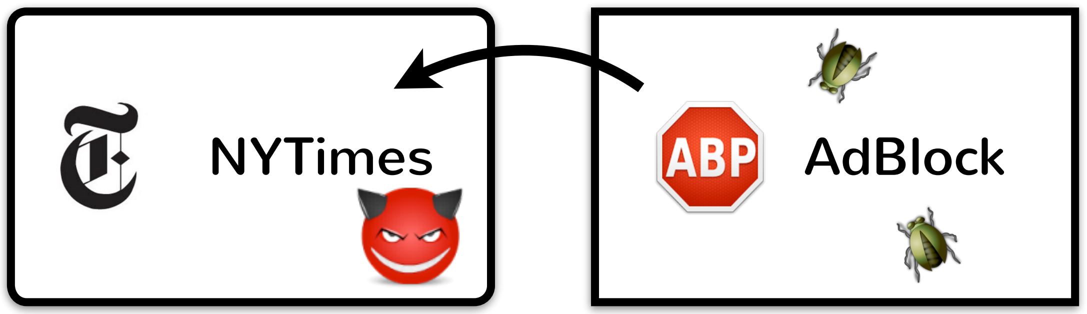
Extension security?



- **Extensions need direct access to app DOMs**
  - ➤ They modify app style, content, behavior, …
- **Extensions need privileged APIs**
  - ➤ To fetch/store cross-origin content, to read/modify history and bookmarks, to create new tabs, etc.

Chrome extension security model

- Trust model: extensions are benign-but-buggy



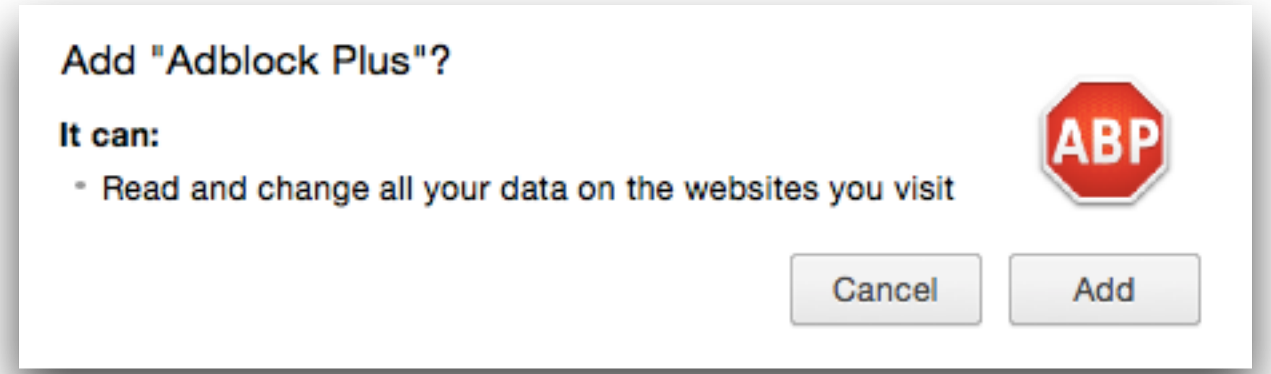
- Privilege separate extension: core and content
  - ➤ Protects vulnerable extension from malicious apps
- Run extensions with least privilege
  - ➤ Limits damage due to exploits

Least privilege via permission system

- Extensions declare necessary permissions

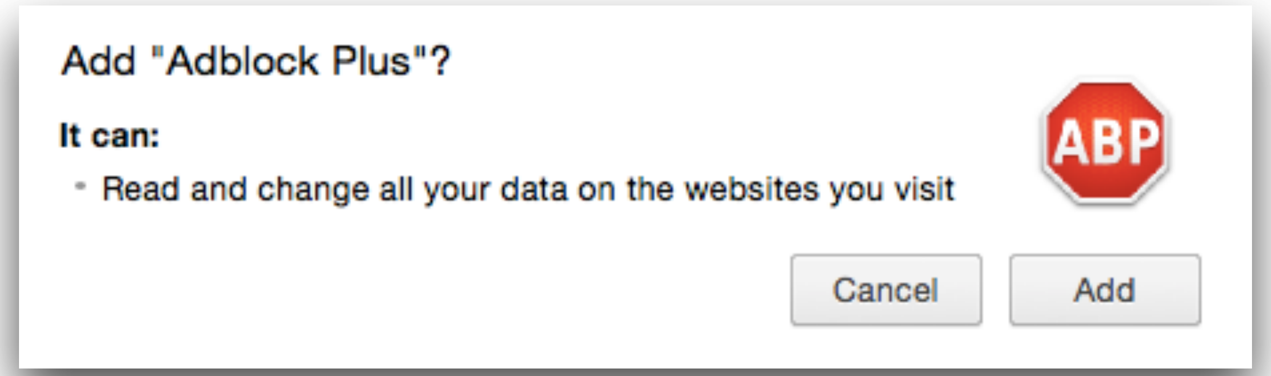
```
{
  "name": “AdBlock Plus”,
  "version": "2.1.10",
...
  "permissions": [
    "http://*/*", "https://*/*", "contextMenus"
  ],
...
```

- Users must grant permissions at install time


Add "Adblock Plus"?

It can:

- Read and change all your data on the websites you visit

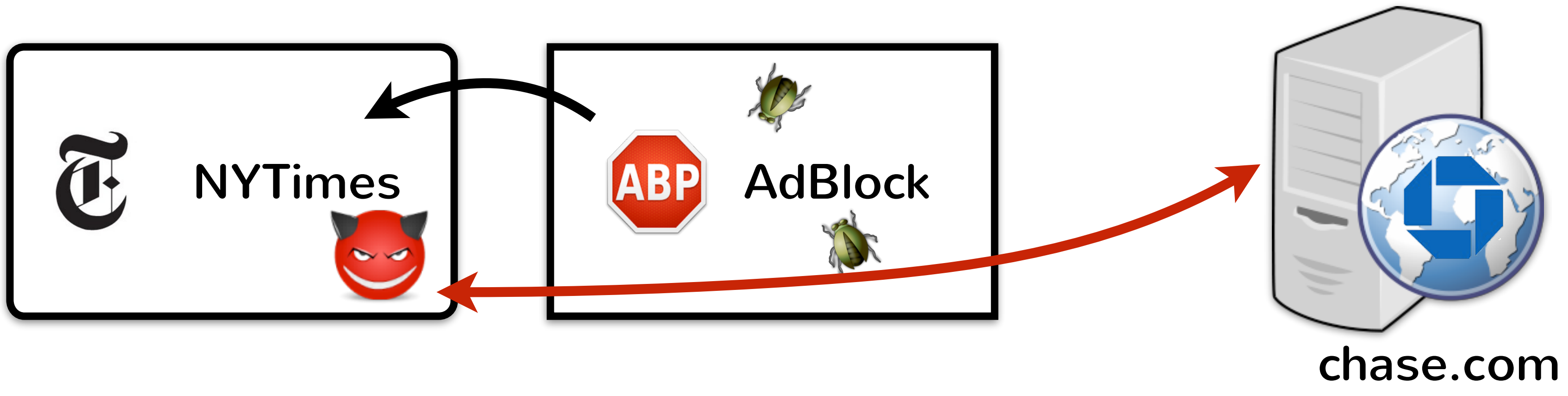
What does  mean?

Add "Adblock Plus"?

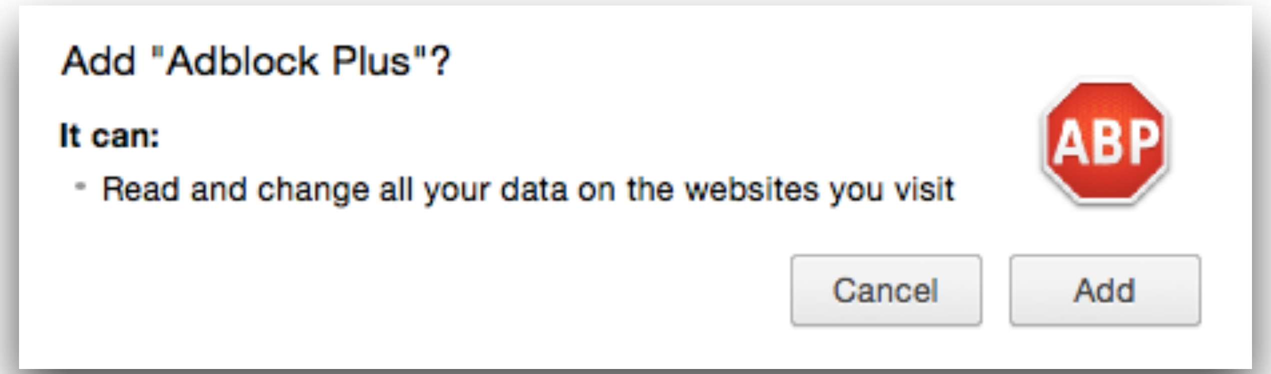
It can:

- Read and change all your data on the websites you visit

- Can read and modify data on any site, regardless of what site you are visiting



- AdBlock must be a special case, right?
    - ➤ 71.6% of top 500 extensions need this privilege!

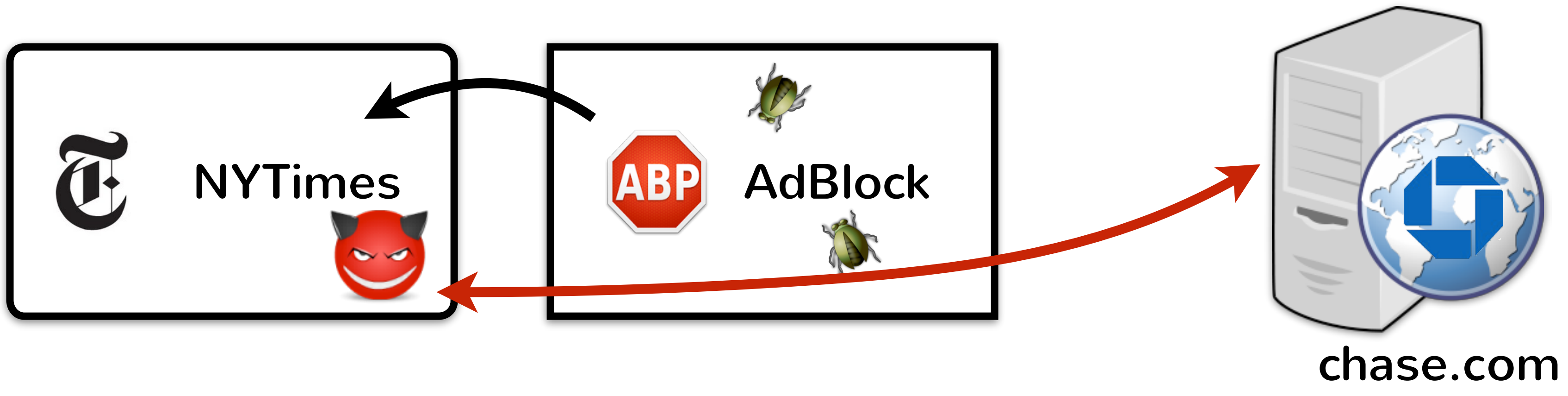
What does  mean?

Add "Adblock Plus"?

It can:

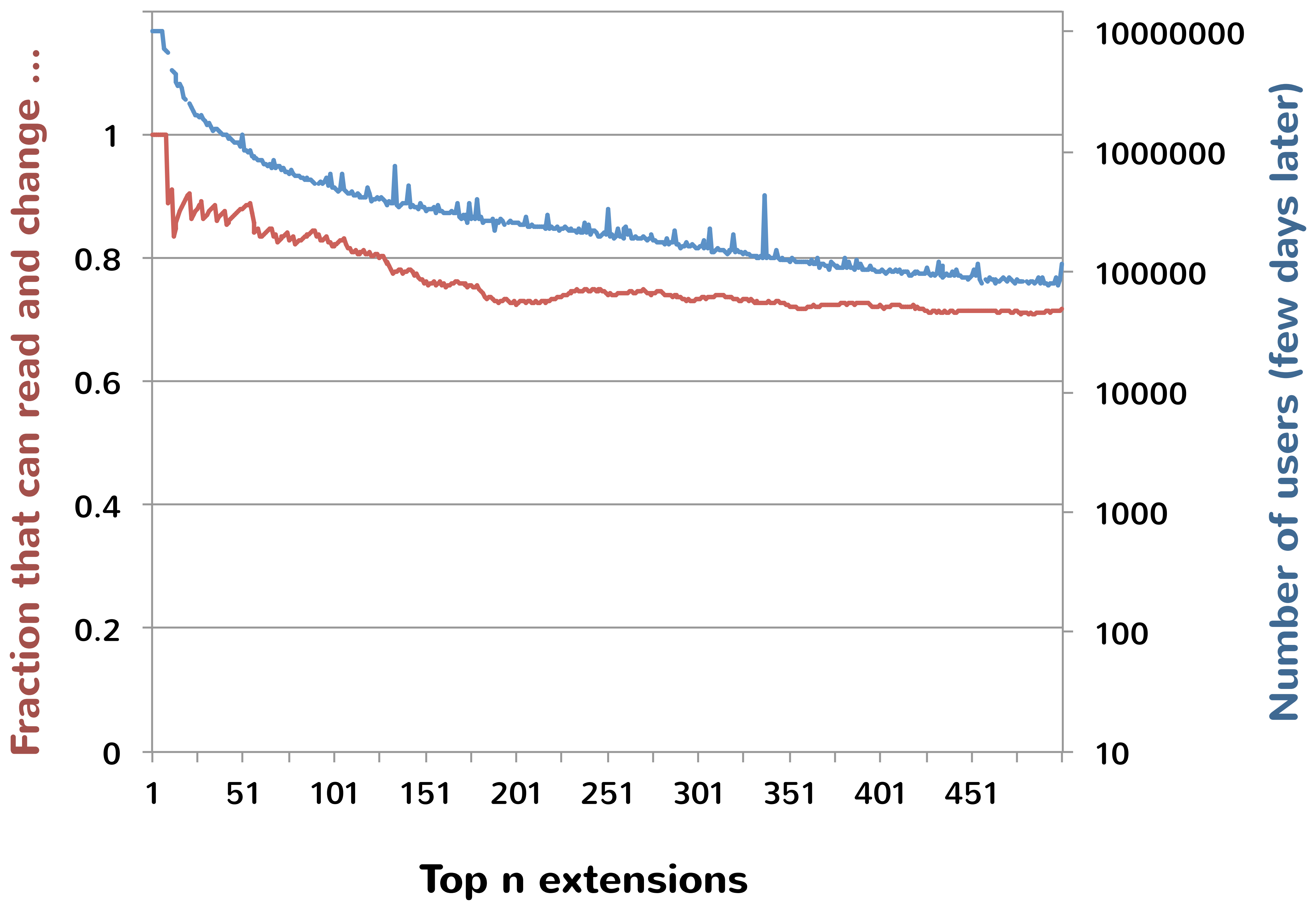
- Read and change all your data on the websites you visit

- Can read and modify data on any site, regardless of what site you are visiting

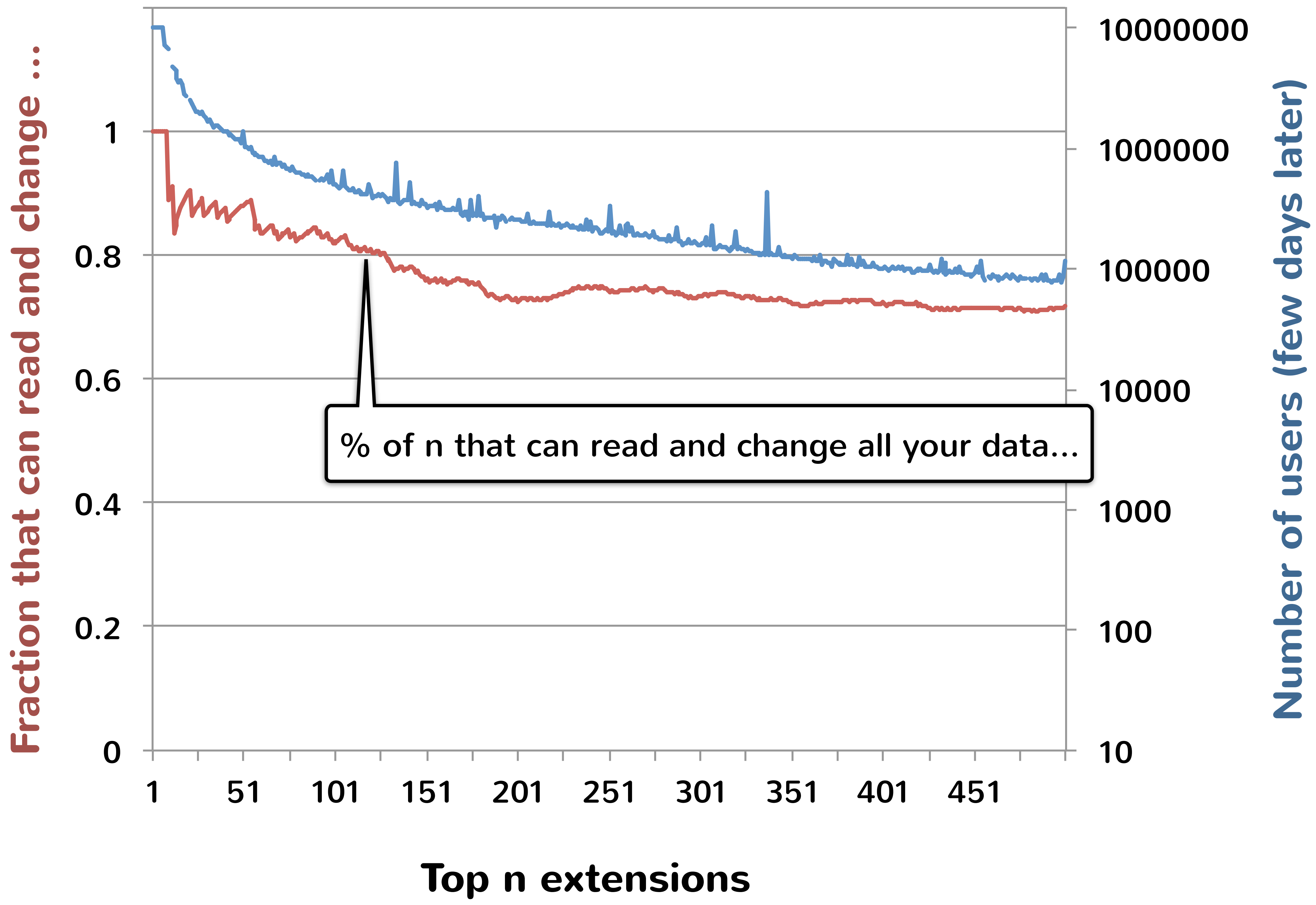


- AdBlock must be a special case, right?
  - 71.6% of top 500 extensions need this privilege!

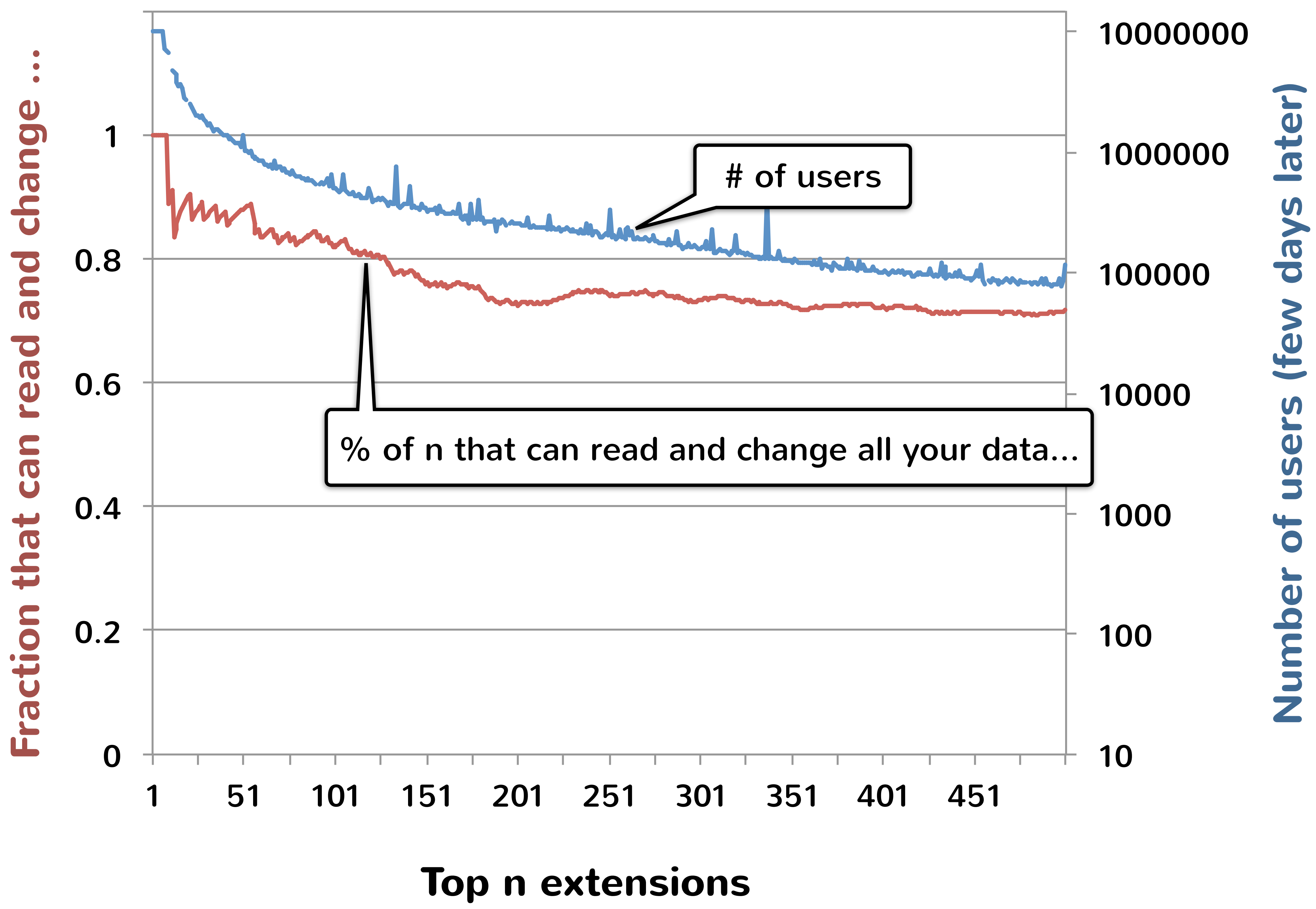
It gets worse with popularity



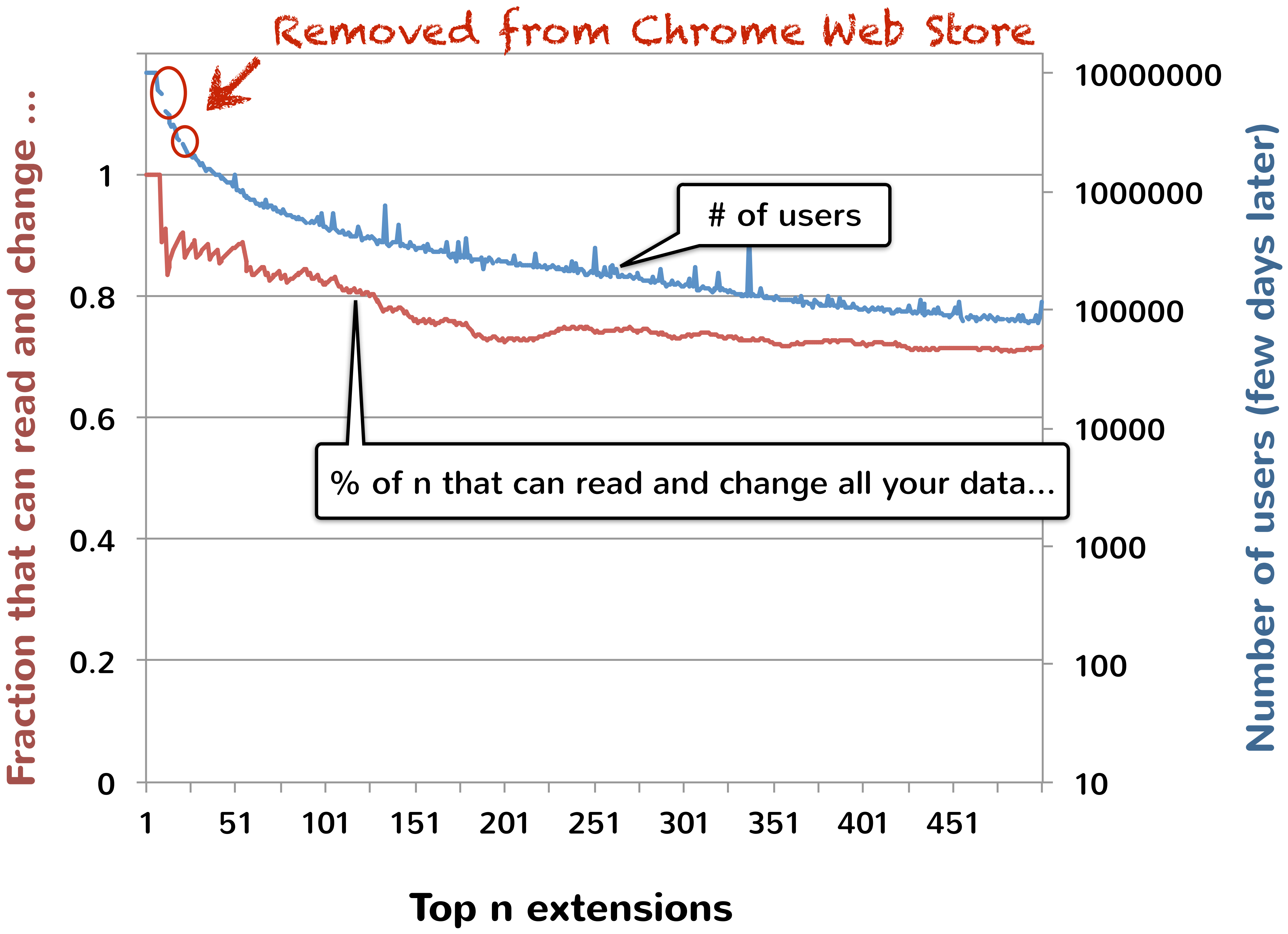
It gets worse with popularity



It gets worse with popularity



It gets worse with popularity



Problem with Chrome’s model

- **Permission requests are meaningless**
  - ➤ Descriptions are broad and context-independent
- **Model encourages principle of most privilege**
  - ➤ Extensions don’t auto-update if they need more privs
- **Threat model is not realistic**
  - ➤ Chrome Web Store listed many malicious extensions
  - ➤ Roughly 5% of Google users run malicious extensions

Problem with Chrome’s model

- **Permission requests are meaningless**
  - ➤ Descriptions are broad and context-independent
- **Model encourages principle of most privilege**
  - ➤ Extensions don’t auto-update if they need more privs
- **Threat model is not realistic**
  - ➤ Chrome Web Store listed many malicious extensions
  - ➤ Roughly 5% of Google users run malicious extensions

Problem with Chrome's model

- **Permission requests are meaningless**
  - ➤ Descriptions are broad and context-independent
- **Model encourages principle of most privilege**
  - ➤ Extensions don't auto-update if they need more privs
- **Threat model is not realistic**
  - ➤ Chrome Web Store listed many malicious extensions
  - ➤ Roughly 5% of Google users run malicious extensions

New extension-system goals

- **Meaningful permission system**
  - Safe behavior should not require permission
  - Permissions requests should be content-specific
- **Model should encourage least privilege**
  - Permissions should be fine-grained
  - Incentivize safe extensions
- **Threat model: extensions may be malicious**
  - Need to also protect user app data from extensions

New extension-system goals

- **Meaningful permission system**
  - Safe behavior should not require permission
  - Permissions requests should be content-specific
- **Model should encourage least privilege**
  - Permissions should be fine-grained
  - Incentivize safe extensions
- **Threat model: extensions may be malicious**
  - Need to also protect user app data from extensions

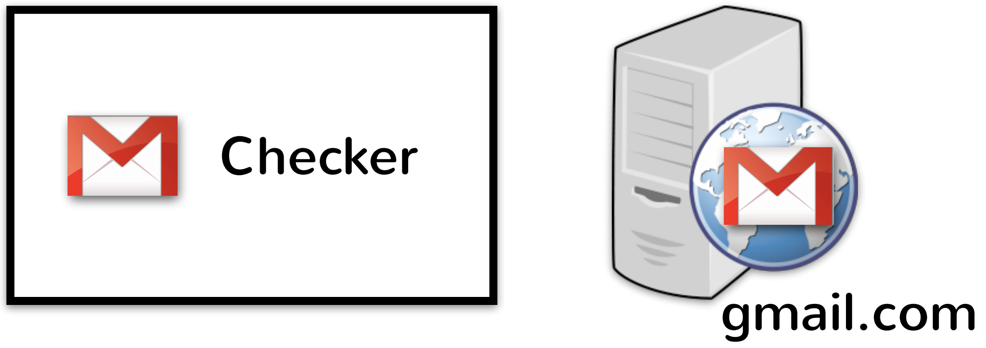
New extension-system goals

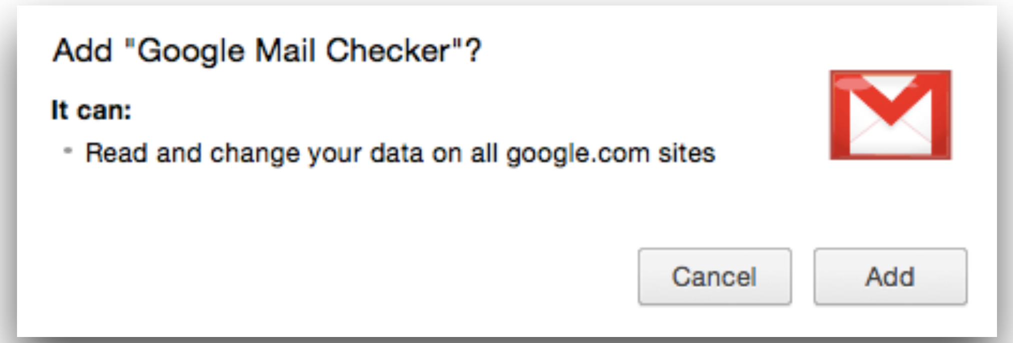
- **Meaningful permission system**
  - ➢ Safe behavior should not require permission
  - ➢ Permissions requests should be content-specific
- **Model should encourage least privilege**
  - ➢ Permissions should be fine-grained
  - ➢ Incentivize safe extensions
- **Threat model: extensions may be malicious**
  - ➢ Need to also protect user app data from extensions

How can we do this?

Insight: it is safe for extension to read user data if it can't arbitrarily disseminate it

- E.g., Google Mail Checker





Add "Google Mail Checker"?

It can:

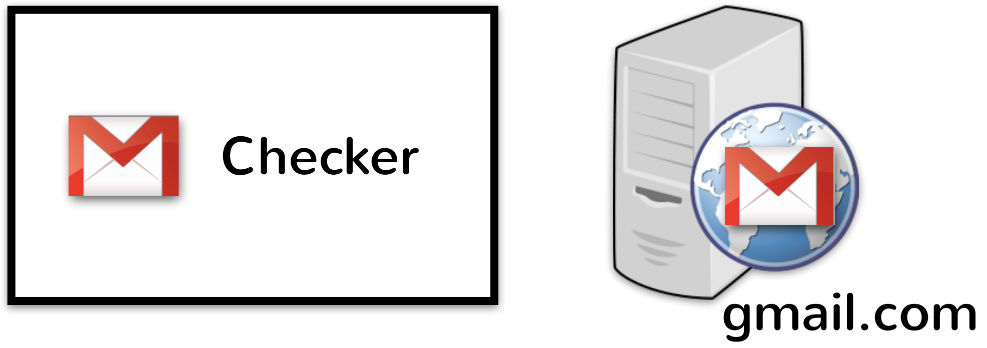
- Read and change your data on all google.com sites

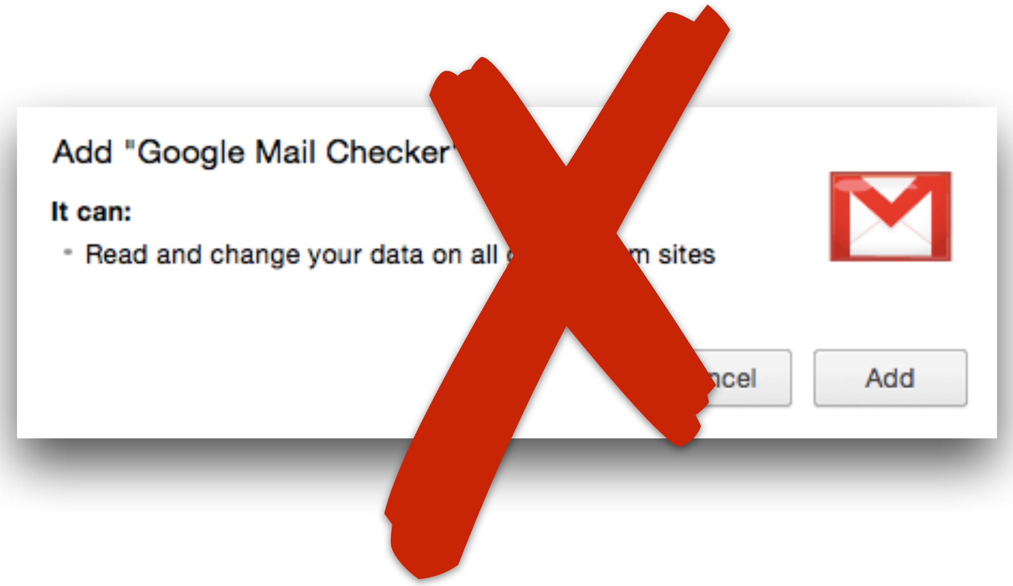
- Taint extensions according to what it reads
- Confine code to protect user's privacy

How can we do this?

Insight: it is safe for extension to read user data if it can't arbitrarily disseminate it

- E.g., Google Mail Checker



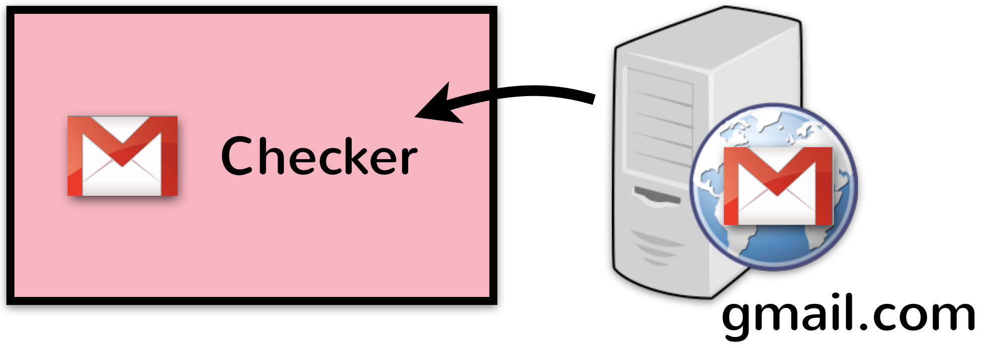

Add "Google Mail Checker"
It can:
Read and change your data on all … sites
Cancel
Add

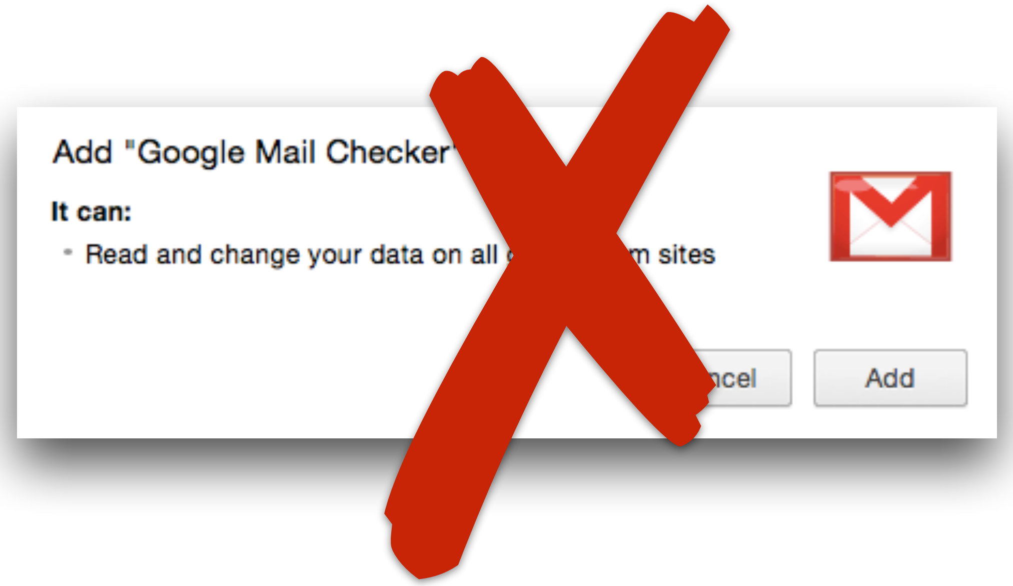
- Taint extensions according to what it reads
- Confine code to protect user's privacy

How can we do this?

Insight: it is safe for extension to read user data if it can't arbitrarily disseminate it

- E.g., Google Mail Checker




Add "Google Mail Checker"

It can:

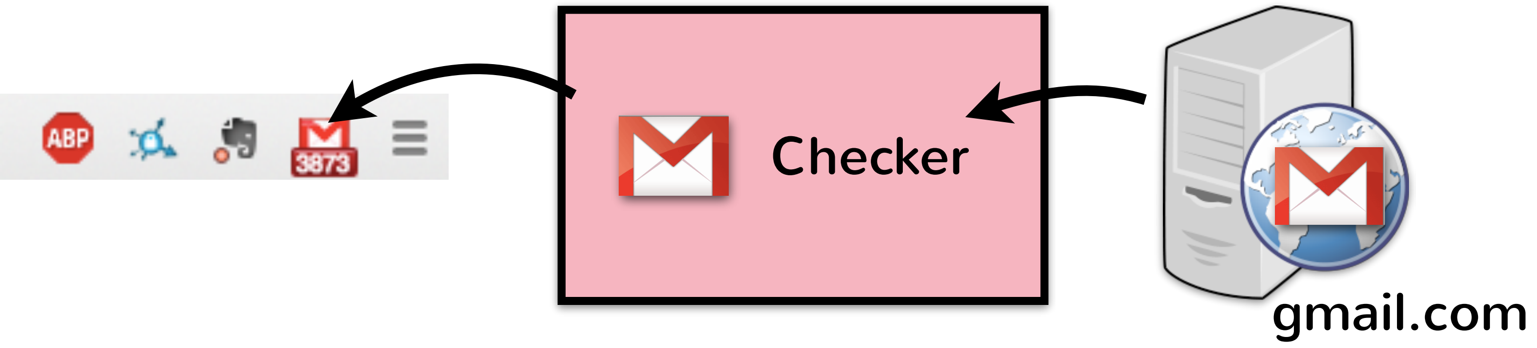
- Read and change your data on all ... m sites

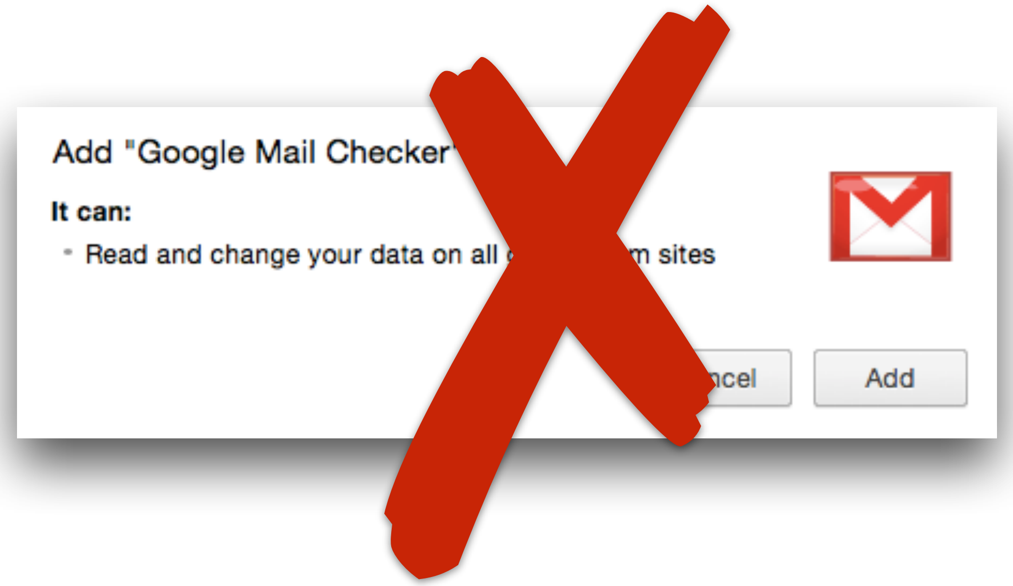
- Taint extensions according to what it reads
- Confine code to protect user's privacy

How can we do this?

Insight: it is safe for extension to read user data if it can't arbitrarily disseminate it

- E.g., Google Mail Checker





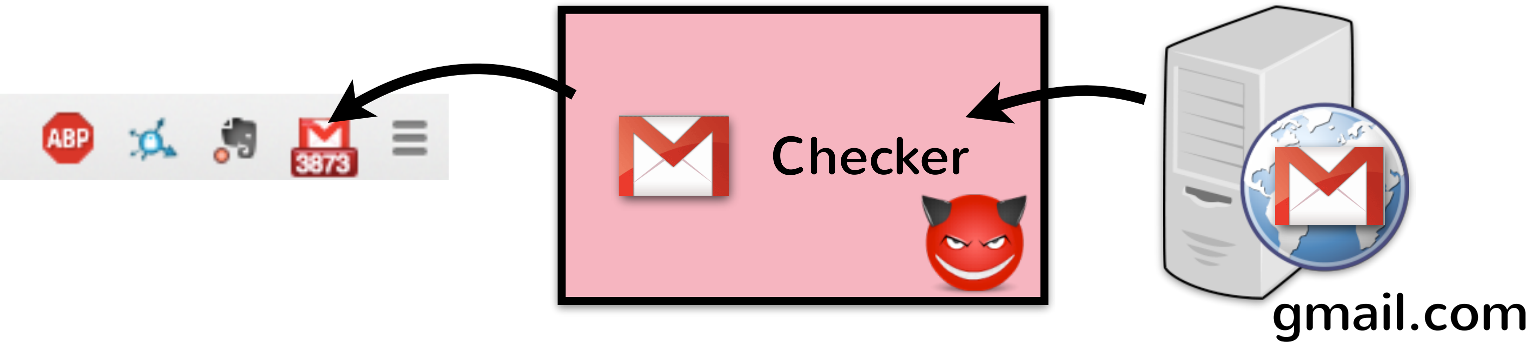
Add "Google Mail Checker"
It can:
Read and change your data on all m sites
ncel
Add

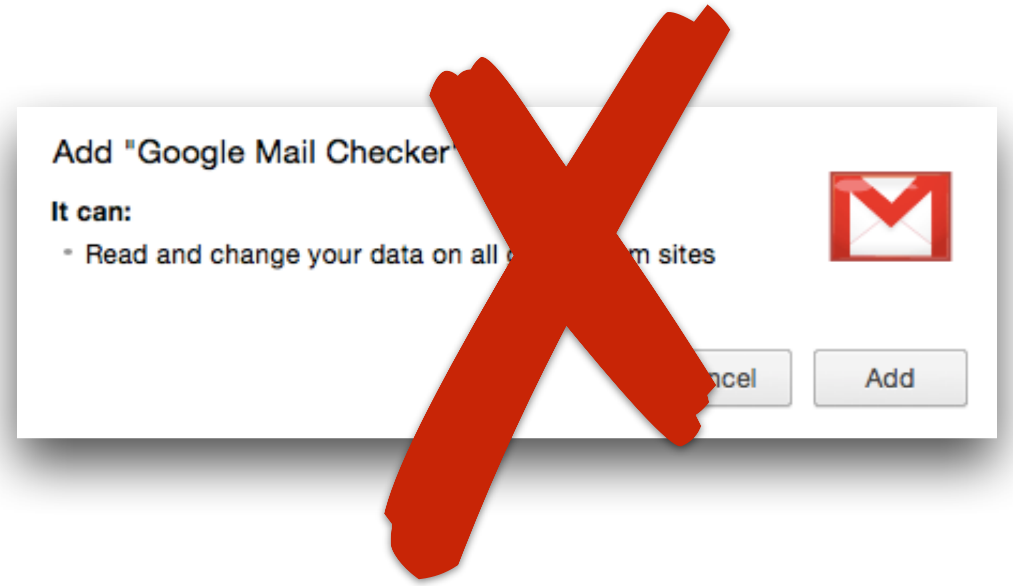
- Taint extensions according to what it reads
- Confine code to protect user's privacy

How can we do this?

Insight: it is safe for extension to read user data if it can't arbitrarily disseminate it

- E.g., Google Mail Checker





Add "Google Mail Checker"

It can:

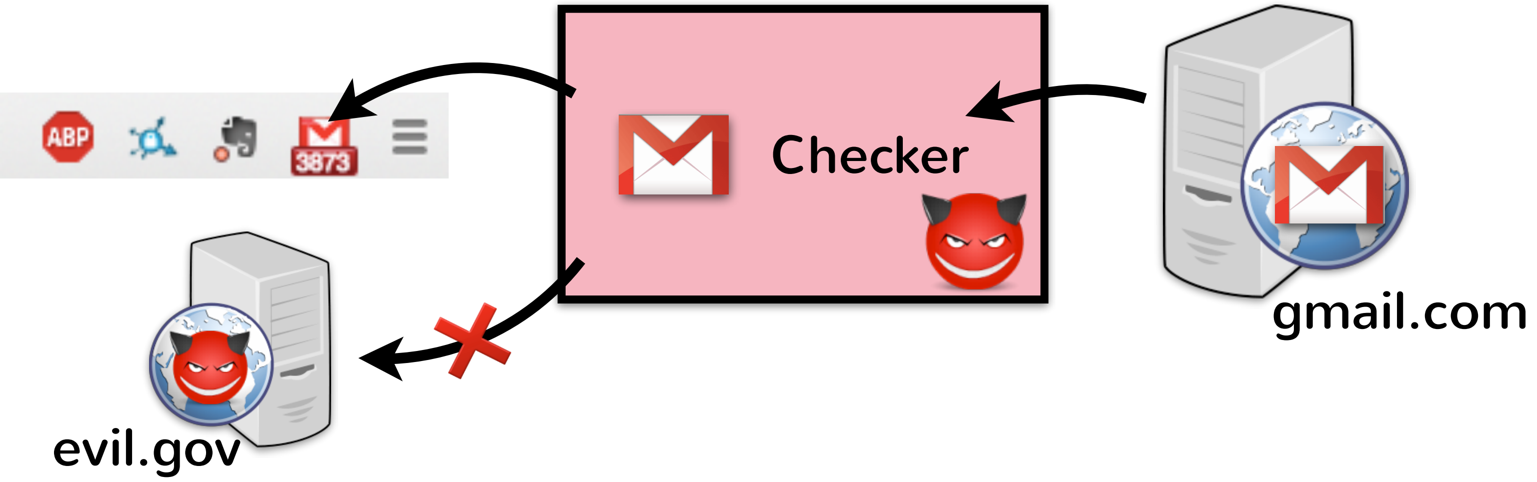
- Read and change your data on all ... sites

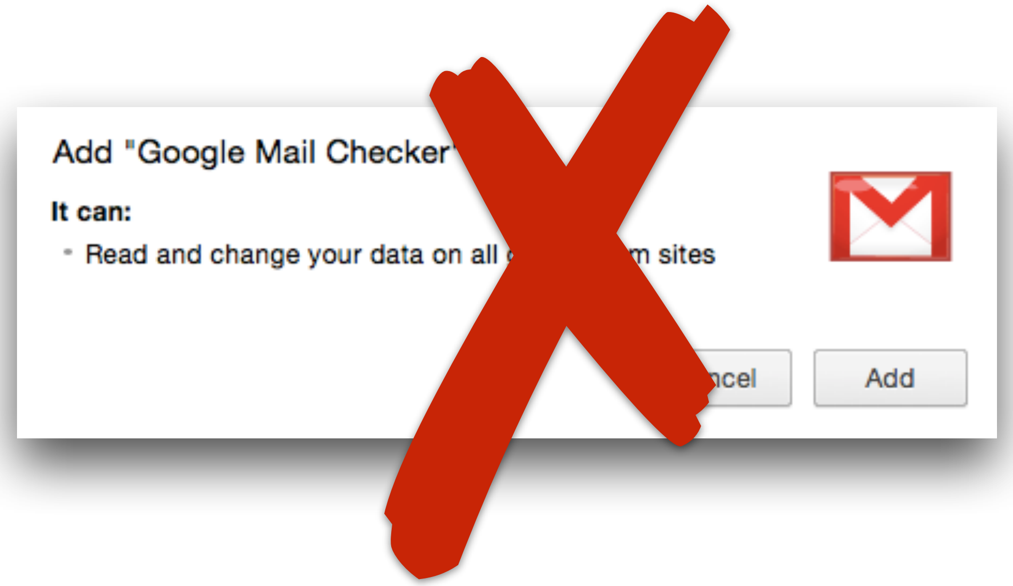
- Taint extensions according to what it reads
- Confine code to protect user's privacy

How can we do this?

Insight: **it is safe for extension to read user data if it can't arbitrarily disseminate it**

- E.g., Google Mail Checker



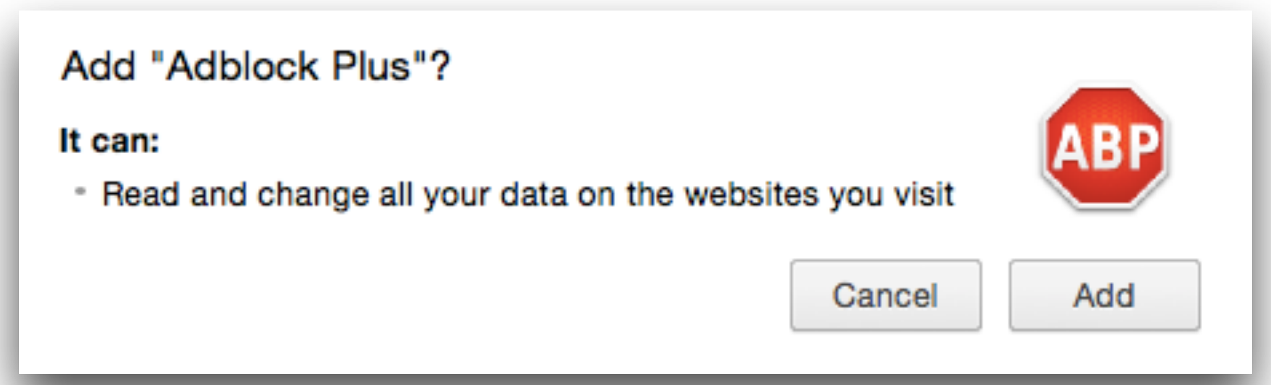

Add "Google Mail Checker"

It can:

- Read and change your data on all m sites

- Taint extensions according to what it reads
- Confine code to protect user's privacy

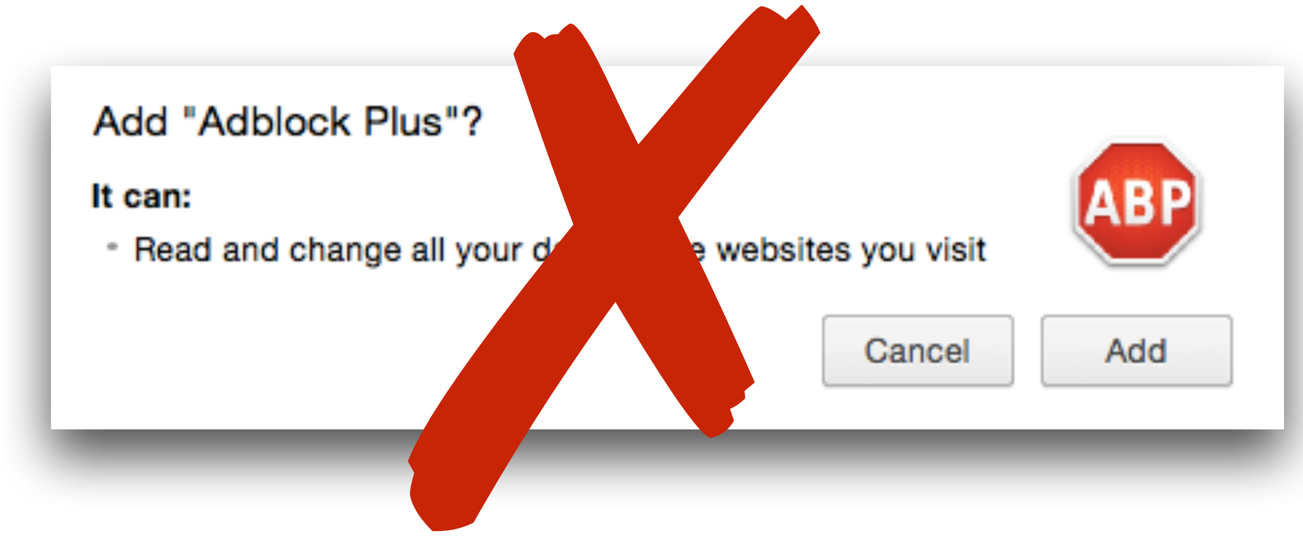
Safely read and modify pages?


Add "Adblock Plus"?

It can:

- Read and change all your data on the websites you visit

Safely read and modify pages?

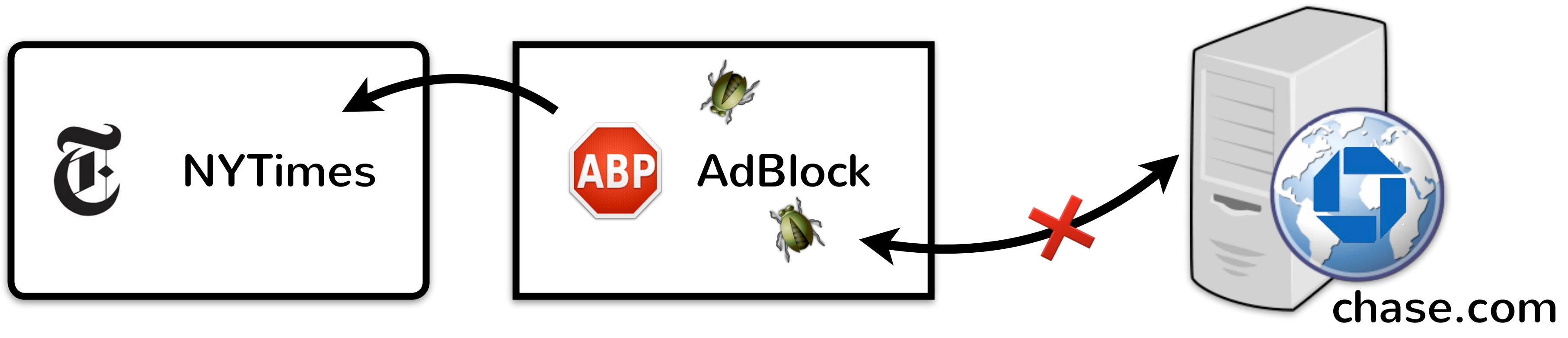

Add "Adblock Plus"?

It can:

- Read and change all your d e websites you visit

Safely read and modify pages?

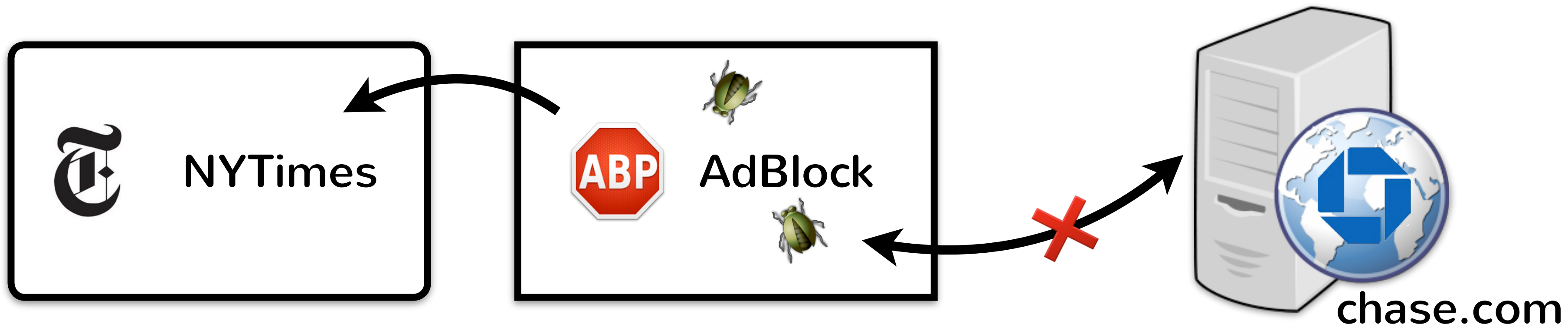
- Idea: tie extension script with app page
  - ➤ Impose at least same-origin policy on extension



- Challenge: read data from page and leak it by injecting content into page’s DOM
- Solution: taint extension, write to isolated DOM
  - ➤ Loads due to extension restricted: confined!

Safely read and modify pages?

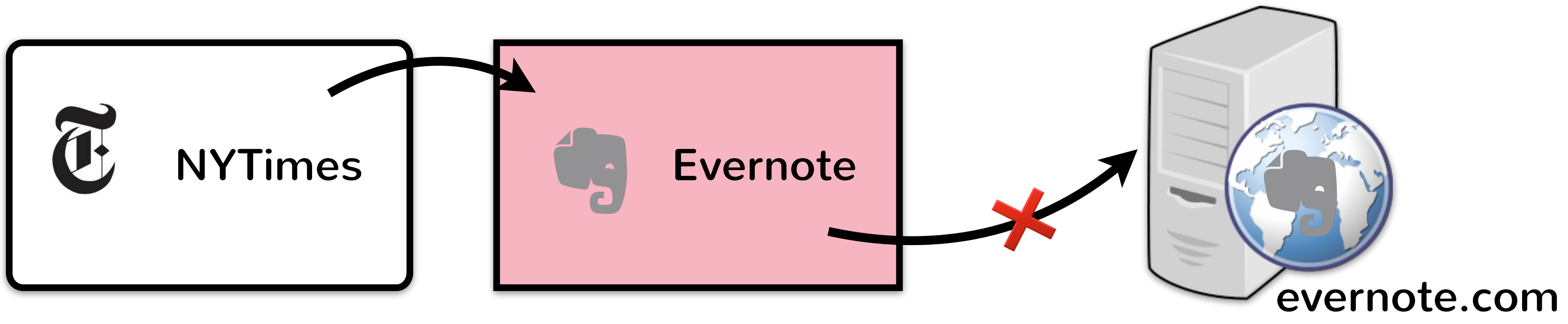
- **Idea:** tie extension script with app page
  - ➤ Impose at least same-origin policy on extension



- **Challenge:** read data from page and leak it by injecting content into page's DOM
- **Solution:** taint extension, write to isolated DOM
  - ➤ Loads due to extension restricted: confined!

Confinement: safe, too restricting

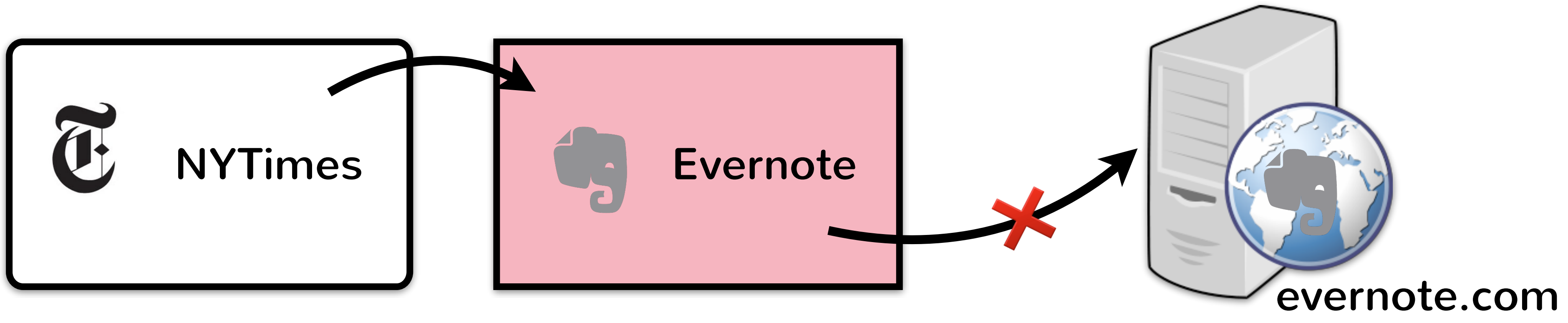
- **Challenge: extensions need to “leak” data**
  - E.g., Evernote is used to save URL, page, etc.
  - Reading DOM taints extension:



- Solution: declassification via sharing menu API

Confinement: safe, too restricting

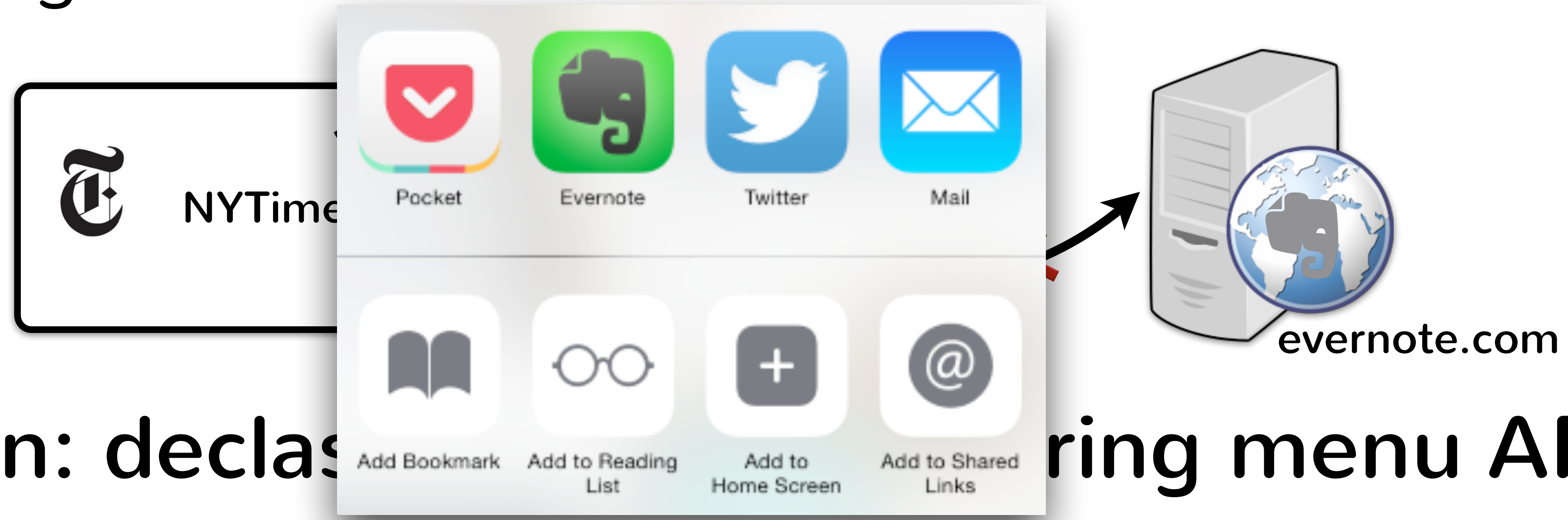
- **Challenge: extensions need to "leak" data**
  - E.g., Evernote is used to save URL, page, etc.
  - Reading DOM taints extension:



- **Solution: declassification via sharing menu API**

Confinement: safe, too restricting

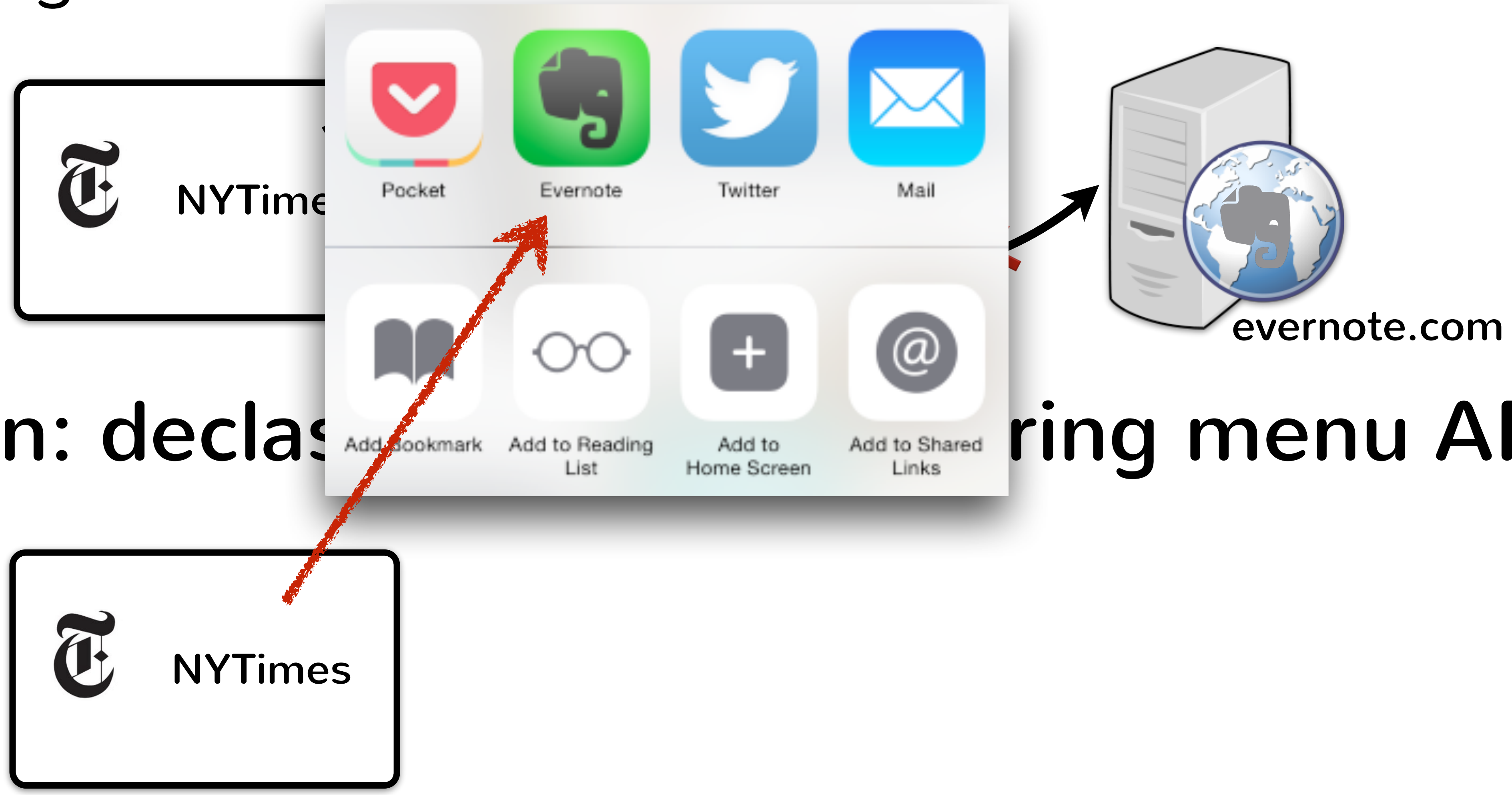
- Challenge: extensions need to "leak" data
  - E.g., Evernote is used to save URL, page, etc.
  - Reading DOM taints extension:



- Solution: declas ring menu API

Confinement: safe, too restricting

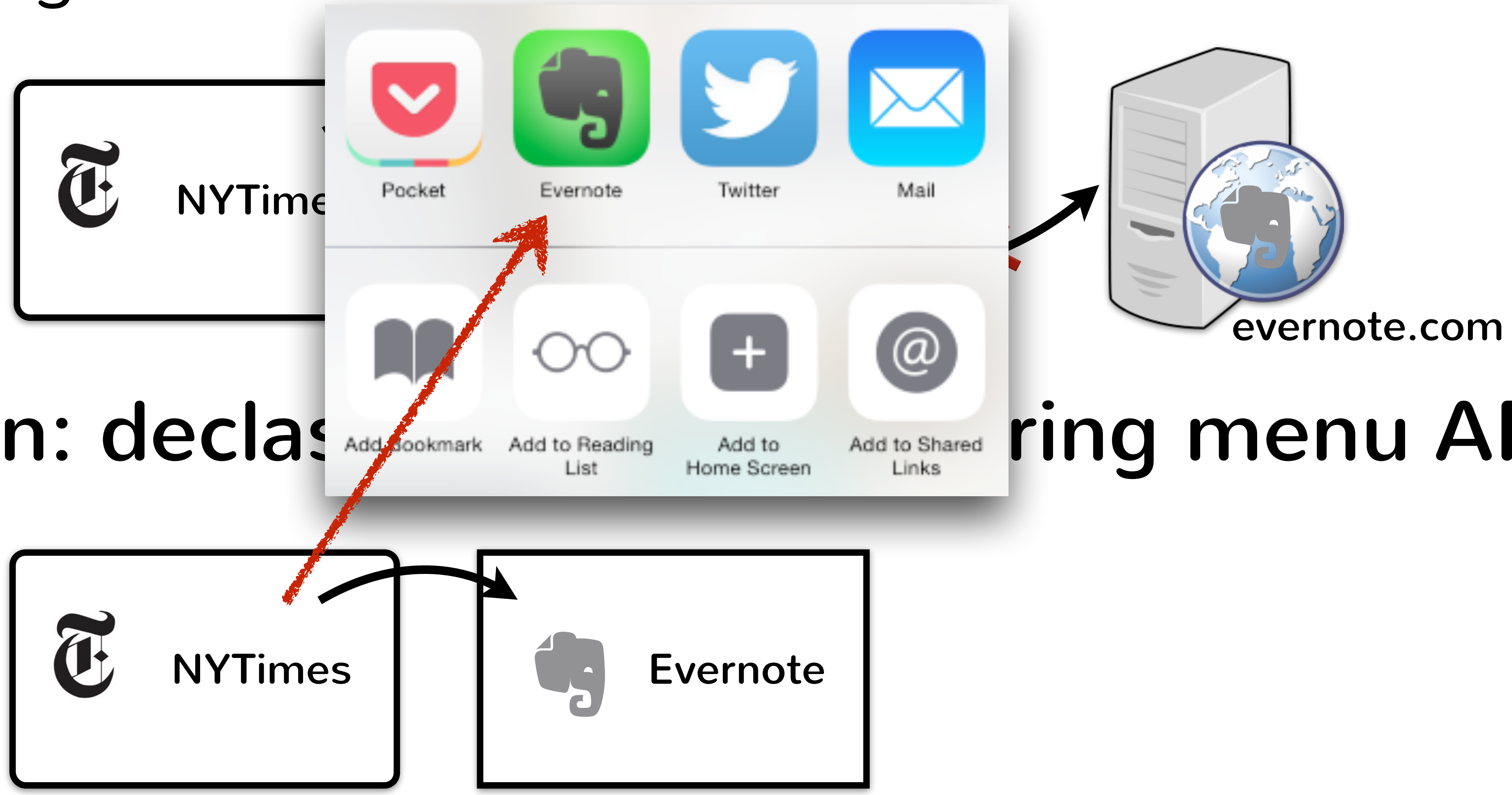
- **Challenge: extensions need to "leak" data**
  - E.g., Evernote is used to save URL, page, etc.
  - Reading DOM taints extension:



- **Solution: declas**[illegible]**ring menu API**

Confinement: safe, too restricting

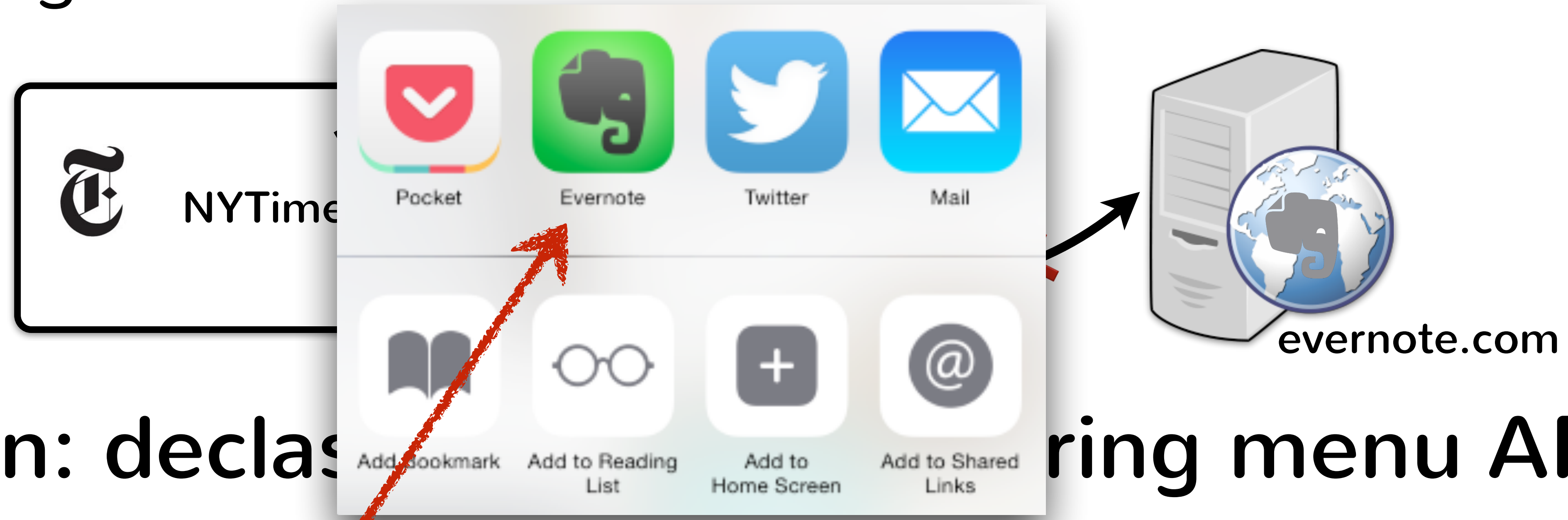
- **Challenge: extensions need to "leak" data**
  - E.g., Evernote is used to save URL, page, etc.
  - Reading DOM taints extension:



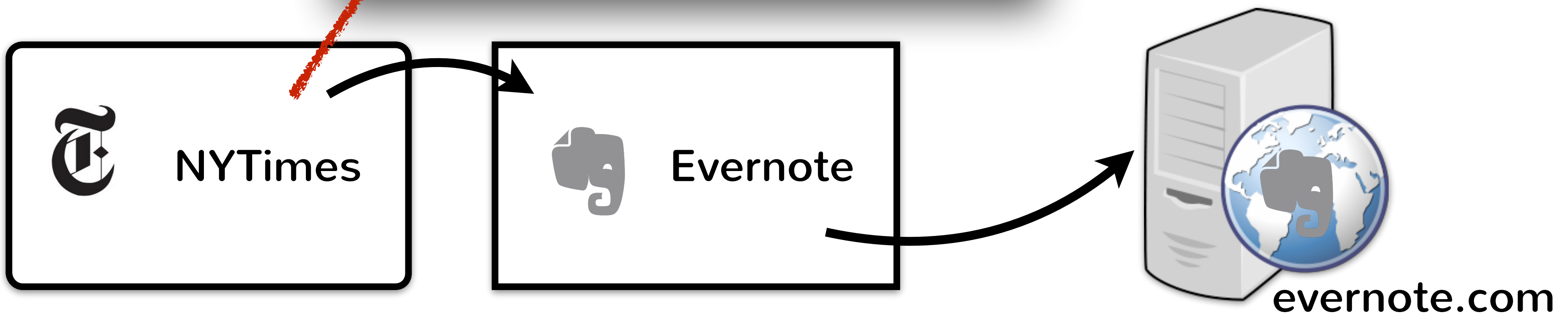
- **Solution: declas ring menu API**

Confinement: safe, too restricting

- Challenge: extensions need to “leak” data
  - E.g., Evernote is used to save URL, page, etc.
  - Reading DOM taints extension:



- Solution: declas ring menu API

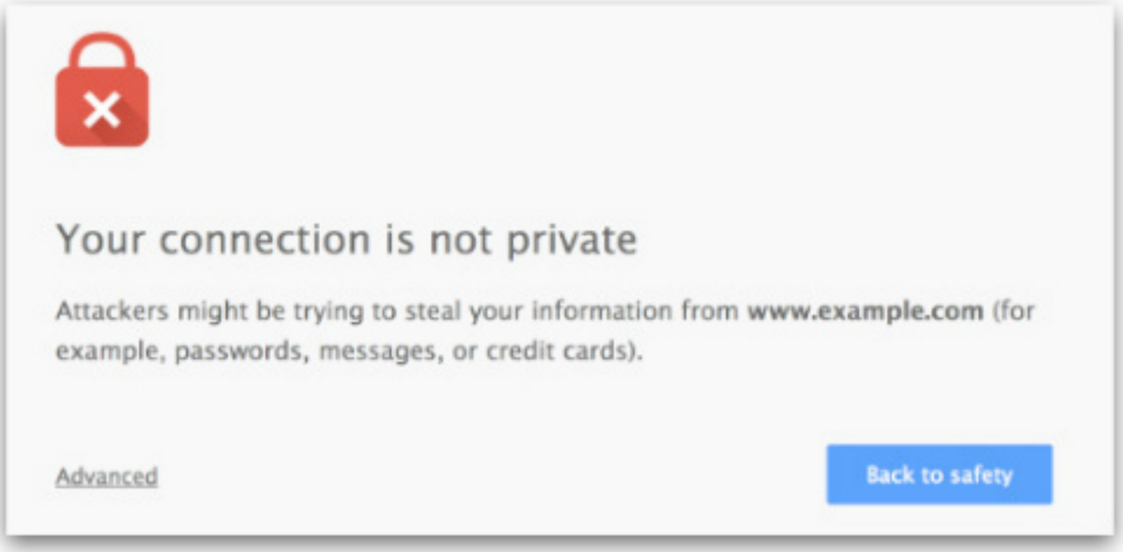


Usable confinement via APIs

- **Crypto API**
  - Convert tainted values to encrypted blobs (LastPass)
- **Declarative CSS API**
  - Taint-oblivious styling changes
- **Network filtering API**
  - Allow/deny network requests given regex (AdBlock)
- …

How can permissions be more meaningful?

- Many extensions can be safe by default
  - ➤ Confinement protects user privacy
  - ➤ Incentivize developers by making warnings rare



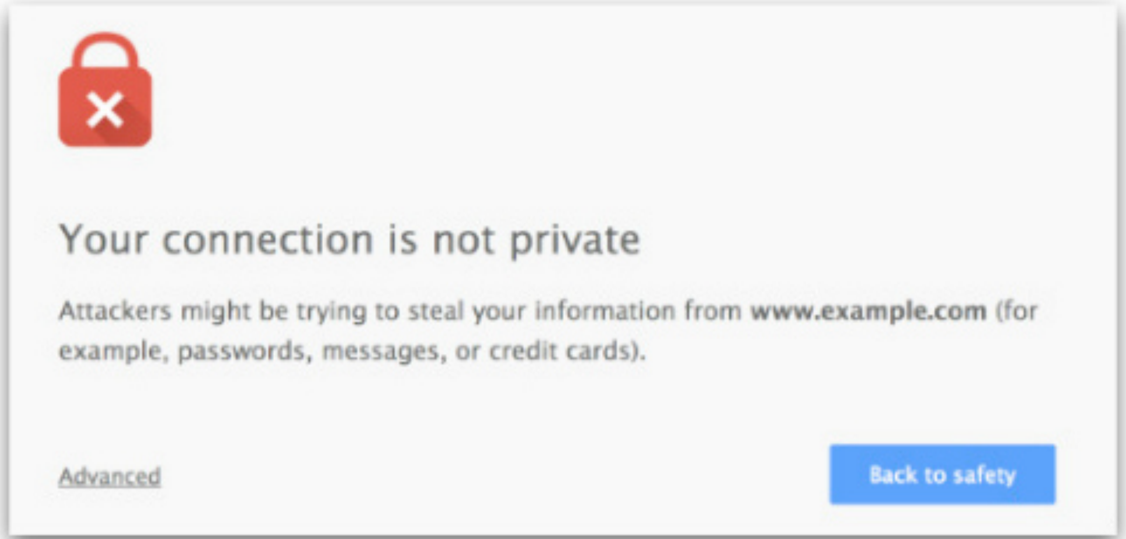
Your connection is not private

Attackers might be trying to steal your information from **www.example.com** (for example, passwords, messages, or credit cards).

- To capture remaining models: need permissions
  - ➤ Use declassification as guide for informing messages: what data is being "leaked"?
    - E.g., URLS, page location, whole page, etc.

How can permissions be more meaningful?

- **Many extensions can be safe by default**
  - ➤ Confinement protects user privacy



Your connection is not private

Attackers might be trying to steal your information from **www.example.com** (for example, passwords, messages, or credit cards).

  - ➤ Incentivize developers by making warnings rare
- **To capture remaining models: need permissions**
  - ➤ Use declassification as guide for informing messages: what data is being “leaked”?
    - E.g., URLS, page location, whole page, etc.

Summary

- **Extensions: most dangerous code in the browser**
    - Third-party, unaudited, highly-privileged JavaScript
- **Rethink extension security systems**
    - Need to protect user privacy from extensions
    - Make user permissions requests rare and clear
- **One direction: confinement + new APIs**
    - Captures many extensions as "safe", makes permission requests rare