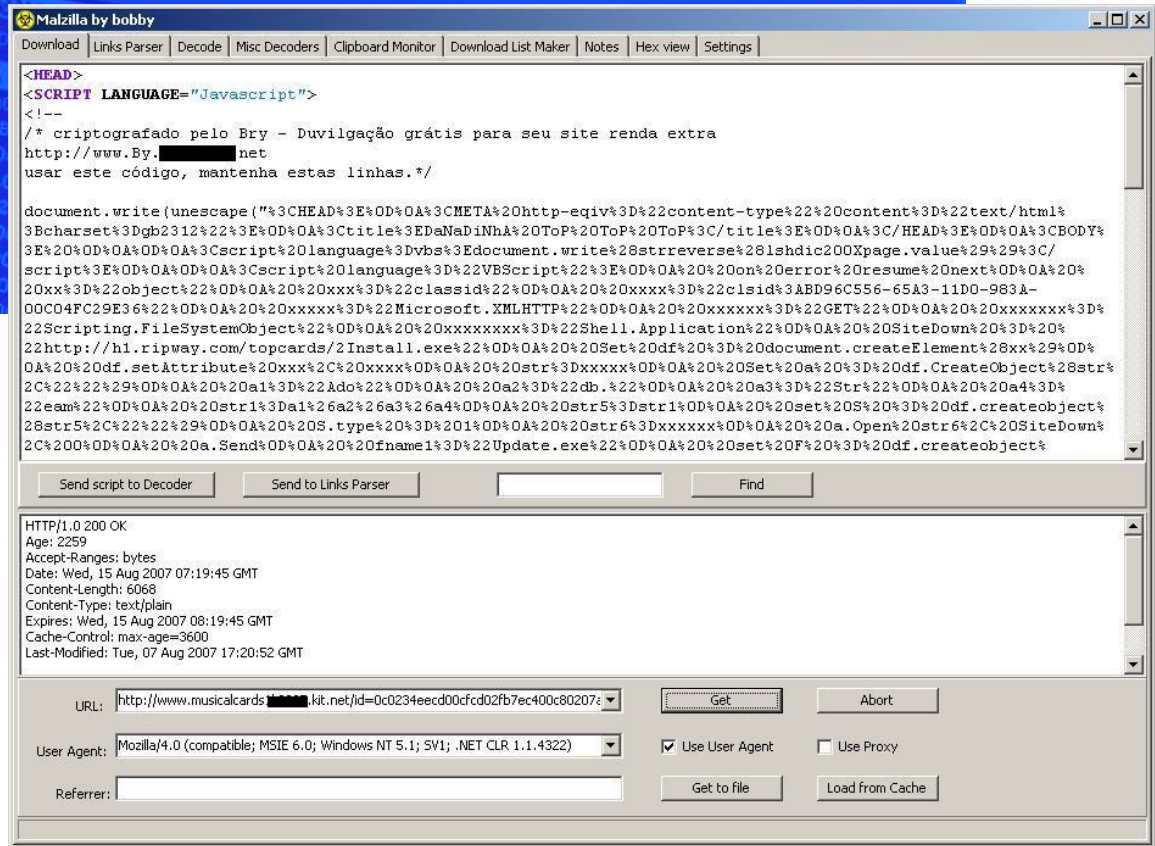


A vertical teal brushstroke on the left side of the page, with a textured, painterly appearance.

Chrome Dump

Il était une fois...



08/06/2023

ChromeDump SSTIC 2023

2

Thug

- HoneyClient basse interaction
- DOM émulé
 - À du mal face au fingerprinting JavaScript & cie.
- <https://github.com/buffer/thug>

Cas d'usages

- Analyse de sites web
 - Scripts d'analyse comportementale des utilisateurs
 - Analyse d'attaques par point d'eau
 - Récupération des codes de **fingerprinting** du navigateur
- Désobscurcissement JavaScript
 - Unpacking partiel « gratuit »
 - Pas de gestion des prédicats opaques

1^{er} essai avec Firefox

- SSTIC 2019 – Under the DOM
- Frida + Firefox
- Instable et difficile à maintenir...

Chrome Devtools Protocol

The screenshot displays the Chrome DevTools interface. The top bar includes tabs for Elements, Console, Sources, Network, Performance, Memory, Application, Security, and Lighthouse. The Elements panel shows the following HTML structure:

```
<html>
  <head>...</head>
  <body> == $0
    <input id="input">
    <script>input.onkeydown = keydown; input.onChange = () => {console.log('change')}</script>
  </body>
</html>
```

The Styles panel on the right shows the following CSS rules:

```
element.style {
}
body {
  user agent stylesheet
  display: block;
  margin: 8px;
}
```

The Protocol monitor panel is active, showing a table of CDP messages:

Method	Request	Response	Request	Response
Overlay.highlightNode	{"highligh...	{}		
Overlay.highlightNode	{"highligh...	{}		
Overlay.hideHighlight	{}	{}		
Page.reload	{"ignoreCa...	{}		
Network.requestWillBeSent		{"requestI...		
Network.responseReceived		{"requestI...		
Page.frameStartedLoading		{"frameId"...		
Target.targetInfoChanged		{"targetIn...		
Runtime.executionContext...		{}		

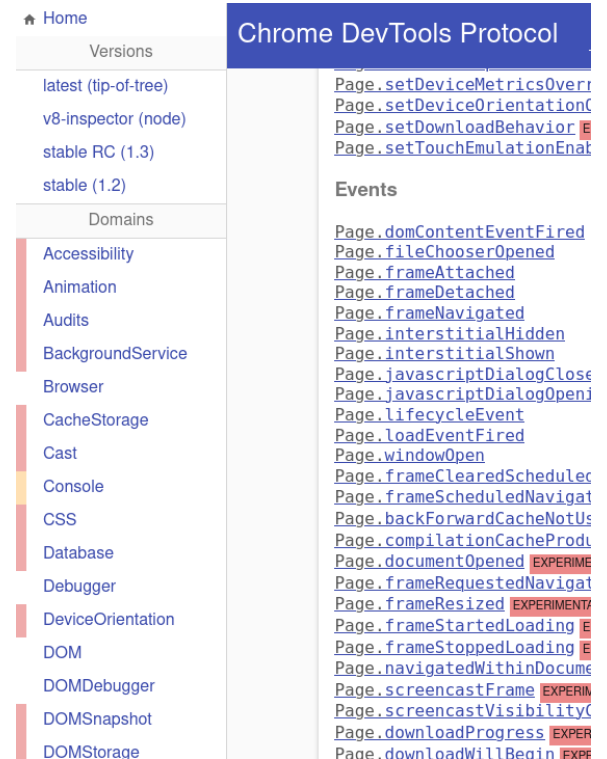
The rightmost column of the Protocol monitor shows the expanded response for the selected message:

```
{requestId: "78E0F1449A4FE365A0020F36719054BA", loaderId: "78E0F1449A4FE365A0020F36719054BA", frameId: "32ADE68F1D3B36D7A5A29312D81E8314", loaderId: "78E0F1449A4FE365A0020F36719054BA", requestId: "78E0F1449A4FE365A0020F36719054BA", response: {...}, timestamp: 135630.50558, type: "Document"}
```

At the bottom of the Protocol monitor, there is a text input field labeled "Send a raw CDP command".

Quelques API intéressantes

- Pour l'analyse JavaScript
 - `Debugger.scriptParsed`
 - `Page.javascriptDialogOpening`
- Pour les ressources échangés sur le réseau
 - `Network.requestWillBeSent`
 - `Network.responseReceived`
- Pour la géolocalisation
 - `Emulation.setGeolocationOverride`
- Pour l'automatisation
 - `Input.dispatchMouseEvent`



The image shows a screenshot of the Chrome DevTools Protocol API list. The interface is divided into two main sections: 'Domains' and 'Events'. The 'Domains' section on the left lists various protocol domains such as Accessibility, Animation, Audits, BackgroundService, Browser, CacheStorage, Cast, Console, CSS, Database, Debugger, DeviceOrientation, DOM, DOMDebugger, DOMSnapshot, and DOMStorage. The 'Events' section on the right lists various protocol events, including `Page.setDeviceMetricsOverr`, `Page.setDeviceOrientation`, `Page.setDownloadBehavior`, `Page.setTouchEmulationEnabl`, `Page.domContentEventFired`, `Page.fileChooserOpened`, `Page.frameAttached`, `Page.frameDetached`, `Page.frameNavigated`, `Page.interstitialHidden`, `Page.interstitialShown`, `Page.javascriptDialogClose`, `Page.javascriptDialogOpen`, `Page.lifecycleEvent`, `Page.loadEventFired`, `Page.windowOpen`, `Page.frameClearedScheduler`, `Page.frameScheduledNavigat`, `Page.backForwardCacheNotUs`, `Page.compilationCacheProdu`, `Page.documentOpened`, `Page.frameRequestedNavigat`, `Page.frameResized`, `Page.frameStartedLoading`, `Page.frameStoppedLoading`, `Page.navigatedWithinDocum`, `Page.screencastFrame`, `Page.screencastVisibility`, `Page.downloadProgress`, and `Page.downloadWillBeIn`. Some events are marked as experimental with a red 'EXPERIMENTAL' label.

2nd essai avec PyChrome

- Pilotage de Chrome via CDP

- <https://github.com/fate0/pychrome>

- Plus maintenu...
- Problèmes de performance conduisant à la perte de scripts
- Pas de suivis multi-onglets

- Discussion éclairante avec Ambroise Terrier

- « pourquoi t'attaque pas directement le protocole via WebSockets ? »

3ème essai : ChromeDump

- AsyncIO pour la performance
- Tornado pour les WebSockets
- Stockage temporaire en RAM ㄟ_(ツ)_ㄟ

Fonctionnalités

- Récupération des Scripts chargés dans V8
- Captures d'écrans lors de modifications visuelles de la page
- Sauvegarde du profil de navigation
- Enregistrement des requêtes émises et des réponses des serveurs

Debugger.scriptParsed

- Évènement lancé par le parseur V8 avant interprétation du JS
- Valable sur les fonctions ré-entrantes comme **eval()** ;
- Pile d'appels JS dans le fichier **jslog.json**
- Scripts JS exécutés dans **/js/**

Network.RequestWillBeSent

- Présente les requêtes envoyées par le navigateur
- Sauvegardées sous **httplogreq.json**
- Liste synthétique des urls dans **urls.txt**

Network.ResponseReceived

- Récupère les ressources chargées sur le réseau
 - JS, HTML, CSS etc..
 - Nécessaire car du JavaScript peut se cacher dans des images
- Fichiers sauvegardés sous **/files/**
- Logs sauvegardés sous **httplogres.json**

Browser.setDownloadBehavior

- Autorise le téléchargement automatique
- Les fichiers téléchargés sont sauvegardés sous **/downloads/**

Page.startScreenCast

- Produit des captures d'écrans
- Les captures sont sauvegardés sous **/screenshots/**



Démo

08/06/2023

ChromeDump SSTIC 2023

16

Et le code source alors ?

- <https://github.com/g4l4drim/ChromeDump>
- Les pull requests & issues sont bienvenues

Questions ?

À la mémoire de Sylvain Gombault

ChromeDump SSTIC 2023

08/06/2023





Références

Browser Fingerprinting

- <https://datadome.co/threat-research/detecting-selenium-chrome/>
- <https://www.f5.com/company/blog/detecting-phantomjs-based-visitors>
- <https://bot.incolumitas.com>

Scraping & CDP

- <https://incolumitas.com/2021/05/20/avoid-puppeteer-and-playwright-for-scraping/>

Devtools & CDP

- <https://chromedevtools.github.io/devtools-protocol/tot/Debugger/#event-scriptParsed>
- <https://learn.microsoft.com/fr-fr/microsoft-edge/webview2/how-to/chromium-devtools-protocol>