// BARBHACK 2026

WEAPONIZING BROWSER EXTENSIONS

...C2 Agent Injection and Fileless Persistence in Chromium

@fir3n0x

WHOAMI

...parce que c'est la tradition

@fir3n0x

Background

→ Formation Ingénieur logiciel

→ Offensive Security

→ MalDev

Contacts

x.com/fir3n0x_

github.com/Fir3n0x

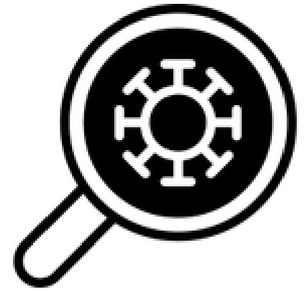
fir3n0x.github.com/

OUTLINE

...ce qu'on va couvrir aujourd'hui

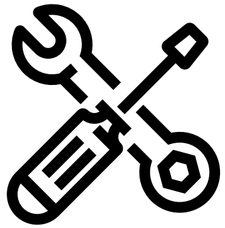
Weaponization

Overview



Persistance furtive

injection



outils

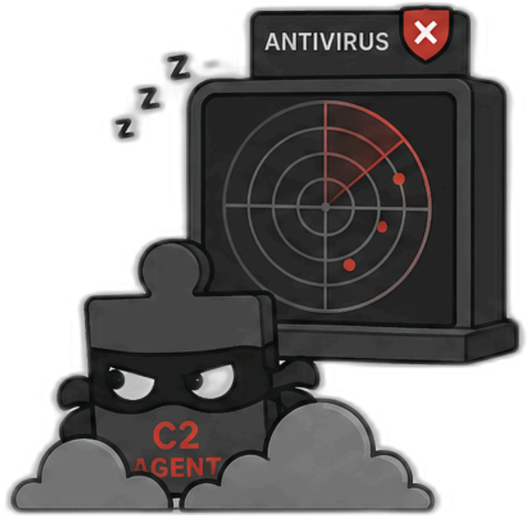
capacités

scénarios

// section 01

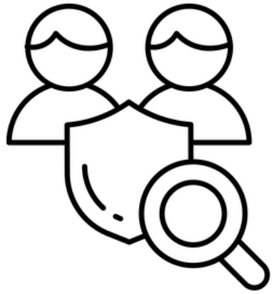
OVERVIEW

...contexte et intérêts d'un agent extension



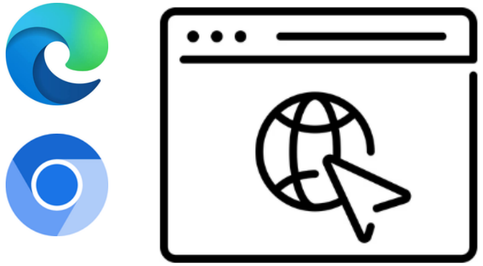
CONTEXTE

...étude de recherche, une alternative aux agents systèmes ?



Détection des implants de plus en plus avancée (tant mieux!)

Étudier une alternative aux agents systèmes en post exploitation



Surface d'attaque moins surveillée avec un agent extension

INTÉRÊTS

...il y a des avantages à utiliser des extensions

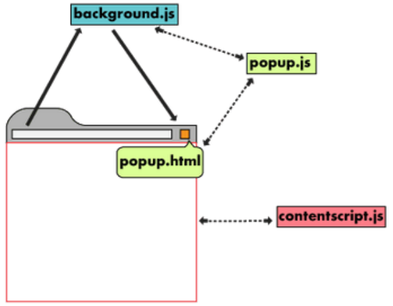
LE NAVIGATEUR EST UNE MINE D'OR

- M365, GitHub, SharePoint, Teams, OneDrive - les données sensibles vivent dans le navigateur;
- Cookies de session, tokens MFA, credential - déjà présents;
- Le navigateur = nouveau poste de travail : messagerie, documents ou outils métier;
- Sessions actives = accès direct aux ressources cloud de la victime.

EXTENSIONS VS AGENTS CLASSIQUES

✓ Processus légitime	✓ Cross-platform
✓ Pas d'injection mémoire	✓ EDR ? Connais pas
✓ Persistance naturelle	✗ Capacités équivalentes

STRUCTURE EXTENSION



...c'est quoi une extension de navigateur ?

Une **extension** est un petit module logiciel ajouté au navigateur :

- **Environnement isolé** (sandbox);
- Déclare ses permissions et capacités via un fichier **manifest.json**;
- Accès à des **API** (chrome, browser).

Se décompose en 3 parties distinctes :

- **manifest.json :** Il définit les métadonnées de base, les permissions demandées, les scripts et ressources de l'extension. Il est **indispensable**;
- **Les content_scripts :** fichiers JavaScript injectés dans les pages web visitées par l'utilisateur;
- **Le Service Worker ou background script :** script JavaScript exécuté en arrière-plan qui réagit aux événements du navigateur, possède l'accès à l'API du navigateur et permet la communication avec des services tiers.

```
{
  "manifest_version": 3,
  "name": "Test Communication Serveur",
  "version": "1.0",
  "description": "Communication avec serveur python",
  "permissions": [
    "alarms"
  ],
  "host_permissions": [
    "https://domain.com/"
  ],
  "background": {
    "service_worker": "background.js"
  },
  "content_scripts": [
    {
      "matches": [
        "<all_urls>"
      ],
      "js": [
        "content.js"
      ]
    }
  ],
  "key": "MIIBIjANBgkqhkiG9w0[...]Sej5JnUDpU5xoCY+wN0+Lu/1K/y/
0zheb24qSSQQ4qm5yDn1g/V8CC42eL/T0Xx2EfMNKb4dHpOs0sCmy2TqLbCp/
cyFYfaDdwIDAQAB"
}
```

// section 02

WEAPONIZATION

...capacités, injection, bypass GPO et quelques outils sympas

INJECTION

...il y a différentes manières de faire l'injection

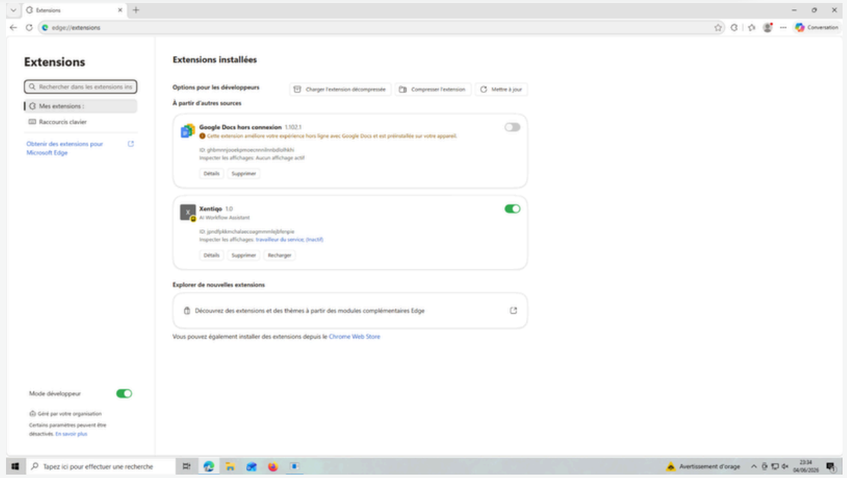
unpacked_extension/

- manifest.json
- content_scripts (.js)
- service worker (.js)

Injection d'une extension dans un navigateur Chromium

→ Installation depuis le store officiel (Edge/Chrome store);
→ Via des stratégies de groupe (GPOs / politiques d'entreprise);
→ Avec le mode développeur :

- Chargement d'une extension non empaquetée depuis un dossier local; ← **méthode étudiée**
- Chargement d'une extension auto-signée au format .crx.



Notes: les extensions téléchargées depuis le store vont se retrouver ici sous forme de dossier C:\\Users\\<user>\\%LOCALAPPDATA%\\Microsoft\\Edge\\User Data\\Default\\Extensions

INJECTION

...facile, suffit d'importer l'extension depuis l'interface

Requiert un **accès initial** : Test sur une VM Windows 11 sans restriction.
Le dossier de l'extension malveillante doit être **présent** sur le disque de la victime.
On injecte depuis l'**interface web**.

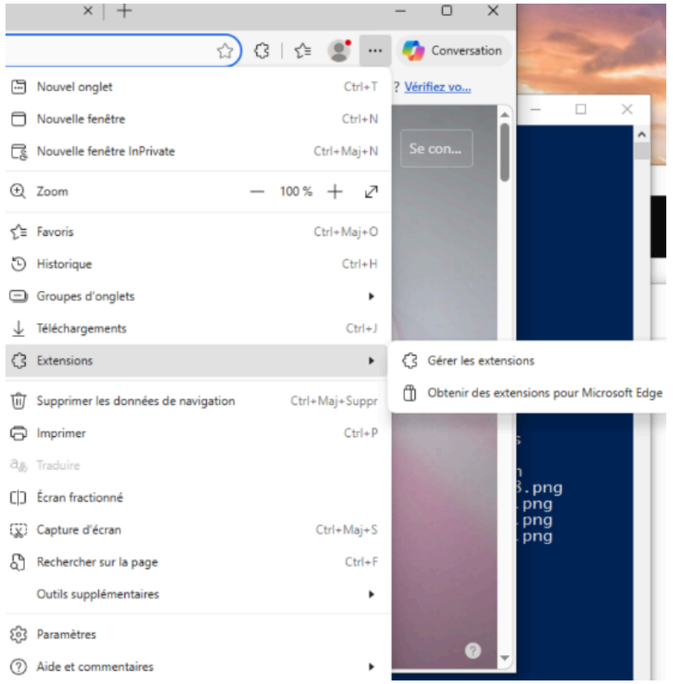
Attaquant

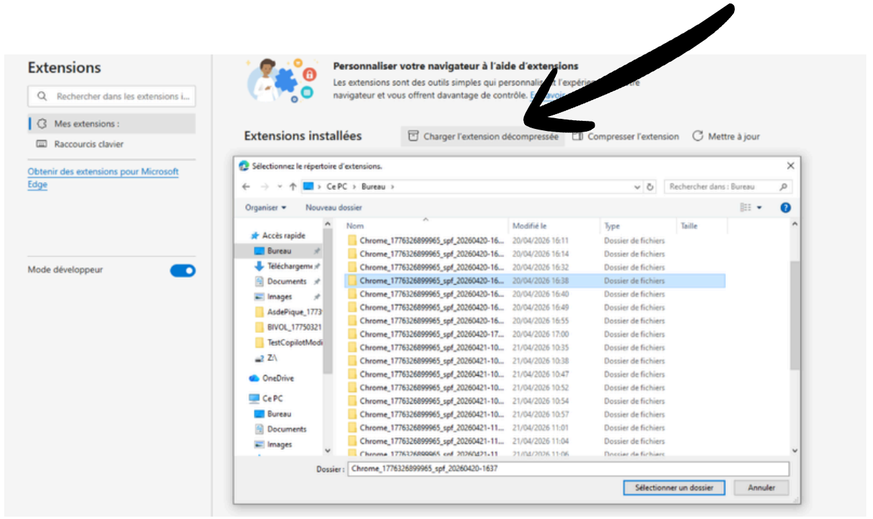
Déploie l'extension

Ordinateur victime

Connexion vers C2

Serveur C2

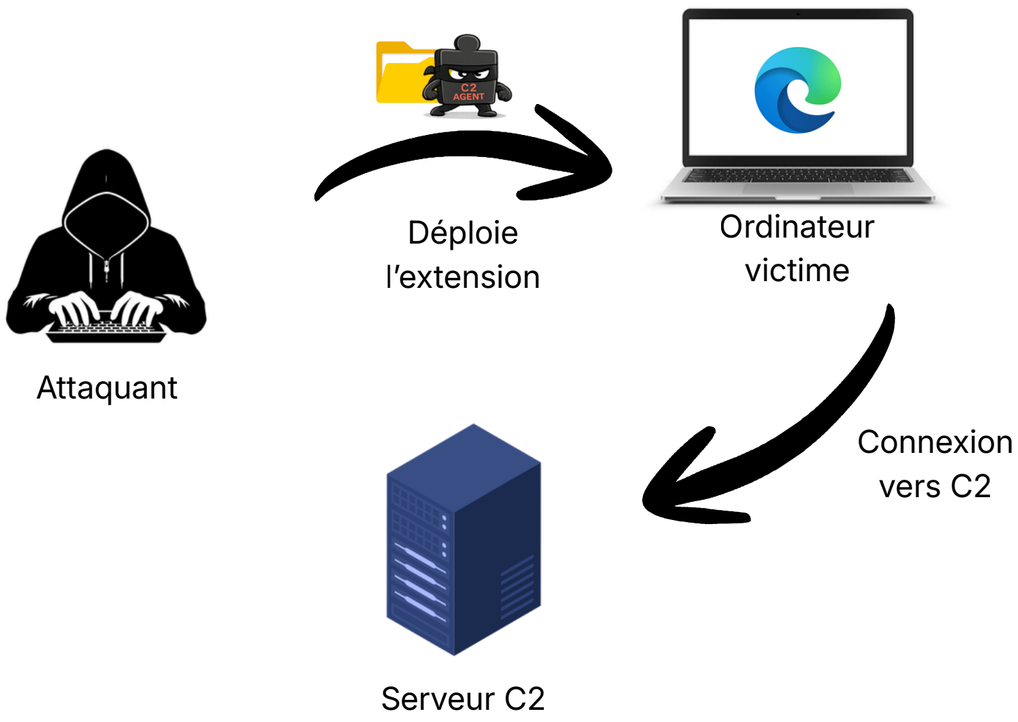




INJECTION

...ça marche pas si bien que ça

Requiert un **accès initial** : Test sur une VM Windows 11 sans restriction.
Le dossier de l'extension malveillante doit être **présent** sur le disque de la victime.
On injecte depuis l'**interface web**.

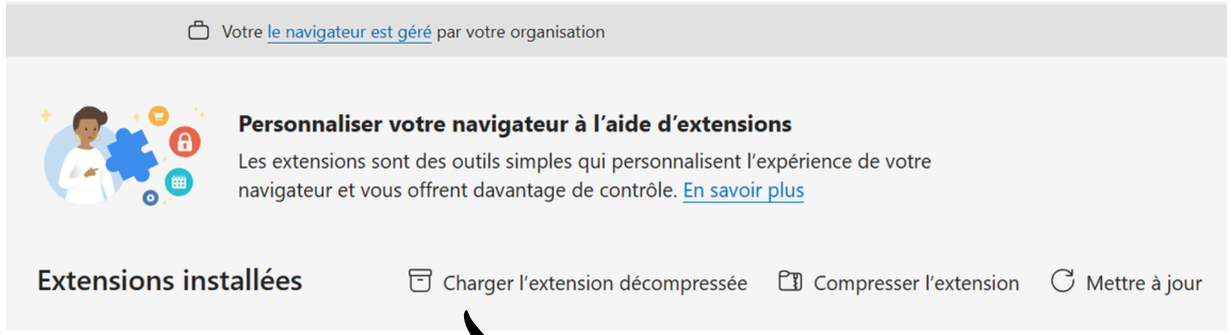


LIMITES /!\

→ Pas toujours accès à l'ordinateur de la victime avec une GUI;
→ Au sein d'une organisation/entreprise, il peut y avoir des GPOs qui empêchent l'injection manuelle d'extension.

INJECTION

...identification du blocage par les politiques de sécurité



ne réagit pas :/

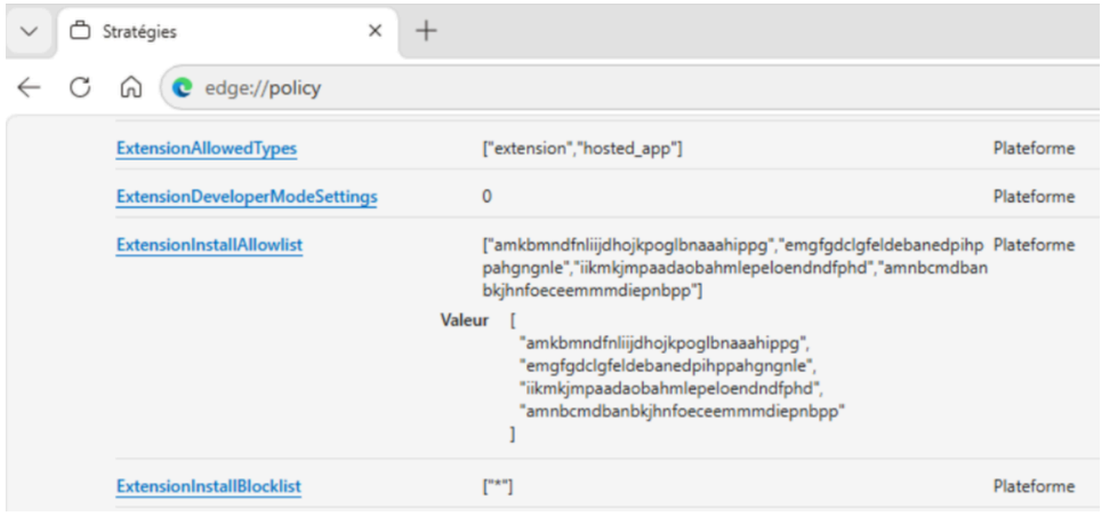
2 défis :

→ Injection de l'agent dans le navigateur;

→ Spoofing de l'ID d'une extension légitime.

2 GPOs bloquantes :

- > **ExtensionInstallAllowlist**
- > **ExtensionInstallBlocklist**

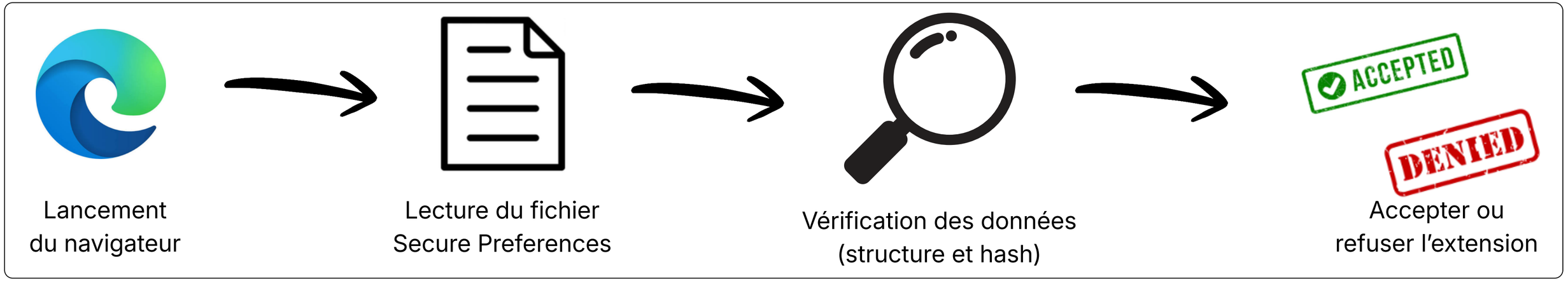


INJECTION DE L'AGENT

...le fichier Secure Preferences

Dans les navigateurs **Chromium**, les données relatives aux extensions sont enregistrées dans le fichier **Secure Preferences**.
Pour Microsoft Edge, ce fichier se localise :

C:\\Users\\<user>\\%LOCALAPPDATA%\\Microsoft\\Edge\\User Data\\Default\\Secure Preferences



On peut modifier ce fichier pour injecter une **extension arbitraire**, en recalculant les bonnes valeurs d'intégrité.

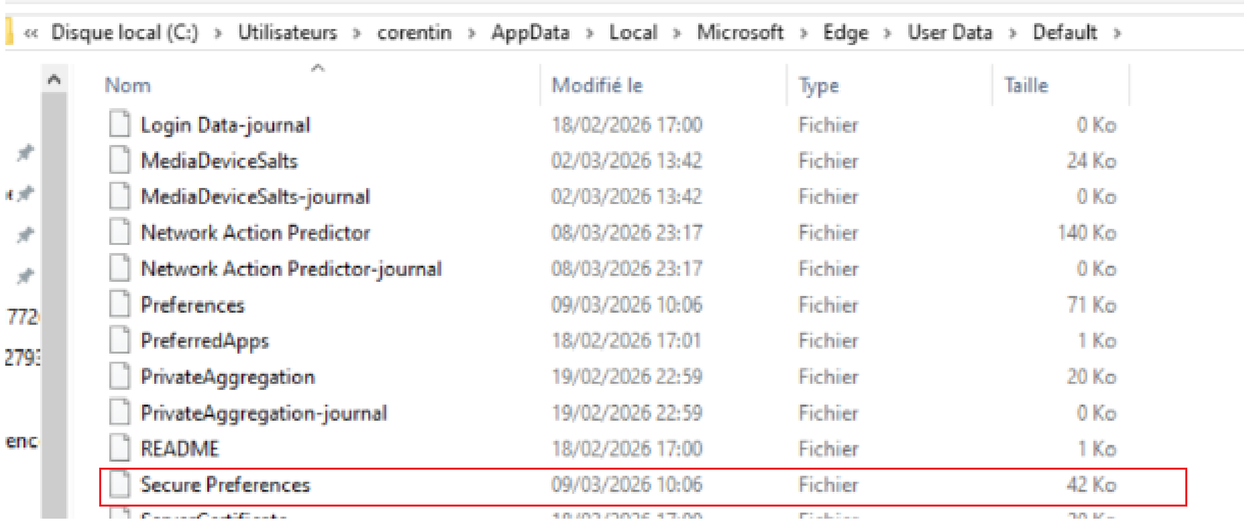
Explication détaillée du mécanisme.

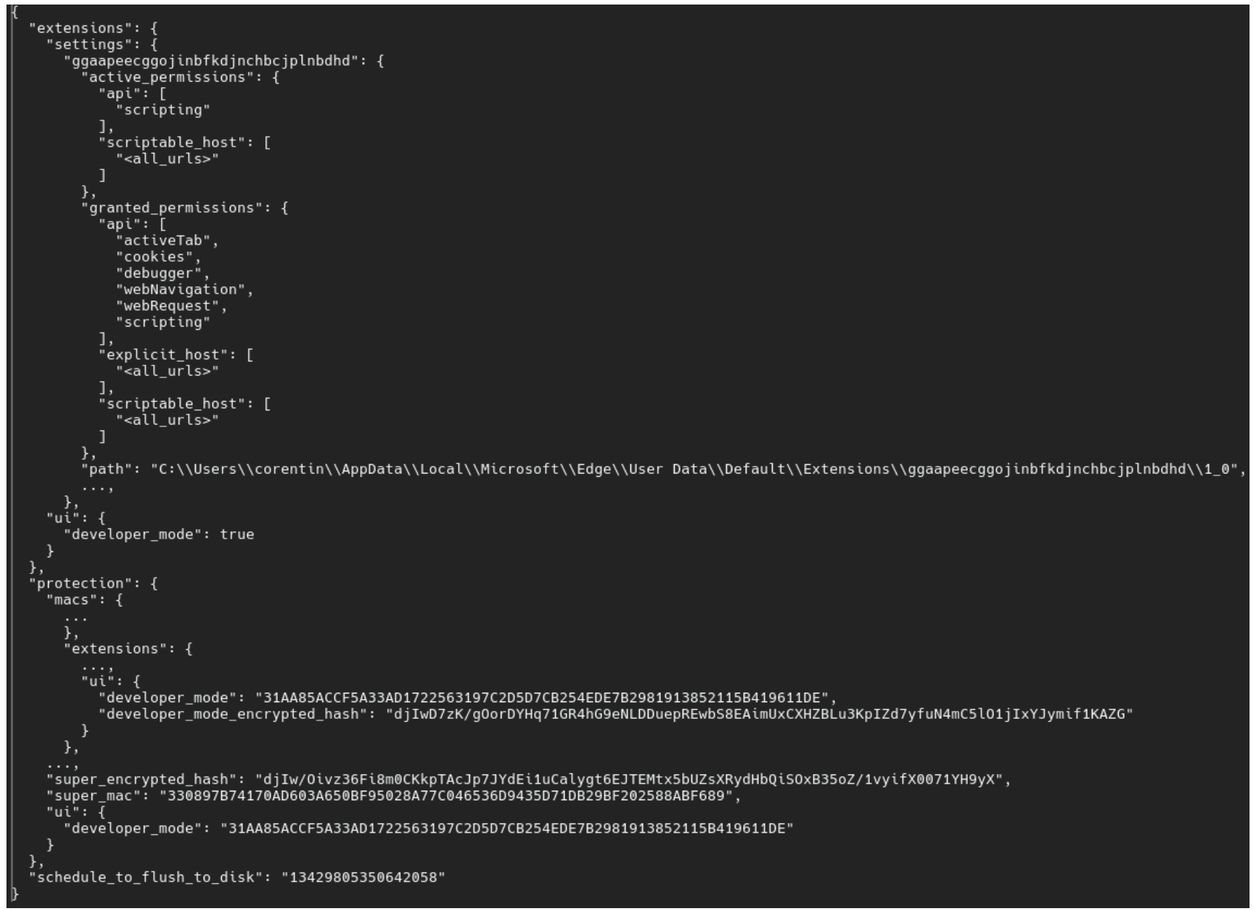
→ https://www.synacktiv.com/publications/lextension-fantome-infiltrer-chrome-par-des-voies-inexplorees
→ https://www.cse.chalmers.se/~andrei/cans20.pdf

INJECTION DE L'AGENT

...bypass ExtensionInstallBlocklist GPO

Injecter les données relatives à l'extension (ID, chemin de déploiement sur le disque,...) et **Calculer** les valeurs HMAC ainsi que le super_mac.





```
{
  "extensions": {
    "settings": {
      "ggaapeecggojinbfkdjnchbcjplnbdhd": {
        "active_permissions": {
          "api": [
            "scripting"
          ],
          "scriptable_host": [
            "<all_urls>"
          ]
        },
        "granted_permissions": {
          "api": [
            "activeTab",
            "cookies",
            "debugger",
            "webNavigation",
            "webRequest",
            "scripting"
          ],
          "explicit_host": [
            "<all_urls>"
          ],
          "scriptable_host": [
            "<all_urls>"
          ]
        },
        "path": "C:\\Users\\corentin\\AppData\\Local\\Microsoft\\Edge\\User Data\\Default\\Extensions\\ggaapeecggojinbfkdjnchbcjplnbdhd\\1_0",
        ...,
      },
    "ui": {
      "developer_mode": true
    }
  },
  "protection": {
    "macs": {
      ...
      },
      "extensions": {
        ...,
        "ui": {
          "developer_mode": "31AA85ACCF5A33AD1722563197C2D5D7CB254EDE7B2981913852115B419611DE",
          "developer_mode_encrypted_hash": "djIwD7zK/gOorDYHq71GR4hG9eNLDDuepREwbS8EAimUxCXHZBLu3KpIZd7yfuN4mC5lO1jIxYJymif1KAZG"
        }
      },
    ...,
    "super_encrypted_hash": "djIw/Oivz36Fi8m0CKkpTAcJp7JYdEi1uCalygt6EJTEMtx5bUZsXRydHbQiSOxB35oZ/1vyifX0071YH9yX",
    "super_mac": "330897B74170AD603A650BF95028A77C046536D9435D71DB29BF202588ABF689",
    "ui": {
      "developer_mode": "31AA85ACCF5A33AD1722563197C2D5D7CB254EDE7B2981913852115B419611DE"
    }
  },
  "schedule_to_flush_to_disk": "13429805350642058"
}
```

→ Précalculer l'ID de l'extension;
→ Calculer le MAC pour protection.macs.extensions.settings[ID_EXT];
→ Calculer le MAC pour le mode développeur;
→ Calculer le super_mac du fichier Secure Preferences.

NÉCESSITE :

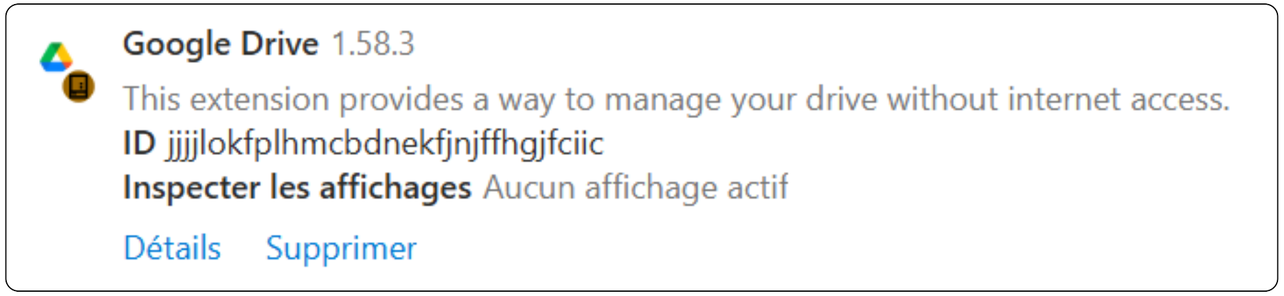
- **SID** (pour Windows → whoami /user);
- **Hardware UUID** (pour macOS → system_profiler SPHardwareDataType).

INJECTION DE L'AGENT

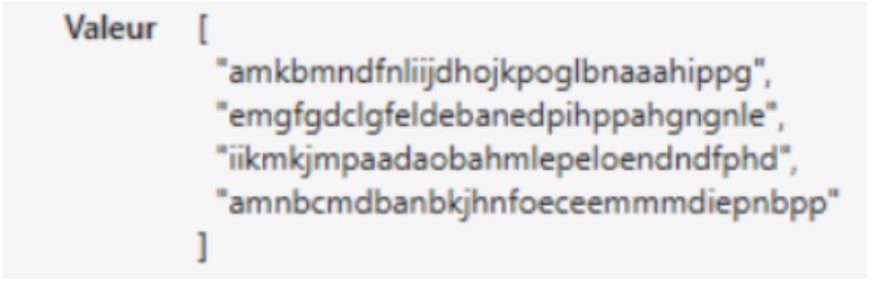
...une autre GPO à contourner

- > Injection de l'agent dans le navigateur.

MAIS mauvais ID d'extension car généré à partir du chemin du dossier (absence de clé publique).



∉



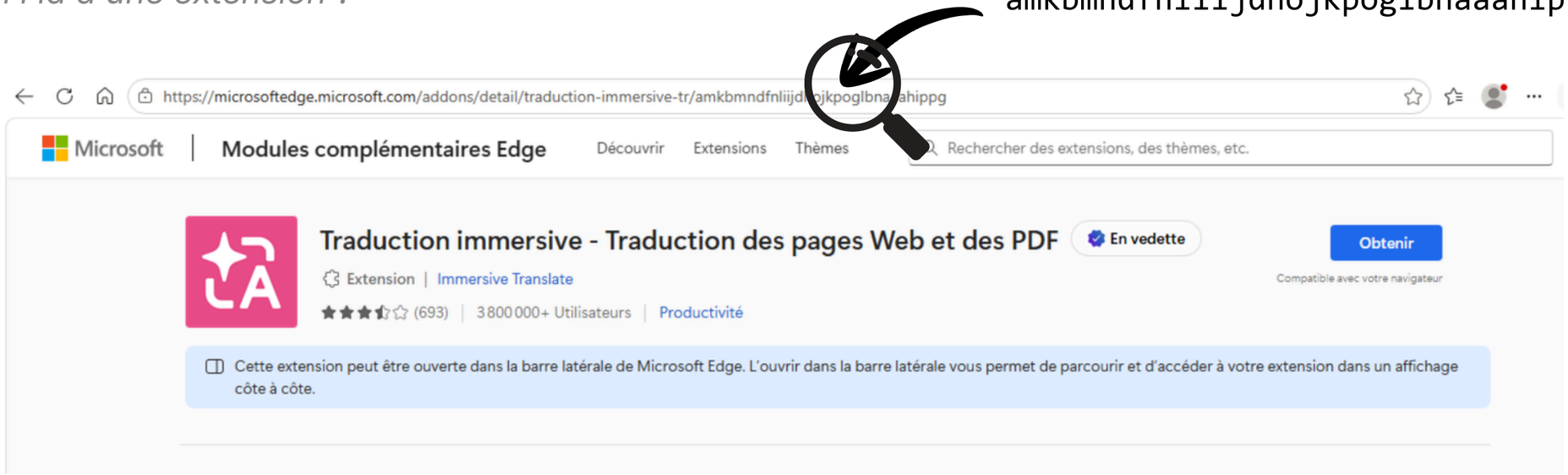
Ces IDs font référence à de vraies extensions disponibles sur le store.

→ Spoofing de l'ID d'une extension légitime.

SPOOFING D'UN ID

...c'est quoi l'id d'une extension ?

amkbmndfnliijdhojkpoglbnaaahippg



Identifie l'extension et est stockée dans le fichier Secure Preferences et est calculée automatiquement par le navigateur à partie de la **clé publique** se trouvant dans le **manifest.json**.

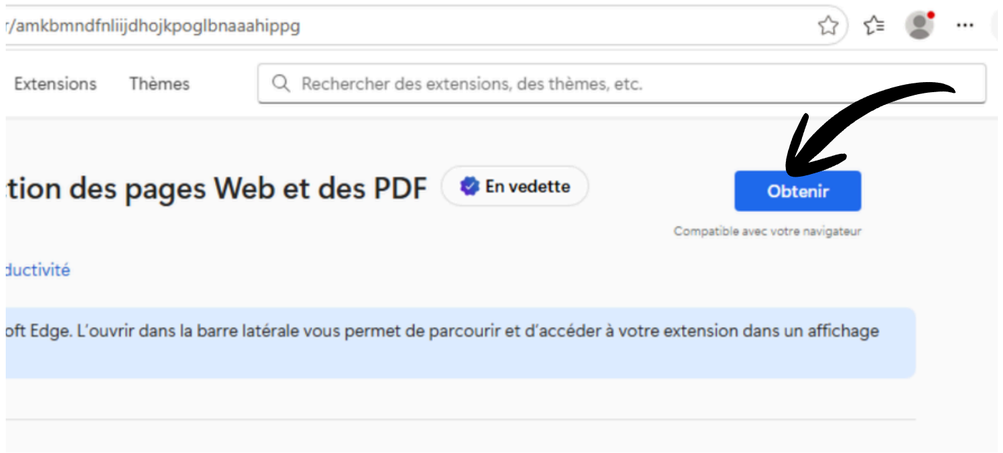
ID = sha256(pub_key) → 32 premiers caractères →
Convertir chaque caractère hex en lettre 0 → a, ..., f → p

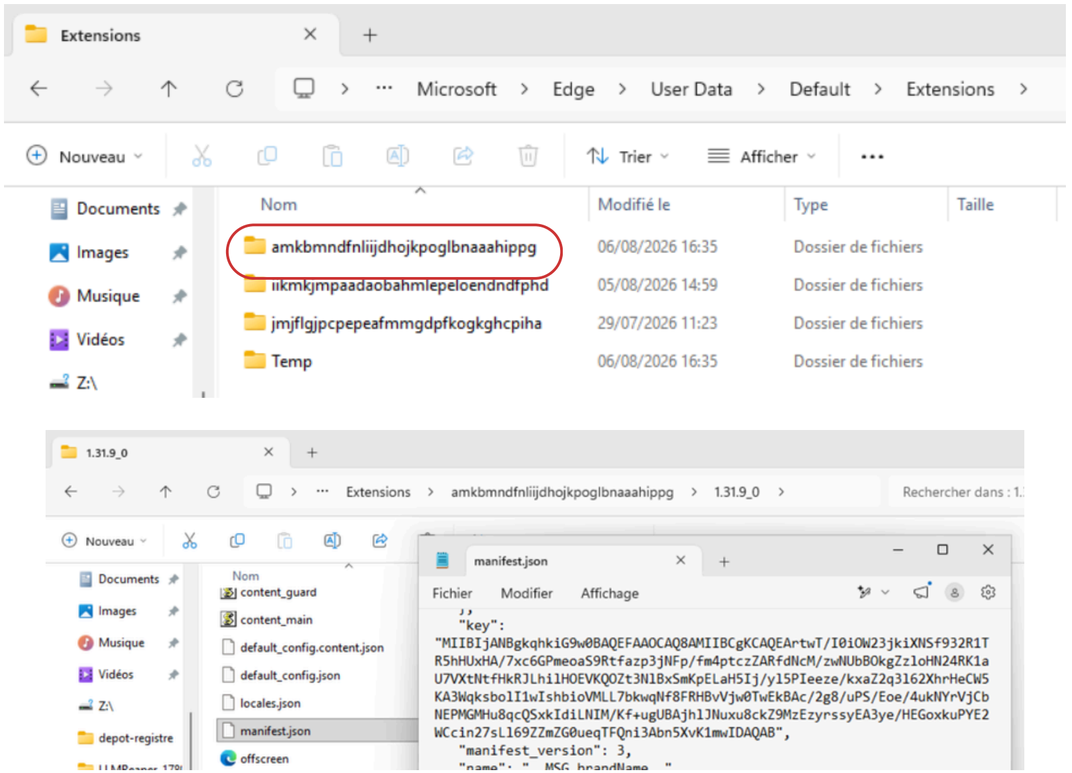
```
  ],
  "key": "MIIBIjANBgkqhkiG9w0[...]Sej5JnUDpU5xoCY+wN0+Lu/1K/y/
0zheb24qSSQQ4qm5yDn1g/V8CC42eL/T0Xx2EfMNKb4dHpOs0sCmy2TqLbCp/
cyFYfaDdwIDAQAB"
}
```

SPOOFING D'UN ID

...récupérer la clé publique d'une extension du store

- > Récupérer le dossier de l'extension.





- > Injecter la clé publique dans le manifest.json de **NOTRE** agent.

clé publique à spoofer

extension malveillante

manifest.json

OUTILS

...présentation de stomp, exécution assistée

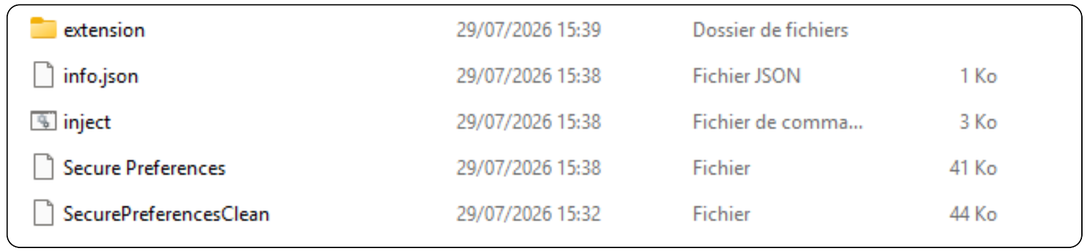
On récupère le fichier **Secure Preferences**, on le modifie en local et on le réinjecte sur la machine victime.

stomp va:

- Récupérer la clé publique à partir d'un id d'extension en entrée (--spoof, optionnel en présence de GPOs);
- Insérer la clé publique récupérée dans le manifest.json (champ 'key');
- Générer le nouveau **Secure Preferences** avec l'extension malveillante (inspiré de l'outil de Synacktiv);
- Packager des fichiers générés par stomp dans une archive .zip avec le fichier **inject.bat** et une backup du Secure Preferences original.

stomp.py

```
python3 stomp.py <EXT> --platform <PLAT> --spoof <ID> --prefs-file SecurePreferences --device-id <SID/UUID> --target-dir <DEPLOY_PATH> --browser <chrome/edge/vivaldi/brave>
```



→ Exécuter **inject.bat.**

→ https://github.com/Fir3n0x/stomp

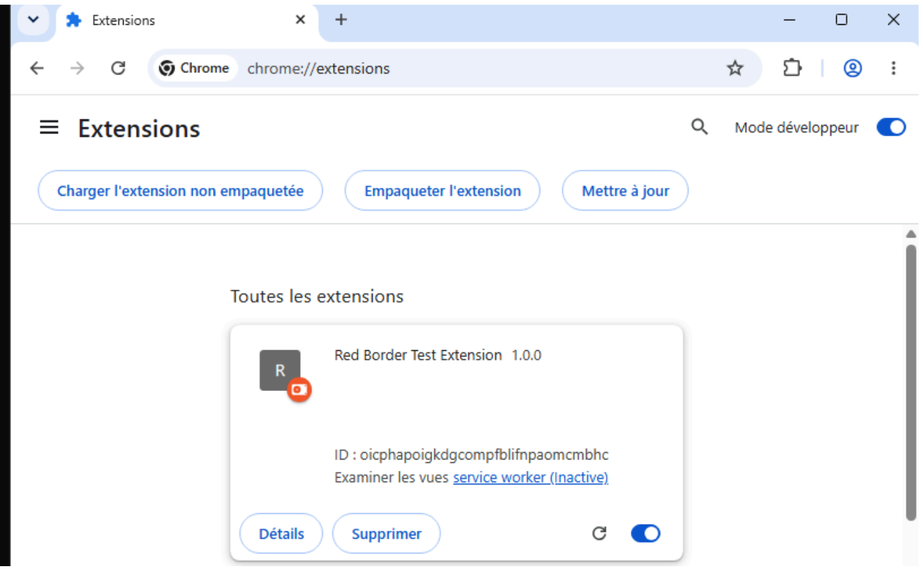
OUTILS

...présentation de BELoader, exécution autonome

Binaire qui **embarque** l'extension, **modifie** le fichier Secure Preferences et **place** l'extension sur le disque.

Il y a également une vérification des GPOs, s'il n'y en a pas → "Error opening Key pair" → Génération d'une paire de clés locale.

```
PS C:\Users\corentin\Desktop> .\BELoader.exe --browser chrome
Fichiers embarqués :
 - content.js
 - icons
 - manifest.json
 - service-worker.js
[*] Browser closed
Every process browser instances were killed...
[!] Error opening Key: Le fichier spécifié est introuvable.
[+] Generated new key pair → ID: oicphapoigkdgcompfblifnpaomcmbhc
[+] Clé publique injectée
[+] Extension MAC: 259DF6D0ED3CB241D45B10E973E4FBBD4C4E6E27F7823DDD0BAC49490A315781
[+] Developer Mode MAC: 63B95BB79D7A522293AAFA647867F03D3B13BD08CD66FF298DB2F4C91D425D36
[+] Preferences patched
[+] Injection successful!
PS C:\Users\corentin\Desktop>
```



→ Exécuter le **binaire.**

→ https://github.com/Fir3n0x/BELoader

OUTILS

...quelles sont les différences ??

Les deux outils ont pour finalité d'injecter une extension pour weaponize un navigateur Chromium, compatible Windows ou MacOS.

	Language	Format	Furtivité	Mise en place	Bypass solution de sécurité
STOMP	Python	Archive (avec .bat)	Discret	Assistée	Oui
BELOADER	Go	Binaire (Lourd ~5 Mb)	Plus bruyant	Autonome	Moyennement en l'état

Microsoft Defender for Endpoint

Si :

- écrit en C (~100 Ko)
- OPSEC (obfuscation, dynamic API resolution, ...)

Alors oui sans problème !

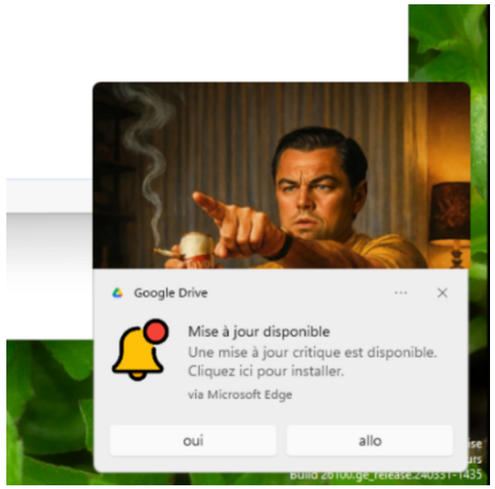
→ https://github.com/Fir3n0x/BELoader

CAPACITÉS

...ce qu'on peut faire, beaucoup de choses au final

bitwarden

KeePass



Activité de l'utilisateur (Interception du traffic réseau).
Exfiltration de données (keylogger, clipboard, documents téléchargés, historique, marque-pages, LLM,...).
Injecter du code JavaScript dans le DOM des pages.
Upload de fichiers dans le dossier de téléchargement par défaut de l'hôte.
Lecture arbitraire de fichiers sur l'hôte.
Envoie de notifications système.
Exécution de fichiers via des notifications système (Forcer le téléchargement et exécuter via notifications).
Exécution de commandes systèmes (via l'API chrome Native Messaging → nécessite un binaire supplémentaier).
...

CAPACITÉS

...concrètement, quelques exemples de l'API chrome

Accès aux données sensibles

`chrome.cookies` Accès à tous les cookies.

`chrome.history` Tout l'historique de navigation.

Interception du traffic

`chrome.webRequest` Listener sur toutes les requêtes/réponses sortantes,etc → harvesting passif.

`chrome.declarativeNetRequest` Version MV3, remplace webRequest.

Injection de code

`chrome.scripting.executeScript()` Injecte du JS dans n'importe quel onglet ouvert. Keylogging, exfiltration de formulaires, manipulation du DOM.

`chrome.tabs` Liste les onglets ouverts.

Infrastructure C2

`chrome.alarms` Beaconing survit aux rechargements du browser.

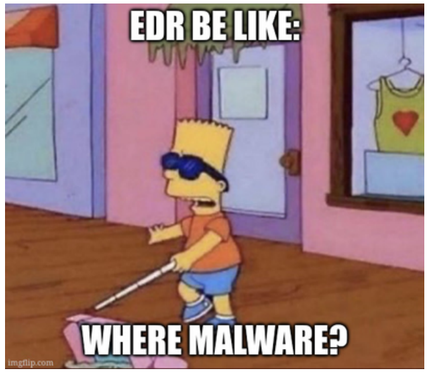
`chrome.storage` Persistance locale pour stocker des données collectées avant exfil.

NE PAS OUBLIER /!\

// Dans le manifest.json :

→ Ajouter les permissions nécessaires;

→ Ajouter '<all_urls>' dans "host_permissions" pour autoriser l'action sur tous les domaines.



→ https://developer.chrome.com/docs/extensions/reference/api - chrome API
→ https://developer.chrome.com/docs/extensions/reference/manifest - manifest format

SCÉNARIOS

...mise en place d'un module RCE, on sort de la sandbox

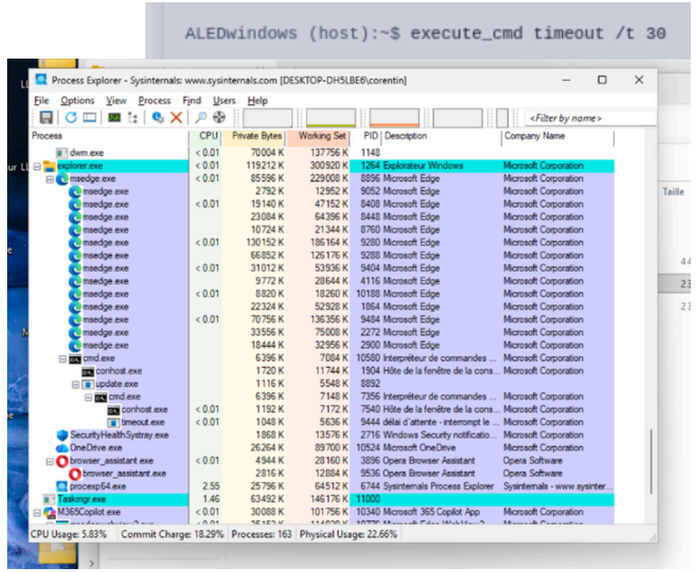
`chrome.sendNativeMessage`

Permet officiellement que l'extension puisse communiquer avec un binaire présent sur le disque de manière ponctuelle.

- > Le binaire doit être enregistré sur le système à partir d'un fichier JSON, placé à un endroit précis :

> Windows → chemin arbitraire référençant par un registre (HKCU\Software\<browser>\NativeMessagingHosts\com.*.* → pointe vers le chemin du JSON);
> MacOS→ ~/Library/Application Support/<browser>/NativeMessingHosts/*.json;
> Linux → ~/.config/<browser>/NativeMessagingHosts/*.json.

sendNativeMessage — lit le JSON associé — lance le processus



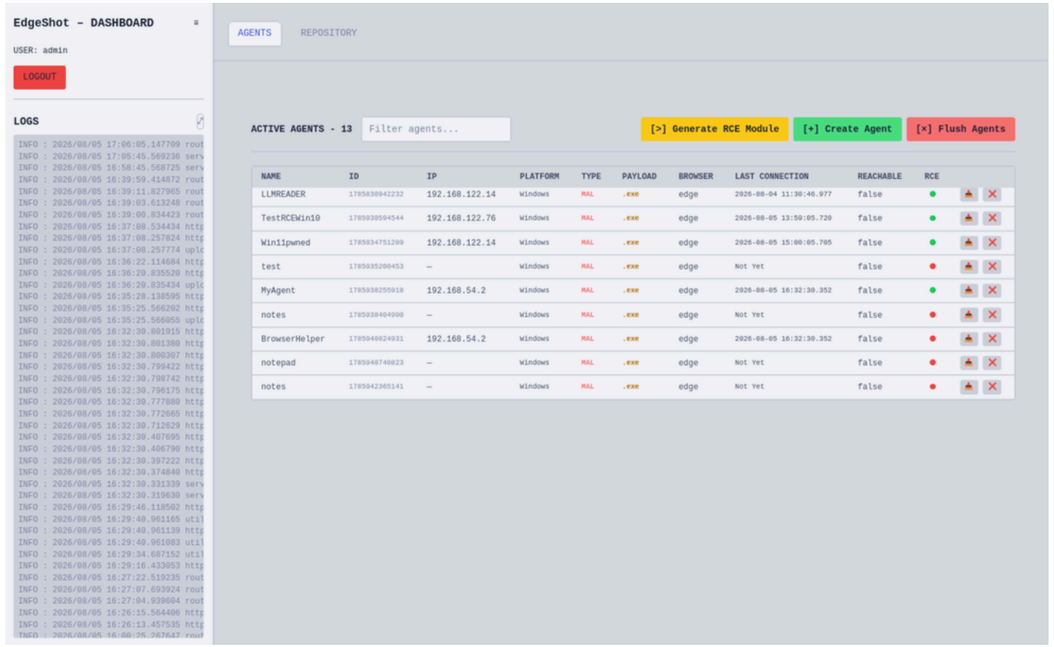
→ **La communication ne transite pas par du traffic réseau, c'est de l'IPC local orchestré par le navigateur (stdin/stdout).**

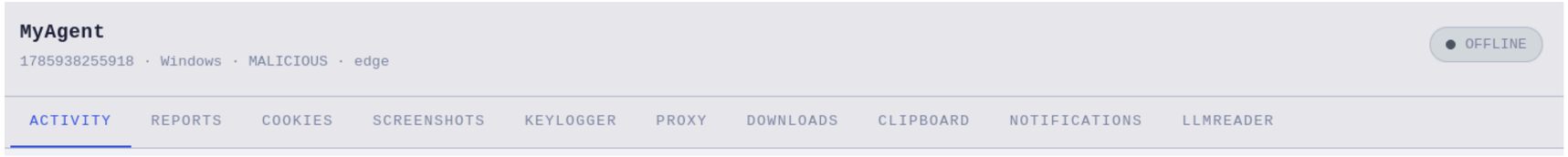
SCÉNARIOS

...on mélange tout dans un C2

stomp

BELoader





// section demo

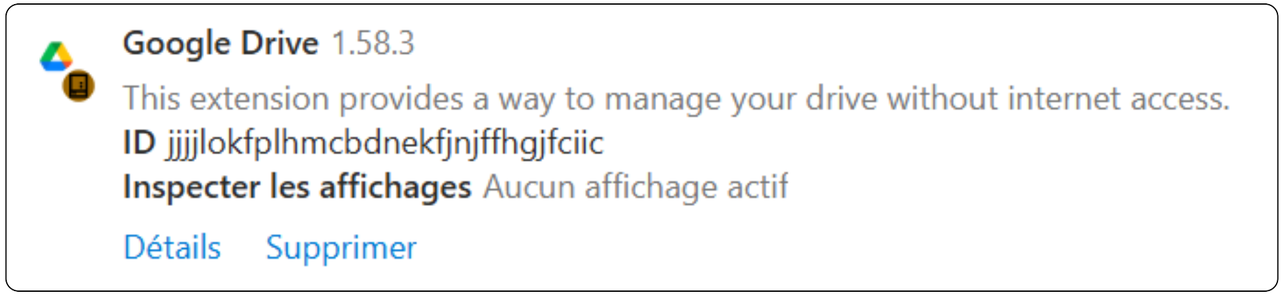
DEMO

...les capacités, module RCE

SCÉNARIOS

...le dossier malveillant est un IOC

Quand on **injecte** une extension dans le navigateur, le **dossier** est présent dans le **SF** et crée un IOC.



Le dossier de l'extension malveillante peut subire une **analyse statique**...

// section 03

PERSISTANCE FURTIVE

...encore une histoire de cache persistant

PERSISTANCE FURTIVE

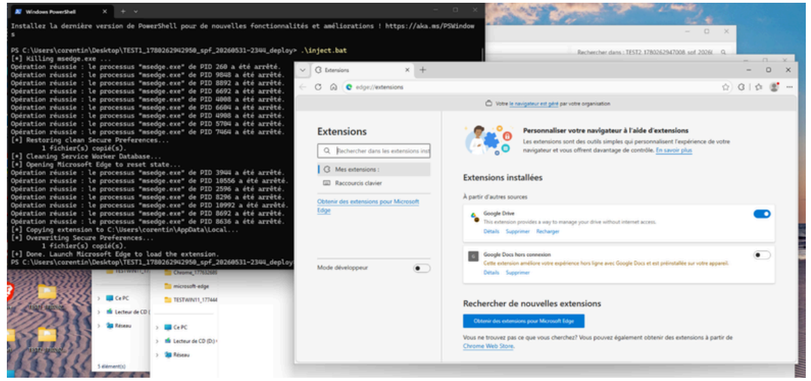
...un comportement inattendu

ACTIVE AGENTS - 2 | Filter agents... | [+] Create Agent | [×] Flush Agents

NAME	ID	IP	BROWSER	LAST CONNECTION	REACHABLE
TEST1	1780262942950	–	edge	Not Yet	false
TEST2	1780262947008	–	edge	Not Yet	false

Comportement particulier avec 2 extensions ayant le même ID et les mêmes noms de fichier.

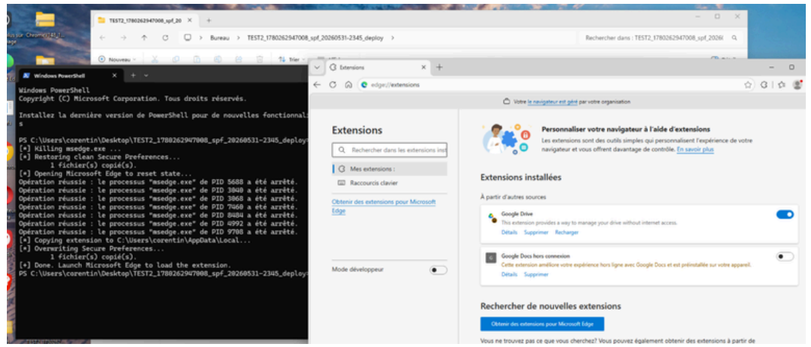
Injection de l'extension TEST1 :



NAME	ID	IP	BROWSER	LAST CONNECTION	REACHABLE
[>] TEST1	1780262942950	192.168.54.2	edge	2026-05-31 23:48:35.436	true
TEST2	1780262947008	–	edge	Not Yet	false

Réception du beacon de l'extension TEST1.

Injection de l'extension TEST2 :



NAME	ID	IP	BROWSER	LAST CONNECTION	REACHABLE
[>] TEST1	1780262942950	192.168.54.2	edge	2026-05-31 23:57:08.722	true
TEST2	1780262947008	–	edge	Not Yet	false

Après avoir écrasé la précédente extension, c'est le code de TEST1 qui ping back vers mon C2 alors que j'ai injecté TEST2 ???

PERSISTANCE FURTIVE

...le service worker et son cache

1 **Edge se lance**

Lit Secure Preferences → identifie les extensions installées.

2 **Service Worker démarre**

background.js est compilé et stocké en cache :

`Service Worker\ScriptCache\`

3 **Enregistrement persistant (LevelDB)**

ID extension → URL script → clé de cache

`Service Worker\Database\`

4 **ScriptCache**

Script compilé indexé par URL, pas par contenu du fichier

`Service Worker\ScriptCache\`

// CLÉ D'INDEXATION DU CACHE

Le ScriptCache est indexé par URL du script, pas par son contenu :

```
// schéma
chrome-extension://<ID>/<fichier>

// exemple réel (background.js est notre script service worker ici)
chrome-extension://amkbmndfnliijdhojkpoglbnaaahippg/background.js
```

// INSIGHT CLÉ

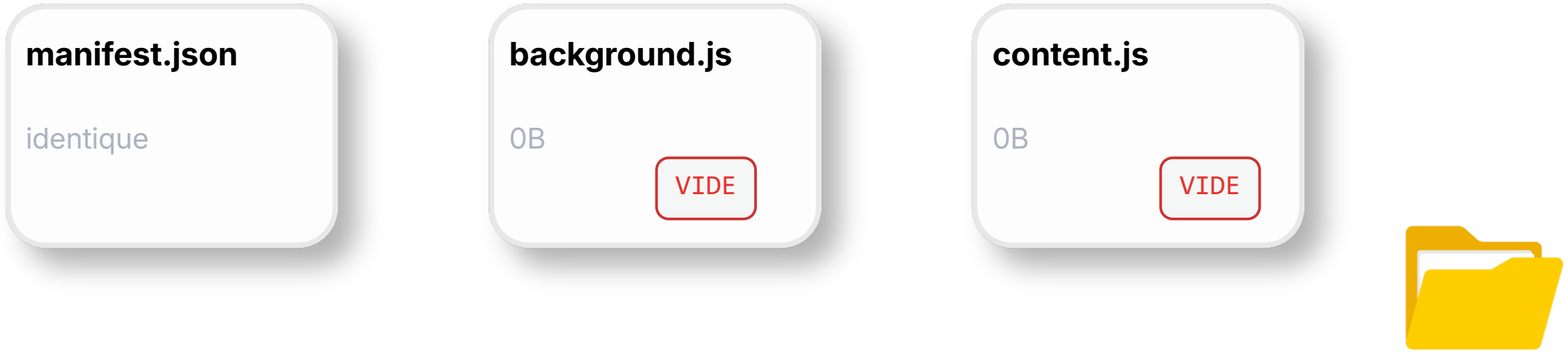
Au redémarrage, Edge vérifie que le dossier existe, puis sert le script depuis le cache, sans vérifier que le fichier sur le disque correspond au cache.

```
%LOCALAPPDATA%\Edge\...\Service Worker\
|- Databases\        ← lien script .. Service Worker
|- ScriptCache\      ← stockage des scripts compilés
|- CacheStorage\     ← cache réseau (hors scope)
```

PERSISTANCE FURTIVE

...et donc ?

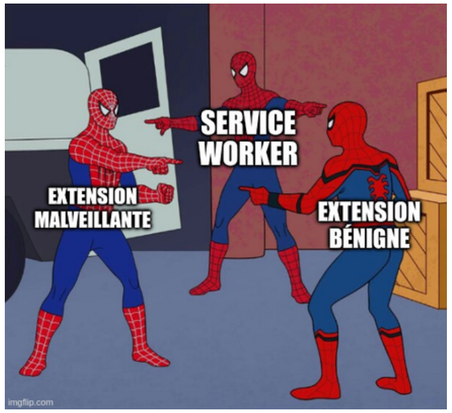
Si on remplace le dossier malveillant par un dossier vide avec le même ID et les mêmes noms de fichiers... le cache n'est jamais invalidé.
Désynchronisation entre le disque et le runtime.



On remplace le dossier malveillant par un bénin, le payload malveillant tourne quand même dans le cache.

PERSISTANCE FURTIVE

...la chaîne d'infection en 4 étapes



La double injection → vulnérabilité **GAP** (**G**host **A**nchor **P**ersistence).

Deploy malicious extension

stomp

Injection de l'extension malveillante via Secure Preferences + HMAC recalculé

Trigger registration

obligatoire

Ouvrir Edge → service worker enregistré en DB dans le profil + background.js compilé en ScriptCache

Stomp with benign extension

GAP

Remplacer l'extension malveillante par la bénigne (même ID, mêmes fichiers, contenu vide), ScriptCache non invalidé et chargement du chemin de déploiement pour cet ID.

Remove the malicious extension folder

fileless

Supprimer le dossier de l'extension malveillante. Zéro artefact malveillant sur le disque. Payload actif dans le cache.

Note: Le cache service worker (ScriptCache) ne s'update pas car :

- *Aucun signal HTTP Cache-Control car n'est pas refresh par le navigateur (automatique);*
- *Pas d'appel explicite unregister();*
- *Pas de version bump (updater officiel).*

Persiste indéfiniment

PERSISTANCE FURTIVE

...testé et validé sur Microsoft Edge, Chrome, Brave et Vivaldi sur Windows

Browser restart — persiste

Service worker restauré depuis le ScriptCache à chaque lancement.

System reboot — persiste

Registration DB + ScriptCache Survivent au redémarrage machine.

Browser update — persiste

Les mises à jour n'invalident pas le cache des extensions locales.

MDE / EDR scan — non détecté

Zéro alerte levée, dossier propre, payload invisible directement sur le disque.


```
// RÉPONSE MSRC - 11 AVRIL 2026
```

"This case has been assessed as not a vulnerability and does not meet Microsoft's bar for servicing. [...] it is not eligible for bounty, and no CVE will be issued."

Submission number : VULN-180712 · Case number : 112111

Les seuls IOCs /!\ :

- Fichier Secure Preferences modifié + dépôt dossier;
- Code du service worker en mémoire qui ne correspond pas à celui du dossier résolu par le fichier Secure Preferences.

→ https://github.com/Fir3n0x/GAP

// section demo

DEMO

...GAP, primitive de persistance furtive

MERCI!

...des questions ?

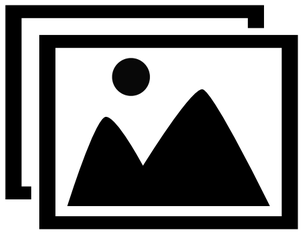
X / TWITTER	@fir3n0x_
GITHUB	github.com/Fir3n0x
BLOG	fir3n0x.github.io/
OUTILS	stomp - BELoader - GAP

TL;DR: Les extensions de navigateur sont utiles pour de la post-exploitation et GAP permet de les camoufler.

RÉFÉRENCES

documentation

→ https://www.synacktiv.com/publications/lextension-fantome-infiltrer-chrome-par-des-voies-inexplorees
→ https://www.cse.chalmers.se/~andrei/cans20.pdf
→ https://developer.chrome.com/docs/extensions/reference/api - chrome API
→ https://developer.chrome.com/docs/extensions/reference/manifest - manifest format
→ https://github.com/Fir3n0x/stomp
→ https://github.com/Fir3n0x/BELoader
→ https://github.com/Fir3n0x/GAP



images

→ https://imgflip.com
→ https://chatgpt.com
→ https://canva.com