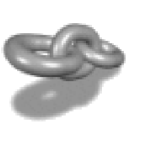
Se connecter
Proposer un contenu
Identifiant
Identifiant
Mot de passe
Mot de passe
☐ Connexion automatique
Se connecter
Pas de compte ? S'inscrire…

18 mar. 2026

52

J'ai mis un proxy entre claude et Internet

Posté par **tito** (**site web personnel**, **Mastodon**) le 18 mars 2026 à 11:12. Édité par **4 personnes**. Modéré par **patrick_g**. **Licence CC By-SA.**
Étiquettes : **intelligence_artificielle**, **proxy**



Je sais que le mot "IA" sur LinuxFr, c'est un peu comme prononcer "systemd" en 2015 ; ça ne laisse personne indifférent. Et je comprends. La merdification est réelle, la bulle est réelle, les externalités sont réelles. Je n'ai aucune envie d'en rajouter une couche. Mais voilà, les lignes sont devenues floues, et j'ai pris le virage du coding assisté. D'abord avec curiosité et prudence, et maintenant les deux pieds dans le plat : ça ne remplace pas ma façon de penser, mais ça m'a ouvert des portes : des concepts que je ne maîtrisais pas, des langages que je n'aurais pas pris le temps de toucher avant ; l'assistant me permet d'explorer, de comprendre, et de construire des outils qui m'aident. Et j'espère qu'ils aident d'autres personnes aussi.

Sauf que voilà. Au début, j'étais prudent. Je vérifiais chaque commande, chaque accès. Et puis petit à petit, j'ai lâché prise. J'ai désactivé les confirmations, laissé l'agent tourner sans supervision, accepté les permissions sans lire. On connaît tous ce moment où on clique "Allow" les yeux fermés parce que c'est la quinzième fois qu'il demande. J'ai fait exactement ce qu'on ne devrait jamais faire en sécurité : faire confiance par défaut.

Et un jour, je me suis dit : je n'ai aucune idée de ce que cet agent envoie sur le réseau. Aucune.

Alors j'ai construit un proxy un peu.. particulier.

Sommaire

Cher journal,

Ça fait un bail que je n'ai pas vraiment contribué à l'open source. Mes derniers vrais projets publics, c'était Kivy et les projets autour… ça remonte à quelques années maintenant, et j'ai pris ma "retraite" sur ces projets.

Mais je n'ai jamais arrêté de coder. J'ai juste réalisé un truc sur moi-même : le code, c'est un peu comme la musique pour moi. J'aime construire des choses. Je m'exprime mieux avec un éditeur et un terminal qu'avec ma voix ou mes mots. C'est probablement pour ça que je suis là à t'écrire un journal au lieu de faire un talk quelque part.

Le constat

On a passé des années à construire des pare-feux, des IDS, du monitoring pour nos serveurs de prod. Sur des entreprises plus grandes, on traque les connexions suspectes… Et puis un agent IA débarque sur notre machine de dev, on lui dit "tiens, refactore-moi ce module", et il fait ce qu'il veut sur le réseau sans qu'on le sache.

C'est quand même un peu absurde, non ?

Le truc, c'est qu'il n'existe pas vraiment d'équivalent à `tcpdump` ou `iptables` pour les agents IA sur nos machines. Pas de couche d'observabilité entre l'agent et Internet. Ou on contrôle, on se fait notre liste d'outils qu'on accepte, ou on fait confiance parce que bon, la sécurité, c'est pas si important… vraiment ?

Greywall et greyproxy

Avec l'équipe de Greyhaven, on a construit deux outils open source :

Greywall est un bac à sable deny-by-default pour les agents IA. Pas de Docker, pas de VM. Ça utilise directement les mécanismes du noyau Linux (namespaces, Landlock, Seccomp, eBPF) pour isoler le processus. Sur Linux, l'isolation réseau passe par un device TUN dans un namespace réseau dédié ; le processus sandboxé ne peut structurellement pas contourner le proxy. Sur macOS, c'est un peu moins élégant en utilisant des variables d'environnement pour forcer un proxy socks5h, si l'outil ne le supporte pas, il ne peut quand même pas sortir. Ça fait le job pour la plupart des outils.

Greyproxy est le plan de contrôle réseau. Un proxy SOCKS5/HTTP avec un dashboard web temps réel. Chaque connexion sortante de l'agent apparaît dans le dashboard. Si aucune règle ne matche, la connexion reste en attente et tu peux l'autoriser ou la refuser en direct, sans relancer la session.

Concrètement, ça donne :

```
greywall -- claude
```

Et hop, Claude Code tourne dans son bac à sable. Tu ouvres `http://localhost:43080` et tu vois en direct chaque domaine qu'il tente de contacter. Tu autorises `api.anthropic.com`, tu autorises `github.com` pour les pushes, tu refuses le reste. Tout est interactif, tout est visible.

Ce que j'ai observé

Au début, c'était juste des connexions supplémentaires. Tiens, c'est quoi ces appels à opencode.ai quand je démarre opencode ? Tiens, pourquoi Claude appelle 2x toutes les 4 minutes un domaine chez Google ? Entre de la télémétrie que l'on ne peut pas désactiver, ou des requêtes qui font "office" de regarder si une nouvelle version est disponible… 2x toutes les 4 minutes. Ce n'est pas le meilleur argument, mais contrairement aux autres sandboxes, au moins ici je le vois en temps réel, et je peux dire oui ou non sur ce que peut accéder la commande.

Le dashboard de greyproxy rend tout ça visible. Tu vois passer les requêtes DNS, les connexions TCP, les domaines contactés. Tu peux construire progressivement une liste d'autorisations adaptée à ton projet. Il y a même un mode apprentissage qui trace les accès filesystem avec `strace` et génère automatiquement un profil de sécurité.

Ce n'est pas un outil pour les paranos. C'est un outil pour ceux qui pensent que l'observabilité, c'est un droit, pas un luxe.

Pourquoi ça compte

Je sais que l'enthousiasme pour l'IA est réellement différent en fonction des gens. Les questions sur la qualité du code généré, la consommation énergétique, la centralisation chez les GAFAM ; tout ça est légitime.

Mais justement. Si on utilise ces outils (et beaucoup d'entre nous le font, même ceux qui restent prudents), autant le faire avec les yeux ouverts. Greywall, c'est pas un outil pour promouvoir l'usage des agents IA. C'est un outil pour que, si tu en utilises un, tu gardes le contrôle.

Il y a une phrase qu'on a mise sur le site et qui résume bien l'idée :

> "The security layer around your tools should be independent of the company selling you the AI."

La couche de sécurité autour de tes outils ne devrait pas dépendre de la boîte qui te vend l'IA. Claude a son propre sandbox intégré, Codex a le sien. Mais tu fais confiance aux entreprises pour te protéger d'elles-mêmes ? C'est un problème d'indépendance, pas de technologie.

Greywall est agnostique. Ça marche avec Claude Code, Codex, Cursor, Aider, Goose, Gemini CLI, Cline, et une dizaine d'autres. Tu changes d'agent, ta couche de sécurité reste la même.

Et après : vers un proxy sémantique

Le greyproxy actuel travaille au niveau des connexions : il voit les domaines, les ports, les IPs. Il ne déchiffre pas le TLS, il ne lit pas le contenu. C'est déjà très utile pour contrôler les accès réseau.

Mais là où ça devient vraiment intéressant, c'est quand on commence à reconstruire les conversations LLM qui passent par le proxy. Pas en cassant le chiffrement ; en instrumentant le flux côté client. L'idée, c'est de construire un proxy sémantique qui comprend ce que l'agent envoie et reçoit, qui peut faire du remplacement de variables d'environnement à la volée (pour ne jamais exposer tes vrais secrets à l'API du LLM), et qui te donne une vision complète de ce que l'IA fait en ton nom.

On en est au début, mais la direction est claire : remettre l'humain au milieu du système. Pas comme un goulot d'étranglement, mais comme un observateur informé qui peut intervenir quand c'est nécessaire. C'est ce qui manque cruellement à des systèmes comme OpenClaw et à la plupart des outils d'orchestration d'agents.

Pour essayer

Installation rapide :

```
# Homebrew
brew tap greyhavenhq/tap && brew install greyw

# Ou via curl (pas taper)
curl -fsSL https://raw.githubusercontent.com/
```

Ça tourne sur Linux et macOS. Sur Linux, il te faut `bubblewrap` et `socat` comme dépendances. Greyproxy s'installe comme service systemd si tu veux qu'il tourne en permanence.

Si tu veux comprendre les détails techniques de l'architecture (les 5 couches de sécurité, pourquoi on a abandonné Docker, comment fonctionne la capture réseau transparente), on a écrit un article technique détaillé ici : https://greyhaven.co/insights/why-we-built-our-own-sandboxing-sytem

La question

J'ai une vraie question pour la communauté. Ceux d'entre vous qui utilisent des agents IA pour coder (même occasionnellement, même à contrecœur) : comment vous gérez la sécurité ? Vous faites confiance par défaut ? Vous avez mis en place quelque chose ? Ou vous préférez ne pas y penser ?

Et pour ceux qui n'utilisent pas d'agents IA : est-ce que le manque de transparence et de contrôle fait partie des raisons ?

Ça m'intéresse vraiment de savoir :)

Aller plus loin

- 🇬🇧 Greywall (138 clics)
- 🇬🇧 Greyproxy (80 clics)
- 🇬🇧 Site web (108 clics)

(22 commentaires). Markdown EPUB

 # J'ai fait pareil ! Avec proxy MITM

Posté par **palkeo** (**site web personnel**) le 18 mars 2026 à 13:09. Évalué à 5 (+4/-0).

> Super de voir de l'intérêt autour de ça, j'ai eu exactement les mêmes questionnements et je me suis fait un outil perso.
> Dans mon cas je fais tourner les agents dans des VMs, et ils sont obligés de passer par un proxy qui déchiffre toutes les communications et relaie. Du coup je vois tout passer en clair, tout est loggé, et je peux écrire des règles pour laisser passer le traffic.
>
> J'en parle ici (en anglais): https://www.palkeo.com/en/blog/foreman.html

Répondre

[^] # Re: J'ai fait pareil ! Avec proxy MITM

Posté par **palkeo** (**site web personnel**) le 18 mars 2026 à 13:11. Évalué à 3 (+2/-0).

> Je sais pas à quel point bubblewrap/landlock marche bien ? Perso je trouve que c'est très puissant de permettre à un agent d'être root sur une VM pour faire ce qu'il veut.

Répondre

[^] # Re: J'ai fait pareil ! Avec proxy MITM

Posté par **Laurent J** (**site web personnel**, **Mastodon**) le 18 mars 2026 à 13:20. Évalué à 5 (+3/-0).

> très puissant ? très dangereux tu voulais dire, non ?
>
> Jamais je laisserai une IA, qui par définition fait des actions non déterministes, faire ce qu'elle veut sur mes machines.

Répondre

[^] # Re: J'ai fait pareil ! Avec proxy MITM

Posté par **palkeo** (**site web personnel**) le 18 mars 2026 à 13:26. Évalué à 4 (+3/-0).

> Si tu isole pas, c'est puissant et dangereux oui (c'est tout le sujet).
>
> Si c'est dans une VM qui à accès à rien, et niveau réseau elle peut juste télécharger des paquets Debian et des images docker, c'est puissant et pas dangereux.

Répondre

[^] # Re: J'ai fait pareil ! Avec proxy MITM

Posté par **tito** (**site web personnel**, **Mastodon**) le 18 mars 2026 à 17:08. Évalué à 3 (+2/-0).

> C'est aussi la limitation, quand on travaille dans des containers, on est oblige de tout réinstaller les outils. Et ça ne marche pas "bien" pour des outils graphiques. C'est pour ça que l'on adopte une position sandbox sans container. (On avait tente avec https://github.com/monadical-sas/cubbi avant, mais il n'y avait que des frictions)

Répondre

[^] # Re: J'ai fait pareil ! Avec proxy MITM

Posté par **tito** (**site web personnel**, **Mastodon**) le 18 mars 2026 à 17:06. Évalué à 3 (+2/-0).

> Ton projet est excellent, effectivement on a les mêmes questionnements, et les mêmes envies :)
> Je vais prendre le temps de tester ta solution, et serais ravis de discuter sur le sujet !

Répondre

 # J'ai pas besoin de faire pareil

Posté par **David Demelier** (**site web personnel**) le 18 mars 2026 à 13:36. Évalué à 1 (+6/-7).

> Parce que je code depuis 2008 et que depuis 2008 je lis les pages de manuel, les forums et les code opensource. Résultat : j'ai capitalisé des connaissances qui me servent encore aujourd'hui.
>
> Quand on me demande quelque chose dans mon travail, il est pas rare que j'ai la réponse rapidement. C'est pas de la science infuse, c'est de l'investissement.
>
> --
>
> AI is a mental disorder

Répondre

[^] # Re: J'ai pas besoin de faire pareil

Posté par **leyouki** (**site web personnel**, **Mastodon**) le 19 mars 2026 à 09:16. Évalué à 3 (+2/-0).

> Rappeler la qualité de l'intelligence humaine et la capacité de chacun·e d'apprendre continuellement et s'adapter, ça fait pas de mal.
>
> Est-ce nécessaire de mépriser les compétences humaines pour échanger sur celles d'un outil qu'on nous a imposé? Comme si rappeler que nous sommes vivants était un affront à l'avancée technologique.

Répondre

[^] # Re: J'ai pas besoin de faire pareil

Posté par **David Demelier** (**site web personnel**) le 19 mars 2026 à 09:27. Évalué à -1 (+1/-4).

 # wsl

Posté par **Nicolas Boulay** (**site web personnel**) le 18 mars 2026 à 14:43. Évalué à 5 (+2/-0).

> Au boulot, on a des machines MS, avec la killerfeature WSL 2.0. On utilise VS code qui utilise un terminal pour copilot. J'ai juste remplacé le terminal par défaut de copilot pour un WSL VM Debian. Il n'a que le répertoire en court de monter, le reste est exclus.
>
> Ainsi, Claude ne peut voir que le répertoire de code, et est root dans son environnement, il peut installer ainsi ce qu'il veut pour travailler (python, ripgrep, …).
>
> --
>
> "La première sécurité est la liberté"

Répondre

[^] # Re: wsl

Posté par **tito** (**site web personnel**, **Mastodon**) le 18 mars 2026 à 21:56. Évalué à 2 (+1/-0).

> Il y a un de l'équipe qui est en train de faire le support pour WSL2.
> Disons que oui ton approche marche, mais seulement pour les fichiers. Ici tu peux également dire oui/non pour les requêtes réseaux, sans redémarrer ton agent.

Répondre

[^] # Re: wsl

Posté par **Nicolas Boulay** (**site web personnel**) le 19 mars 2026 à 10:36. Évalué à 3 (+0/-0).

> Disons que j'ai fais cette VM justement pour ne pas avoir à accepter commande par commande. Cela serait bien si ton projet venait avec une black list associé (ping vers google, et puis ?)
>
> --
>
> "La première sécurité est la liberté"

Répondre

[^] # Re: wsl

Posté par **bunam** le 19 mars 2026 à 09:11. Évalué à 2 (+1/-0).
Dernière modification le 19 mars 2026 à 09:11.

> Du coup il faut démonter : /mnt/c ?

Répondre

[^] # Re: wsl

Posté par **Nicolas Boulay** (**site web personnel**) le 19 mars 2026 à 10:30. Évalué à 3 (+0/-0).

> oui, il y a un moyen assez simple de le faire.
>
> --
>
> "La première sécurité est la liberté"

Répondre

Et donc ...

Posté par **rycks** le 18 mars 2026 à 15:30. Évalué à 10 (+13/-0).
Dernière modification le 18 mars 2026 à 15:30.

> Beau boulot mais je reste sur ma faim
>
> En bref claude cause quoi avec claudette ou mégaclaude ou xclaude ou claude-command-center-usa-first ? nous on voulait savoir ce qu'ils se racontent ...
>
> Quelles sont les données exfiltrées ?
>
> Si t'a essayé de lui laisser par inadvertance un fichier "pot de miel" (genre des login & passwords avec des url de sites que tu héberges toi même et pour lesquels tu vas recevoir une alerte lorsqu'une connexion sera faite) dans la liste de ce qu'il aurait pu consulter pour voir s'il farfouille etc.
>
> C'est bien de mettre un proxy tu nous as mis l'eau à la bouche mais maintenant commence le travail du journaliste d'investigation hop au boulot nous on veut savoir :)
>
> --
>
> eric.linuxfr@sud-ouest.org

Répondre

[^] # Re: Et donc ...

Posté par **tito** (**site web personnel**, **Mastodon**) le 18 mars 2026 à 17:12. Évalué à 9 (+8/-0).

> C'est vrai, tu as raison. Je me suis amuse de faire un Capture the flag dans l'environnent restreint, mais je n'ai rien encore écris a ce propos. Sur le sujet de l'exfiltration de donnée, c'est exactement un des points que je veux ajouter.
>
> On reviendra un peu plus tard avec des belles histoires!

Répondre

si je comprends bien...

Posté par **Psychofox** (**Mastodon**) le 18 mars 2026 à 15:47. Évalué à 5 (+2/-0).

> ...tu as développé un outil comparable à Little Snitch mais pour Linux?

Répondre

[^] # Re: si je comprends bien...

Posté par **bunam** le 19 mars 2026 à 08:51. Évalué à 2 (+1/-0).

> PI (pour ceux qui ont la flemme de chercher) :
>
> https://www.obdev.at/products/littlesnitch/index.html

Répondre

opensnitch

Posté par **RD** le 18 mars 2026 à 21:30. Évalué à 3 (+4/-1).

> 1. OpenSnitch https://github.com/evilsocket/
> 2. Utiliser un client libre connecté à l'API plutôt qu'un client propriétaire
> 3. AppArmor (en plus de la sandbox du client)

Répondre

"Et pour ceux qui n'utilisent pas d'agents IA : est-ce que le manque de transparence et de contrôle fait partie des raisons ?"

Posté par **Stéphane Ascoët** (**site web personnel**) le 19 mars 2026 à 10:57. Évalué à 1 (+0/-0).

> Même réponse que la première intervention que David D. Par contre contrairement à lui, je ne rejette pas le fait que tu aies fait ce projet, et, comme d'autres, j'aurais aimé avoir plus de détails croustillants sur ce que tu as bloqué.
>
> Pour information, un des soirs de cette semaine, le débat était le même dans "le téléphone sonne" de France Inter. Comme d'habitude, énervant et désespérant servage de soupe au capitalisme et notre société de merde sans **aucun** point de vue contraire (à part que Fabienne Saintesse reconnaît être une cruche en informatique, super).
>
> Dommage que RD n'ait pas mis son commentaire dans le fil de celui de Psychofox(snitch)

Répondre

Proxies

Posté par **høm** le 19 mars 2026 à 12:06. Évalué à 1 (+1/-0).

> Projet intéressant. Il faudra que je teste un de ces jours.
>
> Je voulais justement implémenter *isolate* https://github.com/jgaines/isolation un script dédié à ce type d'outils qui s'appuie aussi sur *bubblewrap*, mais se concentre sur les autorisations d'accès au système de fichiers (lecture/écriture, lecture seule, etc.). Malheureusement, je crains qu'*isolate* et *Greywall/Greyproxy* ne fonctionneront pas facilement ensemble, je me trompe ?
>
> Pendant que j'y suis, il y a aussi *rtk* https://github.com/rtk-ai/rtk qui a retenu mon attention. Il s'agit encore d'un "proxy", mais à l'objectif bien différent puisqu'il permet de réduire le nombre de token consommés par les agents.
>
> Permettre à tous ces outils de bien fonctionner ensemble serait incroyable.

Répondre

[^] # Re: Proxies

Posté par **tito** (**site web personnel**, **Mastodon**) le 19 mars 2026 à 14:54. Évalué à 1 (+0/-0).

> Greywall utilise Bubblewrap... donc oui, ils sont incompatibles : on ne peut pas faire du Bubblewrap dans du Bubblewrap 🙂 Mais si tu regardes du côté des profils Greywall, ce sont des profils pour l'accès au système de fichiers (allowRead, allowWrite, denyRead, denyWrite), avec une approche de tout refuser par défaut. Et j'ai tenté d'ajouter un --learning qui permet de capturer tous les accès aux fichiers pour un run, afin de déterminer les règles nécessaires.
>
> Par contre, Greyproxy, tu pourrais le rajouter dans ton script. Configure HTTP_PROXY/ALL_PROXY pour pointer sur Greyproxy, pareil pour le DNS, tun2socks avec iptables si tu veux capturer le trafic qui ne passe pas par le proxy... et tu as un équivalent 1:1 de Greywall.

Répondre

Envoyer un commentaire

Suivre le flux des commentaires

Note : les commentaires appartiennent à celles et ceux qui les ont postés. Nous n'en sommes pas responsables.

Revenir en haut de page

Derniers commentaires

- Je n'avais pas vu le t…
- Zig ≠ Rust
- Re: ça va arriver de …
- Re: Mouais
- suivi AIS
- je suis vieux, alors...
- Re: Je dois avouer u…
- Re: Mouais
- Re: ça va arriver de …
- Re: grep sait le faire
- Sources
- Re: Quel ramasse-m…

Étiquettes (tags) populaires

- intelligence_artificielle
- merdification
- grands_modèles_de…
- états-unis
- sortie_version
- administration_fran…
- politique
- google
- bigtech
- capitalisme
- capitalisme_de_surv…
- android

Sites amis

- April
- Agenda du Libre
- Framasoft
- Éditions D-BookeR
- Éditions Eyrolles
- Éditions Diamond
- Éditions ENI
- La Quadrature du Net
- Lea-Linux
- En Vente Libre
- Grafik Plus
- Open Source Initiative

À propos de LinuxFr.org

- Mentions légales
- Faire un don
- L'équipe de LinuxFr.…
- Informations sur le s…
- Aide / Foire aux que…
- Suivi des suggestion…
- Wiki du site
- Règles de modération
- Statistiques
- API pour le dévelop…
- Code source du site
- Plan du site