

# 21 degrés de liberté – 20

*Aujourd'hui nos employeurs peuvent intercepter nos communications passées sur notre lieu de travail, une nouvelle atteinte à la vie privée que n'auraient pas admise nos parents et dont nos enfants sont victimes.*

*Voici déjà le 20<sup>e</sup> article de la série écrite par [Rick Falkvinge](#). Le fondateur du [Parti Pirate suédois](#) aborde ici le droit légal des employeurs à prendre connaissance de nos messages quand ils sont passés dans l'entreprise.*

*Le fil directeur de la série de ces 21 articles, comme on peut le voir clairement dans les [épisodes précédents](#) que nous vous avons déjà livrés, c'est la perte de certaines libertés dont nous disposions encore assez récemment, avant que le passage au tout-numérique ne nous en prive.*

**Votre patron ne pouvait pas lire votre courrier. Jamais.**

Source : Rick Falkvinge sur [privateinternetaccess.com](http://privateinternetaccess.com)

Traduction Fralang : dodosan, Susyl, goofy.

**Slack vient tout juste de mettre à jour ses conditions générales d'utilisation pour permettre à votre employeur de lire vos conversations privées sur des canaux privés. Nos parents auraient été choqués et horrifiés à l'idée que leurs chefs ouvrent les colis et lisent les messages personnels qui leur étaient adressés. Pour nos enfants dans le monde numérique, cela fait partie de la vie de tous les jours et ne mérite qu'un haussement d'épaule.**



Photo par [philcampbell](#) (CC-BY 2.0)

Le bon vieux système téléphonique, parfois appelé par son abréviation anglaise POTS, est un bon exemple de la façon dont les choses devraient se passer, même dans le monde numérique. Les législateurs avaient vu juste dans l'ensemble à ce sujet.

Lorsque quelqu'un passe un appel téléphonique – un appel à l'ancienne, analogique – on sait que la conversation est privée par défaut. Peu importe à qui appartient le téléphone. C'est la personne qui l'utilise, à cet instant précis, qui a tous les droits sur ses capacités de communication à l'instant T.

L'utilisateur a tous les droits d'utilisation. Le propriétaire n'a aucun droit d'intercepter les communications ou d'interférer avec elles sur la seule base du droit de propriété.

Autrement dit, être propriétaire d'un outil de communication ne donne pas automatiquement le droit d'écouter les conversations privées passant par cet équipement.

Malheureusement, cela ne s'applique qu'au réseau téléphonique. Qui plus est, uniquement à la partie analogique du réseau téléphonique. Si quelque chose est même de loin numérique, le propriétaire peut intercepter pratiquement tout ce qu'il veut, pour n'importe quelle raison.

Cela s'applique particulièrement au lieu de travail. On pourrait soutenir qu'on n'attend aucun respect de la vie privée quand on utilise l'équipement de son employeur. Cela revient précisément à oublier qu'une telle intimité était primordiale pour les POTS, il y a moins de deux décennies, quel que soit le propriétaire de l'équipement.

Certains employeurs installent même des certificats numériques [wildcard<sup>1</sup>](#) sur les ordinateurs de l'espace de travail, dans l'objectif bien particulier de contourner la sécurité de bout en bout entre les ordinateurs des salariés et le monde extérieur, effectuant ainsi une [attaque dite « homme du milieu »](#). En termes politiquement corrects, cette pratique est appelée « interception HTTPS<sup>2</sup> » et non « attaque de l'homme du milieu » quand elle est menée par votre employeur et non par un autre attaquant.

Puisque nous en sommes à comparer analogique et numérique et la façon dont les droits à la vie privée se sont évaporés en passant d'une époque à l'autre, il est intéressant de jeter un coup d'œil aux lois qui régissaient un fort ancien moyen de communication, la correspondance postale. Demandez-vous si votre patron pouvait ouvrir votre courrier simplement parce qu'il vous était adressé sur votre lieu de travail.

Les lois sont un peu différentes selon les pays sur ce point, mais en général, même si votre patron ou entreprise étaient autorisés à ouvrir votre correspondance (c'est le cas [aux USA](#) mais non [en Angleterre](#)), ils n'étaient en général *jamais* autorisés à la lire ([même aux USA](#)) ;

Tout au contraire, pour le courrier électronique, vos employeurs ne se contentent pas de lire la totalité de vos courriels, mais ont souvent [engagé une équipe entière](#) pour le faire. En Europe, la chose est allée jusqu'à la Cour Européenne des Droits de l'Homme, qui a statué qu'il est tout à fait normal pour un employeur de lire la correspondance la

plus privée, pourvu qu'il en informe les employés (ce qui piétine au passage l'espoir d'une confidentialité par défaut)

Il va de soi que ce principe qui s'applique aux courriels maintenant un peu démodés s'applique désormais aussi à tous les moyens de communication d'aujourd'hui, tels que Slack.

De sorte que pour nos enfants de l'ère numérique, l'idée suivant laquelle « le courrier c'est privé et il vous appartient, peu importe si vous le recevez au travail » semble définitivement oubliée. Encore un principe que nos aînés de l'époque analogique tenaient pour acquis, et pour lequel ils n'ont pas cru nécessaire de combattre.

La vie privée demeure de votre responsabilité.

---

## **Notre gitlab évolue en Framagit. C'est très efficace !**

*Warning : cet article parle de forge logicielle qui sert à développer collaborativement du code. Il est donc un peu velu et technique, mais il fera plaisir aux plus « barbu-e-s » d'entre vous !*

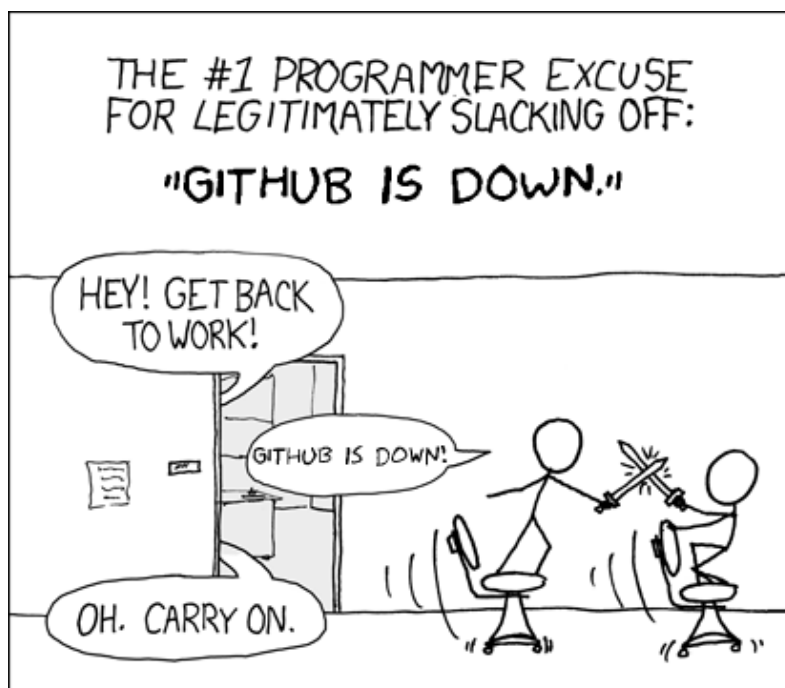
Préviousselaid, chez Framasoft : nous avons besoin d'une forge logicielle comme outil interne à l'asso... parce que même si nous ne développons pas (ou exceptionnellement) de logiciel libre ; les mettre en avant, les améliorer (parfois), les promouvoir et ouvrir des services au monde, ben ça demande de créer, maintenir, échanger et améliorer du code !

Nous nous étions donc installé [Gitlab](#) à la main, sur un coin de serveur, juste pour nous... Étant les seuls utilisateurs, on s'est dit que ce ne serait pas grave s'il n'était pas toujours à jour, à traquer la dernière version... (oui : nous sommes moins exigeants sur nos outils internes que pour les services que nous ouvrons au grand public ^^).

## Franchement, merci Google !

Merci, parce qu'à chaque fois que vous prenez des décisions unilatérales aux dépens de vos utilisateurs-produits, vous nous offrez l'occasion de prouver que le Libre offre des alternatives bien plus respectueuses des personnes qui vous ont confié leur vie numérique (et leur code).

Le jour où nous avons appris que [Google Code fermait ses portes, nous avons donc décidé d'ouvrir les nôtres](#). Cela nous a aussi permis de sensibiliser au fait que, dans le mode des codeurs et développeuses, [GitHub est devenu un point central et monopolistique assez inquiétant](#).



*L'excuse n°1 des programmeurs pour se lâcher sans scrupules :*  
« *GitHub est en panne* »

*– Hé, au boulot les gars ! – Github  
est en panne !  
– Ah bon, continuez alors.*

Forcément, l'ouverture à tous de notre *git* et les nouvelles fonctionnalités des nouvelles versions de Gitlab (une nouvelle version tous les 22 du mois) nous ont incités à mettre à jour plus régulièrement, ce qui prend plusieurs heures à chaque fois... et plusieurs fois par mois, car des versions correctives sont régulièrement publiées.

## **Améliorer le Framagit... une priorité**

Ceci, ajouté à l'utilisation grandissante de notre forge qui allait bientôt poser des problèmes de taille de disques, nous a amenés à migrer (le 17 mars dernier) notre Gitlab vers une machine avec plus de disque et surtout avec une installation utilisant les paquets dits « omnibus ».

Ces paquets omnibus nous ont permis d'installer Gitlab à l'aide d'un simple `apt-get install gitlab-ce` plutôt que de suivre la longue procédure d'installation manuelle. Non seulement l'installation est simplifiée, mais – et c'est surtout là la plus-value que nous en attendions – mettre à jour Gitlab devient tout aussi simple avec une seule commande `apt-get dist-upgrade`.

Résultat : notre Gitlab suit scrupuleusement la publication des nouvelles versions, avec leur lot de nouvelles fonctionnalités !

Alors, il paraît  
que tu es passé  
de GitHub à GitLab ?



Pour fêter cela, nous avons étrenné un nouveau nom de domaine... inspiré par vous ! Avouons-le, «Git point Framasoft point orrrrrrrrueh », ça accroche un peu en bouche. De partout, nous avons entendu parler du « [Framagit](#) » : alors tant qu'à faire, autant l'appeler comme vous le faites déjà. Bien entendu, il n'est nul besoin de modifier vos URL, elles restent valides... mais la nouvelle est à votre disposition !

## Et si on ajoutait de l'intégration continue ?

Derrière ce terme barbare se cache une fonctionnalité très pratique : on crée une « recette » qui sera exécutée dans une machine virtuelle à chaque *push*. Cela peut par exemple permettre de lancer une suite de tests pour vérifier que l'on n'a rien cassé. □

Pour utiliser cette fonctionnalité, il faut disposer de ce que l'on appelle un *runner*, c'est à dire un [logiciel](#) qui va récupérer la recette et l'exécuter. Il est possible

d'installer un *runner* sur n'importe quel ordinateur, même votre ordinateur de bureau.

Pour ceux qui ne souhaitent pas gérer leur *runner* eux-mêmes, Framasoft met à disposition deux *runners* partagés entre tous les utilisateurs de Framagit, que vous pouvez utiliser comme bon vous semble. Notez toutefois que Gitlab indique que quiconque utilise un *runner* partagé peut accéder aux informations des projets utilisant ce *runner* : il vaut mieux monter votre propre *runner* pour vos projets sensibles.

De plus, en utilisant les *runners* partagés de Framasoft, il est possible que votre projet soit mis en file d'attente, en attendant que les recettes précédentes aient fini de s'exécuter... à vous de voir !





Pouhiou-le-moldu-du-code lisant cet article,  
allégorie.

## **Vous voulez des pages Gitlab ? Nous aussi !**

Github permet à tout un chacun d'héberger un site statique. Gitlab propose une fonctionnalité similaire mais hélas, uniquement dans sa version entreprise... Nous utilisons pour notre part la version communautaire qui est la version libre

de Gitlab... donc *sans* les pages Gitlab.

Nous avons donc ouvert un ticket pour demander que cette fonctionnalité soit incluse dans la version communautaire. Si vous aussi vous aimeriez voir cela arriver, aidez-nous tout simplement en votant sur <https://gitlab.com/gitlab-org/gitlab-ce/issues/14605>.

**En attendant, profitez d'une forge logicielle à jour et libre sur [Framagit.org](http://Framagit.org) !**

***Mise à jour du 5/08/2016 :***

*Le tutoriel d'installation de Gitlab est -enfin- disponible [sur le FramacLOUD](#).*

*Notez que cette installation est conjointe à celle de Mattermost (Framateam) puisque c'est ainsi que nous avons procédé ☐*