Le SSH Tunneling

Arthur Naullet

3 juin 2021

Table des matières

1 Introduction

Le ssh tunneling est une technique qui permet à travers une connexion ssh de créer un tunnel pour y faire passer des flux tcp chiffrés. Cette technique a différentes applications. Elle peut servir à : chiffrer un flux, exfiltrer des données, bypasser un proxy, bypasser un firewall. Après une revue du fonctionnement et des applications théoriques du tunnel, nous verrons une application concrète.

2 Rappels sur SSH

Faisons un rappel sur le protocole SSH. Secure Shell (SSH) est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite, tous les segments TCP sont authentifiés et chiffrés. Il devient donc impossible d'utiliser un sniffer (logiciel qui permet de récupérer les informations qui circulent sur le réseau) pour voir ce que fait l'utilisateur. - wikipedia

Le protocole permet donc une connexion chiffrée avec un serveur distant. Mais ce protocole permet bien d'autres choses, comme le tunneling...

3 Multiples Applications du tunneling

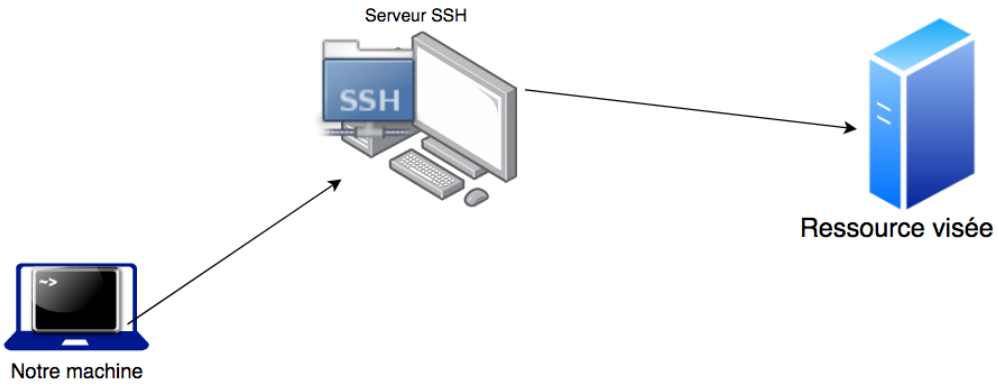


FIGURE 1 – Tunnel SSH

3.1 Qu'est-ce qu'un tunnel

Un tunnel est un genre de "trou" qui va permettre d'envoyer tous flux d'un point A à un point B en passant par un point C. Ici, le point A sera l'utilisateur, le point B un serveur SSH et le point C la ressource/le serveur que l'on souhaite atteindre.

De manière plus précise, le server SSH va jouer le rôle de relay pour le flux TCP entrant et le rediriger vers un autre server.

Nb : Conformément au protocole SSH, le paquet entrant est toujours chiffré. Le paquet sortant sera lui desencapsulé et diriger tel quel.

3.2 Utilisation de la connexion chiffrée

Du fait qu'un tunnel SSH chiffre le flux qui le traverse, on peut imaginer un nombre d'applications assez conséquent. On pourrait prendre pour exemple l'accès à un site web n'implémentant pas l'https, et bien passer par un tunnel SSH permettrait de sécuriser tout de même cette connexion. HTTP n'est qu'un exemple, mais tout protocole non chiffré est valable (HTTP, FTP, IPP etc...).

3.3 Exfiltration de données

L'exemple d'un attaquant ayant infiltré une entreprise et cherchant a exfiltré les données de l'entreprise est très concret. Grâce au tunnel SSH, il pourra faire passer toutes les données qu'il souhaite sur un serveur a l'extérieur de l'entreprise. Exemple, un serveur ftp (C) permettra de recueillir les données du Poste Administrateur (A) à travers la connexion SSH d'une machine (B) elle aussi a l'extérieur de l'entreprise.

3.4 Contournement de protections

Il y a deux exemples assez concrets. Le premier est le contournement d'un firewall qui laisserait passer par exemple uniquement les requêtes voulant atteindre les ports 80, 443, 53 donc http, https, et dns. Dans notre cas, il suffirait de faire tourner un serveur SSH sur le port 80 ou 443 a l'extérieur du réseau ou il y a le firewall et de ce fait, on pourrait utiliser un serveur tiers ne tournant pas sur l'un de ces 3 ports.

Le deuxième exemple, c'est le by-pass de vérification d'un proxy web. En effet si un proxy web refuse les connections a certains sites indiqués dans une base de données, notre serveur SSH n'étant pas dans cette base de données, on pourrait requêter grâce à lui via un tunnel des ressources normalement impossible d'accès.

Ces 2 applications seront présentes dans la partie démonstration pour une explication plus concrète.

4 Fonctionnement du tunnel

4.1 Théorie

D'abord, le paquet est enveloppé dans un en-tête supplémentaire (additional header), c'est ce qu'on appelle l'encapsulation. Cet en-tête supplémentaire contient les informations de routage nécessaires pour envoyer le paquet encapsulé à travers l'inter-réseau intermédiaire. Cette information est très importante car les données utiles sont envoyées dans un protocole différent de celui dans lequel les données ont été créées. Ensuite, un tunnel, qui est un chemin interconnectant 2 points, est créé et le paquet est transmis au point final. À ce moment, le paquet est désencapsulé pour "retrouver sa forme initiale". - wikipedia

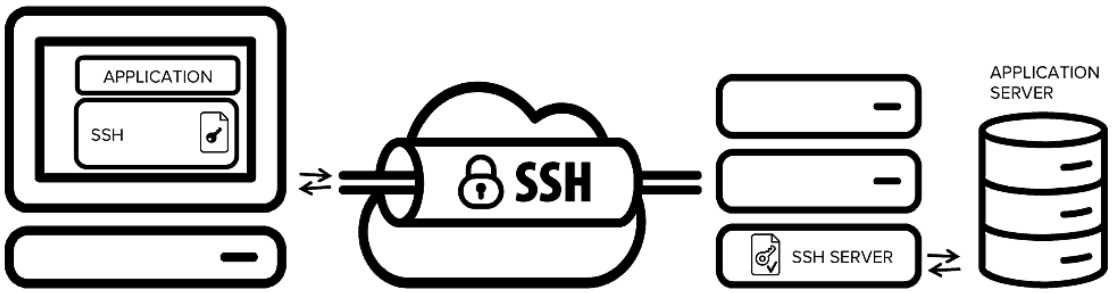


FIGURE 2 – Encapsulation(connexion chiffrée) puis désencapsulation - paquet d'origine

4.2 En pratique

Alors comment créer un tunnel ?

Le programme SSH nous offre deux méthodes pour celà :

4.2.1 Tunnel simple

La première méthode est pratique lorsqu'on cherche a atteindre une ressource spécifique SSH s'utilise ainsi :

```
1 ssh -L localhost:<port local>:<IP du serveur cible>:<port du serveur
    cible> <utilisateur>@<serveur SSH>
```

Aisi tant que "utilisateur" maintiendra la connexion SSH, il aura accès sur `localhost:<Port Local>` la ressource qu'il souhaite atteindre. Le problème est le suivant, dès qu'il voudra atteindre une autre ressource sur un autre serveur par exemple, alors il devra retaper toutes la commandes avec les bonnes modifications. C'est pourquoi il existe une autre méthode.

4.2.2 Mode dynamique

La méthode dynamique va créer un "trou" qui va envoyer tout ce qui est requêté à travers le tunnel SSH, seule condition supplémentaire, il faudra mettre en place un proxy. Elle s'utilise ainsi :

```
1 ssh -D <port local> <utilisateur>@<serveurSSH>
```

5 Cas concret

5.1 Scénario

Voici le scénario, Alice et Bob sont dans le CDI de leur lycée, ils voudraient tous les deux voir la diffusion du lancement de la dernière fusée de spaceX disponible sur `spaceX.com`. Malheusement Bob essaye de se connecter et impossible il a un message d'erreur qui indique "Interdit : le site que vous, essayé d'atteindre, vous a été refusé par le proxy de l'établissement". En effet l'administrateur réseau a estimé (pour X raison) que `spaceX.com` était prohibé. Il a donc rajouté le nom de domaine et l'ip de spaceX à la base donnée des ressources black listées. C'est pourquoi le web proxy refuse l'accès a Bob. On nottera également que le lycée possède un firewall. Ce firewall ne laisse que passer les requêtes à destination de ports 80(http), 443(https) et 53(dns) On peut résumer la situation avec le schéma suivant :

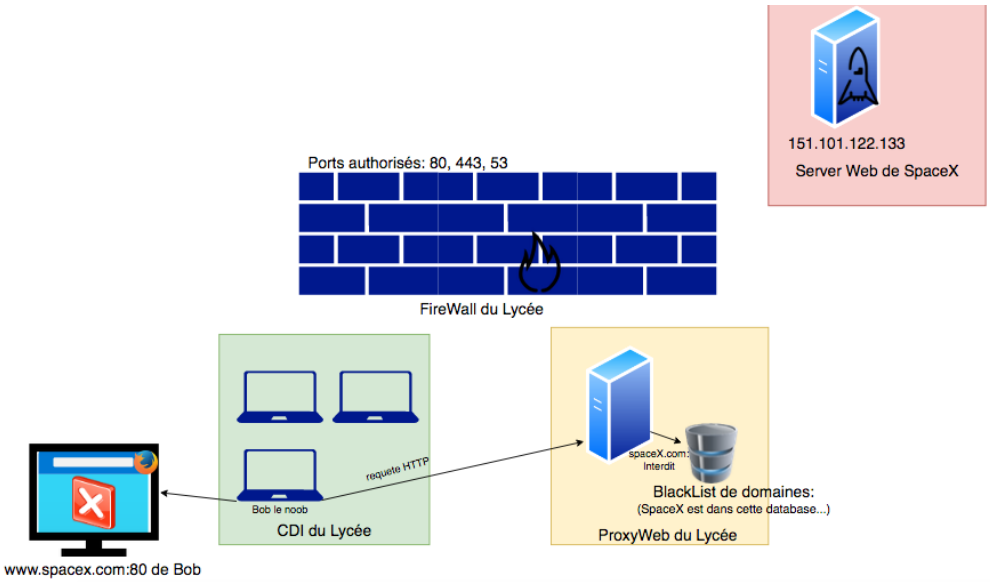


FIGURE 3 – Bob essaye de requeter `spaceX.com` mais l'accès est refusé

5.2 Tunnel spécifique

Alice, un peu plus débrouillarde que Bob décide d'utiliser un tunnel ssh pour pouvoir accéder a `spaceX.com` Elle ouvre alors un terminal et tape la commande suivante :

```
1 ssh -L localhost:1234:151.101.122.133:443 alice@mysuperserver.fr -p 80
```

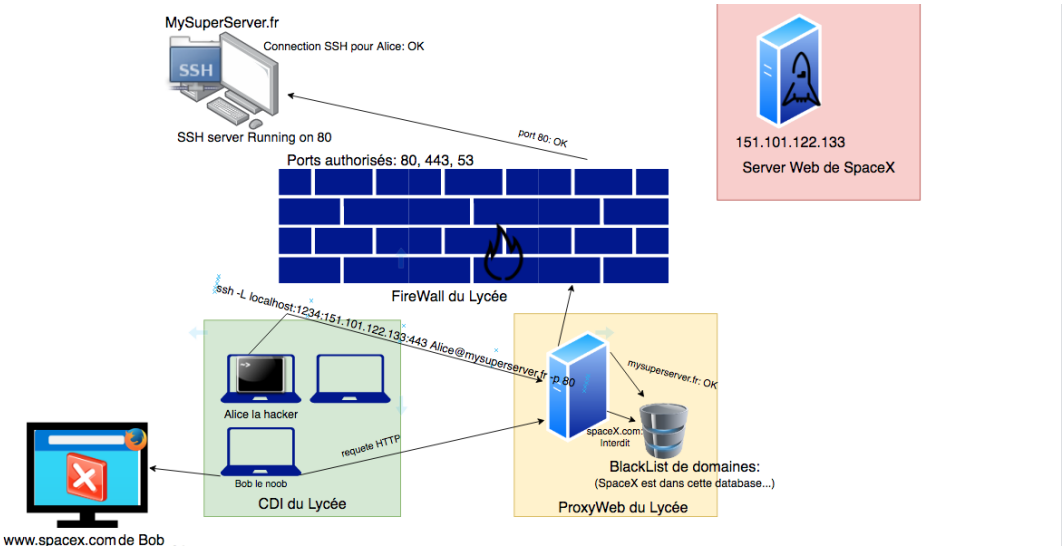


FIGURE 4 – Connexion SSH d'Alice avec -L

Le proxy ne refusera pas la connexion, car `mysuperserver.fr` n'est pas dans sa blacklist.

De plus, le firewall voyant une connexion vers un port 80 ne posera pas de problème lui non plus.

Alice est donc maintenant connectée en SSH.

En se connectant sur son navigateur et en entrant l'url `https://localhost:1234` Alice peut maintenant voir la page de `spaceX.com`.

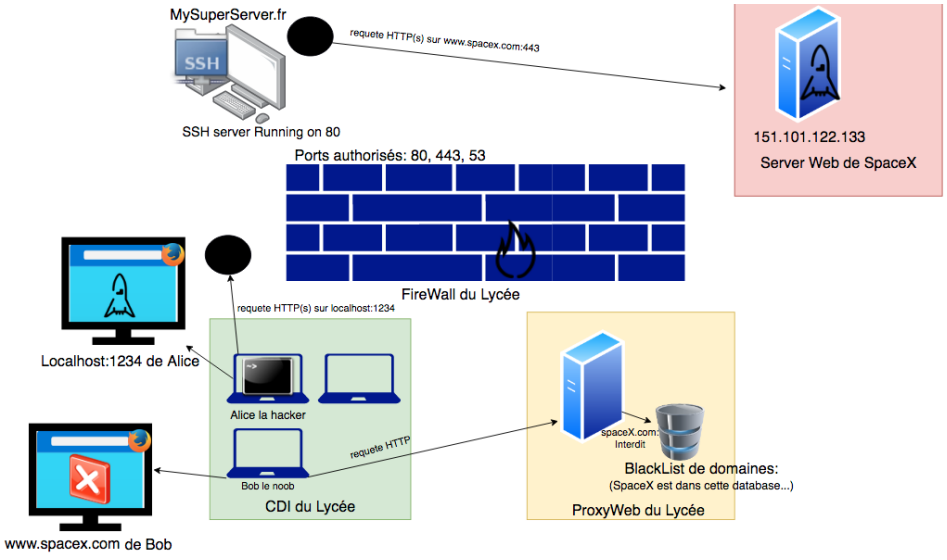


FIGURE 5 – Alice a accès a `spaceX.com` via localhost :1234

5.2.1 Possibles problèmes

Si spaceX utilise un serveur http qui implémente le virtual hosting, Alice risque de voir apparaître des erreurs HTTP du a un VHost inconnu. IL faut alors qu'elle précise a son navigateur de changer le header "`Host`" de la requête http en "`www.spaceX.com`". En effet le header "`Host`" d'Alice pour le moment est `localost:1234`. SpaceX ne connaît pas ce virtual host. En le changeant manuellement, elle pourra alors avoir accès à la ressource souhaitée.

5.3 Tunnel dynamique

Alice a également la possibilité d'utiliser le tunnel SSH en mode dynamique avec la commande suivante :

```
1 ssh -D 1234 alice@mysuperserver.fr -p 80
```

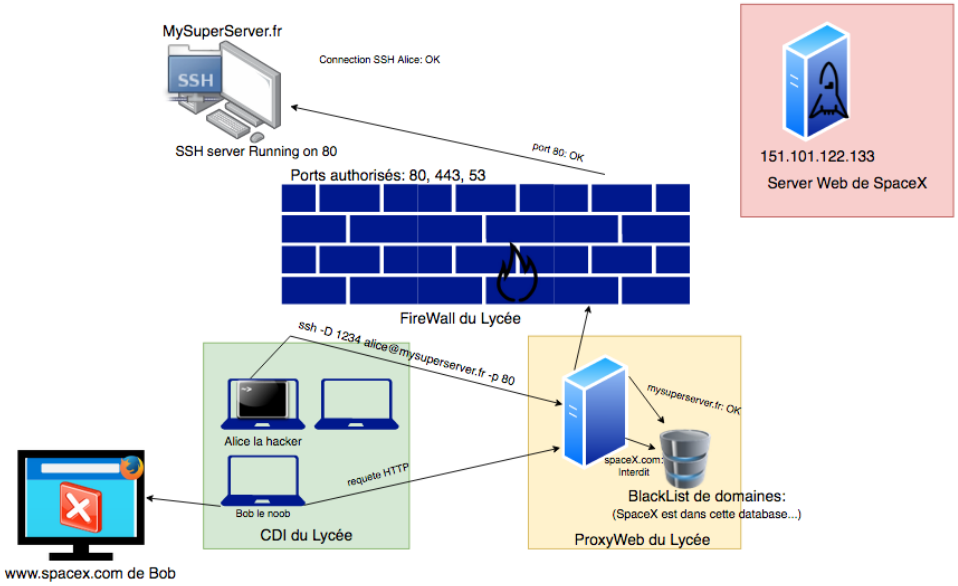


FIGURE 6 – Connexion SSH d'Alice avec -D

Cependant, avec cette méthode Alice devra paramétrer un proxy dans son navigateur web pour pouvoir lui indiquer d'utiliser le tunnel.

le type de proxy sera socks4 ou socks5 (on peut y faire traverser tout type de protocole) et l'adresse `localhost:1234`.

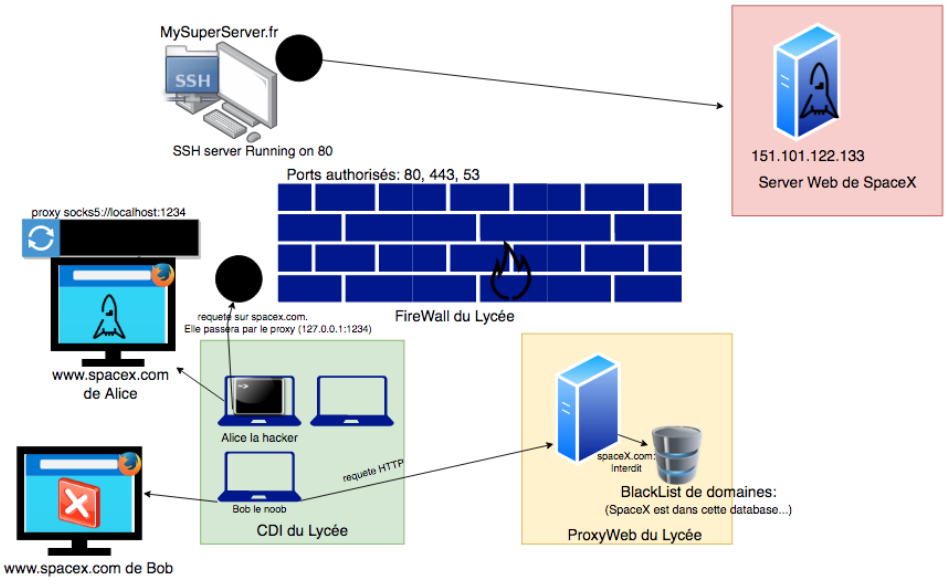


FIGURE 7 – Alice a accès a `spaceX.com` avec set up d'un proxy

6 Démonstration

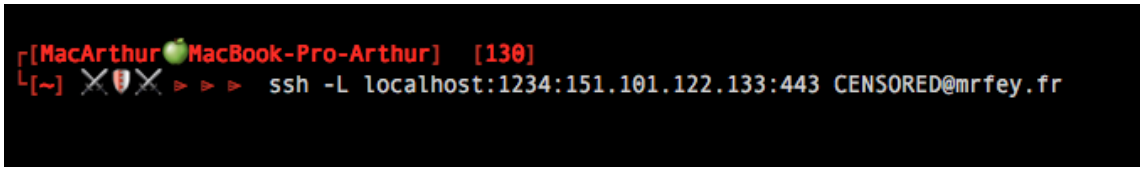
Je vais ici vous montrer un tunnel que j'ai monté. Mon serveur ssh tourne sur mrfey.fr. Il n'y a pas de web proxy ni de firewall dans mon réseau local, mais on peut quand même voir comment cela s'utilise.

Comme Alice, on va accéder à spaceX.com via notre tunnel SSH.

6.1 Tunnel spécifique

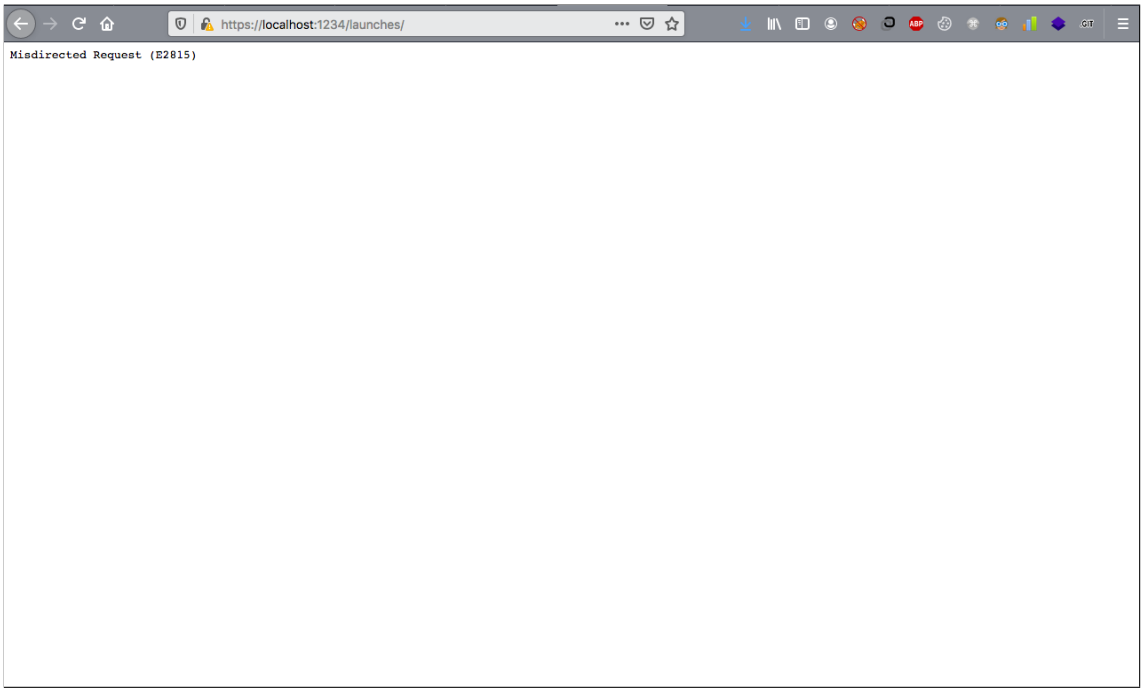
6.1.1 Connexion SSH

Je me connecte en SSH à mon serveur.



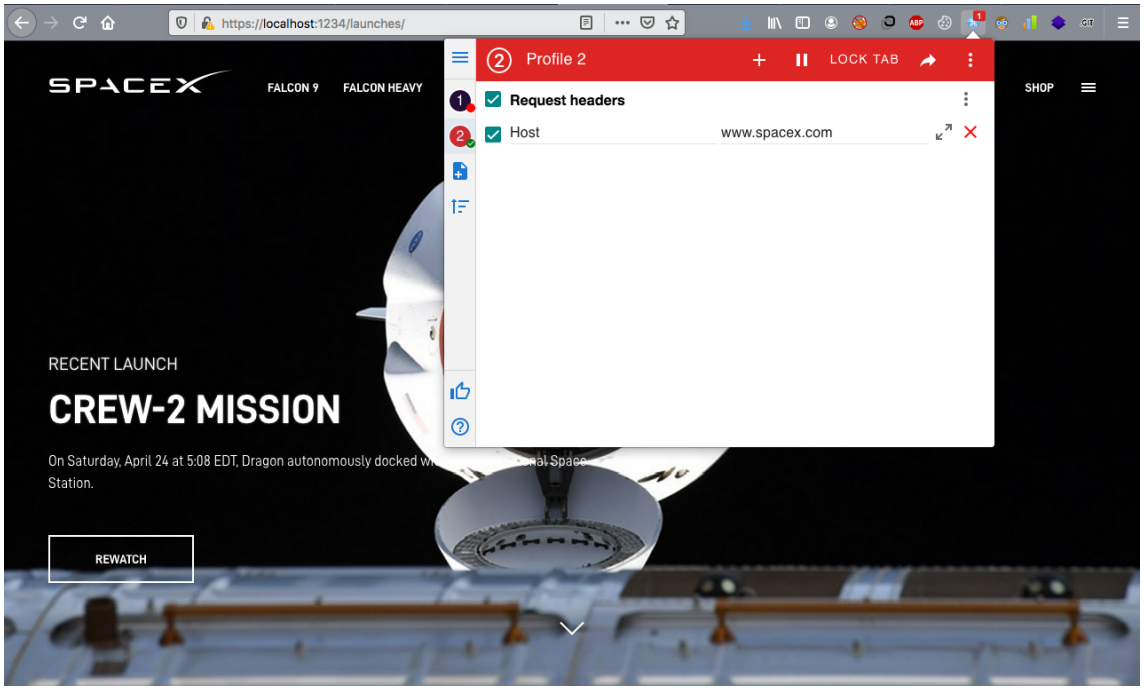
6.1.2 Paramétrage header

On a une erreur (prévisible) dû au mauvais header "Host".



6.1.3 Tunnel opérationnel

On change notre header "Host" en "www.spaceX.com". Et on a bien accès a spaceX.



6.2 Tunnel Dynamique

Cette fois on va monter un tunnel dynamique.

6.2.1 Paramétrage du proxy

On va en premier set up notre proxy sur firefox.

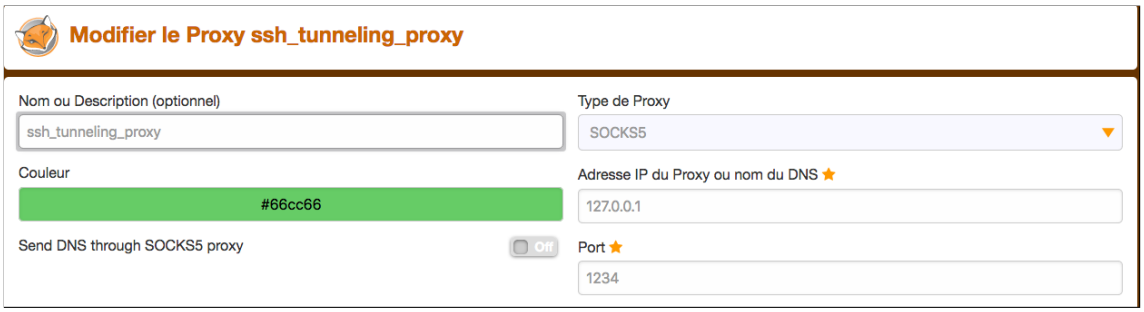
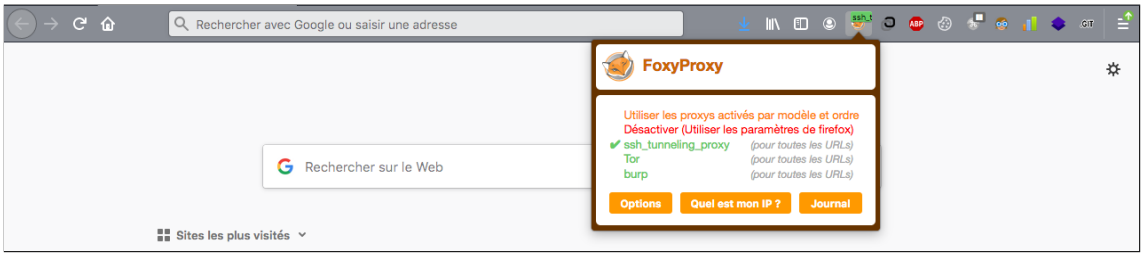
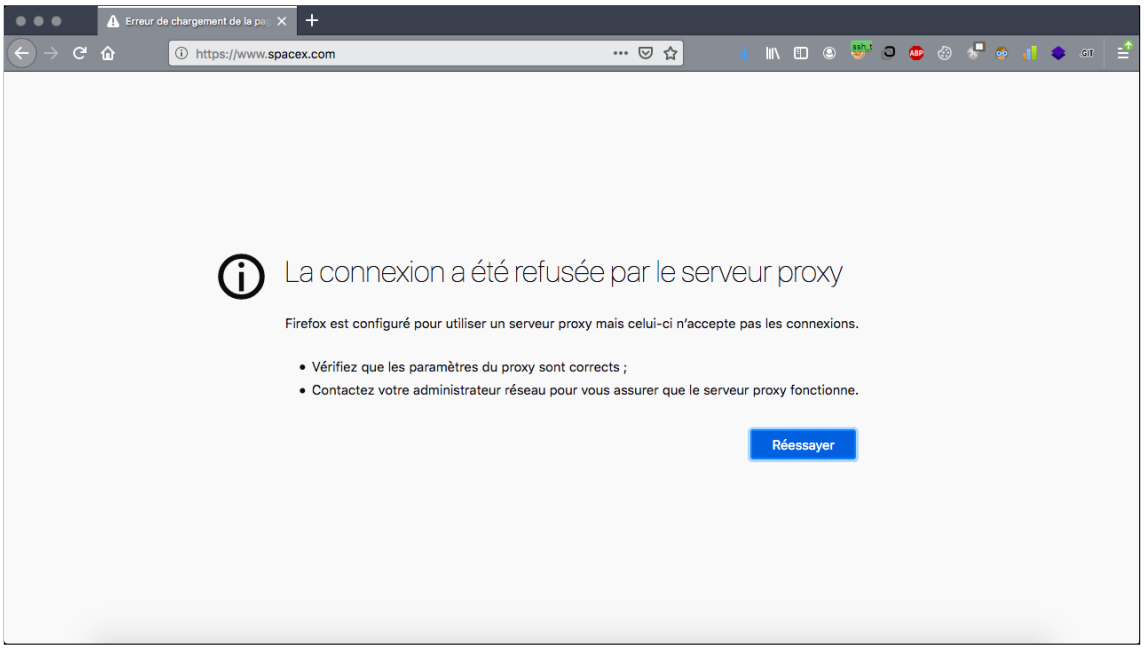


FIGURE 8 – Foxy proxy

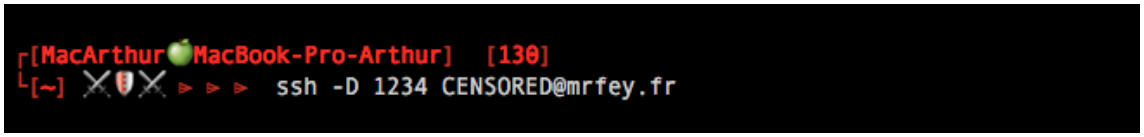


Si on essaye de se connecter l'accès nous est refusé, car on a pas encore monté le tunnel.



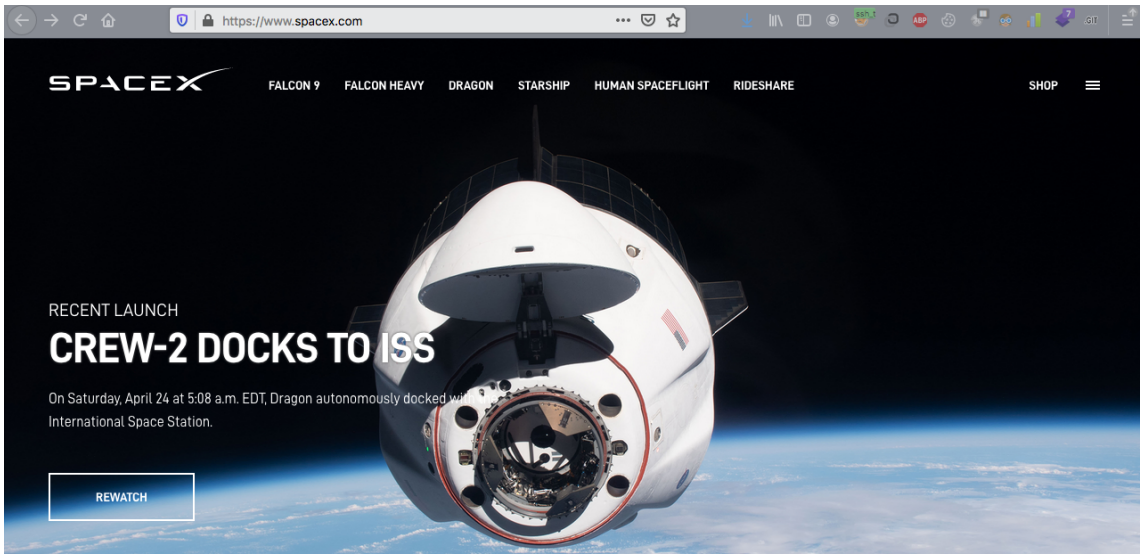
6.2.2 Connexion SSH

On monte alors notre tunnel dynamique.



6.2.3 Tunnel opérationnel

Cette fois on a bien accès a SpaceX en passant par le tunnel.



7 Conclusion

Le SSH tunneling a de multiples applications et différents modes de fonctionnement. Dans mon cas, j'ai souvent eu recours au ssh tunneling dans le cadre de la sécurité informatique mais je suis convaincu que chacun peut y trouver une utilité.

8 Sources

- https://fr.wikipedia.org/wiki/Secure_Shell
- https://youtu.be/MSbAr6iuedE
- https://www.it-connect.fr/chapitres/tunneling-ssh/

Un grand merci a Podalirius pour les tips Latex.