LE TÉLÉTRAVAIL SÉCURISÉ
en SSH avec LINUX et PUTTY

Le télétravail c'est pratique, voire indispensable en cas de problème comme nous pouvons le vivre actuellement, en mars 2020, a cause du confinement dû au virus Covid-19.
Mais sous prétexte de devoir « télétravailler », il ne faut pas le faire n'importe comment et se précipiter sur la facilité d'ouvrir votre entreprise aux 4 vents informatiques pour le plaisir de tous les hackers.
Exemples (trop) rapides: ouvertures de ports et accès RDP/TSE en direct sur votre poste, ou pire en accès VNC, ouverture de votre intranet à l'externe pour le transformer en extranet, etc.
Bien entendu quand on peut identifier des IP entrantes sur un routeur et les filtrer dans le pare-feu, c'est déjà un peu mieux. Mais quand ce n'est pas possible ou plus compliqué avec des IP dynamiques, ou même pour des PC nomades connectés en wifi public ou en 4G, les difficultés commencent et la sécurité se réduit.

Pour mettre toutes les chances de votre coté, beaucoup mieux sécuriser l'accès distant à votre poste de travail, tout en mettant en œuvre une solution « facile » je vous ai fait un petit tutoriel détaillé de qu'il est possible de faire avec des outils libres et gratuits.

Pour cela il vous faut :

- sur votre le lieu de travail :
  - un serveur (une machine) Linux
  - un routeur, une box, paramétrable pour laisser passer un seul port TCP défini en entrée/sortie
  - une IP fixe, ou à défaut un moyen d'accéder à votre entreprise avec un DYNDNS
- chez vous ou ailleurs: un poste sur Windows (ou Linux, ou Mac, ou Android) avec un accès à internet

Le serveur Linux pourra être, au choix et par exemple : soit une machine physique (même un assez vieux PC de type core 2, voire pentium pour les plus téméraires) sur Linux, soit un raspberry PI sur raspbian (ou un nano PC quelconque sur Linux), soit même une machine virtuelle sur Linux dans une infrastructure d'entreprise.

Le tutoriel portera pour l'exemple sur la possibilité d'établir en une seule fois une connexion de l'extérieur :
* en TSE (port 3389) sur un poste de votre entreprise dont l'IP est : **192.168.1.10**
* en HTTP (port 80) sur l'accès à un intranet d'entreprise dont l'IP est : **192.168.1.40**
* en VNC (port 5900) sur la télémaintenance d'un poste de l'entreprise dont l'IP est : **192.168.1.15**

Nous considérerons aussi que l'IP de notre entreprise est fixe en : 78.15.78.15

Vous avez tout cela et vous êtes prêts ? Alors allons-y…

1 - Paramétrer votre serveur Linux :

Je ne vais pas expliquer ici comment installer un Linux, ce n'est pas le propos et il y plein de tutos là-dessus. Vous pouvez installer à peu près ce que vous voulez, Ubuntu, Mint, Suse, Debian…
Comme le Linux ne fera office que de passerelle « SSH », un PC assez vieux sorti d'une cave ou d'une étagère poussiéreuse fera très bien l'affaire : un core duo avec 1 ou 2 GO de RAM et 20/30 GO de disque dur suffit amplement pour un système de base.
Une Raspbian sur un simple Raspberry PI 2, 3 ou 4 model B (entre 20 et 40 euros) fera aussi très bien le boulot. Pas besoin donc d'investissement lourd, juste d'une petite heure pour l'installation.
Pour ma part je baserai mon tutoriel , et les quelques commandes qui vont avec, sur une machine de type GNU/LINUX Debian 10 BUSTER.

Mon sujet commence une fois que votre machine Linux est prête et fonctionnelle sur votre réseau. Cela sous entend que vous y avez mis un système récent, à jour, et que vos avez créé un utilisateur (root ou classique avec un accès sudo) doté d'un mot de passe très (très) fort, pour éviter bien entendu que votre machine soit vulnérable dès le départ car c'est elle qui sera exposée principalement à l'extérieur.
Pour plus de simplicité, notre machine aura également besoin d'une IP fixe interne sur le réseau, pour le routage de port externe effectué par le routeur. Nous lui donnerons une IP sur le même réseau que nos machines : 192.168.1.200

Commençons par l'installation de la seule partie indispensable : un « serveur SSH ». Par l'intermédiaire du gestionnaire de paquet de votre distribution ou en ligne de commande, installez un serveur SSH :

apt-get install openssh-server

L'installation se suffit à elle-même. Par défaut les paramètres du fichier ***/etc/ssh/sshd_config*** doivent être corrects, pour vérification ils doivent notamment comprendre les lignes :
PermitRootLogin no
#AuthorizedKeysFile %h/.ssh/authorized_keys
(ou)
AuthorizedKeysFile %h/.ssh/authorized_keys2

Ensuite nous créons un utilisateur avec un mot de passe super (super) complexe. Lancez un terminal en root (ou sudo), puis tapez :

adduser monutilisateur

Entrez ensuite 2 fois un mot de passe, généré de manière aléatoire, long et complexe. Par exemple :
ohwohl2ahngah{Hiet}eek_i2oothe

Entrez le nom / prénom et autres renseignements… voilà votre utilisateur est créé.
Inutile de noter le mot de passe chez vous, car il ne devra jamais servir à connecter l'utilisateur en console. Perdez-le, c'est même mieux… l'utilisateur devra se connecter uniquement en SSH avec un système de clefs asymétriques et de passphrase.

Toujours dans le terminal nous allons maintenant nous connecter à l'utilisateur et générer ses paramètres de clef SSH.
Pour cela, connectez-vous à l'utilisateur :
su monutilisateur

(vous devez obtenir le prompt de type : monutilisateur@debian-buster:~$)

Créez le répertoire SSH et déplacez-vous dedans :
cd /home/monutilisateur
mkdir .ssh
cd .ssh

Créez ensuite la paire de clef SSH qui vous permettra la connexion sécurisée/chiffrée :
ssh-keygen -b 4096 -t rsa -f /home/monutilisateur/.ssh/clef- monutilisateur

Une passphrase vous est demandée … :
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):

Cette passphrase est importante, ne la négligez pas, elle va vous permettre de sécuriser au mieux votre accès. Entrez une phrase/mot de passe relativement complexe et facile à retenir en mêlant majuscules/minuscules/chiffres/caractères spéciaux.
Exemple : ***JadoreLesFruitsDeSaison2020$***

La commande a généré deux fichiers de clef que vous pouvez voir en listant le répertoire courant .ssh par un :

ls -al

-rw------- 1 monutilisateur monutilisateur 3326 mars 20 18:29 clef-monutilisateur
-rw-r--r-- 1 monutilisateur monutilisateur 748 mars 20 18:29 clef-monutilisateur.pub

Le fichier ***clef-monutilisateur*** est la clef privée, le ***clef-monutilisateur.pub*** la clef publique.

Dupliquez ensuite la clef publique dans un fichier nommé authorized_keys2, cela va permettre à l'utilisateur d'autoriser sa connexion distante sur le serveur grâce à cette paire de clef.
Toujours au sein du répertoire .ssh, cela peut se faire simplement ainsi :

cat clef-monutilisateur.pub > authorized_keys2

Éditez ensuite le fichier *authorized_keys2* pour y ajouter certains paramètres. Il commence par « ssh-rsa » puis une série de caratères.
Insérez devant les éléments suivants :

permitopen="192.168.1.10:3389",permitopen="192.168.1.40:80",permitopen="192.168.1.15:5900",no-pty

(de manière à avoir :
permitopen="192.168.1.10:3389",permitopen="192.168.1.40:80",permitopen="192.168.1.15:5900", no-pty ssh-rsa …)

Explications :
permitopen="192.168.1.10:3389" : permettra la prise en main surs le poste 1.10 en TSE port 3389
permitopen="192.168.1.40:80" : permettra l'accès vers l'extranet 192.168.1.40 port 80
permitopen="192.168.1.15:5900" : permettra la prise en main sur le poste 1.15 en VNC port 5900
no-pty = n'autorise pas le shell, pas de ligne de commande possible quand on se connecte

A noter que ces paramètres ne sont pas indispensables pour que l'accès distant fonctionne, mais ils sont une sécurité supplémentaire pour restreindre l'accès au strict nécessaire et à l'accès de certains postes uniquement.

Voilà, le poste Linux est paramétré. Avant de le quitter, copiez votre fichier de clef privée ***clef-monutilisateur*** sur une clef USB, vous en aurez besoin pour le script windows.

Passons maintenant au routeur.

2 - Paramétrer votre routeur/box :

Dans une entreprise on a généralement un routeur ou une box internet avec un pare-feu pouvant se paramétrer.
Ici aussi, comment paramétrer d'un routeur et naviguer dans les menus ne sera pas le propos, il existe plein de tutoriels ou d'aide sur Internet pour faire cela en fonction des matériels et/ou box.

Voici uniquement ce qu'il est nécessaire de faire.

Notre serveur SSH d'IP 192.168.1.200 écoute par défaut sur le port 22. Pour permettre un accès externe au serveur, il va nous falloir router un port externe vers notre port 22 sur le serveur 192.168.1.200.

Par sécurité, et comme le port 22 est un port très fréquemment testé par les pirates ou les robots, on peut renforcer la sécurité en définissant un autre port externe d'entrée dans l'entreprise. Nous prendrons arbitrairement le port 12322.
Dans votre routeur vous devez donc créer deux règles.
Une règle NAT qui transformera le port d'entrée 12322 vers l'IP 192.168.1.200:22

(input/from WAN) 12322 DESTINATION (output/to LAN) 192.168.1.200 port 22

Et une règle de pare-feu qui en autorise l'accès :

ALLOW (WAN to LAN) 192.168.1.200 port 22

Et c'est tout… votre système, votre entreprise, n'a pas à avoir d'autres ports (et donc d'autres « portes ») ouverts. Le seul port ouvert en externe sera le 12322 et il sera uniquement routé vers une machine Linux à jour et sécurisée.

3 - Préparer le script de connexion externe :

Pour stocker les scripts, vous aurez le choix soit de les préparer directement sur le PC externe d'un collaborateur (préférable), soit de les stocker sur une clef USB à donner au collaborateur pour qu'il s'en serve pour établir la connexion chez lui.

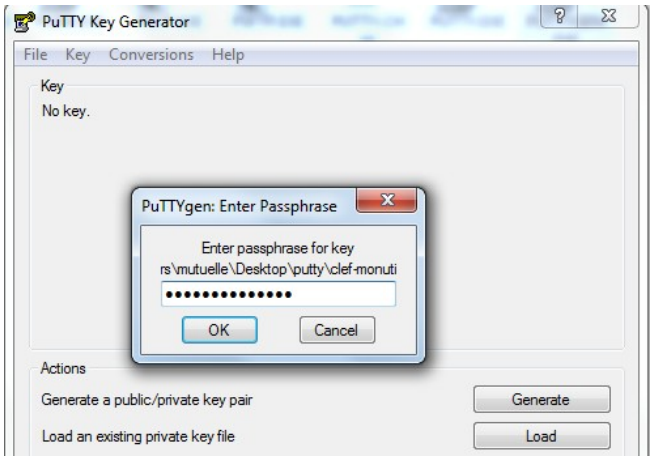
Sur Windows un seul package de programmes sera nécessaire à la connexion, il s'agit d'un package d'outils libres, opensource et gratuits nommé PUTTY.

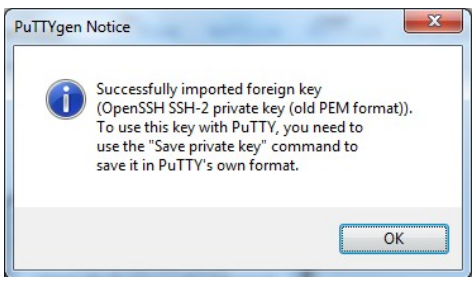
Au moment où ce tutoriel est écrit, vous pouvez trouver le package complet PUTTY (putty.zip) en suivant le lien officiel : https://www.putty.org/
Ou plus précisément sur :
https://the.earth.li/~sgtatham/putty/latest/w32/putty.zip (32 bits)
https://the.earth.li/~sgtatham/putty/latest/w64/putty.zip (64 bits)

Téléchargez le package et dézippez-le dans un répertoire « non UNC », de type c:\connexion par exemple. Vous devez y retrouver les fichiers suivants :

Copiez ensuite la clef privée ***clef-monutilisateur*** que vous avez mis sur une clef USB dans le même répertoire.

Nous allons commencer par convertir votre clef privée pour qu'elle soit compatible avec les outils PUTTY. Pour cela lancez l'outil PUTTYGEN. Cliquez ensuite sur LOAD et chargez votre clef privée ***clef-monutilisateur*** . La passphrase vous est logiquement demandée, tapez là pour décoder et charger la clef.





Cliquez ensuite sur SAVE PRIVATE KEY et donnez lui le même nom avec l'extension .ppk : ***clef-monutilisateur.ppk***

Voilà vous avez votre fichier de clef privée converti en format « putty », ***clef-monutilisateur.ppk***, la seule qu'il vous faudra. Vous pouvez maintenant supprimer la clef Linux ***clef-monutilisateur*** (sur le c:\connexion mais aussi sur votre clef USB).

Passons aux scripts à proprement parler. Pour plus de lisibilité nous en ferons 3, pour procéder à la connexion en 3 étapes.

Le premier consiste à stocker la clef tant que le PC reste allumé à l'aide de l'outil PAGEANT.EXE.
Éditez un fichier nommé ***1-Identification-de-la-clef.bat*** et tapez-y la commande :

START c:\connexion\pageant.exe c:\connexion\clef-monutilisateur.ppk

Le deuxième script va permettre de se connecter au poste Linux en définissant les paramètres de « routage » nécessaires pour accéder aux postes distants.
Editez un fichier nommé ***2-Connexion-poste-linux.bat*** et tapez-y la commande :

c:\connexion\plink.exe -no-antispoof -v -L 13389:192.168.1.10:3389 -L 8000:192.168.1.40:80 -L 15900:192.168.1.15:5900 monutilisateur@78.15.78.15 -P 12322

On définit ici de faire :

- une simple connexion à l'aide de PLINK, outil de connexion SSH
- en enlevant la confirmation antispoof demandant de valider par ENTER
- en visualisant de manière un peu plus verbale ce qu'il se passe (-v)
- en faisant un routage LOCAL <=> DISTANT de nos 3 IP+ports
- en se connectant à notre entreprise (78.15.78.15) en tant qu'utilisateur monutilisateur
- en utilisant (avec -P) le port externe 12322

Le (ou les) troisième(s) script(s), pas forcément nécessaire, consistera ensuite à lancer les commandes selon les besoins de connexion :

- lancer une connexion TSE sur notre poste distant
- et/ou accéder à l'intranet distant avec un navigateur
- et/ou lancer une connexion VNC sur notre poste distant

Éditez par exemple un fichier nommé ***3-Lancement-application.bat*** et tapez-y les commandes (à adapter selon les navigateur, les programmes de gestion VNC...) :

START mstsc /v:127.0.0.1:13389
START "C:\Program Files\Mozilla Firefox\firefox.exe" http://127.0.0.1:8000
START "C:\Program Files\uvnc bvba\UltraVNC\vncviewer.exe" 127.0.0.1:15900

Explications : Quand la connexion PLINK sera établie (script étape N°2), les IP+ports passées en paramètre avec les options « -L » permettent de router le localhost (IP 127.0.0.1, adresse locale) avec certains ports définis vers les IP et les ports distants.
Par exemple : puisqu'on a passé en paramètre -L 13389:192.168.1.10:3389, quand on essaye d'accéder sur la machine locale à 127.0.0.1:13389 PLINK, une fois le lien établi, fait transiter de manière automatique la connexion vers l'adresse 192.168.1.10:3389 du réseau distant.
Concrètement quand on lancera une connexion TSE avec ***mstsc /v:127.0.0.1:13389*** PLINK la transformera de manière transparente en ***mstsc /v:192.168.1.10:3389*** comme si on se trouvait sur le réseau distant.
A noter ici le choix délibéré dans l'option -L de changer les ports « de base », 3389, 80 et 5900, par des ports personnalisés : 13389, 8000 et 15900. Ce choix est fait pour éviter les problèmes de conflits de ports d'écoute sur la machine locale cliente, avec des ports qui seraient déjà utilisés dans

certains cas, par certains services et sur certains PC clients. Pour éviter tout problème, l'utilisation de ports personnalisés qui ne risquent pas d'être déjà utilisés est donc préférable.

A noter aussi qu'un script de lancement d'application ne sera pas obligatoirement nécessaire pour le télétravail, mais quand on prépare une connexion clef en main pour un collaborateur ce sera beaucoup plus simple à appréhender et à utiliser.

Par exemple pour l'accès à l'intranet distant, et une fois la connexion établie, le simple fait de taper dans un navigateur http://127.0.0.1:8000 permettra l'accès au site à distance.

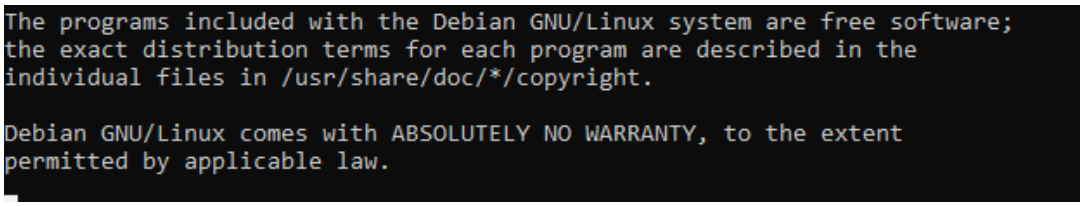
Voilà, vos scripts sont faits… il ne reste plus qu'à les utiliser dans l'ordre :

- lancez le ***1-Identification-de-la-clef.bat*** et tapez votre passphrase. Si vous vous trompez la fenêtre revient sans cesse… si c'est OK, la fenêtre disparaît et un petit icône stockant votre clef se place en bas à droite dans la barre des tâches Windows : 

- lancez ensuite le ***2-Connexion-poste-linux.bat***, la connexion doit s'établir de manière automatique. La toute première fois cependant, un message apparaît et vous demande de stocker le fingerprint du serveur en cache :

```
The server's rsa2 key fingerprint is:
ssh-rsa 4096 **:**:**:**:**:**:**:**:**:**:**:**:**:**:**:**
If you trust this host, enter "y" to add the key to
PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without
adding the key to the cache, enter "n".
If you do not trust this host, press Return to abandon the
connection.
Store key in cache? (y/n)
```

Répondez « y », la question ne vous sera plus posée à l'avenir sur votre poste.
Si vous obtenez la connexion sans soucis, la fenêtre se fige ainsi.



Ne fermez pas cette fenêtre et laissez-la réduite dans votre barre des tâches, elle maintiendra la connexion jusqu'à la fin de votre travail.

- lancez ensuite le script de connexion ***3-Lancement-application.bat*** pour accéder à vos applications. Elle se lanceront de manière automatique en vous invitant à entrer vos paramètres e connexion sur les élément distants, utilisateur+mot de passe pour le TSE, page de navigateur pour l'intranet, mot de passe pour le VNC.

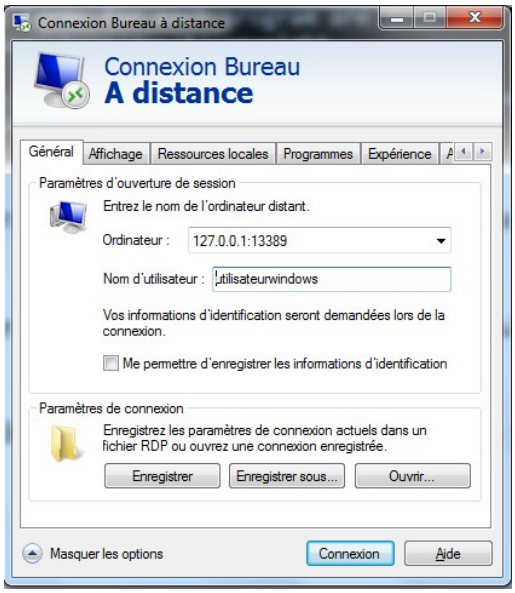
Vous pouvez aussi y accéder de manière plus classique en lançant les programmes associés et en tapant les adresses locales.

TSE

avec

la commande

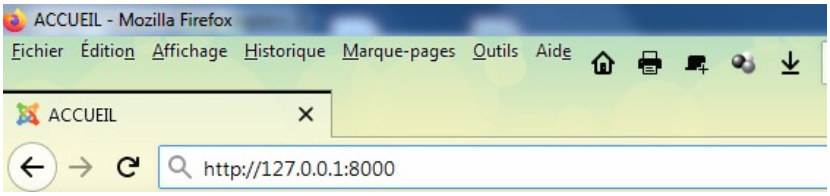
MSTSC :



Votre intranet

avec

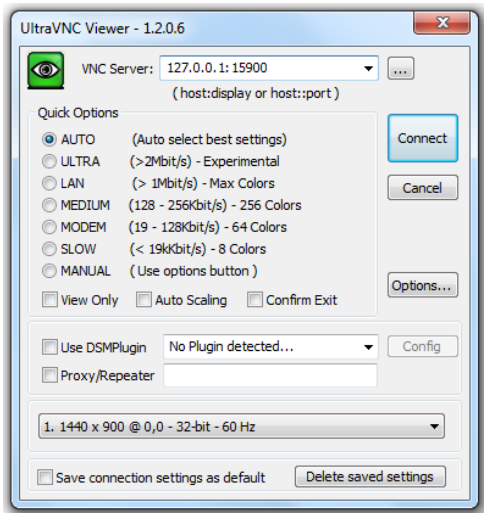
FIREFOX :



VNC

avec

ULTRAVNC :



Vous êtes désormais parés pour un télétravail en toute sécurité.

Cette méthode possède plusieurs avantages :

- elle est sécurisée à différents niveaux qui s'empilent, notamment pour l'accès à un poste :
  - nécessité de posséder le fichier de clef privée
  - nécessité de connaître la passphrase de la clef
  - nécessité de connaître le mot de passe du poste Windows, TSE ou VNC

  Soit 2 niveaux de plus qu'une simple connexion ouverte directement vers un poste.
- elle est chiffrée par le protocole SSH et elle sécurise donc nativement dans un tunnel tous les flux TCP qui ne le sont pas par conception, comme des flux FTP, VNC, HTTP. Elle permet donc également de se connecter à partir de réseaux publics ou partagés sans crainte.
- elle n'expose pas plusieurs ports sur internet, un seul port d'entrée personnalisé suffit (ici le 12322). Malgré cela, grâce à la connexion PLINK et aux multiples options « -L », on peut même imaginer de router plusieurs dizaines de ports et donc plusieurs dizaines d'accès en TCP vers son réseau externe : postes, caméra IP, serveur de mail, etc...

- elle est sécurisée grâce à une simple machine Linux à jour et maintenue qui fait office de passerelle et qui peut même permettre que son propre réseau interne le soit moins, en routant par exemple des connexions vers des postes en fin de vie Windows 7, Windows serveur 2008, voire des Windows XP. Votre Linux pourra même être renforcé contre les attaques par des outils que je ne décrit pas ici, comme FAIL2BAN par exemple, et servir à autre chose sur votre réseau.
- en cas de doute, par exemple sur la corruption d'une clef externe d'un collaborateur, l'accès distant peut être immédiatement supprimé par l'administrateur de la machine Linux en effaçant tout simplement la clef publique stockée dans le .ssh/auhorized_keys2 et/ou en supprimant l'utilisateur Linux créé. Aucun autre effet de bord à craindre.
- elle est LIBRE et GRATUITE. Tous les éléments décrits sont opensource et utilisables sans investissement.

Je ne décris pas de manière détaillée dans ce tutoriel la possibilité de se connecter à partir d'un poste externe en LINUX, MAC ou ANDROID, mais sachez que c'est également faisable car il existe des outils portables sur tous les systèmes pour établir une connexion SSH.

Avec LINUX vous pourrez le faire très simplement avec :
- la commande native ***ssh*** qui est le client ssh linux et qui remplacera PLINK avec les mêmes syntaxes
- un script écrit en BASH pour automatiser la connexion
- ***REMMINA*** et ses plugins pour créer et manager des connexion RDP/TSE ou VNC par exemple

Sur un MAC, la commande ***ssh*** existe aussi. Je vous laisserai le soin de trouver ce qui permet le RDP/TSE ou le VNC.

Et enfin, sur ANDROID il existe également des outils pour tout faire :
- ***CONNECTBOT*** ou ***JUICESSH*** pour établir des connexion SSH avec gestion de clefs privées et routage de ports
- ***REMOTERDP*** et ***ANDROIDVNC*** pour faire du RDP/TSE et du VNC

Sachez qu'il existe bien d'autres possibilités avec ce formidable outil et protocole qu'est SSH, vous n'avez ici qu'une introduction avancée, mais il vous restera encore beaucoup à découvrir.

Je vous souhaite un excellent télétravail, et de belles et nombreuses expériences avec sa mise en œuvre.