VPN Libre/OpenSource

Exemples de mises en oeuvre

G. Arcas <guillaume.arcas@free.fr>
OSSIR / 14 février 2004

VPN Libre/OpenSource

- VPN Niveau 4 : OpenVPN, SSLTunnel
- VPN Niveau 7 : OpenSSH
- VPN Niveau 3 : IPSEC : OpenSWAN, ISAKMD, ipsec-tools
- Niveau 4 et Niveau 7 : userland, pas de modification du noyau, possibilité de chrooter ou d'exécuter sans les privilèges root.
- IPSEC : dans le noyau, protocoles standard.

VPN Libre/OpenSource

- Introduction
- Critères de choix (***arbitraires***)
  - Tunnels chiffrés
  - Nomades – passerelle : VPN SSL
  - Passerelle – passerelle : IPSEC
  - Hybride / AdHoc : SSH
  - Portabilité : Unix, MS Windows, Mac OS X
  - Facilité de mise en oeuvre et d'administration
- Liste non exhaustive !

VPN Libre/OpenSource

VPN SSL avec OpenVPN

VPN Libre/OpenSource

- OpenVPN
  - http://www.openvpn.net
- Application client / serveur
- Connexions sécurisées via SSL
- Authentification forte mutuelle
- Support des OS couramment utilisés : MS Windows, Linux, *BSD, Mac OS X, Solaris

VPN Libre/OpenSource

- OpenVPN 1/13
- 2 modes
  - Tunnel : niveau 3, IP
  - Tap : niveau 2, Bridge
- 2 méthodes d'authentification
  - Clef secrète partagée, Point-à-point
  - Certificats x509, Client(s) / Serveur

VPN Libre/OpenSource

- OpenVPN2/13
- Authentification par clef secrète partagée
  - 1 clef pour le chiffrement, 1 clef pour l'authentification (digest)
  - Il est possible d'utiliser 4 clefs, 1 paire par hôte
  - ☺ Configuration simple (voire simpliste)
  - ☹ Les clefs sont les mêmes à chaque session.
  - ☹ La solution n'est pas aisément « scalable ».

VPN Libre/OpenSource

- OpenVPN 3/13
- Authent. Certificats x509
  - Chaque partie (clients et serveur) possède son certificat et sa clef privée
  - Clefs de session uniques et fréquemment renouvelées
  - Utilisation de CRL
  - Peut nécessiter une PKI si nombre important de clients

VPN Libre/OpenSource

- OpenVPN 4/13
- Installation
  - Installation en Userland : possibilité de cloisonner (chroot), d'exécuter en mode non privilégié.
  - Interfaces TUN / TAP
  - OpenSSL
  - Librairie compression LZO (optionelle)
  - Binaires fournis pour MS Windows
  - Pas de distinction entre binaires client et serveur

VPN Libre/OpenSource

- OpenVPN 5/13
- Authent. Certificats x509
  - Scripts easy-rsa (AC sur machine dédiée)
    - build-ca ⇨ ca.crt + ca.key
    - build-key-server *server* ⇨ server.crt + server.key
    - build-key *client* ⇨ client.crt + client.key
    - build-dh ⇨ dh1024.pem
    - server.crt + server.key : sur le serveur OpenVPN
    - client.crt + client.key : sur chaque client
    - ca.crt sur serveur ET chaque client

VPN Libre/OpenSource

- OpenVPN 6/13
- Configuration Serveur 1/
  - local a.b.c.d / port / proto
  - dev tun || dev tap
  - Fichiers ca.crt / server.crt / server.key / dh
  - server 192.168.1.0 255.255.255.0
  - push « redirct-gateway def1 »
  - push « dhcp-option DNS/WINS »

- OpenVPN 7/13
- Configuration Serveur 2/
  - Journalisation
    - log || log-append filename (syslog par défaut)
    - verb [0-9]
    - status filename
  - Interface de management (mode texte)
    - management ip port
      - Accès par telnet

VPN Libre/OpenSource

- OpenVPN 8/13
- Configuration Serveur 3/
  - duplicate-cn : 1 certificat pour plusieurs clients
  - client-to-client : connexions entre clients
  - client-config repertoire : paramètres propres à un client
  - crl-verify crl (script revoke-full *client_name*)
    - Note : client-config et crl-verify sont lus à chaque nouvelle session sans redémarrage du démon OpenVPN

VPN Libre/OpenSource

- OpenVPN 9/13
- Haute dispo.
  - Plusieurs serveurs :
    - remote server1
    - remote server 2
  - Plusieurs instances sur un même serveur
    - remote server1 8000
    - remote server 1 8001
  - Failover : server1 puis server2 si échec
  - Load-Balancing : remote-random

VPN Libre/OpenSource

- OpenVPN 10/13
- Sécurité renforcée
  - tls-auth : signature (HMAC) des paquets échangés sur le canal de contrôle. Anti DoS
    - Le fichier utilisé doit être le même sur les clients et le serveur (clef secrète partagée).
  - replay-persist filename : stocke les « tables » de session pour éviter les replay même après redémarrage du serveur
  -

VPN Libre/OpenSource

- OpenVPN 11/13
- Interface de management du serveur
  - management localhost 7505
  - telnet localhost 7505
    - Déconnexion d'un client
    - kill cn (cn = CommonName)
    - kill IP:port
    - status
    - Reload / Arrêt du démon : signal SIGTERM/SIGHUP

VPN Libre/OpenSource

- OpenVPN 12/13
- Configuration Client
  - Objectif : qu'elle soit la plus simple possible et reproductible.
    Si paramètres spécifiques ⇨ client-config sur le serveur
  - client / dev tun / remote a.d.c.d port / proto
  - tls-remote cn (avec cn = CommonName serveur)
  - ns-cert-type server

VPN Libre/OpenSource

- OpenVPN 13/13
- Configuration Client
  - Passage de (par) proxy HTTP
  - proto tcp
  - http-proxy ip port stdin|filename basic|ntlm
    - stdin : entrée login/password par utilisateur
    - filename : login/password stockés dans un fichier

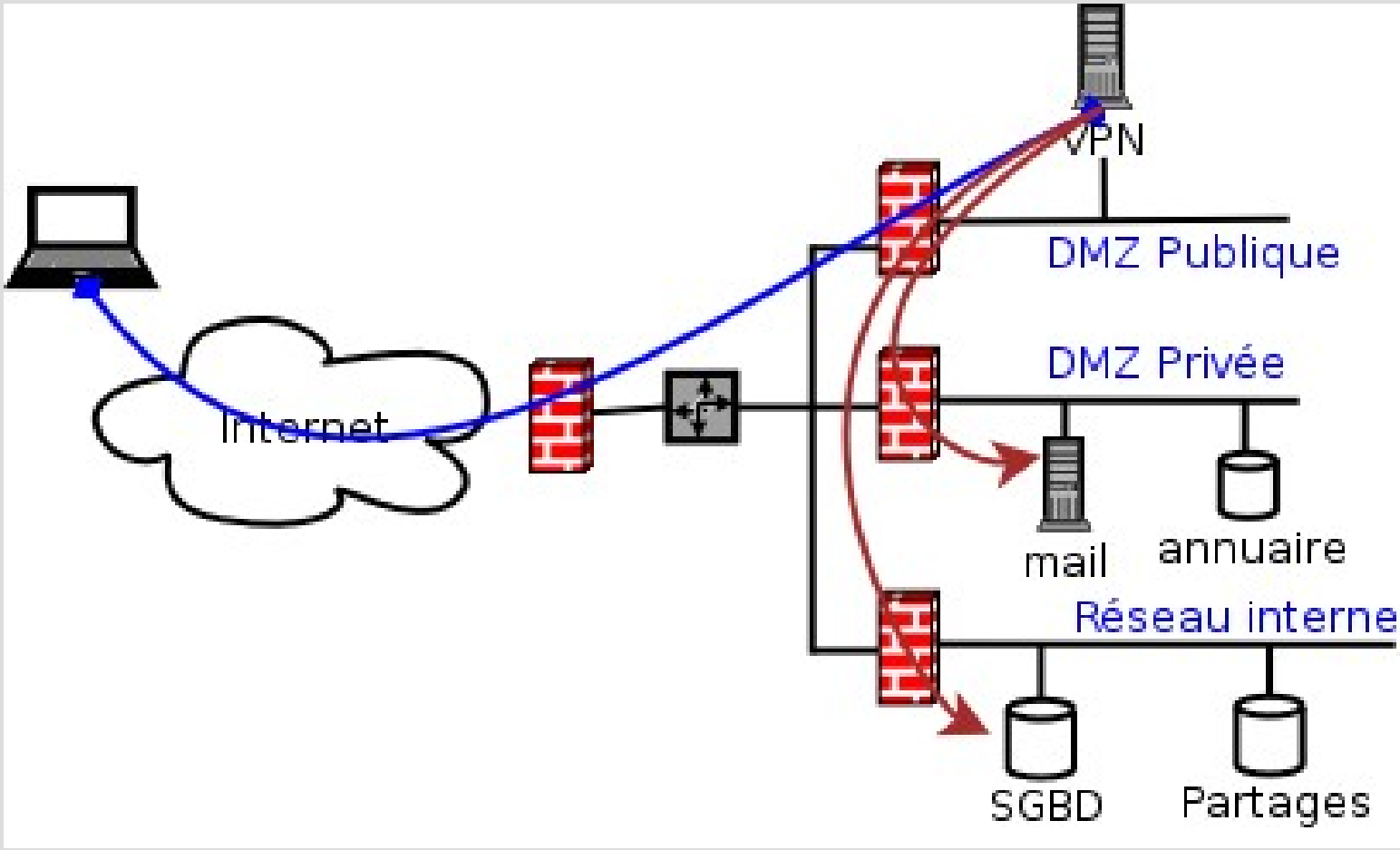
VPN Libre/OpenSource

Exemples OpenVPN

VPN Libre/OpenSource

- Point d'accès OpenVPN « léger »
- Boitier Soekris net4801
- OpenBSD 3.8
- OpenVPN 2.0.2 (paquetage OpenBSD)

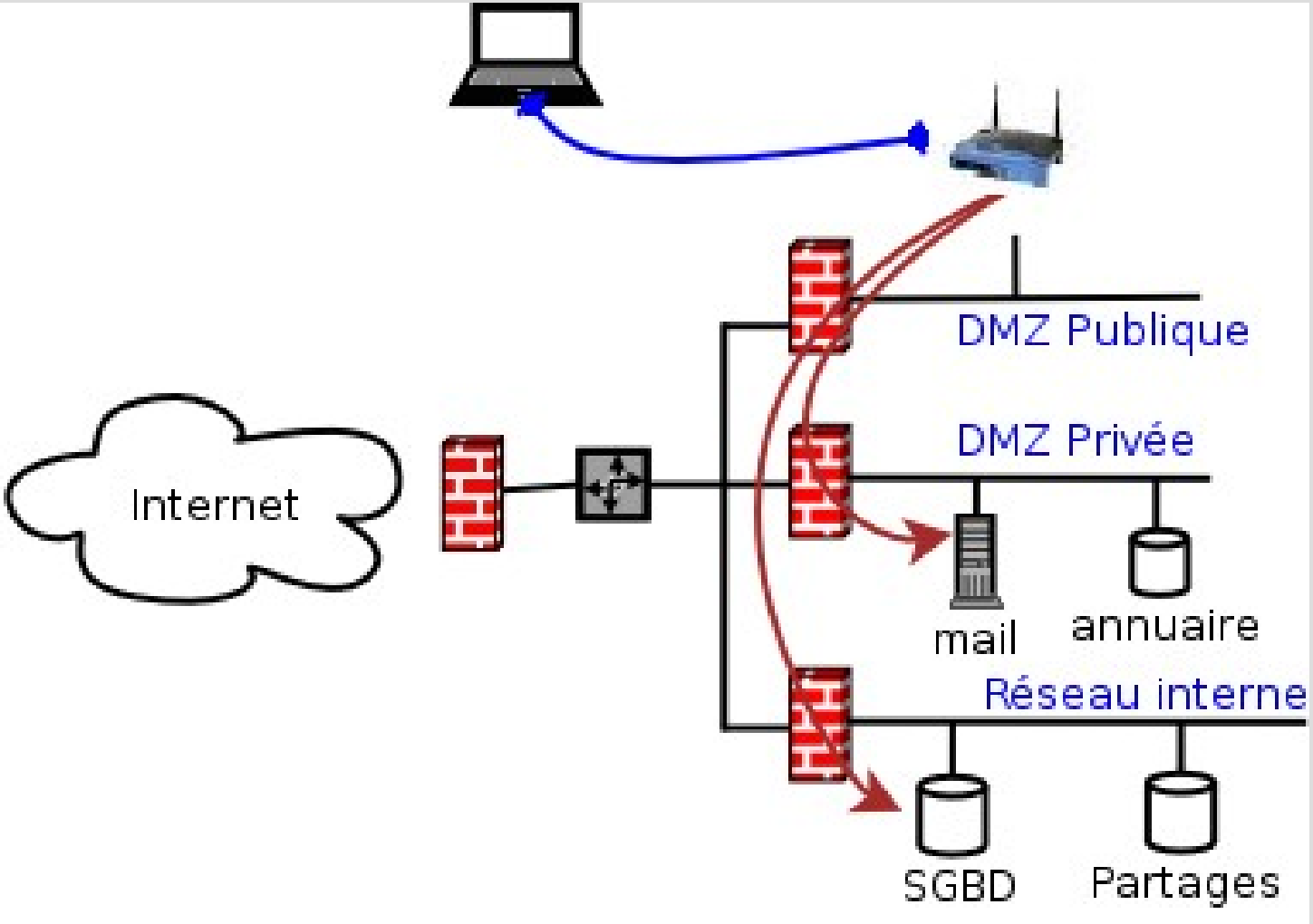
VPN Libre/OpenSource



VPN Libre/OpenSource

- Sécurisation d'un point d'accès sans-fil
- AP
  - Linksys WRT54G
  - OpenWRT RC4
  - OpenVPN 2
- Clients
  - MS Windows XP, Linux, etc.

VPN Libre/OpenSource



VPN Libre/OpenSource

- Cahier des charges
  - Authentification des clients et du serveur par certificat
  - Authentification des utilisateurs possible si mot de passe pour la clef
  - Pas de communication directe entre clients
  - Le serveur devient la route par défaut et attribue les adresses IP (DHCP) aux clients
  - Trafic IP

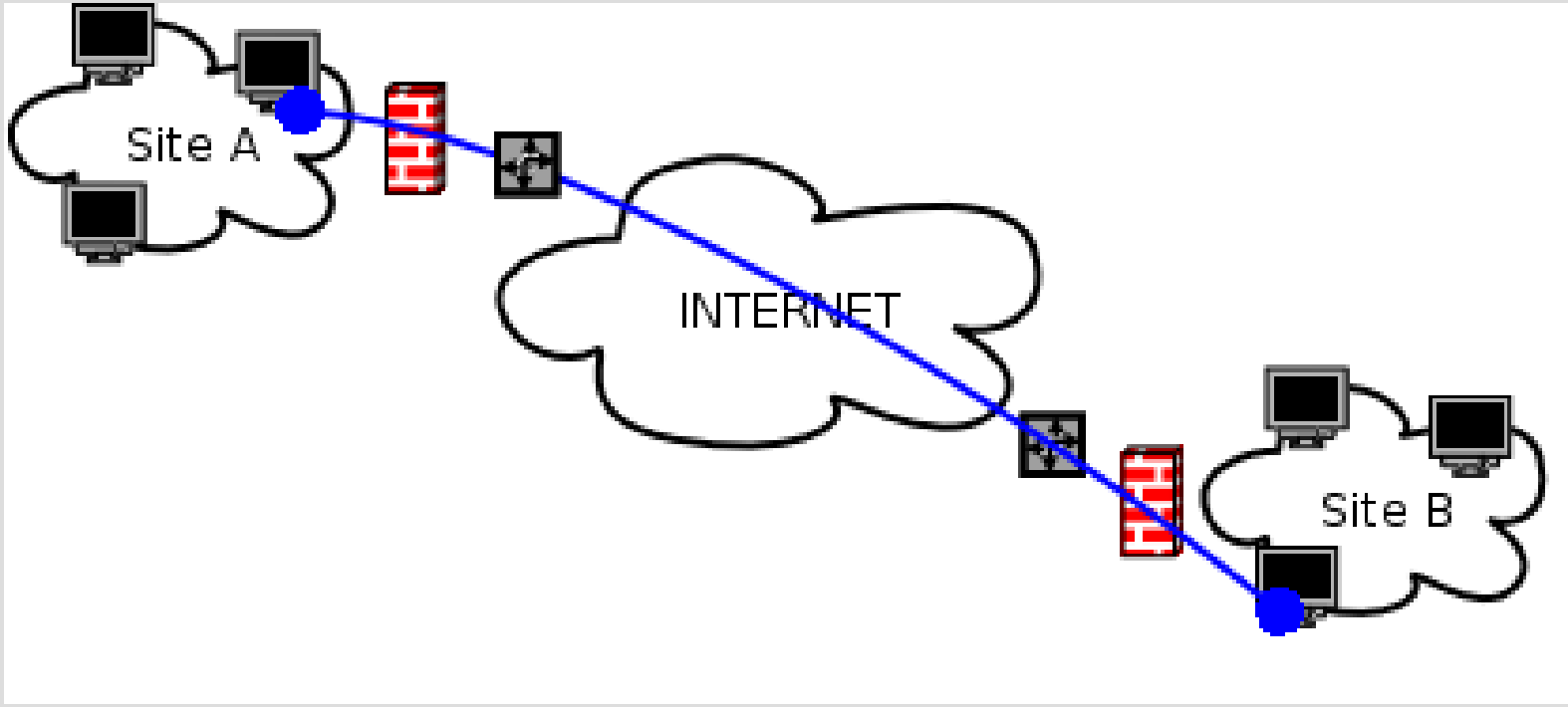
VPN Libre/OpenSource

Tunnel SSH avec OpenSSH 4.x

OpenVPN Libre/OpenSource

- OpenSSH 4.3
  - ☝ Pas personnellement testé sur la durée !
- Permet d'établir un tunnel chiffré entre deux machines via l'interface TUN
  - Côté serveur
    - sshd_config : PermitTunnel yes
  - Côté client :
    - config. interface tun up
    - ssh_config : Tunnel yes + TunnelDevice
- Alternative « légère » à OpenVPN ?

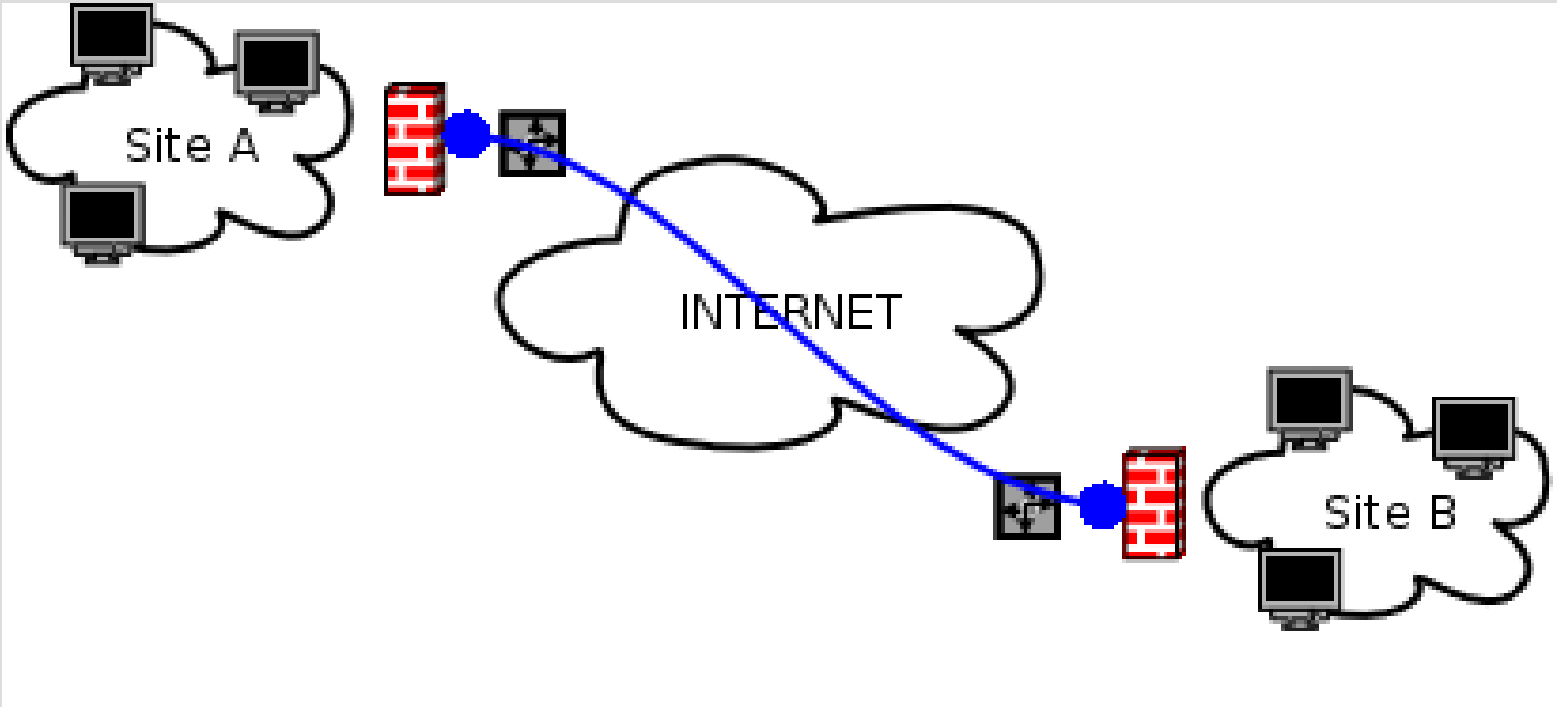
VPN Libre/OpenSource



VPN Libre/OpenSource

VPN IPSEC

VPN Libre/OpenSource



VPN Libre/OpenSource

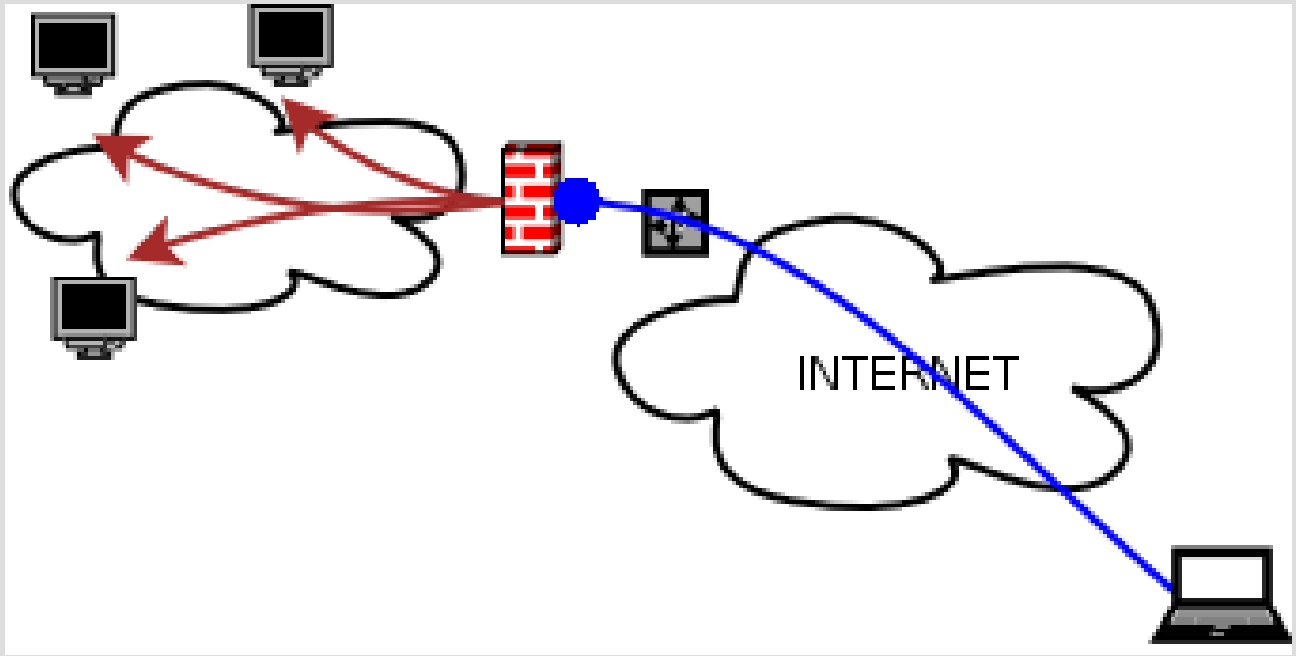
OpenSWAN

VPN Libre/OpenSource

- OpenSWAN
  - http://www.openswan.org
  - « Héritier » du projet FreeSWAN.
    - Tout comme StrongSWAN et SuperFreeSWAN...
  - Mise en oeuvre des protocoles IPSEC pour les noyaux Linux 2.0 à 2.6
    - Nécessite en théorie une recompilation de noyau mais en pratique les principales distributions sont « IPSEC ready ».

VPN Libre/OpenSource

- Mode « Road Warrior »
  - Connexion poste nomade ⇨ passerelle

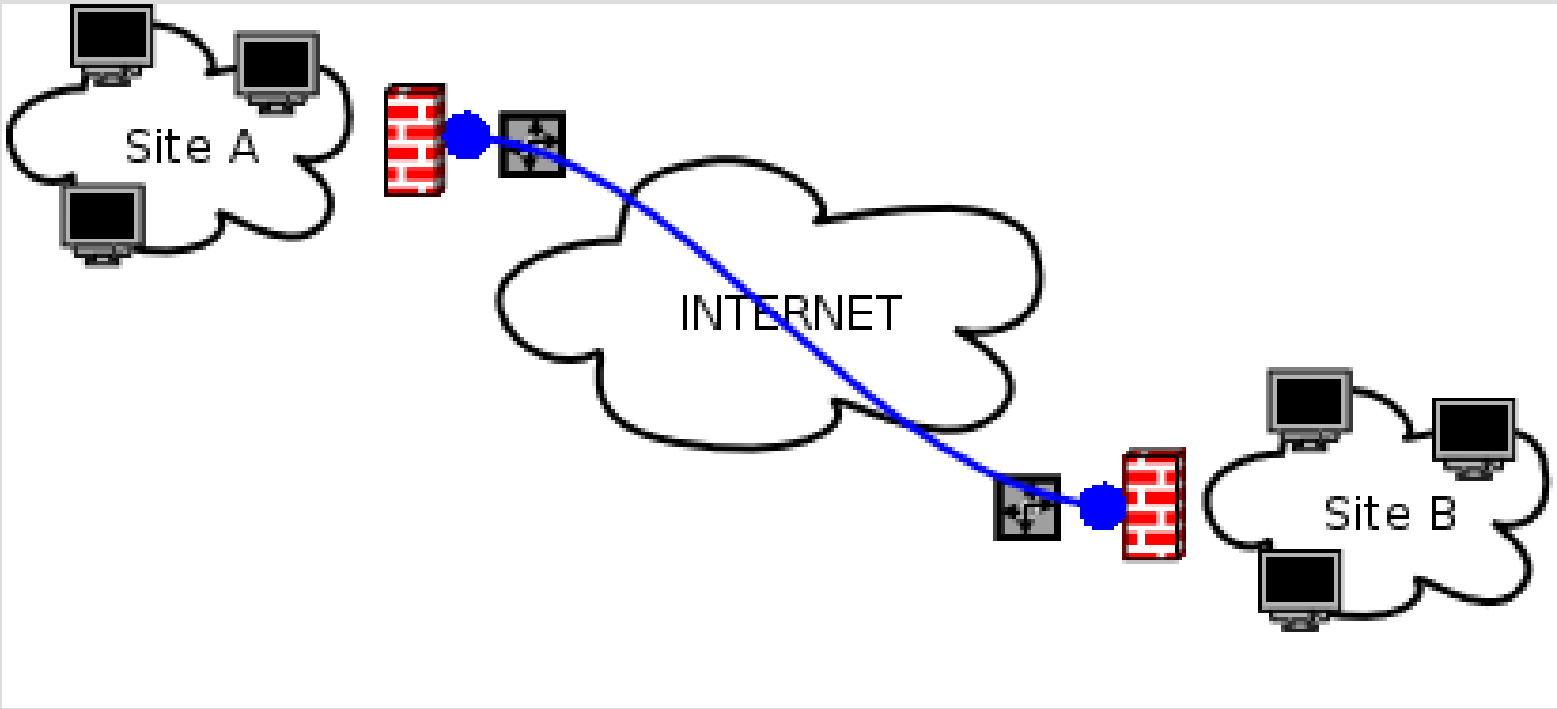


VPN Libre/OpenSource

- OpenSWAN, mode Road Warrior
- Dans ce mode, on définit une partie Gauche (LeftSide) et une partie... Droite.
  - Par convention **L**eft = **L**ocal, **R**ight = **R**emote.
- Fichier ipsec.conf
    - conn road (road = identifiant)
    - Chaque machine doit avoir la clef publique de l'autre : leftrsasigkey et rightrsasigkey sur chaque machine
    - Activation : ipsec auto -up road

VPN Libre/OpenSource

- Mode Net-to-Net



VPN Libre/OpenSource

- OpenSWAN, mode net-to-net
  - conn net-to-net

```
left=192.0.2.2                  # Local side
leftsubnet=172.16.0.0/24        #
leftid=@xy.example.com          #
leftrsasigkey=0s1LgR7/oUM...    #
leftnexthop=%defaultroute
right=192.0.2.9                 # Remote side
rightsubnet=10.0.0.0/24         #
rightid=@ab.example.com         #
rightrsasigkey=0sAQOqH55O...    #
rightnexthop=%defaultroute
auto=add
```

  - ipsec auto --up net-to-net

VPN Libre/OpenSource

- OpenSWAN mode « Opportuniste »
  - Opportunistic Encryption
  - Permet de monter un tunnel sans authentification préalable
  - Utilise IKE et le DNS (RR TXT) pour les échanges de clefs
  - Fonctionnalité « pure » OpenSWAN et par conséquent incompatible avec toute autre implémentation IPSEC
  - Solution attractive si « full openSWAN »

VPN Libre/OpenSource

ISAKMPD

VPN Libre/OpenSource

- OpenBSD/ISAKMPD
- Implémentation de Isakmpd dans OpenBSD par Niklas Hallqvist & Niels Provos en 1998.
  - Isakmpd porté sur Linux.
- IPSec est supporté par le noyau 'GENERIC' fourni par défaut.
- 2 façons de monter un tunnel IPSec :
  - manuellement ipsecadm(8) ou par l'intermédiaire d'un démon isakmpd(8).

VPN Libre/OpenSource

- ISAKMPD
  - Démon permettant d'établir et de gérer les associations de Sécurité pour le traffic IPSec.
  - Gestion des echanges des authentifications et des sessions.
  - Le protocole utilisé pour cette gestion est IKE.
  - Présent par défaut dans OpenBSD
  - Séparation des privilèges
    - fork en 2 processus au démarrage
    - le processus fils est chrooté dans /var/empty.

VPN Libre/OpenSource

- CONFIGURATION
  - tunnel mode et transport mode,
  - Supporte multiple méthodes d'authentification :
    - Passphrase (Shared Key)
    - Public Key
    - Certificats X509
    - Keynote.
  - 'IKE mode-config' : pour les Road-Warriors, permet d'identifier chaque peer et de fournir une configuration réseau (IP/netmask/Nameserver/WINS-server).

VPN Libre/OpenSource

- AVANTAGES
  - Présent par défaut dans OpenBSD
  - Produit qui a fait ses preuves
  - Fourni avec des outils pour la génération de certificats, possibilité de debugger finement.
  - compatible avec beaucoup de plateformes propriétaires : Cf. Démo IPSEC/HSC 2001
  - Configuration pour Road-Warriors : possibilité d'affecté un sous-réseau aux nomades et de distribuer des configurations réseaux.

VPN Libre/OpenSource

- INCONVÉNIENTS
- Configuration indigeste mais fine (mais documentation fournie, claire et explicite) :
- Deux fichiers texte de configuration isakmpd.conf et isakmpd.policy
- Pas de GUI.

VPN Libre/OpenSource

IPSEC-TOOLS

VPN Libre/OpenSource

- Références
  - http://www.openvpn.net
  - http://www.openssh.org
  - http://www.openswan.org
  - http://ipsec-tools.sourceforge.net
  - http://www.hsc.fr/ressources/ipsec/ipsec2001/
- Remerciements
  - Jérome Léonard pour la partie OpenBSD/ISAKMPD
- http://yom.retiaire.org