Composants fondamentaux d'une architecture sécurisée
Tunnels, VPN et IPSec

Carlos Aguilar
carlos.aguilar@enseeiht.fr

IRIT-IRT

Plan

Les tunnels

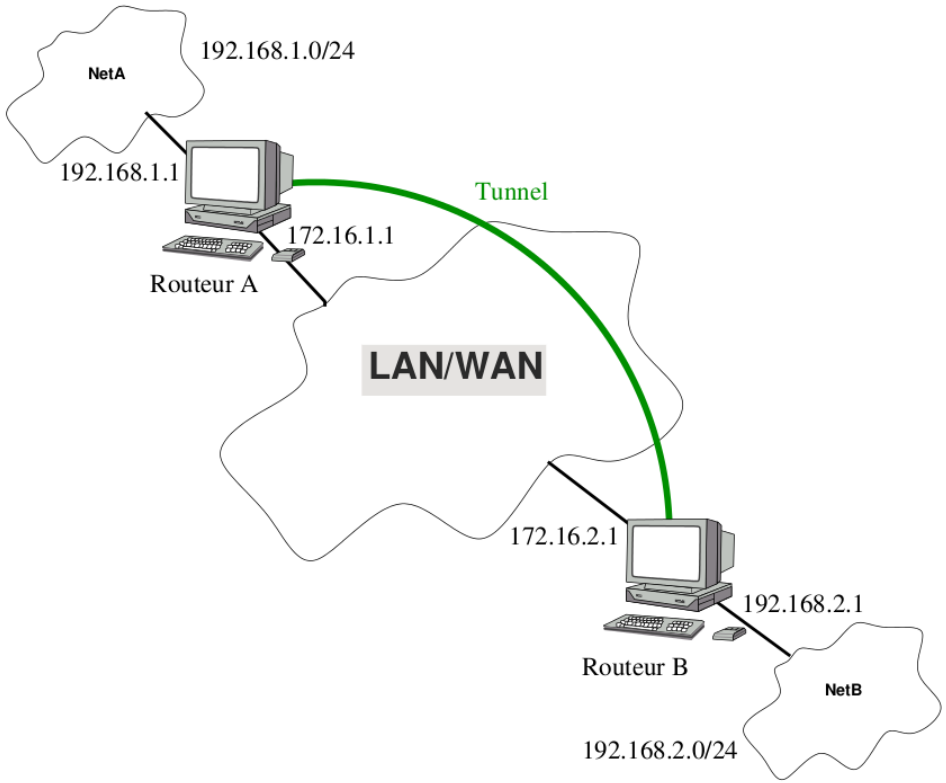
Définition

Encapsulation d'un protocole (généralement de niveau 2 ou 3) dans un protocole de niveau 3

Deux méthodes classiques

- "IP in IP"
  - Prévu dans IPv4 et dans IPv6
  - Tout mélange d'IPv4 et IPv6 pot. possible (4x4, 4x6, 6x4, 6x6)
  - Deux en-têtes IPs qui se suivent
- Generic Routing Encapsulation (GRE)
  - En-tête GRE intercalée entre les deux entêtes IP (dans le cas du IP sur IP)
  - N'importe quel protocole (avec un type valide) peut être encapsulé
  - Permet d'avoir du chiffrement mais c'est pas très robuste/polyvalent

Exemple type



Principe

- On relie deux réseaux avec pot. un adressage privé de manière transparente
- On va créer un lien point à point de niveau 2 ou 3 qui en fait sera assuré par un *autre* lien (généralement de niveau 3)

Et en pratique ?

Sur le routeur A

```
# ip tunnel add montunnel gre remote 172.16.2.1 local 172.16.1.1
# ip link set montunnel up
# ip addr add 10.0.0.1 dev montunnel
# ip route add 192.168.2.0/24 dev montunnel
```

Sur le routeur B

```
# ip tunnel add montunnel gre remote 172.16.1.1 local 172.16.2.1
# ip link set montunnel up
# ip addr add 10.0.0.2 dev montunnel
# ip route add 192.168.1.0/24 dev montunnel
```

Autres exemples

Versions plus hierarchisées de l'exemple type

Un site satellite veut avoir le réseau comme dans le site central
Un individu itinérant veut avoir tout le réseau comme s'il était au boulot

Tunnels plus exotiques ... à des fins moins avouables

IP sur :

- ICMP
- SMTP
- DNS
- HTTP
- etc.

Objectif : émettre/recevoir tout ce qu'on veut alors qu'on a essayé de nous restreindre à un protocole

Plan

Définition

Version courte

Interconnexion de réseaux locaux via un tunnel sécurisé

Oui mais commment

- On met en place un système d'authentification permettant d'appliquer une politique d'accès par rapport à qui peut se connecter au VPN (ça peut être juste deux points en statique ou un groupe de personnes venant d'endroits divers).
- On utilise un protocole dédié (PPTP, L2TP, IPSec, MPLS) pour qu'il y ait encapsulation, chiffrement et contrôle d'intégrité
- On configure le routage au niveau des deux points du tunnel pour connecter les deux réseaux puissent communiquer

PPPoE (PPP over Ethernet)

PPP

Gestion de l'initialisation et du contrôle de connexion

Sécurité

Authentifications PAP CHAP EAP

- PAP seul a éviter sur un lien non protégé
- CHAP a éviter sauf si on est vraiment sûr de l'entropie du mdp
- Souvent on a EAP SRP-SHA1 qui est très bien a préférer

Chiffrement Microsoft Point-to-Point Encryption protocol (MPPE)

- Basé sur RC4 (comme WEP) ce qui est très mauvais
- Choix de clé : 40 ou 128 bits (éviter 40 bits ...)
- Attaques permettent de modifier les données (bit-flipping)
- Pas cassé en soit mais attaqué en pratique (dans PPP)

Point-to-Point Tunneling Protocol (PPTP)

Principe

- On crée un canal de contrôle de la connexion (par TCP 1723)
- On crée un tunnel PPP dans lequel on encapsule les données
- On envoie les messages PPP en les envoyant par GRE (prot. IP 47)

Sécurité

Initialement conçu pour utiliser la sécurité fournie par PPP
Couche de sécurité ajoutée, en pratique considéré cassé (comme PPP)

PPP over SSH

Un VPN fait maison

Utilisation de deux outils courants

- pppd, démon permettant de mettre en place un tunnel PPP
- ssh/sshd, client/serveur SSH permettant de chiffrer ce tunnel

Simple et plutôt efficace (IP sur TCP est pas super, mais SSH compresse)

Et en pratique ?

- Simple parce qu'on peut le faire avec des outils et des protocoles simples
- Pas simple dans le sens on peut le faire avec qqs *one-liners*

Il faut mettre en place le serveur (user dédié, attributions d'adresses, routage) et le client (configuration ppp)

Voir par exemple `http://nsd.dyndns.org/pppossh/`

Alternative plus simple : le VPN de OpenSSH (1/2)

Configuration de sshd_config

```
PermitRootLogin yes
PermitTunnel yes
```

Creation du tunnel chiffré (par PPP) depuis le client

```
sudo ssh -w 0:0 serveurSSH
```

Configuration des interfaces

Au niveau des deux ordinateurs

```
ip link set tun0 up
ip addr add adresseLocaleTunnel/32 peer addresseDistanteTunnel dev tun0
```

Maintenant on a un tunnel bidirectionnel qui marche. Depuis un bout on peut faire `ping adresseAutreBoutTunnel` et on reçoit la réponse de l'autre.

Alternative plus simple : le VPN de OpenSSH (2/2)

Exemple

machineA et machineB on construit un tunnel VPN
machineA veut accéder à réseauB

Routage et NAT

Sur machineB :

```
echo "1" > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Sur machineA :

```
route add -net reseauB tun0
```

DNS

Si reseauB utilise un DNS privé pensez à l'ajouter sur /etc/resolv.conf au niveau de machineA pour qu'elle puisse l'utiliser !

Autres options

OpenVPN

Tunnel en mode SSL/TLS (comme pour https)

IPSec

En mode tunnel ou transport combiné à un protocole permettant de faire un tunnel (L2TP, MPLS)

Plan

Sécurité

Authentification

- Manuelle ou PSK : pour tests seulement
- Signature X509 ou ECDSA : avec une PKI ou des certificats auto-signés
- Autres : Kerberos V5, NTLMv2 (méthodes d'authentification de Microsoft Active Directory)

Chiffrement

- DES : aucune sécurité (seulement pour tests)
- 3DES : sûr mais coûteux, préférer AES
- AES-{CBC,GCM}-{128,192,256} : GCM = confidentialité+intégrité !

Intégrité

- MD5 : dangereux (à n'utiliser que pour des tests)
- SHA-{1,256,384} : HMAC avec SHA (SHA-1 = 160 bits bien sûr)
- AES-{GCM,GMAC}-{128,192,256} : GCM fait les deux (GMAC=GCM sans chiffrement)

Échanges de clés

Échanges de clés : DH ou ECDH (Elliptic Curve Diffie Hellman)

Protocoles

Security Associations (SA)

Permet de se mettre d'accord sur les algorithmes utilisés, s'authentifier et échanger des clés

- Internet Security Association and Key Management Protocol (ISAKMP) framework : qui utilise pour l'authentification et l'échange de clés des algos variés
- pre-shared keys, IKE (v1 ou v2), Kerberized Internet Negotiation of Keys (KINK), ou IPSECKEY DNS records

Authentication Headers (AH)

En-têtes ajoutés aux paquets IP permettant d'authentifier le paquet (prot. intégrité, authentification et prot. anti-rejeu)

Encapsulated Security Payload (ESP)

Chiffrement d'un contenu (le payload) + en-tête donnant authenticité, intégrité, et prot. contre le replay (limitée)

Modes de fonctionnement

Mode transport

- Ajout d'une en-tête (AH ou ESP) entre l'en-tête IP et le payload IP (qui peut contenir une autre en-tête par exemple TCP)
- Chiffrement éventuel du payload IP initial
- Permet de ne rajouter que quelques octets

Mode tunnel

- Encapsulation du paquet IP original (ajout en-tête AH ou ESP et IP par dessus)
- Chiffrement, authentification, contrôle d'intégrité, et protection anti-rejeu éventuels pour l'intégralité du paquet IP original

Fin !

Des questions ?