ip-l2tp

Configuration des tunnels non-gérés statiques L2TPv3

Les commandes ip l2tp sont utilisées pour établir des tunnels statiques non-gérés L2TPv3. Pour les tunnels non-gérés il n'y a pas de protocole de contrôle L2TP donc aucun service en espace utilisateur n'est requis. Les tunnels sont créés manuellement sur le système local et sur le pair distant.

L2TPv3 est prévu pour le tunneling niveau 2. Les tunnels statiques sont utiles pour établir des liens réseaux via les réseaux IP quand les tunnels sont fixés. Les tunnels L2TPv3 peuvent transporter des données de plus d'une session. Chaque session est identifiée par un session_id et le tunnel_id du tunnel parent. Un tunnel doit être créé avant qu'une session puisse être créé dans le tunnel.

En créant un tunnel L2TP, l'adresse IP du paire distant est spécifié, qui peut être soit une IPv4 ou une IPv6. L'adresse IP locale utilisée pour atteindre le paire doit également être spécifié. C'est l'adresse sur laquelle le système local écoute et accepte les paquets de données L2TP du paire.

L2TPv3 définis 2 formats d'encapsulation de paquet : UDP ou IP. L'encapsulation UDP est la plus commune. L'encapsulation IP utilise une valeur de protocole IP dédiée pour gérer les données L2TP sans la couche UDP. Utiliser l'encapsulation IP seulement quand il n'y a pas de périphérique NAT ou firewall dans le chemin réseaux.

Quand une session Ethernet L2TPv3 est créée, une interface réseaux virtuel est créé pour la session, qui doit être configurée et activée, tout comme une autre interface réseaux. Quand une donnée est passée via l'interface, elle est passée dans le tunnel L2TP au paire. En configurant les tables de routage du système ou en ajoutant l'interface à un bridge, l'interface L2TP est comme un câble virtuel connecté au paire.

Établir un pseudo-lien ethernet non-géré L2TPv3 implique de créer les contexts L2TP manuellement dans le système local et sur le paire. Les paramètres utilisé sur chaque site doivent correspondre ou aucune donnée ne passera. Aucune vérification de consistance n'est possible vu qu'il n'y a pas de protocole de contrôle utilisé pour établir les tunnels L2TP. Une fois l'interface réseau virtuelle configurée et activée, les données peuvent être transmises, même si le paire n'a pas été encore configuré. Si le paire n'est pas configuré, les paquets L2TP seront supprimés par le paire.

Pour établir un tunnel L2TP non-géré, utiliser les commande **l2tp add tunnel** et **l2tp add session** décrite dans ce document. Puir configurer et activer l'interface virtuelle.

Noter que les tunnels non-gérés ne gérent que les trames ethernet. Si vous voulez gérer du trafic PPP (L2TPv2), vous avez besoin d'un serveur L2TP qui implémente le protocole de contrôle L2TP. Le protocole de contrôle L2TP permet d'établir des tunnels et session dynamiques et fournissent un détection et correction d'erreurs.

add tunnel Ajouter un tunnel

name NAME Définis le nom de la session de l'interface réseau. Défaut : l2tpethN

tunnel_id ID Définis l'id du tunnel, qui est une valeur entière 32-bits assignée au tunnel par le paire. La valeur utilisé doit correspondre à la valeur tunnel_id utilisée par le paire.

remote ADDR Définis l'adresse IP du paire distant. Peut être spécifié comme IPv4 ou IPv6.

local ADDR Définis l'adresse IP de l'interface locale utilisée pour le tunnel. Cette adresse doit être l'adresse d'une interface locale. Peut être spécifié comme IPv4 ou IPv6.

encap ENCAP Définis le type d'encapsulaion du tunnel (udp ou ip).

udp_sport PORT Définis le port source à utiliser pour le tunnel. Doit être présent quand l'encapsulation udp est utilisée.

udp_dport PORT Définis le port upd de destination pour le tunnel. Doit être présent quand l'encapsulation udp est utilisée.

del tunnel Supprime un tunnel.

tunnel_id ID Définis l'id du tunnel à supprimer. Toutes les sessions dans le tunnel doivent être supprimés avant.

show tunnel Affiche des informations sur les tunnels

tunnel_id ID Définis l'id du tunnel à afficher. Non spécifié, affiche tous les tunnels

add session Ajoute une nouvelle session à un tunnel

name NAME Définis le nom de la session de l'interface réseau. Défaut : l2tpethN

tunnel_id ID Définis l'id du tunnel, qui est une valeur entière 32-bits assignée au tunnel par le paire. La valeur utilisé doit correspondre à la valeur tunnel_id utilisée par le paire.

peer_session_id ID Définis l'id de session du paire, qui est une valeur entière 32-bits assignée à la session par le paire. La valeur utilisé doit correspondre au session_id utilisé par le paire.

cookie HEXSTR Définis une valeur cookie optionnel à assigner à la session. C'est une valeur 4 ou 8 octets, ex : 014d3636deadbeef. La valeur doit correspondre à la valeur de cookie définie par le paire. La valeur du cookie est incorporée dans les paquets de données L2TP et est vérifié par le paire. N'utilise pas de cookie par défaut.

peer_cookie HEXSTR Définis une valeur cookie du paire optionnel à assigner à la session. C'est une valeur 4 ou 8 octets, ex : 014d3636deadbeef. La valeur doit correspondre à la valeur de cookie définie par le paire. Il indique au système local quelle valeur cookie il s'attend à trouver dans les paquets L2TP reçus. N'utilise pas de cookie par défaut.

l2spec_type L2SPECTYPE Définis le type d'en-tête spécifique à niveau 2 de la session : none ou udp

offset OFFSET Définis l'octet dans l'en-tête L2TP où les données utilisateurs commencent dans les paquets de données transmis. Si définis, doit correspondre à la valeur peer_offset utilisé. Défaut : 0

peer_offset OFFSET Définis l'octet dans l'en-tête L2TP où les données utilisateurs commencent dans les paquets de données reçus. Si définis, doit correspondre à la valeur offset utilisé. Défaut : 0

del session Détruit une session

tunnel_id ID Définis l'id du tunnel dans lequel la session est localisée

session_id ID L'id de session à supprimer

show session Affiche des informations sur les sessions

tunnel_id ID Définis l'id du tunnel dans lequel les sessions sont affichées

session_id ID L'id de session à afficher

Exemples

Définir des tunnels L2TP et des sessions
site-A :# ip l2tp add tunnel tunnel_id 3000 peer_tunnel_id 4000 encap udp local 1.2.3.4 remote 5.6.7.8 udp_sport 5000 udp_dport 6000
site-A :# ip l2tp add session tunnel_id 3000 session_id 1000 peer_session_id 2000
site-B :# ip l2tp add tunnel tunnel_id 4000 peer_tunnel_id 3000 encap udp local 5.6.7.8 remote 1.2.3.4 udp_sport 6000 udp_dport 5000
site-B :# ip l2tp add session tunnel_id 4000 session_id 2000 peer_session_id 1000
site-A :# ip link set l2tpeth0 up mtu 1488
site-B :# ip link set l2tpeth0 up mtu 1488
Noter que les adresse IP, ports UDP et id de session/tunnel correspondent et inversés dans chaque site.
Configurer comme interface IP
site-A :# ip addr add 10.42.1.1 peer 10.42.1.2 dev l2tpeth0
site-B :# ip addr add 10.42.1.2 peer 10.42.1.1 dev l2tpeth0
site-A :# ping 10.42.1.2
Configurer comme interfaces bridgés
site-A :# ip link set l2tpeth0 up mtu 1446
site-A :# ip link add br0 type bridge
site-A :# ip link set l2tpeth0 master br0
site-A :# ip link set eth0 master br0
site-A :# ip link set br0 up
En utilisant les VLAN, définis un bridge par vlan et bridger chaque VLAN sur une session L2TP séparé
site-A :# ip link set l2tpeth0 up mtu 1446

site-A :# ip link add brvlan5 type bridge
site-A :# ip link set l2tpeth0.5 master brvlan5
site-A :# ip link set eth1.5 master brvlan5
site-A :# ip link set brvlan5 up

Ajouter l'interface L2TP à une bridge force le bridge à forwarder le trafic sur le lien L2TP comme avec une autre interface. Le bridge apprend les adresses MAC des hôtes attachés à chaque interface et forwards les frames d'un port à un autre. Les adresse IP ne sont pas assignés aux interface l2tpethN. Si le bridge est configurés correctement des 2 côté, il devrait être possible d'atteindre les hôtes dans le réseau bridgé du paire.

Quand des frames ethernet brut son bridgé via un tunnel L2TP, de grandes frames peuvent être fragmentés et forwardés comme fragment IP individuels au destinataire, en fonction du MTU de l'interface physique utilisé par le tunnel. Quand les frames ethernet gérent les protocoles qui sont réassemblés par le destinataire, comme IP, il n'y a pas de problème. Cependant, une telle fragmentation peut causer des problèmes pour les protocoles comme PPPoE où le destinataire s'attend à recevoir des frames ethernet exactement comme transmises. Dans de tels cas, il est important que les frames quittant le tunnel soient réassemblé en une seul frame avant d'être forwardé. Pour le faire, activer le tracking de connection netfilter (conntrack) ou charger manuellement le module degrag netfilter à chaque point de tunnel.

site-A :# modprobe nf_degrag_ipv4
site-B :# modprobe nf_degrag_ipv4

Les tunnels L2TPv3 non-gérés sont supportés par certains équipements réseaux. Dans Linux, les messages L2TP Hello ne sont pas supportés dans les tunnels non-gérés. Les message Hello sont utilisés par L2TP pour détecter les erreurs des liens pour pouvoir automatiser le rétablissement des tunnels dynamiques. Si un paire non-linux supporte les messages Hello dans les tunnels non-gérés, il doit être désactivé pour fonctionner avec Linux.

Linux utilise par défaut le type Layer2SpecificHeader par défaut comme définis dans le protocole L2TPv3 (rfc3931).