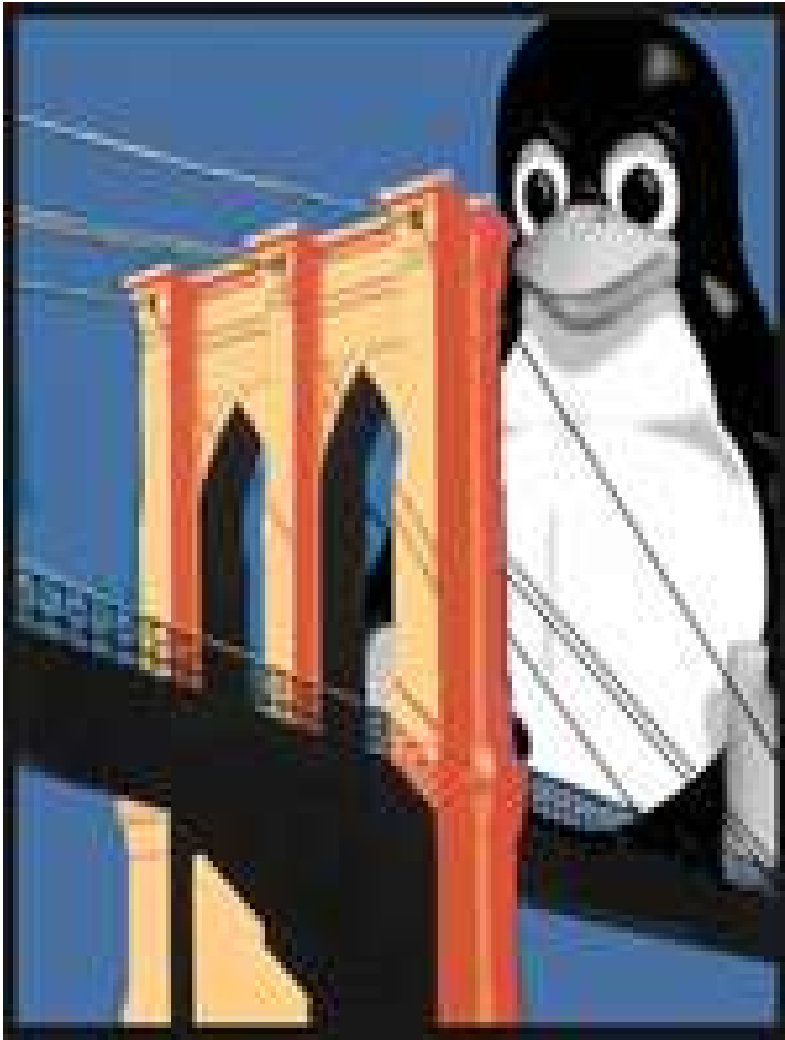


Mathrice - Lille - 21 octobre 2004



Pont-filtrant (bridge-firewall)

Retour d'expérience

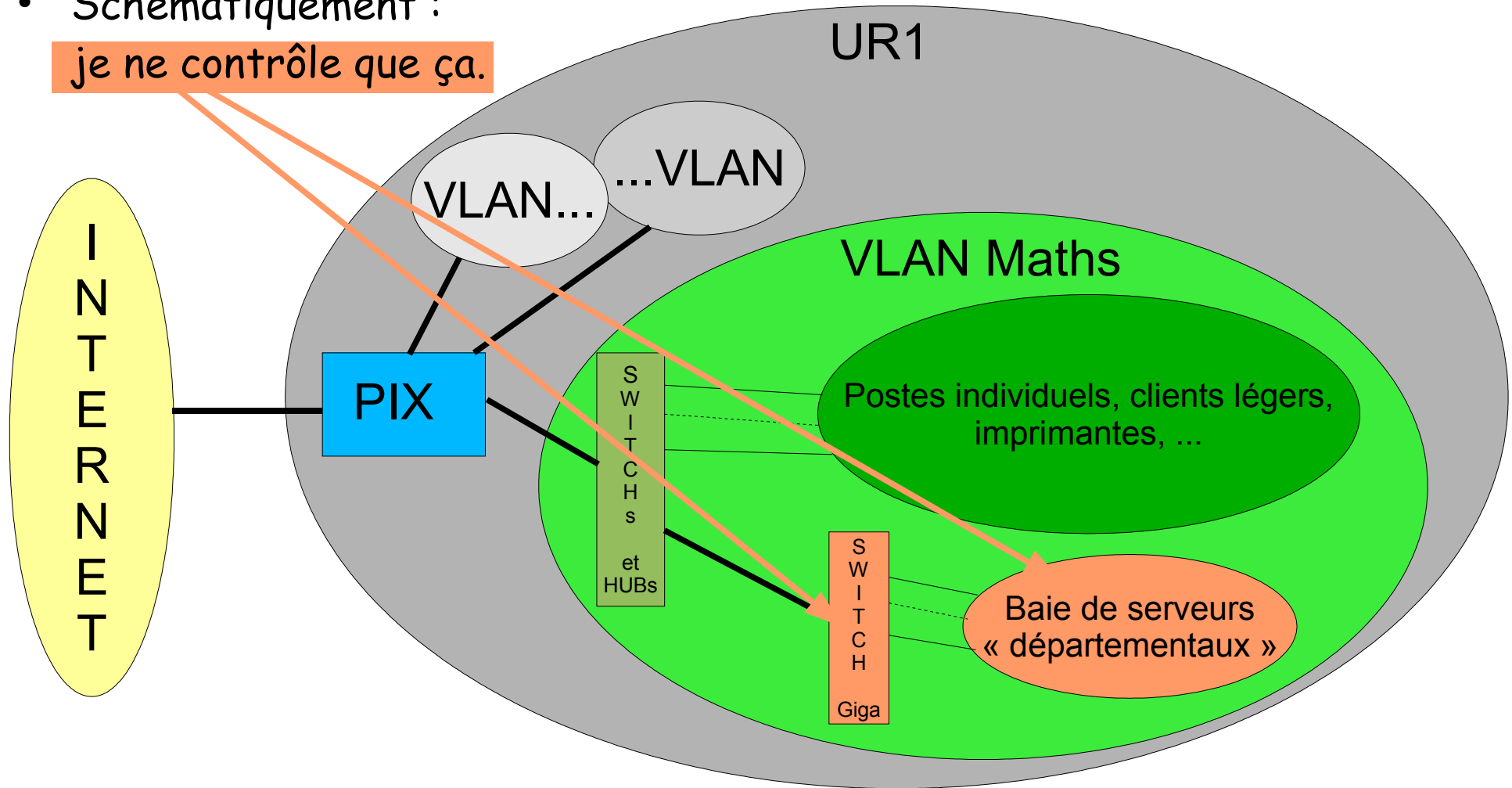


Contexte réseau local (avant)

- Réseau de campus entièrement géré par le CRI jusqu'à la prise.
- VLANs par labos, UFR, ... (beaucoup de commutateurs, mais encore des dizaines de HUBs).
- Filtrage Internet-campus fort (tout interdit en entrée sauf).
- Filtrage inter-VLANs faible, pas d'identification MAC à la prise.
- Un seul VLAN pour l'UFR (UMR) de Maths.
- Pas de possibilité (pour moi) de structurer le trafic au sein du VLAN.
- L'ensemble des serveurs Unix/Linux que je gère n'accède au VLAN que par un seul point, et j'ai installé un commutateur « privatif » entre-eux (commutateur Gigabit).

Contexte réseau local (avant)

- Schématiquement :
je ne contrôle que ça.





Contexte réseau local (avant)

- A priori, pas nécessaire d'installer un dispositif de filtrage puisque le CRI prend cela en charge en amont.
- Et jusqu'à récemment, cela n'avait pas été fait (bien qu'une machine ait été achetée pour cela, elle servait de plate-forme de tests).
- Mais divers incidents m'ont fait reconsidérer la question :
 - Problèmes de scans et autres dispositifs intrusifs depuis d'autres VLANs.
 - Plantages graves du serveur de fichier, cause pas déterminée, mais DoS possible, et pas de métrologie pour diagnostiquer.
 - Modification (à l'initiative de ma Direction) de la politique de filtrage Internet-VLAN_maths : mon VLAN ne devenait plus un sous-réseau de confiance... (des postes utilisateurs sont devenus ouverts en entrée sur Internet, suppression de l'obligation de passer par un délégataire SSH).



Contexte réseau local (avant)

- Le filtrage au niveau des serveurs devenait la seule réponse à ces problèmes.
- Mais le faire machine par machine n'était pas optimum :
 - Multiplication des sources de références
 - Pertes potentielles de performances (service de fichier Gigabit)
- Modifier la structure du VLAN maths et faire prendre en charge ce cas particulier par le CRI n'était pas réaliste (pas techniquement réalisable ?).
- Le garde-barrière « privatif » (tout comme il a été choisi un commutateur « privatif » pour disposer d'un « bus » Gigabit entre les serveurs) semble la solution évidente.

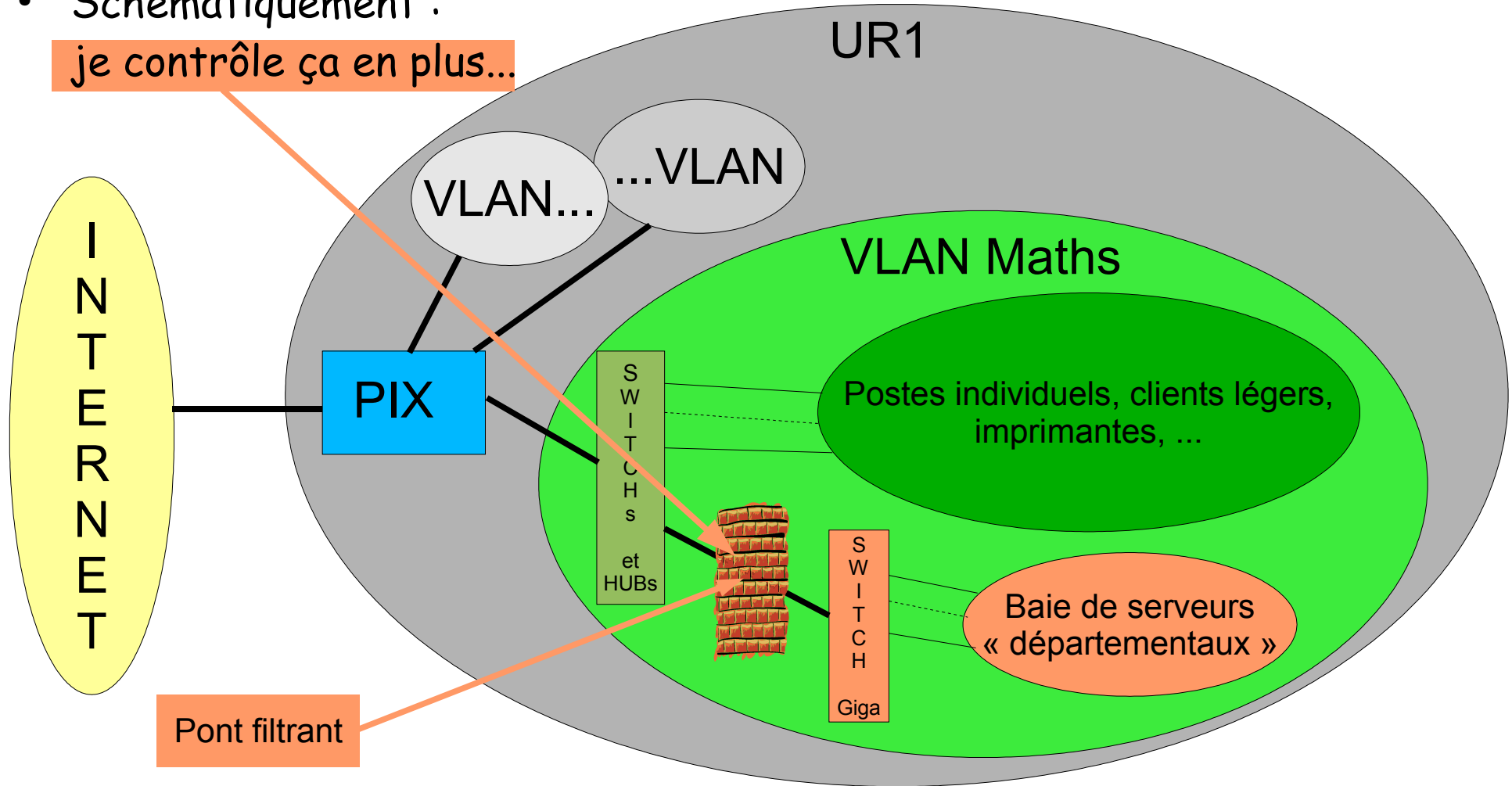


Contexte réseau local (avant)

- Mais, hors de question d'intervenir sur le VLAN lui-même.
 - Et encore moins sur le routage.
 - La technologie « pont-filtrant transparent » s'impose naturellement.
-
- Je décide donc d'installer ceci :

Contexte réseau local (après)

- Schématiquement :
je contrôle ça en plus...





Contexte réseau local (après)

- A la base, un pont-filtrant est... un pont.
- Donc, deux interfaces réseau, un « externe » et un « interne ».
- Il n'agit (en tant que pont) qu'au niveau 1 (ethernet)
- Mais en tant que dispositif filtrant, et avec la technique adoptée (Linux/netfilter-iptables), il pourra agir aux niveaux IP (niveaux avec un X car on peut aller jusqu'à ouvrir les paquets et traiter ce que l'on veut).
- Le pont peut avoir un (des) adresse(s) IP sur ses interfaces passant.
- Mais c'est un raccourci que je n'ai pas adopté :
 - Pour être vraiment transparent.
 - Pour être plus sûr et fiable.



Contexte réseau local (après)

- Il a donc été ajouté un interface dédié à l'administration du pont :
 - Surcoût vraiment faible (quelques dizaines d'€).
 - Avantages certains ; en parlant « iptables » :
 - avec ceci, les chaînes INPUT et OUTPUT ne sont présentes que sur l'interface d'administration.
 - La chaîne FORWARD n'est présente que sur les interfaces du « pont ».
 - Cela simplifie les règles et la gestion (et les fiabilise).
 - Cela permet l'administration du pont-filtrant sur un réseau dédié (non routé si on veut, connecté à rien d'autre ou disjoint mécaniquement du réseau local si on veut, ...).

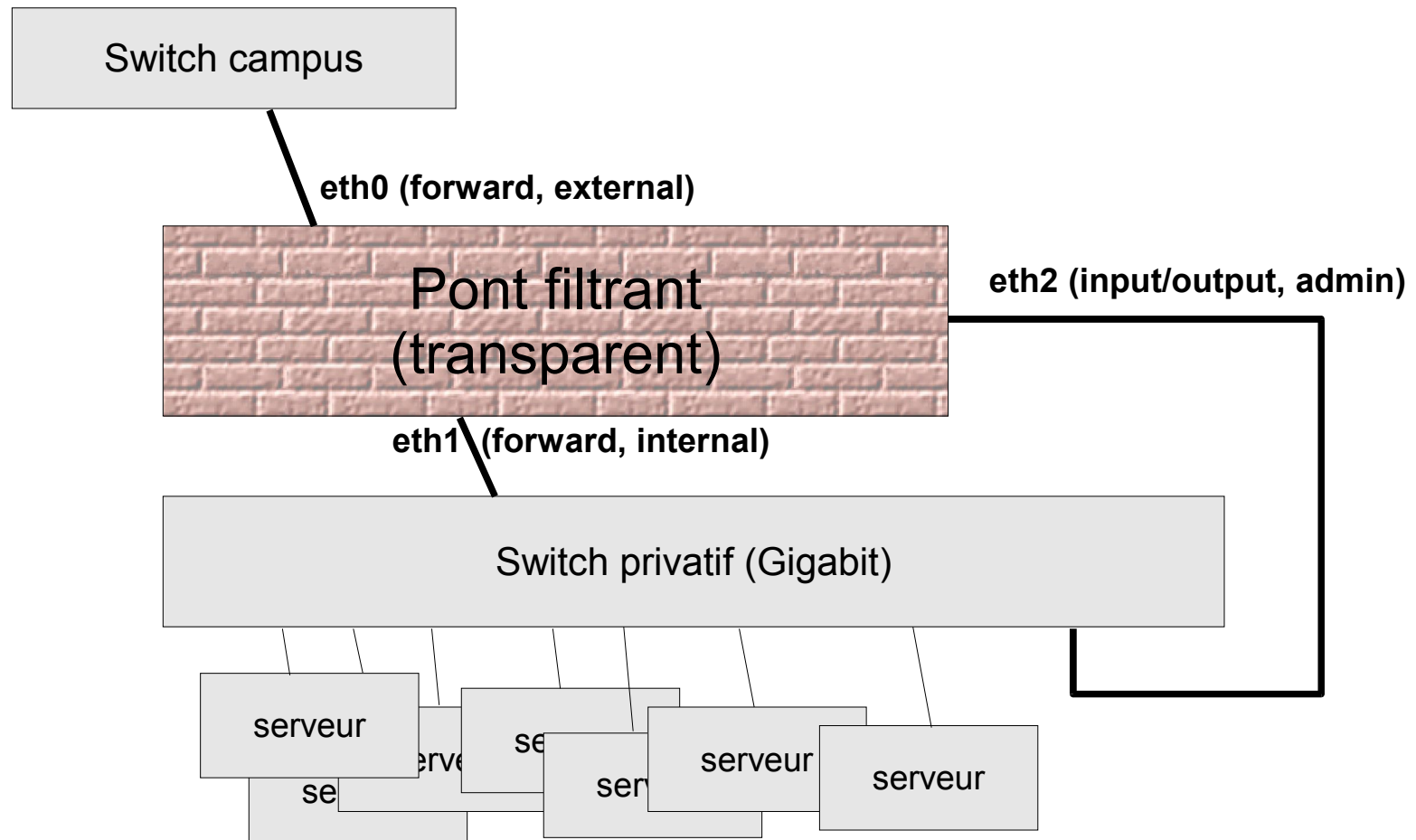


Contexte réseau local (après)

- Il n'y a pas d'interface « DMZ » :
 - Cela ne correspondait pas à un besoin en l'occurrence.
 - Mais la technique utilisée le permettrait.
- Il n'y a pas de prise en compte des VLANs (802.1q) :
 - Cela ne correspondait pas à un besoin en l'occurrence.
 - Cela aurait même été une difficulté (domaine de responsabilité du CRI).
 - Mais la technique utilisée le permettrait.
- Cela donne donc le schéma suivant :

Connectivité (détail)


- En détail, le schéma d'interconnexion du pont-filtrant est le suivant :






Mise en œuvre

- Pas de solution commerciale satisfaisante en « pont-filtrant ».
- Et pourquoi payer ce qui peut ne rien coûter...
- Plate-forme PC (DELL Poweredge 1650, 1Go de mémoire) :
 - Linux (2.4 ou 2.6).
 - ebttables (fonction bridge) (filtrage trames ethernet).
 - iptables (netfilter) (filtrage de paquets IP).
- Évaluation (rapide) de quelques solutions très intégrées (sentry, smoothwall, trustix, labwall, ...) :
 - Manque de souplesse pour faire exactement ce que je voulais...



Mise en œuvre

- Adoption de la distribution « Devil-Linux »
 - Indiquée la plus populaire par DistroWatch pour les firewalls.
 - Sans doute car bon compromis : ni trop, ni trop peu, base fiable et sérieuse, touche finale laissée à l'utilisateur (et facilement possible).
 - Version 1.2b2 rapidement abandonnée (pas assez stable, fonction bridge défectueuse).
 - Utilisation de la dernière version stable 1.0.6.
 - Dernière minute : devil-linux v1.2 sortie le 16 octobre 2004
- Tous les détails sur :
 - <http://www.devil-linux.org/>



Devil-Linux

- Boote depuis un CD :
 - Peu booter depuis une clé USB, mais mon PC ne pouvait pas.
 - On télécharge une image ISO, on grave, ça boote (et ça marche).
- Configuration (/etc) sur disquette ou clé USB :
 - Disquette : lente, faible capacité si on veut ajouter des fichiers importants (ajout d'outils par exemple). Peut être protégée en écriture.
 - Clé USB : plus rapide (bien en cours de mise au point...), plus grosse capacité. Mais plus chère, et penser à en avoir une avec un dispositif de protection écriture. Risque de se faire voler ?
- La configuration « définitive » peut être intégrée au CD bootable (jusqu'à prochaine modification...).



Devil-Linux

- Pas besoin de disque dur.
 - Mais disque dur possible : pour logs, pour ce que l'on veut... (utilisation de LVM)
- Tourne sur i486 et plus (prise en compte SMP).
- Support Netfilter/iptables (et ebtables).
- Fichier de configuration pour personnaliser les fonctionnalités du système.
- Connu de Firewall Builder (voir ci-après).
- Pas de desktop ni interface graphique.
- Support aisée de « cages » (chroot).



Devil-Linux

- Fonctions bridge ou routeur, filtrage (ethernet, IP) ou non.
- La plupart des binaires compilés avec le «GCC Stack Smashing Protector » ...
- ... et noyau modifié « GRSecurity » (divers mécanismes de protection, dont PAX)
- Démons et outils fournis (mais pas obligé de les utiliser !) :
 - Serveur : DNS (Bind), DHCP-relay, MAIL (Cyrus), HTTP (thttpd), FTP (vsftpd), VPN (freeswan, vtun, cipe, openvpn), NTP, Proxy (Squid)
 - IDS (Snort2), Antivirus (Clamav)
 - Logs avec syslog-ng
 - Encore plus en v1.2 (mais est-ce bien raisonnable ?)



Devil-Linux

- Devil-Linux **n'est pas** un firewall « près à l'emploi » :
 - Configuration à base de scripts et fichiers.
 - Pas d'interface graphique.
 - Il faut savoir ce que l'on veut et ce que l'on fait.
 - Et en particulier :
 - Savoir configurer les interfaces réseau.
 - Savoir utiliser ebttables et iptables en mode ligne.
 - Mais réutilisation de scripts existants possible.
 - Générateurs de règles (fwbuilder, ...) possible.
- Mais quand il marche, on sait comment et pourquoi !



Mise en œuvre

- Recommandation empirique :
 - Ne pas configurer un firewall directement « à la volée ».
 - Utiliser des scripts (rejouables, paramétrables).
 - Ou un outil de génération de règles.
 - Si on n'a pas de scripts :
 - Ne pas configurer le firewall directement « à la volée ».
 - Écrire des scripts (rejouables, paramétrables).
 - Si on n'aime pas écrire des scripts :
 - Ne pas configurer un firewall directement « à la volée ».
 - Identifier un générateur de règle (graphique si on préfère) et l'utiliser.
 - Il n'y a rien de plus facile que de rendre son firewall inefficace suite à l'utilisation de commandes iptables « à la volée »...

etables

- Fonctionnalité pont et filtrage au niveau trames ethernet.
- Patch pour noyaux 2.4, et intégré aux noyaux 2.6.
- Syntaxe très proche de celle de iptables.
- Voir « man etables » pour détails.
- On peut très bien laisser le bridge complètement passant, et ignorer etables, mais alors :

```
etables -A INPUT -j ACCEPT  
etables -A OUTPUT -j ACCEPT  
etables -A FORWARD -j ACCEPT
```

permet de verrouiller les règles en mode passant et procurant des compteurs.

etables

- Quelques courts exemples :

- bloquant comme un routeur Ipv4 :

```
etables -P FORWARD DROP
etables -A FORWARD -p Ipv4 -j ACCEPT
etables -A FORWARD -p ARP -j ACCEPT
...idem pour tables INPUT et OUTPUT
```

- Anti-spoofing :

```
etables -A FORWARD -p Ipv4 -ip-src 172.16.2.5 \
-s ! 00:11:22:33:44:55 -j DROP
```

- NATing :

```
etables -t nat -A PREROUTING -d 00:11:22:33:44:55 -i eth0 \
-j dnat -to-destination 00:66:77:88:99:AA
```



iptables

- On ne va pas (trop) revenir dessus, (re)voir présentations passées.
- En bref cependant...
- Filtre de paquets « statefull » : une heuristique interne détermine des « sessions » (états NEW, RELATED, INVALID, ESTABLISHED) en faisant du suivi de connexions (connection-tracking), aussi bien en TCP (trivial) qu'en UDP ou ICMP (association paquet retour) :
 - Ce mécanisme prend de la mémoire, compter 8192 entrées possibles par tranches de 128Mo de RAM (donc, c'est bien d'avoir de la mémoire dans un firewall statefull !). Réglage dans `/proc/sys/net/ipv4/ip_conntrack_max`
 - Et du CPU : sur un Pentium 1Ghz, une charge de fond d'au moins 15% semble en être la conséquence.



iptables

- Pour voir ce suivi de connexion :

- `cat /proc/net/ip_conntrack`

gaffe... y'a du monde...(des milliers d'entrée si un serveur web est derrière par exemple).

- `connwatcher.pl`

perl script qui permet de voir en temps réel.



iptables

- Quelques pointeurs pour documentations netfilter/iptables :
 - <http://www.iptables.org/>
 - <http://www.linuxguruz.com/iptables/>
(plein de docs, tutoriaux, howtos, astuces, scripts, autres pointeurs, ...)
 - <http://iptables-tutorial.frozentux.net/>
(très bon tutorial/manuel iptables)



Mise en œuvre

- Phase de préparation :
 - Installation matérielle du PC.
 - Connexion de ce PC derrière le commutateurs (machine autonome).
 - Configuration de Devil-Linux correspondante à l'architecture retenue :
 - Démarrage du mode bridge.
 - Configuration des interfaces ethernet.
 - Scripts firewall minimum (structure déjà élaborée, mais pas de politique de filtrage, tout passe, sauf sur l'interface d'administration pour laquelle l'accès n'est autorisé qu'à un seul poste (le mien)).
 - Validation avec un mini-switch sur l'interface « interne » et connexion de quelques postes de tests (mon portable, un client-leger, un des serveurs pas essentiel, un serveur utilisateur « à la volée »).
 - Ça marche, phase suivante...

Mise en œuvre

- Phase 1 :

- Mise en production (c'est à dire câblage définitif), mais sans politique de filtrage (sauf interface admin).
- L'opération a été faite « à la volée » :
 - Pas de douleur... pas pire qu'un reboot de switch ou routeur...
- Attention, pour ne pas casser les connexions TCP déjà ouvertes avant, il ne faut pas avoir de règle du type :

```
iptables -A FORWARD -p tcp ! --syn -m state NEW -j DROP
```

(recommandation valable aussi pour le restart ou reboot du firewall)

- Bien entendu, le suivi de connexion est déjà présent, cela permet de valider l'ensemble, mais sans règle autre que :

```
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
iptables -A FORWARD -m state --state NEW -j ACCEPT
```



Mise en œuvre

- Phase 1 (suite) :
 - Cela permet en particulier de voir la tenue de la machine en charge.
 - Quelques constats :
 - Charge CPU de fond de l'ordre de 15% (PIII-1Ghz) avec un peu de trafic.
 - Le reload/restart du firewall seul (exécution d'un script de config) est sans aucune douleur, quelques fractions de secondes.
 - Rebooter le PC prend de 2mn à 3mn : pas un drame (pas de connexions perdues), pas sensible quand même (le plus long étant le test mémoire du BIOS, et le boot sur CD, plus lent que sur disque).
 - Tient la charge en Gigabit (i.e. 200Mb/s à 400Mb/s)
 - J'ai laissé dans cet état deux mois, par prudence.



Mise en œuvre

- Phase 2 :
 - Comme tout marchait bien, pas de modification.
 - Configuration de filtres correspondant aux besoins.
 - Ajout/configuration de quelques outils :
 - syslog-ng (version updatée) : logs iptables sur fichiers séparés, sur disque local.
 - fwlogwatch/fwlogsummary (produit des synthèses des logs du firewall, publication en HTML, serveur accessible uniquement de la station administrateur).
 - Pour ajouter des fichiers, et ne pas regraver le CD de la distribution, création d'un répertoire /etc/sbin, copie des nouveautés dedans, elles se retrouvent sauvées sur la disquette/clé-USB avec la configuration.
 - Le pont-filtrant est bien transparent ! (pas de trace de son adresse MAC externe dans les logs et tables ARP du commutateurs campus).



Mise en œuvre

- Pour appliquer la politique de filtrage :
 - Scripts modulaires maisons.
 - Mais il faut apprendre à utiliser `iptables` en mode ligne
 - Firewall Builder : pour le moment, je n'utilise pas `fwbuilder` pour charger automatiquement la politique, mais pour la formaliser (interface graphique), et pour reporter les règles générées dans mes propres scripts.
 - Il faut apprécier si ne reposer que sur un outil graphique ne constitue pas un handicap potentiel en période de crise par rapport à des scripts maîtrisés modifiables à la console système avec `vi` ... (approche lowtech vs hightech).

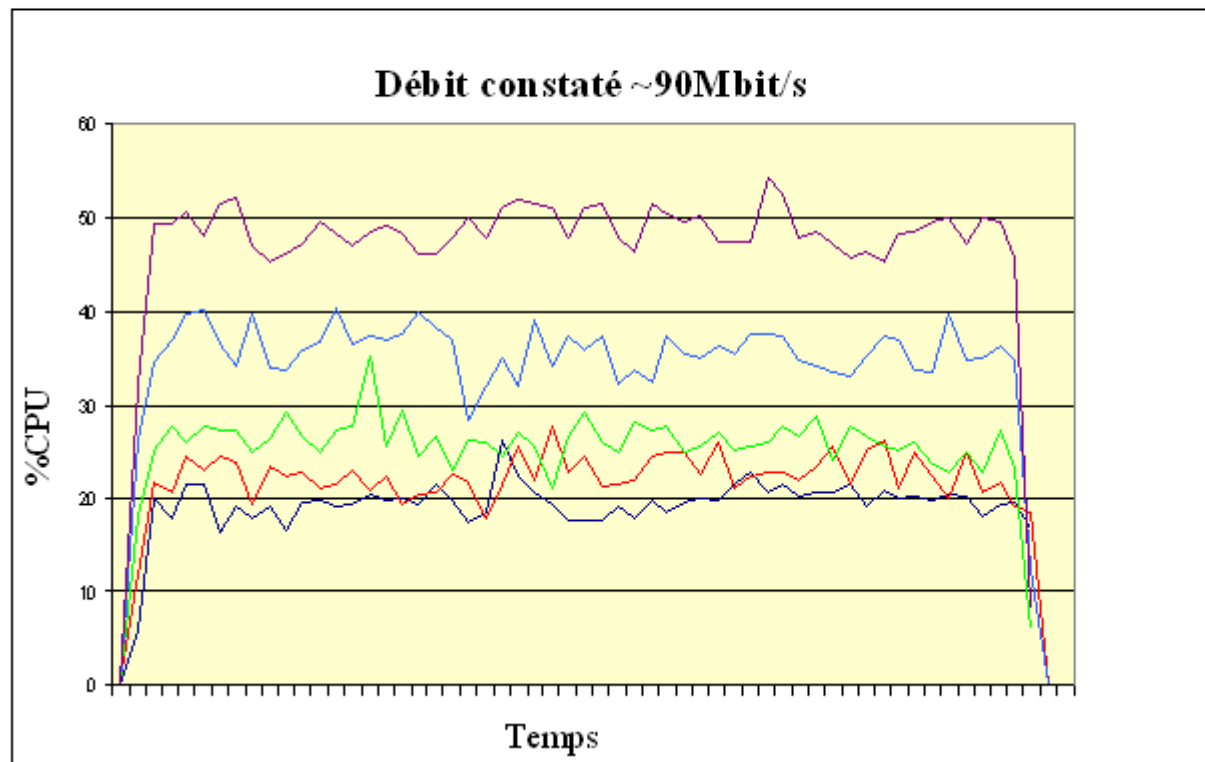


Performances

- Pas de perte de débit notable constatée :
 - Débit moyen = 20Mo/s (pour 28K-paquets/s) (deux sens cumulés) (beaucoup de trafic X11 avec clients-légers).
- CPU chargé (minimum 15-20%), mais avec une dérivée lente quand il y a des pointes de trafic (avec peu de règles et peu de logs, je n'ai jamais dépassé les 30-35% de charge).
- Rapport d'environ un facteur 10x entre trafics sortant et entrant (sortant majoritaire) :
 - Le trafic disque (NFS) reste interne, ce résultat est donc très classique, similaire à celui d'un abonné vis à vis de son FAI (interne=client).
- Pour information, des mesures en provenance de :
<http://labwall.dr15.cnrs.fr/>

Performances

Netfilter: influence du nombre de règles sur le CPU

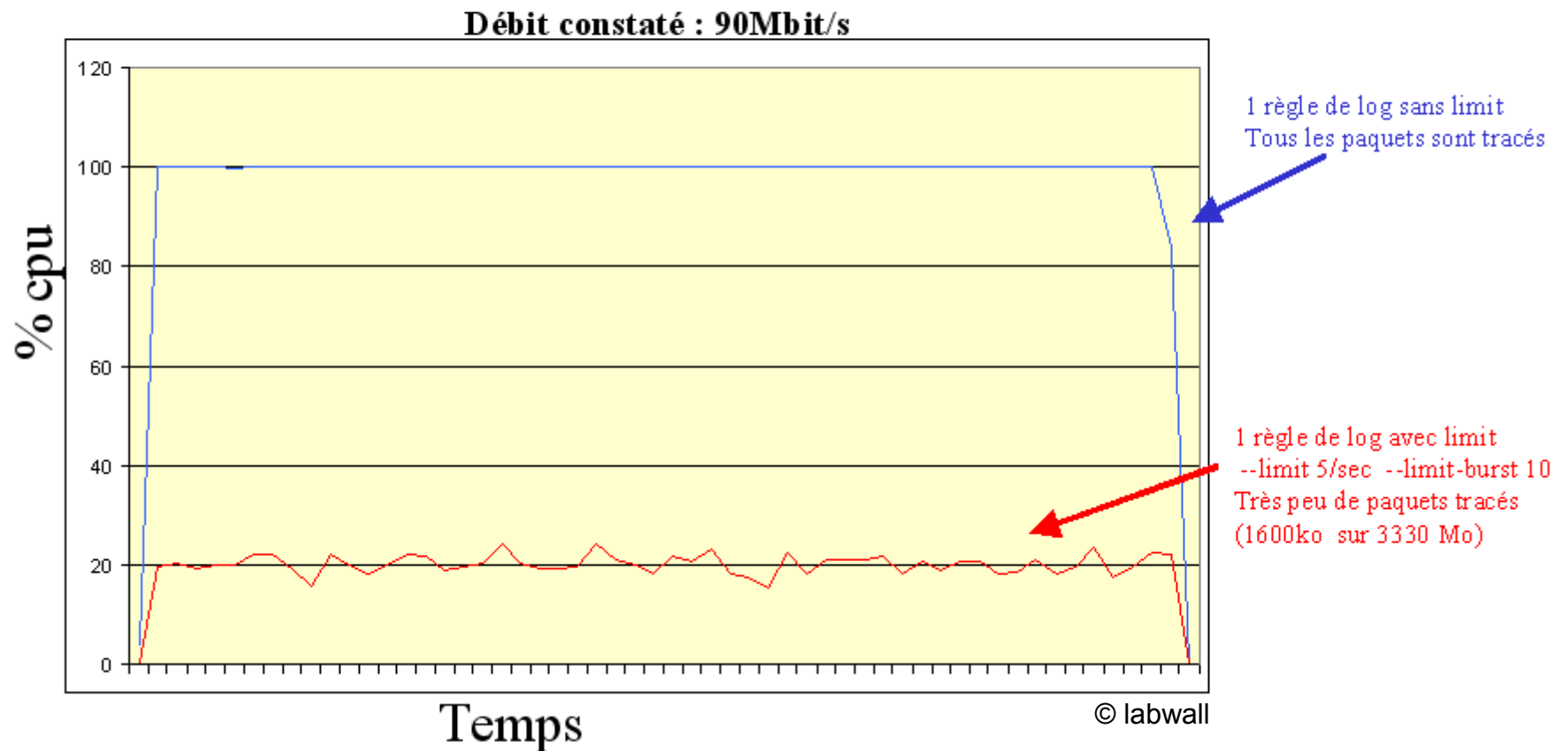


© labwall

1000 règles
500 règles
200 règles
100 règles
0 règle

Performances

Netfilter: influence d'une règle de log sur la consommation CPU





Bilan

- Solution finalement facile à mettre en oeuvre.
- Coût réduit à celui d'un PC bien né, pas forcément dernier cri (penser à avoir assez de mémoire), (bien) moins de 1000€.
- Pour un petit réseau desservi, peut tourner sur un PC de récupération.
- Vraiment transparent et « passif » vis à vis d'une infrastructure réseau type réseau « campus ».
 - Donc, solution idéale lorsque le déploiement si une autonomie de filtrage est souhaitée/nécessaire en sus d'une tutelle réseau qui ne peut la prendre en charge.
- Extensible, et grâce à sa furtivité, sans doute le firewall le plus « sûr », y compris en environnement routé.



Firewall Builder

- Outil graphique de gestion de politique de filtrage sur firewalls
- <http://www.fwbuilder.org/>
- Le GUI créer/utilise un fichier de configuration du/des firewall(s) au format XML.
- Des commandes (en mode ligne ou appelées par le GUI) compile ce fichier XML pour produire des scripts exécutables (et téléchargeables via ssh/scp) par les firewalls :
 - netfilter/iptables (Linux)
 - Ipfiler (*BSD, Solaris)
 - OpenBSD PF
 - Cisco PIX (commercial)



Firewall Builder

- L'interface graphique est agréable, pas trop mal pensé.
- On crée ses propres objets, cela permet de bien formaliser, visualiser la politique de filtrage.
- Mais pas d'impression/rapport graphique... dommage.
- Quelques détails énervants...
- Compileur pertinent (pas d'erreur majeure), mais stratégie éventuellement déroutante, optimisation (nombre de règles traversées, utilisation de tables supplémentaires) très faible.
 - Ça marche, mais cela n'écrit pas ce que l'on aurait fait à la main (du coup, relire un `iptables-save` n'est pas évident...).
 - Mais est-ce vraiment un problème ... ?



Firewall Builder

- Versions Unix GPL
- Versions Windows et MacOSX commerciales (évaluation 30j)
- Compilateurs de règles GPL
 - Sauf PIX, commercial
- Pas beaucoup d'autres (pas du tout ?) produits équivalents (pour iptables) actuellement :
 - On trouve soit des firewalls tout intégrés, avec interface GUI ou Web, mais moins « adaptables »
 - Des configureurs qui ne font rien d'autre que générer des lignes de commandes iptables, et/ou plutôt adaptés à des firewalls personnels.
 - Je regrette toujours la disparition de NP-lite...

Firewall Builder

- Présentation / démonstration en direct...

The screenshot displays the Firewall Builder application window titled "Firewall Builder: test.fwb". The interface includes a menu bar (File, Edit, Object, Rules, Help), a toolbar, and a left-hand navigation pane showing a tree structure with folders for "Firewalls", "Objects", "Services", and "Time". The main area is divided into a rule table and a configuration panel on the right.

Policy	outside	inside	loopback	dmz	NAT				
Source	Destination	Service	Action	Time	Options	Com			
0	net-192.168.1.0	test2	TCP ssh	Accept					
1	test2	internal server	DNS	Accept					
2	Any	test2	Any	Deny					
3	Any	Any	TCP auth	Reject					
4	Any	server on dmz	TCP smtp	Accept					
5	server on dmz	internal server	TCP smtp	Accept					
6	server on dmz	net-192.168.1.0	DNS SMTP	Accept					
7	net-192.168.2.0	net-192.168.1.0	Any	Deny					
8	net-192.168.1.0	Any	Any	Accept					
9	Any	Any	Any	Deny					

The configuration panel on the right shows the "Firewall" settings for "test2". It includes tabs for "General", "Templates", and "SNMP". The "General" tab is active, showing fields for "Name: test2", "Library: User", "Platform: iptables", "Version: - any -", "Host OS: Linux 2.4/2.6", and a "Comment" field. An "Apply Changes" button is located at the bottom right of the panel.



Des questions ?

