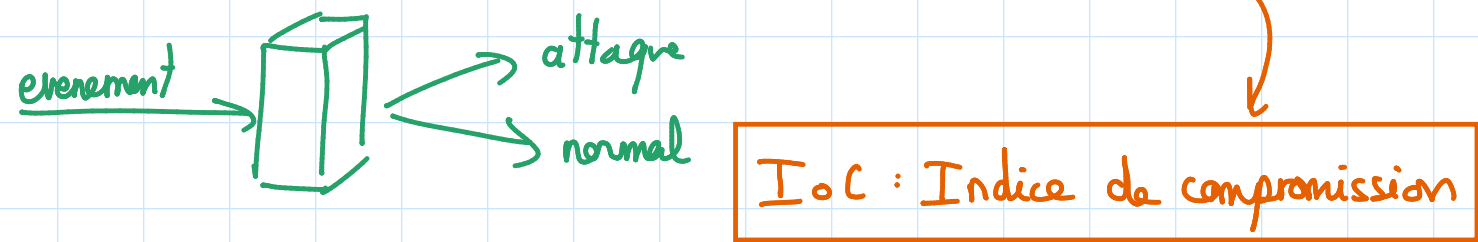


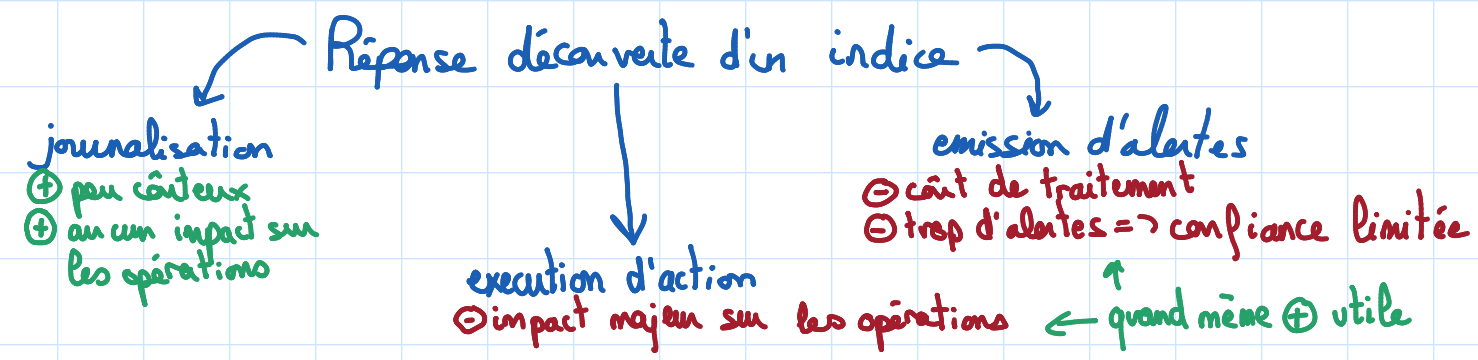
Intrusion: violation de la politique de sécurité
Attaques: tentative d'intrusion
Scénario: Suite des étapes élémentaire d'une attaque
Vulnérabilité (faille): Défaut (bug) exploité par l'attaquant pour mettre en œuvre son attaque

Détection d'intrusion: identifier les indices permettant de croire qu'une intrusion a lieu ou a eu lieu.



IDS
Provenance des données: noyau OS, hôte, réseau...
Méthode d'analyse: recherche de signature, détection d'anomalies...
Réponse à la découverte d'un indice: Journalisation, émission d'alertes, exécution d'actions...

Analyse de signature: correspondance de caractère, expressions régulières, machine à état → à partir d'une BDD de vuln connues ou usages non appropriés
⊕ pas de faux positifs ⊖ mise à jour constante
Analyse sur détection d'anomalie: différence à une norme de référence
→ modélisation de comportement, machine learning...
⊖ beaucoup de faux-positifs



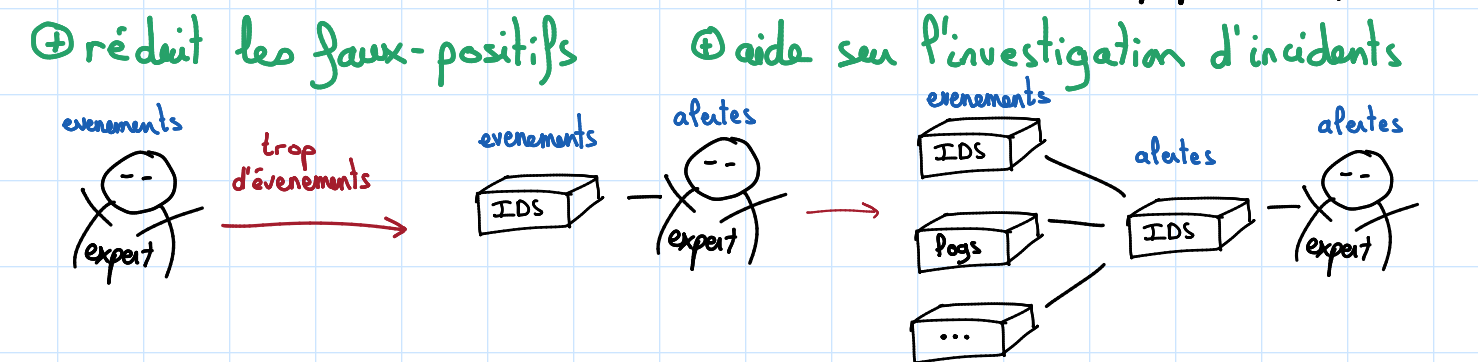
Host-based IDS (HIDS): fichiers, journaux, événements
→ recherche de signatures dans fichiers/registres
→ contrôle d'intégrité fichiers/repertoire
→ analyse comportementale des habitudes utilisateurs
} émissions d'alertes, blocage d'opérations

Kernel-based IDS (KIDS): appels système
→ analyse comportementale des séquences d'appels
} journalisation, émission d'alertes, blocage d'opérations

Network-based IDS (NIDS):
→ recherche de signature dans charge utile
→ contrôle de conformité des protocoles rzo
→ analyse statique de l'utilisation rzo
→ analyse comportementale des interconnexions entre hôtes
} ajout de règles pare-feu, journalisation, perte de paquets rzo, émission d'alertes

Intrusion Prevention System (IPS):
→ IDS mais réponses à une détection est une action
} bloquer l'attaque (pare-feu), réplique intrusive vers l'attaquant

Security Information & Event Management (SIEM):
→ version moderne d'un HIDS+NIDS
→ centralise et corréle les événements avec d'autres équipements (pare-feu)
⊕ réduit les faux-positifs
⊕ aide sur l'investigation d'incidents



construction d'une signature:

- **pré-requis**: connaître la vuln et la manière de l'exploiter.
- **objectif**: identifier les indices qui caractérisent l'attaque
- **conception**: l'emplacement des indices influence le choix du type d'IDS et les besoins en terme de définition de signature

signatures statiques

- regex, motifs, valeur, chaîne de car.
- entête de protocole
- heuristique

signatures dynamique

- Directive particulière
- Machine à état... (et l'automatique dans tout ça?)
- Langage "Turing-complet"

détection d'anomalie

↳ **spécification de référence**

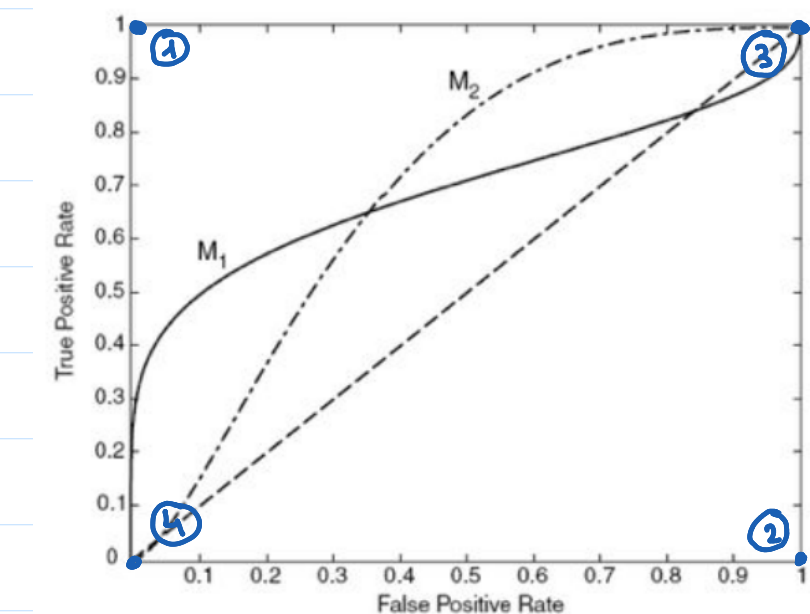
- * **contrôle d'intégrité**: fonction de hachage pour vérifier ou créer les signatures de fichiers de conf., base de registre, répertoires...
- * **respect d'un format/norme**: formats de fichiers, entêtes protocoles rzo
- * **respect d'un cadre d'utilisation**: vérification des valeurs possibles qu'un champs peut prendre

↳ **base sur un modèle**

- * **modèle statistique**: vérifier que valeur ∈ (moyenne ± écart-type)
- * **modèle comportemental**: structures de données (graphes, séquences...), ML...
↳ **défis**: disponibilité et représentativité du jeu de données, comment faire évoluer le jeu de données, comment traiter les événements rares.
↳ **conception**: préparation des données, préparation du modèle, apprentissage, évaluation/validation, utilisation.

Matrice de confusion	Predicted	
	negative (normal)	positive (attack)
Actual	TN	FP
	FN	TP

Mesures de base: $TPR = \frac{TP}{TP+FN}$; $FPR = \frac{FP}{FP+TN}$; $TNR = \frac{TN}{TN+FP}$; $FNR = \frac{FN}{FN+TP}$
Precision: si le système dit positif, ça l'est $\frac{TP}{TP+FP}$
F-Measure: balance entre détection et précision: $\frac{2 \cdot TP}{2 \cdot TP + FP + FN}$



ROC curve
on fixe l'un des 2 paramètres et on regarde comment ça réagit à la variation de l'autre
↓
indicateur de performance
↓
permet la comparaison

① **IDS parfait**

Matrice de confusion	Predicted	
	negative (normal)	positive (attack)
Actual	950	0
	0	50

TPR: 100%
FPR: 0%
TNR: 100%
FNR: 0%

② **Pire IDS**

Matrice de confusion	Predicted	
	negative (normal)	positive (attack)
Actual	0	950
	50	0

TPR: 0%
FPR: 100%
TNR: 0%
FNR: 100%

③ **Le TAP** → tout positif

Matrice de confusion	Predicted	
	negative (normal)	positive (attack)
Actual	0	950
	0	50

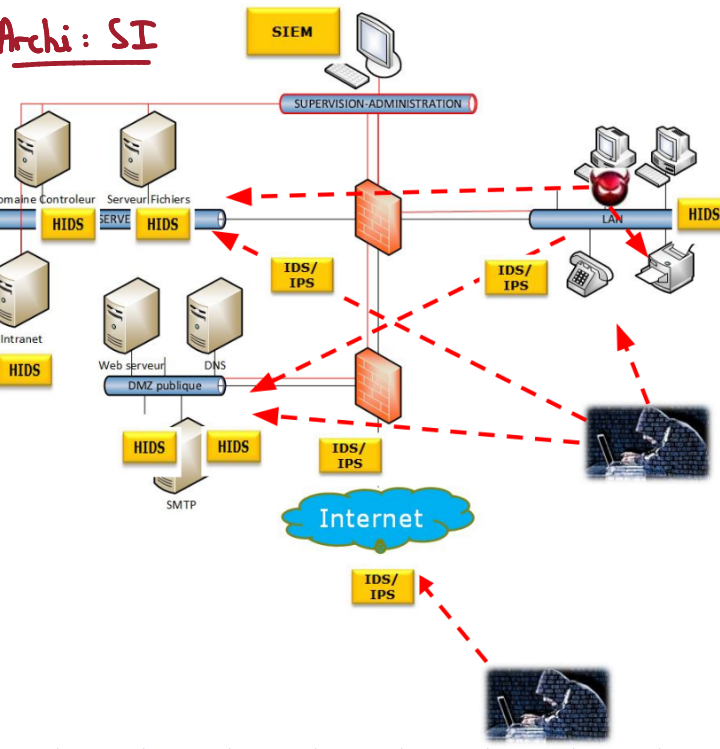
TPR: 100%
FPR: 100%
TNR: 0%
FNR: 0%

④ **Pas d'IDS**

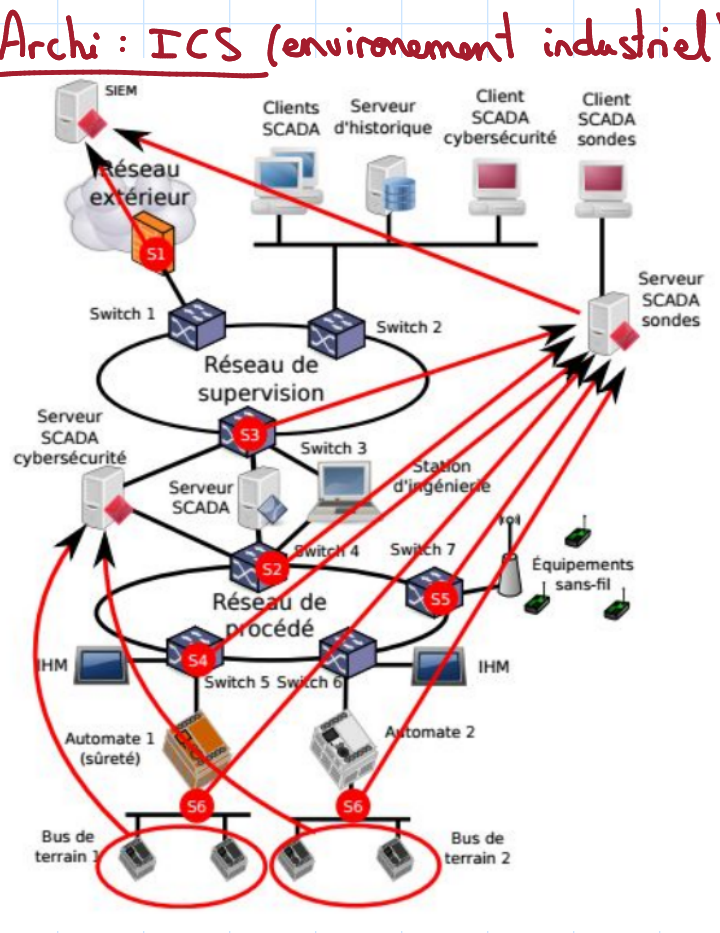
Matrice de confusion	Predicted	
	negative (normal)	positive (attack)
Actual	950	0
	50	0

TPR: 0%
FPR: 0%
TNR: 100%
FNR: 100%

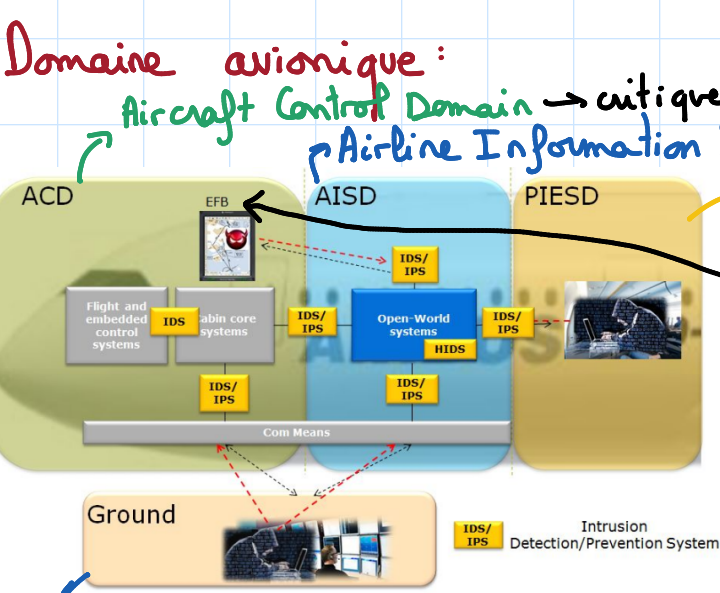
autres critères sur les IDS : performance débit max traitable, coût solution, coût intégration, adaptation à l'environnement, fréquence des mises à jour des signatures, qualité des alarmes/rapports, perf...



- firewall externe → statistique
 - NIPS en coupure
 - NIDS en port mirroring ou tap
 - HIDS sur les postes clients/serveur
 - SIEM depuis le Pro supervision
- détection :
- IPS (mode signature)
 - IDS (mode comportementale)



- sources de menaces : corruption ou erreur durant la maintenance, att. rzo
- positionnement :
- interco Rzo ext et Sys indus (S1)
 - entre SCADA et Rzo procédé (S2)
 - interco Rzo supervision et SCADA (S3)
 - automates (S4, S5, S6 sur bus terrain)
 - centralisation alertes sur SCADA
- détection :
- détection des déviations
 - machine à états pour autoriser des règles entre clients et serveurs

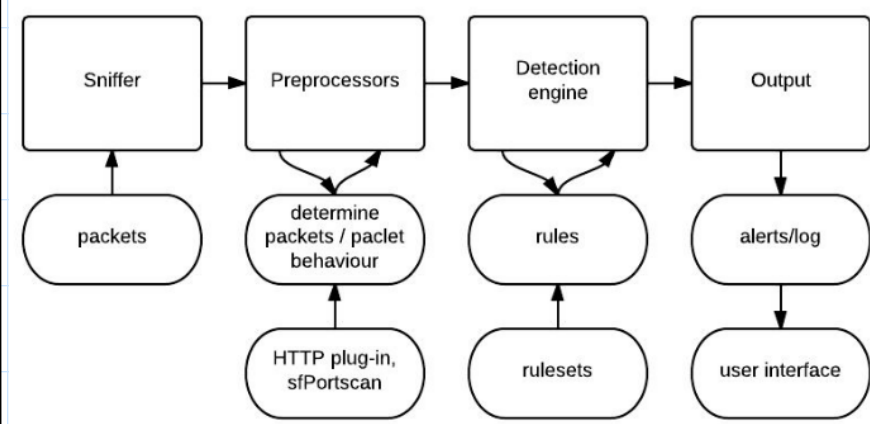


- Domaine avionique :
- Aircraft Control Domain → critique car contrôle l'avion
 - Airline Information System Domain
 - Passenger Information and Entertainment Domain
- EFB : équipement mobile d'aide au pilote pour la préparation de vol
- on va mettre des trucs :
- interfaces de comm, entre domaines
 - sur les systèmes à protéger
 - sur le bus avionique en mode IDS
 - là où les ressources suffisent

- IDS ou IPS :
- ACD : faux positifs a des conséquences (safety et opérationnel) si mauvaise précision.
 - les ACD sont temps réel donc un HIDS est difficile à intégrer + latences IPS
- techniques de détection :
- ACD : pas d'historique d'attaques donc approche comportementale [semi ou non] - supervisé
 - AISD : protocoles standards donc signatures

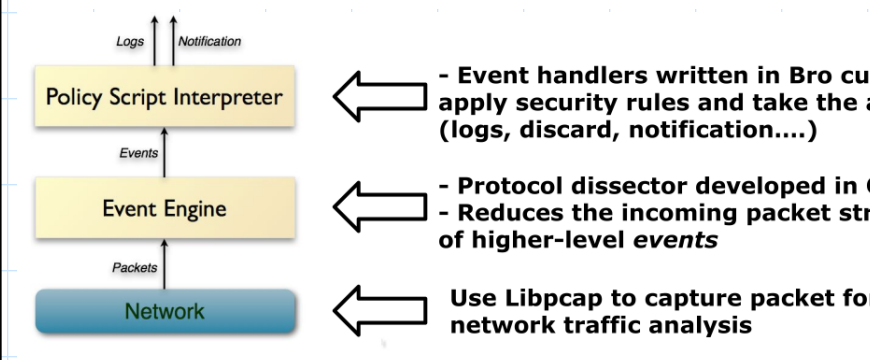
- Tendances : Firewall NexGen et UTM (Unified Threat Management)
- firewall rzo
 - anti spam
 - antivirus
 - filtrage applicatif
 - web reputation
 - fonction DLP (Data Leak Protection)
 - NIPS

solutions Open-source : Snort, Zeek (anciennement Bro IDS), Suricata, Security Onion...



Snort

- real time, intrusion detect°
- packet logging on IP network



Zeek

- Event handlers written in Bro custom script to apply security rules and take the appropriate action (logs, discard, notification....)
- Protocol dissector developed in C/C++ (or binpac) - Reduces the incoming packet stream into a series of higher-level events
- Use Libpcap to capture packet for passive network traffic analysis

Security Onion (linux distro)

↳ intrusion détection, network security monitoring, log management

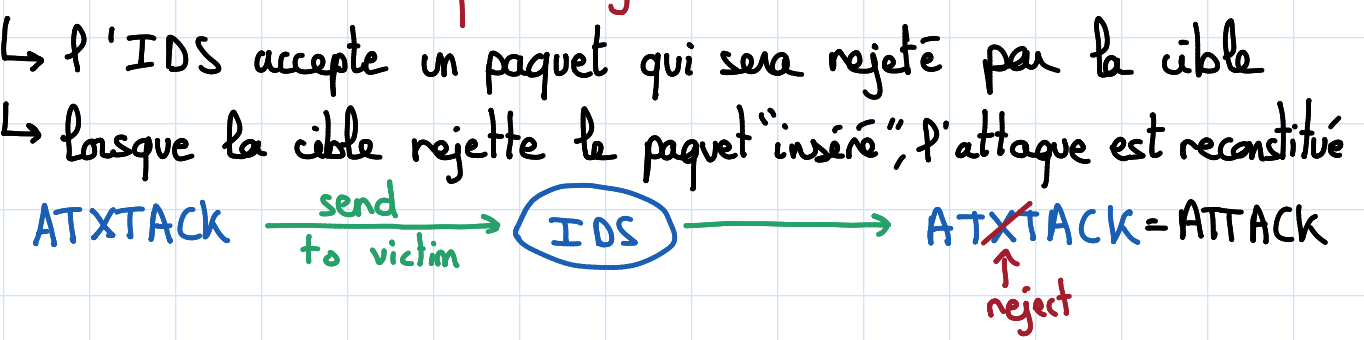
IDS inefficace si flux chiffré

- ↳ on casse le chiffrement avec un proxy en MITM
- ↳ sinon on regarde les autres infos : tailles des packets, sens de la communication, latence...

IDS contournable par encodage

↳ pour que l'IDS ne puisse rien interpréter (par ex ascii ↔ unicode)

IDS contournable par injection



IDS contournable par fragmentation

- ↳ fragmentation IP ou TCP de sorte qu'ils ne veulent rien dire seuls
- ↳ si l'IDS ne manipule pas les out-of-order fragments => vulnérable!

IDS bloquable par DOS

↳ noyer l'IDS sans les alertes

↳ si IPS => blocage rzo (fail-secure) ou laisse tout passer (fail-open)

VPN de openSSH

ssh -w 0:0 user@serveur

au niveau des 2 ordinateurs :

ip link set tun0 up

ip addr add adresseLocaleTunnel/32 peer adresseDistantTunnel dev tun0

↳ depuis l'un des bout on peut ping adresseAutreBoutDuTunnel

↳ cas particulier : machine A et machine B ont un tunnel VPN

machine A veut accéder à réseau B

sur machine A :

```
echo "1" > /proc/sys/net/ipv4/ip-forward
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
route add -net reseauB tun0
```

sur machine B :

+ si machine B utilise un DNS privé il faut penser à l'ajouter

autres options : open VPN (tunnel SSL/TLS), IPSec (L2TP, MPLS)

VPN : encapsulation de couche 2 ou 3 dans la couche 3

↳ "IP in IP" ~ 2 entêtes IP qui se suivent.

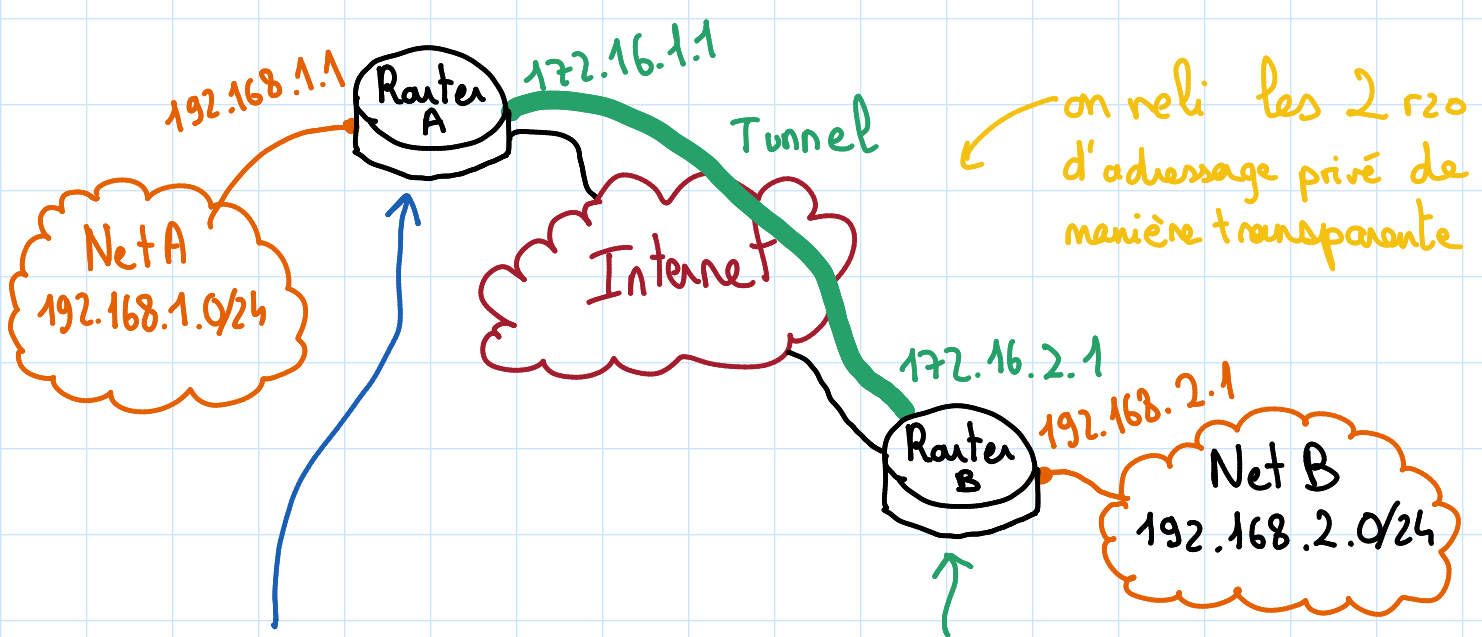
↳ mélange IPv6 et IPv4 possible

↳ Generic Routing Encapsulation (GRE)

↳ dans l'entête IP encapsulant : champ "GRE" qui indique le protocole encapsulé

↳ chiffrement mais pas très robuste/polyvalent

on peut encapsuler ce qu'on souhaite



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

en pratique

tunnel à des fin mais valable avec : ICMP, SMTP, DNS, HTTP
↳ pour contourner les restrictions

VPN : interconnection de réseaux locaux via un tunnel sécurisé

- ① mise en place d'un système d'authentification pour politique d'accès
- ② utilisation d'un protocole dédié (PPTP, L2TP, IPsec, MPLS) pour chiffrer, encapsuler et contrôler l'intégrité
- ③ configuration du routage au niveau des 2 points du tunnel

PPP over Internet

↳ PPP : gestion initialisation et contrôle de connexion

↳ Auth : • PAP → à éviter sur un lien non protégé

• CHAP → à éviter sauf si l'entropie du mdp est bonne

• EAP SRP-SHA1 → à préférer

↳ chiffrement Microsoft Point-to-Point Encryption protocol (MPPE)

↳ basé sur RC4 → mauvais

↳ choix de clé (40 ou 128 bits) → éviter 40

↳ attaques permettant de modifier les données (bit-flipping)

↳ pas cassé en soit mais attaqué en pratique

Point-to-Point Tunneling Protocol (PPTP) ~ utilise la série PPP, considéré cassé

① on crée un canal de contrôle de connexion (TCP 1723)

② on crée un tunnel PPP dans lequel on encapsule les données

③ on send les messages PPP en les envoyant avec GRE (protocole IP47)

PPP over SSH "VPN" fait maison

↳ 2 outils courants : pppd (démon), ssh/sshd

plutôt efficace car IP sur TCP pas super car SSH compressa

↳ simple car fait avec des outils/protocoles, pas simple car qqs one-liners

↳ mise en place serveur (user dédié, adresse, routage) et le client (PPP)

IPSEC

Authentification : { manuelle ou PSK : only 4 tests
signature X509 ou ECDSA : PKI ou auto-signés
autres : Kerberos, NTLMv2 (Microsoft auth. AD)

Chiffrement : { DES : aucune sécurité (4 tests)
3DES : sûr mais coûteux (use AES instead)
AES - {CBC, GCM} - {128, 196, 256} | GCM => confidentiel + intégrité

Sécurité

Intégrité : { MD5 : dangereux
SHA-1, 256, 512 : HMAC avec sha-1 (sha-1 : 160 bits)
AES - {GCM, GMAC} - {128, 196, 256} | GCM fait les 2 (GMAC = GCM sans chiffrement)

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

Security Associations (SA) { Internet Security Association and Key Management Protocol (ISAKMP)
Pre-shared keys, IKE (v1, v2), Kerberos, Internet Negotiation of Keys (KINK), IPsec DNS records

Protocoles

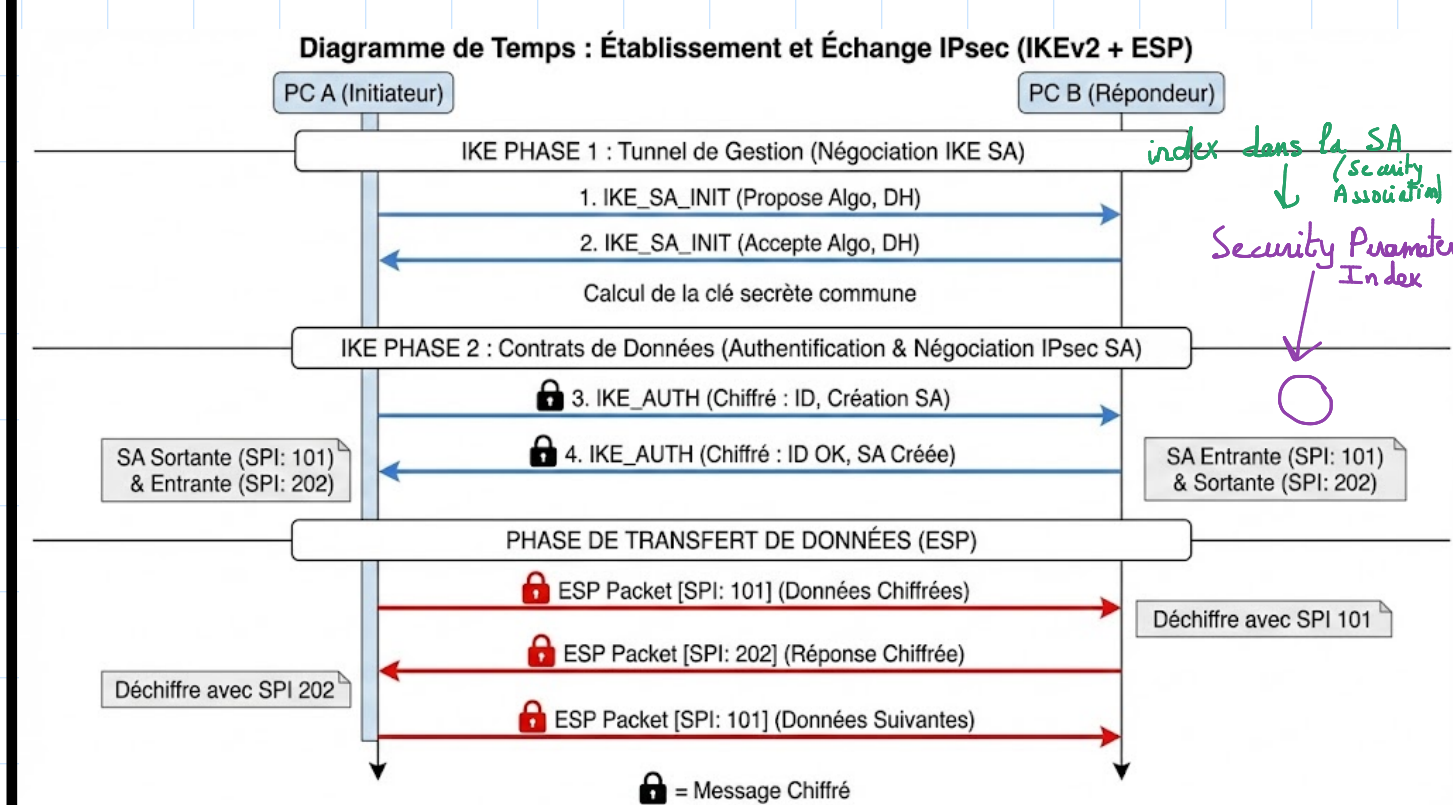
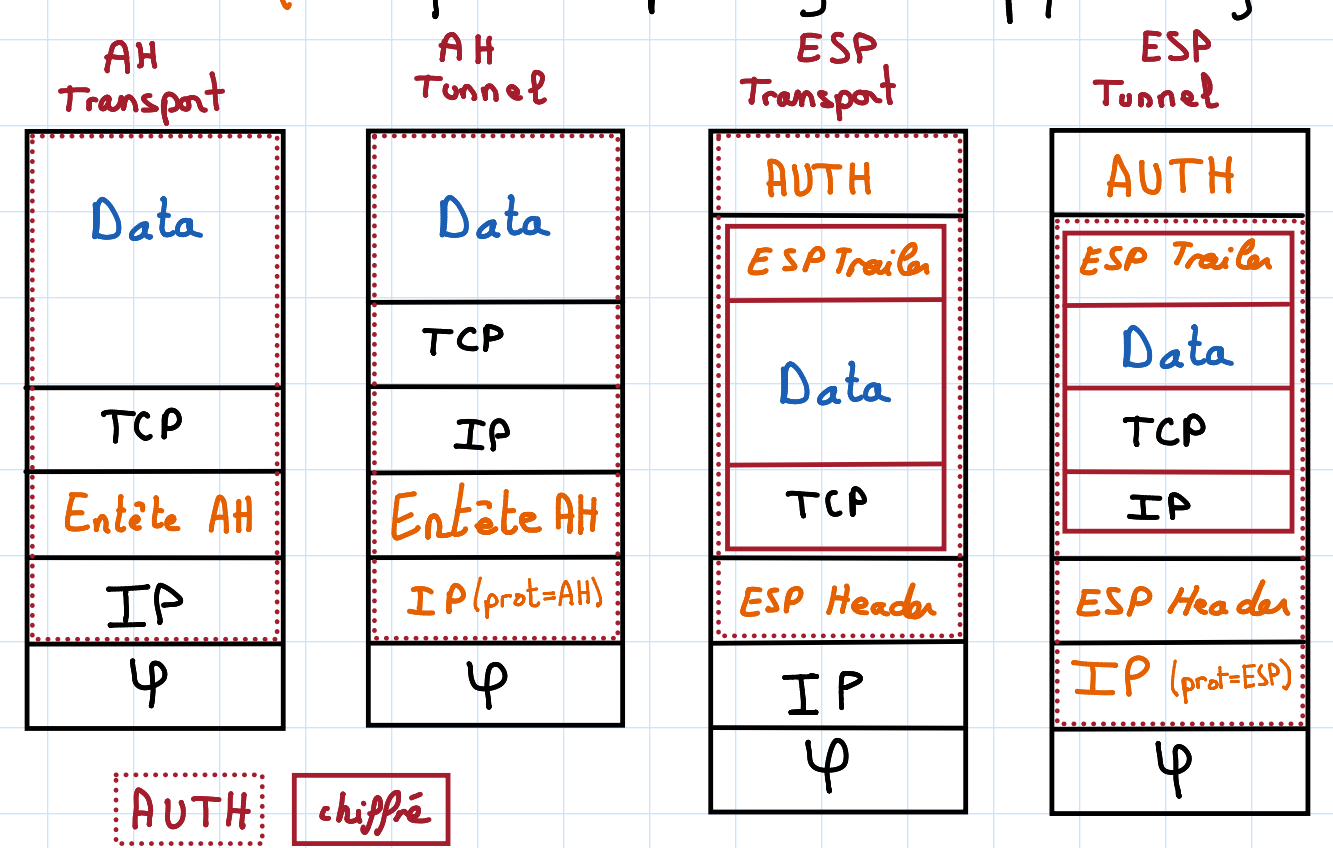
Authentication Headers (AH) { en-têtes aux paquets IP d'authentification
↳ auth, intégrité

Encapsulated Security Payload (ESP) { chiffrement d'un contenu (payload)
↳ auth, intégrité, confidentialité
en-tête donnant authentification, intégrité
protocole contre le rejeu (limitée)

2 mode

Transport { ajout en-tête (AH ou ESP) entre l'en-tête et la payload IP
chiffrement éventuel du payload IP initial.
permet de ne rajouter que quelques octets

Tunnel { encapsulation du paquet IP original (ajout en-tête AH ou ESP autour)
chiffrement, authentification, contrôle d'intégrité, protection anti-rejeu éventuel pour l'intégralité du paquet IP original



AH :

ESP :



SA : IP (src, dst), AH/ESP, cipher suite, spi, secret-key
SP : ports, protocoles, direction
SA est négocié
↳ pour une connexion unidirectionnelle
↳ 2 pour bidirectionnelle

hub : broadcast tout

≠ switch (commutateur) : passe en mode hub si ne connaît pas l'@MAC dest ou qu'il ne fonctionne plus correctement

→ limite dans l'apprentissage d'un port du switch pour ne pas passer en mode hub

→ important d'utiliser des switches, la fonction port security, empêcher l'accès physique aux switches

→ Virtual Local Area Network

VLAN : isolation au niveau 2

→ Private VLAN

PVLANs : en gros tu peux parler qu'au routeur → typiquement réseau publique

équivalent WIFI : Client Isolation

En principe : segmentation par VLAN et interconnection de VLANs par des routeurs (niv. 3)

→ filtre par : protocoles (TCP, ICMP...), ports, IPs, sens de la connexion

stateless : examine chaque paquet indépendamment (ex : peut bloquer les SYN)

Firewalls type routeur

stateful

- garde une trace dans le passé (historique)
- gestion de "transmission" UDP (indisp. pour le DNS)
- gère ICMP, fragmentation, port knocking

applicatif : vérifie le protocole (évite les contournements), typiquement pour les protocoles dont les ports n'est pas facilement prévisibles

→ Access Control List

Firewalls personnel : généralement stateful. ACL selon les applications

problèmes Firewall :

- contrôle IP ⇒ lien implicite machines-utilisateurs
- tunnels & encapsulation : SSH, HTTP connect...
- pas de défense : les paquets sont routés → simplifie les attaques si une faille est trouvée

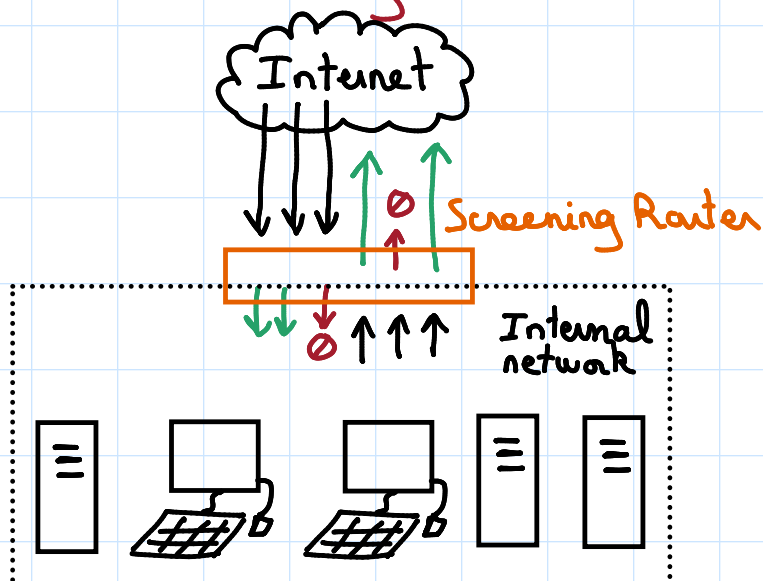
authentification utilisateurs { auth forte contrairement à IP
Pogs + politique en fonction de l'utilisateur

relais applicatifs sécurité par protocole { scan viral pour SMTP
(proxy) Pas de CONNECT on HTTP

paquets traités : seules les infos contenues passent

danger { ① le programme analyse, ② il est complexe ⇒ ③ attaquable
→ il faut réduire les services au minimum
→ audit de bastions/relais
→ prendre en compte la compromission possible

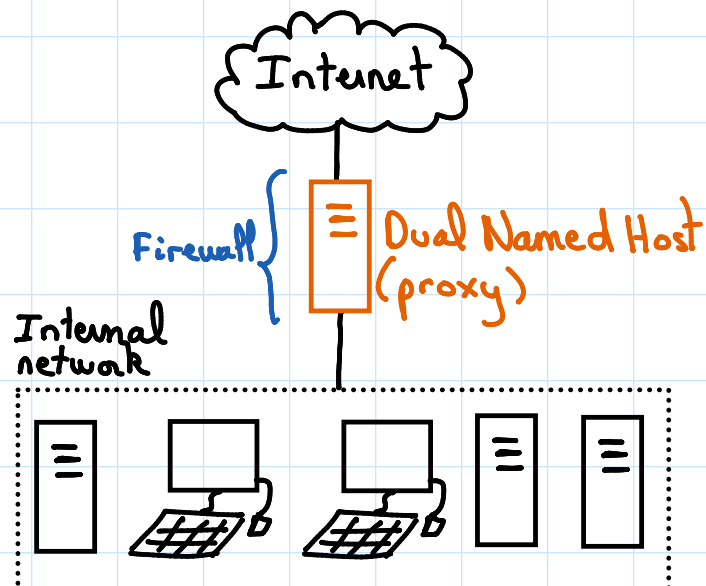
Archi 1 : Screening Router



on détermine une politique arbitraire et on filtre

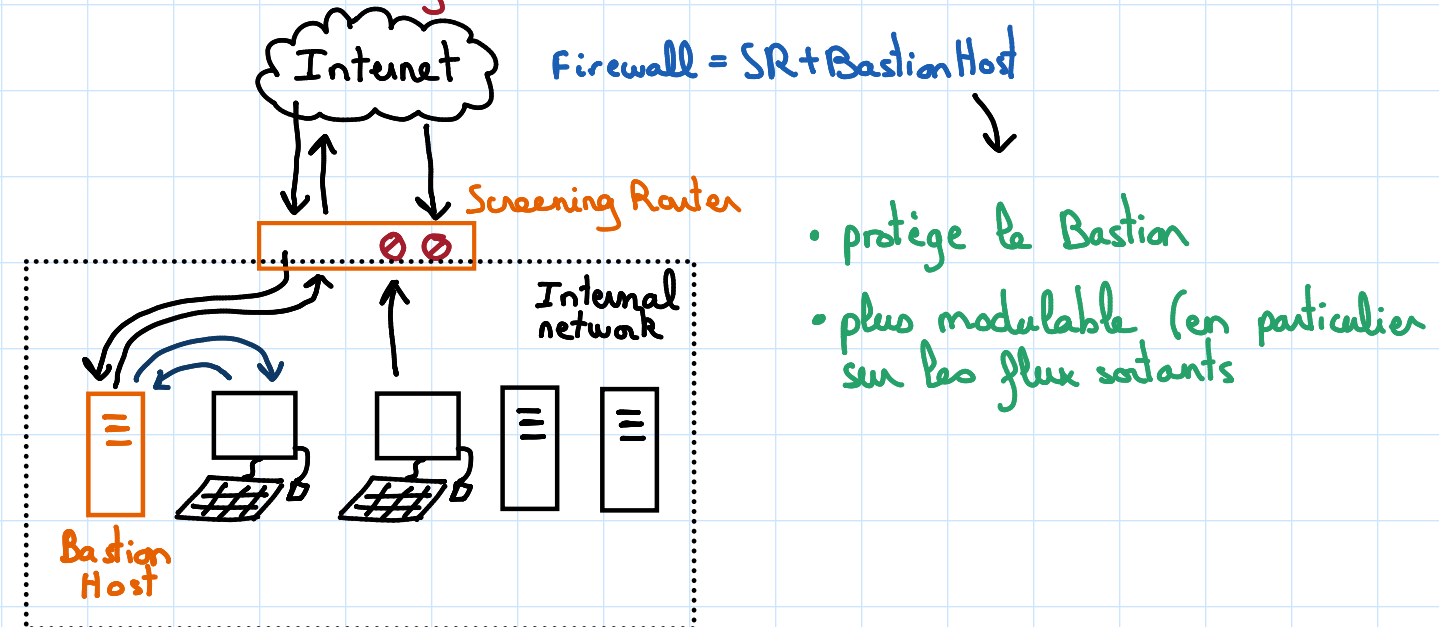
→ le routeur route, il laisse passer ou non mais ne modifie rien

Archi 2 : Bastion / Relai



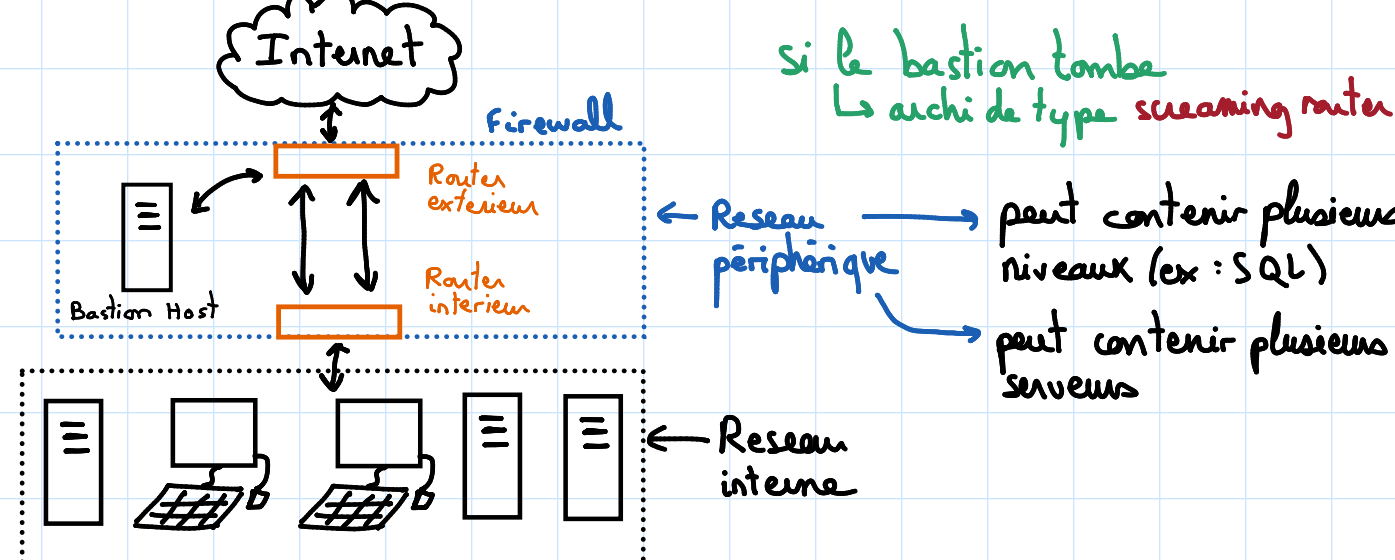
Mise en place de proxy entrants/sortant pour chaque service

Archi 3 : Screening Router + bastion



- protège la Bastion
- plus modulable (en particulier sur les flux sortants)

Archi 4 : DMZ



si la bastion tombe
→ archi de type screening router

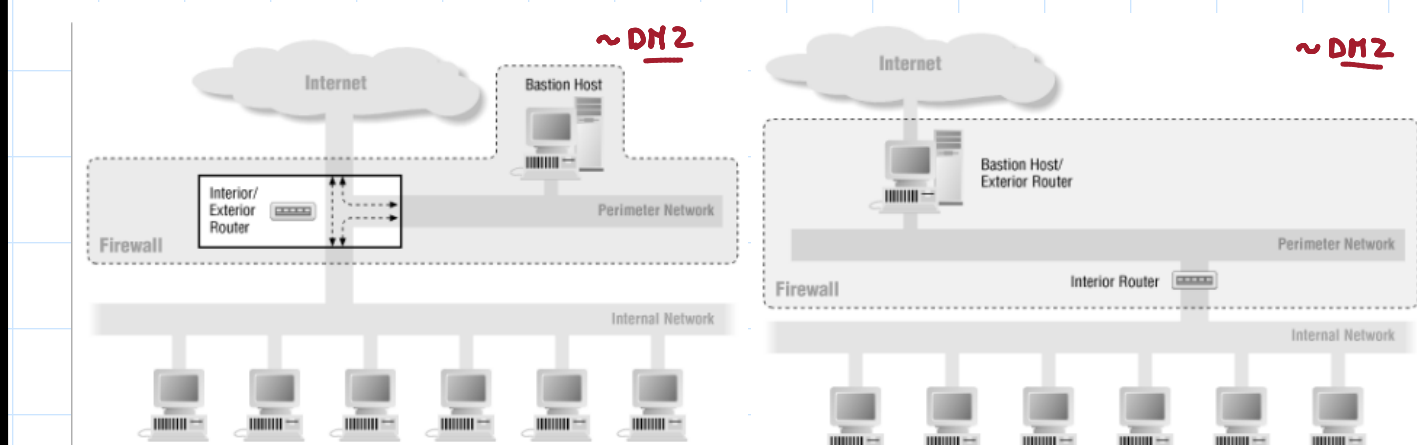
Réseau périphérique

peut contenir plusieurs niveaux (ex : SQL)

peut contenir plusieurs serveurs

Réseau interne

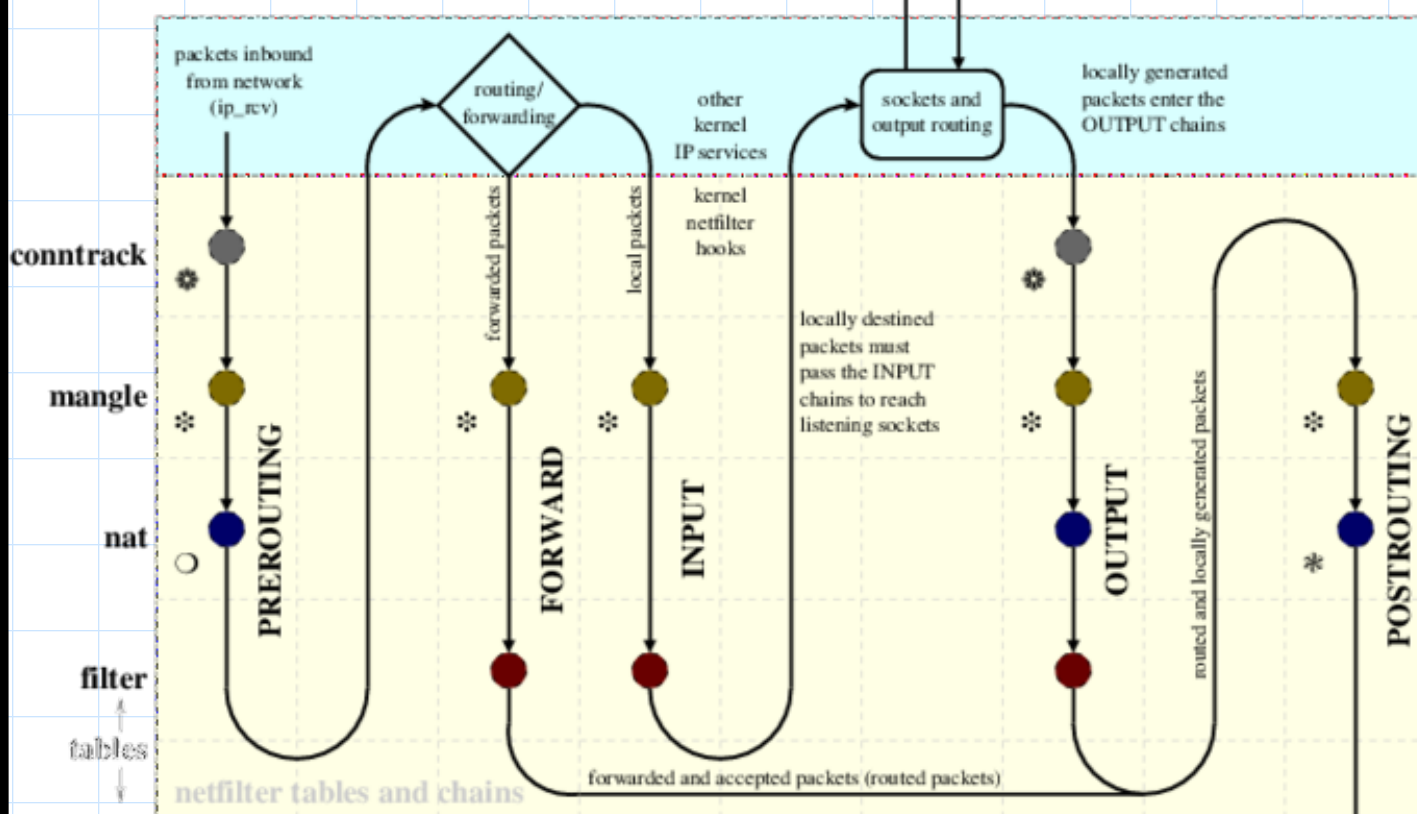
Archi α : variantes



Netfilter Packet Traversal

<http://linux-ip.net/nf/nf-k-traversal.png>

Martin A. Brown, martin@linux-ip.net



Commandes de base	Description
-t table	Si la table n'est pas spécifiée, alors la table par défaut est filter.
-j cible	Exécute l'action définie par cible. ACCEPT/DROP/REJECT
-A chaîne	Append. Ajoute la règle à la fin de la chaîne indiquée.
-I chaîne index	Insert. Ajoute la règle à la place indiquée par index. Les indices commencent à un.
-D chaîne index	Delete. Élimine une règle de la chaîne indiquée.
-F [chaîne]	Flush. Efface toutes les règles de la chaîne et table sélectionnée. Si chaîne n'est pas précisé ça efface toutes les chaînes de la table.
-L [-v]	List. Liste les règles de toutes les chaînes de la table demandée (avec plus de détail si -v est précisé).
-s adresse-ip/masque]	Adresse source du paquet. Peut être une plage si masque est précisé.
-d adresse-ip/masque]	Adresse destination du paquet. Peut être une plage si masque est précisé.
-i interface	Interface d'entrée du paquet.
-o interface	Interface de sortie du paquet.
-P (policy)	-m state --state NEW,ESTABLISHED,RELATED
Protocoles	Description
-p protocole	Protocole du paquet : icmp, udp, tcp, etc.
-p tcp (ou udp) --sport ports	Ports source. Valeur unique ou plage au format portde-part : portdefin.
-p tcp (ou udp) --dport ports	Ports destination. Valeur unique ou plage au format portde-part : portdefin.
-p tcp --syn	Flag SYN. Comme pour beaucoup d'autres options on peut l'inverser. "! --syn" c'est tout paquet avec le flag SYN à zéro.
--icmp-type type	Type ICMP. Généralement echo-request ou echo-reply.