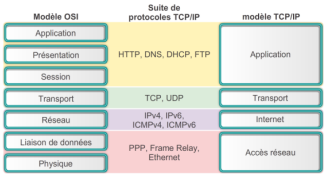
1 OSI



1.1 Attaque lien physique / OSI 1-2

- DOS : brouilleur, émission continue (ethernet, wifi, téléphone)
- Écoute : câble - dispositif dédié, aérien - communication aérienne, aérien - rayonnement parasite (par exemple écouter les ondes qui se propagent dans un radiateur)
- Sniffers logiciel : mode promiscious (on ne jette pas les paquets qui matchent pas notre @MAC) - *ifconfig eth0 promisc*. Contre-mesures : chiffrement, segmentation rzo. Détection : on send des pings à tout le monde - *l'attaquant va écouter l'ensemble des pings donc se prendre une surcharge rzo*, requêtes ARP marl-formé - *remplis la table ARP de l'attaquant*, requête DNSLOOKUP - *il va répondre alors que c'est pas pour lui.*

1.2 Attaque liaison / OSI 1-2

- MAC spoofing : pour contournement des moyens d'écoute ou écouter(en spoofant l'adresse de la personne à écouter) ou spoofer l'adresse de quelqu'un qui est authentifié au portail captif par ex - *ifconfig eth0 hw ether 01 : :ff.*
- MAC flooding : on créer des packets avec @MAC différentes(MAC spoofing pour MAC flooding) pour remplir la table de correspondance port/MAC du switch - le switch va passer en mode "hub". Contre-mesures actuel : plus de mémoire, timeout des entrées.

1.3 ARP spoofing - poisoning / OSI 2-3

écouter - contourner authentification de session - MITM - DOS

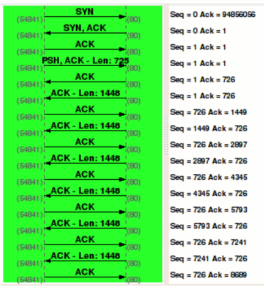
2 solutions :
- IS-AT : ne fonctionne pas tout le temps
- WHO-AS : WHO-AS en unicast (plus discret) à la personne qu'on souhaite empoisoner avec notre MAC source et comme IP source l'IP pour laquelle on essaye de se faire passer. Puis on send des IS-AT similaire pour maintenir la table empoisonné.

Si on spoof 2 caches : écoute ou MITM en activant le routage sur sa machine. MITM + désactivation du routage des packets permet un déni de service.

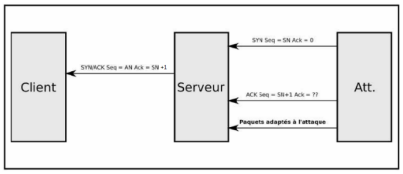
1.4 Attaque réseau / OSI 3-4

1. Fragmentation IP : On peut réduire la MTU avec ICMP et les packets IP seront fragmenté en plein de packets avec charge utile faible donc gros overheading, ce qui va flooder.
2. superposition IP : Idée, au lieu que les fragments se succèdent (début = fin de l'autre), je vais envoyer un fragment qui se superpose avec un fragment précédent pour le ré-écrire → implémentation de la gestion de ce scénario diffère en fonction du manufacturer de l'équipement (pas de standard, pas de RFC, rien). Si le mécanisme de reconstruction de l'IDS (système de détection d'intrustion) est différent de celui de l'appareil ciblé, il est possible que l'IDS reconstruise en ignorant / écrasant la payload malveillante, tandis que l'appareil cible va utiliser un algorithme de reconstruction différent qui lui va préserver la payload
3. IP spoofing : DOS, MITM -> tout en gros
4. DHCP starvation : utiliser l'ensemble de la pool IP.
5. DOS : ping of death/champ TCP résérvé, SYN flood(pleins de SYN avec @IP_{source} différentes pour flooder la mémoire avec des connections semi-ouvertes)/maintient de connection TCP(SlowLoris - on ouvre des connections HTTP par exemple)
6. rDOS (réplication DOS) : décupler avec un serveur, adresse de broadcast (smurfing), dans les 2 cas faut faire du IP spoofing.
7. DDOS : utilisation de botnets.
8. Protocole flood : DNS (jusqu'a x100), NTP (x780)

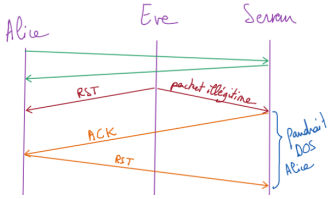
1.5 Attaques TCP / OSI 4



TCP Connection spoofing = TCP Hijacking :



probleme : il faut connaitre l'AN(Ack number) et le SN (Sequence number) et le client va envoyer un RST.

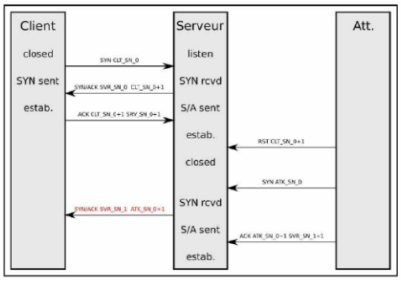


RST Hijacking : envoyer un RST bien formé en se faisant passer pour le serveur. Parler au serveur en se faisant passer pour le client. Probleme : les ACK vers le client vont entrainer un RST du client.

A l'époque les numéro de séquences étaient prévisible.

Attaque de Mitnick : si un serveur A fait confiance(noeud de confiance : en gros pas de handshake) a un serveur B, on DOS le serveur B pour pas avoir de "RST" de sa part et on se fait passer pour lui en reprenant le numéro de séquence correcte qu'on a écouté (qui est incrémenté).

Attaque de Joncheray :



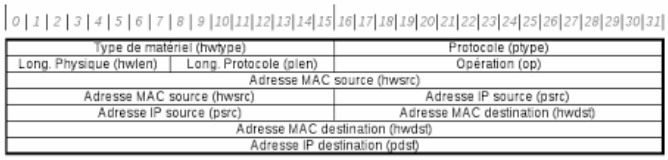
Si on est en MITM on est les rois pour l'attaque de Joncheray il y a tout qui marche on a même pas besoin de désynchroniser les interlocuteurs, on modifie les paquets à la volée, on a juste à faire la traduction des numéros de séquence pour que personne ne s'aperçoive des modifications de payload. Obligé d'être en local pour pouvoir écouter et deviner les numéros de séquence, mais l'avantage c'est qu'on pourra avoir une connexion durable, pas obligé de tout faire en one-shot.

Ack storm : Si les interlocuteurs sont désynchro, chaque packet de A est hors fenetre TCP de B et idem de B vers A. Lors de la réception d'un packet hors fenetre. celui qui reçoit va envoyer un ACK avec sa fenêtre. Ce ACK sera hors fenêtre donc B enverra sa fenêtre etc... ACK pas renvoyé en cas de DROP, processus fini.

1.6 Scan

1. Port : en TCP j'envoie des SYN je prends des reset si c'est fermé, en UDP si je reçois un truc wtf
2. Réseau : je veux savoir qui est sur le réseau
3. Machine : connaître les services et versions des trucs qui tournent sur la machine, e.g. samba

Entete ARP :

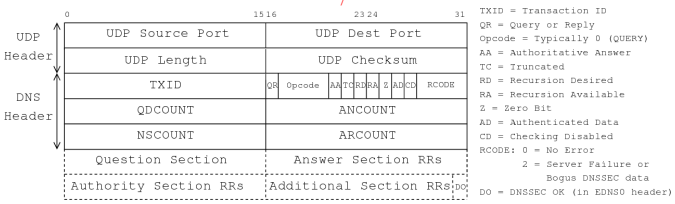


Entete TCP :

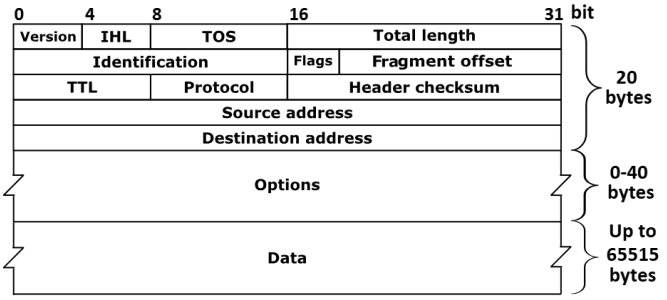
En-tête TCP

Bits	0-15			16-31
0	Port Source			Port Destination
32	Numéro de séquence			
64	Numéro d'accusé de réception			
96	Offset	Réservé	Flags	Taille de fenêtre
128	Total de contrôle			Pointeur d'urgence
160	Options			

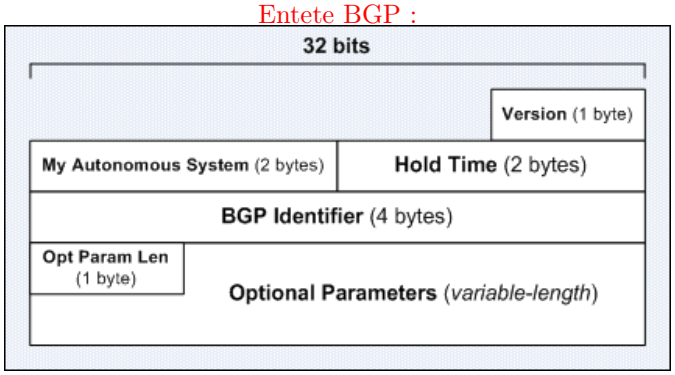
Entete DNS/DNSSEC :



Entete IP :



Entete BGP :



a remplir

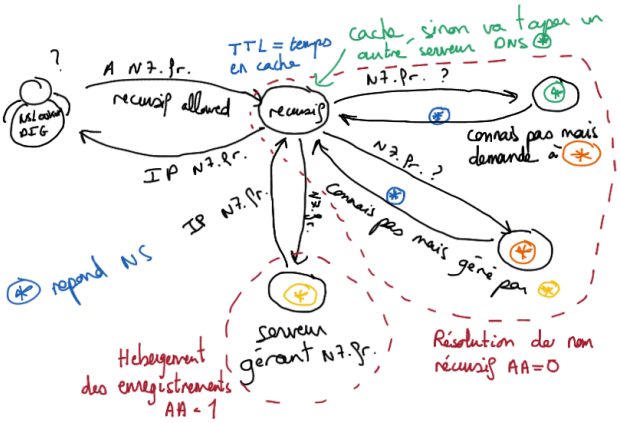
2 DNS (UDP 53)

2.1 Administration

1. Domaines de 1er niveau : *Internet Assigned Numbers Authority* (IANA)
2. Domaines : *Network Information Center* (NIC)
3. Bureaux d'enregistrement désignés par les NIC

2.2 Types de serveurs

1. Hébergement (bind9, NSD)
2. Résolution de noms (bind9, unbound, knot)



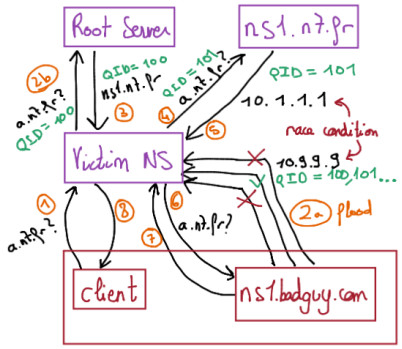
cache négatif : réponses "non existant" en cache

2.3 Champs

1. A : adresse IPv4
2. AAAA : adresse IPv6
3. NS : délégation de zone (Name Server)
4. MX : adresse échangeur de mail
5. SOA : Start of Authority
6. NSEC : Next SECure

Nom	TTL	Classe	Type	Données
n7.fr	84600	IN	SOA	n7.fr | master@n7.fr
a.n7.fr	84600	IN	NS	ns.a.n7.fr délégation sous-zone
ns.a.n7.fr	84600	IN	A	193.48.20.3 glue record
n7.fr	84600	IN	MX	smtp.n7.fr

2.4 Empoisonnement simple : race condition



Il faut le bon port UDP et query ID. Gros TTL est un +
Si on rate la race condition alors le vrai serveur à répondu et sera en cache donc on doit attendre pour réhitérer l'attaque.

2.5 Empoisonnement de Kaminsky

On va empoisonner la délégation de zone (glue record) en gagnant la race condition. Du coup si on rate l'attaque on peut recommencer (pas obligé de réussir du premier coup)

Empoisonnement à la Kaminsky (2008) (1/3)

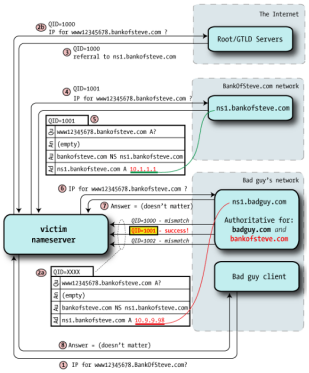
Objectif plus ambitieux et ... plus simple !

Pervertir dans le cache quel est le NS d'un domaine (e.g. `bankofsteve.com`)
Pourquoi plus simple ? On peut tenter l'attaque autant de fois qu'on veut

Attaque

1. Faire ou faire faire une requête vers `[aléa].bankofsteve.com`
2. Envoyer une réponse avec un glue donnant une fausse adresse pour `ns1.bankofsteve.com`
3. Si ça ne marche pas recommencer

Même si le DNS cible est en cache il suffit que `[aléa].bankofsteve.com` ne le soit pas !



on peut pas bruteforce le Qid car 256 donc timeout en 2secondes

Bug du serveur récursif de bind9 : plusieurs requêtes pour un même nom de domaine !!!

Le serveur récursif et l'adversaire tirent au hasard et indépemment un couple de *query id* parmi $m = 2^{16}$ valeurs

$$P(m,n) \approx 1 - e^{-\left(\frac{n^2}{m}\right)}$$

```
>>> import math
>>> m = 2**16 # 65536
>>> n = math.sqrt(m) # 2^(16/2) = 256
>>> 1 - math.exp(-1*(n**2)/m)
0.63
```

2.6 DNSSEC

1. RRSIG : Signature d'un ensemble de registre (par type/nom)
2. DNSKEY : Clé publique pour check les signatures
3. DS : Haché d'une DNSKEY d'un DNS d'un sous-domaine
4. NSEC, NSEC3, NSEC3PARAM : hors-programme

(.) Racine
- DNSKEY(. - KSK) clé publique connue par le résolveur
- DNSKEY(. - ZSK)
- RRSIG(DNSKEY) signature DNSKEY racine (signé par KSK)
- DS(fr.) = hash(DNSKEY(fr. - KSK))
- RRSIG(DS) signé par la racine avec sa ZSK

vérif signature avec DNSKEY(ROOT), compare DS(fr.) avec hash(DNSKEY(fr.))

(fr.) TLD : Top Level Domain
- DNSKEY(fr. - KSK)
- DNSKEY(fr. - ZSK)
- RRSIG(DNSKEY) signature DNSKEY fr. (signé par KSK)
- DS(n7.fr.) = hash(DNSKEY(n7.fr. - KSK))
- RRSIG(DS) signé par .fr avec sa ZSK

vérif signature avec DNSKEY(fr.), compare DS(n7.fr.) avec hash(DNSKEY(n7.fr.))

(n7.fr.) Zone enfant
- DNSKEY(n7.fr. - KSK)
- DNSKEY(n7.fr. - ZSK)
- www.n7.fr. A = 93.184.216.34
- RRSIG(A) signé par n7.fr. (signé par ZSK)

vérif RRSIG(A) avec DNSKEY(n7.fr.)

ZSK : Zone Signing Key (généré par KSK) - KSK : Key Signing Key - chain of trust

Comment prouver qu'un champ n'éxiste pas ?

Nom	Type	Données
n7.fr	SOA	n7.fr | master@n7.fr
a.n7.fr	A	193.48.20.1
a.n7.fr	MX	a.smtp.n7.fr
b.n7.fr	A	193.48.20.2
c.n7.fr	A	193.48.20.3
a.n7.fr	NSEC	b.n7.fr. A MX RRSIG NSEC (chaînage)
b.n7.fr	NSEC	c.n7.fr. A RRSIG NSEC (chaînage)
c.n7.fr	NSEC	n7.fr. A RRSIG NSEC (bouclage, fin de zone)

- DNSSEC : intégrité seulement
- raisonable d'utiliser un solveur récursif public : non car on doit alors faire confiance au solveur et ne pas vérifier la chaine de confiance.

3 BGP : Border Gateway Protocol

BGP : Abstraction de distance pour calculer des routes.
C'est sur du TCP donc ça souffre de tous les mêmes problèmes qu'on a vu (MITM, vol de session, etc.), mais encore une fois on va voir les problèmes spécifiques à BGP.
IGP → interne, souci de performance dans l'AS
EGP → externe, principalement des soucis contractuels (interconnexion et compatibilité)
AS (Système Autonome) : par exemple des FAIs ex : Renater (éducation FR), ou pour transférer : IXP
IXP (internet exchange point) → des entreprises qui sont des AS spécialisées dans l'interconnexion d'AS (e.g. "donne-nous 20 millions de dollars et je t'interconnecte avec l'AS de l'autre côté de l'océan pendant un an")
culture générale : Bellman-Ford $O(S^3)$, Dijkstra $O(Slog(S))$ –> on utilise OSPF
Fonctionnement en Path : Plus un chemin est court et plus l'aggrégation est faible (on annonces un /xx avec xx plus grand) plus on voudra passer par là
Si on annonce du BGP on donne son ID donné par l'IANA (AS Number)
Incidents :

1. suicide de AS7007 : reçoit 23k routes, remplace la path par lui et les désagrège –> beaucoup de routes très intéressantes car précises
2. suicide du Pakistan : Volonté de censurer youtube : annonce d'un niveau plus précis qui prétend router vers YouTube.

Attaque BGP : Il suffit de mentir et prétendre être meilleur => annoncer une meilleure route. Tous les chemins vont passer par nous
BGP-spoofing : facile de balancer n'importe quel numéro de BGP car 0 authentification
DOS BGP : Annoncer pour une zone particulière qu'on est X et dumper tout quand les gens nous utilisent –> d'où le fait d'avoir des looking glass
Solution Pour le spoofing :

1. Authentifier les routeurs ? ==> NOP, ça marche pas Authentifier via les DNS ? ==> NOP Écouter un peu partout (looking glass) et regarder si personne ne balance notre ID ==> mouais pas une vrai solution Solution RFC RPKI : Certificat d'entité finale (EE) pour pouvoir prouver qu'on annonce bien : on crée un Route Origination Authorization (package avec un certif EE, l'IP, son slash). Empêche de rendre les addresses plus précises pour protéger des hijacking. Beaucoup de choses sont réglées niveau transport (avec TLS les communications reroutés par un malveillant ne sont pas lisibles, on aura beau falsifier le site web aura un certif)

à remplir

4 DOS

4.1 Types of DOS

— Distributed Denial Of Service : botnet...
— Denial Of Service sur failles applicatives : ping of death...
— Replicated Denial Of Service : DNS, FTP...

4.2 Attaques

— Attaques locales : diffusion au max de la bande passante (broadcast : ARP par ex), saturation des câbles. Contre Mesure : par dichotomie on découpe le réseaux jusqu'à isoler la source. Prévention : limiter traffic broadcast
— Attaques distantes : plus facile à filtrer si source unique (recherchent souvent phénomène d'amplification), broadcast impossible Cas classique : Botnet
— Attaques DHCP : famine - récupérer toutes les adresses IP. Limitation : Ne bloque pas les machines déjà connectées
— Attaques ICMP : ping flooding Smurf version : ping avec comme IP source celle de la victime en broadcast (victime reçoit ECHO REPLY). Contre Mesure : Filtrage ICMP - pas de reply. Similaire : "Fraggle" en UDP
— Attaques IP : *ping of death* (packet mal formé : un ping de 65535 octets pour créer des fragments IP mais après reconstruction le packet est trop grand ce qui induisait buffer overflow et crash), *Teadrop* (variante avec fragments TCP), *Famine* (paquets fragmentés induisent saturation firewalls, switches -> forte amplification de saturation mémoire).
— TCP RST : Interdire à une machine d'accèder à un service Comment : deviner le numéro de séquence et envoyer un TCP RST (une chance sur 2^{32}). Limites : randomisation du port source et numéro de séquence à l'ouverture de la connexion
— TCP SYN FLOOD : Saturer la file pour les connexions semi ouvertes (elle faisait 10-128 à l'époque, aujourd'hui 1024 ou plus). Contre mesure : augmenter la taille de la file, réduction timeout, filtrage IP (contourné par randomisation source). Syn cookie ! !

Syn Cookies

1. Clé secrète
2. T : compteur sur 5 bits incrémenté toutes les 64 s
3. L : $MAC_{\text{clé}}(S_{addr}, S_{port}, D_{addr}, D_{port}, SN_c, T)$ sur 24 bits
4. SN_s : concaténation de T, MSS, M

Quand le serveur reçoit le SYN/ACK, il extrait MSS, vérifie L et T et alloue l'espace si c'est bon.

— DNS amplification : envoyer requêtes à résolveurs récursifs en remplaçant la source par victime. requête de 60octets $\implies$ 512 octets ($x8$). Jusqu'a $x60$ avec windows 8. Contre mesure : DNSSEC.
— Attaques génériques : coup déséquilibrés sur le calcul : *client chiffre, serveur déchiffre* (x10).
— Attaques génériques : coup déséquilibrés sur la bande passante : *demander un PDF de 40Go*
— Attaques topologie de réseau : Attaques par "éclipse" - prendre le contrôle de noeud afin d'interdire aux membres de différents sous réseaux de communiquer entre eux.
— Attaques BGP : Saturer routeurs - jusqu'a saturer un AS. Souvent par erreur de configuration (Pakistan Telecom et trafic Youtube) Comment : flooding BGP - flux de paquets avec caractéristiques temporelles précises. Contre-mesures : S-BGP, BGPSEC (de la crypto encore).

5 Math - lol

5.1 Lois marginales

On calcule la corrélation avec la covariance On peut pas utiliser Shanon car c'est trop le bordel, on va faire une décomposition matricielle du coup.

5.2 Loi gamma

- Grosse fonction qui permet de faire autant des gaussiennes, des exponentielles etc - $\Gamma_{a,b}(x) = \frac{1}{\beta\Gamma(\alpha)}(\frac{x}{\beta}) \times exp(-\frac{x}{\beta})$ - β : amplitude - $\frac{1}{\alpha}$: distance à la Gaussienne

5.3 Farima

modèle autorégressif (en fonction de ce que j'ai je calcule une variance qui se réduit et me permet de prédire de plus en plus) permettant de considérer une moyenne mouvante (apprentissage autour d'une valeur qui change) $f_{X_\Delta}(v) = \sigma_\epsilon^2 |1 - e^{-i2\pi v}|^{-2d} \frac{|1-\theta e^{-i2\pi v}|}{|1-\phi e^{-i2\pi v}|}$
DDos sur la marginale : askip la forme du traffic change mais pas l'amplitude
Pendant l'attaque DDOS c'est sur le paramètre α qu'on voit la différence
Courbe Receiver Oriented Curve : Courbe qui voit si le système détecte bien ce qui doit être détecté. Le but est d'avoir une courbe qui monte directement en (1,0)
Au finale solution complexe qui dit qu'on a un DDoS mais ON NE SAIT RIEN. On fait beaucoup d'efforts pour finalement pas grand chose (un peu comme ce cours non ?)

6 Questions annales

Quels seraient les principes d'une attaque d'UDP Flooding pour maximiser son efficacité et chances de succès ?
Utiliser des paquets avec des ports aléatoires pour éviter le filtrage. Utiliser des IP sources spoofer pour éviter la détection. Cibler des services qui répondent aux paquets UDP pour amplifier le trafic (par exemple DNS ou NTP). Envoyer un grand volume de paquet en continu pour saturer la BP ou les ressources du serveur
Nous disposons d'un firewall pour sécuriser un réseau d'entreprise des flux malveillants venant de l'Internet mondial. Quels seraient les dimensions/attributs du trafic à considérer pour se prémunir d'une attaque de type UDP flooding ?
Il va falloir surveiller le volume (débit) du trafic (taux de paquets par seconde, ou débit en bit par seconde). Une augmentation soudaine et massive du trafic UDP peut indiquer une attaque. On va aussi vérifier la taille des paquets, car les attaques UDP flooding utilisent souvent des petits paquets pour maximiser le nombre de paquets envoyés par seconde. Une prévalence anormale de paquets de taille fixe ou minimale peut donc être suspecte. On peut aussi surveiller s'il y a une distribution inhabituelle des ports de destination, ou une diversité anormale des adresses IP sources (ou des adresses IP non rentables). De plus, dans une attaque, les paquets UDP sont souvent envoyés sans attendre de réponse, donc on peut surveiller s'il y a un taux élevé de requêtes sans réponse. Pour se protéger, on peut limiter le débit par IP ou par port, filtrer les paquets UDP, et bloquer des IP sources suspectes. De manière générale, il faut faire de la détection d'anomalies, et une inspection approfondie des paquets pour identifier des signatures d'attaque.
Truc de seuil de l'outil de détection
On additionne les moyennes et les écarts maximaux des flots.

1. Flot 1 (poisson) : $moyenne = variance = \lambda_1$
2. Flot 2 : constante = c, (donc moyenne = C et variance = 0)
3. Flot 3 (expo) : moyenne = $\frac{1}{\lambda_2}$,$variance = (\frac{1}{\lambda_2})^2$
4. Flot 4 : Moyenne = a, variance = b

Le seuil minimal est la somme des moyennes + l'écart maximal possible. Soit Seuil = Somme des moyennes + 3*écart type (soit sqrt(somme des variances))
Lorsqu'un système de détection lève une alerte alors qu'il n'y a pas lieu, il s'agit d'un faux positif (une action a été réalisée). Lorsque, au contraire, il ne lève pas d'alerte alors qu'il aurait dû, il s'agit d'un faux négatif