



Surveillance et gestion d'incidents

2

Architecture

Capture et analyse réseau

Commissariat à l'Énergie Atomique et
aux Énergies Alternatives

Février 2020

L'architecture réseau joue un rôle primordial dans la détection d'intrusion réseau.

Elle a un double rôle :

- Faciliter la mise en place de filtrage réseau et de protections (périmétrique et interne);
- Faciliter la mise en place de la surveillance.

Premier pilier de la sécurité d'un système d'information

Une architecture réseau bien pensée à la base doit fournir un socle solide sur lequel mettre en place des protections et de la surveillance.

Intérêts de prendre en compte la sécurité et la surveillance dès la mise en place d'une nouvelle architecture :

- Evolution difficile après le déploiement initial;
- Impact moins lourd sur l'ensemble du SI en cas d'évolution;
- Modularité facilitant l'évolution du SI sans négliger la prise en compte de la surveillance;
- Evolution de la surveillance avec l'évolution du SI.

Une architecture bien pensée doit permettre également de faire de la défense en profondeur facilement.

Machines et services accessibles = surface d'exposition

Plus cette surface est petite, plus les options d'un attaquant sont limitées.

Il faut donc :

- Limiter l'exposition des serveurs et services;
- Rajouter des barrières successives à un attaquant dans le but de :
 - le ralentir;
 - faciliter sa détection.

Bulles réseau et DMZ

Comment segmenter son réseau :

Regrouper les services qui ont un rôle similaire, un même niveau d'exposition face à un attaquant. Typiquement :

- relais-mail, proxy, cache DNS externe
- serveurs de cœur de réseau : AD, Serveur de stockage, mail
- Serveurs d'hébergement web

- Zone d'admin
- Zone de surveillance

- Zone de stations
- Bulles Windows / Linux ?

Bulles réseau et DMZ

Il peut également être intéressant de créer des bulles pour des services que l'on sait être :

- Particulièrement exposés
- Obsolètes ou vulnérables

Principe : on isole au maximum un service qui est exposé

S'il est compromis, on limite la contagion.

Augmenter la segmentation du système d'information à un coût.

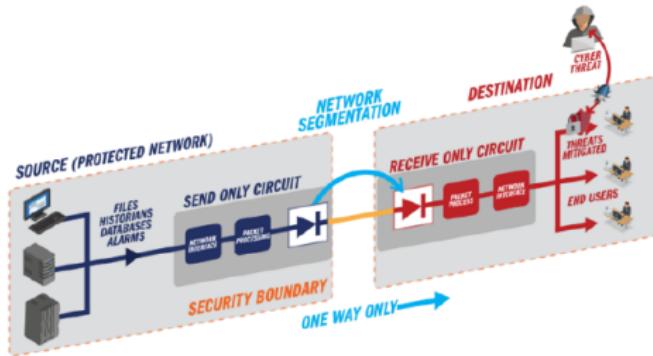
➔ Recherche du meilleur compromis entre administrabilité et sécurité.

La solution des **private VLANs** :

- Évite la multiplication de VLANs;
- Particulièrement adapté pour les regroupements de :
 - stations;
 - imprimantes;
 - interfaces IPMI.

⚠ Vérifier la compatibilité des équipements réseau.

Diodes réseau



- Relier des réseaux de niveaux de sensibilité différents;
- Être complètement furtif/passif (cas des systèmes industriels);
- Permet la remontée d'informations dans un seul sens.

La capture du trafic réseau : définition

C'est l'association de deux actions :

- Écouter ("sniffer") le trafic transitant par une interface réseau
- Enregistrer les données reçues de manière permanente

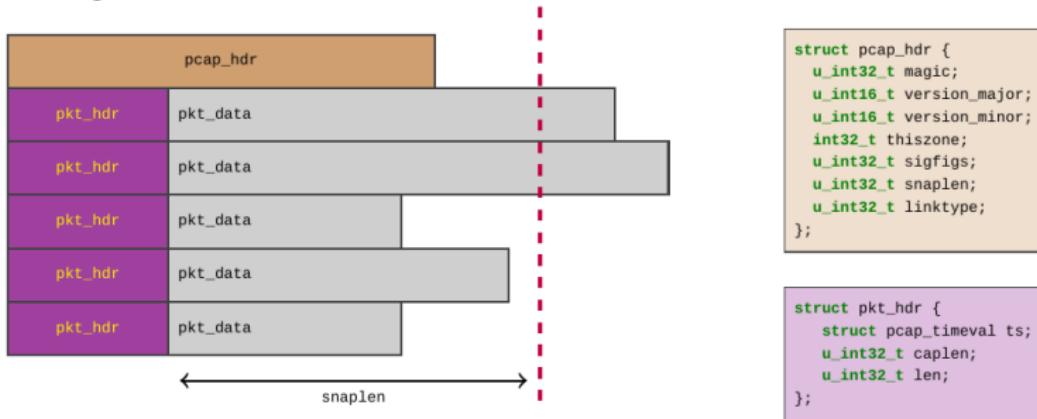
Exemple : pour enregistrer les 500 premiers octets de chaque paquet reçu sur l'interface eth0 dans des fichiers de capture de 20Mo.

```
# tcpdump -n -i eth0 -s 500 -C 20 -w capture.pcap
# ls -alrt
-rw-r--r-- 1 root root 20000035 2009-04-04 23:34 capture.pcap
-rw-r--r-- 1 root root 20000084 2009-04-04 23:35 capture.pcap1
-rw-r--r-- 1 root root 20000278 2009-04-04 23:36 capture.pcap2
-rw-r--r-- 1 root root 7284159 2009-04-04 23:36 capture.pcap3
```

Si l'interface est en mode promiscuous, le driver de la carte réseau fait remonter toutes les trames reçues au niveau 2.

Le format PCAP

Le format PCAP est un format de fichier (popularisé par la *libpcap*) permettant d'enregistrer le trafic réseau.



Remarques

- format simple et universel;
- pas de métainformations sur les données capturées;
- évolution vers le format pcap-ng ?

Les méthodes de capture

Les réseaux actuels étant fortement commutés, différents moyens peuvent être mis en oeuvre pour que la sonde puisse « voir » le trafic réseau.

HUB équipements anciens, en voie de disparition.

Port Mirroring nom générique de la fonctionnalité permettant à un équipement réseau de dupliquer le trafic d'un port donné vers un autre port.

- SPAN ou RITE ;
- uniquement sur les équipements relativement évolués ;
- traitement coûteux pouvant entraîner des problèmes de performances ou de stabilité ;
- pertes de paquets, rejet des trames incorrectes, etc.

Boitiers TAP (« *Test Access Port* »)

Équipement passif conçu spécifiquement pour la capture réseau.

Caractéristiques des boitiers TAP

- Déploiement en coupure du lien réseau que l'on veut capturer;
- Les flux réseaux sont séparés et recopiés au niveau physique :
 - capture les paquets sans la moindre altération;
 - peut nécessiter de re-fusionner les flux (« *channel bonding* »)
- En cas de défaillance du boîtier, le flux réseau original n'est pas altéré (mode dit « *fail-open* »).
- Les boîtier TAP sont similaires à des diodes et ne laissent passer le flux que dans le sens de la duplication.
- Équipement autonome ou mode embarqué (carte PCI).



En plus des caractéristiques basiques :

- 1 Les flux réseaux sont re-fusionnés au niveau du boîtier.
→ plus besoin de « *channel bonding* » au niveau de la sonde.
- 2 Un même boîtier peut fournir plusieurs ports de capture.
- 3 Différentes technologies : cuivre, optique, mixte, etc.



Furtivité des points de capture

Quelle que soit la solution technique retenue (boîtier TAP ou *port mirroring*), le point de capture **ne peut pas** émettre de trames sur le réseau.

- la capture du trafic est complètement invisible ;
- de multiples points de capture peuvent être mutualisés sur un réseau dédié ;

Le problème du très haut-débit (à partir de 10 Gb/s)

- aspects matériels (stockage) ;
- CPU offloading (FPGA, multi-core) ;
- précision de l'horodatage des paquets.

Le choix du positionnement des points de capture est stratégique.

Avec une bonne architecture, les points de capture intéressants à surveiller seront plus facile à identifier :

- en périphérie (capture du trafic depuis/vers le monde extérieur);
- au coeur du réseau (flux métiers/utilisateurs);

Points plus particuliers :

- Serveurs hébergeant des données sensibles;
- Services exposés;
- Accès visiteurs dans les salles de réunion;
- Points d'accès wifi.

Problématiques :

- Espace "tampon" sur la machine de capture à dimensionner correctement;
- Parallélisation possible avec plusieurs points de capture;
- Conserver les données utiles pour l'analyse et jeter le reste;
- Utilité de capturer l'intégralité du trafic chiffré?
 - Méta-données plus intéressantes sur ce type de flux;
 - Conserver quelques centaines d'octets du début de connexion (bannière SSH par exemple).

Un point de capture où il n'y a que du trafic chiffré a moins d'intérêt : aucune donnée utile à analyser, très peu de meta-données.

Il faut surveiller en priorité les endroits où les flux sont en clair.

Stratégies pour les flux chiffrés

- Séparation entre chiffrement et services :
 - Relais SMTP intermédiaire;
 - Reverse proxy HTTP;
- Déchiffrement SSL à la volée :
 - Potentiellement gourmand en ressources;
 - Risque de rompre la chaîne de confiance des certificats.

Archivage des captures :

- Traçabilité des échanges (être capable de remonter dans le temps et de savoir qui communique avec qui);
- Recherche de marqueurs indiquant des signes de compromission;
- Analyse des volumes de données;

Difficulté de conserver des captures complètes :

- Espace de stockage limité;
- Trafic réseau souvent déjà compressé (en-tête Content-Encoding: gzip);
- Généralement on ne peut garder que quelques heures à quelques jours de capture.

Privilégier l'archivage des métadonnées : analyse plus rapide (mais moins exhaustive), possibilité de compression.

- Boîte à outils classique : tcpdump, dumpcap

net2pcap

Outil de capture réseau en *environnement hostile* :

- programme simple, facilement auditible (\rightarrow 400 lignes de C);
- indépendant de la libpcap;
- très performant.

<https://github.com/nbareil/net2pcap>

L'analyse peut se faire de façon asynchrone par rapport à la capture. Cela permet de :

- lisser l'analyse de trafic dans le temps, absorber les pics de trafic
- garantir que tout le trafic sera bien analysé, et qu'aucun paquet ne sera perdu lors de la capture.
- multiplier les analyses sur la capture de trafic et également de les répartir sur une ou plusieurs machines.

L'analyse en temps-différé doit être privilégiée, **y compris dans le cadre des systèmes de type NIDS.**

- séparation des rôles → meilleure fiabilité;
- prise en compte des aspects *forensic*;
- idée d'une détection d'intrusions collaborative.

L'analyse en temps-réel reste néanmoins intéressante pour :

- l'expérimentation (par ex : déployer un NIDS en trois clics);
- lorsqu'on s'intéresse plus à l'aspect quantitatif des événements (par ex : métrologie sur les réseaux haut-débits);
- les statistiques (→ montrer des diagrammes à la hiérarchie!).

Détection par signatures

Rechercher des formes d'événements caractérisant une attaque de manière précise.

- peu de faux-positifs (selon la qualité de la base de signatures);
- caractérisation précise d'une attaque, voire d'un outil;
- la base de signatures doit être à jour (comme pour un antivirus);
- les attaques inconnues ne sont pas détectées, et l'évasion reste toujours possible pour une signature donnée¹.

Détection d'anomalies

Rechercher des événements qui sortent de l'ordinaire (« pourquoi la secrétaire de direction utilise nmap ? »)

- nombreux faux-positifs (à cause du caractère imprévisible des utilisateurs et des réseaux);
- comment définir le seuil de normalité? (→ phase d'apprentissage);
- peut détecter des attaques non connues (mettre en évidence les symptômes sans connaître l'attaque).

Dans sa forme la plus simpliste, cela consiste à rechercher une chaîne de caractères dans un paquet réseau.

Exemple : détection des attaques de type *SQL injection* :

```
bash# ngrep -q -x -d eth0 '(SELECT|UNION)' 'tcp and dst port 80'

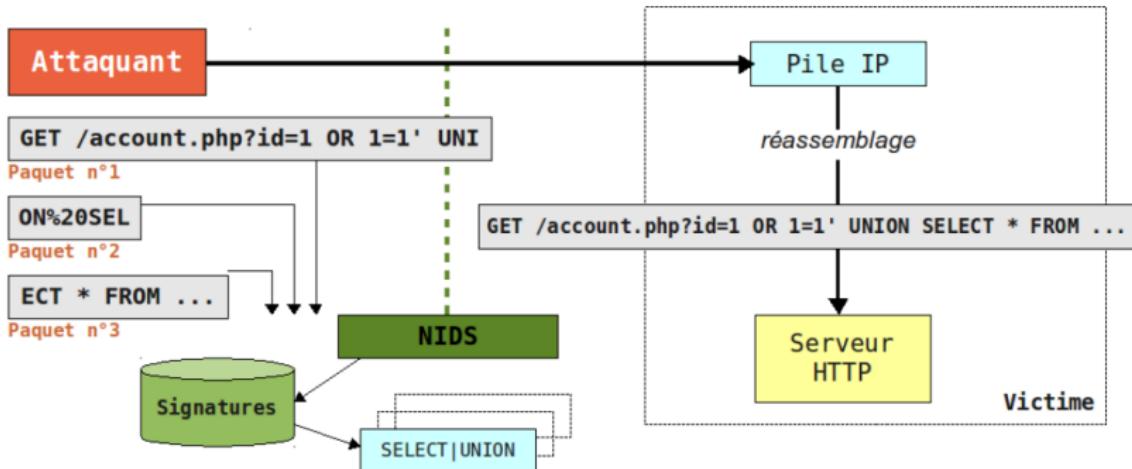
T 10.0.20.53:56422 -> 91.121.40.100:80 [AP]
47 45 54 20 2f 61 63 63      6f 75 6e 74 2e 70 68 70      GET /account.php
3f 69 64 3d 31 25 32 30      4f 52 25 32 30 31 3d 31      ?id=1%20OR%201=1
25 32 37 25 32 30 55 4e      49 4f 4e 25 32 30 53 45      %27%20UNION%20SE
4c 45 43 54 25 32 30 2a      25 32 30 46 52 4f 4d 25      LECT%20*%20FROM%
32 30 61 64 6d 3b 20 48      54 54 50 2f 31 2e 31 0a      20adm; HTTP/1.1.
```

Problèmes

- très peu précis (→ risques importants de faux-positifs);
- le contournement est très simple.

Une technique simple d'évasion

Pour contourner la signature précédente, il suffit simplement de découper le paquet original en plusieurs paquets.



Pour contrer ce type d'évasion, le moteur de détection doit travailler sur des **sessions réseaux complètes**, c'est-à-dire être capable de :

- de réassembler les datagrammes IP fragmentés;
- de fusionner les segments TCP d'une même connexion.

Remarques

- le suivi des sessions est relativement gourmand en ressources;
- problèmes de pollution (bruits, paquets malformés, mauvaise configuration de certains équipements, etc.);
- chaque système a sa propre implémentation de la pile TCP/IP et il est illusoire de penser que le moteur de détection puisse adapter finement son comportement à l'ensemble des machines surveillées¹;

1. « *Eluding IDS : Insertion, Evasion and Denial Of Service* » – T. Ptacek & T. Newsham

Les différences d'interprétation ne se limitent pas aux couches réseaux mais touchent aussi le niveau applicatif¹.

Dans l'exemple précédent, le moteur de détection doit d'abord décoder le protocole HTTP pour chercher l'attaque par *SQL injection* à l'emplacement approprié dans le paquet.

D'autres techniques d'évasion auraient pu être utilisées :

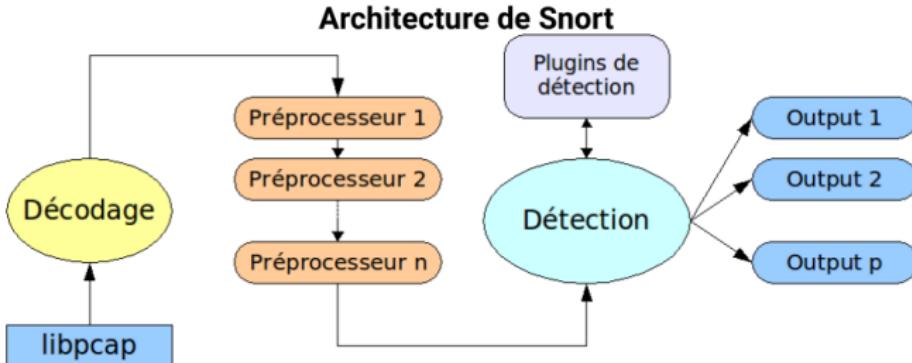
- jeux de caractères différents, encodage, etc.
- différences d'implémentation des protocoles.

1. On rappelle que dans le monde TCP/IP, la couche applicative prend très souvent en charge les deux couches inférieures (**session et présentation**).

- nécessite une quasi-parfaite adéquation entre la sonde et les systèmes surveillés, que ce soit au niveau de la pile TCP/IP ou au niveau applicatif;
- problèmes de pollution (bruits, paquets malformés, mauvaise configuration de certains équipements, etc.);
- le suivi des sessions et le décodage des protocoles demandent des ressources importantes (→ perte de paquets);
- nombre de faux-positifs et pertinence des informations remontées (→ couplage avec un scanner de vulnérabilités);
- précision de la configuration des règles de détection → diminution du nombre de faux positifs;
- qualité de la base de signature (0day ?);
- problème des flux chiffrés (SSH, HTTPS, IPSEC, etc.).

Snort¹ est un NIDS open-source intégrant :

- un moteur de détection par signatures très performant (ces dernières sont décrites par une syntaxe souple et puissante);
- différents préprocesseurs permettant de normaliser les données, aussi bien au niveau réseau qu'au niveau applicatif;
- de nombreux composants additionnels (*plugins*).



1. Projet commencé fin 1998 par Marty Roesch, sponsorisé actuellement par la société SourceFire.

- 1 Ne pas hésiter à passer du temps pour bien comprendre la syntaxe du langage des règles et le rôle des différents modules.
- 2 Il faut rechercher une adéquation la plus complète possible du produit au système d'informations, pour cela :
 - ne pas hésiter à partir d'un jeu de règles complètement vide et à augmenter ce dernier en fonction des besoins;
 - ne pas hésiter à se lancer dans l'écriture de règles spécifiques.
- 3 Rechercher à être le plus pragmatique possible au niveau de la remontée des alertes (en pratique les consoles de supervision fournies avec Snort sont loin d'être idéales).

Question

Si le système de détection d'intrusion est capable (au moins en théorie!) de détecter une attaque, pourquoi ne pas en profiter pour la neutraliser?

- parce que dans ce cas, l'IDS ne sera plus invisible;
- parce que c'est une mauvaise idée en cas de faux-positif;
- parce que l'attaquant n'est peut-être pas celui que l'on croit;
- parce qu'il ne faut pas confondre **protection** et **surveillance**.

L'analyse de flux

On définit un **flux réseau** comme un échange de données délimité dans le temps entre deux points du réseau. La délimitation peut être :

- marquée au niveau du protocole (par exemple, *flags TCP* : SYN, FIN, etc.);
- purement temporelle (*timeout*).

En détection d'intrusions, l'analyse de flux consiste :

- 1 à enregistrer l'ensemble des flux que l'on « voit » transiter au niveau des points de capture;
 - compromis idéal entre la **capture complète** de trames et la **métrologie**
- 2 à faire de la recherche d'anomalies en s'intéressant notamment :
 - aux aspects quantitatifs : débit, volumétrie, durée, etc.
 - aux acteurs : qui a parlé avec qui ? et quand ?

L'analyse de flux est particulièrement adaptée à la détection :

- des scans réseaux;
- de trafic suspect « post-intrusion »;
- de contournement de politiques de sécurité (canaux cachés, P2P, etc.).

Netflow est une technologie développée par Cisco dans les années 1990¹.

Au sens Netflow, un flux est **unidirectionnel**, la majorité du temps une connexion réseau va donc générer deux enregistrements.

Architecture de collecte Netflow

- chaque équipement réseau (supportant la technologie) maintient une table des flux ouverts (pas de stockage permanent);
- dès qu'un flux est fermé, un enregistrement est généré puis envoyé en UDP vers un collecteur central;
- ce dernier enregistre les informations reçues dans une base de données;
- les données peuvent ensuite être analysées depuis une console d'administration avec des outils spécifiques.

Exemples d'outils gratuits : fprobe, flow-tools

1. initialement pour des besoins d'*accounting*.

Caractéristiques et fonctionnalités

- contrairement à Netflow, les flux au sens Argus sont **bi-directionnels**;
- le programme principal (« serveur ») capture le trafic (ou lit un fichier PCAP) et enregistre les flux dans des fichiers sur le disque;
- différents outils en ligne de commandes (« clients argus ») peuvent être utilisés pour consulter/requêter les fichiers.

Argus (2/2)

Exemple : Liste des fichiers Argus générés chaque jour par une capture sur un lien à 150 MBit/s (noter le support gzip natif et la taille des fichiers) :

```
# ls -alrt /argus
-rw-r--r-- 1 argus argus 72763637 fév 29 23:47 db.log.20080229.gz
-rw-r--r-- 1 argus argus 19861748 mar  1 23:46 db.log.20080301.gz
-rw-r--r-- 1 argus argus 18725864 mar  2 23:46 db.log.20080302.gz
-rw-r--r-- 1 argus argus 82636148 mar  3 23:46 db.log.20080303.gz
-rw-r--r-- 1 argus argus 83317377 mar  4 23:46 db.log.20080304.gz
```

Exemple : Extraire les connexions SSH sortantes le 2 avril 2009 :

```
# ra -nn -z -r db.log.20090402.gz - 'tcp and dst port ssh'
XXX
```

Historiquement, les mesures de protection déployées prennent en compte les menaces depuis l'extérieur vers l'intérieur (protection périmétrique).

Actuellement, ces barrières sont devenues relativement solides :

- présence de filtrage réseau;
- serveurs généralement « *secure by default* »;
- (un peu) moins de vulnérabilités coté serveur pour les applications web.

Conséquence : réorientation de la cible coté client

- plus proche des données, des mots de passe, etc.
- ciblage relativement simple (par exemple via les réseaux sociaux);
- la menace interne est souvent négligée (→ l'utilisateur peut être la source ou le vecteur).

Une fois que l'attaquant a un pied dans le SI, le tour est joué (*crunchy networks*¹).

1. Un réseau dur à l'extérieur, mais tendre et moelleux à l'intérieur.

La détection d'extrusion consiste à détecter les machines compromises situées à l'intérieur du périmètre de sécurité, en se focalisant sur **la recherche d'anomalies dans le trafic sortant**.

Remarque

Cet objectif ne peut pas être réalisé si tout le trafic sortant est autorisé; en effet dans ce cas, comment peut-on distinguer ce qui est légitime de ce qui ne l'est pas ?

Architecture réseau « défendable »

C'est une architecture résistante aux intrusions, c'est-à-dire facilitant la détection de comportements anormaux.

Observable n'importe quel point du réseau peut être surveillé;

Contrôlée la liberté de mouvement de l'attaquant doit être limitée;

Minimisée la surface d'exposition doit être réduite;

Maintenue à jour pas d'exploitation possible de vulnérabilités anciennes.

- HoneyPot
- 802.1X
- Logs ACLs
- Tarpit
- QOS, mac tcp-conn, fail2ban, null routage

Commissariat à l'énergie atomique et aux énergies alternatives
Centre de Bruyères-le-Châtel | 91297 Arpajon Cedex
T. +33 (0)1 69 26 40 00 | F. +33 (0)1 69 26 40 00
Établissement public à caractère industriel et commercial
RCS Paris B 775 685 019

CEA/DAM/DIF