Snort : détection d'intrusion

Contributeur : (SISR2-2017- Melvyn Bruneau)

Présentation

Snort un outil qui permet de détecter des comportements anormaux sur un réseau informatique et :

- d'alerter sur une console ou à un destinataire en cas d'activation d'une règle afin d'intervenir le plus rapidement lors d'attaques, d'indisponibilités des services, ou d'équipements
- de journaliser les évènements de manière à pouvoir remonter la démarche ou l'historique d'une attaque

De plus il peut venir en complément d'un pare-feu.

Un tel outil est capable de détecter des événements comme :

- Défaillance d'un service ou d'un système par débordement stack overflow
- tentative de *scan de port* sur une machine
- augmentation subite du trafic sur une interface réseau
- augmentation anormale du temps de réponse d'un équipement ou d'un service
- circulation d'un protocole illicite

Afin de prévenir ce genre de risque, la mise en place de l'outil **snort** qui est libre et gratuit est une solution.

L'installation sera effectuée sur une machine physique connectée sur un port en miroir sur un switch pour que le serveur IDS puisse capturer un maximum de données sur le réseau.

Installation de l'outil snort

La description correspond à une installation sur un os Debian 8. Une fois Debian 8 installé, il faut vérifier que celui-ci est bien à jour. Pour cela effectuer les commandes ***apt-get update***, et ***apt-get upgrade***.

Snort fait partie des packages disponibles dans les sources officielles.

```
apt install snort
```

Ancienne versionpour une installation manuelle de snort

3 Test de la configuration snort

Afin de vérifier la bonne configuration du fichier snort.conf il est possible d'utiliser la commande suivante

```
snort -T -c /etc/snort/snort.conf
```

La configuration du fichier est validée.

Configuration du port miroir

Le miroir de port permet de renvoyer le trafic d'un port d'un switch sur un autre port. Dans notre cas afin d'avoir accès a un maximum de données on va faire un miroir de port entre un port 26 en mode trunk vers le port ou est brassé notre serveur IDS le port 4.

Configuration pour un switch Cisco

Le mirroring de port(s) sur Cisco passe par des ***sessions*** qui prennent une ou plusieurs ***sources*** vers une ***destination***.

La syntaxe est :

```
//pour les ports source
monitor session <numero_session_mirroring> source interface <nom_interface>
<sens_trafic>
//pour le port destination
monitor session <numero_session_mirroring> destination interface
<nom_interface>
```

- La ***source*** peut être un ***port***, une ***plage de ports*** ou un ***vlan***.
- On peut rediriger le trafic reçu (***rx***), transmis (***tx***) ou les deux (***both***)
- La ***destination*** est un port unique.

Exemples

```
monitor session 1 source interface fa 0/1 both
monitor session 1 source interface fa0/2 - fa0/4 tx
monitor session 1 source interface fa0/6 - fa0/8 rx
monitor session 1 destination interface fa0/2
```

On peut vérifier le paramétrage :

```
show monitor session 1
```

Configuration pour le switch D-link

Se rendre dans ***monitoring/ mirror/ port mirror settings*** .

Dans un premier temps il faut le state à *enable* pour activer le miroir de port ensuite dans *target port* il faut sélectionner le port de renvoi des données ici le port 4 enfin sur le port 26 il faut sélectionner *both* pour que le port 4 puisse envoyer et recevoir des paquets. *Rx* c'est pour recevoir, et *Tx* c'est pour la transmission.

Configuration de règles snort

L'outil snort est maintenant opérationnel, il faut donc créer des règles afin de pouvoir détecter les éventuels problèmes sur le réseau (intrusion, panne de services, etc....). IL est possible de récupérer des règles directement sur le site snort.

Pour configurer des règles personnelles il faut aller dans le fichier ***/etc/snort/rules/local.rules***.

Voilà un exemple de règle créée pour la détection de scan de port sur le serveur portefeuille.

Une règle est construite avec différentes valeurs qui sont les suivantes.

Information	Explication
Types d'actions de la règle	il existe différents types d'actions : * alert - génère une alerte en utilisant la méthode d'alerte sélectionnée, et alors journalise le paquet * log - journalise le paquet * pass - ignore le paquet * activate - alerte et alors active une autre règle dynamic * dynamic - reste passive jusqu'à être activée par une règle activate, alors agit comme une règle log
le protocole à analyser entre TCP, UDP, ICMP	Dans notre cas c'est le protocole TCP qui est analysé.
L'adresse IP source avec son masque de sous-réseau que l'on souhaite analyser.	Dans notre cas la valeur est a any pour analyser toutes les adresses IP sources.
Le port utilisé par l'adresse IP source.	Cela peut-être un port défini (80, 22, 21) une plage de port 1 : 80 pour les ports de 1 à 80 par exemple. ou any pour analyser tous les ports.
L'opérateur de direction	Il permet de définir le sens du trafic : il y a trois types de directions : * → Le trafic venant de la source extérieure vers un réseau, ou une machine spécifique dans notre cas c'est l'analyse de tous les adresses IP vers le serveur Portefeuille 172.20.x.x/32 * ← Le trafic inverse c'est-à-dire venant du réseau interne vers l'extérieur * < > Le trafic bidirectionnel qui va analyser les échanges dans les deux sens.
L'adresse IP et le masque de sous réseau de la destination.	

Information	Explication
Le port de l'adresse IP	ici any pour tous les ports.
L'option msg	Elle va permettre d'entrer un message spécifique afin que l'utilisateur puisse comprendre rapidement.
Sid	C'est l'identifiant unique pour identifier les règles il y a trois types : * <100 Réservé pour une utilisation future 100-999,999 Règles incluses dans la distribution de Snort >1 000 000 Utilisé pour les règles locales Dans notre cas la règle a un Sid 1 000 000 2 pour signifier que c'est la deuxième règle .
La révision	Elle indique un numéro de version, permettant de faire évoluer une règle dans le temps Il faut faire très attention lors de la création de règle, c'est sensible à la casse, et le lancement d'analyse ne se lance pas.

Voici ci-dessous des exemples de règles locales :

La règle numéro une va analyser les tentatives d'accès root aux nas, pour ce qui est de la seconde règle c'est l'analyse des scans port sur le serveur de portefeuille. Et enfin la dernière règle va analyser un accès FTP (port 21) non autorisé sur le serveur portefeuille.

Liste des liens internet intéressant pour la construction de règles snort.
http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node31.html#keyword_sid
http://madchat.fr/reseau/ids%7Cnids/L'%E9criture%20de%20r%E8gles%20Snort.htm

Lancement de l'analyse

Snort a besoin d'être lancé pour remonter les alertes lors de problèmes sur le réseau. Il y a plusieurs types de commande de lancement.

- Mode détection d'intrusion qui permet de remontée des alertes et prendre des décisions si nécessaires.

```
 /usr/local/bin/snort -A console -q -u snort -g snort -c
/etc/snort/snort.conf -i eth0
```

Toutes les alertes sont remontées sur la console automatiquement. Exemple en mode d'intrusion lors d'un accès au Nas avec le compte root ou une règle informe une tentative de connexion en tant que utilisateur root. De plus un fichier log va être créé automatiquement lors des alertes.

- Mode écoute est le mode sniffer c'est-à-dire qu'il analyse tous les paquets du réseau qui se lance par la commande snort-v la commande va générer beaucoup d'informations car il écoute tous les communications mais ne produit pas de fichier log.

Lecture des logs

Les fichiers Logs sont stocké dans le répertoire ***/var/log/snort*** au format *Tcpdump log* pour lancer la lecture d'un fichier il faut utiliser la commande

```
tcpdump –r <nomdufichier>
```

.

Dans le log de l'accès au nas par l'utilisateur root on retrouve l'adresse IP source vers le serveur nas avec le protocole FTP et l'utilisateur root.

Test de l'outil snort

Test capture de communication

Pour finaliser l'intégration de l'outil il faut effectuer différents tests. Tout d'abord il faut vérifier que celui-ci capture toutes les données sur le réseau.

Lancement de la commande snort-v et attendre les retours voire si on capture bien des paquets. Au lancement de la commande on obtient l'initialisation.

Ensuite on a le retour de nombreux paquets par exemple celui-ci-dessous lors d'une connexion sur le nas.

On retrouve bien la source vers le Nas avec le protocole TCP. Ensuite il va falloir arrêter la capture un ctrl+C suffit.

Une fois la capture arrêtée on a un récapitulatif sur la capture les paquets entrant et sortant, ainsi que les paquets qui ont été analyse. De plus on a aussi des informations sur les protocoles utilisés sur le réseau TCP, UDP, ARP, etc. … avec le nombre de paquets et le pourcentage.

Le test de la capture est donc validé car il y a eu des captures TCP, UDP, ICMP.

Test de détection d'un scan de port

Le scan de port va permettre de trouver quels ports son ouvert sur une machine afin de trouver un protocole avec lequel on peut essayer d'avoir accès à une machine. Le but de la détection est de remontée l'information à l'utilisateur afin qu'il puisse bloquer le port concerné et évité l'intrusion.

Dans notre cas on à la règle suivante va permettre de remontée une tentative de scan de port vers la machine portefeuille.

Lors de scans de port on doit avoir une alerte qui est remontée dans la console et inscrite dans le fichier log automatique lors du lancement de la détection d'intrusion.

Lancement de la console de détection.

Une fois la console lancée il va falloir lancer des scans de port via une machine ici sous Windows avec un logiciel.

Le scan des ports nous montre qu'il y a différents ports ouverts sur le serveur portefeuille on a donc les ports sur lesquels il peut y avoir des tentatives d'attaque.

Le serveur d'intrusion a joué son rôle car il a remonté le scan des ports sur le serveur portefeuille dans la console.

Les résultats obtenus sont le précèdent on voit bien que l'adresse IP 10.9.0.14 a contacté plusieurs fois le serveur 172.20.20.20 sur différents ports 443, 80, 554, 1720, 3306, 199... En peu de temps avec le message scan port portefeuille. La détection est donc opérationnelle.

Détection d'accès non autorisé

La détection d'accès non autorisé va permettre de remontée les alertes lorsqu'un utilisateur tente de se connecter avec un compte sur une machine ou services.

Par exemple l'accès en root au serveur nas avec la règle suivante.

Il faut relancer la console de détection d'intrusion et essayer une connexion avec l'utilisateur root au nas.

La console ids remonte bien une tentative d'accès au nas avec l'utilisateur root de l'adresse 10.9.0.14. La détection d'accès est donc opérationnelle.

Détection d'un protocole non autorisé

La détection d'un protocole non autorisé est importante, effet il peut permettre d'avertir d'un problème sur le réseau (attaque, scan de port).

L'accès au serveur portefeuille via le port 21 était non autorisé. La règle suivante va donc remonter une alerte lorsqu'il y a une tentative d'accès via le port 21.

Tentative d'accès au serveur portefeuille en telnet avec le port 21.

Le serveur de détection d’intrusion remonte bien l’alerte de tentative d’accès au portefeuille avec le port 21 la détection de protocole non autorisé est opérationnelle.