Déploiement d'une solution de détection d'intrusion

Contexte : Mise en place d'un système IDS avec Suricata sur Debian

MANANTSOA GEORGES LOVATIANA

Sommaire

Introduction

Ce rapport présente la mise en œuvre d'une solution de détection d'intrusion (IDS) basée sur Suricata, déployée sur un système d'exploitation Debian. Dans un contexte où les menaces informatiques sont de plus en plus sophistiquées, il est essentiel de disposer de mécanismes permettant de détecter les activités malveillantes sur le réseau. Ce projet s'est concentré sur la configuration d'un IDS capable de détecter plusieurs types d'attaques, notamment les scans, les tentatives d'intrusion sur les services réseau et les attaques par déni de service.

Objectifs du projet

Les principaux objectifs de ce projet étaient :

1. Installer et configurer Suricata sur une distribution Debian

2. Développer des règles de filtrage personnalisées pour détecter différentes formes d'attaques
3. Configurer un environnement de test comprenant un serveur FTP vulnérable
4. Simuler diverses attaques et évaluer l'efficacité du système de détection
5. Analyser les performances et proposer des améliorations potentielles

Environnement technique

Infrastructure

- Machine virtuelle principale : Debian 11, 2 Go RAM, 20 Go stockage
- Machine virtuelle secondaire (attaquante) : Debian 11, 1 Go RAM, 20 Go stockage
- Réseau virtuel isolé permettant la communication entre les deux machines

Logiciels utilisés

- Suricata 6.0.8 (système de détection d'intrusion)
- vsftpd 3.0.3 (serveur FTP)
- Hping3 (outil de test réseau et d'attaque)
- Hydra (outil de force brute pour les services réseau)

Méthodologie

La méthodologie adoptée pour ce projet a suivi une approche progressive :

1. **Phase de préparation** : Installation des outils nécessaires et configuration de l'environnement de test
2. **Phase de configuration** : Paramétrage de Suricata et création des règles de détection
3. **Phase de test** : Simulation de différentes attaques pour évaluer la capacité de détection
4. **Phase d'analyse** : Évaluation des résultats et ajustement des configurations

Cette approche a permis de développer et d'améliorer itérativement la solution de détection d'intrusion.

Implémentation

Installation et configuration de Suricata

L'installation de Suricata a été réalisée à l'aide des commandes suivantes :

sudo apt update

sudo apt install suricata

La configuration de base a nécessité des modifications du fichier principal :

sudo nano /etc/suricata/suricata.yaml

Les principaux paramètres modifiés sont :

- HOME_NET: "any" pour faciliter les tests
- Définition du chemin des règles : default-rule-path: /etc/suricata/rules
- Inclusion du fichier de règles personnalisées : local.rules

Développement des règles de détection

Des règles personnalisées ont été créées pour détecter différents types d'attaques :

- Détection des pings ICMP

alert icmp any any -> $HOME_NET any (msg:"ICMP ping détecté"; sid:1000001; rev:1;)

- Détection des connexions FTP

alert tcp any any -> $HOME_NET 21 (msg:"Connexion FTP détectée";

flow:established,to_server; sid:1000002; rev:1;)

- Détection des connexions SFTP

alert tcp any any -> $HOME_NET 22 (msg:"Connexion SFTP détectée";

flow:established,to_server; sid :1000003 ; rev :1;)

```
root@debian:/home/lovatiana# grep -n "" /etc/suricata/rules/local.rules
1:alert icmp any any -> $HOME_NET any (msg:"ICMP ping détecté"; sid:1000001; rev:1;)
2:alert tcp any any -> $HOME_NET 21 (msg:"Connexion FTP détectée"; flow:established,to_se
rver; sid:1000002; rev:1;)
3:alert tcp any any -> $HOME_NET 22 (msg:"Connexion SFTP détectée"; flow:established,to_s
erver; sid:1000003; rev:1;)
4:alert tcp any any -> $HOME_NET 21 (msg:"Tentative de connexion FTP échouée"; flow:estab
lished,to_server; content:"530 Login incorrect"; sid:1000004; rev:1;)
5:alert tcp any any -> $HOME_NET any (msg:"Possible SYN port scan detected"; flags:S; thr
eshold: type both, track by_src, count 11, seconds 5; classtype:attempted-recon; sid:1000
002; rev:1;)
root@debian:/home/lovatiana#
```

Configuration du serveur FTP

Pour tester les règles de détection liées au FTP, un serveur vsftpd a été installé et configuré :

sudo apt install -y vsftpd

sudo nano /etc/vsftpd.conf

Les paramètres suivants ont été définis :

sudo adduser ftpuser

```
root@debian:/home/lovatiana# grep ftpuser /etc/passwd
ftpuser:x:1002:1002:,,,:/home/ftpuser:/bin/bash
root@debian:/home/lovatiana#
```

Tests et validation

Scénarios de test

Plusieurs scénarios d'attaque ont été simulés pour évaluer l'efficacité du système de détection :

1. Détection de ping simple

- Depuis la machine attaquante

```
root@debian:/home/georges# ping 192.168.117.145
PING 192.168.117.145 (192.168.117.145) 56(84) bytes of data.
64 bytes from 192.168.117.145: icmp_seq=1 ttl=64 time=1.70 ms
64 bytes from 192.168.117.145: icmp_seq=2 ttl=64 time=1.83 ms
64 bytes from 192.168.117.145: icmp_seq=3 ttl=64 time=1.60 ms
64 bytes from 192.168.117.145: icmp_seq=4 ttl=64 time=1.07 ms
64 bytes from 192.168.117.145: icmp_seq=5 ttl=64 time=1.50 ms
64 bytes from 192.168.117.145: icmp_seq=6 ttl=64 time=0.765 ms
^C
--- 192.168.117.145 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 0.765/1.411/1.831/0.374 ms
root@debian:/home/georges#
```

- Depuis la machine attaquée

```
root@debian:/home/lovatiana# sudo tail -f /var/log/suricata/fast.log
04/07/2025-11:13:23.896314  [**] [1:1000001:1] ICMP ping détecté [**] [Classification: (null)] [P
riority: 3] {ICMP} 192.168.117.147:8 -> 192.168.117.145:0
04/07/2025-11:13:23.896455  [**] [1:1000001:1] ICMP ping détecté [**] [Classification: (null)] [P
riority: 3] {ICMP} 192.168.117.145:0 -> 192.168.117.147:0
```

2. Attaque SYN Flood sur le port FTP

Depuis la machine attaquante

sudo hping3 -S --flood -p 21 [IP_cible]

3. Attaque par force brute sur FTP

Depuis la machine attaquante

echo "password123" > wordlist.txt

echo "admin" >> wordlist.txt

echo "123456" >> wordlist.txt

hydra -l ftpuser -P wordlist.txt [IP_cible] ftp

4. Attaque par inondation ICMP (Ping Flood)

Depuis la machine attaquante

ping -f [IP_cible]

Vérification des alertes

Pour chaque test, les alertes générées par Suricata ont été surveillées :

sudo tail -f /var/log/suricata/fast.log

Analyse des résultats

Efficacité de la détection

Type d'attaque	Détection	Règle déclenchée	Observations
Ping simple	□	sid:1000001	Détection immédiate de chaque requête ICMP
SYN Flood	□	Règle intégrée Suricata	Génération de plusieurs alertes indiquant une attaque DoS
Force brute FTP	□	sid:1000004	Détection des tentatives de connexion échouées
Ping Flood	□	sid:1000001	Nombreuses alertes générées, parfois agrégées

Conclusion

Ce projet a permis de démontrer l'efficacité de Suricata comme solution de détection d'intrusion. Les tests ont confirmé la capacité du système à identifier différentes formes d'attaques réseau, des plus simples (ping) aux plus complexes (force brute, déni de service).

L'approche modulaire de la configuration et des règles de détection offre une grande flexibilité pour adapter le système aux besoins spécifiques de sécurité d'une infrastructure. De plus, la relative simplicité de mise en œuvre rend cette solution accessible même pour des environnements disposant de ressources limitées.

Les améliorations proposées permettraient d'étendre davantage la portée et l'efficacité de la solution, notamment en intégrant des mécanismes de détection plus avancés et des outils de gestion des alertes plus sophistiqués.

Annexes

A. Fichier de configuration complet de Suricata

Configuration principale de Suricata

/etc/suricata/suricata.yaml

Définition du réseau local

vars:

```
  address-groups:
    HOME_NET: "any"
    EXTERNAL_NET: "!$HOME_NET"

  port-groups:
    HTTP_PORTS: "80"
    SHELLCODE_PORTS: "!80"

# Configuration des interfaces d'écoute
af-packet:
  - interface: ens33
    cluster-id: 99
    cluster-type: cluster_flow
    defrag: yes

# Configuration des règles
default-rule-path: /etc/suricata/rules
rule-files:
  - suricata.rules
  - local.rules

# Configuration des sorties
outputs:
  - fast:
      enabled: yes
      filename: fast.log
      append: yes

  - eve-log:
```

```
    enabled: yes
    filetype: regular
    filename: eve.json
    types:
      - alert
      - http
      - dns
      - tls
      - files
      - ssh
      - flow
```

B. Exemple d'alertes générées

04/07/2025-10:15:32.123456 [**] [1:1000001:1] ICMP ping détecté [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.1.10:8 -> 192.168.1.20:0

04/07/2025-10:17:48.654321 [**] [1:1000002:1] Connexion FTP détectée [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.1.10:52146 -> 192.168.1.20:21

04/07/2025-10:22:15.789123 [**] [1:1000004:1] Tentative de connexion FTP échouée [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.1.10:52150 -> 192.168.1.20:21

C. Scripts de test complets

```
#!/bin/bash
# Script de test d'attaques réseau

# Variables
TARGET_IP="192.168.1.20"
FTP_USER="ftpuser"

# Test 1: Ping simple
echo "[+] Test 1: Ping simple"
```

```
ping -c 4 $TARGET_IP
sleep 2

# Test 2: SYN Flood
echo "[+] Test 2: SYN Flood sur port FTP"
sudo hping3 -S --flood -p 21 $TARGET_IP -c 100
sleep 2

# Test 3: Force brute FTP
echo "[+] Test 3: Force brute FTP"
echo "password123" > /tmp/wordlist.txt
echo "admin" >> /tmp/wordlist.txt
echo "123456" >> /tmp/wordlist.txt
hydra -l $FTP_USER -P /tmp/wordlist.txt $TARGET_IP ftp
sleep 2

# Test 4: Ping Flood
echo "[+] Test 4: Ping Flood"
ping -f $TARGET_IP -c 100
sleep 2

echo "[+] Tests terminés"
```