## Introducció als IDS/IPS:

## En el món de la seguretat informàtica, els sistemes IDS i IPS són eines essencials per protegir els equips i les xarxes contra atacs. Aquests sistemes serveixen per detectar comportaments estranys o sospitosos i, en alguns casos, també poden actuar per evitar que l’atac es dugui a terme.

## Un IDS (Intrusion Detection System) és un sistema que només observa. Monitoritza el trànsit de xarxa o l'activitat del sistema i, si veu alguna cosa rara, genera una alerta. Però no pot fer res per aturar l’atac, només pot avisar.

## En canvi, un IPS (Intrusion Prevention System) fa el mateix que un IDS però, a més, pot intervenir. Si detecta una amenaça, pot bloquejar-la automàticament, com per exemple tallar la connexió de l’atacant.

## Per què és important tenir un IDS o un IPS?

## Actualment, qualsevol dispositiu o servidor connectat a internet pot ser atacat. No cal ser una gran empresa per ser víctima d’un atac: fins i tot un simple servidor a casa pot rebre intents d'intrusió.

## Per això és important tenir un sistema IDS o IPS instal·lat, ja que:

## Et permet detectar atacs abans que es converteixin en un problema greu.

## Et genera logs i alertes que poden servir per investigar què ha passat.

## Si és un IPS, pot arribar a bloquejar automàticament l’atacant abans que faci mal.

## En resum, són com una alarma per al teu sistema. I si és un IPS, a més de sonar, tanca la porta abans que entrin.

## Amenaces més comunes que un IDS/IPS pot detectar

Aquests sistemes poden identificar i alertar sobre diversos tipus d’atacs. Alguns exemples reals:

|  |  |  |
| --- | --- | --- |
| Tipus d’atac | Exemple real | Com ajuda un IDS/IPS? |
| Escaneig de ports | Algú fa un nmap per veure quins serveis tens oberts. | IDS t’avisa que un atacant està intentant trobar punts febles. |
| Atacs de força bruta | Intenten entrar al teu servidor SSH provant moltes contrasenyes. | L’IPS pot bloquejar l’IP de l’atacant automàticament. |
| Injecció SQL | Un hacker prova d’accedir a dades modificant una URL. | L’IDS detecta consultes sospitoses i avisa abans que sigui massa tard. |
| Modificació de fitxers del sistema | Un intrús aconsegueix accés root i canvia /etc/passwd. | OSSEC pot detectar aquests canvis i alertar-te. |

## Comparació entre sistemes IDS/IPS disponibles

La diferencia entre **Snort, Suricata i OSSEC**:

|  |  |  |  |
| --- | --- | --- | --- |
| Característica | Snort | Suricata | OSSEC |
| Tipus | IDS/IPS de xarxa | IDS/IPS de xarxa | IDS basat en host |
| Rendiment | Moderat | Alt (Multithreading) | Depèn del sistema |
| Facilitat de configuració | Complexa | Més fàcil que Snort | Relativament senzill |
| Detecció de signatures | Basat en regles (Snort rules) | Compatible amb Snort rules | Monitoritza canvis en fitxers i activitats sospitoses |
| Prevenció activa | Sí | Sí | Sí (amb active response) |
| Cas d’ús principal | Protecció de xarxes | Xarxes grans amb alts volums de tràfic | Protecció de servidors i sistemes operatius |

## Anàlisi de requisits i selecció de la tecnologia més adequada

## Abans d’escollir quin sistema IDS/IPS utilitzar, cal tenir clar què necessitem i quin tipus d’entorn volem protegir. No és el mateix vigilar tota una xarxa que controlar només un servidor.

## Algunes coses que hem de tenir en compte:

## **Monitorització de xarxa o d’host:**

## Si només volem controlar els servidors (per exemple, un Ubuntu amb serveis com SSH o Apache), OSSEC és molt bona opció. En canvi, si volem controlar tot el tràfic de xarxa, millor tirar per Snort o Suricata.

## **Rendiment i volum de tràfic:**

## Si el sistema ha de gestionar molt de tràfic (com en una empresa gran), Suricata és més eficient perquè treballa millor amb múltiples fils (multithreading).

## **Facilitat de configuració:**

## OSSEC és més senzill d’instal·lar i posar en marxa en una màquina concreta. En canvi, Snort i Suricata poden ser més complicats, sobretot si es vol aplicar a tota la xarxa.

## En el nostre cas, com que volem protegir un servidor concret, OSSEC ens encaixa molt bé.

## Justificació de la selecció

## Hem triat OSSEC perquè és un HIDS (Host-based Intrusion Detection System), o sigui, un sistema que es centra en vigilar què passa dins del propi servidor. És ideal per controlar màquines amb serveis exposats a internet.

## Alguns dels motius principals per escollir-lo són:

## Pot detectar canvis sospitosos en fitxers importants, com per exemple si algú modifica /etc/passwd per afegir un usuari il·legalment.

## Inclou la funció de “Active Response”, que permet bloquejar automàticament una IP si es detecta un comportament estrany.

## És lleuger, i es pot instal·lar fàcilment en servidors Linux sense que afecti gaire el rendiment del sistema.

## A més, la configuració és força senzilla, i això el fa perfecte per a entorns educatius o per protegir servidors concrets.

## INSTALACIÓ DE OSSEC

Baixem OSSEC

Imatge que conté text, captura de pantalla, Font

Pot ser que el contingut generat amb IA no sigui correcte.

Descomprimir l’arxiu

Instalacio de OSSEC

Imatge que conté text, captura de pantalla, Font

Pot ser que el contingut generat amb IA no sigui correcte.  
Language: en (Idioma que estarà)

Install type: local (Instal·lació bàsica sense servidor centralitzat)

Enable email notification? no (Notifica per correu si es necesari, en el nostre cas no volem)

Enable integrity check? yes (Monitoritzarà canvis en arxius importants)

Enable rootkit detection? yes (Detectarà possibles rootkits)

Enable active response? yes (Bloquejarà automàticament alguns atacs)

Imatge que conté text, captura de pantalla, Font

Pot ser que el contingut generat amb IA no sigui correcte.

systemctl start ossec: Inicia OSSEC manualment.

systemctl enable ossec: Fa que OSSEC s’iniciï automàticament en arrencar el sistema.

Systemctl status ossec: comprobem que el servei s’esta executant correctament

Imatge que conté text, captura de pantalla, Font

Pot ser que el contingut generat amb IA no sigui correcte.

## Definició de Polítiques de Seguretat

Per fer que OSSEC sigui més útil al nostre entorn, hem decidit afegir una **regla personalitzada** al fitxer local\_rules.xml, que es troba a:

**/var/ossec/etc/rules/**

En el nostre cas, hem creat una regla per detectar si algú intenta connectar-se per SSH amb un usuari sospitós. En concret, volem que salti una alerta si algú fa servir l’usuari hackeruser.

Imatge que conté text, captura de pantalla, Font

Pot ser que el contingut generat amb IA no sigui correcte.

Aquesta regla l’hem afegit dins del grup següent:

**<group name="local,syslog,">**

Després de guardar la regla, hem reiniciat OSSEC per aplicar els canvis amb:

**sudo /var/ossec/bin/ossec-control restart**

Per comprovar que funciona, hem fet una connexió SSH falsa amb l’usuari sospitós:

**ssh hackeruser@IP\_SERVIDOR**

Tot i que aquest usuari no existeix, OSSEC ha detectat l’intent com a sospitós i ha generat una alerta. Per veure-ho, hem utilitzat aquest comandament:

**sudo tail -n 50 /var/ossec/logs/alerts/alerts.log | grep "hackeruser"**

Imatge que conté text, captura de pantalla, Font

Pot ser que el contingut generat amb IA no sigui correcte.

## Resposta Activa

### Objectiu:

El que volem aconseguir amb la **resposta activa** és que, a més de detectar intents d’intrusió, OSSEC sigui capaç de **reaccionar automàticament**. En aquest cas, el que hem configurat és que quan es detecti un intent de connexió sospitós (per exemple, amb l’usuari hackeruser), **OSSEC bloquegi automàticament la IP de l’atacant**.

Això ens permet evitar que la mateixa IP pugui seguir intentant atacar el sistema.

### Configuració:

Per poder activar la resposta activa, primer vam haver de definir la comanda host-deny dins del fitxer principal de configuració d’OSSEC:

**/var/ossec/etc/ossec.conf**

Aquesta comanda fa servir l’script host-deny.sh, que es troba a:

**/var/ossec/active-response/bin/**

El que fa aquest script és **afegir automàticament la IP de l’atacant al fitxer /etc/hosts.deny**, de manera que a partir d’aquell moment no podrà tornar a connectar-se via SSH.

Un cop definida la comanda, vam configurar la resposta activa perquè estigués **relacionada amb la nostra regla personalitzada amb ID 100010**, que és la que detecta intents de connexió amb l’usuari sospitós hackeruser.

Imatge que conté text, captura de pantalla, Font, disseny

Pot ser que el contingut generat amb IA no sigui correcte.

Aquesta configuració fa que OSSEC **executi la resposta activa automàticament després de 3 intents consecutius** detectats per la nostra regla personalitzada amb ID 100010.

Un cop afegits tots els blocs al fitxer de configuració, vam reiniciar OSSEC amb:

**sudo /var/ossec/bin/ossec-control restart**

### Verificació

Per comprovar que tot funcionava, vam fer diverses connexions fallides des d’una màquina externa amb IP 192.168.1.74, utilitzant l’usuari sospitós hackeruser:

ssh [hackeruser@192.168.1.69](mailto:hackeruser@192.168.1.69)

Després d’intentar-ho 3 vegades, OSSEC **va detectar l’activitat sospitosa** i va **executar la resposta activa automàticament**, activant l’script host-deny.sh, tal com es pot veure al log amb aquest comandament:

El log confirmava que la IP de l’atacant havia estat bloquejada correctament.

**sudo grep "host-deny.sh" /var/ossec/logs/active-responses.log**

### Imatge que conté text, captura de pantalla, Font Pot ser que el contingut generat amb IA no sigui correcte. Conclusions Generals

Durant aquest projecte hem pogut comprovar que **OSSEC és una eina molt útil i flexible** per protegir servidors Linux contra possibles atacs. Ens ha permès veure com funciona un sistema IDS/HIDS real i com es pot configurar per adaptar-lo a situacions concretes.

Gràcies a la creació de regles personalitzades i a la configuració de respostes actives, hem pogut:

* **Detectar intents de connexió sospitosa per SSH**, com per exemple amb l’usuari hackeruser.
* **Crear regles a mida**, pensades per al nostre entorn i necessitats de seguretat.
* **Executar accions automàtiques**, com bloquejar IPs de possibles atacants quan es detecten patrons concrets.
* **Fer proves i validacions reals**, utilitzant scripts i simulant atacs per veure que tot funcionava correctament.

Amb aquest projecte hem vist que OSSEC no només és una eina educativa, sinó que també **pot aplicar-se en entorns reals de producció**, per millorar la seguretat dels sistemes d'una manera força eficient.

### Limitacions trobades

Tot i que OSSEC ens ha resultat molt útil durant el projecte, també ens hem trobat amb algunes limitacions que cal tenir en compte:

* **Les respostes actives no s’activen amb qualsevol alerta**. Perquè funcionin, la regla ha de tenir un nivell suficientment alt o s’han de produir diversos intents seguits (per exemple, 3 o més).
* **No es poden fer proves de resposta activa amb la IP local (127.0.0.1)**, ja que OSSEC no bloqueja connexions d’aquesta IP per seguretat. Això complica una mica les proves si només tenim una màquina.
* **La gestió de logs pot ser una mica limitada per defecte**. Per veure més detalls sobre què fa OSSEC, cal afegir configuracions extres com <logall> al fitxer ossec.conf.