Detection Rules Using Snort

WE Innovate X Zero$ploit

Prepared by: Omar Hassan

Supervised by: Eng.Zeyad Mazen

Task:

- *Writing 3 different detection rules for detecting any executable download (IDS MODE)*
- *Using snort as an IPS for one of the rules*

***Setup** :*

- **Setting Up Ubuntu machine**

Setting	Recommended
RAM	2-4 GB
Disk	10-20 GB
CPU	1-2 Cores
Network	NAT/Bridged

***Snort Installation** :*

Sudo apt install snort

Choose suitable interface

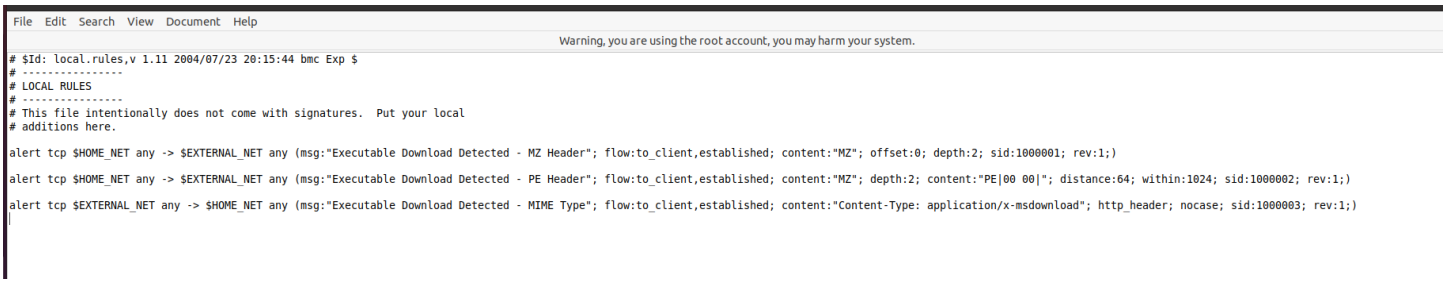
Check installation by : sudo snort –v

***Writing rules in Snort** :*

changing into the rules directory : cd /etc/snort/rules/

sudo nano local.rules

And Add this to the file :



Explanation of each rule :

alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"Executable Download Detected - MZ Header"; flow:to_client,established; content:"MZ"; offset:0; depth:2; sid:1000001; rev:1;)

This rule alerts you if someone downloads a file that starts with the MZ signature, which is how almost all Windows executable files (like .exe, .bat, .com, .dll) begin , no matter what the file is named.

Why it works: Even if the file is renamed to something like document.txt, if it's actually a Windows executable, it will still start with MZ.

alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"Executable Download Detected - PE Header"; flow:to_client,established; content:"MZ"; depth:2; content:"PE|00 00|"; distance:64; within:1024; sid:1000002; rev:1;)

This rule goes a step deeper and checks inside the file for the actual PE (Portable Executable) header, which confirms that the file is a real Windows program not just something that starts with "MZ".
Why it works: It avoids false positives by confirming the deeper structure of real EXE files.

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Executable Download Detected - MIME Type"; flow:to_client,established; content:"Content-Type: application/x-msdownload"; http_header; nocase; sid:1000003; rev:1;)

This rule listens to the HTTP response headers and alerts if a file is being downloaded with a MIME type of an executable (like application/x-msdownload).
Why it works: Even if the file is renamed (like resume.pdf), the server may still tell the truth about its type in the HTTP response headers.

*****Testing** :***

Run Snort in a terminal:
sudo snort -i <interface> -c /etc/snort/snort.conf -A console

Create a fake PE file locally:
echo -n -e "MZ$(printf '=%.0s' {1..64})PE\x00\x00" > fake.exe

Serve it via HTTP:
sudo apt install apache2
sudo cp fake.exe /var/www/html/
sudo service apache2 start

Download it from another device or same machine (adjust IP):
curl http://<ubuntu-ip>/fake.exe -o /dev/null

****Using Snort as an IPS (using NFQUEUE)** :***

Create iptables rule to redirect traffic
sudo iptables -I INPUT -p tcp --dport 80 -j NFQUEUE --queue-num 0
sudo iptables -I OUTPUT -p tcp --sport 80 -j NFQUEUE --queue-num 0

Run Snort in Inline Mode

sudo snort -Q --daq nfq --daq-var queue=0 -c /etc/snort/snort.conf -i ens33

Explanation:
-Q → Inline mode
--daq nfq → Use NFQUEUE
--daq-var queue=0 → Match the iptables queue
-c → Path to config
-i ens33 → Interface

Use a Rule that Drops & put it inside local.rules
drop tcp any any -> any 80 (msg:"BLOCK EXE DOWNLOAD"; flow:to_client, established; content:".exe"; h