NIDS Rule Creation & Testing Lab

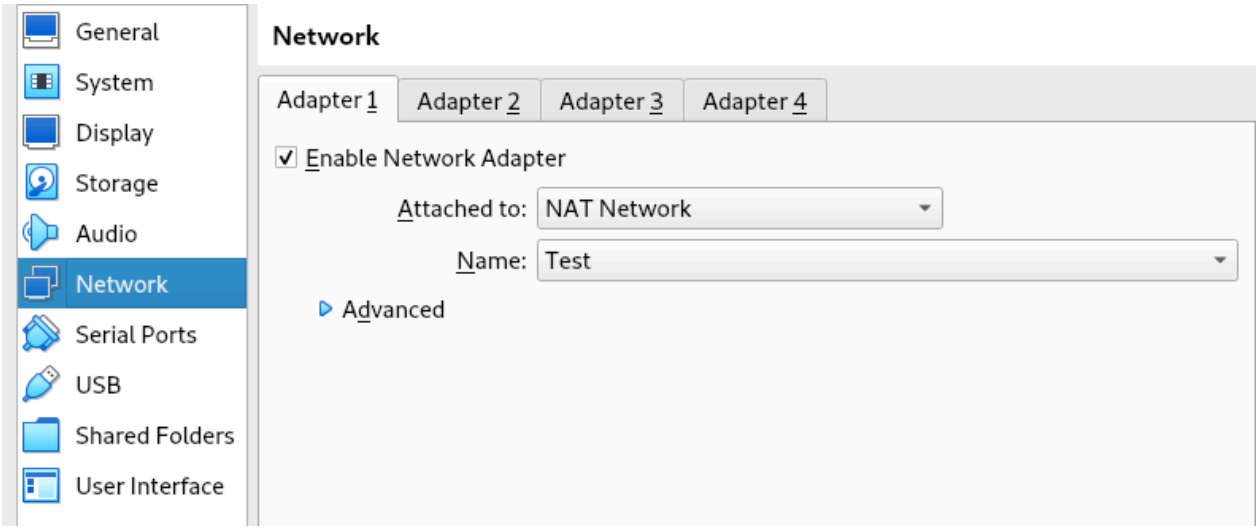
This is a detailed report on how to setup Snort, a Network Intrusion Detection System (NIDS), VM setup and network configuration and how to detect SSH brute-force attack with custom snort rules.

Setup & Installation:

We will install Virtualbox and setup two Virtual machines (Ubuntu Server & Kali Linux).

Network Configuration:

Make sure that both the machines have Bridged or NAT Network enabled. In our case we chose NAT Network and named our network 'Test' for both VMs.



Installation of Snort and other tools:

On our Ubuntu machine we will install Snort and OpenSSH server by using the following commands:

```
sudo apt install -y snort
```

After typing this command in the terminal we will be prompted to select a network range.

Since my IP is '192.168.1.9', I will set the network range to '192.168.1.0/24' which I intend to monitor. Also make sure to enter name of your primary interface (e.g: enp0s3 or eth0)

We also need to install openssh-server, we will use the following command:

```
sudo apt install -y openssh-server
```

Create a Custom NIDS Rule:

To create our own custom rules we have to edit the local.rules file located in '/etc/snort/rules/local.rules', I will use nano to edit this file. Make sure to prepend 'sudo' to these commands.

`alert tcp any any -> $HOME_NET 22 (msg:"SSH Brute-Force Attempt Detected";

flow:to_server,established; detection_filter:track by_src, count 5, seconds 60; sid:1000002;

rev:1;)` , add this command to local.rules files without quotes.

What this rule basically does is that it detects/alerts us if there have been 5 or more connection attempts to port 22 from the same source IP

within 60 seconds timeframe.

Explanation:

alert -> Action we want the snort to take (Generate alert in this case).

tcp -> Protocol (Applies this rule to Tcp traffic)

any any -> Source IP and Port (any means all)

$HOME_NET 22 -> Destination IP and Port (Traffic going to our IP(HOME_NET) on port 22(SSH).

Options inside:

msg:"SSH Brute-Force Attempt Detected" -> Message that will be displayed once the rule is triggered.

flow:to_server,established -> Match only traffic to the server side of a connection that is already established (after TCP handshake).

detection_filter:track by_src, count 5, seconds 60 -> Trigger if the same source IP (`by_src`) makes 5 or more connection attempts in 60 seconds.

sid: 10000001 -> Snort ID (unique identifier for your custom snort rule).

Always use number greater than 10000001.

rev:1 -> Revision number of the rule.

Test the Rule:

Before running snort run 'ifconfig' command on Ubuntu.

```
meer@meer:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.9  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::a00:27ff:fe03:a3b0  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:03:a3:b0  txqueuelen 1000  (Ethernet)
        RX packets 247  bytes 42838 (42.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 236  bytes 42289 (42.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 88  bytes 6860 (6.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 88  bytes 6860 (6.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

As we can see the network interface is ‘enp0s3’ and ip address is ‘192.168.1.9’.

We will run snort in console mode because we want to see the alerts in real-time:

sudo snort -A console -q -c /etc/snort/snort.conf -i enp0s3

To perform the attack from other VM or our host machine make sure to have hydra installed: Hydra is a password cracker tool used to simulate brute-force attacks.

On our Attacker VM, we will first create a dummy password file:

echo “pass123/npass/npassword/nqwerty/nroot” > pass.txt

Now we will run hydra using the command:

hydra -l non_existent_user -P pass.txt ssh://192.168.1.9

```
┌──(kali㉿kali)-[~/Desktop]
└─$ hydra -l non_existent_user -P pass.txt ssh://192.168.1.9
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-bi
hese *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-09-02 22:14:53
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 5 tasks per 1 server, overall 5 tasks, 5 login tries (l:1/p:5), ~1 try per task
[DATA] attacking ssh://192.168.1.9:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-09-02 22:14:57
```

We will check our Ubuntu machine if snort was able to generate alerts:

```
meer@meer:~$ sudo snort -A console -q -c /etc/snort/snort.conf -i enp0s3
09/03-02:14:25.221491  [**] [1:10000001:1] SSH Brute-Force Attempt Detected [**] [Priority: 0] {TCP} 192.168.1.8:45674 -> 192.168.1.9:22
09/03-02:14:25.222700  [**] [1:10000001:1] SSH Brute-Force Attempt Detected [**] [Priority: 0] {TCP} 192.168.1.8:45714 -> 192.168.1.9:22
09/03-02:14:25.331606  [**] [1:10000001:1] SSH Brute-Force Attempt Detected [**] [Priority: 0] {TCP} 192.168.1.8:45718 -> 192.168.1.9:22
09/03-02:14:25.326376  [**] [1:10000001:1] SSH Brute-Force Attempt Detected [**] [Priority: 0] {TCP} 192.168.1.8:45674 -> 192.168.1.9:22
09/03-02:14:25.336633  [**] [1:10000001:1] SSH Brute-Force Attempt Detected [**] [Priority: 0] {TCP} 192.168.1.8:45672 -> 192.168.1.9:22
09/03-02:14:25.328530  [**] [1:10000001:1] SSH Brute-Force Attempt Detected [**] [Priority: 0] {TCP} 192.168.1.8:45700 -> 192.168.1.9:22
09/03-02:14:25.338487  [**] [1:10000001:1] SSH Brute-Force Attempt Detected [**] [Priority: 0] {TCP} 192.168.1.8:45714 -> 192.168.1.9:22
```