NCY-2 Assignment 3

SNORT

Submitted by
Abdul Sami Qasim (22i-1725)
Ahmad Abdullah (22i-1609)

Submitted to
Prof. Abdullah Abid

Date of Submission: 20/10/2024

Introduction

This assignment involved using snort (an IDS) to detect some well known attacks and produce alerts upon their detection.

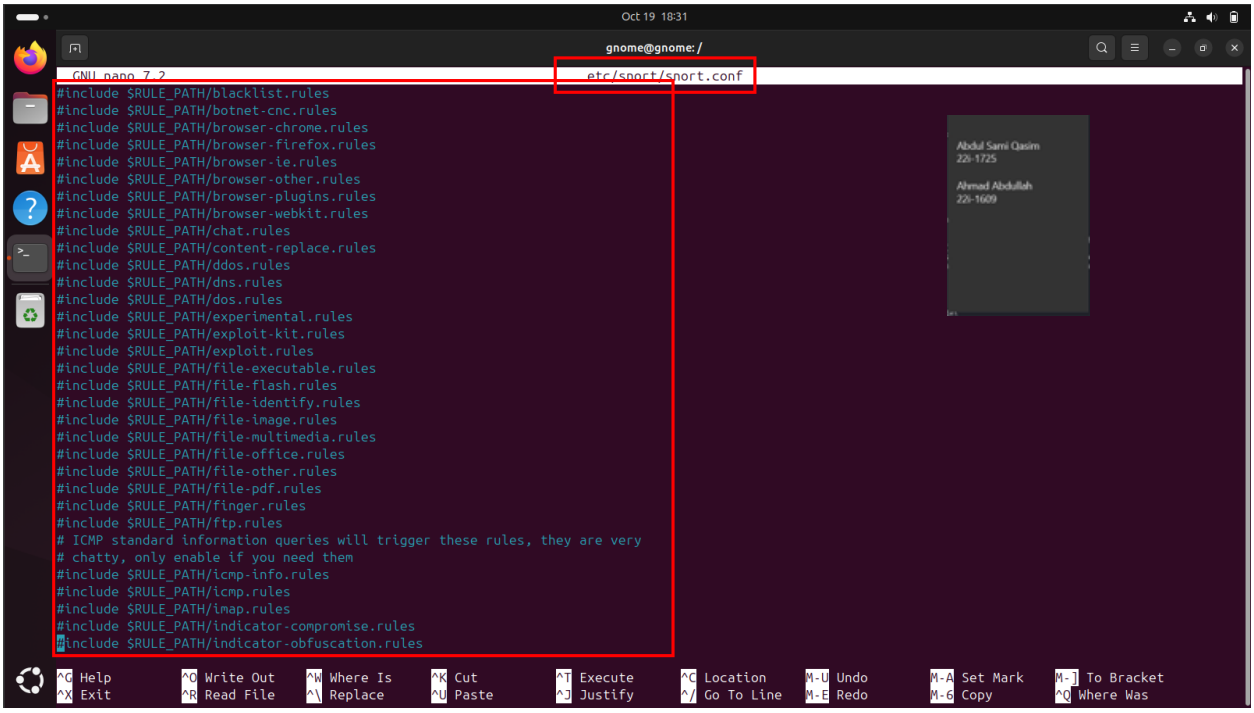
Installation and Configuration

To install snort on our devices, we used the command-line interface and input this command

“sudo apt install snort”

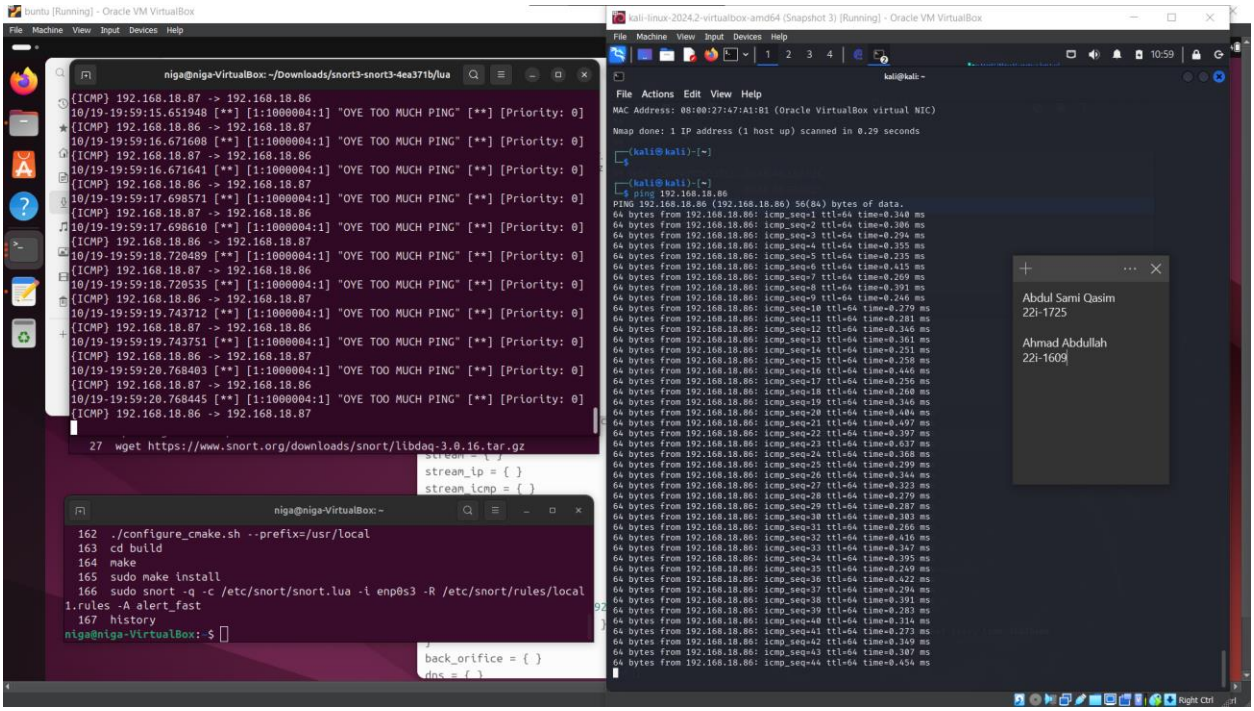
After the installation, a simple command like “snort -V” is able to give the output to see if snort is correctly installed or not.

The next step was to configure snort so that we are able to do our task as smoothly as possible. Firstly, we had to comment on all the other rules in the *snort.conf* file so that snort only checks packets based on our rules only.



Attacks and Simulation

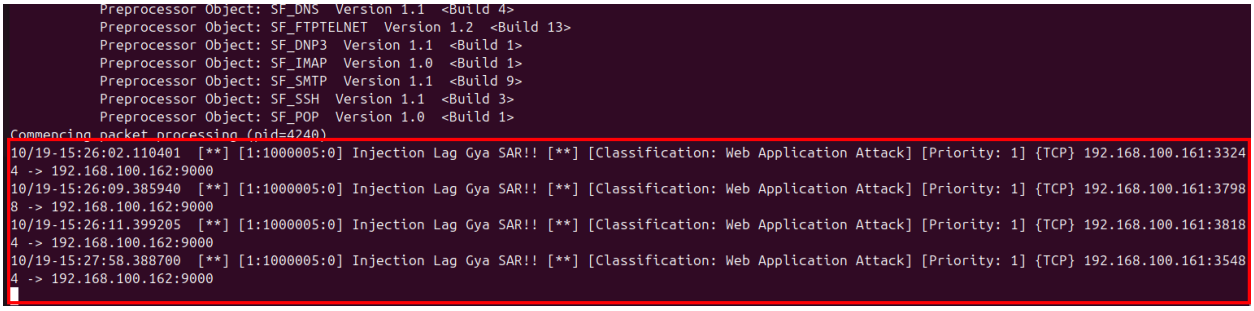
1. DDOS/DOS Attack



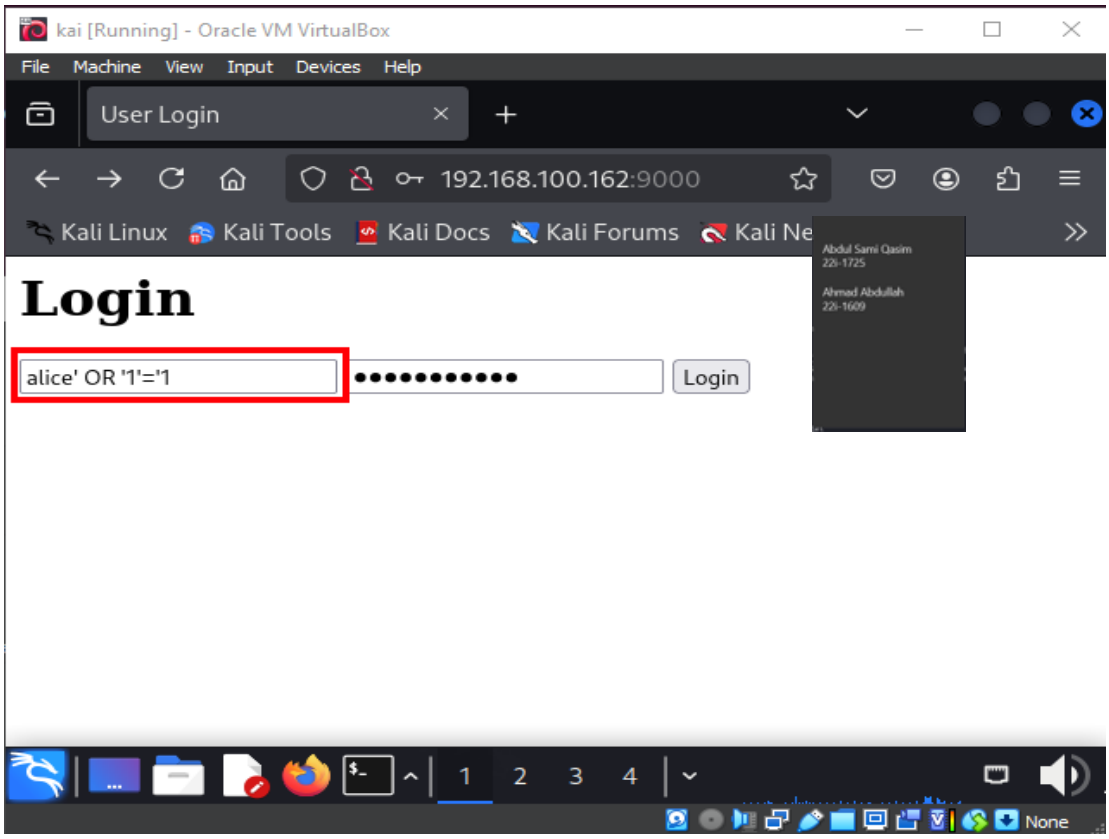
For the DOS attack simulation, we made a rule that generated an alert when more than 5 packets arrived within a minute, yes the bar is set low but it helped us in testing and simulating it.

2. SQLi

We checked the original sqli.rules file to get the idea of what alerts for SQL Injections look like and found out that the rules were only based on uri and not on the content itself. URI-based rules are good, but many SQLi are also done on Login forms and these types of injections are detected only when we make rules that detect content-based injections.

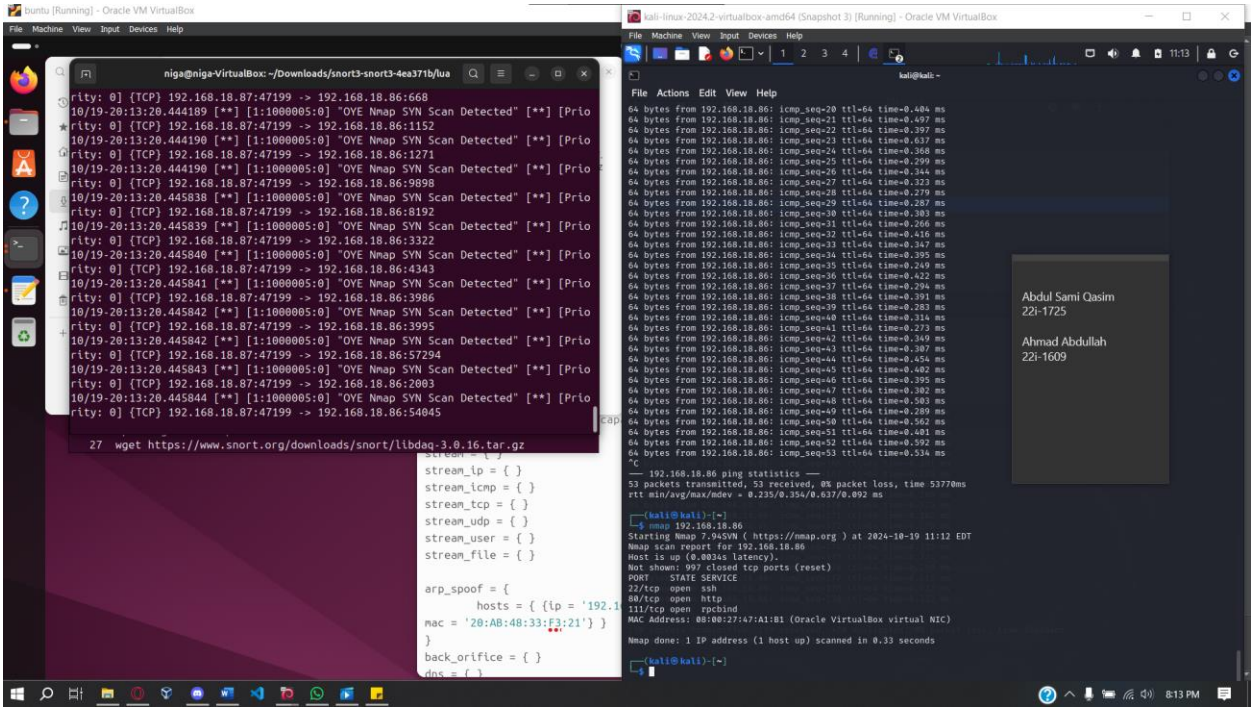


1 Snort SQLi ALERT



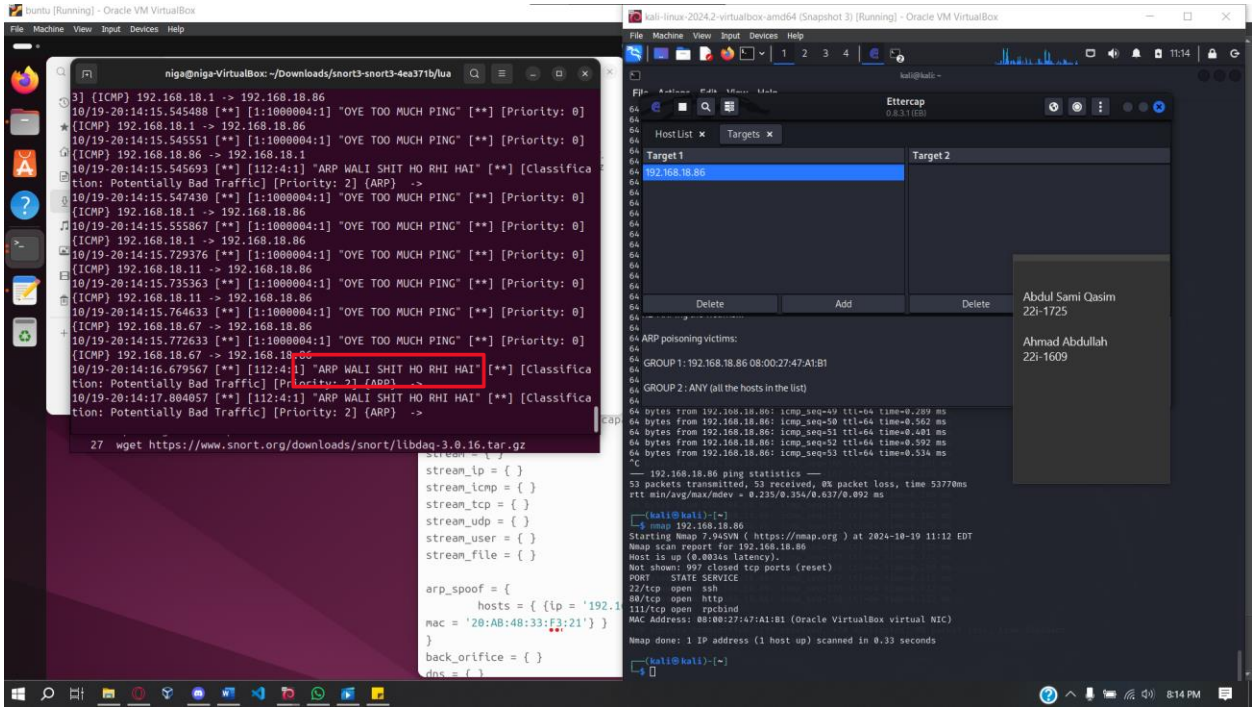
2 Attacking Machin POV

3. Port Scanning



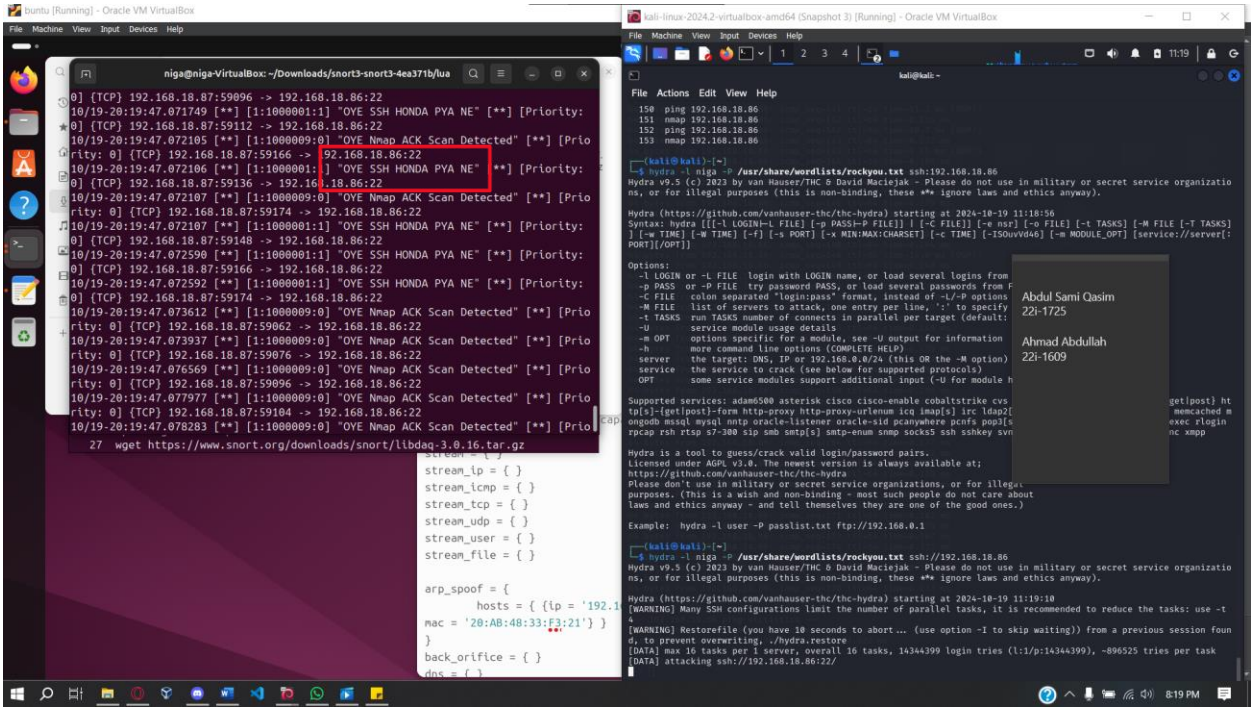
In port scanning we made 5 rules to detect nmap syn, ack, null, fin and xmas flags.

4. Man in the middle



For this we made two rules, one working on the arp packets (apparently snort3 can detect arp packets) and the second rule works on gid:112 which represents the arpspoof preprocessor (the highlighted alert is being generated by the preprocessor rule). On Kali, I'm using the Ettercap GUI to simulate the attack, this is done by first scanning the hosts in the network, flagging the host (ubuntu VM) as the target and then selecting ARP poisoning from the menu.

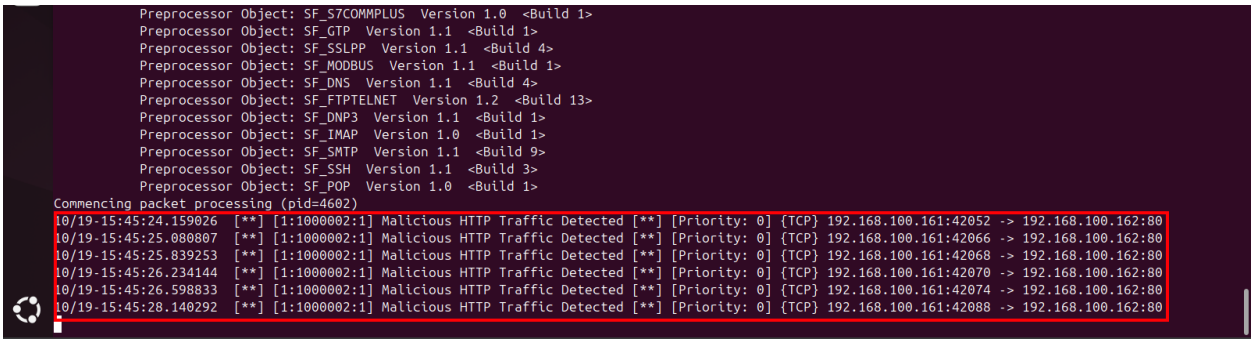
5. Brute Force



For this, we made a rule to detect packets containing SSH in them and if more then 5 packets that fit this category arrive from one source in under 60 seconds, an alert is generated that a brute-force attempt is being made. The highlighted alert is generated as such and the attack was simulated by using hydra.

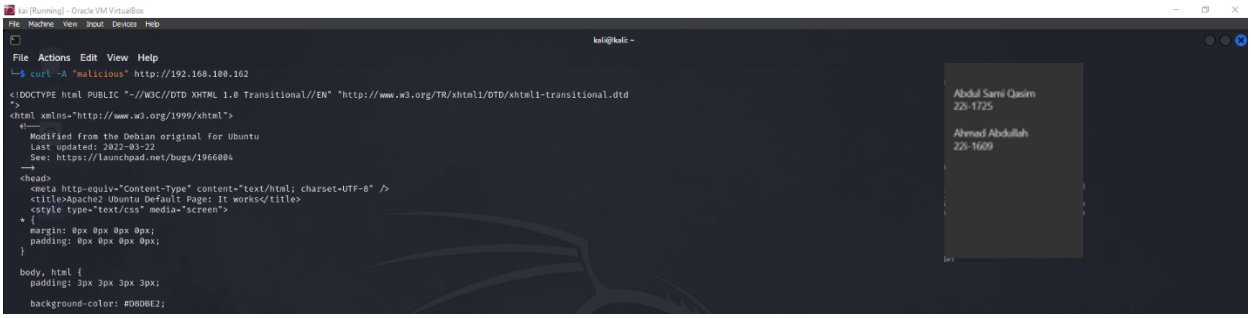
6. Crafted Malicious Traffic

We simulated a web-based malicious traffic. The alert contained an exact match of what request we made but it can be further improved based on requirement. We can say it is a base rule for more sophisticated rule that will detect different kinds of malicious traffic.



3 SNORT Malicious Traffic Alerts

We ran a command “curl -A “Malicious” http://192.168.100.162” from our attacking machine and SNORT picked the packet and alerted.

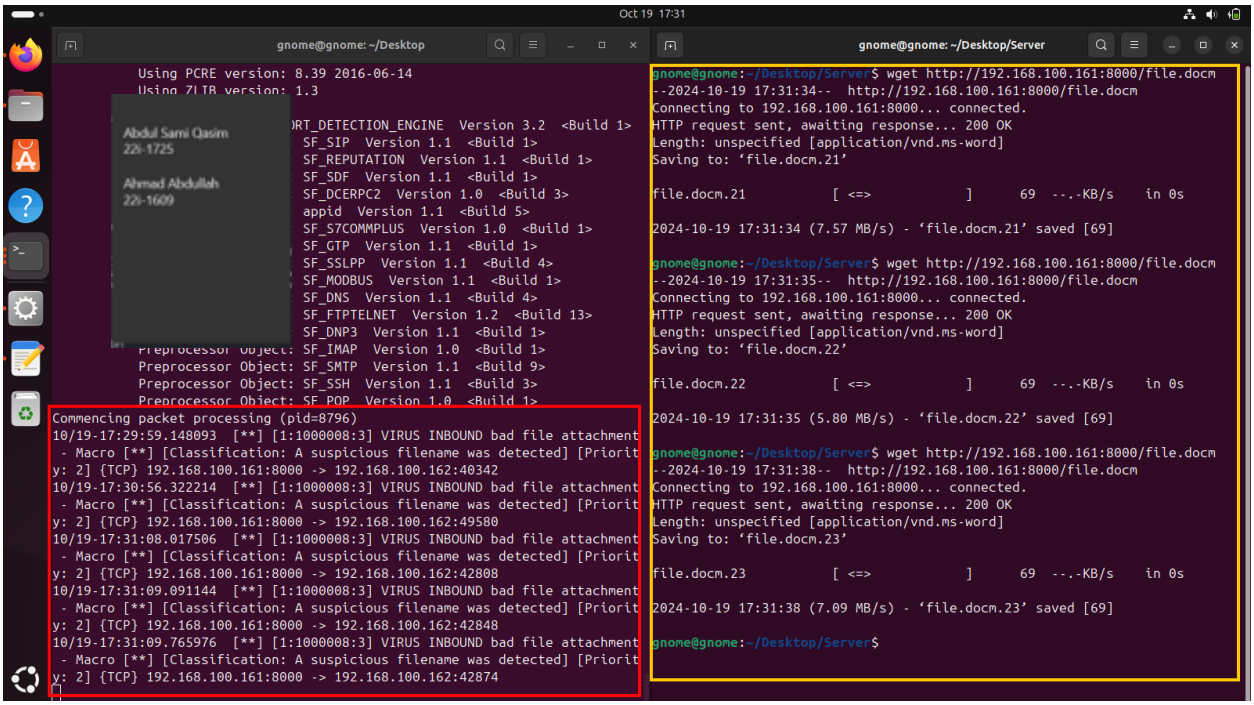


4 Attacking Machine POV

7. Research-Based

For this, we looked at many rule files to find a *virus.rules* which is a rule file that detects suspicious files with ‘exe, bat, etc’. This rule file was missing modern MS Office extensions with macros. So we added those extensions to our custom rules.

On the left, we downloaded a file with the .docm extension, and on the right, SNORT alerted us to a suspicious file download.



Hosted a server on attacking machine with the malicious file so that our Ubuntu should download it .