**NAME :** Mansi Dwivedi

**BRANCH :** IT

**UID :** 2019140016

**BATCH :** A

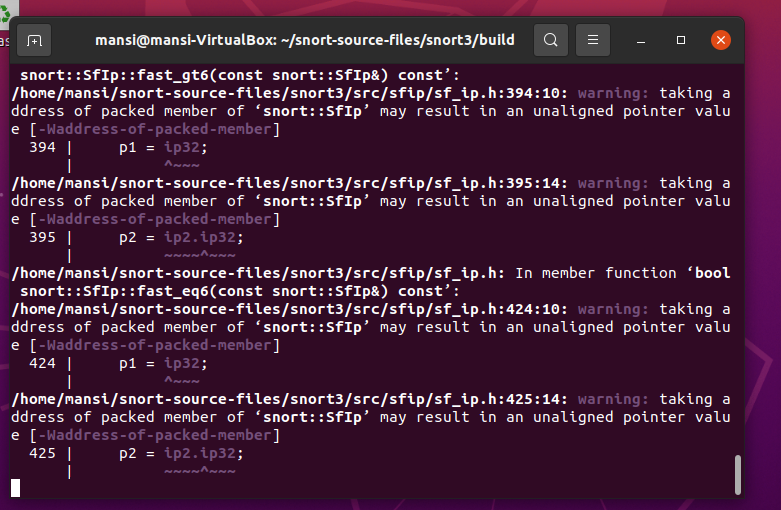
**COURSE :** CSS LAB

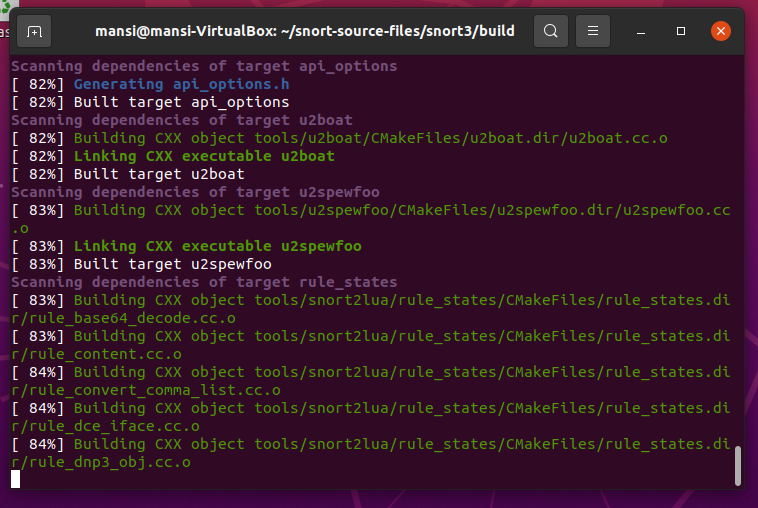
**EXPERIMENT :** 6

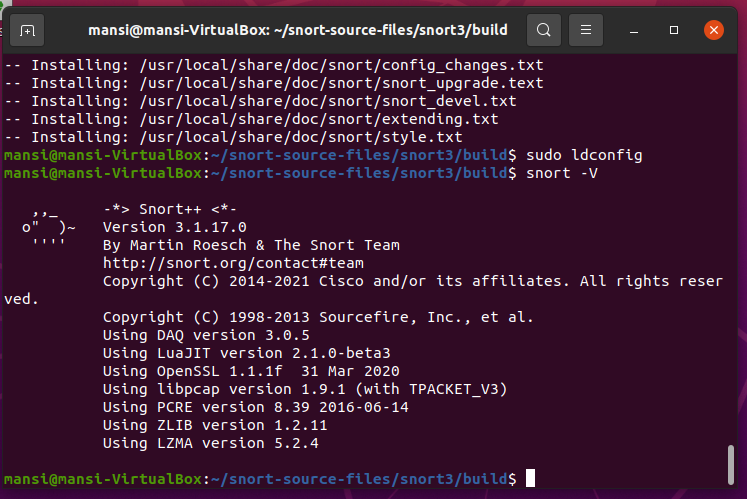
**AIM :** To explore the Snort Intrusion Detection Systems and study Snort IDS, a signature-based intrusion detection system used to detect network attacks. Snort can also be used as a simple packet logger. Use snort as a packet sniffer and write my own IDS rules.

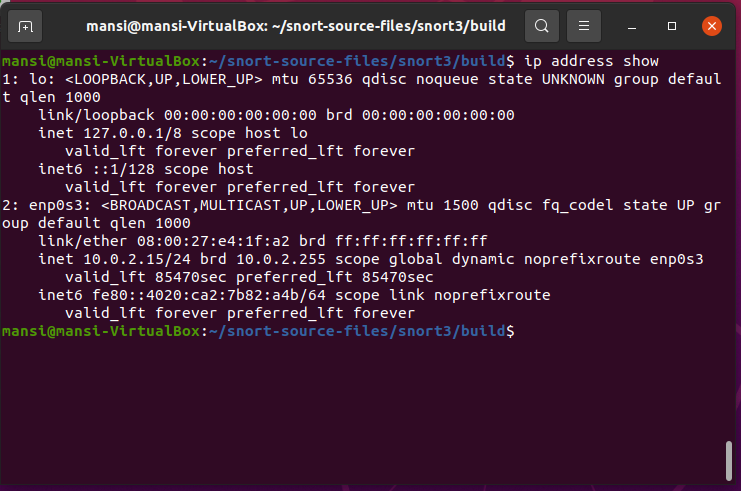
**OUTPUT :**

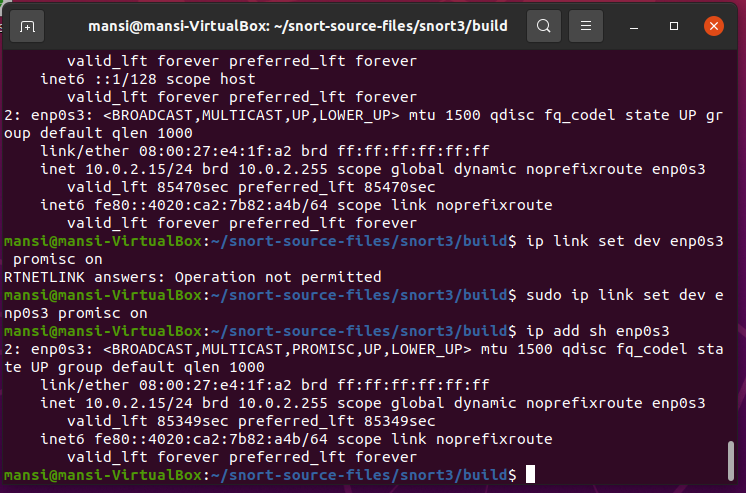
* Installing Snort and various other dependencies it requires.

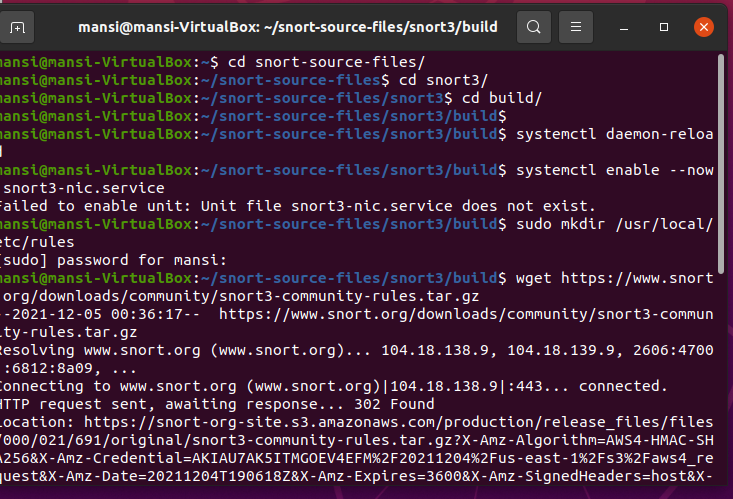
****

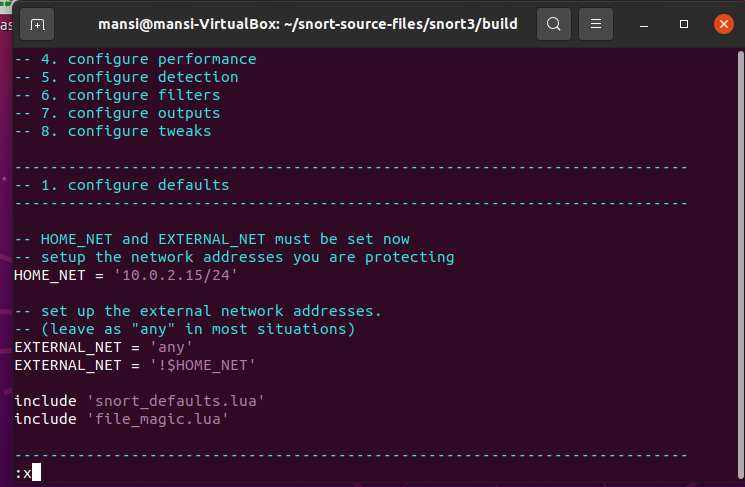
****

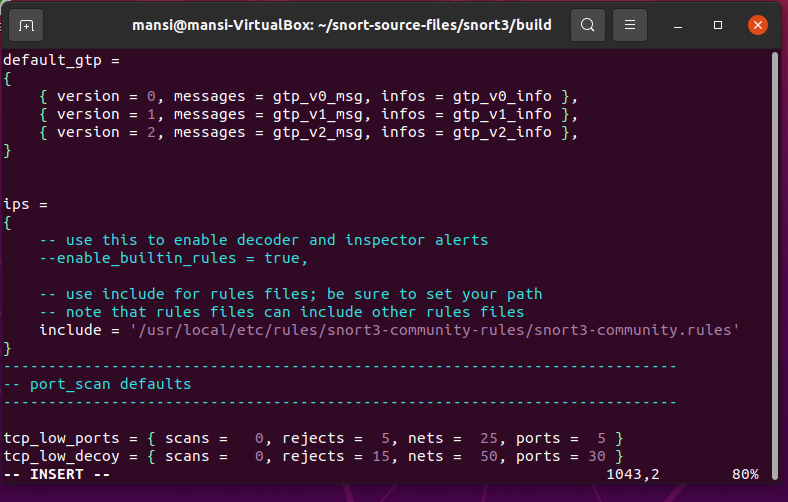
****

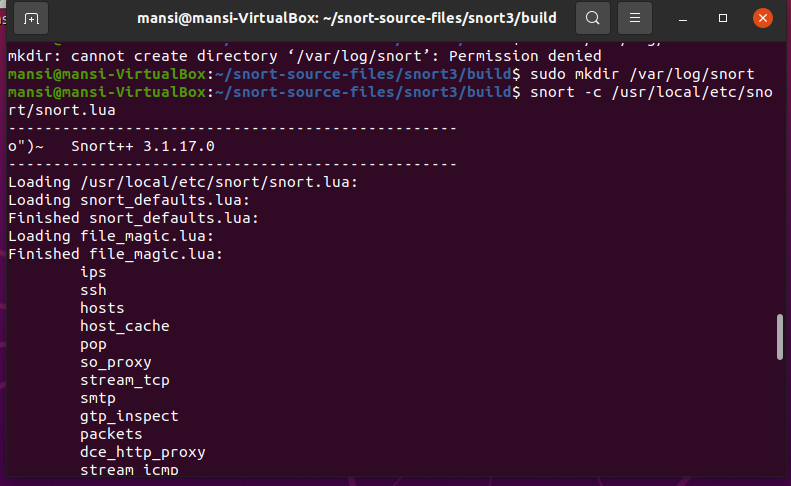
****

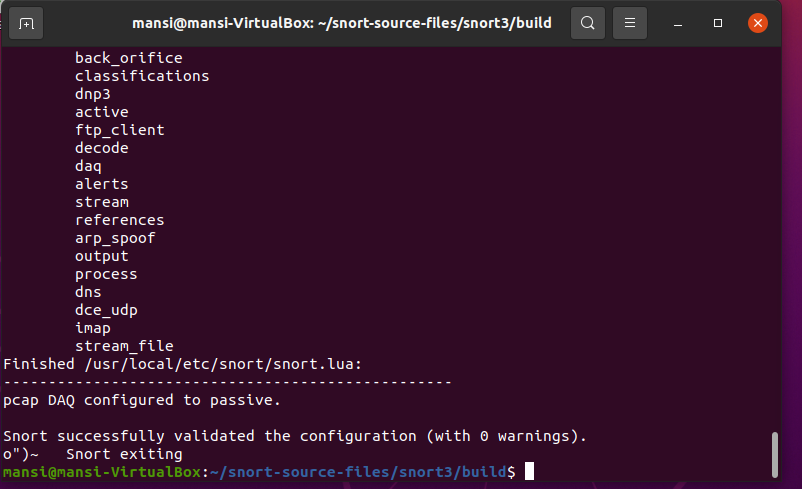
****

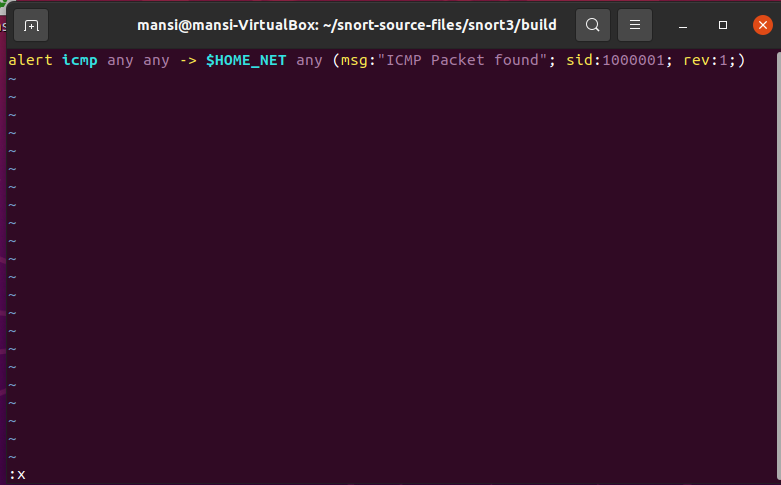
****

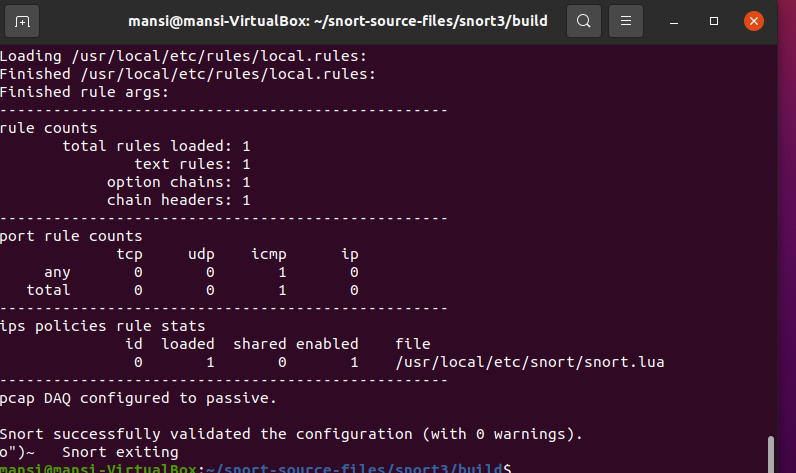
****

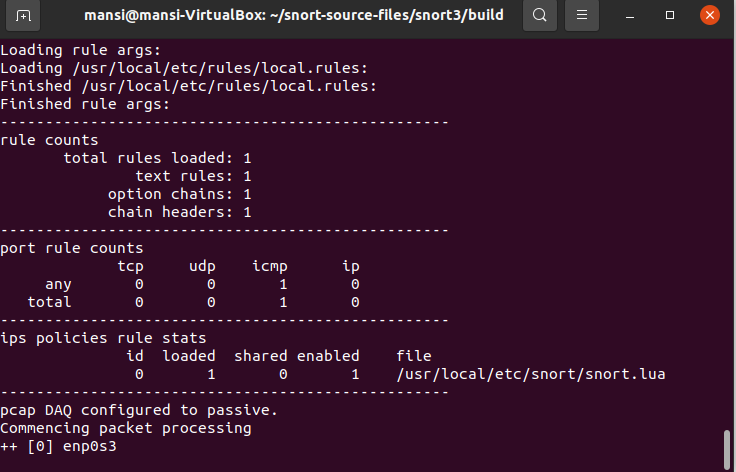
****

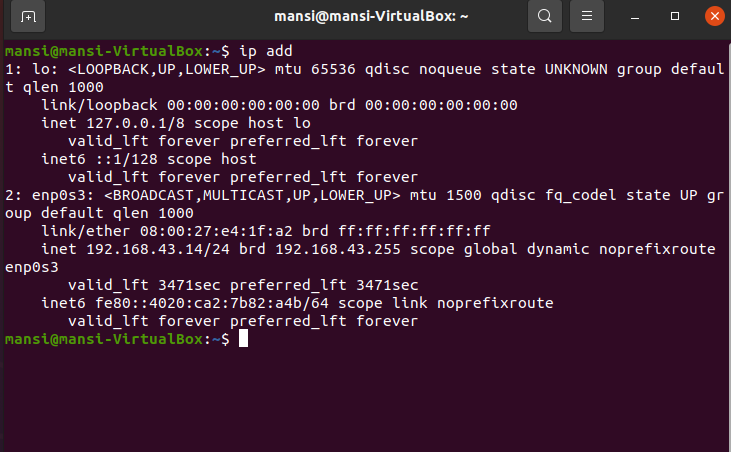
****

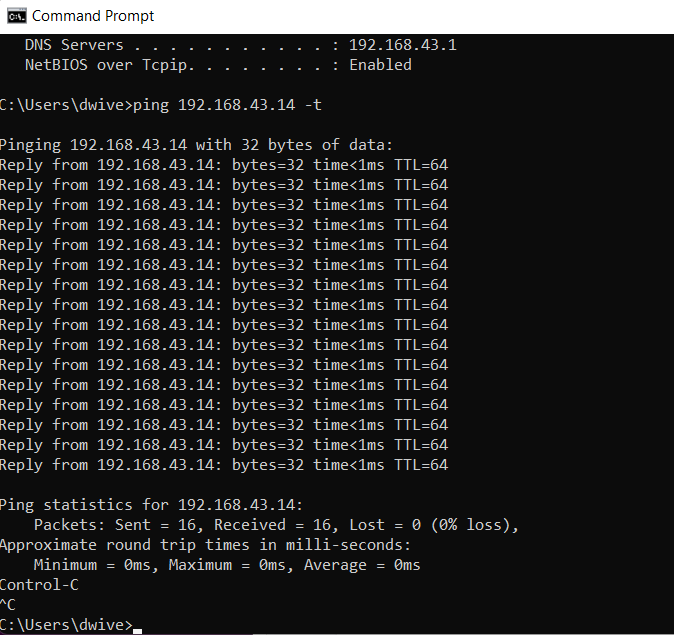
****

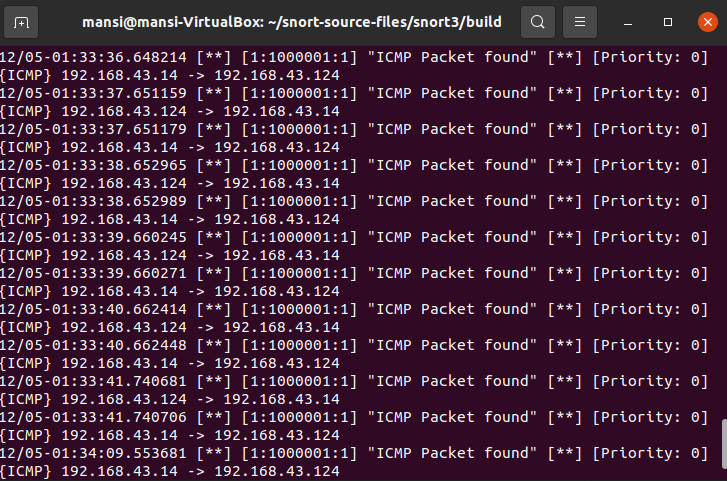
****

****

****

****

****

****

1. **What is a zero-day attack?**

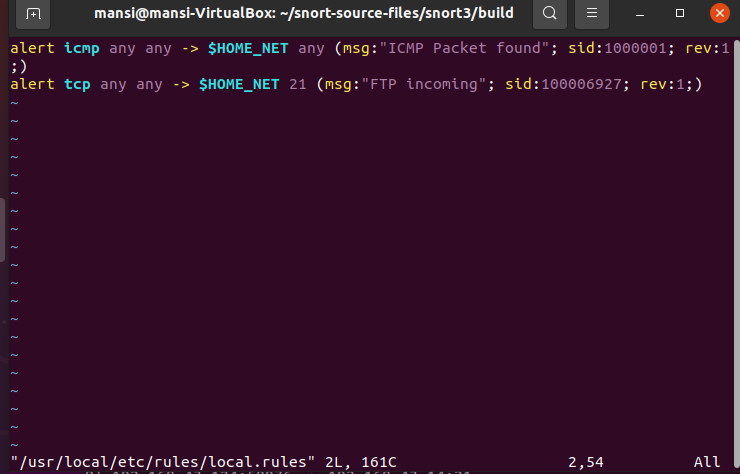
According to what I found on the internet a Zero-day attack can be perceived in two ways: The first one is where these attacks are said to be attacks that target the back door or vulnerabilities of any software that has been patched or declared publicly, on the other hand, the second one states that these attacks take advantage of a vulnerability or bug in the software on the day it was released itself hence the name Zero-day.

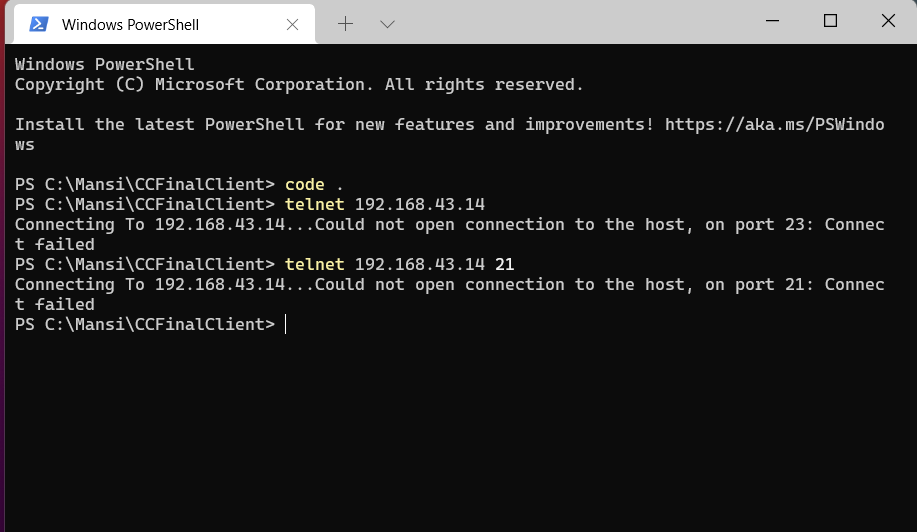
But the general definition describes zero-day attacks (or zero-day exploits) as attacks that target publicly known but still unpatched vulnerabilities.

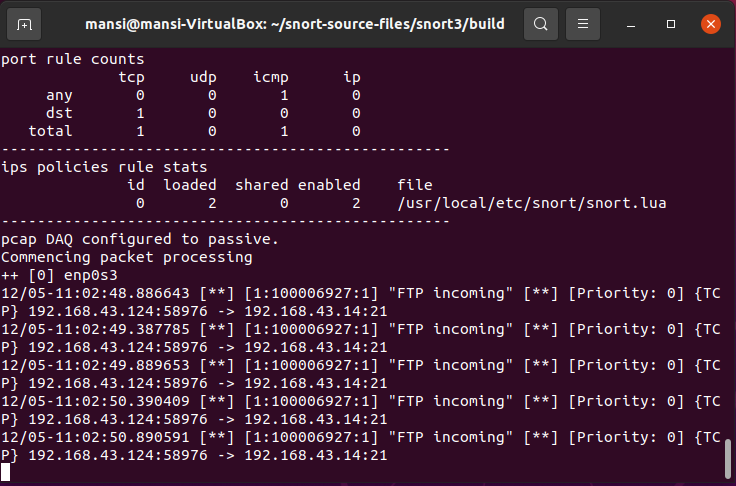
1. **Can Snort catch zero-day network attacks? If not, why not? If yes, how?**

For answering this question after researching I found this research paper “Signature-Based Intrusion Detection for Zero-Day Attacks: (Not) A Closed Chapter?” which states that Snort clearly is able to detect zero-days’ (a mean of 17% detection). The detection rate is however on overall greater for theoretically known attacks (a mean of 54% detection). As a basis for analysis, the detected attacks are categorized according to the vulnerabilities they exploit. (Buffer Error, command injection, PHP file inclusion, etc) A theoretical estimate of Snort’s potency at detecting attacks can be gained by observing the vulnerability coverage of its ruleset. There were 9128 signatures in the tested Snort ruleset. At the time of the release of this ruleset, there were 21166 vulnerabilities disclosed in the US National Vulnerability Database (NVD). Of these vulnerabilities, 9104 were of high severity as defined by the Common Vulnerability Scoring System [20]. A high severity vulnerability can loosely be seen as a vulnerability that can be remotely exploited to gain privileges of a host (e.g., user or admin). Metasploit exploits typically cohere to such vulnerabilities. These are also a focus area of the Snort ruleset due to their severity. The ratio between Snort signatures, disclosed vulnerabilities, and disclosed vulnerabilities of high severity seems to be rather consistent over time: the number of Snort alarms in the official ruleset is about as large as the number of disclosed high vulnerabilities, and a bit less than half of the total number of disclosed vulnerabilities. No formal conclusions regarding vulnerability coverage (and thus detection rate) should however be made from this dataset as multiple Snort rules can address the same vulnerability. Similarly, a single Snort rule sometimes covers multiple vulnerabilities. However, it serves to illustrate that the coverage of the chosen ruleset is neither larger nor smaller than a typical Snort Sourcefire rule set.

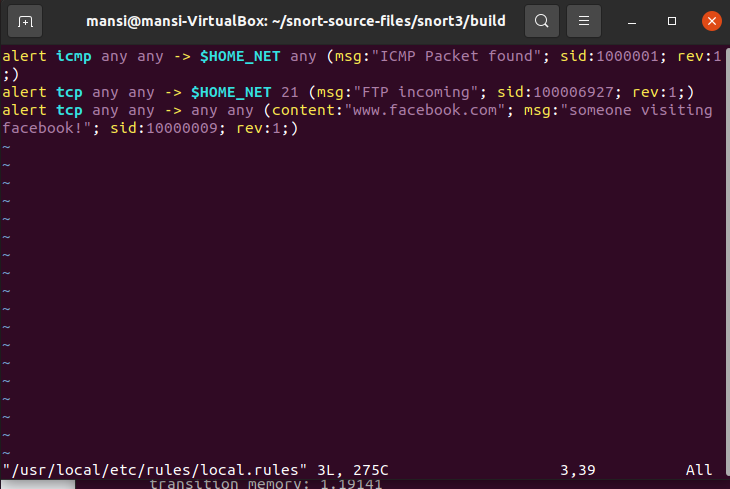
1. **Given a network that has 1 million connections daily were 0.1% (not 10%) attacks. If the IDS has a true positive rate of 95%, and the probability that an alarm is an attack is 95%. What is the false alarm rate?**
2. **Write and add another snort rule and show me you trigger it**

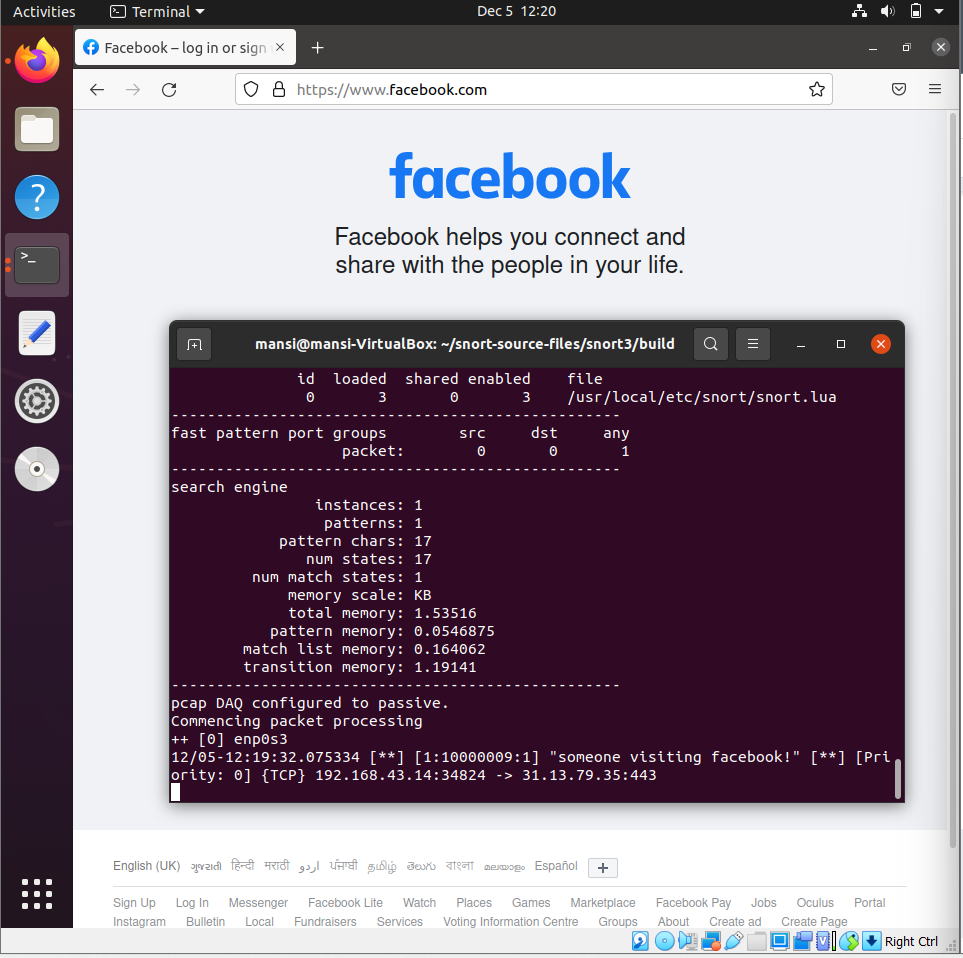
****

****

****

**EXTRA**

****

****