Task 1

SNORT is an open-source intrusion prevention system which utilizes a pattern-based approach for detecting malicious network activity in real-time. It has three main functions, packet sniffing, packet logging, and a full-blown network intrusion detection system. SNORT uses a rule-based language that combines anomaly, protocol, and signature inspection methods to detect potentially malicious activity. Using these tools, SNORT can detect denial-of-service attacks, distributed denial of service attacks, common gateway interface attacks, buffer overflows, stealth port scans, routing attacks, spoofing attacks, server message block probes, and efforts to get an operating system’s fingerprint.

SNORT uses a rule language to determine which network packets should be collected and what actions should be taken when malicious packets are detected. This language can produce three types of rules. The first are alert rules, which simply notify a specific set of users when a certain type of activity is discovered. The second are logging rules, which logs each individual alert once it has been generated. And finally, there are pass rules. Which allows the system to ignore or drop malicious packets.

SNORT is built on the packet capture library (libpcap), which is a widely used tool for sniffing TCP/IP traffic, content searchers, and analyzers. The benefits of Snort include high accuracy, which the open-source nature of the project has greatly assisted in. It is also highly adaptable, once again with the open-source nature of the project allowing contributors to create new features as they are needed. And finally, Snort has real-time protection measures which allow a quick response to potential threats. Snort works by monitoring network traffic in real-time and analyses it using the Misuse Detection Engine BASE. It also utilizes both anomaly-based and signature-based detection methods.

Task 2

Complete!

Task 3

Pretty much what was in the directions. Created a new rule on the SUBSCRN interface. Utilized any for the external nets, http ports, and home net (should have been the Prod-Joomla IP address, 172.16.10.100, but professor comments on discord suggested leaving it at any). Finally added the rules (msg:”LetMeCry Exploit”;sid:1).

Final result: alert tcp any any -> any 80 (msg:”LetMeCry Exploit”;sid:1)

Finally, I activated the rule on the main interface panel.

Task 4

The rule generates an alert anytime a TCP get request is sent to Prod-Joomla on port 80.