CSCI 530 - Lab-7
Rishit Saiya (rsaiya@usc.edu)

1. The smallest value for ping's size that triggers an alert is 801.

2. In `/var/log/snort` directory, I find one of the file named '`alert`' and several files whose names starting with `snort.log`. Mostly in general, there are two output mechanisms in Snort that are: Logging and Alerting.

The whole information of the packet is logged in the files pertaining to `snort.log`. If the log is assumed as an action in the ruleset, details of the packet will be logged in `snort.log` files. It can include details such as IP Address Protocol, size of packet, source IP Address, destination IP Address, etc. These files play a vital role in terms of debugging any crash, any incident where the authorities want to co-relate these log files through different solutions.

The process of Alerting is contingent on actions defined in rules. If an action mentioned in rule is "alert", then an alert will be generated and outputted in the alert file. In the alert file we can see the details of the alert that includes but is not limited to, what attack (or attempted attack) IDS has detected based on rule, details about the attack, CVE information. Its purpose is to alert an administrator or concerned stakeholders about the particular event (attack, attempted attacks). Furthermore, it also provides the IP address of the source and target of the snort attack. In a nutshell, if an alert is triggered, the event is written to the alert facility and related detailed information is written to the output facility.

3. Xref is a reference which points to Mitre website. Mitre consists of a database of predominantly known vulnerabilities. Each vulnerability is represented by Common Vulnerabilities and Exposures (CVE). Each CVE is assigned with a unique reference number and it consists of complete details of vulnerabilities. The given CVE-2002-0606 is about the Buffer Overflow vulnerability that was found in 3Cdaemon 2.0 FTP server. This vulnerability was essentially allowing attackers to cause a Denial of Service (crash) and possibly execute arbitrary code via verbose commands such as login.

4.

```
alert tcp $HOME_NET 23 -> $EXTERNAL_NET any (msg:"TELNET login incorrect"; content:"Login incorrect";)
```

Action: `alert`

Protocol: `tcp`

Source: `$Home_NET` (It specifies the protected network and includes all the IP address that fall in this category)

Source port: `23` (Telnet port)

Destination: `$EXTERNAL_NET` (It specifies the outside network. It includes all the IP address that are part of outside/exterior network)

Destination Port: `any` (Open to availability, not connecting to a specific port)

Message embedded inside the alert: `(msg:"TELNET login incorrect"; content:"Login incorrect";)`
Content embedded inside the alert: This is packet payload content to look for. (alert will be generated if packet contains ‘Login incorrect’)

This alert is for checking incorrect logins. This rule will generate an alert if it finds incorrect login attempts on the telnet port. Here source IP address is all the IP addresses that are part of protected/internal network and port is telnet port. Destination IP address is outside the network and the port can be anything.

Content: "Login incorrect"
If the packet contains ‘Login incorrect’, this alert will be generated. It means wrong attempts are being tried. So IDS will check the packet based on the rule. In this case the packet contains an incorrect login, therefore an alert is generated.

msg: "TELNET login incorrect”
Things written in a message “Telnet Login incorrect” will be logged into the alert file. It is based on the output of rule structure and action. Since an incorrect packet is found, this message will be logged to the alert file.