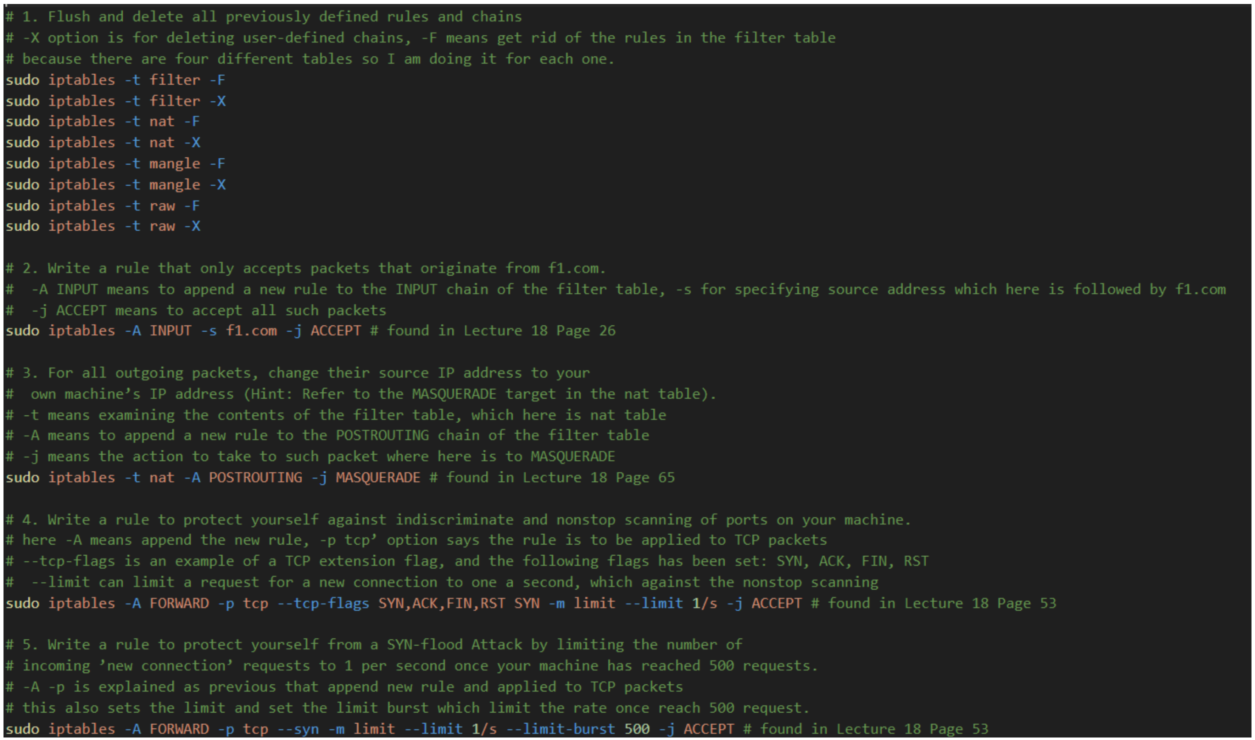
Explanation:



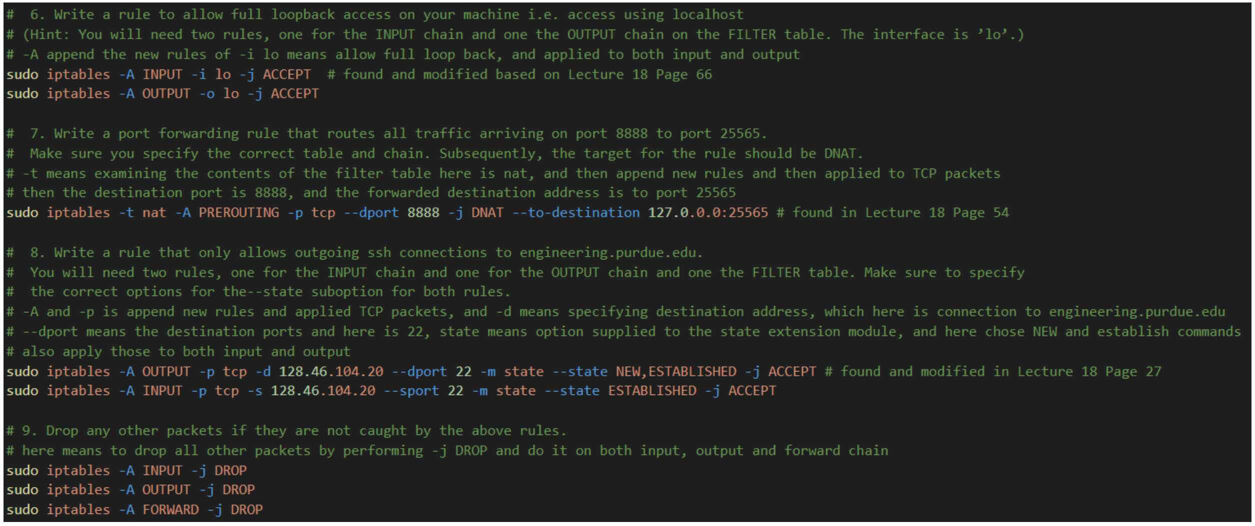
```
# 1. Flush and delete all previously defined rules and chains
# -X option is for deleting user-defined chains, -F means get rid of the rules in the filter table
# because there are four different tables so I am doing it for each one.
sudo iptables -t filter -F
sudo iptables -t filter -X
sudo iptables -t nat -F
sudo iptables -t nat -X
sudo iptables -t mangle -F
sudo iptables -t mangle -X
sudo iptables -t raw -F
sudo iptables -t raw -X

# 2. Write a rule that only accepts packets that originate from f1.com.
#  -A INPUT means to append a new rule to the INPUT chain of the filter table, -s for specifying source address which here is followed by f1.com
#  -j ACCEPT means to accept all such packets
sudo iptables -A INPUT -s f1.com -j ACCEPT # found in Lecture 18 Page 26

# 3. For all outgoing packets, change their source IP address to your
#  own machine’s IP address (Hint: Refer to the MASQUERADE target in the nat table).
# -t means examining the contents of the filter table, which here is nat table
# -A means to append a new rule to the POSTROUTING chain of the filter table
# -j means the action to take to such packet where here is to MASQUERADE
sudo iptables -t nat -A POSTROUTING -j MASQUERADE # found in Lecture 18 Page 65

# 4. Write a rule to protect yourself against indiscriminate and nonstop scanning of ports on your machine.
# here -A means append the new rule, -p tcp’ option says the rule is to be applied to TCP packets
# --tcp-flags is an example of a TCP extension flag, and the following flags has been set: SYN, ACK, FIN, RST
#  --limit can limit a request for a new connection to one a second, which against the nonstop scanning
sudo iptables -A FORWARD -p tcp --tcp-flags SYN,ACK,FIN,RST SYN -m limit --limit 1/s -j ACCEPT # found in Lecture 18 Page 53

# 5. Write a rule to protect yourself from a SYN-flood Attack by limiting the number of
# incoming ’new connection’ requests to 1 per second once your machine has reached 500 requests.
# -A -p is explained as previous that append new rule and applied to TCP packets
# this also sets the limit and set the limit burst which limit the rate once reach 500 request.
sudo iptables -A FORWARD -p tcp --syn -m limit --limit 1/s --limit-burst 500 -j ACCEPT # found in Lecture 18 Page 53
```



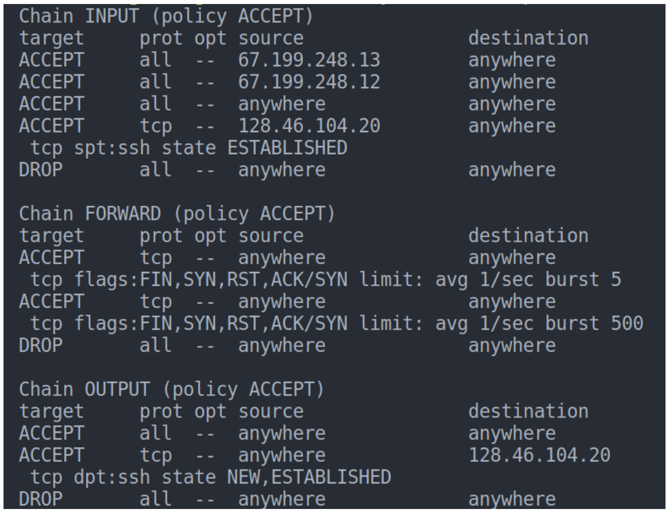
```
#  6. Write a rule to allow full loopback access on your machine i.e. access using localhost
# (Hint: You will need two rules, one for the INPUT chain and one the OUTPUT chain on the FILTER table. The interface is ’lo’.)
# -A append the new rules of -i lo means allow full loop back, and applied to both input and output
sudo iptables -A INPUT -i lo -j ACCEPT  # found and modified based on Lecture 18 Page 66
sudo iptables -A OUTPUT -o lo -j ACCEPT

#  7. Write a port forwarding rule that routes all traffic arriving on port 8888 to port 25565.
#  Make sure you specify the correct table and chain. Subsequently, the target for the rule should be DNAT.
# -t means examining the contents of the filter table here is nat, and then append new rules and then applied to TCP packets
# then the destination port is 8888, and the forwarded destination address is to port 25565
sudo iptables -t nat -A PREROUTING -p tcp --dport 8888 -j DNAT --to-destination 127.0.0.0:25565 # found in Lecture 18 Page 54

#  8. Write a rule that only allows outgoing ssh connections to engineering.purdue.edu.
#  You will need two rules, one for the INPUT chain and one for the OUTPUT chain and one the FILTER table. Make sure to specify
#  the correct options for the--state suboption for both rules.
# -A and -p is append new rules and applied TCP packets, and -d means specifying destination address, which here is connection to engineering.purdue.edu
# --dport means the destination ports and here is 22, state means option supplied to the state extension module, and here chose NEW and establish commands
# also apply those to both input and output
sudo iptables -A OUTPUT -p tcp -d 128.46.104.20 --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT # found and modified in Lecture 18 Page 27
sudo iptables -A INPUT -p tcp -s 128.46.104.20 --sport 22 -m state --state ESTABLISHED -j ACCEPT

# 9. Drop any other packets if they are not caught by the above rules.
# here means to drop all other packets by performing -j DROP and do it on both input, output and forward chain
sudo iptables -A INPUT -j DROP
sudo iptables -A OUTPUT -j DROP
sudo iptables -A FORWARD -j DROP
```

Output:



```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  67.199.248.13        anywhere
ACCEPT     all  --  67.199.248.12        anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  128.46.104.20        anywhere
 tcp spt:ssh state ESTABLISHED
DROP       all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere
 tcp flags:FIN,SYN,RST,ACK/SYN limit: avg 1/sec burst 5
ACCEPT     tcp  --  anywhere             anywhere
 tcp flags:FIN,SYN,RST,ACK/SYN limit: avg 1/sec burst 500
DROP       all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             128.46.104.20
 tcp dpt:ssh state NEW,ESTABLISHED
DROP       all  --  anywhere             anywhere
```