

SQUID

Ajay Aji-22UBCA7312

Experiment: 4

Aim: To create and configure Squid -proxy server

Description: SQUID – PROXY SERVER

Squid is a full-featured web proxy cache server application which provides proxy and cache services for HyperText Transport Protocol (HTTP), File Transfer Protocol (FTP), and other popular network protocols. Squid can implement caching and proxying of Secure Sockets Layer (SSL) requests and caching of Domain Name Server (DNS) lookups, and perform transparent caching. Squid also supports a wide variety of caching protocols, such as Internet Cache Protocol (ICP), the HyperText Caching Protocol (HTCP), the Cache Array Routing Protocol (CARP), and the Web Cache Coordination Protocol (WCCP). The Squid proxy cache server is an excellent solution to various proxy and caching server needs, and scales from the branch office to enterprise-level networks while providing extensive, granular access control mechanisms, and monitoring of critical parameters via the Simple Network Management Protocol (SNMP). When selecting a computer system for use as a dedicated Squid caching proxy server for many users ensure it is configured with a large amount of physical memory as Squid maintains an in-memory cache for increased performance.

Port No: 3128

Package name: squid

Configuration file: /etc/squid/squid.conf

Procedure:

1. At a terminal prompt, enter the following command to install the Squid server:

```
$sudo apt install squid
```

2. Squid is configured by editing the directives contained within the /etc/squid/squid.conf configuration file.

3. Change the access as shown below: `acl localnet src 192.168.234.139(your ip address) acl blocksite dstdomain "/etc/squid/blocksite" http_access deny blocksite http_access allow localnet #http_access deny all http_access allow all`

4. To block access to the website we must configure using "/etc/squid/blocksite" we edit the file by running: `$cd /etc/squid $sudo gedit blocksite`

5. Add the websites to block: in this case, I am blocking youtube, facebook, google

6. To check the actual functioning of the proxy server go to the browser and click settings, search proxy in connection settings.

7. To configure Proxy access to the internet

8. Select Manual Proxy configuration
9. Type your HTTP Proxy(IP Address) and Port number as 3128.
10. Select SOCKS v5 CONNECTING TO WEBSITE
11. Search for the blocked websites 12. Access is denied to the above websites.

Result:

```
root@UBUNTU:~# sudo apt install squid
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libdbi-perl libecap3 squid-common squid-langpack
Suggested packages:
  libnids-perl libnet-daemon-perl libsql-statement-perl squidclient squid-cgi squid-purge resolvconf snbcclient winbind
The following NEW packages will be installed:
  libdbi-perl libecap3 squid-common squid-langpack
0 upgraded, 4 newly installed, 0 to remove and 114 not upgraded.
Need to get 3,809 kB of archives.
After this operation, 12.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 libecap3 amd64 1.0.1-3.2ubuntu4 [17.0 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 squid-langpack all 20200403-1 [170 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 squid-common all 5.9-0ubuntu0.22.04.2 [704 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy/main amd64 libdbi-perl amd64 1.643-3build3 [741 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 squid amd64 5.9-0ubuntu0.22.04.2 [2,678 kB]
Fetched 3,809 kB in 3s (1,207 kB/s)
Selecting previously unselected package libecap3:amd64.
(Reading database ... 232090 files and directories currently installed.)
Preparing to unpack .../libecap3_1.0.1-3.2ubuntu4_amd64.deb ...
Unpacking libecap3:amd64 (1.0.1-3.2ubuntu4) ...
Selecting previously unselected package squid-langpack.
Preparing to unpack .../squid-langpack_20200403-1_all.deb ...
Unpacking squid-langpack (20200403-1) ...
Selecting previously unselected package squid-common.
Preparing to unpack .../squid-common_5.9-0ubuntu0.22.04.2_all.deb ...
Unpacking squid-common (5.9-0ubuntu0.22.04.2) ...
Selecting previously unselected package libdbi-perl:amd64.
Preparing to unpack .../libdbi-perl_1.643-3build3_amd64.deb ...
Unpacking libdbi-perl:amd64 (1.643-3build3) ...
Selecting previously unselected package squid.
Preparing to unpack .../squid_5.9-0ubuntu0.22.04.2_amd64.deb ...
Unpacking squid (5.9-0ubuntu0.22.04.2) ...
Setting up squid (5.9-0ubuntu0.22.04.2) ...
Setting up squid-langpack (20200403-1) ...
Setting up libdbi-perl:amd64 (1.643-3build3) ...
Setting up libecap3:amd64 (1.0.1-3.2ubuntu4) ...
```

```
GNU nano 6.2 /etc/squid/squid.conf
http_access allow localhost manager
http_access deny manager

# This default configuration only allows localhost requests because a more
# permissive squid installation could introduce new attack vectors into the
# network by proxying external TCP connections to unprotected services.
http_access allow localhost

# The two deny rules below are unnecessary in this default configuration
# because they are followed by a "deny all" rule. However, they may become
# critically important when you start allowing external requests below them.

# Protect web applications running on the same server as Squid. They often
# assume that only local users can access them at "localhost" ports.
http_access deny to_localhost

# Protect cloud servers that provide local users with sensitive info about
# their server via certain well-known link-local (a.k.a. APNs) addresses.
http_access deny to_linklocal

# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
include /etc/squid/conf.d/*.conf

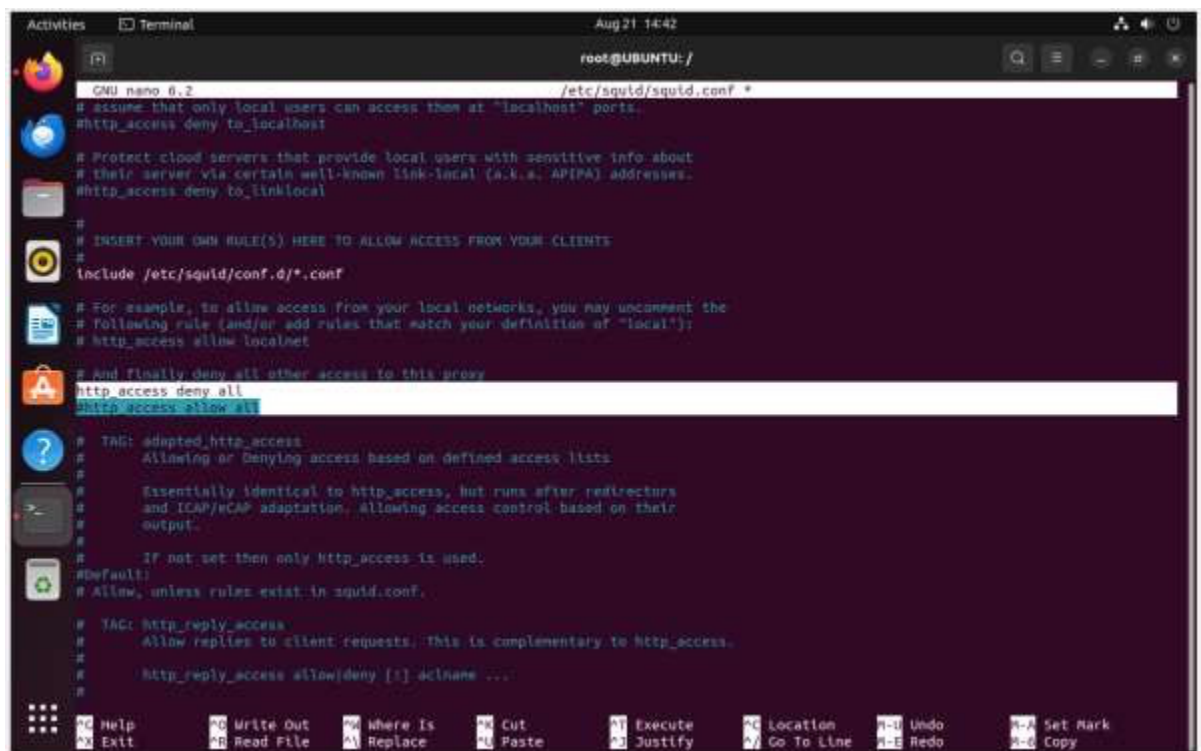
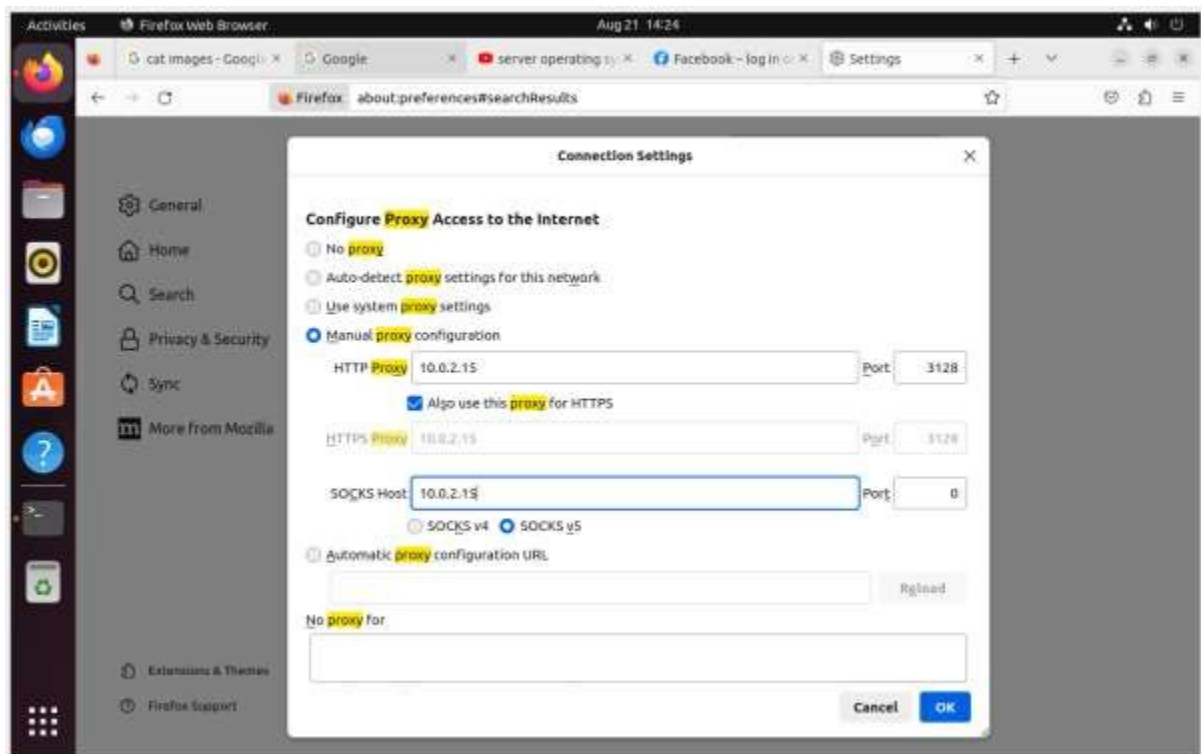
# For example, to allow access from your local networks, you may uncomment the
# following rule (and/or add rules that match your definition of "local"):
# http_access allow localnet

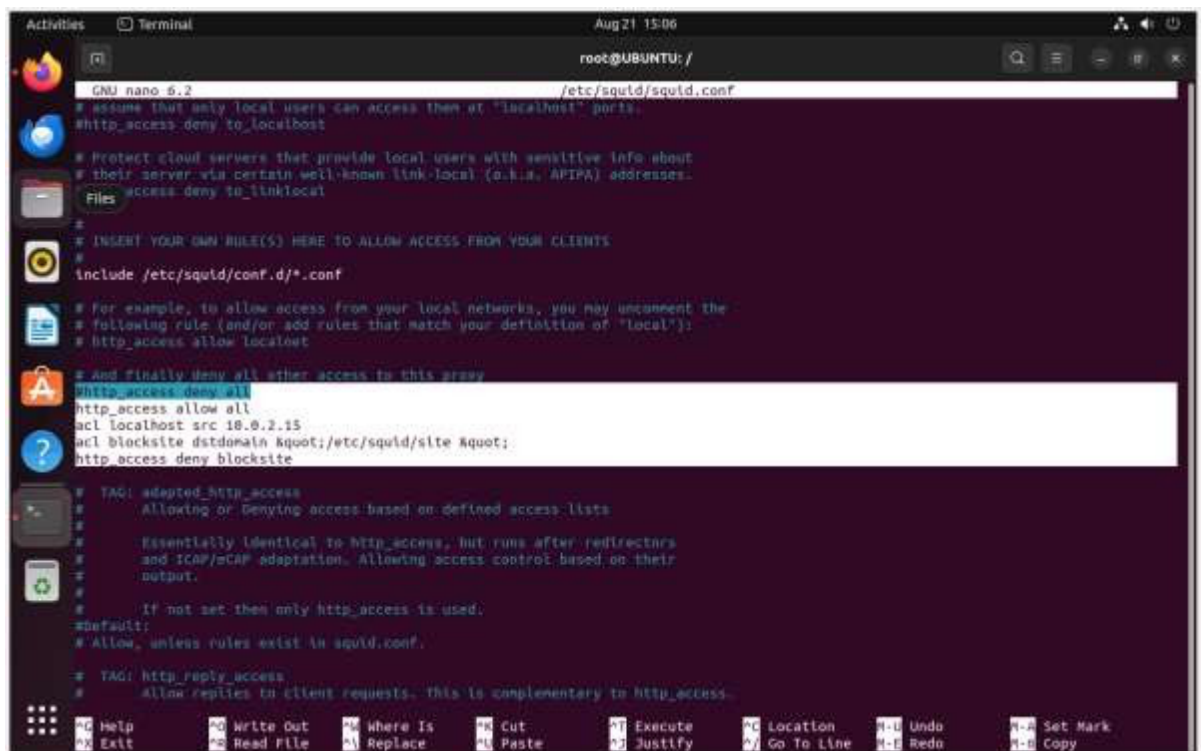
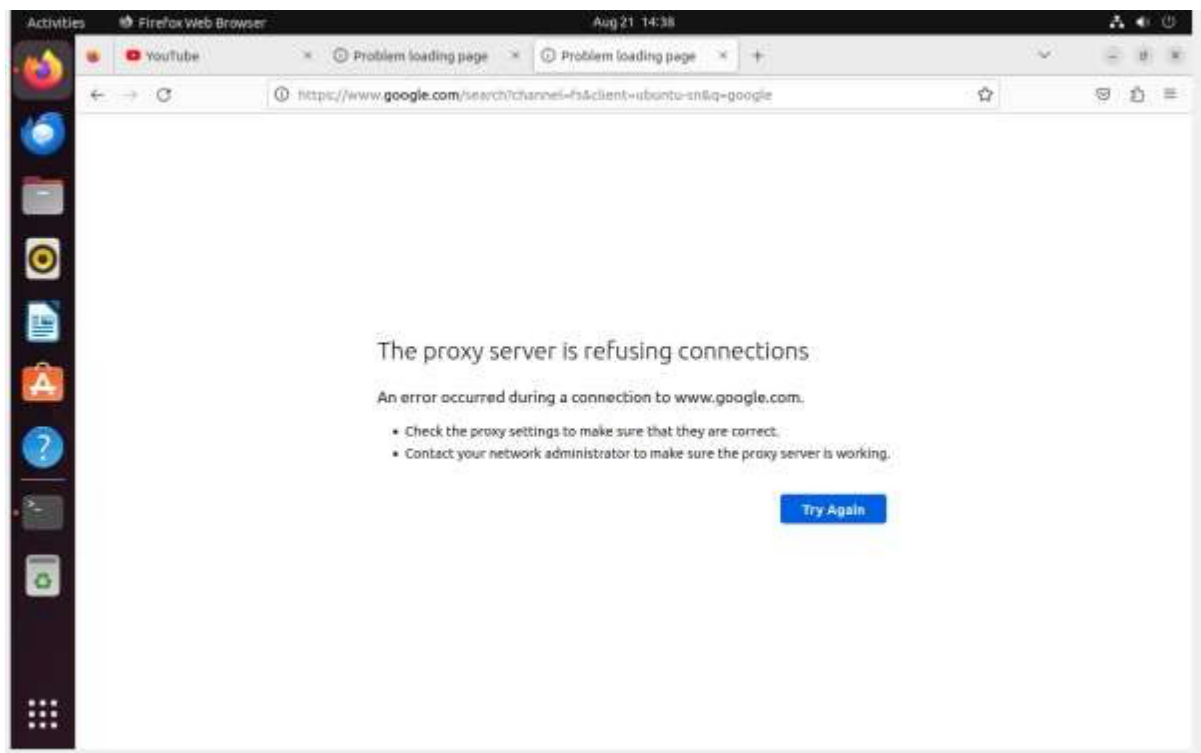
# And finally deny all other access to this proxy
http_access deny all

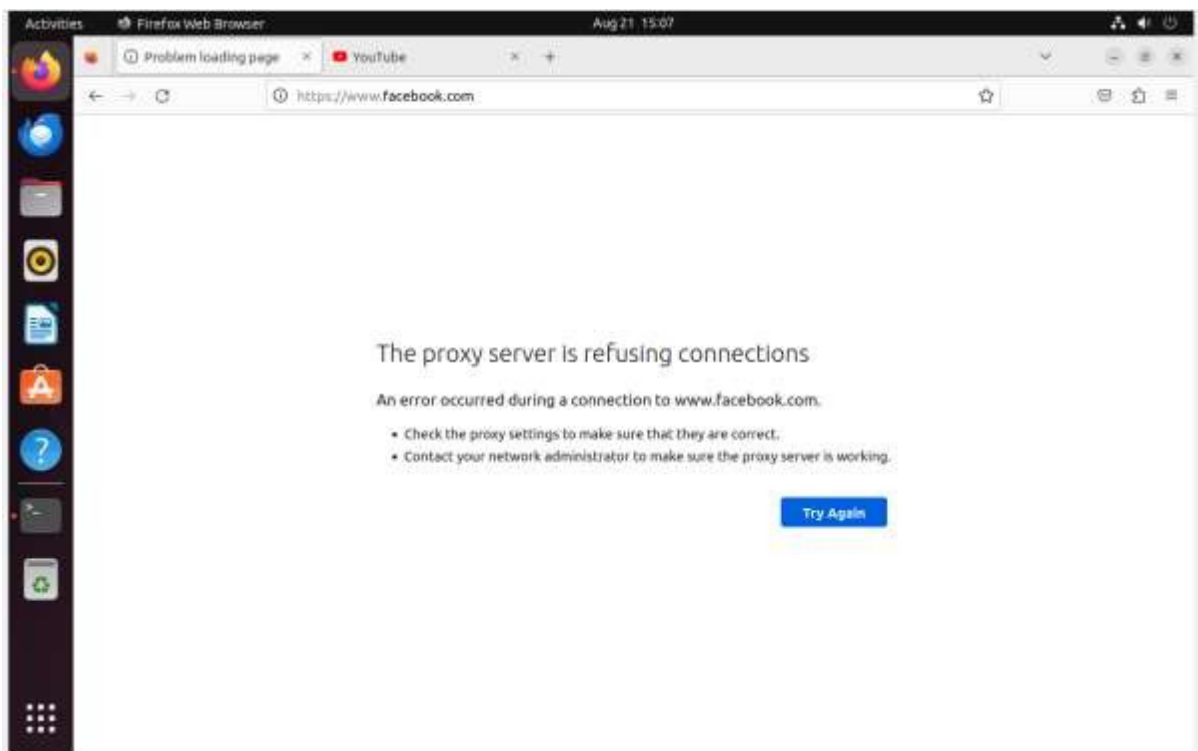
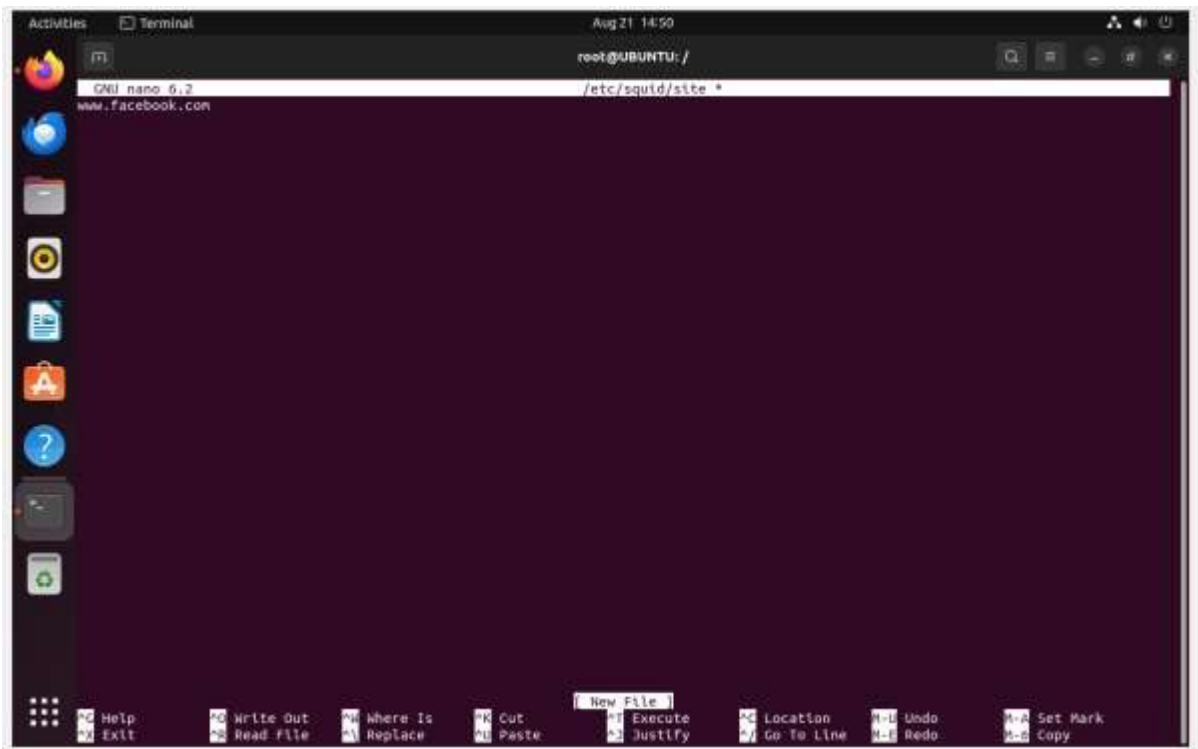
# End of file
http_access allow all

# This is the default configuration for squid. It is a good starting point
# for understanding how squid works. It is not a complete configuration
# file. It is missing many options and defaults. It is not a good idea
# to use it as a template for a new configuration. It is better to use
# the defaults and modify them as needed.

# Essentially identical to http_access, but runs after redirects
```







```
Activities  Terminal  Aug 21 15:08  root@UBUNTU: /

Setting up squid-common (5.9-0ubuntu0.22.04.2) ...
Setting up squid (5.9-0ubuntu0.22.04.2) ...
Setcap worked! /usr/lib/squid/pinger is not setuid!
Skipping profile in /etc/apparmor.d/disable: usr.sbin.squid
Created symlink /etc/systemd/system/multi-user.target.wants/squid.service → /lib/systemd/system/squid.service.
Processing triggers for ufw (0.36-1-4ubuntu0.1) ...
Rules updated for profile 'samba'

Processing triggers for nanodb (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
root@UBUNTU:~# ifconfig
enp8s1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::baf:b255:e5ba:c37c prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:0e:3c:4f txqueuelen 1000 (Ethernet)
    RX packets 75749 bytes 107990004 (107.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 17003 bytes 1813540 (1.8 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 3490 bytes 451639 (451.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3490 bytes 451639 (451.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@UBUNTU:~# nano /etc/squid/squid.conf
root@UBUNTU:~# sudo systemctl restart squid
root@UBUNTU:~# nano /etc/squid/squid.conf
root@UBUNTU:~# sudo systemctl restart squid
root@UBUNTU:~# nano /etc/squid/squid.conf
root@UBUNTU:~# sudo systemctl restart squid
root@UBUNTU:~# nano /etc/squid/site
root@UBUNTU:~# sudo systemctl restart squid
```

Conclusion:

All the commands have been executed and the output has been obtained successfully.