

HW 05 – Linux

By :- Faraz Ahmed

Contents

1. Adding Firewall Rules	3
2. Linux Server Setup.....	4
a. Install and configure the UbuntuWebServer VM	4
b. Install and Configure the RockyDBServer VM	7
3. User and Group Creation.....	11
4. Using Linux Hardening.....	14
5. Updated Topology	22
6. EAS 595 Additional Tasks.....	23

1. Adding Firewall Rules

- To allow AdminNet to access pfSense Router webConfiguration GUI, we need to a new firewall rule in AdminNet as highlighted (or checked) in Figure 1 below. In Source, we have to enter “AdminNet Address” and in destination, we have to enter “Address of pfSense” and “HTTPS” should be the default port.

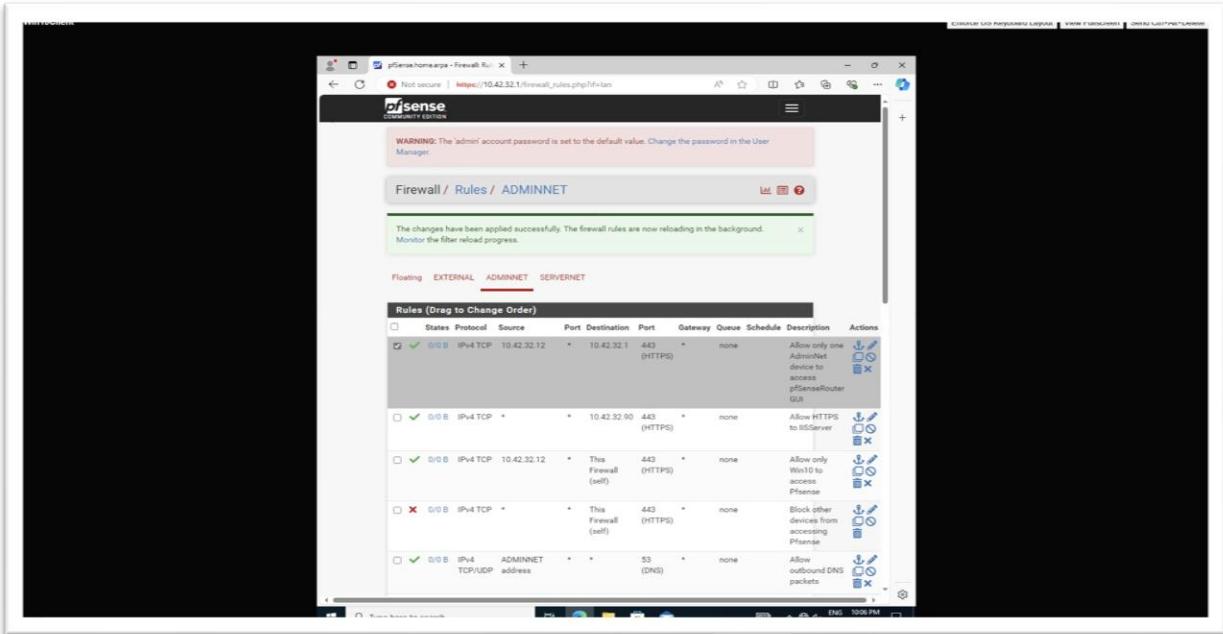


Figure 1: Screenshot of New “Highlighted” Firewall Rules in AdminNet.

- As we can see from figure 2, all new 6 firewall rules from which one is used to block “All Inbound Traffic from External devices to ServerNet” then another one is to allow “AdminNet to access ServerNet” and then rest is to allow “HTTP/HTTPS, DNS and ICMP(Ping) from ServerNet” as shown in highlighted (or checked) Figure 2.

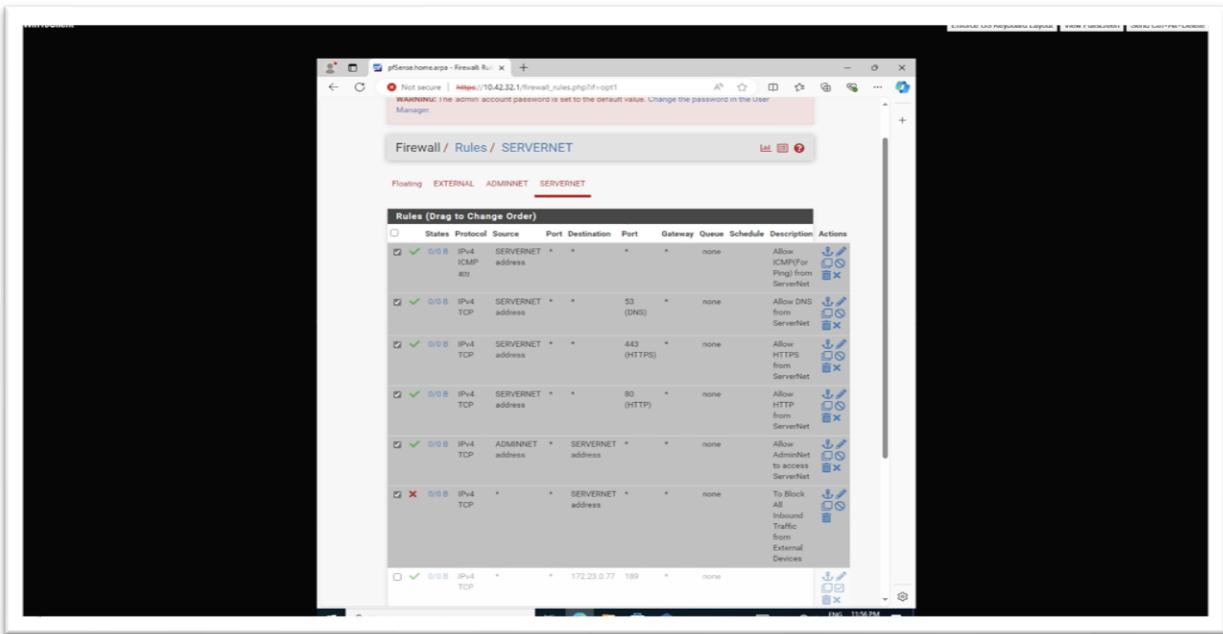
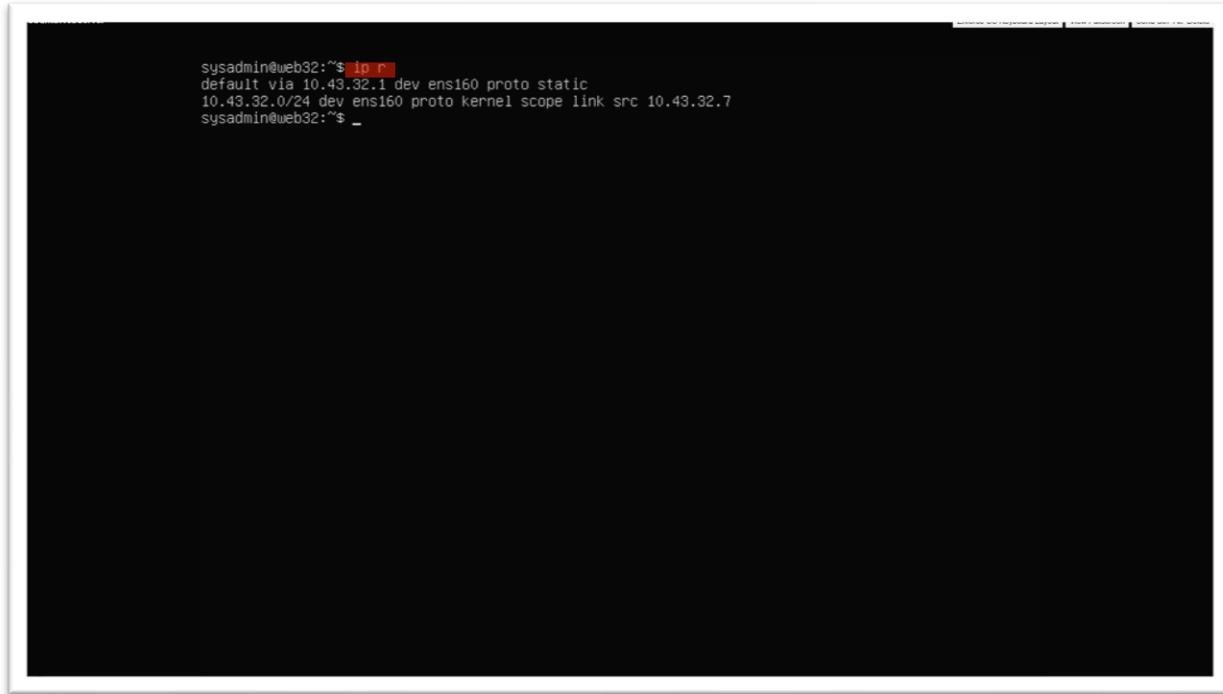


Figure 2: Screenshot of New “Highlighted” Firewall Rules in ServerNet.

2. Linux Server Setup

a. Install and configure the UbuntuWebServer VM

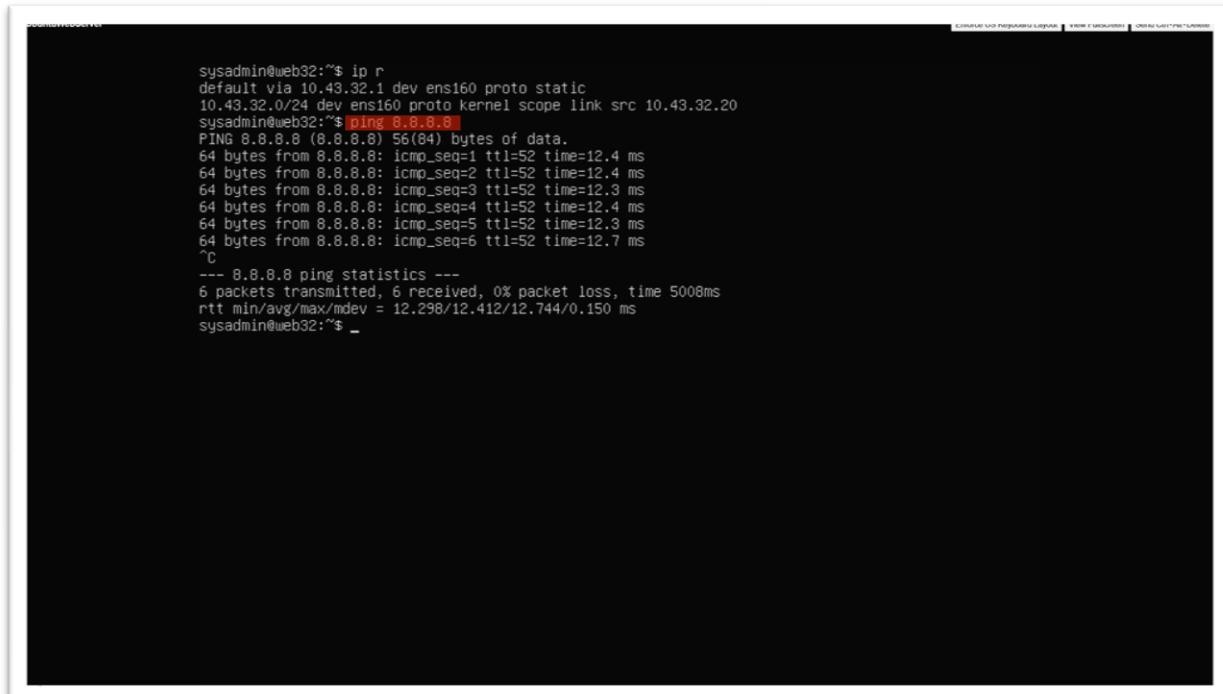
- As we can notice in Figure 3, we type “ip r” command to display connection properties of UbuntuWebServer VM as highlighted below.



```
sysadmin@web32:~$ ip r
default via 10.43.32.1 dev ens160 proto static
10.43.32.0/24 dev ens160 proto kernel scope link src 10.43.32.7
sysadmin@web32:~$ _
```

Figure 3: Screenshot of “ip r” command in UbuntuWebServer VM.

- Now to ping dns.google i.e. “ping 8.8.8.8” to check connection of device with DNS of Google as highlighted in Figure 4.



```
sysadmin@web32:~$ ip r
default via 10.43.32.1 dev ens160 proto static
10.43.32.0/24 dev ens160 proto kernel scope link src 10.43.32.20
sysadmin@web32:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=52 time=12.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=52 time=12.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=52 time=12.3 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=52 time=12.4 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=52 time=12.3 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=52 time=12.7 ms
^C
--- 8.8.8.8 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 12.298/12.412/12.744/0.150 ms
sysadmin@web32:~$ _
```

Figure 4: Screenshot of “ping 8.8.8.8” command in UbuntuWebServer.

- Now, we enter “sudo apt update” to update the operating system for this VM as highlighted in Figure 5.

```

sysadmin@web32:~$ sudo apt update
Ign:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Ign:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Ign:3 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 http://archive.ubuntu.com/ubuntu jammy-security InRelease
Ign:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Ign:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Ign:3 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 http://archive.ubuntu.com/ubuntu jammy-security InRelease
Ign:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Ign:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Ign:3 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 http://archive.ubuntu.com/ubuntu jammy-security InRelease
Err:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Temporary failure resolving 'archive.ubuntu.com'
Err:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Temporary failure resolving 'archive.ubuntu.com'
Err:3 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Temporary failure resolving 'archive.ubuntu.com'
Err:4 http://archive.ubuntu.com/ubuntu jammy-security InRelease
Temporary failure resolving 'archive.ubuntu.com'
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/jammy/InRelease Temporary failure resolving 'archive.ubuntu.com'
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/jammy-updates/InRelease Temporary failure resolving 'archive.ubuntu.com'
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/jammy-backports/InRelease Temporary failure resolving 'archive.ubuntu.com'
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/jammy-security/InRelease Temporary failure resolving 'archive.ubuntu.com'
W: Some index files failed to download. They have been ignored, or old ones used instead.
sysadmin@web32:~$
```

Figure 5: Screenshot of “sudo apt update” to update OS.

- After that, we enter “sudo apt install open-vm-tools” to install VMWare tools in this VM as highlighted in Figure 6.

```

sysadmin@web32:~$ sudo apt install open-vm-tools -y
[sudo] password for sysadmin:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
open-vm-tools is already the newest version (2:11.3.5-1ubuntu4).
open-vm-tools set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
sysadmin@web32:~$
```

Figure 6: Screenshot of “sudo apt install open-vm-tools” to install VMWare Tools.

- Now we enter “sudo systemctl status open-vm-tools” to check the status of VMWare tools if its installed and working or not as highlighted in Figure 7.

```

sysadmin@web32:~$ sudo apt install open-vm-tools -y
[sudo] password for sysadmin:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
open-vm-tools is already the newest version (2:11.3.5-1ubuntu4).
open-vm-tools set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
sysadmin@web32:~$ sudo systemctl status open-vm-tools
● open-vm-tools.service - Service for virtual machines hosted on VMWare
   Loaded: loaded (/lib/systemd/system/open-vm-tools.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2024-10-02 21:36:40 UTC; 6min ago
       Docs: http://open-vm-tools.sourceforge.net/about.php
           Main PID: 795 (vmtoolsd)
             Tasks: 4 (limit: 9409)
            Memory: 7.6M
              CPU: 976ms
            CGroup: /system.slice/open-vm-tools.service
                    └─795 /usr/bin/vmtoolsd

Oct 02 21:36:40 web32 systemd[1]: Started Service for virtual machines hosted on VMWare.
sysadmin@web32:~$ 

```

Figure 7: Screenshot of “sudo systemctl status open-vm-tools” to check status of VMWare tools.

- After that we enter “sudo apt install apache2 libapache2-mod-php php php-mysql php-xml php-mbstring php-apcu php-intl php-gd php-cli php-curl imagemagick inkscape git openssh-server” to install apache2 service. And we can observe the result in Figure 8 and Figure 9.

```

Ign:4 http://archive.ubuntu.com/ubuntu jammy-security InRelease
Ign:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Ign:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Ign:3 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 http://archive.ubuntu.com/ubuntu jammy-security InRelease
Ign:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Ign:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Ign:3 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 http://archive.ubuntu.com/ubuntu jammy-security InRelease
Err:1 http://archive.ubuntu.com/ubuntu jammy InRelease
  Temporary failure resolving 'archive.ubuntu.com'
Err:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
  Temporary failure resolving 'archive.ubuntu.com'
Err:3 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
  Temporary failure resolving 'archive.ubuntu.com'
Err:4 http://archive.ubuntu.com/ubuntu jammy-security InRelease
  Temporary failure resolving 'archive.ubuntu.com'
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
H: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/jammy/InRelease  Temporary failure resolving 'archive.ubuntu.com'
H: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/jammy-updates/InRelease  Temporary failure resolving 'archive.ubuntu.com'
H: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/jammy-backports/InRelease  Temporary failure resolving 'archive.ubuntu.com'
H: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/jammy-security/InRelease  Temporary failure resolving 'archive.ubuntu.com'
H: Some index files failed to download. They have been ignored, or old ones used instead.
sysadmin@web32:~$ sudo apt install open-vm-tools -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
open-vm-tools is already the newest version (2:11.3.5-1ubuntu4).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
sysadmin@web32:~$ 

```

Figure 8: Screenshot of result of “sudo apt install apache2 libapache2-mod-php php php-mysql php-xml php-mbstring php-apcu php-intl php-gd php-cli php-curl imagemagick inkscape git openssh-server” in installation of apache2 service.

```

aspell-autobuildhash: processing: en [en-variant_1].
aspell-autobuildhash: processing: en [en-variant_2].
aspell-autobuildhash: processing: en [en-w_accents-only].
aspell-autobuildhash: processing: en [en-wo_accents-only].
aspell-autobuildhash: processing: en [en_AU-variant_0].
aspell-autobuildhash: processing: en [en_AU-variant_1].
aspell-autobuildhash: processing: en [en_AU-w_accents-only].
aspell-autobuildhash: processing: en [en_AU-wo_accents-only].
aspell-autobuildhash: processing: en [en_CA-variant_0].
aspell-autobuildhash: processing: en [en_CA-variant_1].
aspell-autobuildhash: processing: en [en_CA-w_accents-only].
aspell-autobuildhash: processing: en [en_CA-wo_accents-only].
aspell-autobuildhash: processing: en [en_GB-ise-w_accents-only].
aspell-autobuildhash: processing: en [en_GB-ise-wo_accents-only].
aspell-autobuildhash: processing: en [en_GB-ize-w_accents-only].
aspell-autobuildhash: processing: en [en_GB-ize-wo_accents-only].
aspell-autobuildhash: processing: en [en_GB-variant_0].
aspell-autobuildhash: processing: en [en_GR-variant_1].
aspell-autobuildhash: processing: en [en_US-w_accents-only].
aspell-autobuildhash: processing: en [en_US-wo_accents-only].
Processing triggers for php8.1-cli (8.1.2-1ubuntu2) ...
Processing triggers for libapache2-mod-php8.1 (8.1.2-1ubuntu2) ...
Processing triggers for libgd-pixbuf-2.0-0:amd64 (2.42.8+dfsg-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.

sysadmin@web32:~$
```

Figure 9: Screenshot of result of “`sudo apt install apache2 libapache2-mod-php php php-mysql php-xml php-mbstring php-apcu php-intl php-gd php-cli php-curl imagemagick inkscape git openssh-server`” in installation of apache2 service.

b. Install and Configure the RockyDBServer VM

- We enter “`ip r`” command to display connection properties of RockyDBServer as highlighted in Figure 10.

```

[sysadmin@localhost ~]$ ip r
default via 10.42.32.1 dev ens160 proto static metric 100
10.42.32.1 dev ens160 proto static scope link metric 100
10.43.32.0/24 dev ens160 proto kernel scope link src 10.43.32.30 metric 100
[sysadmin@localhost ~]$ _
```

Figure 10: Screenshot of “`ip r`” command in RockyDBServer.

- After that, we enter “ping 8.8.8.8” to ping the DNS of Google to communicate with it as shown in Figure 11.

```
[sysadmin@localhost ~]# ip r
default via 10.42.32.1 dev ens160 proto static metric 100
10.42.32.1 dev ens160 proto static scope link metric 100
10.43.32.0/24 dev ens160 proto kernel scope link src 10.43.32.38 metric 100
[sysadmin@localhost ~]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=52 time=12.6 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=52 time=12.8 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=52 time=12.7 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=52 time=12.8 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 12.642/12.717/12.787/0.065 ms
[sysadmin@localhost ~]#
```

Figure 11: Screenshot of “ping 8.8.8.8” command in RockyDBServer.

- Now we enter “sudo yum update” to update the operating system of RockyBDServer. So we can see the result of that command in Figure 12 and 13.

Transaction ID	Name	Size	Time
(307/333)	sssd-krb5-common-2.9.4-4.e18_10.x86_64.rpm	1.6 MB/s 285 kB	00:00
(308/333)	sssd-ldap-2.9.4-4.e18_10.x86_64.rpm	1.9 MB/s 258 kB	00:00
(309/333)	sssd-nfs-idmap-2.9.4-4.e18_10.x86_64.rpm	1.2 MB/s 129 kB	00:00
(310/333)	sssd-proxy-2.9.4-4.e18_10.x86_64.rpm	2.3 MB/s 163 kB	00:00
(311/333)	sudo-1.9.5p2-1.e18_9.x86_64.rpm	1.4 MB/s 1.0 MB	00:00
(312/333)	systemd-libs-239-82.e18_10.2.x86_64.rpm	1.7 MB/s 1.1 MB	00:00
(313/333)	systemd-239-82.e18_10.2.x86_64.rpm	2.3 MB/s 3.6 MB	00:01
(314/333)	systemd-pam-239-82.e18_10.2.x86_64.rpm	1.7 MB/s 512 kB	00:00
(315/333)	tar-1.30-9.e18.x86_64.rpm	1.7 MB/s 838 kB	00:00
(316/333)	teamd-1.31-4.e18.x86_64.rpm	1.6 MB/s 129 kB	00:00
(317/333)	tpm2-tss-2.3.2-6.e18.x86_64.rpm	1.4 MB/s 274 kB	00:00
(318/333)	systemd-udev-239-82.e18_10.2.x86_64.rpm	1.5 MB/s 1.6 MB	00:01
(319/333)	trousers-0.3.15-2.e18.x86_64.rpm	1.1 MB/s 152 kB	00:00
(320/333)	trousers-lib-0.3.15-2.e18.x86_64.rpm	1.3 MB/s 167 kB	00:00
(321/333)	tuned-2.22.1-4.e18_10.noarch.rpm	1.4 MB/s 366 kB	00:00
(322/333)	usbutils-015-1.e18.x86_64.rpm	1.3 MB/s 111 kB	00:00
(323/333)	tzdata-2024a-1.e18.noarch.rpm	901 kB/s 474 kB	00:00
(324/333)	util-linux-user-2.32.1-46.e18.x86_64.rpm	1.6 MB/s 182 kB	00:00
(325/333)	vdo-6.2.9.7-14.e18.x86_64.rpm	1.3 MB/s 665 kB	00:00
(326/333)	virt-what-1.25-4.e18.x86_64.rpm	636 kB/s 37 kB	00:00
(327/333)	which-2.21-28.e18.x86_64.rpm	777 kB/s 49 kB	00:00
(328/333)	xfsdump-3.1.8-7.e18_9.x86_64.rpm	1.2 MB/s 332 kB	00:00
(329/333)	util-linux-2.32.1-46.e18.x86_64.rpm	1.9 MB/s 2.5 MB	00:01
(330/333)	yum-4.7.8-28.e18.noarch.rpm	1.4 MB/s 288 kB	00:00
(331/333)	zlib-1.2.11-26.e18.x86_64.rpm	1.5 MB/s 182 kB	00:00
(332/333)	xfsprogs-5.0.0-12.e18.x86_64.rpm	1.4 MB/s 1.1 MB	00:00
(333/333)	linux-firmware-20240610-122.git98df68d2.e18_10.noarch.rpm	10 MB/s 363 MB	00:36
Total		12 MB/s 729 MB	01:02
Rocky Linux 8 - AppStream		819 kB/s 1.6 kB	00:00
Importing GPG key 0x6D745A60:			
Userid : "Release Engineering <infrastructure@rockylinux.org>"			
Fingerprint: 7B51 C470 A929 F454 CEBE 37B7 15AF 5DAC 6D74 5A60			
From : /etc/pki/rpm-gpg/RPM-GPG-KEY-rockyofficial			
Is this ok [y/N]: y			
Key imported successfully			

Figure 12: Screenshot of result for “sudo yum update” command in RockyBDServer.

```

libsoup-2.62.3-5.el8.x86_64
libstemmer-0.10.565svn.el8.x86_64
libxcb-1.13.1-1.el8.x86_64
libxcbcommon-0.9.1-1.el8.x86_64
pixman-0.38.4-4.el8.x86_64
policycoreutils-python-utils-2.9-26.el8_10.noarch
protobuf-c-1.3.0-8.el8.x86_64
python3-audit-3.1.2-1.el8.x86_64
python3-bind-32:9.11.36-16.el8_10.noarch
python3-cairo-1.16.3-6.el8.x86_64
python3-distro-1.4.0-2.module+el8.18.0+1592+61442852.noarch
python3-gobject-3.28.3-2.el8.x86_64
python3-libsemanage-2.9-9.el8_6.x86_64
python3-magic-5.33-26.el8.noarch
python3-pexpect-4.3.1-3.el8.noarch
python3-ply-3.9-9.el8.noarch
python3-policycoreutils-2.9-26.el8_10.noarch
python3-psutil-5.4.3-11.el8.x86_64
python3-ptyprocess-0.5.2-4.el8.noarch
python3-pydbus-0.6.0-5.el8.noarch
python3-setools-4.3.0-5.el8.x86_64
python3-setuptools-39.2.0-8.el8_10.noarch
python3-systemml-2.34-8.el8.x86_64
python3-tracer-1.1-1.el8.noarch
python3-unbound-1.16.2-5.el8_9.6.x86_64
setroubleshoot-plugins-3.3.14-1.el8.noarch
setroubleshoot-server-3.3.26-6.el8.x86_64
sscg-3.0.0-7.el8.x86_64
tracer-common-1.1-1.el8.noarch
unbound-libs-1.16.2-5.el8_9.6.x86_64
xkeyboard-config-2.28-1.el8.noarch

Complete!
[sysadmin@localhost ~]$
[sysadmin@localhost ~]$
[sysadmin@localhost ~]$
[sysadmin@localhost ~]$

```

Figure 13: Screenshot of result for “sudo yum update” command in RockyBDServer.

- Now to enter “sudo yum install open-vm-tools” to install VMWare Tools in RockyBDServer and the result is in Figure 14.

```

Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing : libxslt-1.1.32-6.el8.x86_64
  Installing : libtool-ltdl-2.4.6-25.el8.x86_64
  Running scriptlet: libtool-ltdl-2.4.6-25.el8.x86_64
  Installing : xmлsec1-1.2.25-8.el8_10.x86_64
  Installing : xmлsec1-openssl-1.2.25-8.el8_10.x86_64
  Installing : libpciaccess-0.14-1.el8.x86_64
  Installing : libdrm-2.4.115-2.el8.x86_64
  Installing : fuse-common-3.3.0-19.el8.x86_64
  Installing : fuse-2.9.7-19.el8.x86_64
  Installing : libmspack-0.7-0.3.alpha.el8.4.x86_64
  Installing : open-vm-tools-12.3.5-2.el8.x86_64
  Running scriptlet: open-vm-tools-12.3.5-2.el8.x86_64
  Verifying : libdrm-2.4.115-2.el8.x86_64
  Verifying : libmspack-0.7-0.3.alpha.el8.4.x86_64
  Verifying : open-vm-tools-12.3.5-2.el8.x86_64
  Verifying : xmлsec1-1.2.25-8.el8_10.x86_64
  Verifying : xmлsec1-openssl-1.2.25-8.el8_10.x86_64
  Verifying : fuse-2.9.7-19.el8.x86_64
  Verifying : fuse-common-3.3.0-19.el8.x86_64
  Verifying : libpciaccess-0.14-1.el8.x86_64
  Verifying : libtool-ltdl-2.4.6-25.el8.x86_64
  Verifying : libxslt-1.1.32-6.el8.x86_64
  Installed: fuse-2.9.7-19.el8.x86_64
              libdrm-2.4.115-2.el8.x86_64
              libpciaccess-0.14-1.el8.x86_64
              libxslt-1.1.32-6.el8.x86_64
              xmлsec1-1.2.25-8.el8_10.x86_64
  fuse-common-3.3.0-19.el8.x86_64
  libmspack-0.7-0.3.alpha.el8.4.x86_64
  libtool-ltdl-2.4.6-25.el8.x86_64
  open-vm-tools-12.3.5-2.el8.x86_64
  xmлsec1-openssl-1.2.25-8.el8_10.x86_64

Complete!
[sysadmin@localhost ~]$

```

Figure 14: Screenshot of Result for “sudo yum install open-vm-tools” command in RockyBDServer.

- So to verify the installation of VMWare Tools, we use “sudo systemctl enable vmtoolsd” and “sudo systemctl start vmtoolsd” as highlighted and we can see the result as shown in Figure 15.

```
[sysadmin@localhost ~]$ sudo systemctl enable vmtoolsd
[sysadmin@localhost ~]$ sudo systemctl start vmtoolsd
[sysadmin@localhost ~]$ [ 1426.217417] MET: Registered protocol family 48
```

Figure 15: Screenshot of verify VMWare Tools by “sudo systemctl start vmtoolsd”.

- Finally, now we install additional software packages by putting “sudo yum install mariadb-server” and the result of that command is shown in Figure 16.

```
RockyBDServer
[sysadmin@localhost ~]$ sudo yum install mariadb-server
perl-Goptopt-Long-1:2.50-4.el8.noarch
perl-HTTP-Tiny-0.074-3.el8.noarch
perl-IO-1.38-422.el8.x86_64
perl-IO-Socket-IP-0.39-5.el8.noarch
perl-IO-Socket-SSL-2.066-4.module+e18.9.8+1517+e71a7a62.noarch
perl-MIME-Base64-3.15-396.el8.x86_64
perl-Math-BigInt-1:1.9998.11-7.el8.noarch
perl-Math-Complex-1.59-422.el8.noarch
perl-Mozilla-Ca-20160104-7.module+e18.9.8+1521+e0101edce.noarch
perl-Net-SSLeay-1.88-2.module+e18.9.8+1517+e71a7a62.x86_64
perl-PathTools-3.74-1.el8.x86_64
perl-Pod-Escapes-1:1.07-395.el8.noarch
perl-Pod-Perldoc-3.28-396.el8.noarch
perl-Pod-Simple-1:3.35-395.el8.noarch
perl-Pod-Usage-4:1.69-395.el8.noarch
perl-Scalar-List-Utils-3:1.49-2.el8.x86_64
perl-Socket-4:2.027-3.el8.x86_64
perl-Storable-1:3.11-3.el8.x86_64
perl-Term-ANSIColor-1.06-396.el8.noarch
perl-Term-Cap-1.17-395.el8.noarch
perl-Text-ParseWords-3.30-395.el8.noarch
perl-Text-TabsWrap-2013.0523-395.el8.noarch
perl-Time-Local-1:1.280-1.el8.noarch
perl-URI-1.73-3.el8.noarch
perl-Unicode-Normalize-1.25-396.el8.x86_64
perl-constant-1.33-396.el8.noarch
perl-interpreter-4:5.26-3-422.el8.x86_64
perl-libnet-3.11-3.el8.noarch
perl-libs-4:5.26-3-422.el8.x86_64
perl-macros-4:5.26-3-422.el8.x86_64
perl-parent-1:8.237-1.el8.noarch
perl-podlators-4.11-1.el8.noarch
perl-threads-1:2.21-2.el8.x86_64
perl-threads-shared-1.58-2.el8.x86_64

Complete!
[sysadmin@localhost ~]$
```

Figure 16: Screenshot of result of “sudo yum install mariadb-server” command in RockyBDServer.

3. User and Group Creation

- On “terminal”, we enter “sudo adduser fahmed29 (my UBTName) to add new user as highlighted in Figure 17. And enter the password as “Change.me!” and press “Enter” for rest of the information to get default values in that.

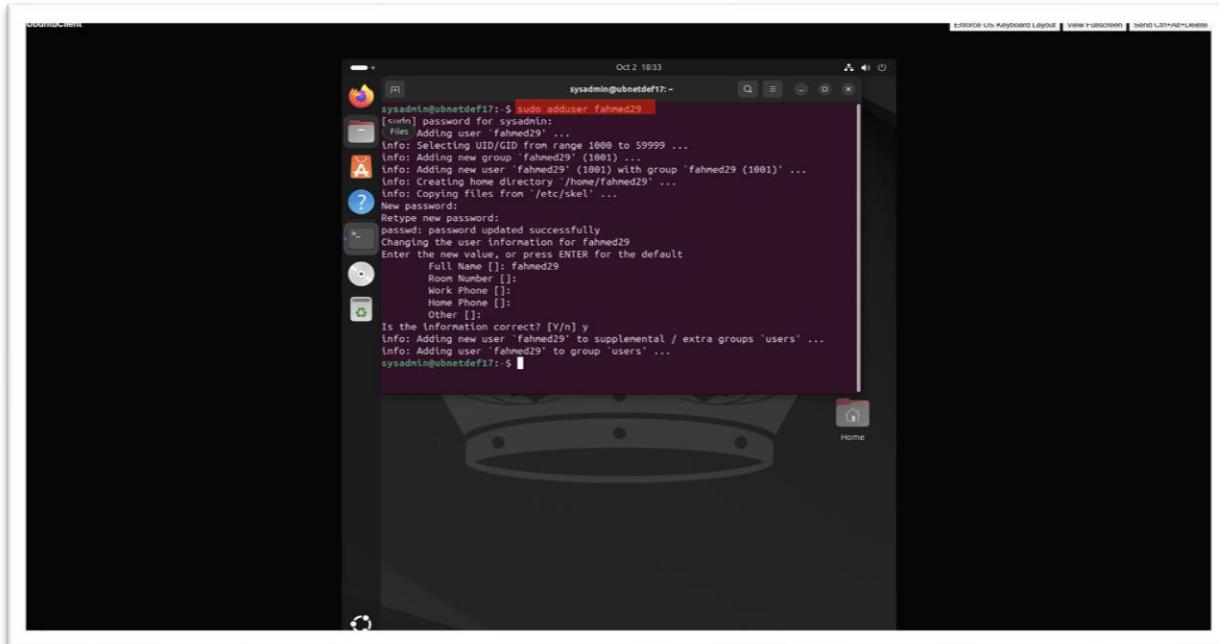


Figure 17: Screenshot of making new user using “sudo adduser fahmed29” command.

- Now we can also make the new user using different commands- useradd command “sudo useradd -n -s /bin/bash kpcleary” where “-m” to create a home directory and “-s /bin/bash” to set the default shell to bash as shown in Figure 18. Also we can enter “sudo passwd kpcleary” to add password to that new user as highlighted in Figure 18.

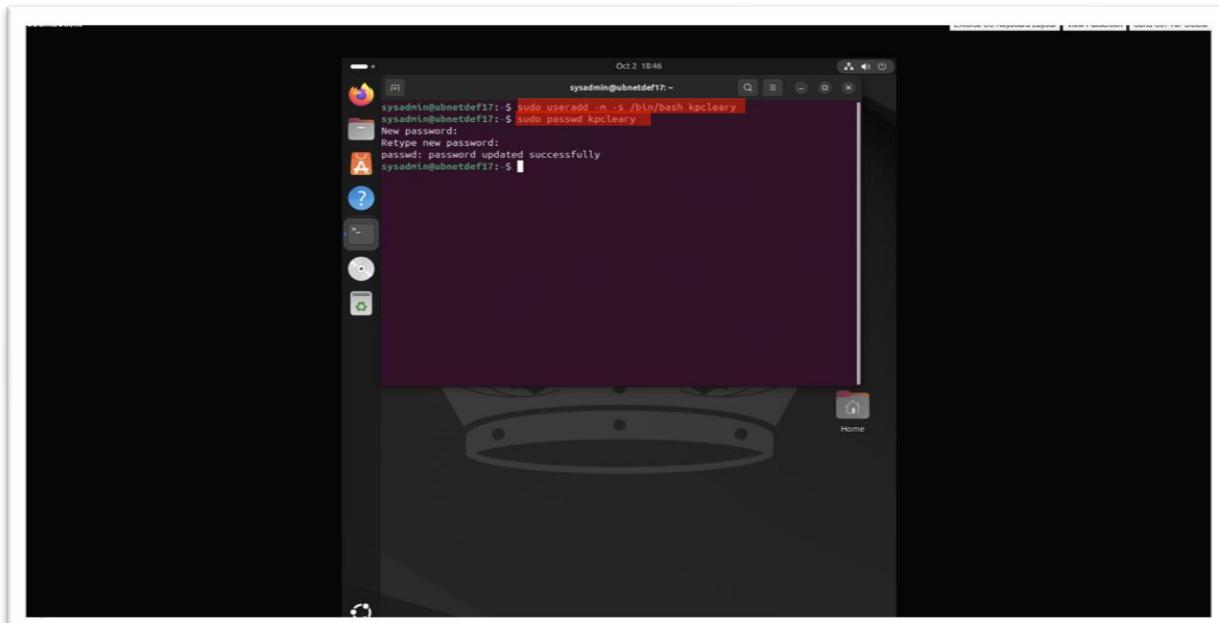
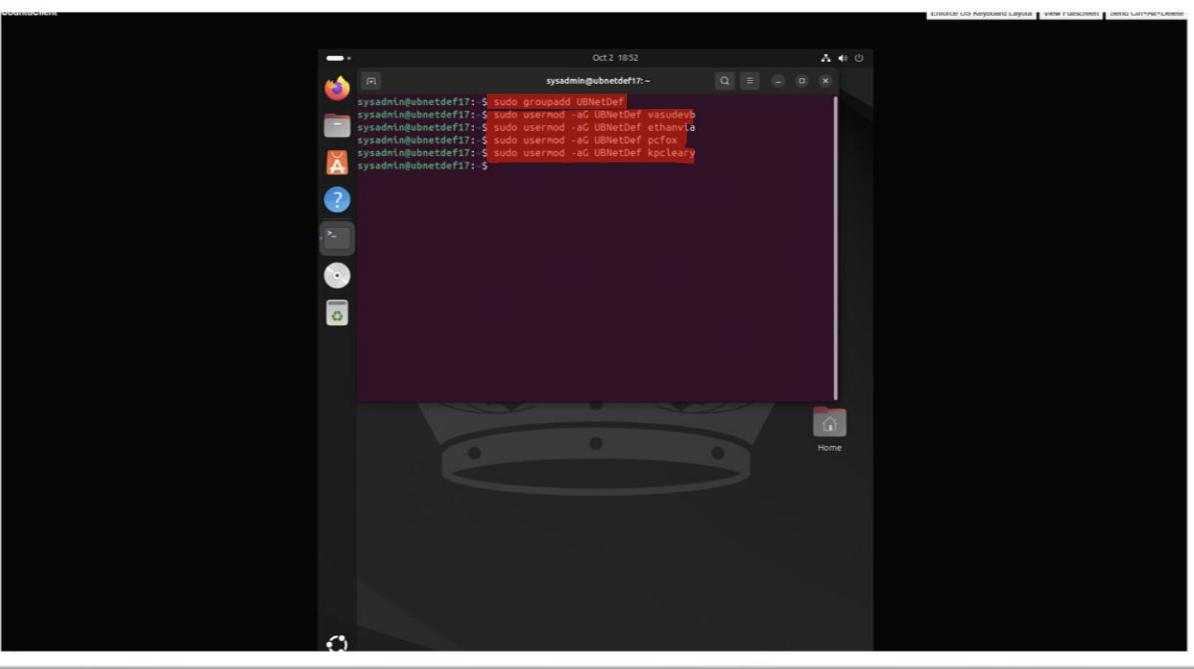


Figure 18: Screenshot of “sudo useradd -n -s /bin/bash kpcleary” command to add new user.

- Now to create a new group- UBNetDef using “sudo groupadd UBNetDef” as highlighted below and add new users vasudevb, ethanvia, pfox and kpcklary using command “sudo usermod -aG UBNetDef username” as highlighted in Figure 19.



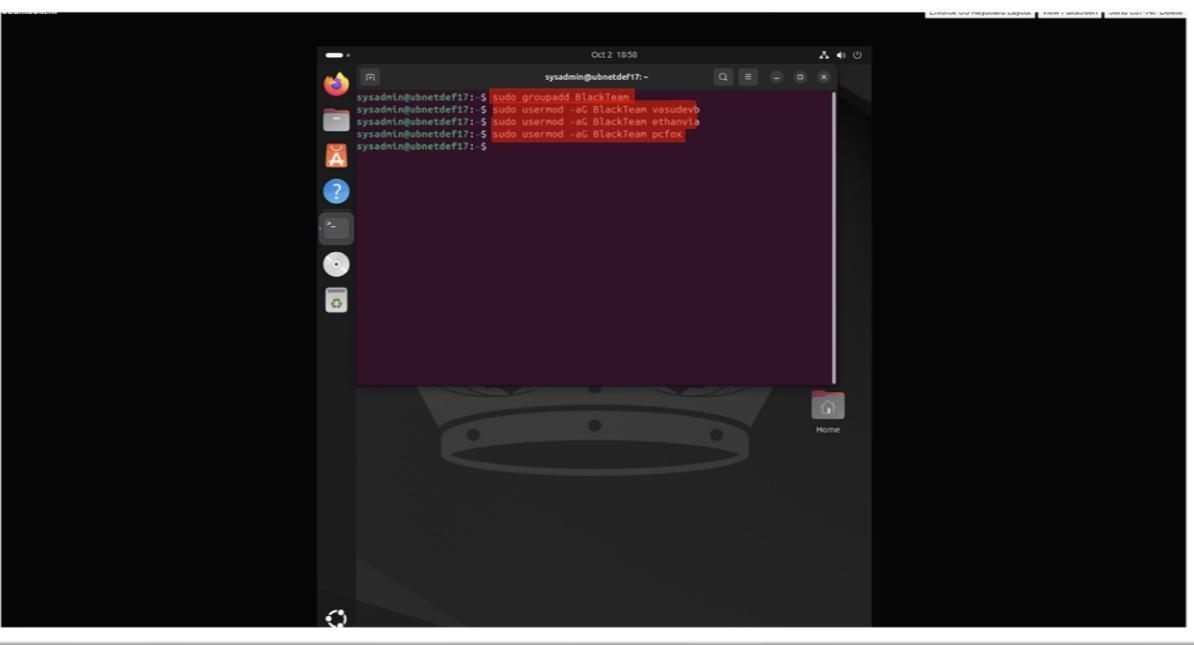
```
Oct 2 18:52
sysadmin@ubnetdef17:~
```

The terminal window shows the following commands being run:

```
sysadmin@ubnetdef17:~$ sudo groupadd UBNetDef
sysadmin@ubnetdef17:~$ sudo usermod -aG UBNetDef vasudevb
sysadmin@ubnetdef17:~$ sudo usermod -aG UBNetDef ethanvia
sysadmin@ubnetdef17:~$ sudo usermod -aG UBNetDef pfox
sysadmin@ubnetdef17:~$ sudo usermod -aG UBNetDef kpcklary
sysadmin@ubnetdef17:~$
```

Figure 19: Screenshot of commands “`sudo groupadd UBNetDef`” to create new group and add users in that.

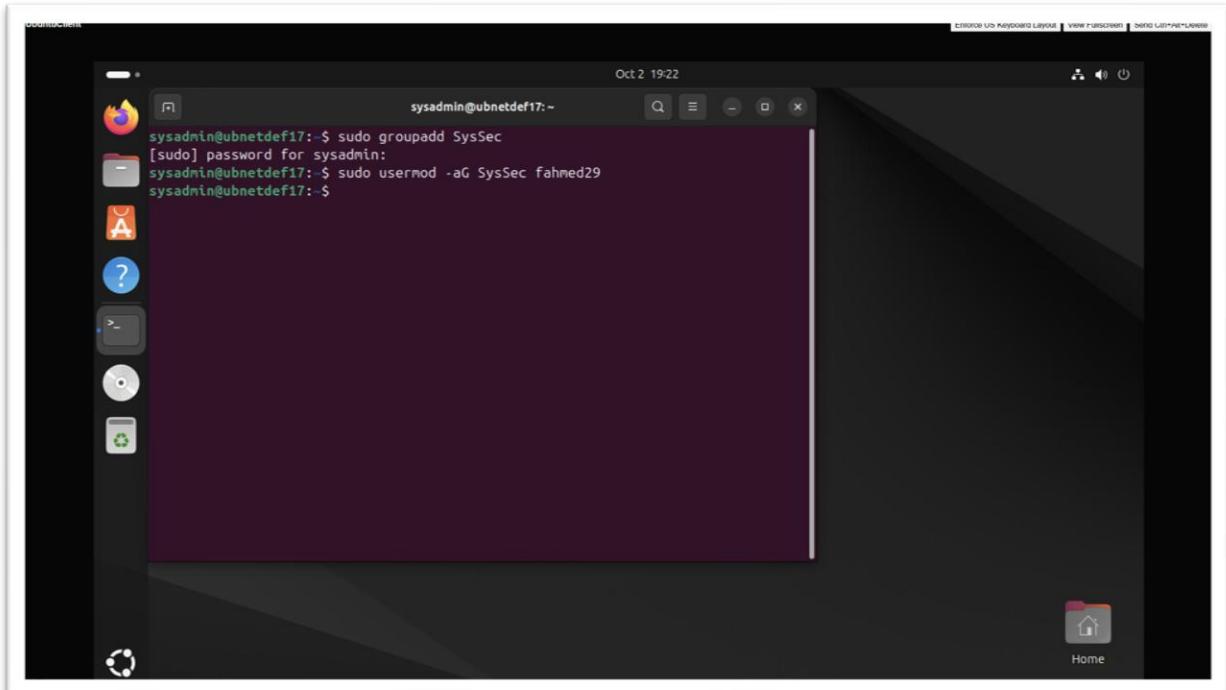
- After that, we create new group- BlackTeam using “sudo groupadd BlackTeam” as highlighted below and also add new users vasudevb, ethanvia and pfox using command “sudo usermod -aG BlackTeam username” as highlighted in Figure 20.



```
Oct 2 18:58
sysadmin@ubnetdef17:~$ sudo groupadd BlackTeam
sysadmin@ubnetdef17:~$ sudo usermod -aG BlackTeam vasudevb
sysadmin@ubnetdef17:~$ sudo usermod -aG BlackTeam ethanvia
sysadmin@ubnetdef17:~$ sudo usermod -aG BlackTeam pfox
sysadmin@ubnetdef17:~$
```

Figure 20: Screenshot of commands “`sudo groupadd BlackTeam`” to create new group and also add users to it.

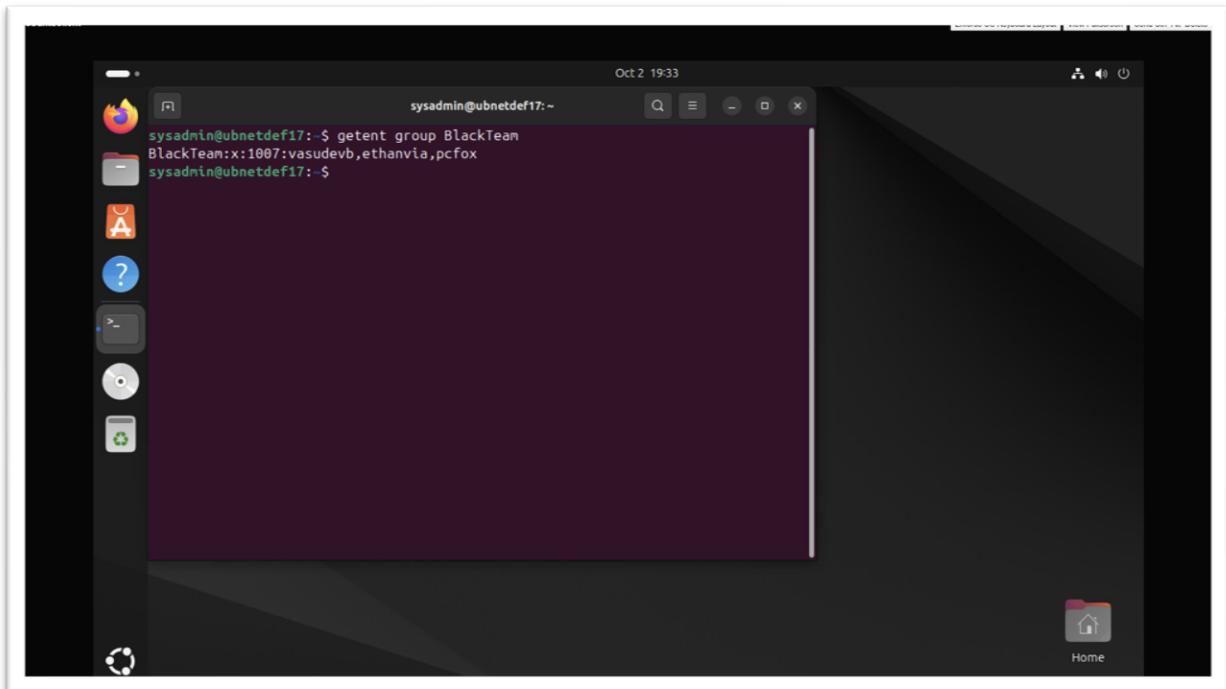
- Now, we create new group “SysSec” using same commands and adding new user “fahmed29” by also using same adding command.



```
sysadmin@ubnetdef17: ~
sysadmin@ubnetdef17: $ sudo groupadd SysSec
[sudo] password for sysadmin:
sysadmin@ubnetdef17: $ sudo usermod -aG SysSec fahmed29
sysadmin@ubnetdef17: $
```

Figure 21: Screenshot of commands “`sudo groupadd SysSec`” to create new group and also add users to it.

- So now, we use “getent group BlackTeam” command to list and verify all members or users of that group.



```
sysadmin@ubnetdef17: ~
sysadmin@ubnetdef17: $ getent group BlackTeam
BlackTeam:x:1007:vasudevb,ethanvia,pcfox
sysadmin@ubnetdef17: $
```

Figure 22: Screenshot of using command “`getent group BlackTeam`” to list all users.

4. Using Linux Hardening

- Now to edit the database of password in location “/etc/login.defs” so we enter “sudo nano /etc/login.defs” as highlighted in Figure 23.

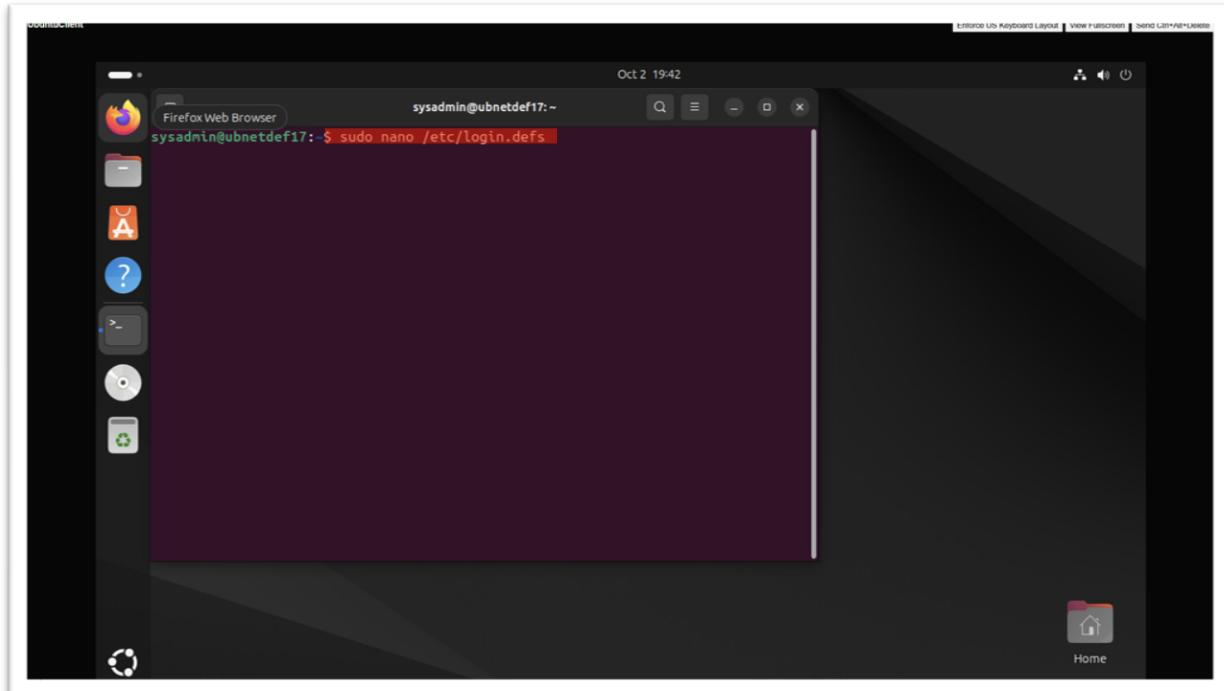


Figure 23: Screenshot of “sudo nano /etc/login.defs” to edit password policy.

- Now edit the number in “PASS_MAX_DAYS” to 70 days and then press ctrl+X then save it and press Enter to exit Sudo as shown below.

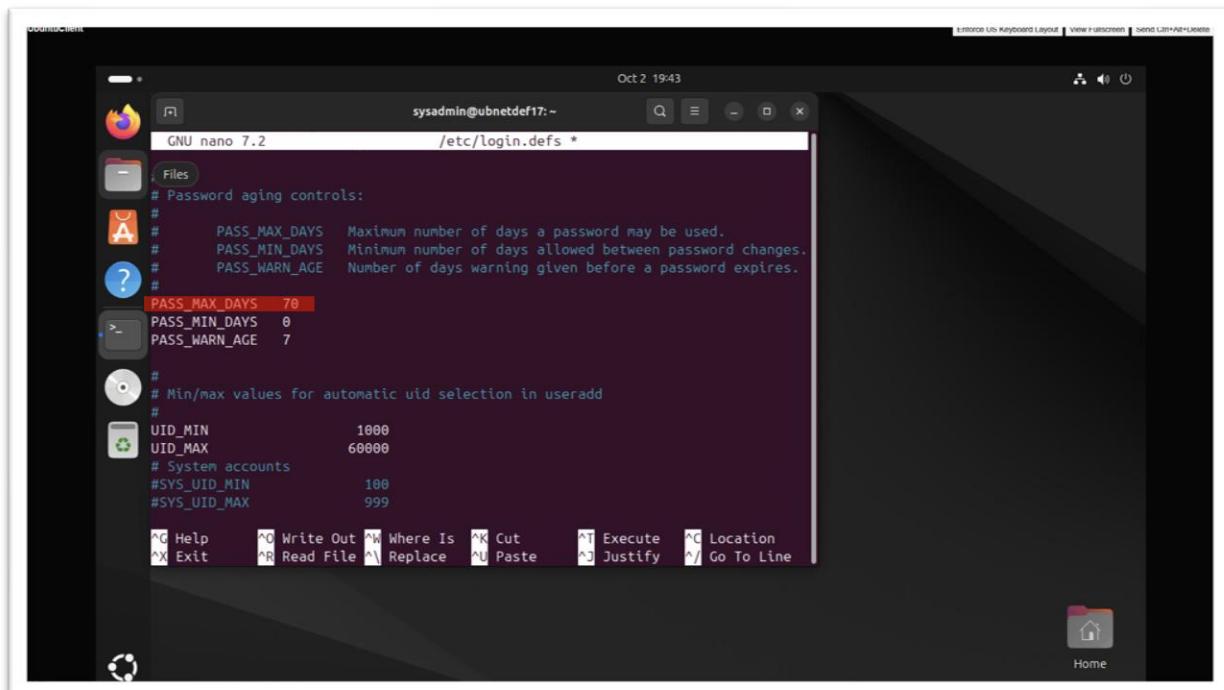


Figure 24: Screenshot of editing “PASS_MAX_DAYS” to 70 days to change password policy.

- Now enter “Apply Security Updates Only” to the “sysadmin” i.e. “`sudo apt install unattended-upgrades`” and we can observe that it is able to run as shown in Figure 25.

Figure 25: Screenshot of “sudo apt install unattended-upgrades” to Apply Security Updates Only.

- Now we change the user to fahmed29 as shown in Figure 26 and we enter “sudo vim” in that but we don’t get output as we can observe in Figure 27.

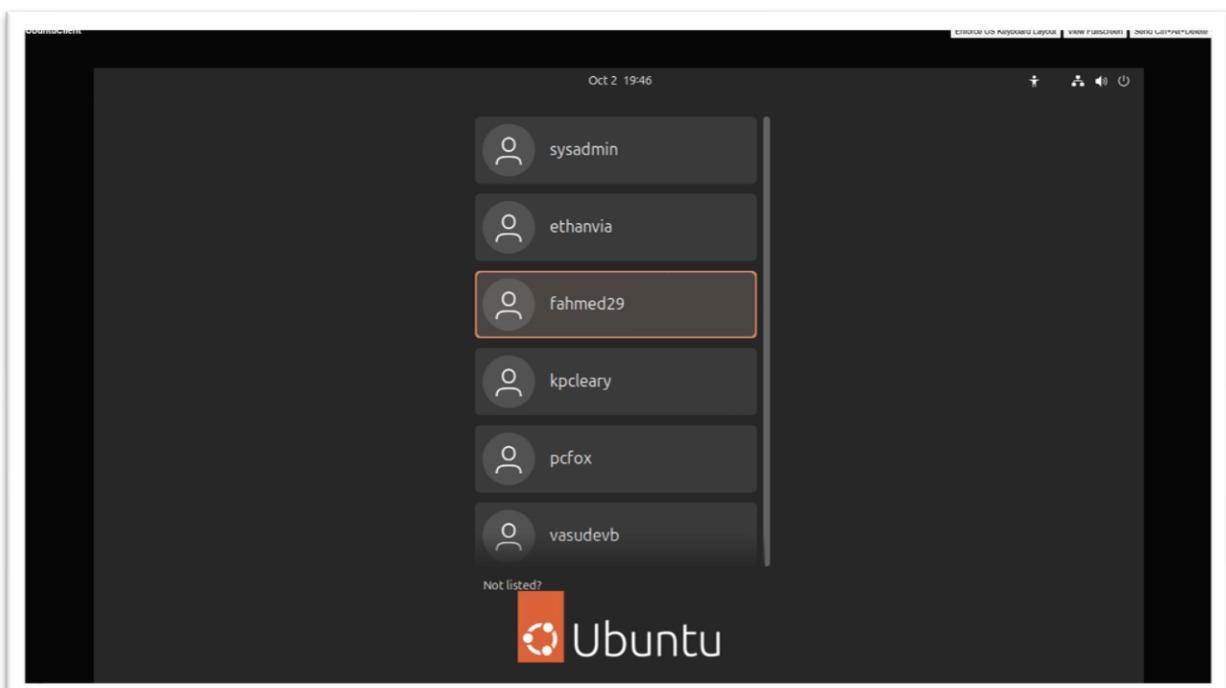


Figure 26: Screenshot of selecting different user “fahmed29”.

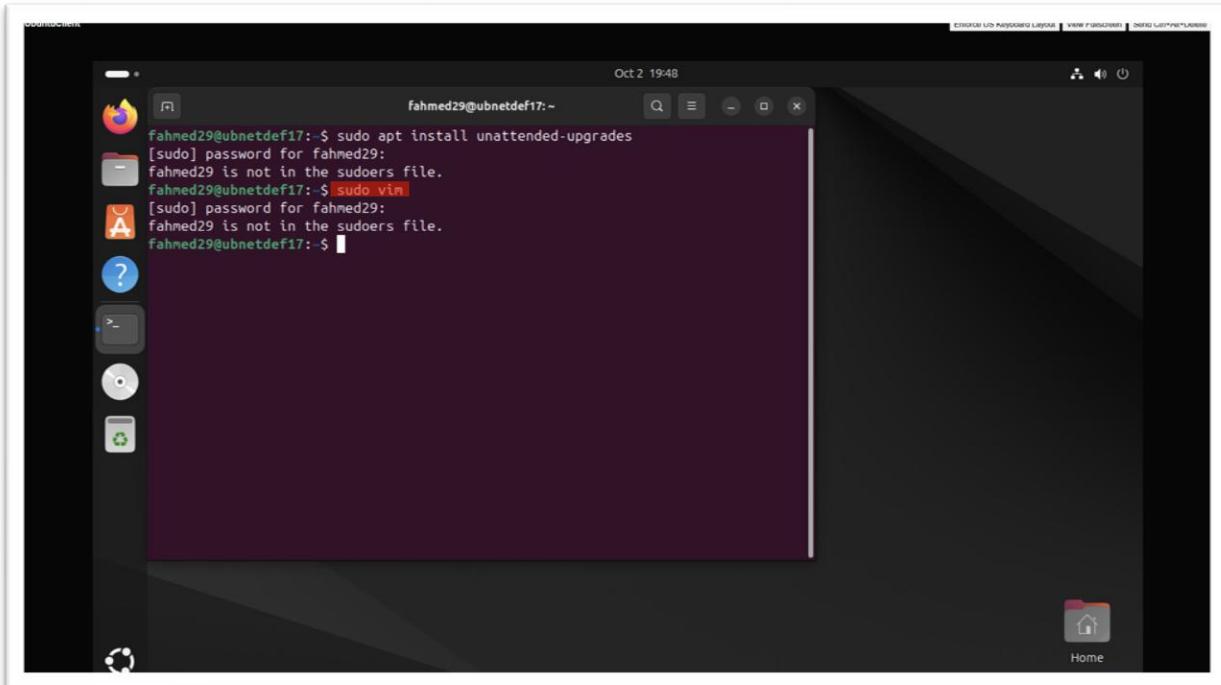


Figure 27: Screenshot of “sudo vim” to verify the output in different user.

- Now we type command “sudo visudo” to edit all the permissions as highlighted in Figure 28.

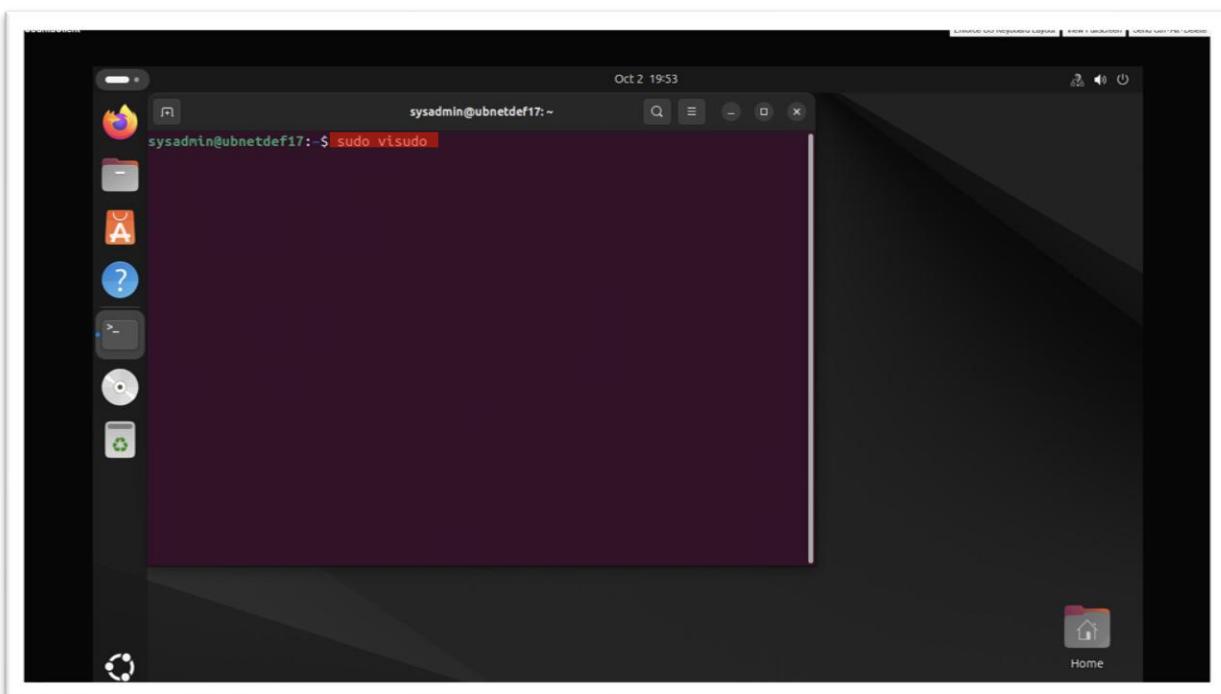


Figure 28: Screenshot of “sudo visudo” to edit permissions.

- So, we type new line under “group sudo permission” – “%BlackTeam ALL=(ALL:ALL) ALL” as shown in Figure 29 to give full permission.

```

sysadmin@ubnetdef17:~$ nano /etc/sudoers.tmp
GNU nano 7.2          /etc/sudoers.tmp

# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL
# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
%BlackTeam ALL=(ALL:ALL) ALL
# See sudoers(5) for more information on "@include" directives:
@includedir /etc/sudoers.d

[ Wrote 57 lines ]

```

Figure 29: Screenshot of entering “%BlackTeam ALL=(ALL:ALL) ALL” to give all permissions.

- Now we enter “sudo chmod 700 \$(which whoami)” to give file permissions for whoami to allow only root user to read, write, and execute it as highlighted in Figure 30.

```

sysadmin@ubnetdef17:~$ sudo chmod 700 $(which whoami)

```

Figure 30: Screenshot of using “sudo chmod 700 \$(which whoami)” to give root permissions.

- So after that, we use “`sudo chmod 750 /etc/hostname`” to give file permissions for `/etc/hostname` to allow members of the BlackTeam group to read and execute it as highlighted in Figure 31.

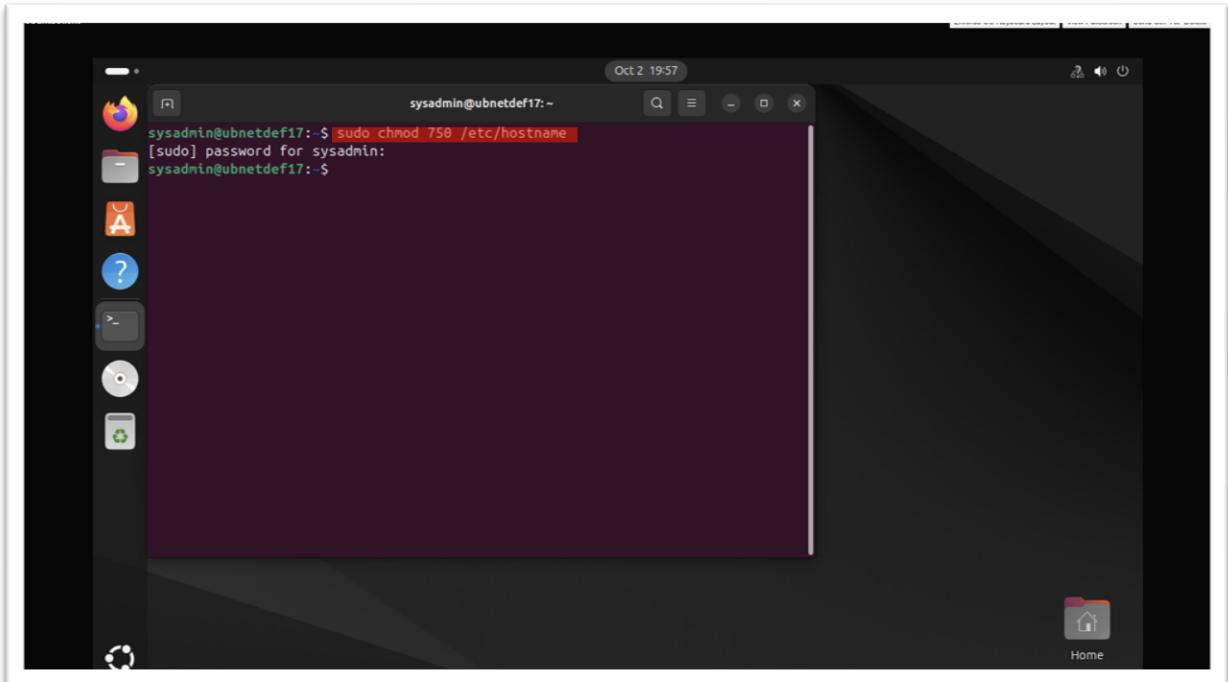


Figure 31: Screenshot of entering “`sudo chmod 750 /etc/hostname`” to give BlackTeam permissions.

- After that, type “`sudo chown pcfox:pcfox /etc/hostname`” to change the ownership of `/etc/hostname` to `pcfox` in highlighted in Figure 32.

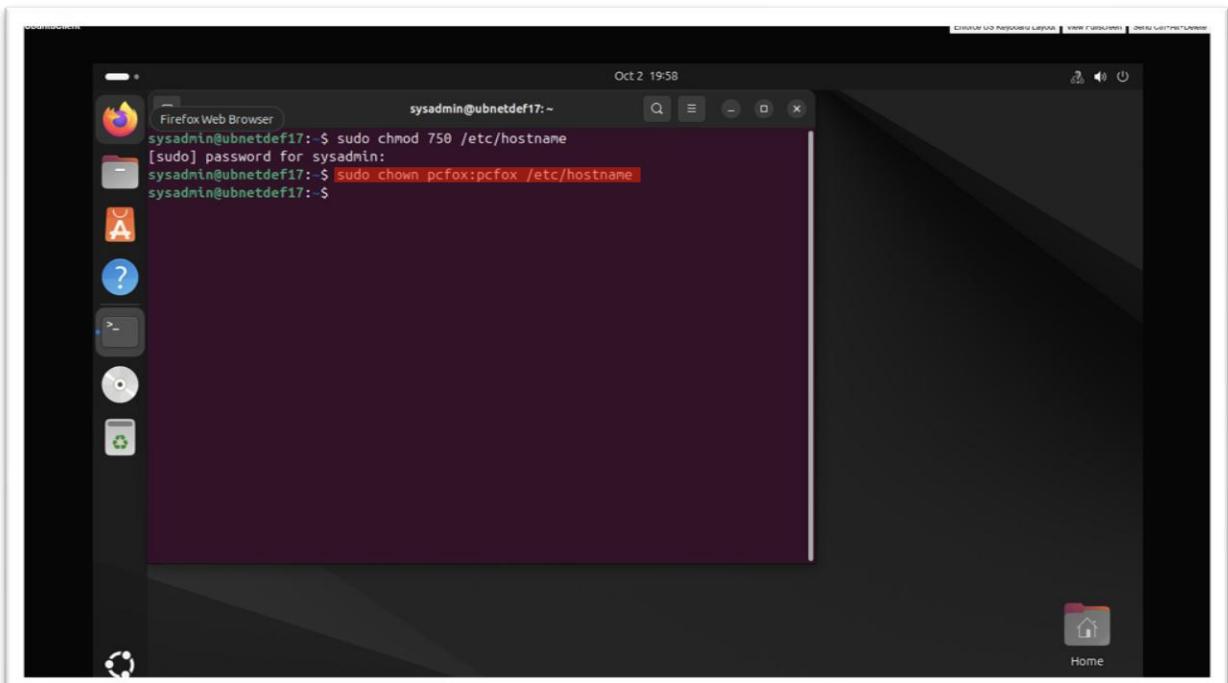


Figure 32: Screenshot of “`sudo chown pcfox:pcfox /etc/hostname`” to change ownership.

- Now, enter “sudo apt install libpam-pwquality” to install this library to edit password properties in highlighted in Figure 33.

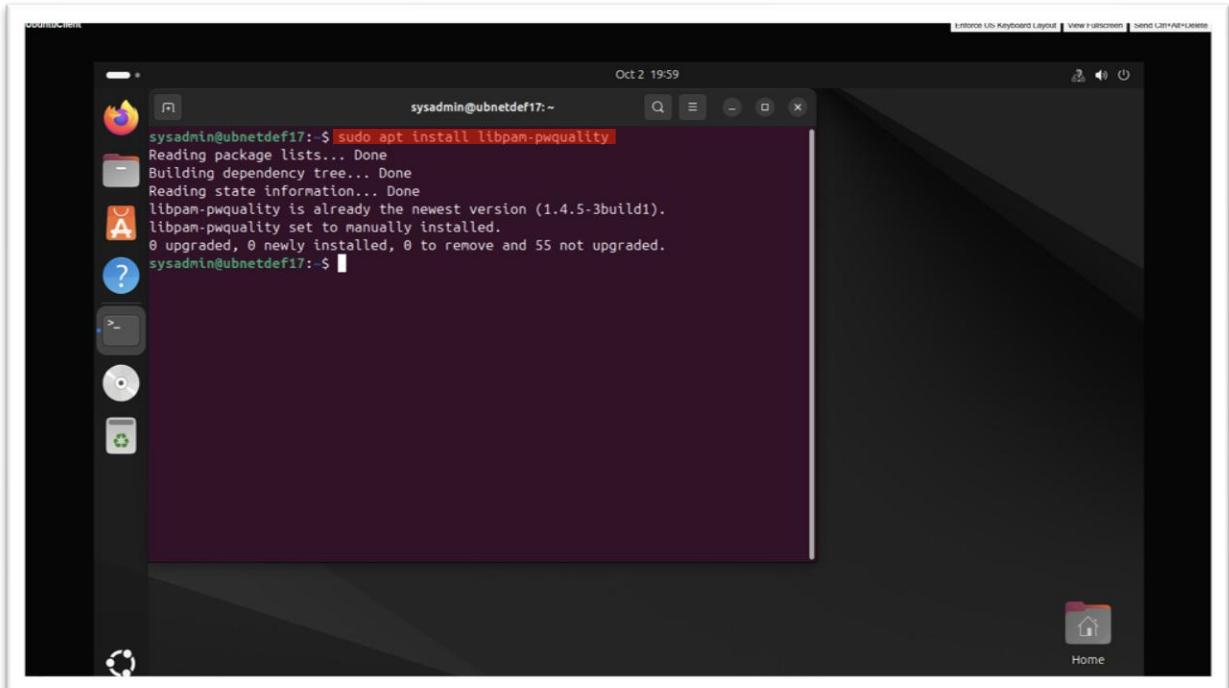


Figure 33: Screenshot of “sudo apt install libpam-pwquality” to install library.

- Now we edit the security file to change password properties by entering “sudo nano /etc/security/pwquality.conf” as highlighted in Figure 34.

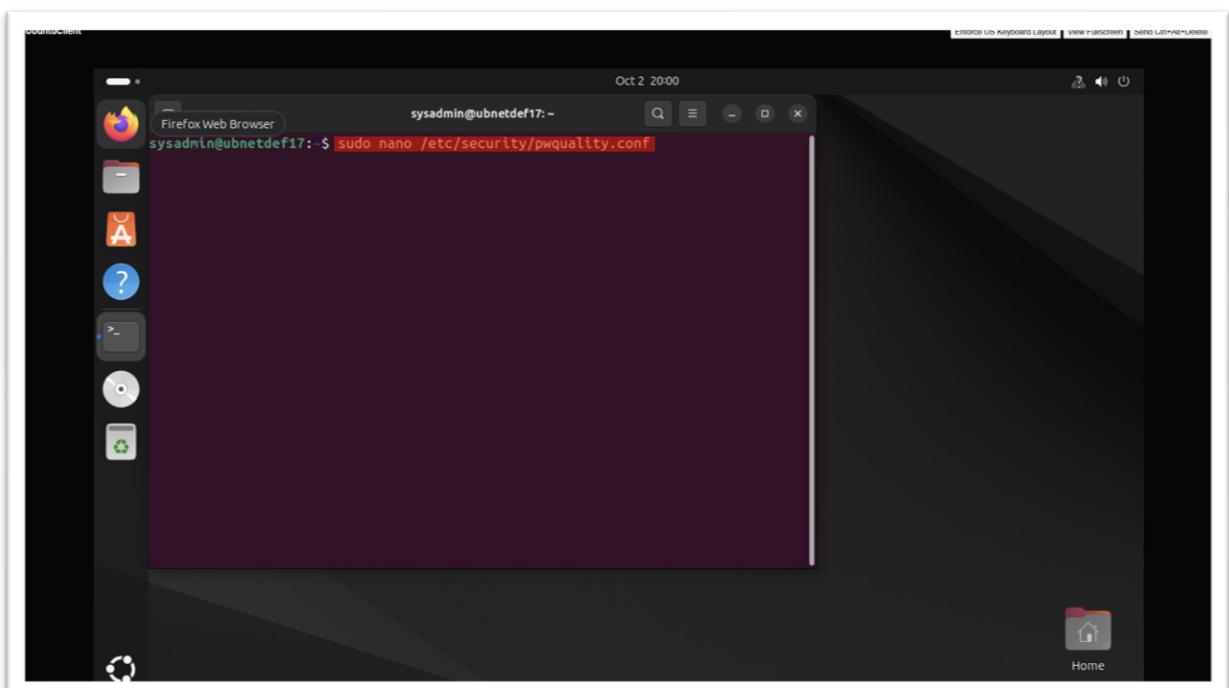
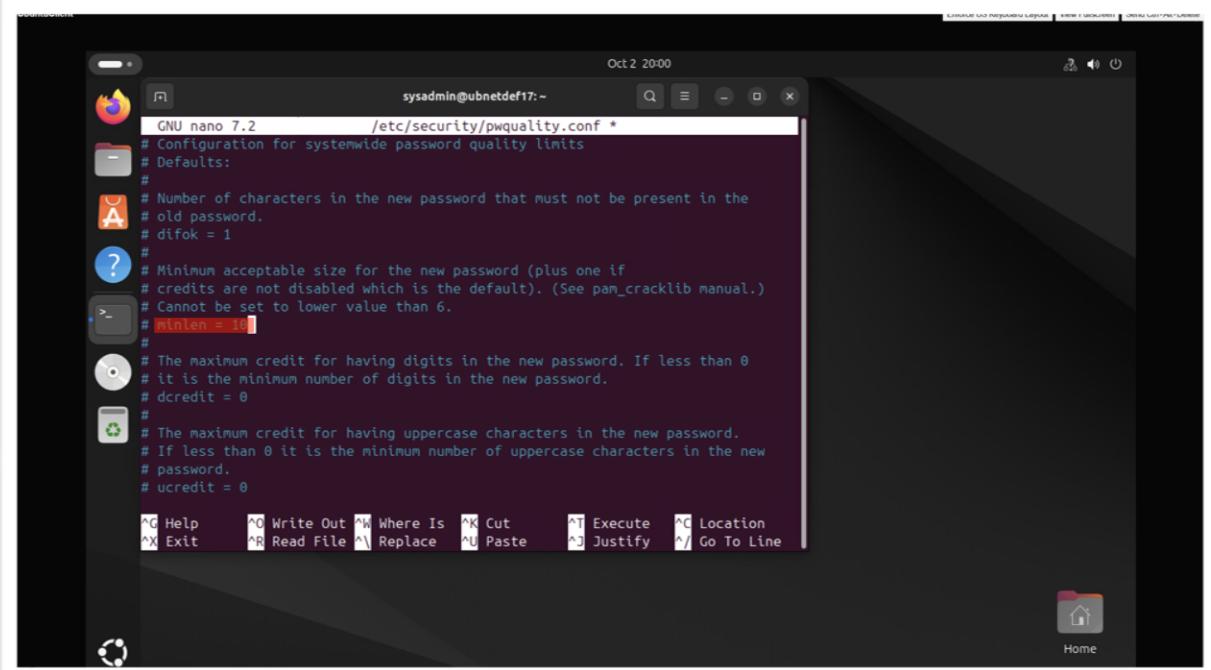


Figure 34: Screenshot of typing “sudo nano /etc/security/pwquality.conf” to edit password properties.

- Now we change “minlen= 10” to accept minimum length of 10 characters in shown in Figure 35.

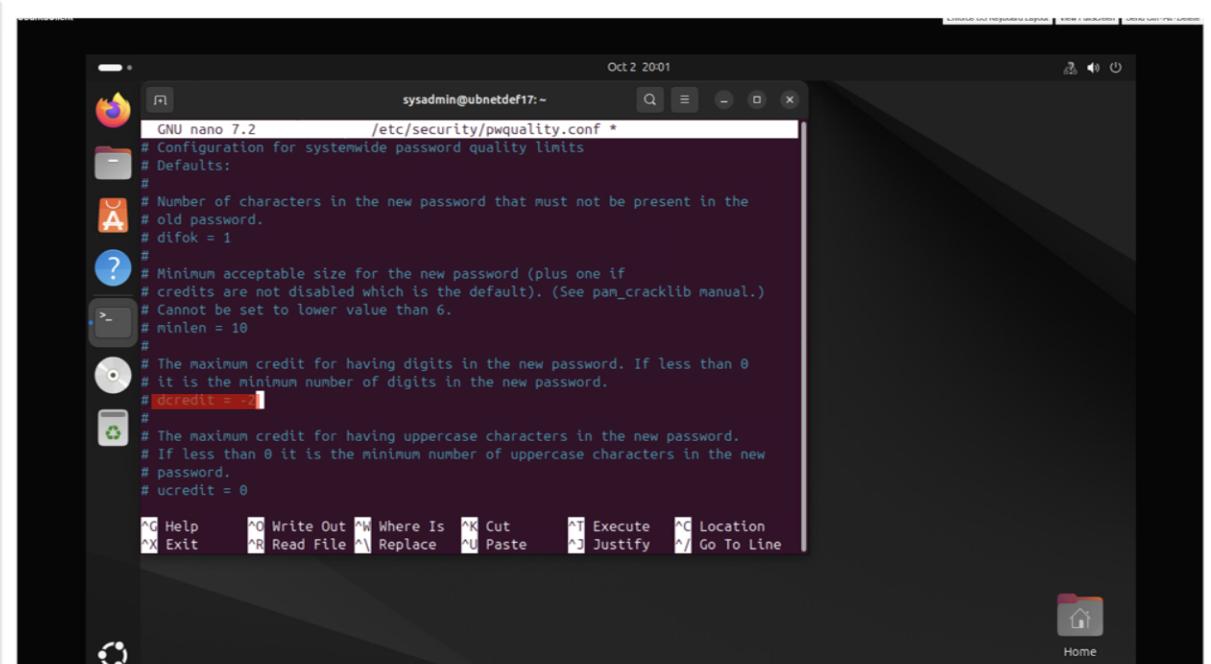


```
GNU nano 7.2          /etc/security/pwquality.conf *
# Configuration for systemwide password quality limits
#
# Defaults:
#
# Number of characters in the new password that must not be present in the
# old password.
# difok = 1
#
# Minimum acceptable size for the new password (plus one if
# credits are not disabled which is the default). (See pam_cracklib manual.)
# Cannot be set to lower value than 6.
# minlen = 10
#
# The maximum credit for having digits in the new password. If less than 0
# it is the minimum number of digits in the new password.
# dcredit = 0
#
# The maximum credit for having uppercase characters in the new password.
# If less than 0 it is the minimum number of uppercase characters in the new
# password.
# ucredit = 0

^O Help      ^O Write Out  ^W Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File  ^\ Replace   ^U Paste    ^J Justify  ^/ Go To Line
```

Figure 35: Screenshot of changing value of “minlen to 10” to accept minimum length.

- After that we modify value of “dcredit= -2” to accept password with atleast two digits as shown in Figure 36.

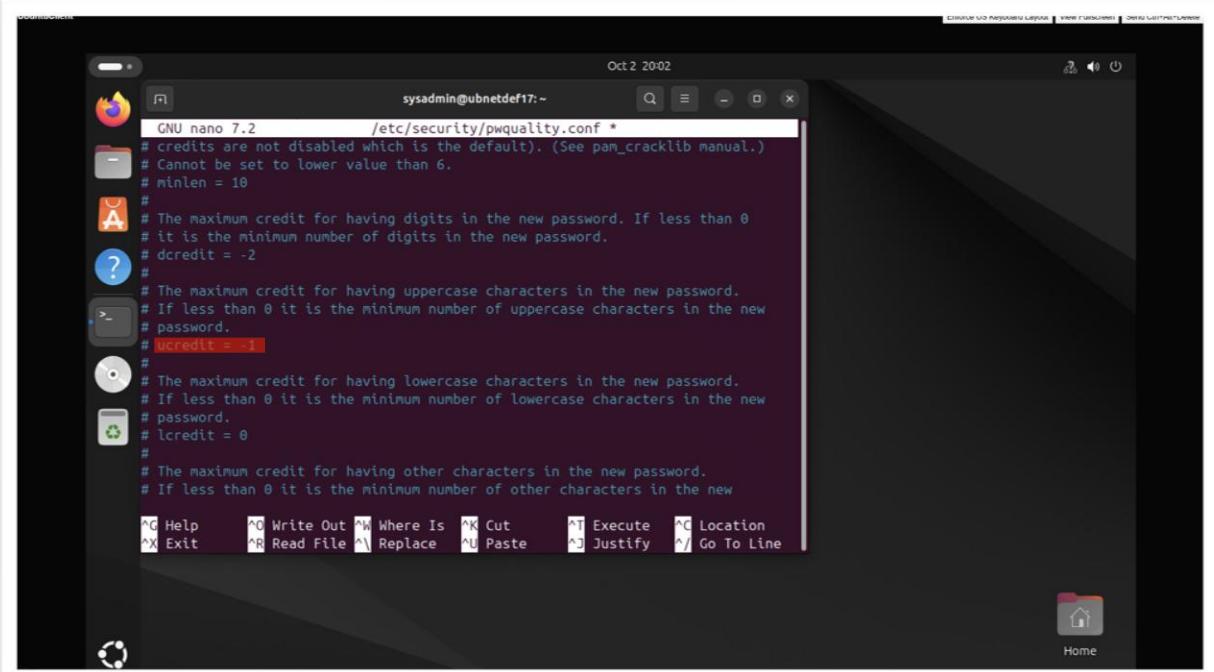


```
GNU nano 7.2          /etc/security/pwquality.conf *
# Configuration for systemwide password quality limits
#
# Defaults:
#
# Number of characters in the new password that must not be present in the
# old password.
# difok = 1
#
# Minimum acceptable size for the new password (plus one if
# credits are not disabled which is the default). (See pam_cracklib manual.)
# Cannot be set to lower value than 6.
# minlen = 10
#
# The maximum credit for having digits in the new password. If less than 0
# it is the minimum number of digits in the new password.
# dcredit = -2
#
# The maximum credit for having uppercase characters in the new password.
# If less than 0 it is the minimum number of uppercase characters in the new
# password.
# ucredit = 0

^O Help      ^O Write Out  ^W Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File  ^\ Replace   ^U Paste    ^J Justify  ^/ Go To Line
```

Figure 36: Screenshot of changing value of “dcredit to -2” to accept two digit.

- Finally, we modify value of “ucredit= -1” to accept passwords with atleast one upper case letter as shown in Figure 37.



The screenshot shows a terminal window titled "sysadmin@ubnetdef17:~" running the nano text editor on a dark-themed desktop environment. The file being edited is "/etc/security/pwquality.conf". The content of the file includes configuration for password quality, such as minimum length ("minlen = 10"), maximum digit credit ("dcredit = -2"), and maximum uppercase character credit ("ucredit = -1"). The "ucredit = -1" line is specifically highlighted with a red rectangle. The terminal window has a standard Linux-style menu bar at the top and a command-line interface at the bottom with various keyboard shortcuts.

Figure 37: Screenshot of changing value of “ucredit to -1” to accept atleast one UpperCase.

5. Updated Topology

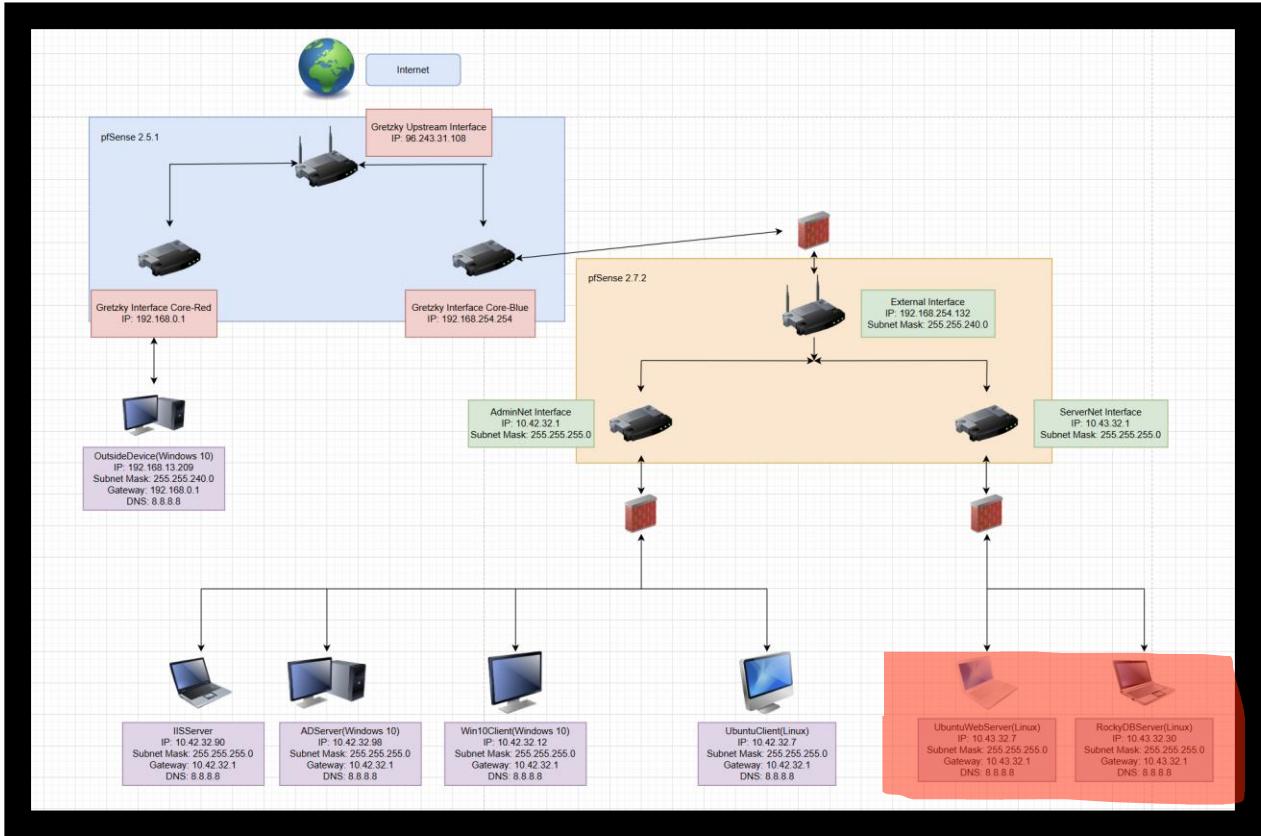
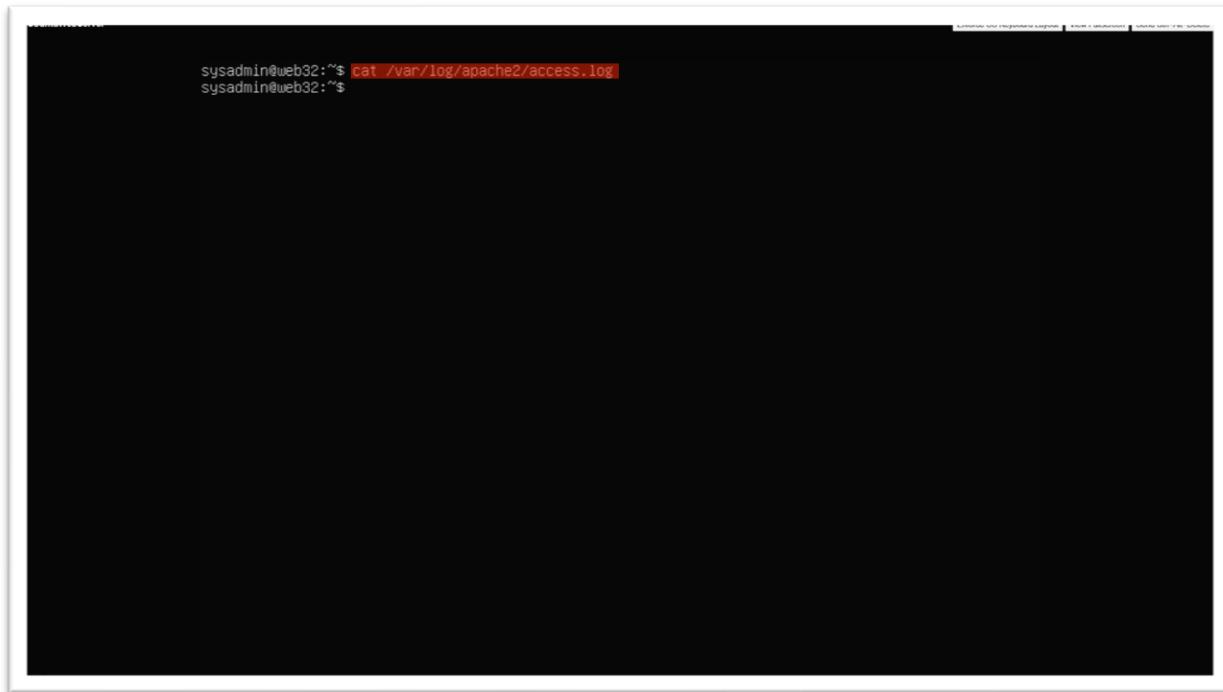


Figure 38: Screenshot of Updated Topology

- Two new devices “UbuntuWebServer” and “RockyBDServer” are added to the ServerNet Interface (as shown in Figure 38).
- One more new device “OutsideDevice” is connected in different network to “Gretzky Core-Red”
- Now, we have two different network interfaces,
 - One is “Blue” which is main system to connect every device from old topology
 - This one is “Red” which connects “OutsideDevice” to the network as shown in Figure 38.
- Because of two different network systems, there are three Gretzky Interface- Red, Blue and Upstream.

6. EAS 595 Additional Tasks

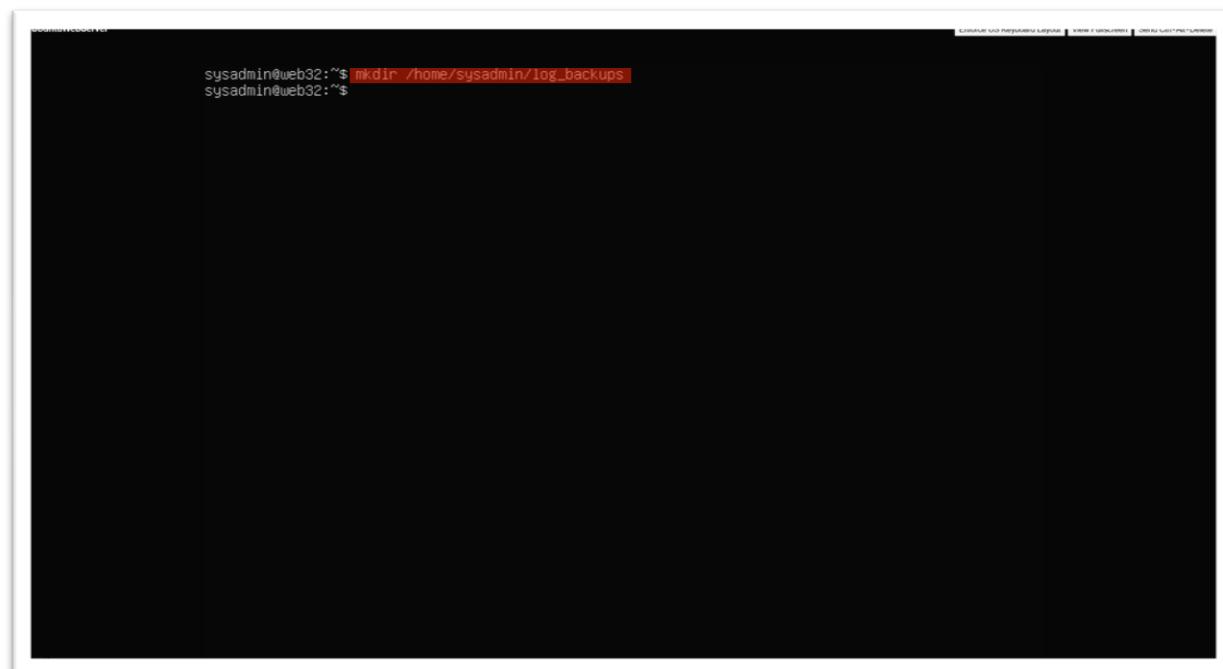
- Now type “cat /var/log/apache2/access.log” to verify that access logs exist as shown in Figure 39. And from same figure we can see that there is an access log that exist but its empty as no logs exists for now.



```
sysadmin@web32:~$ cat /var/log/apache2/access.log
sysadmin@web32:~$
```

Figure 39: Screenshot of “cat /var/log/apache2/access.log” to verify access logs.

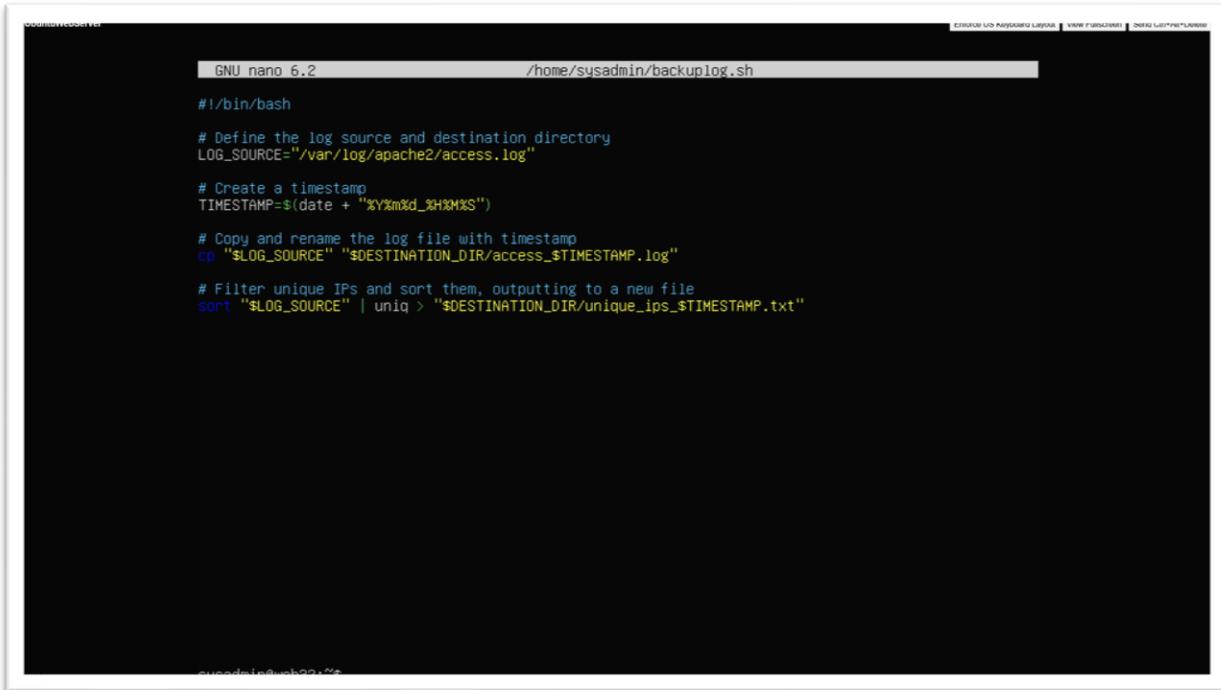
- Now we create a directory to store logs in the sysadmin user's home directory. By entering command “mkdir /home/sysadmin/log_backups”



```
sysadmin@web32:~$ mkdir /home/sysadmin/log_backups
sysadmin@web32:~$
```

Figure 40: Screenshot of “`mkdir /home/sysadmin/log_backups`” to create a directory in user’s home.

- After that we enter “`nano /home/sysadmin/backuplog.sh`” to create a script named `backuplog` and edit the script to make it like as shown in Figure 41.



The screenshot shows a terminal window titled "gnome-terminal" with the command "GNU nano 6.2" and the file path "/home/sysadmin/backuplog.sh". The script content is as follows:

```
#!/bin/bash

# Define the log source and destination directory
LOG_SOURCE="/var/log/apache2/access.log"

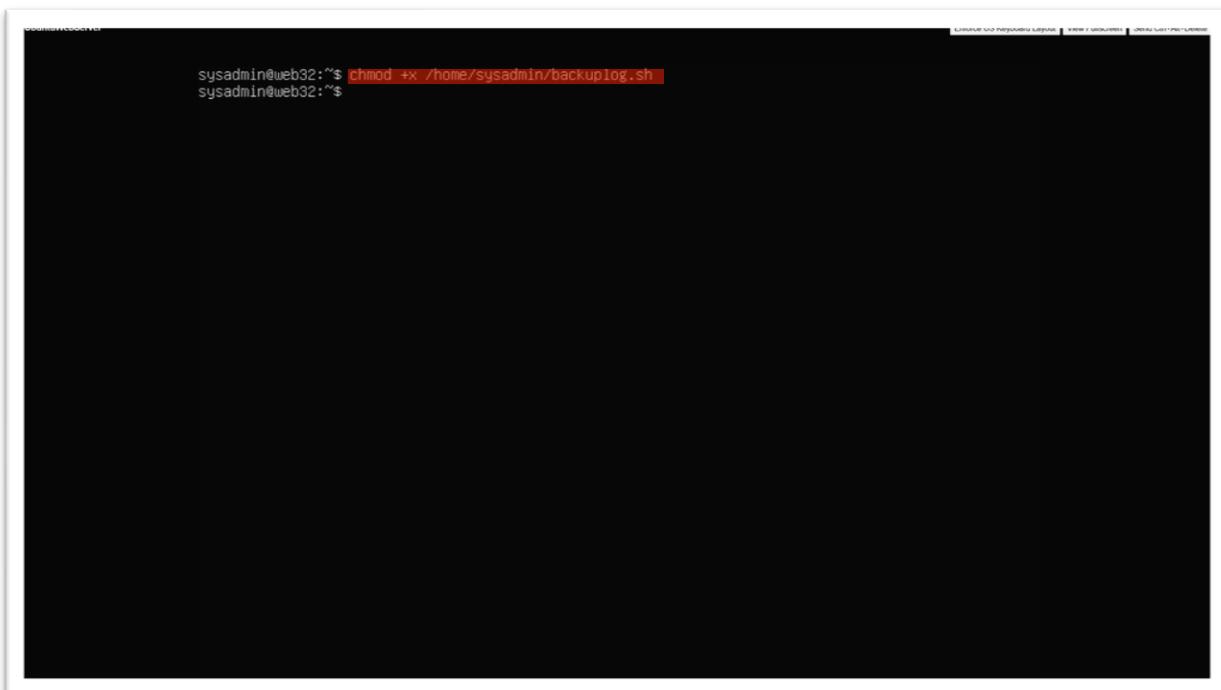
# Create a timestamp
TIMESTAMP=$(date + "%Y%m%d_%H%M%S")

# Copy and rename the log file with timestamp
cp "$LOG_SOURCE" "$DESTINATION_DIR/access_$TIMESTAMP.log"

# Filter unique IPs and sort them, outputting to a new file
sort "$LOG_SOURCE" | uniq > "$DESTINATION_DIR/unique_ips_$TIMESTAMP.txt"
```

Figure 41: Screenshot of editing the script for `backuplog`.

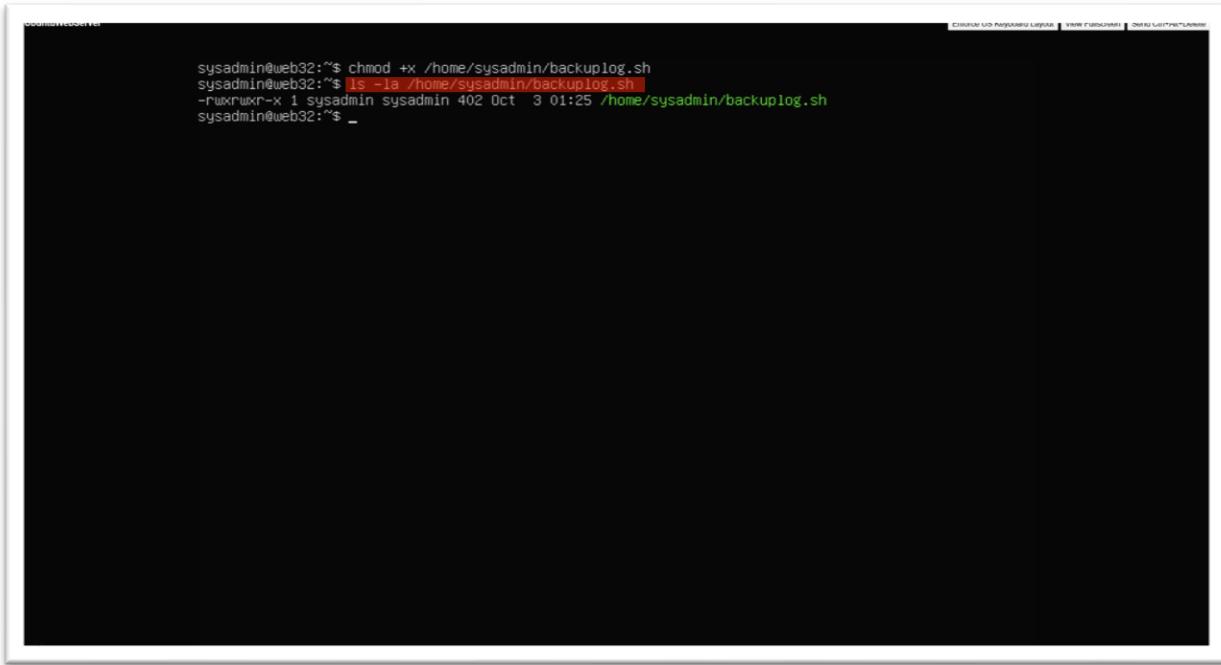
- Now we get proper permissions to that script using “`chmod +x /home/sysadmin/backuplog.sh`” as highlighted in Figure 42.



The screenshot shows a terminal window titled "gnome-terminal" with the command "sysadmin@web32:~\$ chmod +x /home/sysadmin/backuplog.sh". The output shows the command was successful.

Figure 42: Screenshot of “chmod +x /home/sysadmin/backuplog.sh” to give permissions to the script.

- Now we can verify the permissions by using command “ls -la /home/sysadmin/backuplog.sh” as highlighted below in Figure 43 and as we can observe the output consists of “-rwxrwxr-x” which proves the permissions given to that script.



```
sysadmin@web32:~$ chmod +x /home/sysadmin/backuplog.sh
sysadmin@web32:~$ ls -la /home/sysadmin/backuplog.sh
-rwxrwxr-x 1 sysadmin sysadmin 402 Oct  3 01:25 /home/sysadmin/backuplog.sh
sysadmin@web32:~$ _
```

Figure 43: Screenshot of “ls -la /home.sysadmin/backuplog.sh” to verify permissions for script.

- As we want to edit Cron Job, we type “crontab -e” (choose nano if asked so after entering that command) as highlighted in Figure 44.

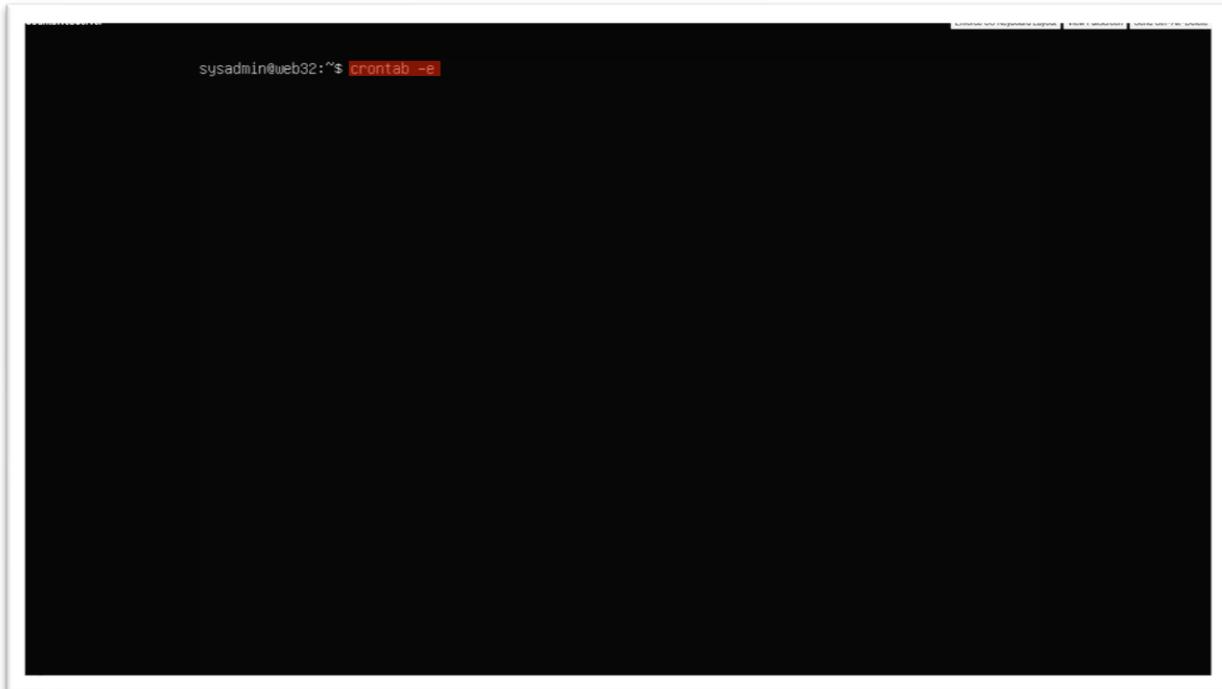


Figure 44: Screenshot of editing Cron Job using “crontab -e”.

- Now enter a new line at the end of nano- “5 4 * * * /home/sysadmin/backuplog.sh” to schedule the script to run daily at 4:05 AM as highlighted below.

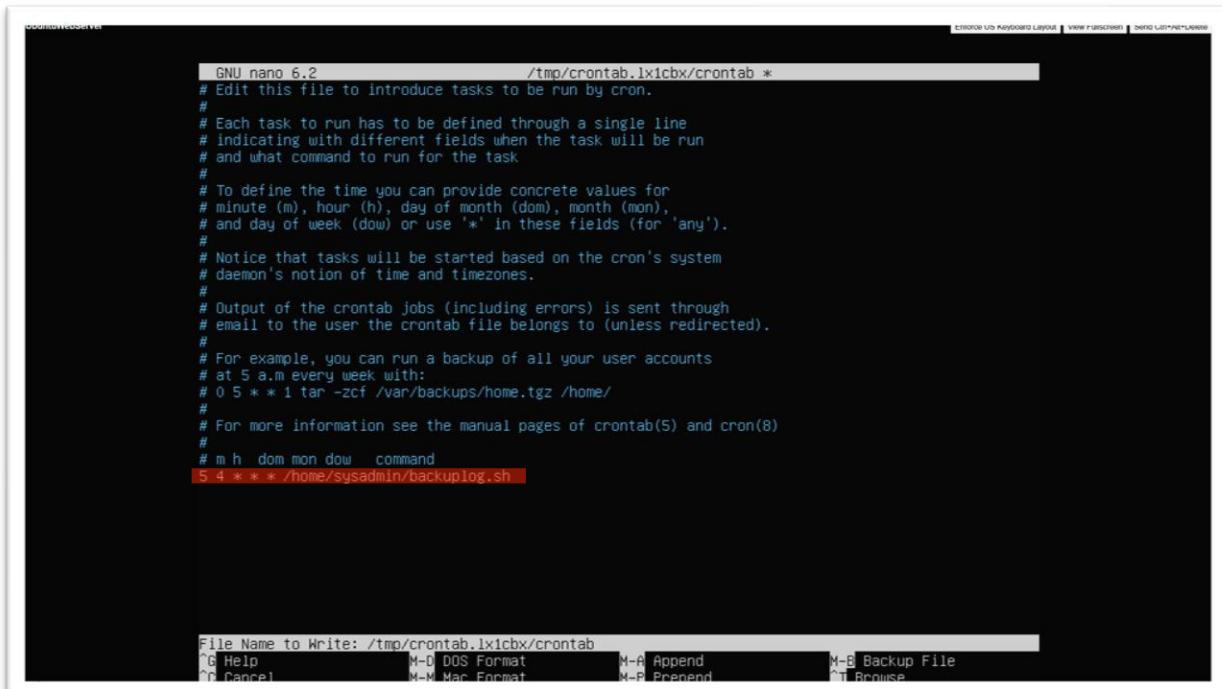
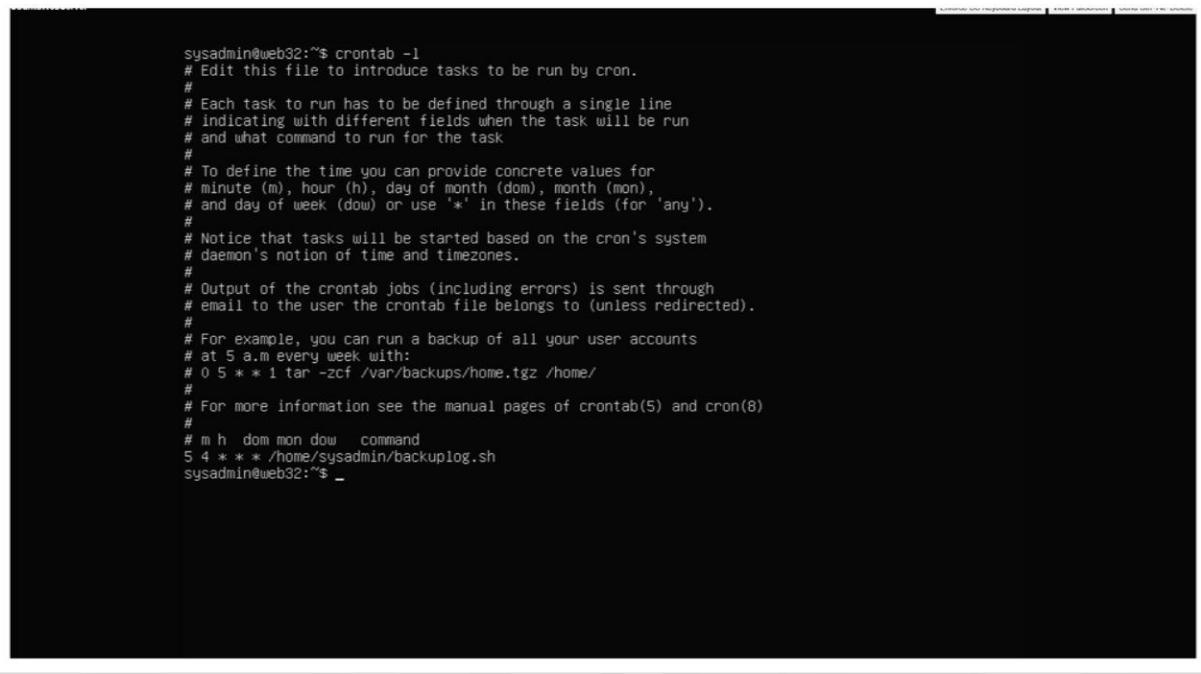


Figure 45: Screenshot of entering a new line “5 4 * * * /home/sysadmin/backuplog.sh” to make script run daily at 4:05 AM.

- After that, we check the list by inputting “crontab -l” and the final result as shown in Figure 46.



```
sysadmin@web32:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
5 4 * * * /home/sysadmin/backuplog.sh
sysadmin@web32:~$ _
```

Figure 46: Screenshot of Final Result by inputting “crontab -l”.