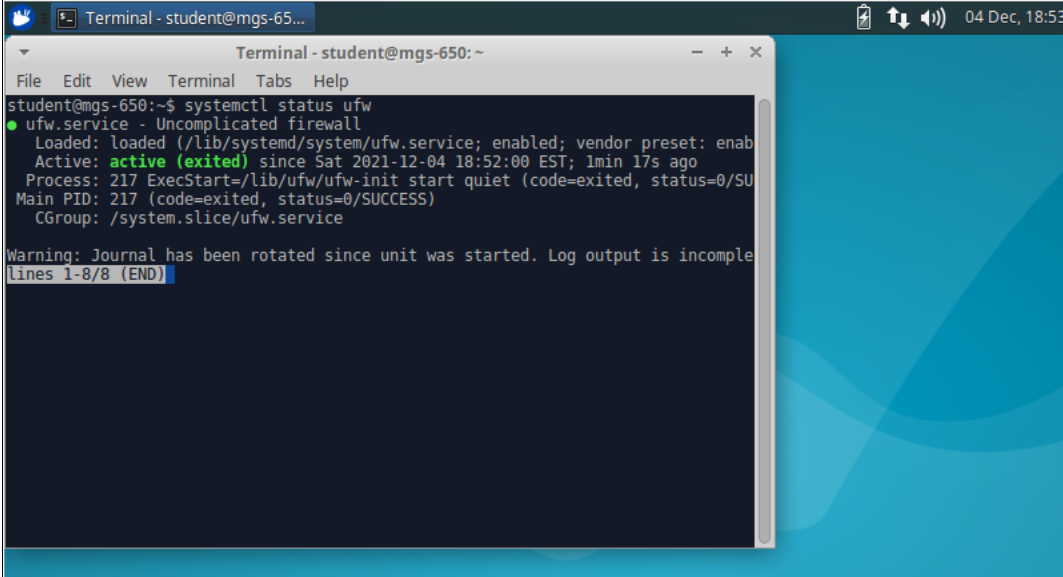


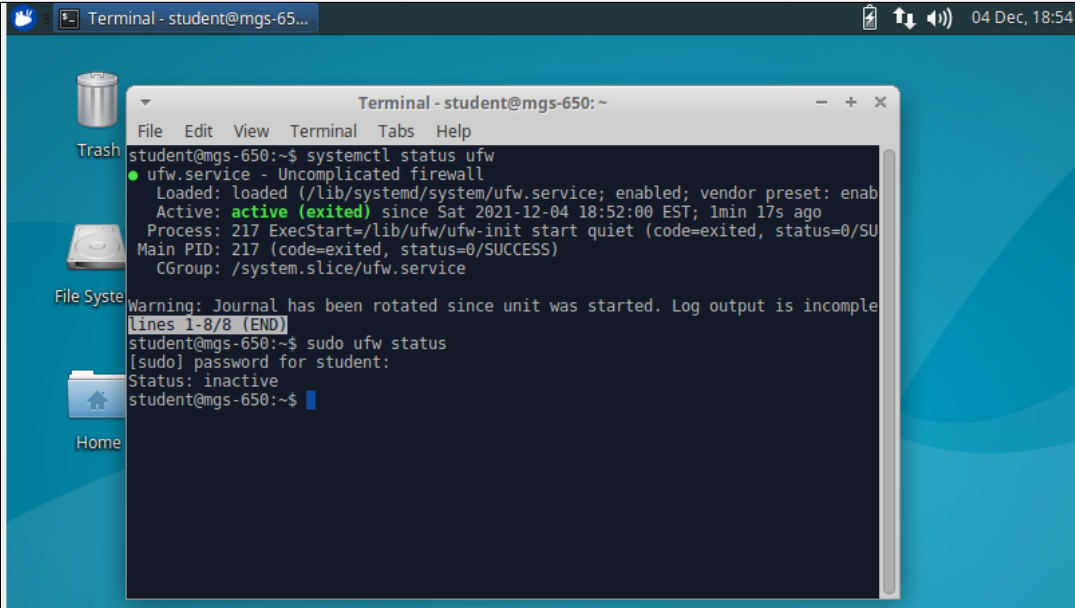
MGS 650 Information Assurance
Lab 5: System Hardening
Submitted by: Akhilesh Anand Undralla
12/06/2021



```
Terminal - student@mgs-650: ~
File Edit View Terminal Tabs Help
student@mgs-650:~$ systemctl status ufw
● ufw.service - Uncomplicated firewall
   Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor preset: enab
   Active: active (exited) since Sat 2021-12-04 18:52:00 EST; 1min 17s ago
   Process: 217 ExecStart=/lib/ufw/ufw-init start quiet (code=exited, status=0/SU
   Main PID: 217 (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/ufw.service

Warning: Journal has been rotated since unit was started. Log output is incomple
lines 1-8/8 (END)
```

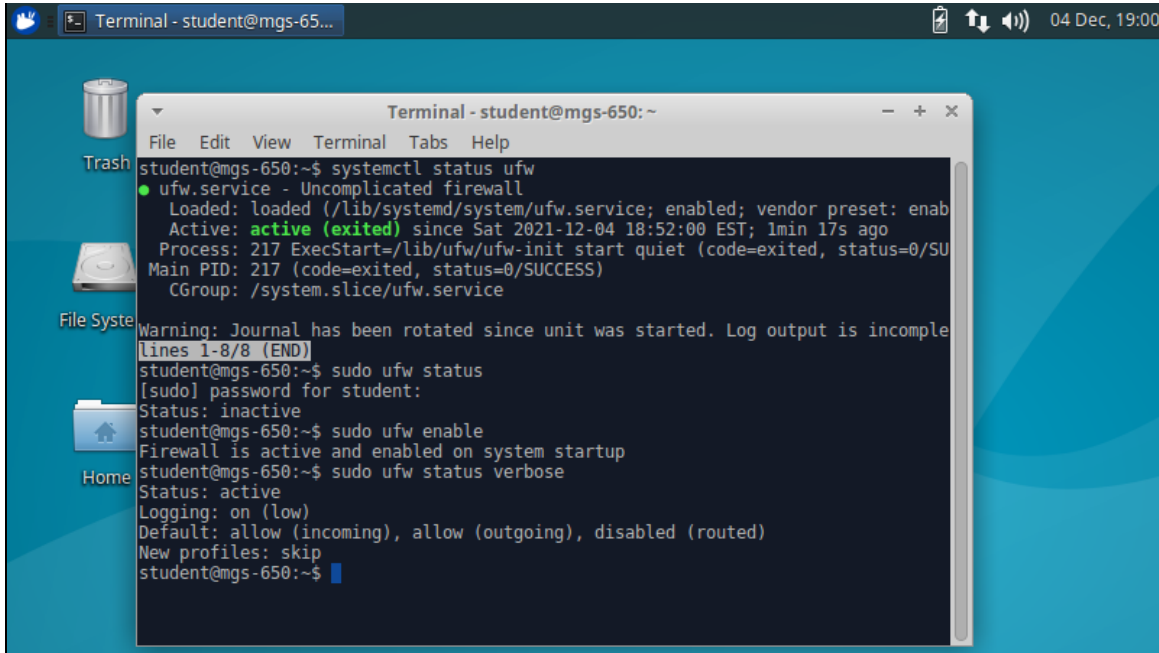
Figure 1.0: checking status of service using the command **systemctl status ufw**



```
Terminal - student@mgs-650: ~
File Edit View Terminal Tabs Help
student@mgs-650:~$ systemctl status ufw
● ufw.service - Uncomplicated firewall
   Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor preset: enab
   Active: active (exited) since Sat 2021-12-04 18:52:00 EST; 1min 17s ago
   Process: 217 ExecStart=/lib/ufw/ufw-init start quiet (code=exited, status=0/SU
   Main PID: 217 (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/ufw.service

Warning: Journal has been rotated since unit was started. Log output is incomple
lines 1-8/8 (END)
student@mgs-650:~$ sudo ufw status
[sudo] password for student:
Status: inactive
student@mgs-650:~$
```

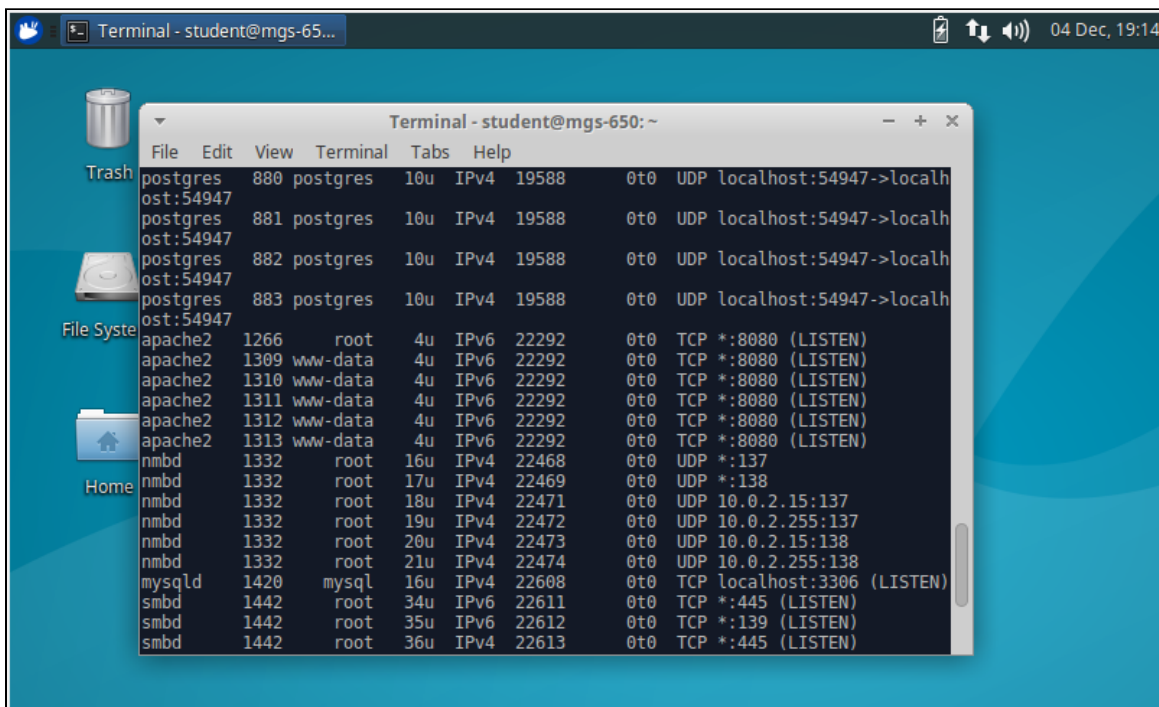
Figure 1.1: checking status of service and firewall using the command **systemctl status ufw** and **sudo ufw status**

A terminal window titled "Terminal - student@mgs-650: ~" is open on a desktop. The terminal shows the output of the command `systemctl status ufw`, which indicates that the ufw service is loaded and active (exited). The user then runs `sudo ufw status`, which shows the firewall is inactive. Finally, the user runs `sudo ufw enable`, which successfully enables the firewall. The terminal output is as follows:

```
student@mgs-650:~$ systemctl status ufw
● ufw.service - Uncomplicated firewall
   Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor preset: enab
   Active: active (exited) since Sat 2021-12-04 18:52:00 EST; 1min 17s ago
   Process: 217 ExecStart=/lib/ufw/ufw-init start quiet (code=exited, status=0/SU
   Main PID: 217 (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/ufw.service

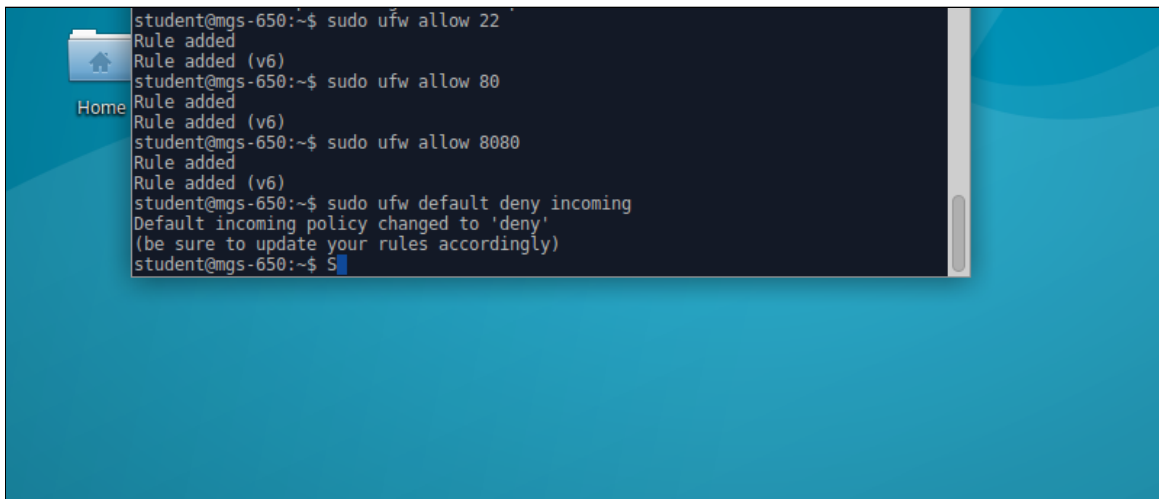
Warning: Journal has been rotated since unit was started. Log output is incomplete
lines 1-8/8 (END)
student@mgs-650:~$ sudo ufw status
[sudo] password for student:
Status: inactive
student@mgs-650:~$ sudo ufw enable
Firewall is active and enabled on system startup
student@mgs-650:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: allow (incoming), allow (outgoing), disabled (routed)
New profiles: skip
student@mgs-650:~$
```

Figure 1.2: Enabling the firewall using `sudo ufw enable` and checking the status with the command `sudo ufw status verbose`

A terminal window titled "Terminal - student@mgs-650: ~" is open on a desktop. The terminal shows the output of the command `sudo lsof -i -P`, which lists all currently running processes that are using network ports. The output is a table with columns for command, PID, user, file type, protocol, local address, and remote address. The data is as follows:

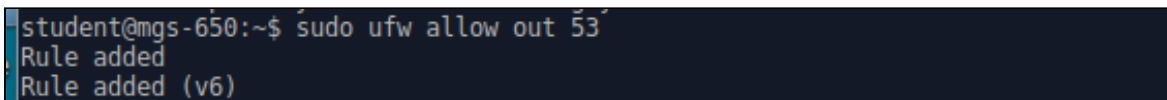
command	PID	user	file type	protocol	local address	remote address
postgres	880	postgres	10u	IPv4	19588	0t0 UDP localhost:54947->localh
postgres	881	postgres	10u	IPv4	19588	0t0 UDP localhost:54947->localh
postgres	882	postgres	10u	IPv4	19588	0t0 UDP localhost:54947->localh
postgres	883	postgres	10u	IPv4	19588	0t0 UDP localhost:54947->localh
postgres	1266	root	4u	IPv6	22292	0t0 TCP *:8080 (LISTEN)
postgres	1309	www-data	4u	IPv6	22292	0t0 TCP *:8080 (LISTEN)
postgres	1310	www-data	4u	IPv6	22292	0t0 TCP *:8080 (LISTEN)
postgres	1311	www-data	4u	IPv6	22292	0t0 TCP *:8080 (LISTEN)
postgres	1312	www-data	4u	IPv6	22292	0t0 TCP *:8080 (LISTEN)
postgres	1313	www-data	4u	IPv6	22292	0t0 TCP *:8080 (LISTEN)
nmdb	1332	root	16u	IPv4	22468	0t0 UDP *:137
nmdb	1332	root	17u	IPv4	22469	0t0 UDP *:138
nmdb	1332	root	18u	IPv4	22471	0t0 UDP 10.0.2.15:137
nmdb	1332	root	19u	IPv4	22472	0t0 UDP 10.0.2.255:137
nmdb	1332	root	20u	IPv4	22473	0t0 UDP 10.0.2.15:138
nmdb	1332	root	21u	IPv4	22474	0t0 UDP 10.0.2.255:138
mysqld	1420	mysql	16u	IPv4	22608	0t0 TCP localhost:3306 (LISTEN)
smbd	1442	root	34u	IPv6	22611	0t0 TCP *:445 (LISTEN)
smbd	1442	root	35u	IPv6	22612	0t0 TCP *:139 (LISTEN)
smbd	1442	root	36u	IPv4	22613	0t0 TCP *:445 (LISTEN)

Figure 1.3: `sudo lsof -i -P` command shows currently-running programs that are using the network

A terminal window with a blue background and a 'Home' icon on the left. The terminal shows the following commands and output:

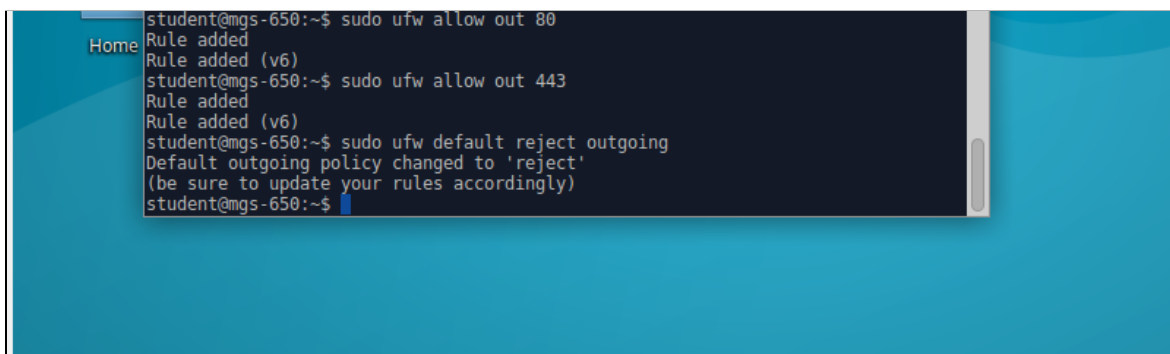
```
student@mgs-650:~$ sudo ufw allow 22
Rule added
Rule added (v6)
student@mgs-650:~$ sudo ufw allow 80
Rule added
Rule added (v6)
student@mgs-650:~$ sudo ufw allow 8080
Rule added
Rule added (v6)
student@mgs-650:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
student@mgs-650:~$ S
```

Figure 1.4: Firewall rule that allows inbound SSH traffic (22), web services(80 & 8080) with `sudo ufw allow 22` and default policy to deny incoming traffic using `sudo ufw default deny incoming`

A terminal window showing the following commands and output:

```
student@mgs-650:~$ sudo ufw allow out 53
Rule added
Rule added (v6)
```

Figure 1.5: Allow the three ports 53 for outgoing communications with `sudo ufw allow out 53`

A terminal window with a blue background and a 'Home' icon on the left. The terminal shows the following commands and output:

```
student@mgs-650:~$ sudo ufw allow out 80
Rule added
Rule added (v6)
student@mgs-650:~$ sudo ufw allow out 443
Rule added
Rule added (v6)
student@mgs-650:~$ sudo ufw default reject outgoing
Default outgoing policy changed to 'reject'
(be sure to update your rules accordingly)
student@mgs-650:~$
```

Figure 1.6: Allow the ports (80, 443) for outgoing communications and default policy to reject remaining with `sudo ufw default reject outgoing`

```
(Be sure to update your rules accordingly)
student@mgs-650:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), reject (outgoing), disabled (routed)
New profiles: skip

To Action From
--
22 ALLOW IN Anywhere
80 ALLOW IN Anywhere
8080 ALLOW IN Anywhere
22 (v6) ALLOW IN Anywhere (v6)
80 (v6) ALLOW IN Anywhere (v6)
8080 (v6) ALLOW IN Anywhere (v6)

53 ALLOW OUT Anywhere
80 ALLOW OUT Anywhere
443 ALLOW OUT Anywhere
53 (v6) ALLOW OUT Anywhere (v6)
80 (v6) ALLOW OUT Anywhere (v6)
443 (v6) ALLOW OUT Anywhere (v6)

student@mgs-650:~$
```

Figure 1.7: Current Firewall status after setting network rules (verbose)

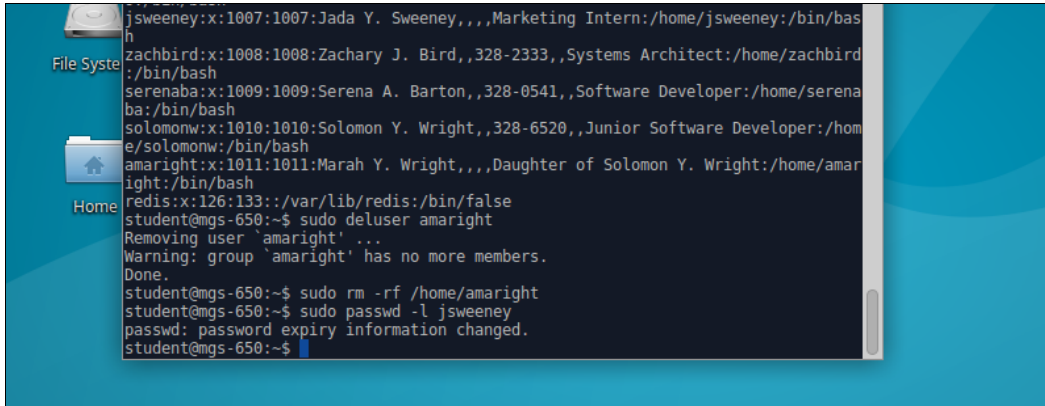
```
student@mgs-650:~$ sudo apt purge samba
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  attr liblvm4.0 linux-headers-4.10.0-28 linux-headers-4.10.0-28-generic
  linux-image-4.10.0-28-generic linux-image-extra-4.10.0-28-generic
  python-dnspython samba-dsdb-modules samba-vfs-modules snapd-login-service
  tdb-tools
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  samba*
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 11.6 MB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 240190 files and directories currently installed.)
Removing samba (2:4.3.11+dfsg-0ubuntu0.16.04.34) ...
Purging configuration files for samba (2:4.3.11+dfsg-0ubuntu0.16.04.34) ...
Processing triggers for libc-bin (2.23-0ubuntu1.3) ...
Processing triggers for man-db (2.7.5-1) ...
student@mgs-650:~$
```

Figure 1.8: Disabling smbd as a running service using `sudo apt purge samba`

```
student@mgs-650:~$ systemctl status vsftpd
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset: e
   Active: active (running) since Sat 2021-12-04 18:52:05 EST; 53min ago
   Main PID: 784 (vsftpd)
   CGroup: /system.slice/vsftpd.service
           └─784 /usr/sbin/vsftpd /etc/vsftpd.conf


Dec 04 18:52:05 mgs-650 systemd[1]: Starting vsftpd FTP server...
Dec 04 18:52:05 mgs-650 systemd[1]: Started vsftpd FTP server.
lines 1-9/9 (END)
student@mgs-650:~$ sudo systemctl disable vsftpd
Synchronizing state of vsftpd.service with SysV init with /lib/systemd/systemd-s
ysv-install...
Executing /lib/systemd/systemd-sysv-install disable vsftpd
inserv: warning: current start runlevel(s) (empty) of script 'vsftpd' overrides
LSB defaults (2 3 4 5).
inserv: warning: current stop runlevel(s) (0 1 2 3 4 5 6) of script 'vsftpd' ov
errides LSB defaults (0 1 6).
student@mgs-650:~$
```

Figure 1.9: Disabling vsftpd as a running service using `sudo apt purge vsftpd`



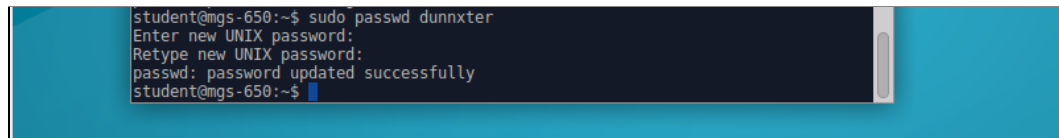
```
jsweeney:x:1007:1007:Jada Y. Sweeney,,Marketing Intern:/home/jsweeney:/bin/bash
h
zachbird:x:1008:1008:Zachary J. Bird,,328-2333,,Systems Architect:/home/zachbird
:/bin/bash
serenaba:x:1009:1009:Serena A. Barton,,328-0541,,Software Developer:/home/serena
ba:/bin/bash
solomonw:x:1010:1010:Solomon Y. Wright,,328-6520,,Junior Software Developer:/hom
e/solomonw:/bin/bash
amaright:x:1011:1011:Marah Y. Wright,,Daughter of Solomon Y. Wright:/home/amar
ight:/bin/bash
redis:x:126:133:/:/var/lib/redis:/bin/false
student@mgs-650:~$ sudo deluser amaright
Removing user 'amaright' ...
Warning: group 'amaright' has no more members.
Done.
student@mgs-650:~$ sudo rm -rf /home/amaright
student@mgs-650:~$ sudo passwd -l jsweeney
passwd: password expiry information changed.
student@mgs-650:~$
```

Figure 2.0: Deleting user account with `sudo deluser amaright`, and deleting home folder with `sudo rm -rf /home/amaright`



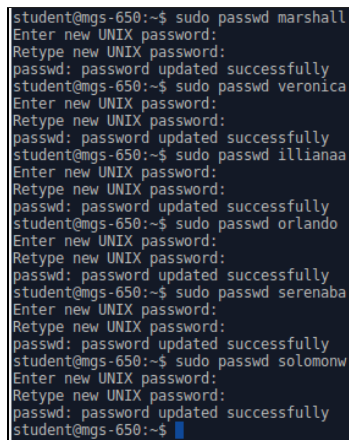
```
sshd:*:17495:0:99999:7:::
mysql:!17495:0:99999:7:::
postgres:*:17495:0:99999:7:::
ftp:*:17495:0:99999:7:::
patcpett:$6$5oG72HSq$Nfpfw1xLCJvUR1Ux8Kvrf.dv8e0rxJFS0dFYMiuB2Pj/28wmSYHFNUhtq
fcA/Mh129s4Y6u6bpvt0jN24I91:17496:0:99999:7:::
dunnxter:17496:0:99999:7:::
illianaa:17496:0:99999:7:::
marshall:17496:0:99999:7:::
veronica:17496:0:99999:7:::
orlando:17496:0:99999:7:::
jsweeney:!$6$Ctw7AgVG$A55jo8op7U0Na6N5nT5hiJBo3TIks.GQhXw0F65VKrDIjISNGAFj89qr7J
XH5NLFPMR0WIMf0uIFuYt/tAbJ/0:17498:0:99999:7:::
zachbird:$6$ngiAx0b0$jZg/ToBRDID3LVilVhQCGPX//TMupnJDMHSMffuIS9FL8EsH5iYChZZiYbi
2RlKvYwan0rHtLLKgdJDWq4Ban0:17496:0:99999:7:::
serenaba:17496:0:99999:7:::
solomonw:17496:0:99999:7:::
redis:*:17496:0:99999:7:::
student@mgs-650:~$
```

Figure 2.1: Checking password hashes stored in the `/etc/shadow` file



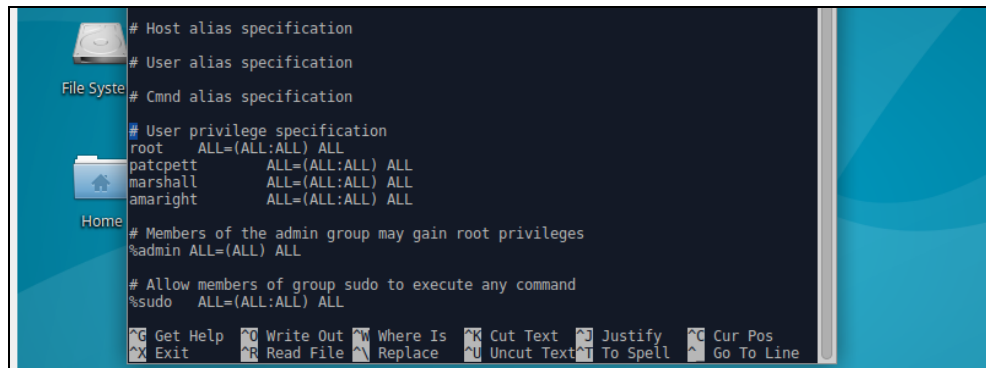
```
student@mgs-650:~$ sudo passwd dunnxter
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
student@mgs-650:~$
```

Figure 2.2: Setting password for user who don't have a password using `sudo passwd dunnxter`



```
student@mgs-650:~$ sudo passwd marshall
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
student@mgs-650:~$ sudo passwd veronica
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
student@mgs-650:~$ sudo passwd illianaa
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
student@mgs-650:~$ sudo passwd orlando
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
student@mgs-650:~$ sudo passwd serenaba
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
student@mgs-650:~$ sudo passwd solomonw
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
student@mgs-650:~$
```

Figure 2.3: Setting password for users who don't have a password.



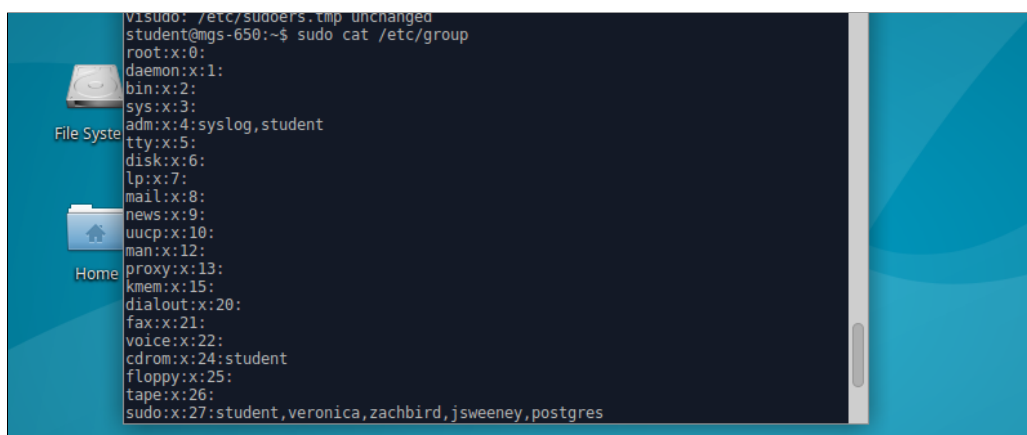
```
# Host alias specification
# User alias specification
# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
patcpett    ALL=(ALL:ALL) ALL
marshall    ALL=(ALL:ALL) ALL
amaright    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin    ALL=(ALL) ALL

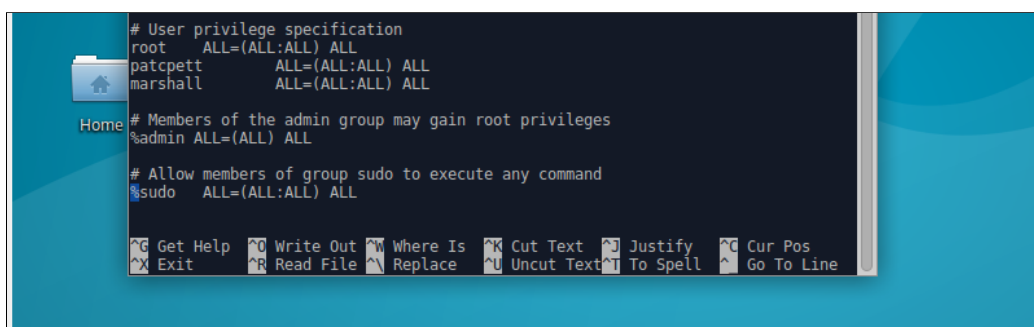
# Allow members of group sudo to execute any command
%sudo    ALL=(ALL:ALL) ALL
```

Figure 2.4: Checking user who can execute commands as root via sudo using `sudo visudo`



```
visudo: /etc/sudoers.tmp unchanged
student@mgs-650:~$ sudo cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,student
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:student
floppy:x:25:
tape:x:26:
sudo:x:27:student,veronica,zachbird,jsweeney,postgres
```

Figure 2.5: Checking users in each group using `sudo cat /etc/group` file

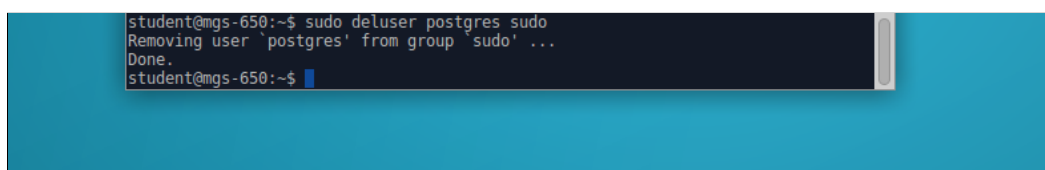


```
# User privilege specification
root    ALL=(ALL:ALL) ALL
patcpett    ALL=(ALL:ALL) ALL
marshall    ALL=(ALL:ALL) ALL

# 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
```

Figure 2.6: Checking sudo users details after removing `amaright` from sudoers file



```
student@mgs-650:~$ sudo deluser postgres sudo
Removing user 'postgres' from group 'sudo' ...
Done.
student@mgs-650:~$
```

Figure 2.7: Removing postgres user from the sudo group with `sudo deluser postgres sudo`

Executive Summary

Initial steps taken before the security configuration assessment were to update all the software, configure firewall rules, remove unnecessary programs (samba, vsftpd), check user access and secure stored password hashes. This can partially and potentially can help towards system hardening by minimising attack vectors such as outdated packages with known security vulnerabilities, unnecessary network access [Figure 1.1 - 1.7], unnecessary running/installed programs [Figure 1.8 & 1.9], unrestricted user access [Figure 2.0, 2.2 - 2.7] and unencrypted data respectively [Figure 2.1].

The security configuration assessment using Center for Internet Security (CIS) Critical Security Controls (CSC) framework was performed to assess mgs650's internal network security posture. The CIS benchmark report generated has various configuration recommendations that applied for system hardening and auditing a Linux system. Overall completion of the assessment indicated recommendations for each control with scoring details for all the 6 controls defined for CIS Ubuntu Linux 16.04 LTS Benchmark v1.0.0. It is one of the CIS Controls that recommends secure configurations for hardware and software on mobile devices, laptops, workstations, and servers. The Center for Internet Security Configuration Assessment Tool (CIS-CAT) suggests security configuration benchmarks distributed by CIS as well as NIST under the Security Content Automation Protocol (SCAP) program, as an initiative to enable automation and standardization of technical security operations.

After the scan, six recommendation categories generated falls under the following configuration baselines:

- Initial Setup
- Services
- Network Configuration
- Logging and Auditing
- Access, Authentication and Authorization
- System Maintenance

For this scan, a Level 1 profile is chosen which is intended for servers and provides a practical and prudent way to secure a system within acceptable technology means and not too much performance impact.

Non-native file system types such as freevxfs, hfs, jffs32 FAT, hfsplus, squashfs, udf are supported under Linux's in-built functionality. Steps must be taken to ensure a check list of filesystems that are needed after considering the system environment since Internet access to cloud storage and standard network connectivity may use non-standard file system formats. Next, if a filesystem is not needed remove/disable support for unneeded file system types. This reduces the local attack surface of the system. An alerting system such as Advanced Intrusion Detection Environment (AIDE) must be set up to detect unauthorized changes to current file state/configuration files against a snapshot and prevent accidental or malicious misconfigurations. Steps must be taken to ensure permissions on bootloader config are configured preventing non-root users to read and write the boot parameters. This will reduce exploits security upon boot. Additional process hardening recommendation after initial setup of the system is to set up a hard limit on core dumps to prevent exposing confidential information from a core file.

Inorder to ensure legitimate incoming traffic from other machines, specific port communications i.e., port 22, port 80, port 8080 are allowed [Figure 1.4] and all the remaining ports are denied/blocked by changing the default policy to 'deny'. Similarly, outgoing traffic rules [Figure 1.6] are modified to allow port 43, port 50, port 443 that specifically allow port communications from our machine to other machines. However, this negates a report result 3.6.5 that was needed to ensure firewall rules exist for all open ports. Hence, this risk is accepted in this scenario.

Although there are password hashes starting "*", "!" which are invalid password hashes and a user cannot normally log in with those accounts [Figure 2.1] (here mysql, sshd, postgres), it is also recommended that the shell field in the password file be set to /sbin/nologin. This will also prevent the account from potentially being used to run any commands by locking the 'sync', 'shutdown', and 'halt' users. These users are traditionally shipped with a null password in unix and linux distributions which are used to properly shut down a system without having to provide the root password. This is useful in the case of a desktop workstation, but is detrimental in the case of a server system (also, this machine is assumed operating as a web server) and creates a risk of Denial of Service attack. So, these accounts must be either removed or locked like other default accounts to prevent use.