Project Report

Project 2: Cloud Server Hardening & Secure Access

Student Details

Name: Rahul Issar
Semester: 5th
ERP: 6604685
Program: B.Tech CSE (Cybersecurity Specialization)

1. Objective

The objective of this project is to **deploy, secure, and harden a cloud-based Linux (Ubuntu) server** using standard cybersecurity best practices.
The project focuses on securing SSH access, applying strict firewall rules, configuring intrusion prevention mechanisms, and auditing all user activity to ensure accountability.

2. Environment Setup

Component	Description
Cloud Platform	Google Cloud Platform (Free Trial)
Instance Type	e2-micro VM (Always Free Eligible)
Operating System	Ubuntu 22.04 LTS
Tools & Utilities	ufw, fail2ban, auditd
Access Client	Windows PowerShell / GCP Browser SSH
User Created	adminuser (with sudo privileges)

3. Implementation Steps

Step 1 – System Update

- sudo apt update && sudo apt upgrade -y
- sudo apt install vim curl unzip htop -y

✅ Ensures latest security patches and packages are installed.

Step 2 – Create IAM-Like User

- sudo adduser adminuser
- sudo usermod -aG sudo adminuser
- echo "adminuser ALL=(ALL) ALL" | sudo tee /etc/sudoers.d/adminuser
- sudo chmod 440 /etc/sudoers.d/adminuser

✅ Creates a dedicated administrative user with restricted sudo privileges.

Step 3 – Enable SSH Key Authentication

- sudo mkdir -p /home/adminuser/.ssh
- echo "ADD YOUR KEY " | sudo tee /home/adminuser/.ssh/authorized_keys
- sudo chown -R adminuser:adminuser /home/adminuser/.ssh

- sudo chmod 700 /home/adminuser/.ssh
- sudo chmod 600 /home/adminuser/.ssh/authorized_keys

✅ Enforces password-less key-based authentication for secure SSH access.

Step 4 – Disable Root Login and Password SSH

- sudo sed -i 's/^#\?PermitRootLogin.*/PermitRootLogin no/' /etc/ssh/sshd_config
- sudo sed -i 's/^#\?PasswordAuthentication.*/PasswordAuthentication no/' /etc/ssh/sshd_config
- sudo systemctl restart ssh

✅ Prevents root login and disables password authentication to reduce attack surface.

Step 5 – Configure Firewall (UFW)

- sudo apt install ufw -y
- sudo ufw default deny incoming
- sudo ufw default allow outgoing
- sudo ufw allow OpenSSH
- sudo ufw enable
- sudo ufw status verbose

✅ Firewall activated – only SSH (port 22) is allowed; all other incoming traffic denied.

Step 6 – Install and Configure Fail2ban

- sudo apt install fail2ban -y
- sudo systemctl enable --now fail2ban
- sudo tee /etc/fail2ban/jail.d/sshd.local > /dev/null <<'EOF'
- [sshd]
- enabled = true
- port = ssh
- logpath = /var/log/auth.log
- maxretry = 3
- bantime = 600
- EOF
- sudo systemctl restart fail2ban
- sudo fail2ban-client status sshd

✅ Fail2ban monitors SSH logs, detects brute-force attempts, and temporarily bans offending IPs.

Step 7 – Enable Auditing (auditd)

- sudo apt install auditd audispd-plugins -y
- sudo systemctl enable --now auditd
- sudo tee /etc/audit/rules.d/50-project.rules > /dev/null <<'EOF'
- -w /var/log/auth.log -p wa -k logins
- -w /etc/sudoers -p wa -k sudo_config
- -w /etc/sudoers.d/ -p wa -k sudo_config
- -a always,exit -F arch=b64 -S execve -F euid=0 -k sudo_cmds
- EOF

- sudo augenrules --load
- sudo systemctl restart auditd
- sudo ausearch -k sudo_cmds | tail -n 10

✅ Auditd tracks all login attempts and sudo command usage for accountability.

---

4. Verification and Deliverables

# Screenshot Title	Command Executed	Expected Result
1 **Disabled Root Login**	`sudo cat /etc/ssh/sshd_config	grep -E 'PermitRootLogin`	
2 **UFW Active Status**	sudo ufw status verbose	Shows Status: active and OpenSSH ALLOW
3 **Fail2ban Ban Logs**	sudo fail2ban-client status sshd and sudo tail -n 15 /var/log/fail2ban.log	Shows banned IPs and ban events
4 **Auditd Log Sample**	`sudo ausearch -k logins	tail -n 10`	

---

5. Attached Screenshots

1. 🖼️ Disabled Root Login –

```
student-02-46746861e8d7@secure-ubuntu:~$ sudo cat /etc/ssh/sshd_config
#### Google OS Login control. Do not edit this section. ####
TrustedUserCAKeys /etc/ssh/oslogin_trustedca.pub
AuthorizedPrincipalsCommand /usr/bin/google_authorized_principals %u %k
AuthorizedPrincipalsCommandUser root
AuthorizedKeysCommand /usr/bin/google_authorized_keys
AuthorizedKeysCommandUser root
#### End Google OS Login control section. ####

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
KbdInteractiveAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the KbdInteractiveAuthentication and
# PasswordAuthentication.  Depending on your PAM configuration,
# PAM authentication via KbdInteractiveAuthentication may bypass
# the setting of "PermitRootLogin without-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and KbdInteractiveAuthentication to 'no'.
UsePAM yes

#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
PrintMotd no
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed
```

2. UFW Active Status –

```
student-02-46746861e8d7@secure-ubuntu:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22/tcp (OpenSSH)           ALLOW IN    Anywhere
22/tcp (OpenSSH (v6))      ALLOW IN    Anywhere (v6)
```

3. Fail2ban Ban Logs –

```
student-02-46746861e8d7@secure-ubuntu:~$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 3
|  |- Total failed:     6
|  `- File list:        /var/log/auth.log
`- Actions
   |- Currently banned: 1
   |- Total banned:     1
   `- Banned IP list:   152.59.62.99
```

4. Auditd Log Sample –

```
student-02-46746861e8d7@secure-ubuntu:~$ sudo ausearch -k sudo_cmds | tail -n 10
type=EXECVE msg=audit(1762292779.347:436): argc=4 a0="sudo" a1="ausearch" a2="-k" a3="sudo_cmds"
type=SYSCALL msg=audit(1762292779.347:436): arch=c000003e syscall=59 success=yes exit=0 a0=5da17d63f840 a1=5da17d63f880 a2=5da17d5efc70 a3=8 items=2 ppid=2403 pid=10993 auid=242948011 uid=242948011 gid=1000 euid=0 suid=0 fsuid=0 egid=1000 sgid=1000 fsgid=1000 tty=pts0 ses=2 comm="sudo" exe="/usr/bin/sudo" subj=unconfined key="sudo_cmds"
----
time->Tue Nov  4 21:46:19 2025
type=PROCTITLE msg=audit(1762292779.420:441): proctitle=617573656172636800 2D6B007375646F5F636D6473
type=PATH msg=audit(1762292779.420:441): item=1 name="/lib64/ld-linux-x86-64.so.2" inode=3175 dev=08:01 mode=0100755 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(1762292779.420:441): item=0 name="/usr/sbin/ausearch" inode=54569 dev=08:01 mode=0100755 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1762292779.420:441): cwd="/home/student-02-46746861e8d7"
type=EXECVE msg=audit(1762292779.420:441): argc=3 a0="ausearch" a1="-k" a2="sudo_cmds"
type=SYSCALL msg=audit(1762292779.420:441): arch=c000003e syscall=59 success=yes exit=0 a0=59ec37244f48 a1=59ec3723a9e0 a2=59ec37247100 a3=0 items=2 ppid=10995 pid=10996 auid=242948011 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts1 ses=2 comm="ausearch" exe="/usr/sbin/ausearch" subj=unconfined key="sudo_cmds"
```

---

6. Results and Observations

- Root and password-based SSH logins completely disabled.
- SSH key-based authentication working as expected.
- UFW firewall actively restricts incoming connections.

- Fail2ban detects and bans brute-force attack IPs.
- Auditd records logins and sudo command usage.

✅ **System is now secure, monitored, and hardened against unauthorized access.**

---

7. Conclusion

This project successfully demonstrates how to **secure a cloud-hosted Linux server** using open-source tools.
It provided hands-on experience in:

- Linux server hardening techniques
- SSH security and key management
- Host-based firewall configuration
- Intrusion prevention using Fail2ban
- Security auditing and log analysis with Auditd

The hardened server setup follows cybersecurity best practices for cloud infrastructure.

---

8. Student Details

Rahul Issar
Semester – 5th
ERP – 6604685
B.Tech CSE (Cybersecurity Specialization)

---

9. References

- Ubuntu Official Documentation – https://ubuntu.com/server/docs
- Fail2ban GitHub – https://github.com/fail2ban/fail2ban
- Auditd Documentation – https://linux.die.net/man/8/auditd
- Google Cloud Compute Engine Docs – https://cloud.google.com/compute/docs

---

Submitted By: Rahul Issar
Department of Computer Science and Engineering
Cybersecurity Specialization