LAB ASSIGNMENT

Securing a Database Server Environment

Objectives

By the end of this lab, you will be able to:

- Create and configure a virtual machine.
- Install and harden a minimal Debian Linux system.
- Set up a firewall using UFW with a least-privilege policy.
- Deploy Fail2ban to protect against brute-force attacks.
- Install and secure a PostgreSQL database server.

Part 1 — Virtual Machine Setup

Create a new virtual machine using VirtualBox (or VMware) with the following specifications:

Setting	Required value
RAM	2 GB minimum
CPU	2 virtual cores
Disk	20 GB, dynamic allocation
Network adapter	Bridged
Boot image	Debian 12 (Bookworm) minimal ISO

Tasks

1. Create the VM with the specifications above.
2. Boot from the Debian ISO and start the installation wizard.

Part 2 — Debian Linux Installation & Hardening

Perform a minimal Debian installation (no graphical interface) and apply basic hardening.

Installation guidelines

- Set the hostname to: db-server
- Create a non-root administrator account (e.g. dbadmin).
- Use a separate partition layout: /, /var, /home, /tmp, swap.
- Do NOT install a desktop environment — select only SSH server and standard utilities.

Hardening tasks

3. Update the system fully after installation.
4. Disable any unnecessary services (check with systemctl).
5. Harden SSH: change the default port, disable root login, restrict access to dbadmin only, and enforce key-based authentication.
6. Enable automatic security updates.
7. Verify SSH access on the new port from your host machine.

> **Note:** Enable automatic updates BEFORE configuring the firewall.

Part 3 — Firewall Configuration (UFW)

Configure UFW to enforce a strict least-privilege policy: deny all incoming traffic by default, and open only the ports strictly needed.

Tasks

8. Install UFW.

9. Set the default policy: deny all incoming, allow all outgoing.
10. Allow SSH on your custom port (with a rate-limit rule).
11. Allow PostgreSQL (port 5432) from localhost only.
12. Enable UFW and verify the active rules.
13. From your host, run a port scan (e.g. nmap) — confirm port 5432 appears as filtered.
14. Enable UFW logging.

> **Warning:** Always add your SSH rule BEFORE enabling UFW to avoid locking yourself out.

Part 4 — Brute-Force Protection (Fail2ban)

Fail2ban monitors log files and automatically bans IP addresses showing suspicious behavior (repeated authentication failures).

Tasks

15. Install Fail2ban and enable it at startup.
16. Create a local configuration file (do not edit jail.conf directly).
17. Configure the SSH jail: custom port, max 3 retries, ban for 2 hours.
18. Configure a PostgreSQL jail targeting authentication failure log entries.
19. Restart Fail2ban and verify both jails are active.
20. Simulate an SSH brute-force attack from a test machine (3+ failed logins). Verify the IP is banned using fail2ban-client.
21. Unban the test IP and confirm it is released.

> **Hint:** Use the command: fail2ban-client status sshd to inspect the SSH jail status.

Part 5 — PostgreSQL Installation & Security

Install PostgreSQL and apply security hardening at every layer: network, authentication, access control, encryption, and logging.

Tasks

22. Install the postgresql and postgresql-contrib packages.
23. Verify the service is running and check the installed version.
24. Set a strong password for the postgres superuser.
25. Create a dedicated application user (no superuser privileges) and a corresponding database.
26. In postgresql.conf — configure the server to listen on localhost only, enable SSL, and activate connection logging.
27. In pg_hba.conf — restrict access using scram-sha-256 authentication. Remove any trust entries.
28. Generate a self-signed SSL certificate and enable SSL for PostgreSQL.
29. Revoke all default privileges from the PUBLIC role on the application database.
30. Test the SSL connection: psql with sslmode=require, then run \conninfo.
31. Trigger 5+ failed login attempts and verify they appear in both the PostgreSQL log and Fail2ban.

> **Security rule:** Never use trust authentication in production. Prefer scram-sha-256 over md5 for all network connections.

Reflection Questions

Answer the following questions briefly in your lab report:

32. Why is it recommended to change the default SSH port? What are the limits of this practice?
33. What is the difference between md5 and scram-sha-256 in pg_hba.conf?
34. What would happen if listen_addresses is set to '*' without any firewall rule?
35. How does Fail2ban interact with UFW to ban an IP address?