Q A T A R U N I V E R S I T Y
College of Engineering
Dept. of Computer Science & Eng.

Operating Systems Lab (CMPS 405)

PBL Project Phase # 1

Spring 2025 Semester

Shell Scripting

Due to 13 March 2025

يعد الغش مخالفة أكاديمية وفقا للوائح والقوانين المعمول بها في جامعة قطر، وقد تصل عقوبة هذه المخالفة في بعض الحالات إلى الفصل النهائي من الجامعة وعلى الطلاب تجنب القيام أو المشاركة في أي عمل يخالف ميثاق النزاهة الأكاديمية وإجراءات الاختبارات المعمول بها في جامعة قطر

Cheating is an academic violation according to Qatar University rules and regulations, and in some cases, it may result in final dismissal from the university. Students should not under any circumstances commit or participate in any cheating attempt or any act that violates student code of conduct.

It is the right of the instructor to test the student's undersetting of the project in any way during the demo and discussion session. So, a project that is 100% working might be may be graded (-100%) due to student not being able to explain a functionality they have implemented.

Project Objectives:

1. **Advanced User and Group Management:** Implement nested groups, supplementary groups, and role-based permissions to enforce strict access controls.
2. **Security Hardening:** Configure SSH key-based authentication, ACLs, and security policies to mitigate unauthorized access.
3. **Automation with Error Handling:** Develop scripts that handle edge cases, log activities, and recover from failures.
4. **Real-World System Monitoring:** Create integrated solutions for tracking system metrics, service statuses, and user activities.
5. **Practical Troubleshooting:** Validate configurations manually and resolve misconfigurations in simulated real-world scenarios.

Environment Setup:

- **Virtual Machines:** Use **Ubuntu Server 22.04 LTS** for three VMs:
  - **VM1 (Server):** (4GB RAM, 4 CPUs) Hosts user accounts, shared directories, and critical services (SSH, MySQL).
  - **VM2 (Client - Development Team):** (2GB RAM, 2 CPU each) Used for development tasks and directory monitoring.
  - **VM3 (Client - Operations Team):** (2GB RAM, 2 CPU each) Used for system monitoring and reporting.

Note: [If it is not possible to install it on the same machine, feel free to install it on different physical machines].

Tasks:

Server Side Tasks (VM1):

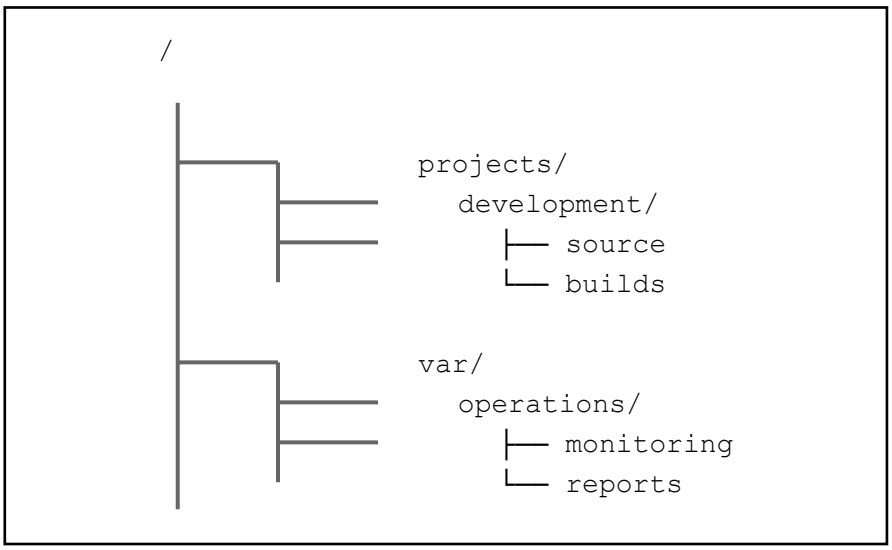
Server-Setup requirement:

- Install MySQL server, configure MySQL server to start automatically on boot. Checks and displays the status of the MySQL service.

Task 1: Advanced User and Group Configuration

1. **Nested Group Creation:**
   - Create groups: developers, dev_leads, operations, ops_admin, monitoring.
   - Assign users with nested group memberships:
     - dev_lead1 → developers + dev_leads
     - ops_lead1 → operations + ops_admin
     - ops_monitor1 → operations + monitoring
   - Add the following users: dev_lead1 and ops_lead1 to the Sudoers group.
   - **Script Requirement:** Automate group assignments using a Bash script (user_setup.sh).

2. **Shared Directories with ACLs:**
   - Create directory structure:



   - Apply **ACLs** to grant permissions:
     - developers group: Full access to /projects/development/source, read-only for /projects/development/builds.

        - monitoring group: Read-only access to /var/operations/reports.
    - **Script Requirement:** Write a script (dir_perms.sh) to enforce permissions.

Task 2: Security Hardening

1. **SSH Key-Based Authentication:**
    - Configure SSH to **disable password authentication** for dev_lead1.
    - Generate and deploy SSH keys for dev_lead1 to access VM1.
    - **Script Requirement:** Script (ssh_config.sh) must auto-detect and block password-based login attempts for dev_lead1.
2. **Automatic Security Updates:**
    - Enable unattended (unattended-upgrades package) security updates.
    - Log update statuses in /var/log/security_updates.log.

3. **Configure Message of the Day (MOTD)**
    - Configure a login message that welcomes users upon SSH access to VM1 terminal. The message should state, "Welcome to the Ubuntu Administration Lab."

Task 3: System Monitoring & Automation

1. **Metric Collection Script (system_monitor.sh):**
    - Collect every hour:
        - CPU/Memory usage, disk I/O, top 5 resource-heavy processes.
        - Status of MySQL, SSH.
    - Save logs to /var/operations/monitoring/metrics_<timestamp>.log.
    - **Edge Case Handling:** If Apache/MySQL fails, restart the service and alert via log.
2. **File Activity Monitor (file_audit.sh):**
    - Use inotifywait to monitor /projects/development/ for changes.
    - The changes include current user, action type (create/modify/delete), and log these changes to file path (/var/log/file_changes.log).

Task 4: MySQL User Management and Database Exploration

1. **MySQL User Creation:**
    - Create two MySQL users: dev_lead1, ops_lead1
    - Assign each user appropriate privileges, ensuring least-privilege access principles.
    - Verify that the users were successfully created.
2. **User Authentication and Verification**

- Log into MySQL using each newly created user.
- Confirm authentication and user session details.

3. **Database and Table Exploration**
   - Under each user, list all available databases.
   - List all tables within each database accessible to the user.
4. **Logging and Audit Trail**
   - Implement logging to track user logins and database interactions.
   - Store logs in /var/log/mysql_audit.log for auditing purposes.

Script Requirement: Write a script (MySQL_login_<USER_NAME>.sh) to automate all previous steps.

Task5: Network Connectivity Monitoring

Write a shell script called (Network.sh) that pings both Client1VM and Client2VM from ServerVM with 10 packets, each having a size of 500 bytes, every 10 seconds. The results should be appended to a file called TimeStamp.txt, where the timestamp is based on the current time when the ping is executed. Change the permission for each Timestamp file to read and write for the owner only. Finally, all ping results must be stored into a directory called PingDirectory.

Client side Tasks (VM2) and (VM3):

Task 1: Development Team (VM2)

1. **Login Attempt Monitoring (login_audit.sh):**
   - Log invalid SSH attempts to VM1 in invalid_attempts.log.
   - Block IP after 3 failed attempts using iptables.
2. **Permission Cleanup Script (fix_perms.sh):**
   - Find files with 777 permissions, change to 700, and log actions to perm_changes.log.

Task 2: Operations Team (VM3)

1. **Resource Reporting Script (resource_report.sh):**
   - Hourly, gather:
     - Process tree, zombie processes, CPU/memory usage.
     - Top 5 resource-consuming processes.
   - Securely copy the report to VM1 using SCP.

2. **Quota Enforcement (quota_check.sh):**
    - Check disk usage for /shared directory.
    - Enforce quotas:
        - dev_lead1: 5GB hard limit (6GB warning).
        - ops_lead1: 3GB hard limit (4GB warning).
    - Email alerts to admin@qu.edu.qa if thresholds are exceeded.

Instructions & Deliverables: Please read carefully

1. **Submission Requirements:**
    - **Word Document:**
        - Cover page with group member details (Name, ID).
        - Task distribution table (tasks assigned per member + contribution %).
    - **Scripts:** All scripts must be error-free and include inline comments explaining logic.
    - **Zip File:** Named as Student1_Student2_Student3_Student4.zip.
2. **Anti-Plagiarism Measures:**
    - **Demo Session:** Randomly assigned tasks must be executed live.
    - **Script Defense:** Students must explain every line of code during the demo.
3. **Penalties:**

- Copying and/or plagiarism (-100%) which includes:
    - **Inappropriate interaction with any other student, outside agency, website, or software that generates assessment responses.**
- Shell Script should be error free (Errors) (-25%).
- In case of late submission, (-10%) for each day of delay (Max 3 days delay).
- A group of three to four students can work on the project.
- Team members are required to meet regularly for discussion and workload distribution.
- It is the right of the instructor to use any way of testing the student in the discussion and demo session, and according to that in some cases (100%) graded project may be down to (-100%) graded project.
- Discussion & Demo 50%. + Student Work 50%.
- One demo after the submission of the second project phase.