**Department of Computer Science**

**Project Progress Report – Academic Year 114**

**Title: Linux System Automated Backup and Monitoring**

**Advisor:** Professor

**Team Leader:** Gao En-Zai

**Date: March 29, 2025**

## Table of Contents

1. **Abstract**
2. **Research Background and Objectives**
3. **Project Content**
4. **Research Methods**
5. **Flowchart**
6. **Planned Tasks and Expected Outcomes**
7. **Work Allocation and Project Schedule (Gantt Chart)**
8. **Problems Encountered and Solutions**
9. **Equipment Requirements**
10. **References**

**1. Abstract**

This study aims to develop a complete automated backup and management system. The system not only performs data backups automatically, but also periodically deletes expired files and manages backup logs. By combining Linux basic commands, shell scripting, Docker containerization, and systemd service management, we implement a fully functional and efficient backup system. The learning process also emphasizes high availability designs such as error detection and notification, and automatic restart mechanisms. These techniques provide a strong foundation for penetration testing and cybersecurity practices.

Ultimately, the study delivers a one-click deployment script (deploy.sh) that automates the entire process—from backup, cleanup, to notification—and can be easily ported to other systems.

**2. Research Background and Objectives**

Before advancing to penetration testing and higher-level security practices, mastering basic Linux operations and system administration is crucial. Kali Linux is widely used in penetration testing, with many built-in tools. However, effective use of Kali requires solid Linux fundamentals.

Thus, this project focuses on learning **Ubuntu** as the foundational Linux system to build skills in OS management, command-line usage, scripting, and containerization.

**Objectives:**

* Learn Ubuntu basics: file system, user management, permission control.
* Master essential system administration tools (Docker, systemd, cron, logrotate) and apply them to automated backups.
* Understand automated scheduling, error notifications, and log management in Linux.
* Build a strong foundation for future use of Kali Linux in penetration testing and defense.

## 3. Project Content

The project designs and implements an **automated backup and management system**, divided into key steps:

1. **Linux Basic Commands**
   * Practice commands: mkdir, chmod, tar, find for backups and file cleanup.
2. **Shell Scripts & Systemd**
   * Write backup.sh to automate backups and configure systemd for scheduled execution.
   * Backup files into .tar.gz and clean up files older than 7 days.
3. **Docker Containerization**
   * Package backup logic into a Docker container for portability and consistency across environments.
4. **Log & Expired File Management**
   * Use logrotate for log rotation and find to delete expired backups.
5. **Error Handling & Restart**
   * Configure systemd for automatic restart when backups fail.
6. **One-click Deployment**
   * Implement deploy.sh to automate installation, scheduling, Docker setup, and error handling.

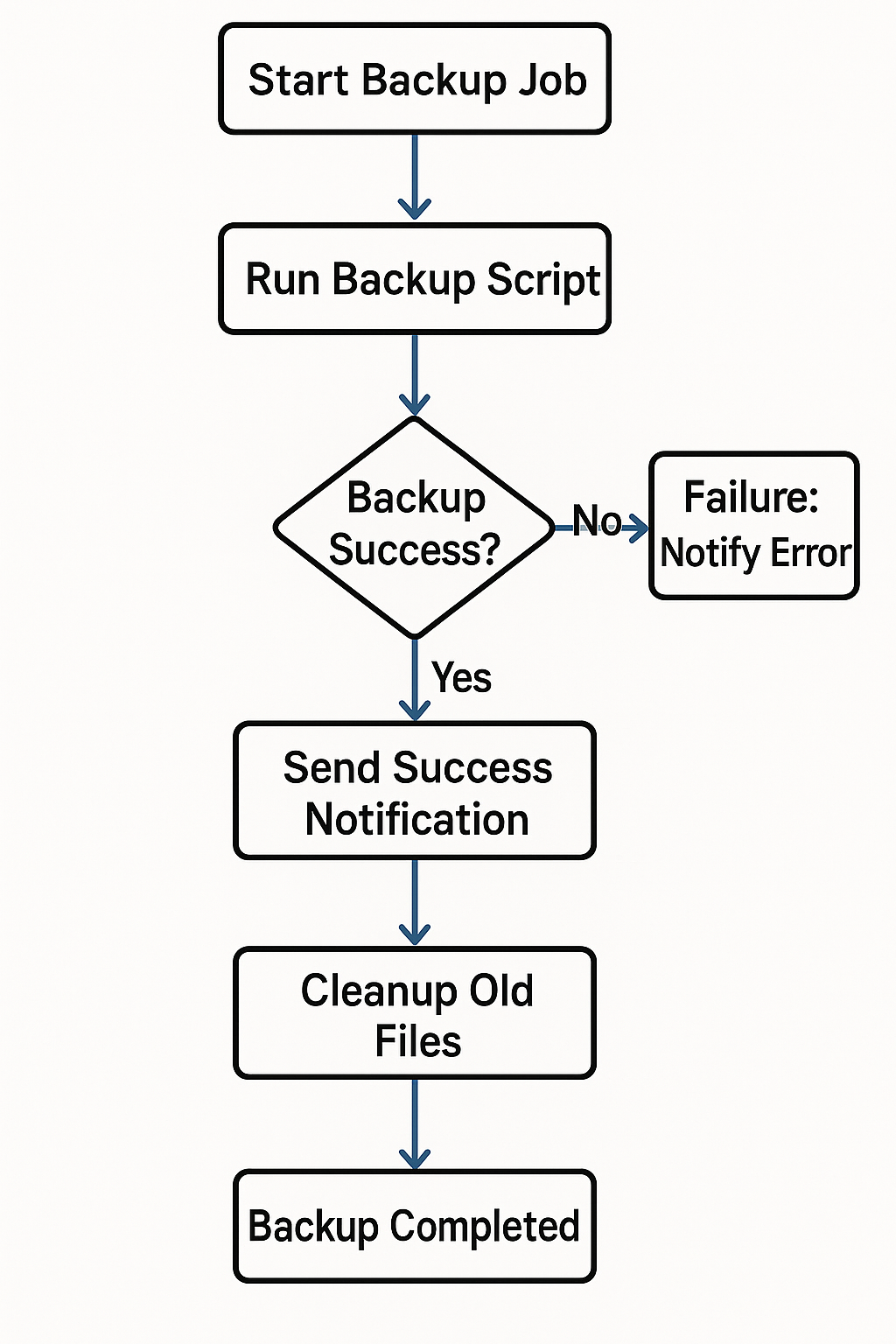
**4. Research Methods**

The penetration testing process adopted in this study includes:

1. **Environment Setup:** VirtualBox with Kali Linux and OWASP BWA hosts.
2. **Port and Service Scanning:** Use nmap, netstat, and arp -a to probe services and connections.
3. **Reverse Shell Injection:** On Kali, use nc -lvnp 4444 to wait for callbacks. On the target, execute .sh backdoor script or bash -i.
4. **Privilege Escalation & Data Access:** Gain root privileges, retrieve sensitive files and logs.
5. **Web Analysis:** Explore /var/www/ directory for projects such as bwapp, webgoat, and evil to evaluate vulnerabilities.

**5. Flowchart**

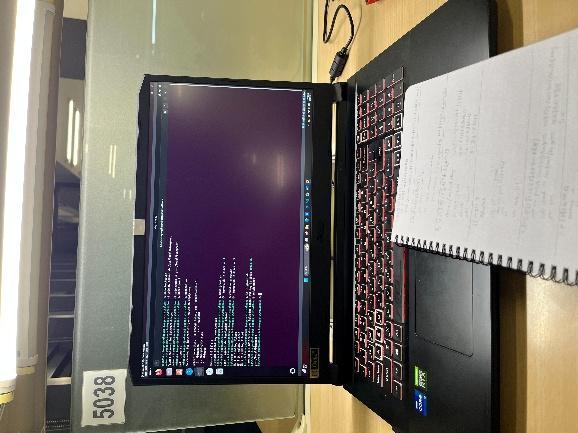
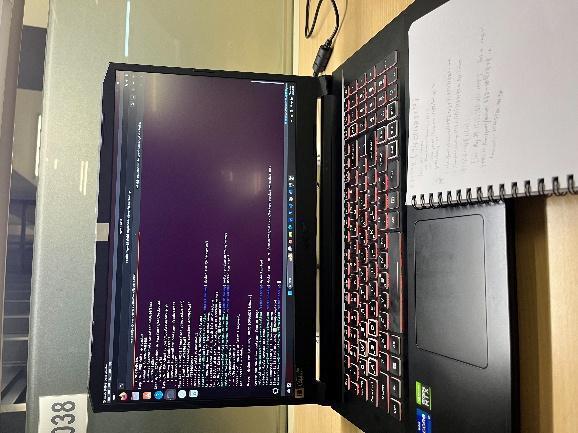
The following is the flowchart of the automated backup system:

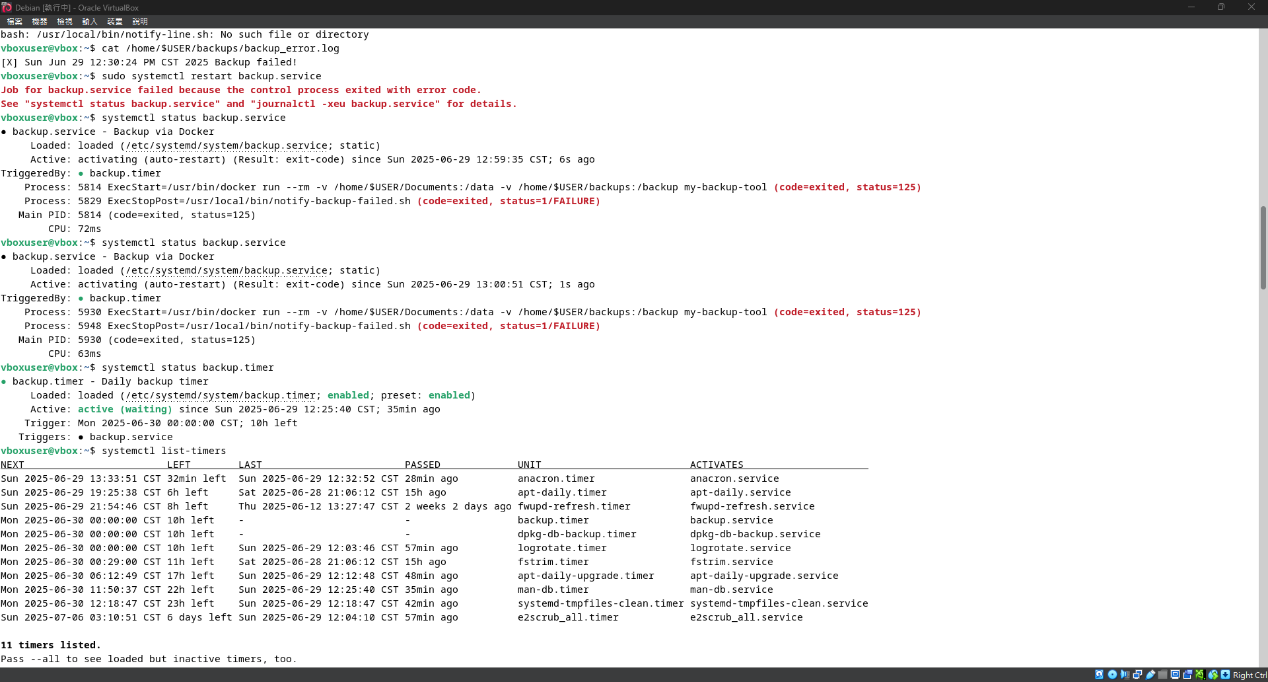
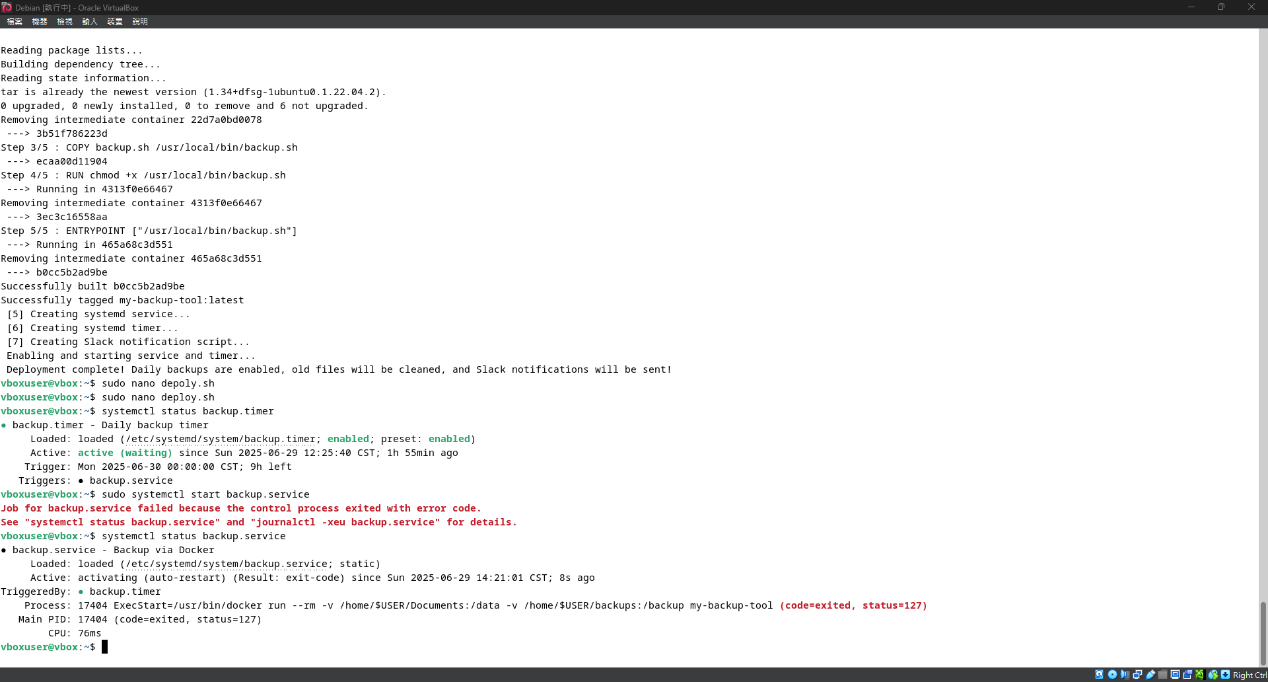
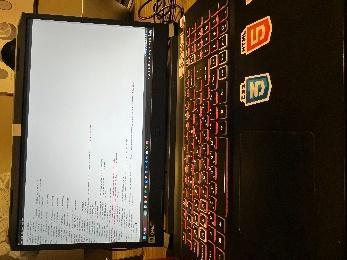
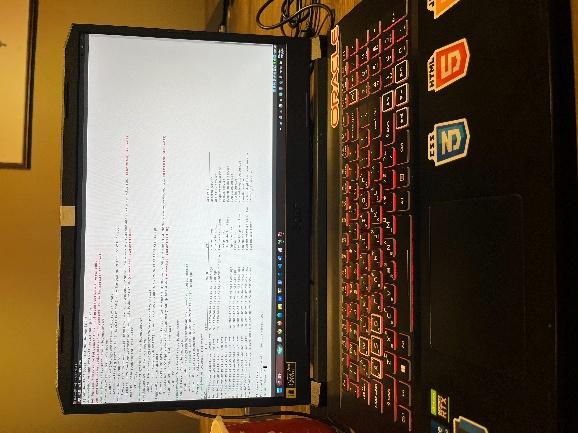
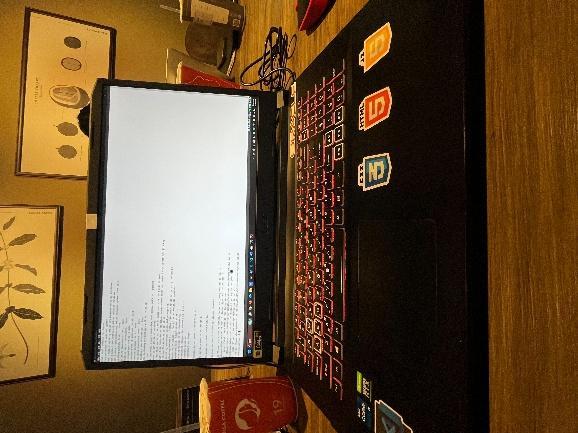


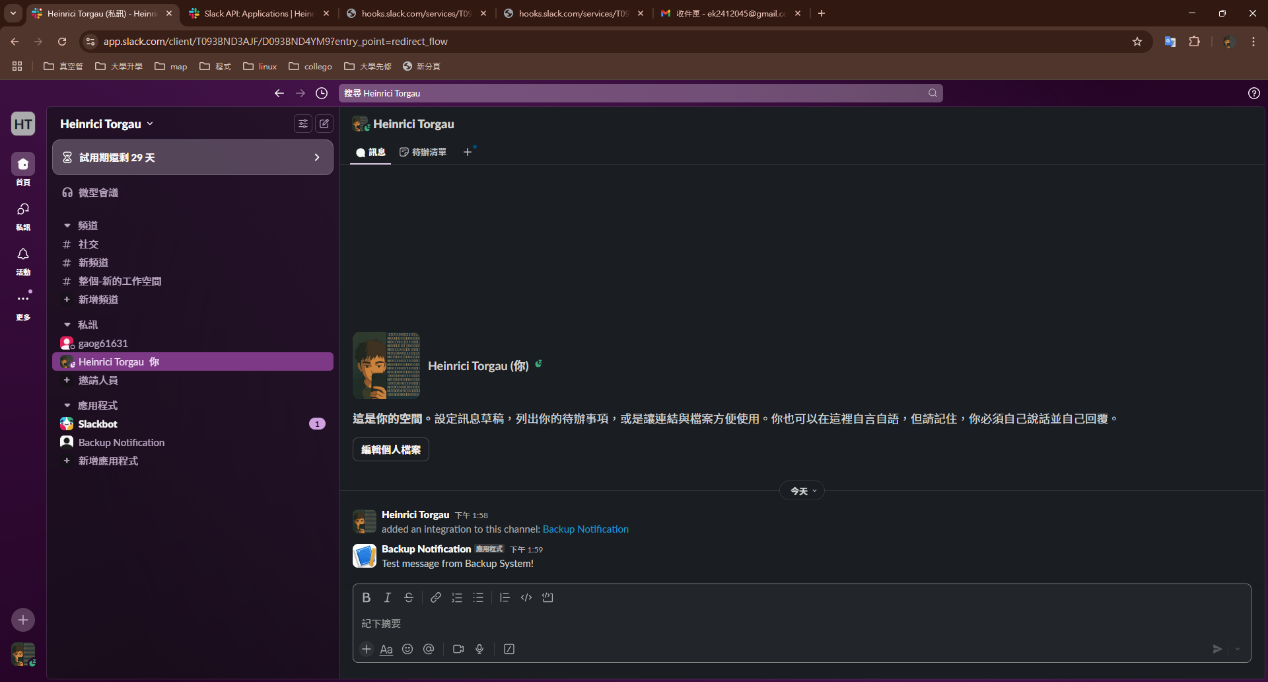
**6. Planned Tasks and Expected Outcomes**

| **Task** | **Expected Outcome** |
| --- | --- |
| Backup script | Automated backup & compression |
| systemd service | Backup + scheduled tasks configured |
| Log & cleanup | logrotate + expired file cleanup |
| One-click deployment | Full automation with deploy.sh |

**Experimental Screenshots**







## Supplementary Scripts

### 1. backup.sh

* Compress /data → /backup/documents-YYYYMMDD.tar.gz.
* Delete backups older than 7 days.
* Send Slack notification.

### 2. deploy.sh

* Automates directory setup, script installation, Docker image build, systemd service and timer creation, and Slack integration.
* Enables daily automated backups.

### 3. backup.service and backup.timer

* **backup.service**: runs the Docker backup container, restarts on failure.
* **backup.timer**: triggers backup daily (OnCalendar=daily, Persistent=true).

**7.Work Allocation & Schedule (Gantt Chart)**

| **Day(s)** | **Task** | **Status** |
| --- | --- | --- |
| 1–5 | Linux basics & scheduling | Completed |
| 6–10 | Shell scripts & systemd | Completed |
| 11–13 | Docker backup container & automation | Completed |
| 14–15 | Cleanup & log management | Completed |
| 16 | Docker Compose backup system | Completed |
| 17 | Report & system structure integration | Completed |
| 18 | Error monitoring & auto-restart | Completed |
| 19 | One-click deployment script | Completed |

**8.Problems Encountered and Solutions**

* **Docker folder mount issue**: resolved by aligning container and host paths + setting permissions.
* **systemd timer not triggering**: resolved by reloading systemd and checking logs.

**9.Equipment Requirements**

| **Equipment** | **Purpose** | **Quantity** |
| --- | --- | --- |
| Ubuntu server | Run backup system | 1 |
| Docker environment | Execute backup scripts | 1 |
| Network connection | Send LINE/Slack notifications | 1 |
| Backup disk | Store backup archives | 1 |

10.References

1. Docker Official Docs:<https://docs.docker.com/>
2. systemd Official Docs:<https://www.freedesktop.org/wiki/Software/systemd/>
3. logrotate Manual:<https://linux.die.net/man/8/logrotate>