GROUP 4

NAMES	INDEX NUMBER
Saraphine Efui	052441360251
Koomson Kwabena Gyimah Godfred	052441360142
Kusi Francis	052441360372
Yakubu Ebenezer	052441360071
Solomon Agyei	052441360239

Report: Secure File Transfer from VM to Cloud Droplet

GITHUB LINK : https://github.com/saraphineefui100-prog/Group-4-Cap-Stone-Project-

Introduction

The purpose of this project was to design and implement a secure, automated mechanism for transferring files from a Virtual Machine (VM) to a cloud droplet. The solution had to meet requirements in networking, security, scripting, and cloud deployment. Specifically, the project leverages scp and rsync for data transfer, SSH keys for authentication, firewall rules for access control, and a Bash script with logging and error handling for automation.

This report presents the methodology used, the implementation details, security configurations, and verification through logs and screenshots. The final deliverables include the transfer script, project documentation, and system configuration screenshots.

Skills Applied

1. Networking (Week 4): Configured secure file transfer using scp and rsync. Verified connectivity between the VM and the droplet using SSH. Implemented a pre-flight connectivity check in the script.

2. Security (Week 5): Generated and deployed SSH key pairs to replace password authentication. Hardened SSH settings by editing /etc/ssh/sshd_config to disable root login and password-based authentication. Configured UFW (Uncomplicated Firewall) to allow SSH only from the VM's public IP address.

3. Scripting (Week 6): Created transfer_files.sh to automate file transfers. Implemented robust error handling with set -euo pipefail. Logged all transfer attempts to ~/myproject/logs/transfer.log. Added alerting mechanisms to notify on failures.

4. Cloud (Week 7): Prepared the droplet with a dedicated user (deploy). Set up appropriate directory permissions (/srv/inbox). Verified firewall and SSH configurations on the cloud host.

Implementation

1. Script: transfer_files.sh – A Bash script was developed to automate the secure transfer of files. Key features include configurable parameters, support for both rsync and scp, logging, error handling, and optional alerts.

2. SSH Key Authentication – SSH keys were generated and copied to the droplet's authorized_keys. Root login and password authentication were disabled to reduce attack surface.

3. Firewall Configuration (UFW) – The droplet firewall was configured to allow SSH only from the VM's IP, enhancing security.

4. Documentation – A README was prepared describing SSH setup, UFW configuration, and script usage.

5. Verification Screenshots – Screenshots included transfer logs and UFW status.

Results

The project resulted in a robust file transfer system with secure authentication using SSH keys, hardened droplet access with firewall rules, and automated file transfer through a reusable Bash script. Logs and alerts provided auditing and monitoring capabilities. The solution successfully demonstrated secure file transfer from VM to cloud droplet.

Conclusion

This project demonstrated the integration of networking, security, scripting, and cloud concepts into a practical solution for secure file transfers. By combining rsync/scp with SSH key authentication, firewall hardening, and a scripted workflow, a reliable and secure pipeline was created.

The deliverables—script, documentation, and screenshots—provide a reproducible and auditable method for transferring files between a VM and a cloud droplet. The successful outcome reinforces best practices in secure system administration and cloud deployment.

Deliverables Submitted

- ~/myproject/scripts/transfer_files.sh

- ~/myproject/docs/project_readme.md

- Screenshots: tail transfer.log, sudo ufw status

- (Optional) GitHub repository

```
saraphine@saraphine-VirtualBox:~$ sudo ufw status
[sudo] password for saraphine:
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
8000                       ALLOW       Anywhere
80                         ALLOW       Anywhere
OpenSSH                    ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
8000 (v6)                  ALLOW       Anywhere (v6)
80 (v6)                    ALLOW       Anywhere (v6)
OpenSSH (v6)               ALLOW       Anywhere (v6)

saraphine@saraphine-VirtualBox:~$
```

```
saraphine@saraphine-VirtualBox:~$ tail ~/myproject/logs/transfer.log
2025-08-16T01:01:17+0000 [INFO] Starting transfer from /path/to/local/folder to saraphi
ne@154.161.246.254:/srv/incoming via rsync
ssh: connect to host 154.161.246.254 port 22: Connection timed out
rsync: connection unexpectedly closed (0 bytes received so far) [sender]
rsync error: error in rsync protocol data stream (code 12) at io.c(232) [sender=3.4.1]
2025-08-16T12:07:12+0000 [INFO] Starting transfer from /path/to/local/folder to saraphi
ne@154.161.246.254:/srv/incoming via rsync
Warning: Identity file /home/saraphine/.ssh/id_rsa not accessible: No such file or dire
ctory.
ssh: connect to host 154.161.246.254 port 22: Connection timed out
rsync: connection unexpectedly closed (0 bytes received so far) [sender]
rsync error: unexplained error (code 255) at io.c(232) [sender=3.4.1]
2025-08-16T12:09:25+0000 [ERROR] Transfer failed
saraphine@saraphine-VirtualBox:~$
```