For the role of Software Engineer I will schedule your interview directly with our founder. But before this you have to complete a assignment which his given below. As soon as you complete the assignment kindly mail me back so that i can schedule your interview with the founder.

The assignment details are included below:

The Assignment

Build a lightweight Linux agent (preferably in Golang) that:

Collects installed packages from a Linux endpoint (e.g., via dpkg/rpm/apk depending on distro). The agent itself should be installed via a proper package format (e.g., .deb or .rpm) and show up under the system's package manager.

Performs 10 Linux security configuration checks mapped against the CIS Benchmark for Ubuntu 22.04 LTS or RHEL 9 (Level 1). Example checks include:

Password complexity and expiration policies enforced

Root login disabled over SSH

Unused filesystems (e.g., cramfs, squashfs) disabled

Firewall enabled and configured (ufw or firewalld)

Time synchronization (chrony or ntpd) configured

Auditd service running

SELinux/AppArmor enabled

No world-writable files

GDM/desktop auto-login disabled

[plus 1–2 additional checks of your choice]

Communicates securely with the cloud:

The agent should send collected data and CIS check results to AWS.

You may use API Gateway → Lambda → DynamoDB/S3 (preferred), or a simpler REST ingestion service if that's faster for you.

Data should be sent in JSON format.

Exposes REST APIs from the backend to retrieve this data (e.g., /hosts, /apps, /cis-results).

Provides a basic frontend to display:

Host details (hostname, OS version, etc.)

Installed packages

CIS check results (pass/fail with evidence)

⚡ Note: You do not need physical Linux hardware. You can run and test everything on an AWS EC2 instance (Ubuntu or RHEL) or any other VM environment you prefer.

Expected Effort

MVP version: 12–18 hours (1–2 focused days).

Polished version with installer refinements, nicer UI, and better error handling: 20–30 hours.

We are fine with the MVP scope for this exercise.

Interview & Demo

You will demo your project during your interview slot.

The demo should cover:

How the agent works (collection + communication).

The CIS checks you implemented.

How results flow to AWS and APIs.

The frontend view.

Design decisions, challenges, and potential improvements.