Operations Security Procedure

Version 1 - Approved by Rishabh

Contents

1. Objective

The objective of this procedure is to provide guidelines to ensure the operational security of Superhawk AI's services through procedures for backup, change management, logging, and vulnerability management.

2. Scope

This procedure covers all systems within our production environment that are owned or operated by Superhawk AI, including cloud assets hosted on AWS and GCP, code repositories in GitHub, and employee devices used to access production systems. It applies to engineering, infrastructure, and operations personnel, as well as contractors who have access to production systems or data.

3. Operations Security Procedure

3.1. Change Management Procedure

3.1.1. Use of Version Control Systems

- All software developed in the service of Superhawk AI or any subdomain of Superhawk AI's products should be version controlled i.e. the latest version of our software as well as any previous version of our software are readily available.
- Superhawk AI uses Git with repositories hosted on GitHub for all source code and infrastructure-as-code. Engineers and authorized contractors create feature branches for development, run local and automated tests, and open GitHub pull requests to propose changes. All changes must be tested in local or CI environments before being merged and deployed.

3.1.2. Initiating Planned Changes

- Planned changes are initiated by creating feature or change branches in GitHub and documenting the change in the associated issue or ticket in the company's task tracking system. For infrastructure or network changes, changes must be tracked in the ticketing system with a documented owner, approver, impact assessment, and rollback plan.
- Feature branches are usually developed locally by engineers or contractors and pushed to GitHub when ready for review. A pull request is created to merge changes into the main branch when the change is ready for testing or release. Any authorized engineer or designated reviewer may initiate a pull request.
- Network or operational changes not represented in code (e.g., cloud console changes) must be recorded in the ticketing system with clear steps, owner, approver, and rollback instructions.

3.1.3. Approving Planned Changes

- Pull requests in GitHub must be reviewed and approved by at least one person who is not the author. Automated CI checks (unit tests, linters, and any configured security scans)

should run on pull requests and their results reviewed before approval. Reviewers verify that prerequisites are met and use the checklist below as applicable.
- For changes tracked in the ticketing system, the ticket must include approval from the designated approver (e.g., engineering lead) before changes are applied to production.
- Before approving and merging a pull request or approving a ticket, the reviewer checks that all prerequisites are met. Typical checks that the reviewer is encouraged to perform are listed below. Not all items in the list are necessary (depending on the type of change), nor is the list exhaustive. Please use your judgment to determine what is necessary, depending on the change at hand. Below are some questions to ask:
    - Does the proposed change solve the problem it set out to solve? Are all requirements for solving the problem met? If not, were reasonable trade-offs made?
    - Are there any unintended consequences of this change to other parts of the system?
    - Does the change adversely affect any related or unrelated user experience?
    - Are there any algorithmic or logical errors in the proposed change?
    - Does the proposed change require changes in the environment itself (like adding production environment variables etc)?
    - Could the change create performance issues for itself (or other parts of the system)?
    - Could the proposed change be achieved in a more extensible, robust, or less disruptive way?

3.1.4. Unplanned Changes

- Sometimes, it becomes necessary to apply unplanned changes, like hotfixes, to the production system in order to maintain Superhawk AI's operational effectiveness. This is usually done to address a situation where the production system is in an undesired state - either from a customer-experience standpoint (like critical bugs, system-down, etc.) or from a security standpoint.
- Depending on the urgency of the fix required, unplanned changes may skip the requirements of a peer review/approval. Such requests are peer-reviewed post facto.
- When emergency changes are required, engineers create a branch or patch and document the change via a GitHub pull request or a ticket. The commit and pull request or ticket must describe the issue, the fix applied, and the rollout steps. Emergency changes must be reviewed after deployment by a qualified reviewer who is not the author.
- Because code and infrastructure changes are tracked in GitHub and through the ticketing system, emergency changes can be rolled back using standard git workflows or by restoring previous infrastructure configurations.

3.2. Backup Procedure

- Production customer data stored on AWS and GCP uses provider-managed backup services and snapshot capabilities. Superhawk AI configures daily automated snapshots/backups where available and documents backup configurations in infrastructure-as-code repositories on GitHub.
- The backup retention period shall be configured to a minimum of 7 days.
- Restoration shall be performed on the backup to ensure that data is readable/accessible. This exercise shall be performed at least once every year.

- The restoration tests shall be documented. The backup snapshot that was restored shall be documented along with any sanity checks performed to ensure that the restoration was successful.
- Restoration tests shall be performed by the administrators of Superhawk AI's production infrastructure along with the Information Security Officer.
- In an unlikely event of a natural or human-induced disaster, a disaster recovery plan needs to be in place for the systems to recover from the failure and be up and running. This can be achieved in the form of tabletop exercises which must be carried out by the Engineering Head and the Information Security Officer.

3.3. Vulnerability Management Procedure

- Superhawk AI performs various internal vulnerability scans and package monitoring on a constant basis.
- The Information Security Officer must ensure that Superhawk AI also performs external vulnerability scans/penetration tests periodically.
- All vulnerabilities detected by vulnerability scanners are tracked together along with their severity.
- It is the responsibility of the Infra operations person to ensure that vulnerabilities are remediated by the engineering team within defined SLAs (Process Config page).
- It is important to track the SLAs for the remediation of vulnerabilities. In case SLA is breached, it is the responsibility of the Information Security Officer along with engineering leads to ensure appropriate actions are taken. E.g. If a vulnerability needs more time to remediate, ensure that the justification for the same is documented.
- Vulnerabilities that are identified through external assessments may also be tracked along with other vulnerabilities for their closure if required.
- The engineering team addresses the reported vulnerabilities and tracks them to resolution. Resolution statuses can include (but are not limited to) the following:
    - Fixed: This means that the reported vulnerability has been fixed via a patch or system changes.
    - Inaccurate/Incorrect/False-positive: This means that the reported vulnerability has been thoroughly investigated, but found to be invalid.
    - Vulnerable but section unused: This means that the reported vulnerability affects parts of the codebase/system that are not in use, and consequently the vulnerability is no longer a threat.
    - Acceptable risk: This means that the reported vulnerability has been analyzed and deemed to not pose any debilitating risk to the system. This is a rare-case scenario, and should only occur when there are extenuating circumstances or extremely high remediation costs.

4. Document Security Classification

Company Internal. Please refer to the Data Classification Policy for more details.

5. Non-Compliance

Compliance with this policy shall be verified through various methods, including, but not limited to, automated reporting, audits, and feedback to the policy owner. Any staff member found to be in violation of this policy may be subject to disciplinary action, which may include termination of employment or contractual agreement. The disciplinary action shall depend on the extent, intent, and repercussions of the specific violation.

6. Responsibilities

The Information Security Officer is responsible for approving and reviewing policy and related procedures. Supporting functions, departments, and staff members, including engineering, infrastructure, and contractors with production access, shall be responsible for implementing the relevant sections of the policy in their area of operation.

7. Schedule

This document shall be reviewed annually or when significant organizational changes occur (for example, changes to cloud providers, major architecture changes, onboarding of contractors with production access, or changes in applicable compliance frameworks such as SOC 2, ISO 27001, or GDPR).

End of Operations Security Procedure. For version history, please see the next page.

Version History

	Version	Log	Date
1	Current	Policy version approved by Rishabh	13 Mar, 2026
1		New policy version created	13 Mar, 2026