6.857 — Problem Set 1 — Problem 1

Introduction

This document describes a proposed security policy for a platform similar to GitHub, a distributed version control hosting service that enables worldwide collaboration on software projects. The goal of the policy is to protect the confidentiality, integrity, availability, and authenticity of source code, metadata, and repository management operations. For this assignment, we assume that GitHub acts both as the hosting provider and as the maintainer of the Git version control system.

The policy outlines security goals, system roles, trust boundaries, and access control requirements that should guide an implementor of a GitHub-like platform.

Security Goals

- **Integrity:** Only authorized principals may modify repository data, commit histories, settings, or access permissions.
- **Availability:** Repository access, authentication services, and automation pipelines should remain available with minimal downtime.
- **Confidentiality:** Private repositories and sensitive metadata must be accessible solely to authorized users.
- **Authenticity and Non-Repudiation:** Commit attribution should reliably identify authors (e.g., using signed commits) and repository actions should be auditable.
- **Secure Collaboration:** The system must support safe distributed contribution through fine-grained access controls.

Roles and Principals

Platform Operator / GitHub Maintainer

Responsible for maintaining infrastructure, authentication, backups, audit logs, and enforcing system-wide security settings.
Permissions: Full administrative control of platform-level configuration, ability to suspend or investigate compromised accounts.

Repository Owner

Creates and manages repositories and controls access rights.
Permissions: Full read/write/admin privileges on their repositories, including branch protections and deployment policies.

Collaborator / Contributor

Authorized by repository owners to contribute code.
Permissions (configurable): Read, write, or admin access to specific repositories.

Organization Administrator

Manages teams and permissions for multiple repositories.
Permissions: Add/remove members, configure organization-wide settings such as single sign-on and repository defaults.

Authenticated User

Registered users on the platform.
Permissions: Read public repositories, fork, file issues, open pull requests.

Anonymous User

Users without authentication.
Permissions: Read-only access to public repositories.

Automated Integrations / Bots / CI Systems

Automated tools requiring controlled access.
Permissions: Limited-scope tokens following least privilege.

Authentication and Access Control Policies

Authentication

Authentication must support multi-factor authentication (MFA), SSH keys, OAuth tokens, and optional hardware security keys. MFA should be required for high-privilege operations.

Authorization

Authorization follows the principle of least privilege. Repository permissions must distinguish between read, write, and admin privileges. Branch protection rules should prevent force-pushes and enforce reviews and CI verification.

Credential Security

Passwords must be stored using strong cryptographic hashing. Access tokens must be stored encrypted and should never be retrievable once created.

Data Protection and Backup

All private repository content should be encrypted at rest. Backups should be performed regularly, verified for integrity, and stored securely. Deleted repositories should be recoverable for a fixed grace period.

Logging and Auditing

Administrative activity, permission changes, and destructive operations must be logged. Audit logs must be immutable, retained for a defined duration, and accessible to repository owners for their repositories.

Incident Response

The platform must support rapid response to suspected compromise, including locking accounts or repositories. Restoration to a secure state must be supported through version control and verified backups.

Third-Party Integrations

External services must authenticate via scoped tokens and must not bypass authorization checks. Tokens must be revocable at any time.

Conclusion

This security policy outlines the architectural and procedural requirements necessary to support a secure, reliable, and collaborative version control hosting platform. By enforcing strict access control, reliable authentication, data protection, and auditability, the platform can maintain trust while supporting global software development.