COALFIRE CONTROLS

Report on GitHub, Inc.’s GitHub Enterprise Cloud service Relevant to Security Throughout the Period April 1, 2024 to September 30, 2024

SOC 3® - SOC for Service Organizations: Trust Services Criteria for General Use Report

Table of Contents

Section 1

Independent Service Auditor’s Report

Independent Service Auditor's Report

To: GitHub, Inc. ("GitHub")

Scope

We have examined GitHub's accompanying assertion titled "Assertion of GitHub, Inc. Management" (assertion) that the controls within the GitHub Enterprise Cloud service (system) were effective throughout the period April 1, 2024 to September 30, 2024, to provide reasonable assurance that GitHub's service commitments and system requirements were achieved based on the trust services criteria relevant to security (applicable trust services criteria) set forth in TSP Section 100, *2017 Trust Services Criteria for Security, Availability, Processing Integrity, Confidentiality, and Privacy (With Revised Points of Focus—2022)* (2017 TSC).

The description of the boundaries of the system indicates that complementary user entity controls that are suitably designed and operating effectively are necessary, along with controls at GitHub, to achieve GitHub's service commitments and system requirements based on the applicable trust services criteria. Our examination did not include such complementary user entity controls and we have not evaluated the suitability of the design or operating effectiveness of such controls.

GitHub uses subservice organizations to provide data center colocation and infrastructure hosting services. The description of the boundaries of the system indicates that complementary subservice organization controls that are suitably designed and operating effectively are necessary, along with controls at GitHub, to achieve GitHub's service commitments and system requirements based on the applicable trust services criteria. The description of the boundaries of the system presents the types of complementary subservice organization controls assumed in the design of GitHub's controls. Our examination did not include the services provided by the subservice organizations, and we have not evaluated the suitability of the design or operating effectiveness of such complementary subservice organization controls.

Service Organization's Responsibilities

GitHub is responsible for its service commitments and system requirements and for designing, implementing, and operating effective controls within the system to provide reasonable assurance that GitHub's service commitments and system requirements were achieved. GitHub has also provided the accompanying assertion about the effectiveness of controls within the system. When preparing its assertion, GitHub is responsible for selecting, and identifying in its assertion, the applicable trust services criteria and for having a reasonable basis for its assertion by performing an assessment of the effectiveness of the controls within the system.

Service Auditor's Responsibilities

Our responsibility is to express an opinion, based on our examination, on management's assertion that controls within the system were effective throughout the period to provide reasonable assurance that the service organization's service commitments and system requirements were achieved based on the applicable trust services criteria. Our examination was conducted in accordance with attestation standards established by the American Institute of Certified Public Accountants. Those standards require that we plan and perform our examination to obtain reasonable assurance about whether management's assertion is

fairly stated, in all material respects. We believe that the evidence we obtained is sufficient and appropriate to provide a reasonable basis for our opinion.

We are required to be independent and to meet our other ethical responsibilities in accordance with relevant ethical requirements relating to the engagement.

Our examination included:

- Obtaining an understanding of the system and the service organization's service commitments and system requirements.
- Assessing the risks that controls were not effective to achieve GitHub's service commitments and system requirements based on the applicable trust services criteria.
- Performing procedures to obtain evidence about whether controls within the system were effective to achieve GitHub's service commitments and system requirements based on the applicable trust services criteria.

Our examination also included performing such other procedures as we considered necessary in the circumstances.

Inherent Limitations

There are inherent limitations in the effectiveness of any system of internal control, including the possibility of human error and the circumvention of controls.

Because of their nature, controls may not always operate effectively to provide reasonable assurance that the service organization's service commitments and system requirements were achieved based on the applicable trust services criteria. Also, the projection to the future of any conclusions about the effectiveness of controls is subject to the risk that controls may become inadequate because of changes in conditions or that the degree of compliance with the policies or procedures may deteriorate.

Opinion

In our opinion, management's assertion that the controls within the GitHub Enterprise Cloud service were effective throughout the period April 1, 2024 to September 30, 2024, to provide reasonable assurance that GitHub's service commitments and system requirements were achieved based on the applicable trust services criteria if complementary subservice organization controls and complementary user entity controls assumed in the design of GitHub's controls operated effectively throughout that period is fairly stated, in all material respects.

Coalfire Controls LLC

Greenwood Village, Colorado
November 26, 2024

Section 2

Assertion of GitHub, Inc. Management

GitHub

88 Colin P Kelly Jr Street,
San Francisco, CA 94107
Tel: 415-448-6673 (main)

Assertion of GitHub, Inc. ("GitHub") Management

We are responsible for designing, implementing, operating and maintaining effective controls within the GitHub Enterprise Cloud service (system) throughout the period April 1, 2024 to September 30, 2024, to provide reasonable assurance that GitHub's service commitments and system requirements were achieved based on the trust services criteria relevant to security (applicable trust services criteria) set forth in TSP Section 100, *2017 Trust Services Criteria for Security, Availability, Processing Integrity, Confidentiality, and Privacy (With Revised Points of Focus—2022)* (2017 TSC). Our description of the boundaries of the system is presented in attachment A and identifies the aspects of the system covered by our assertion.

The description of the boundaries of the system indicates that complementary user entity controls that are suitably designed and operating effectively are necessary, along with controls at GitHub, to achieve GitHub's service commitments and system requirements based on the applicable trust services criteria.

GitHub uses subservice organizations for data center colocation and infrastructure hosting services. The description of the boundaries of the system indicates that complementary subservice organization controls that are suitably designed and operating effectively are necessary, along with controls at GitHub, to achieve GitHub's service commitments and system requirements based on the applicable trust services criteria. The description of the boundaries of the system presents the types of complementary subservice organization controls assumed in the design of GitHub's controls. The description of the boundaries of the system does not disclose the actual controls at the subservice organizations.

We have performed an evaluation of the effectiveness of the controls within the system throughout the period April 1, 2024 to September 30, 2024, to provide reasonable assurance that GitHub's service commitments and system requirements were achieved based on the applicable trust services criteria if complementary subservice organization controls and complementary user entity controls assumed in the design of GitHub's controls operated effectively throughout that period. GitHub's objectives for the system in applying the applicable trust services criteria are embodied in its service commitments and system requirements relevant to the applicable trust services criteria. The principal service commitments and system requirements related to the applicable trust services criteria are presented in attachment B.

There are inherent limitations in any system of internal control, including the possibility of human error and the circumvention of controls. Because of these inherent limitations, a service organization may achieve reasonable, but not absolute, assurance that its service commitments and system requirements are achieved.

We assert that the controls within the system were effective throughout the period April 1, 2024 to September 30, 2024 to provide reasonable assurance that GitHub's service commitments and system requirements were achieved based on the applicable trust services criteria.

GitHub, Inc.

Attachment A

GitHub, Inc.’s Description of the Boundaries of Its GitHub Enterprise Cloud service

Type of Services Provided

GitHub (“the Company”) is an independently operated subsidiary of Microsoft and generated its first commit in 2007. It is headquartered in San Francisco, CA, with additional offices in Bellevue, WA, and Oxford, the UK. GitHub currently employs approximately 3,000 employees, with approximately 95 percent of the workforce being remote.

GitHub is a web-based software development platform built on the Git version control software. Primarily used for software code, GitHub offers the distributed version control and source code management functionality of Git with additional features and enhancements. Specifically, it provides access control and several collaboration features, including bug tracking, feature requests, task management, GitHub Teams, pull requests, discussions, issues, pages, projects, docs, and wikis. GitHub Enterprise Cloud service with Data Residency is an optional version of GitHub Enterprise Cloud service that offers features that provide customers control over the location in which their data is stored. Customers can optionally add Copilot Business, Copilot Enterprise, Actions, and GitHub Advanced Security. In this report, unless specified otherwise, statements about GitHub Enterprise Cloud service apply to all in-scope products represented above.

The following are descriptions of GitHub Enterprise Cloud service features.

Organizations

An organization is a collection of user accounts that owns repositories. Organizations have one or more owners, who have administrative privileges for the organization. When a user creates an organization, it does not have any repositories associated with it. At any time, members of the organization with the owner role can add new repositories or transfer existing repositories.

Code Hosting

GitHub is one of the largest code hosts in the world, with millions of projects. Private, public, or open-source repositories are equipped with tools to host, version, and release code. Unlimited private repositories allow keeping the code in one place, even when using Subversion or working with large files using Git Large File Storage.

Changes can be made to code in precise commits, allowing for quick searches on commit messages in the revision history to find a change. In addition, blame view enables users to trace changes and discover how the file, and code base, has evolved.

With sharing, changes can be packaged from a recently closed milestone or finished project into a new release. Users can draft and publish release notes, publish pre-release versions, attach files, and link directly to the latest download.

Code Management

Code review is a critical path to better code and is fundamental to how GitHub works. Built-in review tools make code review an essential part of team development workflows.

A pull request is a living conversation where ideas can be shared, tasks assigned, details discussed, and reviews conducted. Reviews happen faster when GitHub shows a user exactly what has changed. Diffs compare versions of source code side by side, highlighting the parts that are new or have been edited or deleted.

Pull requests also enable clear feedback, review requests, and comments in context with comment threads within the code. Comments may be bundled into one review or in reply to someone else's comments inline as a conversation.

GitHub allows customers to protect important branches by setting branch protection rules, which define whether collaborators can delete or force push to the branch and set requirements for any pushes to the branch, such as passing status checks or a linear commit history. Protected branches allow for better quality code management. Repositories can be configured to require status checks, such as continuous integration (CI) tests, reducing both human error and administrative overhead.

Project Management

Project boards allow users to reference every issue and pull request in a card, providing a drag-and-droppable snapshot of the work that teams do in a repository. This feature can also function as an agile idea board to capture early ideas that come up as part of a standup or team sync, without polluting the issues.

Issues enable team task tracking, with resources identified and tasks assigned within a team. Issues may be used to track a bug, discuss an idea with an @ mention, or start distributing work. Issue and pull request assignments to one or more teammates make it clear who is doing what work and what feedback and approvals have been requested.

Milestones can be added to issues or pull requests to organize and track progress on groups of issues or pull requests in a repository.

Teams (User Management)

NOTE: This section refers to teams, not the plan called GitHub Team.

Building software is as much about managing teams and communities as it is about code. Users set roles and expectations without starting from scratch. Customized common codes of conduct can be created for any project, with pre-written licenses available right from the repository.

Teams enable the ability to organize people, provide level-up access with administrative roles, and tune permissions for nested teams. Discussion threads keep conversations on topic using moderation tools, like issue and pull-request locking, to help teams stay focused on code. For maintaining open-source projects, user blocking reduces noise and keeps conversations productive.

Documentation

GitHub allows for documentation to be created and maintained in any repository, and wikis are available to create documentation with version control. Each wiki is its own repository, so every change is versioned and comparable. With a text editor, users can add documents in the text formatting language of choice, such as Textile or GitHub Flavored Markdown.

GitHub Enterprise Cloud service with Data Residency

GitHub Enterprise Cloud service with Data Residency is GitHub Enterprise Cloud service's platform with robust data residency features for enterprises. With improved enterprise-grade features and more control over where code is stored, GitHub Enterprise Cloud service with Data Residency will help more enterprise customers meet their security and compliance needs and addresses:

- The ability to store code and repository data in a preferred region

- Enhanced user control, allowing organizations to manage and control user accounts
- Unique namespaces specific to a company on ghe.com isolated from the open-source community
- Enhanced availability and support for zone-based business continuity and disaster recovery

GitHub Copilot

GitHub Copilot is an artificial intelligence (AI)-powered coding assistant that helps developers write code faster. GitHub Copilot (including the Copilot for Non– GitHub Enterprise Cloud service customers product offering) is available through:

- GitHub Copilot Business for GitHub organization or enterprise accounts, which gives control over Copilot policies, including which members can use Copilot
- GitHub Copilot Enterprise for enterprise accounts on GitHub Enterprise Cloud service, which includes all Copilot Business features along with additional AI features on GitHub. With this subscription plan users can choose to assign either Copilot Enterprise or Copilot Business to each individual organization in the enterprise.

GitHub Copilot features include, but are not limited to, the following.

Copilot Feature	Short Description
Code Completion	Autocomplete-style suggestions from Copilot in supported Integrated Development Environments (IDEs) (Visual Studio Code, Visual Studio, JetBrains IDEs, Azure Data Studio, and Vim/Neovim).
Copilot Chat	A chat interface that lets users ask coding-related questions. GitHub Copilot Chat is available on the GitHub website, in GitHub Mobile, and in supported IDEs (Visual Studio Code, Visual Studio, and JetBrains IDEs). Users can also use skills with Copilot Chat.
Copilot in the CLI	A chat-like interface in the terminal, where users can ask questions about the command line. Users can ask Copilot to provide command suggestions or explanations of commands.
Copilot pull request summaries	AI-generated summaries of the changes that were made in a pull request, which files they impact, and what a reviewer should focus on when they conduct their review.
Copilot Extensions	Enables developers to build and deploy to the cloud with their preferred tools and services from the GitHub Marketplace or through private tooling created by organizations. Extensions are supported in GitHub Copilot Chat on GitHub.com, in Visual Studio, and in Visual Studio Code.
Copilot text completion (beta) (Copilot Enterprise only)	AI-generated text completion to help you write pull request descriptions quickly and accurately.
Copilot knowledge bases (Copilot Enterprise only)	Create and manage collections of documentation to use as context for chatting with Copilot. When users ask a question in Copilot Chat on GitHub.com or in Visual Studio Code, users can specify a knowledge base as the context for their question.

GitHub Copilot Administrator features include, but are not limited to, the following.

Copilot Feature	Short Description
Policy management	Manage policies for Copilot in the user's organization or enterprise.
Access management	Enterprise owners can specify which organizations in the enterprise can use Copilot, and organization owners can specify which organization members can use Copilot.
Usage data	Review Copilot usage data within an organization or enterprise to inform how to manage access and drive adoption of Copilot.
Audit logs	Review audit logs for Copilot in an organization to understand what actions have been taken and by which users.
Exclude files	Configure Copilot to ignore certain files. This can be useful if users have files that they don't want to be available to Copilot.

GitHub Actions

GitHub Actions automates continuous integration/continuous delivery (CI/CD) software workflows by enabling the build, test, and deployment of code directly from GitHub, with code reviews, branch management, and issue triaging customized to work the way developers need.

GitHub Actions initiates workflows for events like push, issue creation, or a new release, and actions can be combined and configured for the services used, built, and maintained by the community.

GitHub Actions supports additional options to do things like build containers, deploy web services, or automate notifications to users of open-source projects using GitHub developers' existing GITHUB_TOKEN in collaboration with other Enterprise Cloud features. GitHub Actions is available on hosted runners for major operating systems (OSs) (Linux, macOS, Windows) and silicon (x86, Advanced RISC Machine [ARM], graphics processing unit [GPU]). GitHub Actions can run directly on a virtual machine (VM) or inside a container. Customers can use their own VMs, in the cloud or on-premises, with self-hosted runners.

GitHub Advanced Security

GitHub Advanced Security provides features that help improve and maintain the quality and security of code. Code scanning searches for potential security vulnerabilities and coding errors in code using CodeQL or a third-party tool. Secret scanning detects secrets (e.g., keys and tokens) that have been inadvertently checked into repositories. If push protection is enabled, secret scanning also detects secrets and blocks contributors from pushing them to repositories. Dependency review and Dependabot detect vulnerable versions of dependencies and warn about the associated security vulnerabilities. Alerts from all features can be centrally reported and tracked.

System Boundaries

The scope of this report includes GitHub's Enterprise Cloud, and the supporting production systems, infrastructure, software, people, procedures, and data. The following Enterprise Cloud features are included in the scope of this report: Issues, Discussions, Pages, Projects, Docs, GitHub Advanced Security, GitHub Teams, Dependabot, Copilot, GitHub Enterprise Cloud service with data residency, and GitHub Actions, as well as pull requests, wikis, and audit logging.

Subservice Organizations

GitHub uses multiple subservice organizations in conjunction with providing its Enterprise Cloud product. GitHub uses subservice organizations to provide colocation data center services, as well as to provide infrastructure hosting. These subservice organizations are excluded from the scope of this report. The expected controls for which they are responsible are found in a subsequent section titled Subservice Organizations.

The system description in this section of the report details GitHub Enterprise Cloud service and its associated features noted above. Any other Company services are not within the scope of this report. The accompanying description includes only the policies, procedures, and control activities at the Company and does not include the policies, procedures, and control activities at any subservice organizations (see below for further discussion of the subservice organizations).

The Components of the System Used to Provide the Services

The boundaries of GitHub Enterprise Cloud service are the specific aspects of the Company's infrastructure, software, people, procedures, and data necessary to provide its services and that directly support the services provided to customers. Any infrastructure, software, people, procedures, and data that indirectly support the services provided to customers are not included within the boundaries of GitHub Enterprise Cloud service.

The components that directly support the services provided to customers are described in the subsections below.

Infrastructure

The Company utilizes a third party cloud service provider and GitHub-managed colocation data centers to provide the resources to host GitHub Enterprise Cloud service. The Company leverages the experience and resources of the third party cloud service provider to scale quickly and securely as necessary to meet current and future demand. However, the Company is responsible for designing and configuring the GitHub Enterprise Cloud service architecture within the third party cloud service provider and GitHub-managed colocation data centers to ensure security and resiliency requirements are met.

The in-scope hosted infrastructure also consists of multiple supporting tools.

Software

Software consists of the programs and software that support GitHub Enterprise Cloud service (OSs, middleware, and utilities). The list of software and ancillary software used to build, support, secure, maintain, and monitor include the following applications:

- Solutions to assist in development, deployment, and scalability
- Network architecture configuration and management
- Application and infrastructure monitoring
- CI/CD
- Dependency management, static code analysis, secret scanning
- Endpoint management and security

- In-house developed tool for data center asset inventory and management
- Endpoint security
- Account and license storage
- Single sign-on (SSO) configuration management
- Endpoint inventory and management
- Infrastructure configuration management
- Cloud asset inventory
- Chat Operations (ChatOps) and daily work communications
- Security information and event management (SIEM), logging system, intrusion detection
- Vulnerability scanning
- OS baseline
- Customer support
- AI models

People

The Company develops, manages, and secures GitHub Enterprise Cloud service via separate departments. The responsibilities of these departments are defined in the following table:

People	
Group/Role Name	**Function**
Customer Support	Responsible for providing technical and account-related support to GitHub Enterprise Cloud service customers and for resolving customer issues via email, chat, social media, and phone from developers and customer entities around the globe.
Engineering	Responsible for working with the Product Management team to plan and coordinate releases, and accountable for building, testing, and deploying GitHub Enterprise Cloud service code and feature changes. Responsible for maintaining service availability, including performance and scale monitoring and reporting, incident command, and on-call readiness for any production issues. Responsible for configuration management; building, testing, and deploying software relevant to the operation and management of production assets; patching and remediating vulnerabilities reported by the Security team; and data center operations management. Responsible for managing Git and database storage backups and restores.
Legal	Responsible for negotiating contractual obligations with third parties and technology partners/suppliers, legal terms and conditions, and ensuring compliance with internal contractual standards.
People Operations	Responsible for talent acquisition, diversity and inclusion, learning and development, and employee engagement on everything from benefits and perks to career development and growth.
Privacy	Responsible for determining which privacy laws and regulations apply to GitHub and determining the best way to comply with them, ultimately ensuring GitHub can offer its products to every developer anywhere in the world.

People	
Group/Role Name	**Function**
Product Management	Responsible for understanding customer requirements; collecting, defining, and clarifying feature requests and development efforts; and managing feature rollouts and related customer communication efforts.
Security	Responsible for ensuring the security of GitHub products. Security consists of multiple teams with specific missions: Threat Hunting Operations and Response Incident Response (THOR IR); Product Security Incident Response Team (PSIRT); Security Lab; Security Operations; Secure Access Engineering; Security Telemetry; Vulnerability Management; Cloud and Enterprise Security and Governance, Risk, Compliance; and Communication (GRCC). These teams manage security incident detection and response, monitoring, vulnerability scanning, network and application layer penetration testing, security architecture, security engineering and operations, access management, endpoint asset management, and risk and compliance oversight.
Senior Leadership	Responsible for the overall governance of GitHub. This group includes the Chief Executive Officer (CEO), Chief Financial Officer (CFO), Chief Revenue Officer (CRO), Chief Technology Officer (CTO), Chief People Officer (CPO), Head of Design, Chief Operations Officer (COO), Chief of Staff, Vice President, Senior Vice President of Engineering, Vice President - Chief Information Security Officer, Chief Product Officer, Chief Legal Officer, and Vice President of Communities.

Policies, Standards, and Procedures

GitHub maintains policies, standards, and procedures necessary to securely operate GitHub Enterprise Cloud service. Policies and standards are centrally managed in The Hub, GitHub's centralized internal communication platform. The Hub is backed by a repository, which is used to implement annual reviews and control changes to policies and standards. Once a change has been approved by the owner of the policy or standard, it is automatically updated on The Hub. Procedures are developed and documented within the GitHub repositories maintained by every team to provide end-user documentation and guidance on the multitude of operational functions performed daily by GitHub security and product engineers, developers, administrators, and support personnel. These procedures are drafted in alignment with the overall policies and standards and are updated as necessary to reflect changes in the business.

GitHub policies and standards establish controls to enable security, efficiency, availability, and quality of service. The GitHub Information Security Management System (ISMS) Policy and related policies define information security practices, roles, and responsibilities. The ISMS outlines the security roles and responsibilities for the organization and expectations for employees, contractors, and third parties utilizing GitHub systems or data.

This overarching security policy is supported by several dependent security policies, standards, and procedures applicable to the operation and management of security across the organization. Security-related policies, standards, and procedures are documented and made available to individuals responsible for their implementation and compliance.

Below is the current inventory of security and audit-related policies and standards that inform procedures operating in support of the GitHub ISMS Policy objectives:

Procedures	
Policy	**Associated Standards and Procedures**
GitHub ISMS	No associated standards or procedures
GitHub ISMS Scope	No associated standards or procedures
GitHub ISMS Statement of Applicability (SOA)	• ChatOps Command Security and Risk Standard • Controls Monitoring Standard • Controls Monitoring Standard Operating Procedure (SOP) • Data Classification Standard • Domain Management Standard • Endpoint Security Standard • Enterprise Administration Standard • External File Sharing Standard • Git Systems Server Site Failure plan • High-Risk Application Access Standard • Organization Administration Standard • Production VPN Access Standard • Repository Security Baseline Configuration Standard • Reviewing Pull Requests • Server Operating System Standard
Corporate Data Retention Policy	• Audit Video Retention Standard • Corporate Data Retention Standard • Product Telemetry Data Retention Standard • Slack Retention Standard
Contractor Termination Policy	No associated standards or procedures
Full-Time Employee Termination Policy	No associated standards or procedures
Identity and Access Management (IAM) Policy	IAM Standard • IAM Onboarding SOP • IAM Entitlements SOP • IAM Privileged Systems and Elevated Access SOP • Granting Slack Access to Contractors and Consultants SOP • IAM Non-Human Accounts in Okta • IAM Offboarding SOP • IAM On-Leave SOP
Physical and Environmental Protection Policy	Production Datacenter Standard • Datacenter Physical Access SOP • Production Media Destruction SOP • Datacenter Access compliance guidelines
Privacy Statement	No associated standards or procedures

Procedures	
Policy	**Associated Standards and Procedures**
Private Information Removal Policy – External Customer Facing Policy	No associated standards or procedures
Secure Coding Policy	Secure Coding Standard • Secure Coding – Dotcom • Secure Coding – General Guidance • Security Requirements for New Applications
Security Awareness and Privacy Training Policy	Security Awareness Training Standard
Security Event Logging and Monitoring Policy	• Security Event Logging Standard • Security Event Logging SOP • Security Event Monitoring and Alerting Standard
Security Incident Response and Data Breach Notification Policy	• Data Breach Notification Standard • Security Incident Response Standard • Security Incident Response Procedure • Data Breach Notification Procedure • Security Concern Reporting Procedure
Security Policy Exception Policy	No associated standards or procedures
Security Risk Management Policy	• Security – GRCC Vendor Risk Assessment Process • Security Risk Reporting – Standard Operating Procedure
System and Services Acquisition Policy	• Vendor Security Standard • Purchasing Workflow • Vendor Security Reviews SOP • Procurement Workflow • Vendor Offboarding Checklist • Decommissioning a GitHub-Owned App • Vendor Offboarding SOP • Encryption Standard
Vulnerability Management Policy	• Container Hardening Standard • Exception Handling Process • Vulnerability Management Process • Checklist for Docker Baseline Security • Database Hardening Standard • OS Hardening Standard • Patch Management Standard • FedRAMP Vulnerability Reporting Standard • FedRAMP Annual Vulnerability Exception Review • FedRAMP Monthly Vulnerability Management Reporting Procedure
Background Checks Policy	No associated standards or procedures

Procedures	
Policy	**Associated Standards and Procedures**
IT Asset Management Policy	No associated standards or procedures
Network Policy	No associated standards or procedures
Resilience Program Policy	Resiliency Standard
Security Document Management Policy	Security Document Management Standard

Data

Data refers to transaction streams, files, data stores, tables, and output used or processed by the Company. GitHub uses repository data to connect users to relevant tools, people, projects, and information. Repositories are categorized as public, private, or open source. Public repositories can be viewed by anyone, including people who are not GitHub users. Private repositories are only visible to the repository owner and collaborators that the owner specified. GitHub aggregates metadata and parses content patterns to deliver generalized insights within the product. It uses data from public repositories and uses metadata and aggregate data from private repositories when a repository's owner has chosen to share the data with GitHub through an opt-in.

If a private repository is opted in for data use to take advantage of any of the capabilities of the security and analysis features, then GitHub will perform a read-only analysis of that specific private repository's Git contents. If a private repository is not opted in for data use, its private data, source code, or trade secrets are classified internally as restricted, and they are maintained as confidential and private consistent with GitHub's Terms of Service.

Customer data is managed, processed, and stored in accordance with relevant data protection and other regulations and with specific requirements formally established in client contracts.

The Company has deployed secure methods and protocols for transmission of confidential or sensitive information over public networks.

Subservice Organizations

The Company uses subservice organizations for data center colocation services and infrastructure hosting. The Company’s controls related to GitHub Enterprise Cloud service cover only a portion of the overall internal control for each user entity of GitHub Enterprise Cloud service. The description does not extend to the colocation services for IT infrastructure provided by the subservice organizations.

Although the subservice organizations have been carved out for the purposes of this report, certain service commitments, system requirements, and applicable criteria are intended to be met by controls at the subservice organizations. Controls are expected to be in place at the subservice organizations related to physical security and environmental protection. The subservice organizations’ physical security controls should mitigate the risk of unauthorized access to the hosting facilities. The subservice organizations’ environmental protection controls should mitigate the risk of fires, power loss, climate, and temperature variabilities.

The Company management receives and reviews the subservice organization's SOC 2 reports, International Organization for Standardization (ISO) 27001 and 27701 Certifications, and Payment Card Industry Attestation of Compliance (PCI AOC) as they are issued, and at least annually. In addition, through its operational activities, Company management monitors the services performed by the subservice organizations to determine whether operations and controls expected to be implemented are functioning effectively. Management also communicates with the subservice organizations to monitor compliance with the service agreement, stay informed of changes planned at the hosting facility, and relay any issues or concerns to management of the subservice organizations.

Attachment B

Principal Service Commitments and System Requirements

Principal Service Commitments and System Requirements

Commitments are declarations made by management to customers regarding the performance of GitHub Enterprise Cloud service. System requirements are the specifications about how GitHub Enterprise Cloud service should function. System requirements should:

- Support service commitments, ensuring it functions in a way that fulfills promises made to users, their customers, vendors, and business partners
- Follow rules and regulations, operating within all relevant laws and any guidelines set by industry groups
- Meet additional goals, helping the organization achieve other objectives important to building trust with those it serves

The Company's principal service commitments and system requirements related to GitHub Enterprise Cloud service include the following across all products as a foundation, and subsequent sections building upon that foundation per product:

Trust Services Category	Service Commitments
GitHub Enterprise Cloud service	
Security	• The GitHub leadership team will continually establish reasonable and appropriate security measures, along with identifying and mitigating risks to the entity and customers. • The GitHub leadership team is committed to building a culture of security within GitHub and the software development and open-source communities. • GitHub attracts, develops, and attempts to retain competent individuals in all areas of the organization to accomplish effective technical and human security measures. • GitHub designs reasonable and appropriate security controls to effectively mitigate risks and conducts periodic internal and external assessments to analyze the design and operating effectiveness of the security controls. Where control design or operational flaws are identified, GitHub works to remediate those flaws in a timely manner. • GitHub will utilize reasonable and appropriate security measures related to configuration and software development change management, highlighted specifically by the core use cases, and features directly built into the GitHub platform. • GitHub will utilize reasonable and appropriate security and legal measures to manage vendor and business partner risks. • GitHub will utilize reasonable and appropriate security measures to safeguard sensitive information against unauthorized access, use, modification, destruction, or disclosure. • If GitHub becomes aware of a security incident, GitHub will notify the customer without undue delay.
GitHub Enterprise Cloud service with Data Residency	
Security	• GitHub is committed to taking appropriate measures to ensure core customer content, such as repositories, user-generated content, and GitHub Actions data, is stored within the chosen region. • GitHub will utilize Azure availability-zone-based disaster recovery capabilities to ensure customers have access to their code and data in the regions they select.

Trust Services Category	Service Commitments
GitHub Copilot	
Security	• GitHub Copilot Business/Enterprise communications will be encrypted in transit between GitHub and the customer, using Transport Layer Security (TLS) version 1.2 and above with HTTP Strict Transport Security (HSTS) enabled. • As documented in the Copilot Product Terms, prompts will not be retained by GitHub when using Copilot Business Chat and Code Completion in the IDE. • Copilot Business will not retain suggestions produced by Copilot when using Copilot Business Chat and Code Completion in the IDE. • Copilot Business will not retain user content (e.g., code from the user's editor) when using Copilot Business Chat and Code Completion in the IDE. • As documented in the Copilot Product Terms, GitHub will allow customer administrators to enable the Duplicate Detection feature to filter suggestions that match existing public code. • Copilot Business does not use the customer's code to improve the model. • Copilot Business does not send information to OpenAI, the software company.

The Company designs its processes and procedures to provide a secure environment for customer data. GitHub's security commitments and system requirements are documented and communicated to customers in the Data Protection Agreement and at the other resources listed below:

- Security at GitHub: https://github.com/security
- GitHub General Privacy Statement: https://help.github.com/en/articles/github-privacy-statement
- Customer Terms: https://github.com/customer-terms