cloud security alliance®

SaaS Security Capability Framework
Working Group

Control Framework

SaaS Security Capability Framework (SSCF)

Acknowledgments

Co-Chairs

Romke de Haan
Boris Sieklik
Jonathan Villa

Lead Authors

Romke de Haan
Boris Sieklik
Jonathan Villa
Idan Fast
Dennis Faire
Joseph Longo
Shlomi Matichin
Uli Petersen
Daniel Rosenberg
Brian Soby

Contributors

Adam Fisher (Team Lead)
Sing Ambikapathi
Rohit Bansal
Sergei Beliachkov
Ken Cody
Eduard Hurtos
Kane Narraway
John B. Oseh
Govindaraj Palanisamy
Michael Roza
Anh Rucker
Vic Salemme
Heinrich Smit
Srinivas Tatipamula
David Tessier

CSA Global Staff

Andy Ruth
Lefteris Skoutaris

Graphic Design

Stephen Lumpe
Stephen Smith

The permanent and official location for the SaaS Security Capability Framework (SSCF) Working Group is: https://cloudsecurityalliance.org/research/working-groups/saas-security-capability-framework-sscf

Table of Contents

Executive Summary

The widespread adoption of Software-as-a-Service (SaaS) applications has fundamentally transformed how organizations deliver business solutions. This transformation introduces significant security challenges due to a critical gap in existing Third-Party Risk Management (TPRM) processes, which primarily concentrate on suppliers' organizational security rather than evaluating the security capabilities of individual SaaS applications. This oversight leads to substantial security exposure, as SaaS applications often lack standardized security capabilities that application owners can consistently configure or consume. The absence of such standardization results in misalignment between vendors, application users, security, and TPRM teams, leading to delays, wasted resources, and unnecessary risk exposure.

To address these challenges, the Cloud Security Alliance (CSA), in collaboration with the CSA SaaS Working Group (which includes diverse industry players such as MongoDB and GuidePoint Security), developed the SaaS Security Capability Framework (SSCF). This technical framework outlines configurable, consumable, and customer-facing security controls provided by SaaS vendors to their customers. The SSCF was designed to serve three purposes:

- For **TPRM teams**, to serve as a baseline of security capabilities during SaaS vendor assessment, simplifying risk assessments and procurement processes
- For **SaaS vendors**, to standardize assessment responses by serving as a consistent framework, reducing custom questionnaires and assessment overhead
- For **SaaS Security Engineering teams**, to provide a baseline implementation checklist, streamlining and accelerating the SaaS security program

The foundation of the SSCF is its emphasis on SaaS provider and customer responsibilities, aligning with the Shared Security Responsibility Model (SSRM). The SSCF covers critical security domains, including logging mechanisms, identity and access management, management control policies, secure authentication protocols, and API management.

The SSCF enhances existing certifications by translating high-level principles for the SaaS vendor into actionable security capabilities that customers can directly configure and enforce. It complements established compliance frameworks, such as SOC 2 and ISO 27001, by balancing robust security with the resource constraints of early-stage SaaS companies. It provides enforceable specifications alongside flexible implementation guidelines.

1. Introduction

1.1 What is the SSCF?

Organizations are increasingly adopting Software-as-a-Service (SaaS) applications. Regardless of the security controls a SaaS provider implements within its organization and products, the customer adopting the SaaS platform is ultimately responsible for operating it securely. This includes activities such as monitoring audit logs, ensuring secure configuration, conducting access reviews, and managing their identities. Currently, inconsistent security capabilities among SaaS providers hinder the ability of customers to implement security controls at scale. This in turn leads to operational complexities, increased risk, and burdensome vendor assessments.

To systematically address these challenges and foster a more secure SaaS ecosystem, the Cloud Security Alliance (CSA), in collaboration with industry players such as MongoDB and GuidePoint Security, and drawing on the collective expertise of the CSA SaaS Working Group, has developed the SaaS Security Capability Framework (SSCF).

The SSCF is meant to serve as a baseline of configurable, consumable, and customer-facing security controls and documentation provided by SaaS vendors to their customers. It complements established compliance frameworks, such as SOC 2 and ISO 27001, by highlighting controls whose implementation is typically owned by the customer in the Shared Security Responsibility Model (SSRM).

1.2 Problem Statement

The SaaS ecosystem lacks a defined, technical-level industry standard for the minimum required security capabilities of SaaS applications. This presents considerable challenges for SaaS customers and SaaS providers, including large enterprises, established SaaS providers, and less mature SaaS startups.

Large enterprises, which oversee hundreds of SaaS applications, encounter operational complexity, heightened security risks, and extensive vendor assessment burdens due to the diverse configurability, visibility, and integration options among providers. This creates friction in vendor onboarding, compliance assurance, and risk management. Meanwhile, SaaS startups aiming to attract enterprise clients struggle to identify and implement the most impactful security features to satisfy procurement requirements.

1.3 Scope and Objectives

The SSCF enables SaaS providers and customers to systematically assess, establish, and mature their security posture for SaaS applications. It enables customers to have consistent security features on their side of the shared responsibility model.

1.3.1 Scope

The scope of the SaaS Security Controls Framework (SSCF) focuses on customer-facing security controls within SaaS platforms and services. These are controls that can be directly influenced, managed, or utilized by SaaS customers, or that support them in fulfilling their security implementation responsibilities under the Shared Security Responsibility Model (SSRM). Examples include capabilities such as logging, access monitoring, and configuration of security settings.

By providing standardized controls, SaaS vendors empower customers to operate their products securely. These controls provide visibility into the security posture and settings of SaaS tenants, offer actionable audit trails of user activity, and enable the scalable and cost-effective implementation of security control, helping organizations proactively safeguard their cloud footprint.

The SSCF addresses security features and functionalities that are not comprehensively covered by existing standards or frameworks. To maintain relevance and avoid redundancy, the framework intentionally excludes topics that are already well-defined by established industry standards, as well as those that are not visible to the customer via the SaaS application or part of the customer's shared security implementation responsibilities. For example:

- **Encryption mechanisms for data at rest:** Encryption mechanisms for data stored and processed within the SaaS provider's infrastructure are controlled by the SaaS vendor. Encryption is covered in depth by existing standards, compliance frameworks, and contractual obligations.
- **Monitoring of the application backend infrastructure:** Monitoring of the application backend is typically the responsibility of the SaaS vendor.
- **Data usage (e.g., in AI features):** Any reuse of the customer data by the vendor should be covered contractually, as required by various privacy regulations (e.g., DPA).
- **Process support (e.g., incident handling):** Process support is typically handled in the security procedures within the SaaS vendor's shared responsibility and is contractually defined with the customer.
- **GenAI configuration:** While some platforms do offer controls around the configuration of their GenAI features, such as AI assistants and AI copilots, this is a use-case-specific and product-specific approach and is difficult to apply across all SaaS platforms. As an example,

some SaaS platforms, such as AI chatbots, are AI-centric, requiring configuration to turn off AI that does not apply to these vendors. Therefore, the SSCF does not define controls specific to GenAI. These can, however, be found in other standards, such as AICM.

- **Data model controls:** Some controls are specific to certain data models. For example, when dealing with applications that contain file storage features, a common capability is to label files according to the sensitivity of the data they contain. While this may be a desirable capability in some platforms, it is not generically applicable to all applications with file storage features. For example, SaaS platforms such as ticketing or purchasing systems typically do not label files for sensitivity, despite having file storage features. We chose to exclude features that lack consistent usage and security applicability and that are not generic across all SaaS platforms, from the first version of this standard.
- **Data residency:** Data residency is typically contractually enforced and is a standard part of existing security frameworks.

1.3.2 Objectives

The core objective of the SSCF is to provide guidance and set a standard on expected customer-facing security controls within SaaS applications.

1.4 Assessment Guidelines

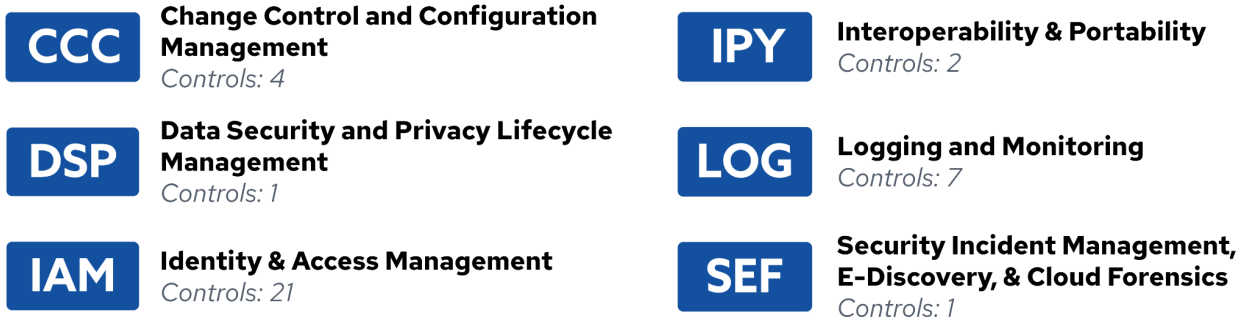
The SSCF offers a consistent approach for implementing security controls across key domains. The SSCF is designed to ensure that security capabilities are implemented effectively and measured consistently. This standard does not aim to dictate low-level implementation details of each control. Therefore, the methodology for assessing SaaS applications against the SSCF should consider using the following as a basis for performing assessments:

- Assessments using this framework should be designed to be scalable and adaptable in order to accommodate both small SaaS implementations and large enterprise platforms with complex integrations
- The SSCF provides Control Specifications and Implementation Guidelines in order to balance prescriptive security requirements with the flexibility to address diverse business models, technology stacks, and regulatory landscapes
    - **Control Specifications** are mandatory requirements that are either implemented, not implemented, or not applicable to the SaaS platform
    - **Implementation Guidelines** are recommended best practices that should be used but are not a mandatory requirement

An assessment framework for evaluating controls against the SSCF standard is on the project roadmap and is expected to be released at a later date.

1.5 SSCF Structure

This section displays an image illustrating the security domains in version 1.0 of the SSCF, along with the number of controls included.



1.6 SSCF Domain Descriptions

The SSCF v1.0 includes six SaaS security domains, aligning with the CSA Cloud Control Matrix (CCM) v4 domain naming conventions. These domains are listed below, along with a description of each one's unique purpose and use.

1.6.1 Change Control and Configuration Management (CCC)

Change Control and Configuration Management (CCC) are critical components in a SaaS environment. They ensure stability, compliance, and reliability while supporting agility and innovation. Implementing robust and secure CCC controls reduces the risks associated with system changes, updates, and misconfigurations, which are leading causes of security incidents and data breaches in SaaS environments.

Under the Shared Security Responsibility Model (SSRM), the SaaS provider owns the platform and is responsible for ensuring changes to the application and infrastructure are secure, tested, and do not disrupt service. The SaaS customer, though not responsible for the application itself, must manage how it's configured and used, particularly with regard to settings that affect security, compliance, and user experience. The SaaS customer is responsible for setting secure configurations in the SaaS application (e.g., enforcing MFA, session timeout settings); continuously reviewing settings to maintain alignment

with emerging threats, security policies, and regulatory needs; and ensuring integrations and APIs are configured securely.

1.6.2 Data Security and Privacy Lifecycle Management (DSP)

Data Security and Privacy Lifecycle Management (DSP) is a fundamental area of SaaS controls. Typically, most of the tasks relating to data security are handled by the SaaS provider. Given the variety of potential SaaS use cases, it is challenging to create a set of SaaS customer-specific controls that would universally apply to all SaaS applications. Nevertheless, the SSCF requires customer-facing controls for file upload functionality. SaaS customers are responsible for utilizing these to enable or disable file uploads and define acceptable file types.

1.6.3 Identity and Access Management (IAM)

Identity and Access Management (IAM) domain controls are paramount within SaaS applications. These controls are foundational to protecting customer data and ensuring the integrity and availability of the SaaS platform itself. IAM settings vary depending on the SaaS provider, but the configuration of these settings is the responsibility of SaaS customers. Within the SSCF, the IAM section contains several controls that allow SaaS customers to increase the security of identities used in their SaaS platforms. By implementing these IAM security controls, SaaS customers build trust, meet compliance requirements, and significantly reduce the risk of unauthorized access or data breaches, which can have severe reputational and financial consequences. Effective IAM controls enable secure user provisioning, authentication, authorization, and auditing, forming the fundamentals of a secure SaaS offering.

1.6.4 Interoperability and Portability (IPY)

Interoperability and Portability (IPY) in SaaS applications often translates to the ability to massively export data out of the platform. These large data exports are often a crucial element of customer exit strategies, but also represent a security risk, so the SSCF mandates a configuration around these features when available.

In addition, many SaaS applications encourage connectivity with other applications but often lack support for the security team to perform governance of that interconnectivity. The SSCF requires basic integration attribution to support security teams' governance of integrations.

1.6.5 Logging and Monitoring (LOG)

Effective logging and monitoring control implementation is critical to allow for visibility, traceability, and accountability within a SaaS product. Under the SSRM, logging and monitoring responsibilities are distributed between the SaaS service provider and the SaaS service customer. The provider is accountable for implementing controls over the underlying infrastructure, application layer, and platform services. Meanwhile, the customer is responsible for configuring tenant-level settings, monitoring activities of their users and connected accounts, and responding to security alerts relevant to their environment.

By providing relevant security events and system activity to SaaS customers, the SaaS providers enable customers to detect abnormal behaviors, detect potential incidents, and maintain compliance with regulatory and contractual obligations. Continuous monitoring enables proactive threat detection, supports forensic analysis, and contributes to the overall resilience of SaaS platforms. It also plays a vital role in meeting audit requirements by maintaining a detailed and tamper-evident record of system events. As the threat landscape evolves, robust logging and monitoring become foundational to timely response and recovery.

1.6.6 Security Incident Management, E-Discovery, and Cloud Forensics (SEF)

A swift and effective response to security incidents relies on a clear partnership between the SaaS provider and the customer, as defined by the SSRM. While the provider is responsible for detecting and managing incidents within its infrastructure, it is imperative that standardized mechanisms exist to ensure timely communication and actionable guidance for affected SaaS customers. Therefore, the SSCF requires providers to allow setting a security contact that will be notified during incidents, enabling rapid collaboration to contain threats and minimize impact.

1.7 SSCF Components and Spreadsheet Structure

This section describes the components of the SSCF, as they appear in this document and the accompanying spreadsheet. A change log section is also present for version control purposes and tracking changes to the standard.

For version 1.0, the primary component of the SSCF is the list of controls. This means that the initial release will focus on defining the specific security capabilities that SaaS vendors should implement and that enterprise customers should expect to be configurable and usable.

The overall framework structure is presented in a tabular format, detailing each control across various attributes, including Control Domain, Control Title, Control ID, Control Specification, and Implementation Guidelines. This systematic organization aims to ensure that both SaaS providers and customers can efficiently navigate, understand, and apply the framework's security capabilities.

1.7.1 Controls and Implementation Guidelines

The SSCF v1.0 provides a detailed overview of the essential security controls required for SaaS environments. Organized in a tabular format, each row defines a unique security control with key attributes. These attributes include the Control Domain, categorizing the control within a broader security area, a specific Control Title, and a unique Control ID for easy reference.

Further details are provided through the Control Specification, which offers a comprehensive understanding of the technical requirements and scope of each control. The SSCF is formatted such that the controls represent "what" **must** be implemented and constitute requirements, and the Implementation Guidelines represent "how" controls **should** be implemented and constitute recommendations in the form of best practices.

The Implementation Guidelines section within the SSCF provides practical recommendations and actionable steps for effectively implementing the framework's controls. These guidelines are a collaborative effort, drawing on the collective experience and insights of SaaS Service Providers (SSPs) and SaaS Service Customers (SSCs) who are actively involved in securing SaaS applications. This collaborative approach aims to help organizations tailor and apply the SSCF controls within their specific SaaS environments.

It is essential to recognize that while these guidelines offer valuable insights and articulate best practices, they are not intended to serve as a rigid, prescriptive how-to manual. The precise operationalization of the SSCF controls will inherently depend on a multitude of factors unique to each organization. These factors include, but are not limited to, their specific SaaS architecture, the diverse types of technologies employed, the prevailing risk landscape, applicable regulatory requirements, and their internal organizational policies and procedures.

Consequently, the CSA does not provide highly detailed or universally prescriptive guidance that would apply to every single organization or SaaS service implementation. Instead, these guidelines are designed to serve as a flexible and adaptable resource, empowering security practitioners to tailor their implementation strategies to effectively meet their unique security objectives and address their specific operational contexts.

1.8 Target Audience

The SSCF is designed to help SaaS vendors standardize SaaS customer controls. Once implemented, these controls serve a diverse audience within the cybersecurity and business landscape.

On the SaaS customer side, it primarily caters to security professionals, compliance officers, risk management teams, and decision-makers responsible for assessing and managing SaaS vendors and their associated security postures.

The framework is particularly valuable for organizations that manage extensive SaaS portfolios, providing them with a standardized methodology to:

- Evaluate security responsibilities
- Implement appropriate security controls
- Maintain a consistent security posture across various SaaS environments

On the SaaS vendor side, it provides a standardized approach to controls required by larger enterprise customers. For smaller SaaS vendors, this can translate into fewer resources required for supporting varying customer requirements.

By offering a clear baseline for SaaS security, the SSCF aims to simplify the SaaS procurement process, alleviate the assessment burden on security teams, and ultimately enhance enterprise security in complex multi-cloud settings, while providing clarity on security requirements to SaaS vendors.

The broader objective of the SSCF is to foster trust, efficiency, and integrity within the global SaaS ecosystem by establishing standardized security practices.

1.9 Shared Security Responsibility Model (SSRM)

The SaaS Shared Security Responsibility Model (SSRM) is a framework that divides security duties between the SaaS provider and the SaaS customer. The SaaS provider is responsible for the security of the cloud, meaning they secure the core application and the infrastructure it runs on. The SaaS customer is responsible for security in the cloud. This includes securing their own data, managing user accounts and access, and correctly configuring the security settings provided within the service. This clear separation of roles ensures that all security responsibilities are owned, preventing gaps in protection.

Within the context of the SSCF, the SaaS provider is responsible for implementing the security controls highlighted in this standard and making them available to SaaS customers. The SaaS customers are responsible for utilizing and configuring these controls appropriately.

The SSCF controls are configurable, consumable, and customer-facing.

1.10 Document Control and Version History

Version	Date	Author(s)	Description of Changes	Approved By
1.0	2025-09-24	Idan Fast Dennis Faire Romke de Haan Joseph Longo Shlomi Matichin Uli Petersen Daniel Rosenberg Boris Sieklik Brian Soby Jonathan Villa	Initial release	WG co-chairs - Romke de Haan - Boris Sieklik - Jonathan Villa

2. SSCF Controls by Domain

This section lists the controls for each domain of the SSCF v1.0.

2.1 Change Control and Configuration Management (CCC)

Control ID	Control Title	Control Specification (Requirement / Must)	Implementation Guideline (Recommendation / Should)
CCC-SaaS-01	Programmatic Configurations Querying	The SaaS platform must support programmatic querying of all current security configurations. If these concepts exist on the SaaS platform, they must be readable via programmatic querying: • Authentication • RBAC assignments • Entitlements • Permissions • Resource ACLs • Application-specific security concepts • Configurations affecting security log coverage (e.g., enabling/disabling streams)	The output of the API should be in a machine-readable format.

CCC-SaaS-02	Configurations Documentation	The SaaS platform must provide up-to-date documentation of all customer-visible, security-relevant configurations and must make it readily available to SaaS customers. If these concepts exist on the SaaS platform, configuration documentation must include, but not be limited to: • Authentication • RBAC assignments • Entitlements • Permissions • Resource ACLs • Application-specific security concepts • Audit configuration	Documentation should be available via the SaaS provider's website, within the platform, or on a community page. Public documentation is recommended. Documentation should also include versioning and change logs to support traceability and audit requirements. Any custom language or non-industry standard terms should be explained in detail. Documentation should include: • Security onboarding documentation • Focus on the impact of the security configuration • Add default security settings, if applicable, and describe dependencies between configurations, such as one configuration overriding another • Publish the customer responsibility matrix against the shared responsibility model, ensuring the customer is aware of what domain areas are supported by the SaaS platform
CCC-SaaS-03	New Configuration Updates	The SaaS platform must provide notifications about software updates, including new or existing security configuration options, to SaaS customers.	Updates should have a subscription mechanism that includes notification of relevant release documentation and changes made to the SaaS platform. It is recommended that console notifications are visible when these changes take place, in addition to a subscription mechanism.
CCC-SaaS-04	Security Configuration Guidelines	The SaaS platform must provide SaaS customers with best practice security guidelines for relevant security configurations of the SaaS platform.	Security configuration guidelines should consider best practices in domains like IDP configuration, interface configuration, principles like least privilege, secure SSO configuration, avoidance of

			long-lived sessions, and application policies. The SaaS platform should create a subscription mechanism that includes notification of relevant changes. Security configuration guidelines should be available via the SaaS provider's website, within the platform, or on a community page.

2.2 Data Security and Privacy Lifecycle Management (DSP)

Control ID	Control Title	Control Specification (Requirement / Must)	Implementation Guideline (Recommendation / Should)
DSP-SaaS-01	Blocking Malicious File Uploads	If the SaaS platform allows unrestricted file attachments, it must provide administrative configuration that limits the acceptable file types using an allow list and must provide an option to disable any file uploads.	If the platform exposes the ability to upload files externally (e.g., through a support portal), this control is meant to disallow files that may contain malicious code (e.g., office documents). We recommend that SaaS providers consider adding file scanning capabilities.

2.3 Identity and Access Management (IAM)

Control ID	Control Title	Control Specification (Requirement / Must)	Implementation Guideline (Recommendation / Should)
IAM-SaaS-01	User Access Visibility	The SaaS platform must have a user management service that allows administrators to identify users via both UI and programmatic means, as well as their authentication mechanisms.	The SaaS platform should include details like: • User login mechanisms • Last login • Last activity
IAM-SaaS-02	User Permissions Enumeration	The SaaS platform must support enumeration and programmatic querying of all assigned user	User permissions enumeration is SaaS platform-specific, and provider discretion is advised on

		entitlements. The platform must provide information about: • Access permissions • Roles • Groups • Application-specific entitlements • Data access entitlements • All entitlements for security configuration access	entitlements implementation. The SaaS platform should allow SaaS administrators to see the entitlements assigned to each user.
IAM-SaaS-03	Network Access Restriction	The SaaS platform must support restricting logins/access from outside a SaaS customer's network. The SaaS platform must offer a minimum of two distinct access rule sets, enabling customers to assign specific user groups to more stringent restrictions (e.g., to further limit administrator-level users to a narrower range of networks).	The SaaS platform should allow for IP restrictions to be separately applied for user logins and integrations, or other non-human connections, including APIs. For example, the platform may support IP allowlisting for a SaaS customer's instance or the use of a customer-assigned domain.
IAM-SaaS-04	Single Sign-On Support	The SaaS platform must support federated authentication using the most current version of an industry-standard protocol, such as SAML or OIDC. If SAML is used, then SSO support for SAML must include IdP-initiated and SP-initiated flows.	*This cell intentionally left blank*
IAM-SaaS-05	Single Sign-On Enforcement	The SaaS platform must support the option of disabling alternative login methods if federated authentication is enabled for users. The SaaS platform must be able to disable specific users from this enforcement.	The SaaS platform should allow administrators to set up break-glass accounts with alternative login methods, such as username and password, or sign in with a SaaS customer managed account (enterprise account).
IAM-SaaS-06	NHI Governance	The SaaS platform must support the identification of Non-Human Identities (NHIs) in UI and via programmatic means. The SaaS platform must identify NHIs, their type, source/target counterparties, NHI issuance date and	The SaaS platform should be able to differentiate between NHI types (e.g., service accounts), such as third-party integrations, AI agents, marketplace integrations, or custom integrations. The SaaS platform should provide

		expiration, if any, and entitlements.	programmatic access to additional attributes that affect the lifecycles of NHIs. For example: • Creation dates of the NHI • If applicable, the identity it is delegated from • Access expiration • Authentication type (e.g., secret key, certificate, username and password) This would also include application connections, such as users' tokens on mobile devices. The SaaS platform should show all entitlements assigned to NHIs (including actions they can take), and NHI accounts should have UI access disabled by default.
IAM-SaaS-07	NHI Revocation	The SaaS platform must support manual and programmatic revocation of Non-Human Identities (NHIs) by SaaS platform administrators and authorized SaaS platform users.	NHI revocation should ensure that session invalidation propagates across all access tokens, refresh tokens, and active sessions for the NHIs. An example of an authorized user would be a user who created the credential.
IAM-SaaS-08	User Credentials Management	The SaaS platform must support administrative control of all credentials issued to SaaS platform users. Administrative control refers to the ability to view, remove, and reset all authentication factors associated with users and user-provisioned credentials.	Examples of such credentials are user credentials (e.g., password, authenticator apps, application credentials issued in the context of users, passkeys, SMS based factor phone numbers) and user-provisioned credentials (e.g., SSH keys, OAuth refresh tokens, api keys, api tokens, OIDC). Changes in credentials should terminate all active sessions for the user and force re-authentication. Changes in permissions should take effect immediately. SaaS platform administrators should not have access to the private credentials (e.g., private keys, passwords, tokens) and the ability to gain access should not be

			possible.
IAM-SaaS-09	User Provisioning and Deprovisioning	The SaaS platform must support automated user provisioning and deprovisioning. The SaaS platform must have a mechanism to limit the programmatic access of user provisioning and deprovisioning operations.	The suggested implementation is SCIM. Alternative programmatic methods, such as API calls, are also permissible.
IAM-SaaS-10	Security Auditing Role	The SaaS platform must provide a Security Auditing role for read-only access to all security settings, including log access in UI and via programmatic means. The Security Auditing role must allow visibility into security configurations and logging data. It must not enable viewing or modifying customer data or making changes to any configurations.	*This cell intentionally left blank*
IAM-SaaS-11	Password Rules	The SaaS platform must define password strength requirements or configuration controls to comply with NIST guidelines. If implemented through configuration controls, the following must be configurable: • Set password length • Password reuse • Toggle special characters required • Password expiry	The SaaS platform can support password strength assessment, including already compromised passwords. SaaS platform administrators should consider industry-wide accepted standards, such as NIST, while configuring this feature. This control applies to SaaS platform users and is not needed for users with delegated authentication (SAML).
IAM-SaaS-12	Multi-Factor Authentication	The SaaS platform must support the use of multi-factor authentication. The SaaS platform must allow SaaS platform administrators to toggle on and off each factor. The SaaS platform must have the capability to configure MFA enforcement so the user may not sign in without MFA.	It is recommended to disallow vulnerable MFA methods (such as SMS) and support phishing-resistant methods.
IAM-SaaS-13	Disabling Anonymous Access	If the SaaS platform supports anonymous access, it must provide a mechanism to disable it globally.	*This cell intentionally left blank*

IAM-SaaS-14	Disabling External Access for Unmanaged Users	If the SaaS platform supports access for external unmanaged users, it must provide a mechanism to disable it globally.	External unmanaged users can also be referred to as guest users.
IAM-SaaS-15	Session Revocation / Single Sign Out	The SaaS platform must support a process to invalidate a user's sessions via programmatic means. Invalidation of the user session must have the capability to revoke user and application sessions (all device and UI sessions) in real-time.	This is commonly referred to as a universal log-out.
IAM-SaaS-16	Entitlements Change Enforcement	The SaaS platform must support immediate enforcement of entitlement changes. Where not possible, forced re-authentication is allowed.	The SaaS platform should provide session blocklisting.
IAM-SaaS-17	Temporary Account Suspension	The SaaS platform must support programmatic suspension and reactivation of accounts without requiring their deletion. When account suspension is invoked, the SaaS platform must suspend or revoke active sessions.	Associated NHIs should be suspended when account suspension is triggered. The SaaS platform should allow for the suspension and reactivation of accounts, including any associated NHIs. Upon reactivation, all NHIs linked to the account should be restored without any of them being revoked.
IAM-SaaS-18	Scopes Requirements	If the SaaS platform supports a scoped protocol such as OAuth, then granular scopes must be created that allow for least privilege operations. Read and write scopes are separated. SaaS platform administrative actions, such as managing data, must be scoped separately.	While read and write scopes should be provided separately, the application may provide scopes such as manage or administrative, which combine lower-level scopes.
IAM-SaaS-19	Third Party Allowlisting	The SaaS platform must provide administrative controls to determine which third-party integrations can be connected by users.	If the SaaS platform does not have the ability to manage an allow list, it should allow SaaS platform administrators to block the installation of third-party applications globally by regular users.

IAM-SaaS-20	Inactive Session Timeout	The SaaS platform must support the configuration of inactive UI session timeout settings. The inactive session timeout must allow SaaS platform administrators to set the inactive UI session timeout within the UI of the SaaS platform or the security configuration API.	The SaaS platform should have a default inactive session timeout in minutes or hours, not days or weeks.
IAM-SaaS-21	Restricting User Invites	If users can be provisioned or invited by users other than administrators, the SaaS platform must support restricting this capability to specific roles.	The SaaS platform should support the invitation of collaboration users. If such functionality is available, SaaS platform administrators can restrict by role those users authorized to issue invitations.

2.4 Interoperability and Portability (IPY)

Control ID	Control Title	Control Specification (Requirement / Must)	Implementation Guideline (Recommendation / Should)
IPY-SaaS-01	Export Capability	If the SaaS platform offers mass data export functionality, it must allow SaaS platform administrators to disable this functionality for non-administrative users.	Export capability should be disabled by default for non-administrative users.
IPY-SaaS-02	Integration Attribution	The SaaS platform must implement a mechanism to allow or deny connections based on the verified creator of the integration. When SaaS platform users are prompted to accept integrations (e.g., via a consent screen), the acceptance process must display at least one verifiable attribute of the application's creator, such as their email address domain. This verifiable ownership attribute must be visible both during the acceptance process and in the list of connected integrations, as stipulated in **IAM-SaaS-06**. Verification of the integration creator must be performed by the SaaS platform using an application-appropriate mechanism.	An example of a SaaS platform-appropriate mechanism can be confirmed by the email address of the creator or the domain used in the OAuth consent (for relevant flows).

2.5 Logging and Monitoring (LOG)

Control ID	Control Title	Control Specification (Requirement / Must)	Implementation Guideline (Recommendation / Should)
LOG-SaaS-01	Logged Events Scope	The SaaS platform must provide security logs to SaaS customers. Events from both NHIs and humans must be captured in logs, including: • Sign in attempts (fail + pass) • All configuration changes • Creating integrations, including into other SaaS platforms • Creation, deletion, and/or modification of API keys • OAuth access key generation using a refresh token • User impersonation (including by local administrators or user-to-user role assumption) • Creation and modification of user accounts and their permissions • Each authentication step, including MFA stages and the factor used • Bulk export and mass data reporting activity	The logs should be in a machine-readable format (e.g., JSON). Logs should include: • All configuration changes that impact the customer UI and configuration • Non-administrative changes • Sharing of objects With logs for user impersonation, the user ID does not need to include email address or full name, just a unique identifier of the impersonating user.
LOG-SaaS-02	Log Records Mandatory Fields	The SaaS platform logs must contain the following security-relevant information: • Timestamp • User ID/username, or NHI ID (if applicable) • Impersonation user ID, whether from a customer or SaaS provider • IP address • User agent (if applicable) • Source of change context (API/UI/App) • Action	The logs should: • Describe the source of change (API vs UI vs 3rd party app vs SaaS provider), making changes to customer-visible configurations • Describe target resource (field/display names) • Describe the session identifier If items like an IP address are not applicable, they can be excluded.

		• Target resource • Non-sensitive session identifier	For clarity, this only applies to the SaaS platform logs (not backend activity). A non-sensitive session identifier is a unique identifier representing an authenticated session, not a confidential session value.
LOG-SaaS-03	Programmatic Logs Delivery	The SaaS platform must support programmatic log delivery via a push or pull mechanism.	The logs should be in a machine-readable format (e.g., JSON). Common delivery mechanisms include pulling logs from the SaaS platform API endpoint or automatic delivery from the SaaS platform by way of a webhook or cloud storage bucket. For SaaS platforms where logs may be delivered out of order and a pull mechanism from the customer is available, customers should be able to query based on log delivery time, as opposed to event time. This prevents gaps for out-of-order logs that are continuously retrieved.
LOG-SaaS-04	Logs Retention	The SaaS platform logs must be retained and be made available to customers. Logs must be made available to the customer for a minimum of 7 days.	It is recommended that logs are available for 30 days or longer for critical log types such as login events.
LOG-SaaS-05	Logs Delivery Latency	The SaaS platform logs must be delivered without undue delay or latency. Logs must be made available and deliverable to or by the customer without undue delay but at most within 24 hours.	The SaaS platform should allow throttling mechanisms to allow the timely delivery of logs to customers.
LOG-SaaS-06	Log Events Documentation	The SaaS platform must provide up-to-date documentation for log format, log event types, and all log fields and their meaning.	Documentation should be available via the SaaS provider's website, within the platform, or on a community page. Public documentation is recommended. Documentation should also include

			versioning and change logs to support traceability and audit requirements. Any custom language or non-industry standard terms should be explained in detail.
LOG-SaaS-07	Log Integrity	If the SaaS platform allows logs to be mutable, it must provide an administrative mechanism for logs to be made immutable.	The SaaS platform can still be compliant if they have a specific use case that needs mutable logs, provided that a mechanism exists to disable mutability. This is specifically relevant for platforms that deal with data where logs are redirected to a data storage layer that the customer fully controls (e.g., storage buckets, database tables).

2.6 Security Incident Management, E-Discovery, and Cloud Forensics (SEF)

Control ID	Control Title	Control Specification (Requirement / Must)	Implementation Guideline (Recommendation / Should)
SEF-SaaS-01	Security Event Notification	The SaaS platform must allow setting a security contact who will be notified in case of a security incident.	The SaaS platform should send periodic emails to SaaS customers if this contact is not set.

Glossary and Acronyms

Acronyms

API – Application Programming Interface
A set of protocols and tools that allow different software applications to interact and exchange data.

CCC – Change Control and Configuration Management
A control domain within CCM, CAIQ, and the SSCF focused on ensuring consistent, secure handling of system changes and configurations.

CCM – Cloud Controls Matrix
A cybersecurity framework by the CSA for cloud-specific security controls.

CSA – Cloud Security Alliance
A leading industry group promoting best practices for cloud computing security.

DSP – Data Security and Privacy Lifecycle Management
A control domain within the SSCF addressing the protection of data from creation through deletion.

IAM – Identity and Access Management
A control domain with the SSCF and a framework for ensuring that only authorized identities can access specific resources.

IPY – Interoperability and Portability
A control domain within the SSCF that ensures SaaS systems can integrate easily and allow data to move between systems.

LOG – Logging and Monitoring
A control domain within the SSCF focused on capturing and analyzing security-relevant system and user activity.

MFA – Multi-Factor Authentication
A login method requiring two or more independent authentication factors.

NHI – Non-Human Identity
Digital identities not tied to human users, such as APIs, bots, or service accounts.

NIST – National Institute of Standards and Technology
A U.S. agency creating standards for technology and security practices.

OAuth – Open Authorization
An open protocol for delegated access to systems without sharing login credentials.

RBAC – Role-Based Access Control
A method of assigning access rights based on roles rather than individuals.

SaaS – Software as a Service
A software distribution model where applications are hosted by a provider and accessed online.

SCIM – System for Cross-Domain Identity Management
An open standard to automate the exchange of user identity data between systems.

SEF – Security Incident Management, E-Discovery, and Cloud Forensics
A control domain within the SSCF dealing with incident response, digital investigations, and legal discovery in cloud systems.

SOC 2 – System and Organization Controls Related to Security, Availability, Processing Integrity, Confidentiality, and/or Privacy
An audit standard evaluating how service providers securely manage customer data.

SSCF – SaaS Security Capability Framework
A security framework specifying essential customer-facing controls in SaaS applications.

SSPM – SaaS Security Posture Management
A solution for monitoring and maintaining secure configurations across SaaS platforms.

SSRM – Shared Security Responsibility Model
A model dividing security obligations between cloud providers and their customers.

TPRM - Third-Party Risk Management
The process of identifying, assessing, monitoring, and mitigating risks that third-party vendors, suppliers, partners, and other external entities introduce to an organization's data, operations, and finances.

WG – Working Group
A team of experts developing or maintaining a security framework or standard.

Terms

Access Control List (ACL)
Defines permissions assigned to specific users or systems for accessing or modifying resources.

API Key
A credential used to authenticate API requests and track usage of API-based services.

Authentication
The process of verifying the identity of a user, device, or system.

Authorization
Determines what an authenticated user is allowed to do within a system.

Cloud Forensics
Applying forensic techniques to investigate security events in cloud environments.

Configuration Management
The discipline of handling and documenting changes to system configurations securely and consistently.

Control ID
A unique reference number is assigned to each control within the SSCF framework.

Control Specification
The core requirement of a security control that must be implemented and mandatory explanatory elements that clarify the intent and context of a security control.

Data Residency
The physical or geographic location where data is stored, which is often subject to regulatory requirements.

E-Discovery
Identifying, collecting, and producing digital information for legal or regulatory purposes.

End-to-End Encryption (E2EE)
A security mechanism that encrypts data at the origin and only decrypts it at the final destination.

Entitlement
A defined permission or access right assigned to a user or system entity.

Implementation Guideline
Recommended practices for securely and effectively implementing a control.

Immutable Logs
Logs that cannot be altered or deleted, ensuring trust in audit records.

Incident Response
A structured process for detecting, responding to, and recovering from security incidents.

ISO, International Organization for Standardization
A global standards body publishing technical and business best practices and audit requirements for certifications.

Least Privilege
The practice of granting users only the access necessary for their roles or functions.

Log Redaction
Removing sensitive information from logs to comply with data privacy standards.

Log Retention
The duration for which log data is stored and made available for analysis or auditing.

Multi-Tenancy
A software architecture where a single instance serves multiple customers with isolated data.

Non-Human Identities (NHIs)
Digital entities, such as bots or APIs, that interact with SaaS systems without human involvement.

OAuth Token
A credential issued during OAuth flows to authorize access without sharing passwords.

Programmatic Access
Automated system access using code or APIs rather than manual user interaction.

Refresh Token
A credential used to renew access tokens without re-authenticating the user.

SaaS Security Capability Framework (SSCF)
A structured set of SaaS-specific security controls focused on customer-manageable capabilities.

Security Configurations
Settings that affect the security of a SaaS application, such as access controls and encryption.

Session Timeout
A mechanism that ends inactive user sessions after a defined period to enhance security.

Shared Security Responsibility Model (SSRM)
Outlines which party—the provider or customer—is responsible for each aspect of cloud security.

Single Sign-On (SSO)
Allows users to log in once and access multiple systems without re-authenticating.

Suspended Account
An account is temporarily disabled (but not deleted) due to policy violations or security concerns.

Tenant
An individual or organization with isolated access to shared infrastructure in a multi-tenant system.

User Impersonation
The act of an authorized person assuming another user's identity for troubleshooting, which is usually audited.

Webhook
An automated message is sent from one application to another when specific events occur.

For a more comprehensive glossary of terms on cloud security, navigate to the CSA Cloud Security Glossary.