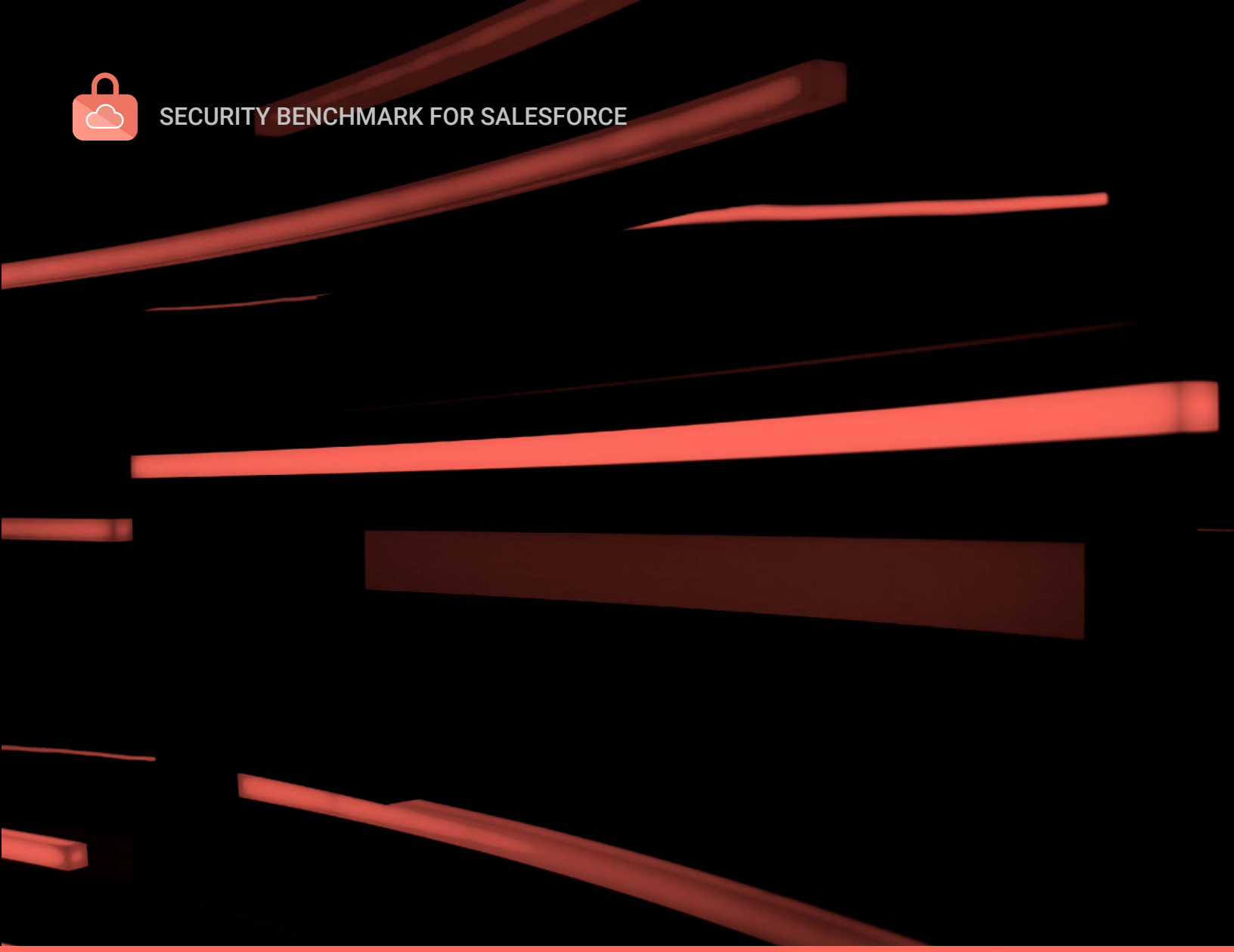


Security Benchmark for Salesforce:

A comprehensive security standard for Salesforce environments

Independent, prescriptive and auditable security controls for evaluating the security posture of Salesforce organizations.

Table of Conents

What SBS Is? 03

OAuth Security 09

Integrations 13

Access Controls 17

Authentication 29

Code Security 34

Customer Portals 39

Data Security 44

Deployments 48

Security Configuration 54

File Security 57

Salesforce Logging Architecture 60

Conclusion 69

Authors 71

01

WHAT SBS IS

What SBS Is

Salesforce has evolved far beyond its CRM origins to become a mission-critical business platform for many of the world's largest enterprises and government agencies. It is widely recognized as one of the most secure enterprise SaaS platforms in operation, with strong default controls, mature identity primitives, and extensive compliance
certifications. However, incident reporting from 2024 and 2025, including analysis published by Google Threat Intelligence, have shown that Salesforce environments are increasingly being compromised through inconsistent implementation and operation of Salesforce security controls at scale. These campaigns have demonstrated repeated abuse of overprivileged users, ungoverned administrator access, and OAuth-based integrations, where legitimate Salesforce features were leveraged to enable data exfiltration, lateral access, and delayed detection when governance and operational discipline broke down.

The Security Benchmark for Salesforce (SBS) addresses a critical gap between what Salesforce makes possible and what is consistently implemented, measured, and defended in real-world enterprise environments. While Salesforce provides extensive security mechanisms, there is no widely accepted, practitioner-defined baseline for what constitutes an adequate Salesforce security posture at enterprise scale. Without such a benchmark, organizations lack an objective way to assess preparedness, compare posture, or understand how their Salesforce environments will withstand, contain, and be investigated during a security incident. SBS defines a set of verifiable conditions that systematically raise security posture over time, under the assumption that security incidents will occur. Its purpose is to reduce breach impact by limiting blast radius, preserving forensic visibility, and enabling effective incident response.

Purpose and Scope

SBS defines the conditions that must be true for a Salesforce environment to be considered securely operated in the context of modern threat activity. It is intended for chief information security officers, security architecture teams, auditors, system integrators, and security tooling as a common reference for evaluating Salesforce security posture in a consistent, auditable, and repeatable manner.

The benchmark applies to Salesforce environments that function as critical business platforms and data systems. It addresses security across organizational, technical, and operational domains, including identity and privilege management, integration governance, data access visibility, development practices, and change management. SBS is designed to support prevention, detection, and recovery by enabling organizations to understand how specific Salesforce security controls reduce breach likelihood, limit impact, and accelerate post-incident assessment.

Control Structure

Each SBS control is expressed as a normative requirement and includes clearly defined audit procedures and remediation guidance. Controls are written to support objective verification and are designed to be evaluated independently or as part of a broader security assessment.

SBS focuses on platform-relevant security outcomes and avoids prescribing specific vendors, tools, or implementation approaches. Where controls depend on optional Salesforce capabilities or external security functions, those dependencies are documented explicitly.

Relationship to Other Frameworks

SBS is complementary to established security frameworks such as NIST and ISO. While those frameworks define program-level security principles, SBS provides Salesforce-specific security requirements that translate those principles into concrete, auditable expectations for Salesforce environments.

SBS is not a certification program and does not replace regulatory or compliance obligations. It is intended to serve as a platform-specific benchmark that organizations may map to broader governance and risk management frameworks.

Governance and Independence

SBS is maintained by an editorial board and contributors drawn from the Salesforce and security practitioner communities. It is an independent initiative and is not official, endorsed, or supported by Salesforce.

Controls At-a-Glance

This page provides a quick reference to all SBS control statements. Each control statement is a concise, single-sentence summary of the requirement. For full details including rationale, audit procedures, and remediation steps, refer to the individual control sections.

Access Controls

SBS-ACS-001: Enforce a Documented Permission Set Model All permission sets, permission set groups, and profiles must conform to a documented model maintained in a system of record and enforced continuously.

SBS-ACS-002: Documented Justification for All `API-Enabled` Authorizations Every authorization granting the "API Enabled" permission must have documented business or technical justification recorded in a system of record.

SBS-ACS-003: Documented Justification for `Approve Uninstalled Connected Apps` Permission The "Approve Uninstalled Connected Apps" permission must only be assigned to highly trusted users with documented justification and must not be granted to end-users.

SBS-ACS-004: Documented Justification for All Super Admin–Equivalent Users All users with simultaneous View All Data, Modify All Data, and Manage Users permissions must be documented in a system of record with clear business or technical justification.

SBS-ACS-005: Only Use Custom Profiles for Active Users All active users must be assigned custom profiles. The out-of-the-box standard profiles must not be used.

SBS-ACS-006: Documented Justification for `Use Any API Client` Permission The "Use Any API Client" permission must only be assigned to highly trusted users with documented justification and must not be granted to end-users.

SBS-ACS-007: Maintain Inventory of Non-Human Identities Organizations must maintain an authoritative inventory of all non-human identities, including integration users, automation users, bot users, and API-only accounts.

SBS-ACS-008: Restrict Broad Privileges for Non-Human Identities Non-human identities must not be assigned permissions that bypass sharing rules or grant administrative capabilities unless documented business justification exists.

SBS-ACS-009: Implement Compensating Controls for Privileged Non-Human Identities Non-human identities with permissions that bypass sharing rules or grant administrative capabilities must have compensating controls implemented to mitigate risk.

SBS-ACS-010: Enforce Periodic Access Review and Recertification All user access and configuration influencing permissions and sharing must be formally reviewed and recertified at least annually by designated business stakeholders, with documented approval and remediation of unauthorized or excessive access.

SBS-ACS-011: Enforce Governance of Access and Authorization Changes All changes to Salesforce user access and authorization must be governed through a documented process that requires approval, records business justification, and produces an auditable record of the change.

SBS-ACS-012: Classify Users for Login Hours Restrictions Organizations must maintain a documented classification of users who require login hours restrictions or equivalent monitoring, and must either enforce those restrictions or implement monitoring and alerting for off-hours authentication.

Authentication

SBS-AUTH-001: Enable Organization-Wide SSO Enforcement Setting Salesforce production orgs must enable the org-level setting that disables Salesforce credential logins for all users.

SBS-AUTH-002: Govern and Document All Users Permitted to Bypass Single Sign-On All users who do not have the "Is Single Sign-On Enabled" permission must be explicitly authorized, documented in a system of record, and limited to approved administrative or break-glass use cases.

SBS-AUTH-003: Prohibit Broad or Unrestricted Profile Login IP Ranges Profiles in Salesforce production orgs must not contain login IP ranges that effectively permit access from the full public internet or other overly broad ranges that bypass network-based access controls.

SBS-AUTH-004: Enforce Strong Multi-Factor Authentication for External Users with Substantial Access to Sensitive Data All Salesforce interactive authentication flows for external human users with substantial access to sensitive data must enforce multi-factor authentication that includes at least one strong authentication factor.

Code Security

SBS-CODE-001: Mandatory Peer Review for Salesforce Code Changes All Salesforce code changes must undergo peer review and receive approval before merging into any production-bound branch.

SBS-CODE-002: Pre-Merge Static Code Analysis for Apex and LWC Static code analysis with security checks for Apex and Lightning Web Components must execute successfully before any code change is merged into a production-bound branch.

SBS-CODE-003: Implement Persistent Apex Application Logging Organizations must implement an Apex-based logging framework that writes application log events to durable Salesforce storage and must not rely on transient Salesforce debug logs for operational or security investigations.

SBS-CODE-004: Prevent Sensitive Data in Application Logs Application logging frameworks must not capture, store, or transmit credentials, authentication tokens, personally identifiable information (PII), regulated data, or other sensitive values in log messages or structured log fields.

Customer Portals

SBS-CPORTAL-001: Prevent Parameter-Based Record Access in Portal Apex AuraEnabled methods exposed to customer portal users must not accept user-supplied record identifiers or parameters that directly determine query scope or record access.

SBS-CPORTAL-002: Restrict Guest User Record Access Guest users in customer portals must not have Create, Read, Update, or Delete permissions on standard or custom objects except as strictly required for unauthenticated user flows.

SBS-CPORTAL-003: Inventory Portal-Exposed Apex Classes and Flows Organizations must maintain an authoritative inventory of all Apex classes and Autolaunched Flows exposed to Experience Cloud sites, documenting which components are accessible to external and guest users.

SBS-CPORTAL-004: Prevent Parameter-Based Record Access in Portal-Exposed Flows Autolaunched Flows exposed to customer portal users must not accept user-supplied input variables that directly determine which records are accessed.

SBS-CPORTAL-005: Conduct Penetration Testing for Portal Security Organizations with Experience Cloud sites must conduct penetration testing of portal security controls before initial go-live and subsequently after major releases or on a defined cadence.

Data Security

SBS-DATA-001: Implement Mechanisms to Detect Regulated Data in Long Text Area Fields The organization must implement a mechanism that continuously or periodically analyzes the contents of all Long Text Area fields to identify the presence of regulated or personal data.

SBS-DATA-002: Maintain an Inventory of Long Text Area Fields Containing Regulated Data The organization must maintain an up-to-date inventory of all Long Text Area fields that are known or detected to contain regulated or personal data.

SBS-DATA-003: Maintain Tested Backup and Recovery for Salesforce Data and Metadata Salesforce production orgs must maintain a documented backup and recovery capability for Salesforce data and metadata, and must test restoration on a defined schedule.

SBS-DATA-004: Require Field History Tracking for Sensitive Fields The organization must maintain a documented list of sensitive fields and ensure Field History Tracking is enabled for each listed field on all in-scope objects.

Deployments

SBS-DEP-001: Require a Designated Deployment Identity for Metadata Changes Salesforce production orgs must designate a single deployment identity that is exclusively used for all metadata deployments and high-risk configuration changes performed through automated or scripted release processes.

SBS-DEP-002: Establish and Maintain a List of High-Risk Metadata Types Prohibited from Direct Production Editing Salesforce production orgs must maintain an explicit list of high-risk metadata types that must never be edited directly in production by human users, defaulting at minimum to the SBS baseline list while allowing organizations to extend or refine it as needed.

SBS-DEP-003: Monitor and Alert on Unauthorized Modifications to High-Risk Metadata Salesforce production orgs must implement a monitoring capability that detects and reports any modification to high-risk metadata performed by a user other than the designated deployment identity.

SBS-DEP-004: Establish Source-Driven Development Process Meaningful Salesforce metadata changes must be deployed through a source-driven, automated, and deterministic deployment process, except where the platform does not provide programmatic deployment support.

SBS-DEP-005: Implement Secret Scanning for Salesforce Source Repositories Organizations using source-driven development for Salesforce must implement automated secret scanning on all repositories containing Salesforce metadata, configuration, or deployment scripts to detect and prevent the exposure of credentials, access tokens, and other sensitive authentication material.

SBS-DEP-006: Configure Salesforce CLI Connected App with Token Expiration Policies Organizations must configure the Connected App used for Salesforce CLI authentication with refresh token expiration of 90 days or less and access token timeout of 15 minutes or less.

Event Monitoring

SBS-MON-001: Enable Event Monitoring Log Storage Organizations using Salesforce Event Monitoring must ensure that storage of required Event Monitoring logs is enabled for all event types necessary to support the organization's security monitoring and compliance policies.

SBS-MON-002: Retaining Event Logs Organizations must retain security event logs for the defined retention period and implement measures to protect the logs from tampering and unauthorized deletion to ensure forensic availability.

SBS-MON-003: Monitor for Suspicious Logins Organizations must continuously monitor and alert on anomalous login patterns to promptly detect and mitigate compromised accounts and application credentials.

SBS-MON-004: Monitor for Suspicious API Activity Organizations must continuously monitor and alert on all API activity to establish a baseline, detect anomalous and
malicious activity, and identify potential application and integration abuse in a timely manner.

SBS-MON-005: Monitor API Usage Against Limits Organizations must implement continuous, real-time monitoring and alerting on current API usage against defined Salesforce limits to proactively prevent service disruptions.

File Security

SBS-FILE-001: Require Expiry Dates on Public Content Links Organizations must ensure that Public Content links have an appropriate expiry date.

SBS-FILE-002: Require Passwords on Public Content Links for Sensitive Content Organizations must ensure that Public Content links to sensitive content have a password.

SBS-FILE-003: Periodic Review and Cleanup of Public Content Links Organizations must implement a recurring process to review all active Public Content links and remove or remediate links that are no longer required, lack appropriate controls, or were created outside of current policy.

Foundations

SBS-FDNS-001: Centralized Security System of Record The organization must maintain a centralized system of record documenting all Salesforce security configurations, exceptions, justifications, and SBS-required inventories.

Integrations

SBS-INT-001: Enforce Governance of Browser Extensions Accessing Salesforce Organizations must enforce a centrally managed mechanism that restricts which browser extensions are permitted to access Salesforce, and must not allow the use of unmanaged or uncontrolled extensions.

SBS-INT-002: Inventory and Justification of Remote Site Settings Organizations must maintain an authoritative inventory of all Remote Site Settings and document a business justification for each endpoint approved for Apex HTTP callouts.

SBS-INT-003: Inventory and Justification of Named Credentials Organizations must maintain an authoritative inventory of all Named Credentials and document a business justification for each external endpoint and authentication configuration approved for use in Salesforce.

SBS-INT-004: Retain API Total Usage Event Logs for 30 Days The organization must retain API Total Usage event log data (EventLogFile EventType=ApiTotalUsage) for at least the immediately preceding 30 days using Salesforce-native retention or automated external export and storage.

OAuth Security

SBS-OAUTH-001: Require Formal Installation of Connected Apps Organizations must formally install all connected apps used for OAuth authentication rather than relying on user-authorized OAuth connections.

SBS-OAUTH-002: Require Profile or Permission Set Access Control for Connected Apps Organizations must control access to each formally installed connected app exclusively through assigned profiles or permission sets.

SBS-OAUTH-003: Add Criticality Classification of OAuth-Enabled Connected Apps All OAuth-enabled Connected Apps must be recorded in an authoritative system of record and assigned a documented vendor criticality rating reflecting integration importance and data sensitivity.

SBS-OAUTH-004: Due Diligence Documentation for High-Risk Connected App Vendors Organizations must review and retain available security documentation for all high-risk Connected App vendors and explicitly record any missing documentation as part of the vendor assessment.

Security Configuration

SBS-SECCONF-001: Establish a Salesforce Health Check Baseline Salesforce production orgs must define and maintain a Salesforce Health Check baseline—including Salesforce's native baseline XML or an equivalent customized baseline—and ensure it reflects the organization's intentional security configuration posture.

SBS-SECCONF-002: Review and Remediate Salesforce Health Check Deviations Salesforce production orgs must periodically review Health Check results against the defined baseline and remediate deviations or formally document approved exceptions.

02

OAUTH SECURITY

OAuth Security

This section defines controls related to OAuth-enabled Connected Apps, third-party integrations, and external access to Salesforce environments. These controls ensure that organizations maintain visibility, governance, and lifecycle management over external systems that authenticate to Salesforce via OAuth, reducing the risk of unauthorized access, data exfiltration, and stale integration pathways.

SBS-OAUTH-001: Require Formal Installation of Connected Apps

HIPAA GDPR NIST CCPA/CPRA SOC 2 ISO 27001

Control Statement:
Organizations must formally install all connected apps used for OAuth authentication rather than relying on user-authorized OAuth connections.

Description:
The organization must ensure that any connected app used for OAuth authentication is formally installed in the Salesforce org as a managed or unmanaged connected app rather than implicitly created through user-initiated OAuth flows. Connected apps that appear only as user-authorized OAuth connections without formal installation expose the org to unmanaged security settings and prevent centralized governance.

Risk: Critical
Without formal installation, Connected Apps operate outside organizational control—inheriting security configuration from the external app developer rather than the Salesforce administrator. This establishes an unmanaged security boundary: refresh token lifetimes, session policies, and IP restrictions cannot be enforced, allowing tokens to persist indefinitely and enabling unauthorized access from any location. Attackers who compromise a user-authorized OAuth token gain persistent access that administrators cannot revoke or constrain through standard Connected App policies.

Audit Procedure:

1. Enumerate all user-authorized OAuth connected apps via Setup or the Tooling/Metadata API.

2. Identify all connected apps that are not formally installed as managed or unmanaged connected apps.

3. Flag any connected app that is used but not formally installed as noncompliant.

Remediation:

1. Formally install any connected app that appears only as a user-authorized OAuth connection.

2. Configure the installed connected app's policies, including refresh token and session security settings.

3. Remove the user-authorized OAuth connections that are now superseded by the installed connected app.

Default Value:
When a user first authenticates to a connected app via OAuth, Salesforce automatically creates a user-authorized OAuth entry that is not formally installed.

SBS-OAUTH-002: Require Profile or Permission Set Access Control for Connected Apps

HIPAA GDPR NIST CCPA/CPRA SOC 2 ISO 27001

Control Statement:
Organizations must control access to each formally installed connected app exclusively through assigned profiles or permission sets.

Description:
All formally installed connected apps must govern user access through explicit profile or permission set assignments. No connected app may rely on unrestricted or unmanaged access models.

Risk: Critical
Without explicit profile or permission set access control, Connected Apps may allow any user in the org to authenticate—bypassing the principle of least privilege and creating an uncontrolled access boundary. This enables unauthorized users to establish OAuth sessions with external systems, potentially exfiltrating data or performing actions beyond their intended scope. The lack of access scoping also eliminates audit visibility into who is authorized to use each integration, preventing detection of unauthorized access patterns.

Audit Procedure:

1. Enumerate all formally installed connected apps via Setup or the Metadata API.

2. For each installed connected app, verify that access is granted only through assigned profiles or permission sets.

3. Flag any connected app that lacks access scoping via profiles or permission sets as noncompliant.

Remediation:

1. For each connected app lacking profile or permission set access control, create or update profiles or permission sets to define which users are authorized to access the app.

2. Assign the appropriate profiles or permission sets to the connected app configuration.

3. Verify that no users can access the connected app without explicit authorization.

Default Value:
Salesforce does not require profile or permission set access control for connected apps by default. Access models vary based on connected app configuration.

SBS-OAUTH-003: Add Criticality Classification of OAuth-Enabled Connected Apps

GDPR CCPA/CPRA SOC 2 ISO 27001

Control Statement:
All OAuth-enabled Connected Apps must be recorded in an authoritative system of record and assigned a documented vendor criticality rating reflecting integration importance and data sensitivity.

Description:
Organizations must maintain a complete, authoritative inventory of all OAuth-enabled Connected Apps and assign each an explicit vendor criticality rating based on operational importance and the sensitivity of accessible Salesforce data.

Risk: High
Without a complete inventory and criticality classification, organizations lose visibility into their third-party integration landscape—preventing effective risk assessment, prioritization of security controls, and governance of external system connectivity. Security teams cannot identify which integrations access sensitive data, scope the impact of a vendor compromise, or respond effectively to incidents involving Connected Apps. This impairs detection, investigation, and response capabilities for integration-related security events.

Audit Procedure:

1. Retrieve a list of all Connected Apps with active OAuth configurations from Salesforce Setup.

2. Retrieve the organization's authoritative system of record for integration and vendor management.

3. Compare the Salesforce Connected App list to the system of record and confirm every OAuth-enabled Connected App appears in the inventory.

4. Verify each listed Connected App has an assigned vendor criticality rating documented in the system of record.

5. Flag any apps missing from the inventory or lacking a documented criticality rating as noncompliant.

Remediation:

1. Add any missing OAuth-enabled Connected Apps to the system of record.

2. Document and assign a vendor criticality rating to each Connected App based on operational importance and data sensitivity.

3. Implement a recurring process to synchronize Connected App changes with the system of record.

Default Value:
Salesforce does not automatically maintain or enforce an external inventory or criticality classification for Connected Apps.

SBS-OAUTH-004: Due Diligence Documentation for High-Risk Connected App Vendors

GDPR CCPA/CPRA ISO 27001

Control Statement:
Organizations must review and retain available security documentation for all high-risk Connected App vendors and explicitly record any missing documentation as part of the vendor assessment.

Description:
For each Connected App vendor classified as high-risk, the organization must collect, review, and store relevant security documentation—including terms of use, privacy policy, trust center or security overview, and any published information security guidelines—and must explicitly document when a required artifact does not exist.

Risk: Moderate
Without documented due diligence for high-risk vendors, organizations may onboard integrations without understanding the vendor's security posture, data handling practices, or contractual obligations. This increases the likelihood of undiscovered risks but does not directly enable unauthorized access—other controls (SBS-OAUTH-001, SBS-OAUTH-002) still govern the technical security boundary. Missing documentation primarily impacts audit readiness, risk assessment accuracy, and the organization's ability to make informed decisions about vendor relationships.

Audit Procedure:

1. Retrieve the list of Connected App vendors classified as high-risk from the organization's system of record.

2. For each high-risk vendor, verify that the following documents, where available, are stored in the designated repository:

   - Terms of use
   - Privacy policy
   - Trust center or security overview
   - Published information security guidelines

3. Confirm that any missing documentation is explicitly recorded as unavailable in the vendor assessment.

4. Flag any high-risk vendor lacking required documentation or missing explicit acknowledgment of unavailable documents as noncompliant.

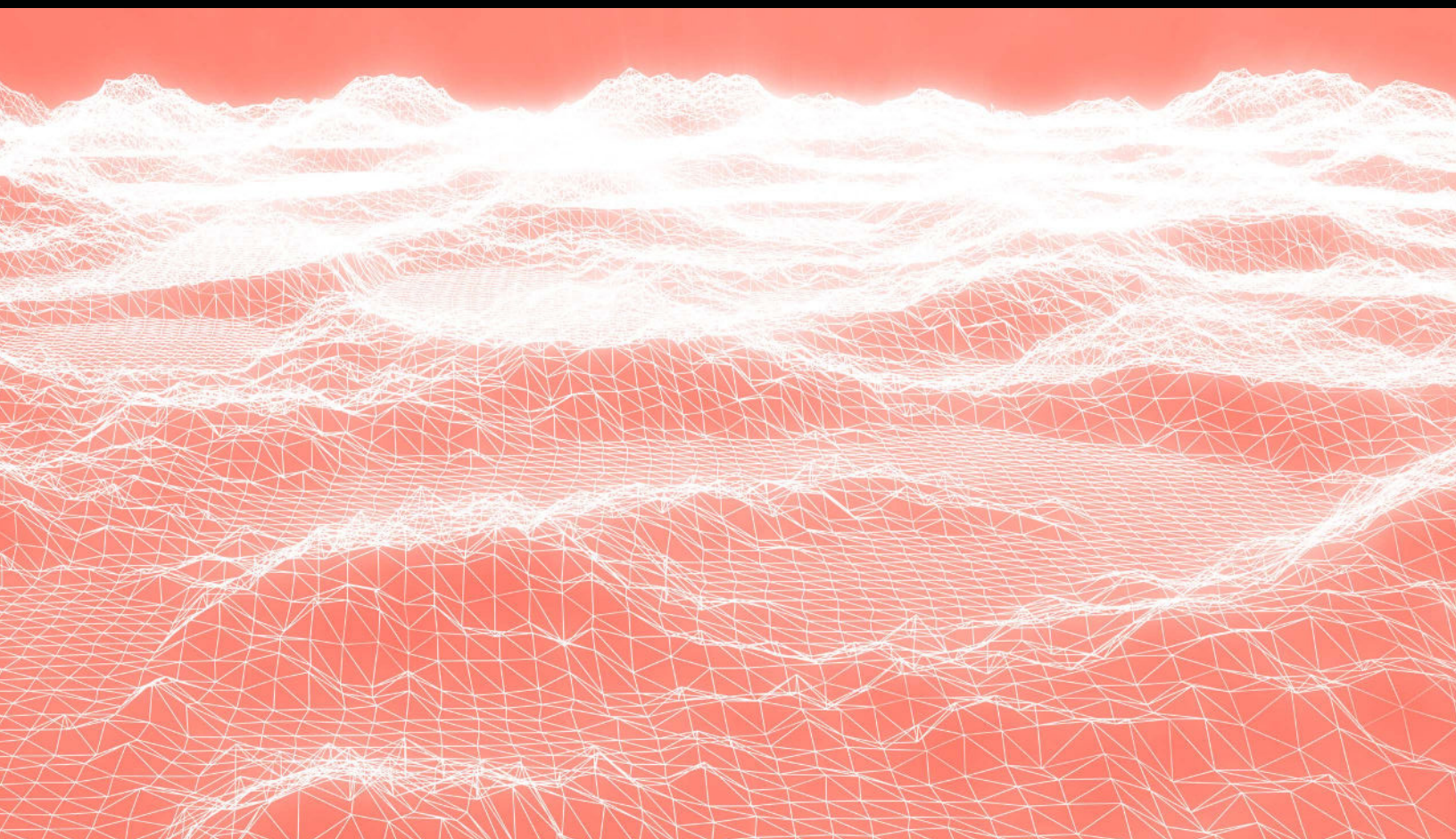
Remediation:

1. Collect and store all required documentation for each high-risk vendor.

2. Where documentation does not exist, record this absence in the vendor assessment.

3. Update the vendor management process to ensure ongoing due diligence for high-risk vendors.

Default Value:
Salesforce does not manage or enforce vendor due diligence requirements for Connected App providers.



03

INTEGRATIONS

Integrations

This section defines controls related to outbound connectivity from Salesforce to external systems, including Remote Site Settings and Named Credentials. These controls ensure that organizations maintain visibility and governance over approved external endpoints, authentication mechanisms, and data flows initiated by Salesforce, reducing the risk of unauthorized data transmission, dependency on untrusted services, and configuration drift in integration pathways.

SBS-INT-001: Enforce Governance of Browser Extensions Accessing Salesforce

SOC 2 ISO 27001

Control Statement:
Organizations must enforce a centrally managed mechanism that restricts which browser extensions are permitted to access Salesforce, and must not allow the use of unmanaged or uncontrolled extensions.

Description:
Organizations must deploy a centrally managed governance mechanism—such as Chrome Browser Cloud Management, MDM policies, or configuration profiles—that enforces an allow-list or blocklist for browser extensions accessing Salesforce domains.

Risk: Moderate
Without centralized governance over browser extensions, malicious or cloned extensions—increasingly common with AI-generated code—can harvest session tokens, exfiltrate data, and execute unauthorized operations within authenticated Salesforce sessions. However, this control provides governance and defense-in-depth rather than establishing a Salesforce-native security boundary: exploitation requires a malicious extension to be installed and an authenticated session to exist. Other controls (SSO, session management) still provide protection, and this governance mechanism operates outside Salesforce via MDM or browser management.

Audit Procedure:

1. Request evidence of a browser-extension governance mechanism applied to user devices (e.g., Chrome Browser Cloud Management, Intune configuration profile, Jamf configuration profile, Active Directory GPO, or equivalent).

2. Require a screenshot, exported policy file, or screen capture demonstrating that extension controls are active and enforceable (e.g., an allow-list or blocklist configuration for Chrome extensions).

3. Verify that the mechanism explicitly restricts installation or execution of unapproved extensions that can access Salesforce domains.

4. Flag the organization as noncompliant if no enforceable governance mechanism exists or if extension governance is based solely on policy, awareness, or voluntary user behavior.

5. Download API Total Usage logs (EventLogFile - ApiTotalUsage, available in free tier of Event Monitoring) and analyze for indicators of unauthorized browser extension activity:

   - Review USER_AGENT field for patterns indicating browser extensions (e.g., extension identifiers, non-standard user agents).

   - Identify API call patterns characteristic of auto-refresh extensions (e.g., Inspector Reloader) such as regular-interval repeated requests.

   - Flag any anomalous patterns for investigation against approved extension inventory.

Remediation:

1. Implement a centrally managed browser or device management solution capable of enforcing extension restrictions (e.g., Chrome Browser Cloud Management, Intune, Jamf, or GPO-based controls).

2. Define and apply an allow-list or blocklist policy governing which extensions are permitted to interact with Salesforce.

3. Remove or disable any unapproved browser extensions from managed devices.

4. Apply enforcement policies to all corporate-managed devices accessing Salesforce.

Default Value:
Salesforce provides no mechanism to prevent or detect browser extension usage; unmanaged browser extensions are permitted by default, including those capable of accessing Salesforce data and authenticated sessions.

SBS-INT-002: Inventory and Justification of Remote Site Settings

GDPR SOC 2 ISO 27001

Control Statement:
Organizations must maintain an authoritative inventory of all Remote Site Settings and document a business justification for each endpoint approved for Apex HTTP callouts.

Description:
All Remote Site Settings configured in Salesforce must be recorded in the organization's system of record along with a clear business justification demonstrating why the endpoint is required and trusted for use in Apex HTTP callouts.

Risk: Moderate
Without a documented inventory and justification for Remote Site Settings, unvetted or insecure endpoints may be authorized for Apex HTTP callouts—exposing the organization to data leakage, dependency risks, or communication with untrusted services. However, this control provides governance documentation rather than detection or prevention capability: it supports audit readiness and informed decision-making, but other controls are required to detect or prevent actual data exfiltration through approved endpoints.

Audit Procedure:

1. Enumerate all Remote Site Settings via Salesforce Setup or the Metadata API.
2. Retrieve the organization's system of record for approved outbound endpoints.
3. Compare the Salesforce list to the system of record to confirm each Remote Site Setting is documented.
4. Verify that each documented Remote Site Setting includes a clear business justification.
5. Flag any Remote Site Settings missing from the inventory or lacking justification as noncompliant.

Remediation:

1. Add all undocumented Remote Site Settings to the system of record.
2. Document a valid business justification for each endpoint.
3. Remove or disable any Remote Site Settings that cannot be justified.
4. Implement a recurring process to reconcile Remote Site Settings with the system of record.

Default Value:
Salesforce does not require or maintain business justification for Remote Site Settings and does not enforce an external inventory.

SBS-INT-003: Inventory and Justification of Named Credentials

GDPR SOC 2 ISO 27001

Control Statement:
Organizations must maintain an authoritative inventory of all Named Credentials and document a business justification for each external endpoint and authentication configuration approved for use in Salesforce.

Description:
All Named Credentials defined in Salesforce—regardless of authentication type or use case—must be recorded in the organization's system of record, including the endpoint URL, authentication model, and a clear business justification demonstrating why the connection is required and trusted for Apex callouts, External Services, or external data access.

Risk: Moderate
Without a documented inventory and justification for Named Credentials, undocumented or unjustified configurations may expose the organization to data leakage, unauthorized integrations, or reliance on insecure or untrusted endpoints. However, this control provides governance documentation rather than detection or prevention capability: it supports audit readiness and informed decision-making about authenticated external connections, but other controls are required to detect or prevent actual misuse of approved credentials.

Audit Procedure:

1. Enumerate all Named Credentials using Salesforce Setup, Metadata API, Tooling API, or Connect REST API.
2. Retrieve the organization's system of record for approved external endpoints and integration credentials.
3. Compare the Salesforce list to the system of record to confirm all Named Credentials are documented.
4. Verify that each documented Named Credential includes:
   - The external endpoint URL
   - The authentication type (named principal or per-user)
   - The business justification for the integration
5. Flag any Named Credentials missing from the inventory or lacking justification as noncompliant.

Remediation:

1. Add any undocumented Named Credentials to the system of record.

2. Document a valid business justification for each Named Credential.

3. Remove, disable, or reconfigure any Named Credentials that cannot be justified or that reference untrusted endpoints.

4. Establish a recurring reconciliation process to ensure Named Credentials remain fully inventoried and justified.

Default Value:
Salesforce does not maintain or enforce an external inventory or business justification for Named Credentials.

SBS-INT-004: Retain API Total Usage Event Logs for 30 Days

HIPAA GDPR NIST CCPA/CPRA SOC 2 ISO 27001

Control Statement:
The organization must retain API Total Usage event log data (EventLogFile EventType=ApiTotalUsage) for at least the immediately preceding 30 days using Salesforce-native retention or automated external export and storage.

Description:
If the organization's Salesforce does not provide at least 30 days of ApiTotalUsage EventLogFile availability in Salesforce, the organization must automatically export newly available ApiTotalUsage event log files at least once every 24 hours to an external log store that retains a minimum of 30 days of data.

Risk: High
Without retained API Total Usage logs, organizations lose visibility into REST, SOAP, and Bulk API activity—including user identity, connected app, client IP, resource accessed, and status codes. This materially degrades the ability to detect anomalous API behavior, investigate security incidents, attribute unauthorized access, and determine the scope of potential breaches. The absence of this visibility creates a significant gap in incident detection and response capabilities.

Audit Procedure:

1. Determine whether the organization relies on Salesforce-native retention (Event Monitoring/Shield/Event Monitoring add-on) or an external log store as the system of record for ApiTotalUsage EventLogFile data.

2. If the organization relies on Salesforce-native retention, verify that EventLogFile data is retained for at least 30 days (for example, confirm the org is entitled to and configured for Event Log File retention that is at least 30 days and can retrieve ApiTotalUsage EventLogFile data within the preceding 30-day window).

3. If the organization relies on an external log store (including all orgs with only 1-day ApiTotalUsage availability in Salesforce):

   - Verify an automated process exists that retrieves EventLogFile entries where EventType='ApiTotalUsage' and downloads the associated log files at least once every 24 hours.

   - Inspect job schedules/run history and confirm successful executions covering at least the last 30 days (no missed days).

   - From the external log store, retrieve ApiTotalUsage logs for (a) the oldest day in the preceding 30-day window and (b) the most recent day, and confirm both are accessible and attributable to the organization.

   - Verify access to the external log store is restricted to authorized roles and service identities responsible for monitoring and investigations.

Remediation:

1. If the organization has only 1-day ApiTotalUsage EventLogFile availability in Salesforce, implement an automated daily export that downloads newly available ApiTotalUsage log files and stores them externally for at least 30 days.

2. If the organization uses Salesforce-native retention, ensure the configured retention period for Event Log Files is not less than 30 days.

3. Restrict access to the retained logs (Salesforce-native or external) to authorized personnel and designated service identities.

Default Value:
Enterprise, Unlimited, and Performance Edition organizations have free access to the ApiTotalUsage event type with 1-day data retention, while organizations with Shield/Event Monitoring add-on retain Event Log Files for 30 days by default (and may be eligible to extend retention).

04

ACCESS CONTROLS

Access Controls

This section defines controls related to permission sets, permission set groups, profiles, and access governance within Salesforce environments. These controls ensure that organizations maintain a structured, documented, and enforced approach to authorization management, reducing privilege sprawl and unauthorized access risks.

SBS-ACS-001: Enforce a Documented Permission Set Model

NIST SOC 2 ISO 27001

Control Statement:
All permission sets, permission set groups, and profiles must conform to a documented model maintained in a system of record and enforced continuously.

Description:
The organization must define, document, and enforce a standardized permission set model within its system of record. A permission set model defines how the organization structures permissions—for example, using permission set groups to represent personas or departments, and permission sets to represent specific actions or capabilities. The specific structure is determined by the organization, but all profiles, permission sets, and permission set groups must conform to the documented model. No permission constructs may exist outside the defined model, and compliance must be evaluated and enforced on a near real-time basis.

Example models:

- Permission set groups represent job roles (Sales Rep, Service Agent), and individual permission sets represent capabilities (View Reports, Edit Accounts)
- Permission set groups represent departments (Sales, Marketing), and permission sets represent access tiers (Standard, Advanced)
- Permission sets represent business functions with no grouping hierarchy

Risk: High
Without a documented and enforced permission set model, organizations lose visibility into their authorization structure—accumulating ad hoc permission constructs created for one-time needs that are never reviewed or removed. This results in privilege sprawl, inconsistent access patterns, and inability to audit who has what access and why. Security teams cannot assess authorization posture, detect drift, or investigate access-related incidents when no authoritative model exists to compare against. The lack of continuous enforcement means unauthorized or excessive permissions can persist indefinitely without detection.

Audit Procedure:

1. Obtain the organization's documented permission set model from the designated system of record.
2. Enumerate all Profiles, Permission Sets, and Permission Set Groups using Salesforce Setup, Metadata API, or Tooling API.
3. Compare each enumerated item against the documented model to determine whether:
   - Its purpose or persona aligns with the model.
   - Its included permissions conform to the model's structure and boundaries.
   - Its naming and classification match the documented conventions.
4. Identify any profiles, permission sets, or permission set groups that do not conform to the model.
5. Verify that the organization has a process or automation that enforces model compliance in near real time (e.g., continuous scanning, pipelines, or governance workflows).

Remediation:

1. Update or deprecate noncompliant profiles, permission sets, and permission set groups to align with the documented permission set model.
2. Migrate users off legacy or misaligned authorization constructs.
3. Implement or enhance automated enforcement to ensure continuous alignment with the defined model.
4. Update the system-of-record documentation as the model changes.

Default Value:
Salesforce does not enforce any specific permission set model. Profiles, permission sets, and permission set groups can be created without structure or alignment unless governed by the organization.

SBS-ACS-002: Documented Justification for All `API-Enabled` Authorizations

NIST SOC 2 ISO 27001

Control Statement:
Every authorization granting the `API Enabled` permission must have documented business or technical justification recorded in a system of record.

Description:
All profiles, permission sets, and permission set groups that grant the `API Enabled` permission must be recorded in a designated system of record with a documented business or technical justification for requiring API access. Any authorization lacking documented rationale is noncompliant.

Risk: High
Without documented justification for API-enabled authorizations, organizations lose visibility into which users and systems can programmatically access Salesforce data at scale. The `API Enabled` permission enables large-scale data extraction, bulk modification, and automated operations—capabilities that create significant exposure when granted without oversight. Undocumented API access paths accumulate over time, preventing security teams from assessing data exfiltration risk, investigating suspicious API activity, or enforcing least privilege across automated access patterns.

Audit Procedure:

1. Enumerate all profiles, permission sets, and permission set groups that include the API Enabled permission using Salesforce Setup, Metadata API, Tooling API, or an automated scanner.

2. Compare the enumerated list against the organization's designated system of record for API-enabled authorizations.

3. Verify that every profile, permission set, and permission set group granting "API Enabled" has a corresponding entry in the system of record.

4. Confirm that each entry includes:

   - A clear business or technical justification for API access, and
   - Any applicable exception or approval documentation.

5. Flag as noncompliant any authorizations lacking documentation or justification.

Remediation:

1. Remove the `API Enabled` permission from any profile, permission set, or permission set group that lacks a documented justification and is not required for business operations.

2. For any authorization that legitimately requires API access, add or update the rationale in the system of record to clearly justify the need.

3. Reconcile and update the system of record to ensure complete and accurate inventory of all API-enabled authorizations.

Default Value:
Salesforce does not require or maintain a system of record for API-enabled authorizations. The `API Enabled` permission is disabled by default for standard profiles but may be granted by administrators.

SBS-ACS-003: Documented Justification for `Approve Uninstalled Connected Apps` Permission

HIPAA GDPR NIST CCPA/CPRA SOC 2 ISO 27001

Control Statement:
The `Approve Uninstalled Connected Apps` permission must only be assigned to highly trusted users with documented justification and must not be granted to end-users.

Description:
All profiles, permission sets, and permission set groups that grant the `Approve Uninstalled Connected Apps` permission must be recorded in a designated system of record with a documented business or technical justification. This permission should only be assigned to highly trusted users, such as administrators and developers involved in managing or testing connected app integrations. Any authorization lacking documented rationale is noncompliant.

Risk: Critical
The `Approve Uninstalled Connected Apps` permission allows users to bypass Connected App usage restrictions and self-authorize any OAuth application without administrator approval. This establishes an uncontrolled security boundary: users with this permission can grant external applications access to Salesforce data without oversight, enabling data exfiltration, unauthorized integrations, and potential account compromise. Unlike other permissions that require additional failures to exploit, this permission directly enables unauthorized third-party access the moment it is misassigned—making it a primary security boundary that must be tightly controlled.

Audit Procedure:

1. Enumerate all profiles, permission sets, and permission set groups that include the `Approve Uninstalled Connected Apps` permission using Salesforce Setup, Metadata API, Tooling API, or an automated scanner.

2. Compare the enumerated list against the organization's designated system of record for this permission.

3. Verify that every profile, permission set, and permission set group granting "Approve Uninstalled Connected Apps" has a corresponding entry in the system of record.

4. Confirm that each entry includes:
    - A clear business or technical justification for requiring this permission,
    - Identification of the user role or persona (e.g., administrator, developer, integration manager),
    - Any applicable exception or approval documentation, and
    - Confirmation that the use case is limited to testing or managing connected app integrations.
5. Verify that the permission is not assigned to end-user profiles or permission sets intended for general business users.
6. Flag as noncompliant any authorizations lacking documentation, justification, or assigned to unauthorized user populations.

Remediation:

1. Remove the `Approve Uninstalled Connected Apps` permission from any profile, permission set, or permission set group that lacks a documented justification or is assigned to end-users.
2. For any authorization that legitimately requires this permission (e.g., administrators or developers testing connected apps), add or update the rationale in the system of record to clearly justify the need and identify the specific role or use case.
3. Ensure that connected apps required for business operations are properly installed and allowlisted rather than relying on this permission for end-user access.
4. Reconcile and update the system of record to ensure complete and accurate inventory of all assignments of this permission.

Default Value:
The `Approve Uninstalled Connected Apps` permission is not granted by default in Salesforce. This permission was introduced in September 2025 as part of Connected App Usage Restrictions changes. Organizations must explicitly assign this permission to users who require it for legitimate testing or integration management purposes.

SBS-ACS-004: Documented Justification for All Super Admin–Equivalent Users

HIPAA GDPR NIST CCPA/CPRA SOC 2 ISO 27001

Control Statement:
All users with simultaneous `View All Data`, `Modify All Data`, and `Manage Users` permissions must be documented in a system of record with clear business or technical justification.

Description:
All users who hold simultaneous authorization for `View All Data`, `Modify All Data`, and `Manage Users`—collectively constituting Super Admin–level access—must be identified and documented in the system of record with a clear business or technical justification. Any user with this combination of permissions who lacks documented rationale is noncompliant.

Risk: High
Without documented justification for Super Admin–equivalent users, organizations lose visibility into who possesses unrestricted access to the entire Salesforce environment. These users can read and modify all data, manage user accounts, and alter the security posture of the org without oversight. Undocumented Super Admin access prevents security teams from assessing breach impact, investigating administrative actions, or maintaining accountability for the most sensitive operations. The inability to identify and justify these users also prevents effective access reviews and creates persistent exposure from forgotten or orphaned administrative accounts.

Audit Procedure:

1. Enumerate all users who simultaneously possess the following permissions through any profile, permission set, or permission set group:
    - `View All Data`
    - `Modify All Data`
    - `Manage Users`
2. Compile a list of all users meeting the criteria for Super Admin–equivalent access.
3. Compare the list against the organization's system of record.
4. Verify that each Super Admin–equivalent user has corresponding documentation that includes:
    - A clear business or technical justification for requiring this level of access, and

- Any relevant exception or approval records.

5. Flag as noncompliant any users with Super Admin–equivalent access lacking documentation or justification.

Remediation:

1. Remove one or more of the Super Admin–equivalent permissions from any user who does not have a documented business or technical justification.

2. For users who legitimately require this level of access, add or update rationale within the system of record.

3. Reassess user access to ensure alignment with least privilege, reducing broad permissions where narrower privileges are sufficient.

Default Value:
Salesforce does not limit the number of users who may receive these permissions, and does not maintain any system of record regarding administrative access.

SBS-ACS-005: Only Use Custom Profiles for Active Users

NIST SOC 2 ISO 27001

Control Statement:
All active users must be assigned custom profiles. The out-of-the-box standard profiles must not be used.

Description:
Any regular user that can access the org, must use a custom profile. If a user has one of the standard profiles (e.g. “System Administrator”, “Standard User”, “Salesforce - Minimum Access”), the user is non-compliant. This only affects personal users, not machine users that use the default “API Only” permission sets.

Risk: High
Standard profiles are managed by Salesforce, not the organization—meaning Salesforce can enable permissions and object access on these profiles when features are released or platform updates occur without administrator approval. This creates an uncontrolled change vector: users assigned to standard profiles may gain new capabilities unexpectedly, bypassing the organization’s authorization governance. Standard profiles are also overly permissive by default (e.g., “Standard User” grants “View Setup,” “System Administrator” grants developer-level permissions), making it impossible to enforce least privilege. Without custom profiles, organizations cannot investigate authorization changes or maintain accountability for who approved which permissions.

Audit Procedure:

1. Enumerate all human users that are “Active” (`IsActive = true` on the user flag)

2. Flag all users noncompliant that use a standard profile (`IsCustom = false` on the profile metadata)

Remediation:

1. Setup a custom profile for each standard profile that is used

2. Manage permissions and object access on these profiles to be compliant with the other controls of the SBS

3. Assign the new custom profiles to your active users, following the principle of “least privilege access”

Default Value:
Salesforce does not require to create and assign custom profiles.

SBS-ACS-006: Documented Justification for `Use Any API Client` Permission

HIPAA GDPR NIST CCPA/CPRA SOC 2 ISO 27001

Control Statement:
The `Use Any API Client` permission, which bypasses default behavior in orgs with “API Access Control” enabled, must only be assigned to highly trusted users with documented justification and must not be granted to end-users.

Description:
All profiles, permission sets, and permission set groups that grant the `Use Any API Client` permission must be recorded in a designated system of record with a documented business or technical justification. This permission should only be assigned to highly trusted users, such as administrators and developers involved in managing or testing connected app integrations. Any authorization lacking documented rationale is noncompliant.

Risk: Critical
The `Use Any API Client` permission allows users to bypass API Access Control entirely, authorizing any OAuth-connected application without requiring it to be pre-vetted or allowlisted. This establishes an uncontrolled security boundary: users with this permission can grant data access to arbitrary external applications, enabling data exfiltration, unauthorized integrations, and potential account compromise without administrator oversight. Granting this permission to unauthorized personnel completely defeats the purpose of API Access Control, creating a direct path to unauthorized third-party access that requires no other control to fail.

Audit Procedure:

1. Enumerate all profiles, permission sets, and permission set groups that include the `Use Any API Client` permission using Salesforce Setup, Metadata API, Tooling API, or an automated scanner.

2. Compare the enumerated list against the organization's designated system of record for this permission.

3. Verify that every profile, permission set, and permission set group granting `Use Any API Client` has a corresponding entry in the system of record.

4. Confirm that each entry includes:

   - A clear business or technical justification for requiring this permission,
   - Identification of the user role or persona (e.g., administrator, developer, integration manager),
   - Any applicable exception or approval documentation, and
   - Confirmation that the use case is limited to testing or managing connected app integrations.

5. Verify that the permission is not assigned to end-user profiles or permission sets intended for general business users.

6. Flag as noncompliant any authorizations lacking documentation, justification, or assigned to unauthorized user populations.

Remediation:

1. Remove the `Use Any API Client` permission from any profile, permission set, or permission set group that lacks a documented justification or is assigned to end-users.

2. For any authorization that legitimately requires this permission (e.g., administrators or developers testing connected apps), add or update the rationale in the system of record to clearly justify the need and identify the specific role or use case.

3. Ensure that connected apps required for business operations are properly vetted and allowlisted rather than relying on this permission for end-user access.

4. Reconcile and update the system of record to ensure complete and accurate inventory of all assignments of this permission.

Default Value:
The `Use Any API Client` permission is not granted by default in Salesforce. Organizations must explicitly assign this permission to users who require it for legitimate testing or integration management purposes.

SBS-ACS-007: Maintain Inventory of Non-Human Identities

NIST SOC 2 ISO 27001

Control Statement:
Organizations must maintain an authoritative inventory of all non-human identities, including integration users, automation users, bot users, and API-only accounts.

Description:
Non-human identities operate without direct human oversight and often possess persistent credentials with elevated access. Organizations must maintain a complete and current inventory of all such identities to enable effective governance, access reviews, and incident response. The inventory must include identity type, purpose, owner, creation date, and last activity date.

Risk: High
Without a comprehensive inventory of non-human identities, organizations cannot detect, investigate, or respond to security incidents involving automated access. Non-human identities are frequently created for integrations or automation projects and then forgotten—accumulating as orphaned accounts with persistent credentials and elevated access. Security teams cannot assess which automated systems access Salesforce data, identify compromised integration credentials, or scope the impact of a vendor breach. This loss of visibility prevents effective governance of automated access and creates persistent security exposure from untracked machine accounts.

Audit Procedure:

1. Request the organization's inventory of non-human identities
2. Query Salesforce for all users where `IsActive = true` and any of the following conditions apply:
   - Username contains "integration", "api", "bot", "automation", or "service"
   - Profile name contains "Integration", "API", or similar indicators
   - User has "API Only User" permission enabled
   - User is associated with Einstein Bot or Flow automation
3. Compare the inventory to the query results to identify discrepancies
4. Verify the inventory includes: identity name, type, purpose, business owner, creation date, and last login date
5. Confirm the inventory is reviewed and updated at least quarterly

Remediation:

1. Query Salesforce to identify all potential non-human identities using the criteria in the audit procedure
2. For each identified identity, document: name, type (integration/bot/API), purpose, business owner, creation date
3. Establish a process to update the inventory when non-human identities are created, modified, or deactivated
4. Implement quarterly reviews of the inventory to identify and deactivate unused accounts
5. Store the inventory in an authoritative system of record accessible to security and compliance teams

Default Value:
Salesforce does not provide a built-in inventory or classification system for non-human identities. Organizations must create and maintain this inventory manually or through third-party tools.

SBS-ACS-008: Restrict Broad Privileges for Non-Human Identities

NIST SOC 2 ISO 27001

Control Statement:
Non-human identities must not be assigned permissions that bypass sharing rules or grant administrative capabilities unless documented business justification exists.

Description:
Non-human identities should follow the principle of least privilege and be granted only the minimum permissions necessary to perform their intended function. Permissions that bypass object-level or record-level security (such as View All Data, Modify All Data) or grant administrative capabilities (such as Manage Users, Modify Metadata) create significant security risk when assigned to automated accounts. Organizations must document a specific business justification for any non-human identity that requires such permissions.

Risk: High
Without documented justification for broad non-human identity privileges, organizations lose visibility into which automated systems can bypass sharing rules or perform administrative operations. Non-human identities operate without human judgment, making over-privileged automation a high-impact target—compromised credentials can result in complete data extraction, system-wide configuration changes, or persistent backdoor access. Many non-human identities are granted excessive permissions during initial setup and never reviewed, creating long-lived security exposure that security teams cannot detect, investigate, or remediate without knowing which identities have which privileges and why.

Audit Procedure:

1. Using the non-human identity inventory from SBS-ACS-006, identify all non-human identities
2. For each non-human identity, query assigned permissions through profiles, permission sets, and permission set groups
3. Flag any non-human identity with one or more of the following permissions:
   - View All Data
   - Modify All Data
   - Manage Users
   - Author Apex
   - Customize Application

- Any permission that bypasses sharing rules or grants administrative access

4. For each flagged identity, verify that documented business justification exists explaining why the permission is required

5. Confirm the justification was approved by appropriate stakeholders (security, compliance, or management)

Remediation:

1. For each non-human identity with broad privileges, evaluate whether the permission is genuinely required for the identity's function

2. Remove broad privileges that are not necessary; replace with more granular permissions where possible

3. For non-human identities that legitimately require broad privileges, document:

   - Specific business function requiring the permission
   - Why more granular permissions cannot satisfy the requirement
   - Business owner and technical owner
   - Approval from security or compliance team

4. Implement a formal approval process for granting broad privileges to non-human identities

5. Establish periodic review (at least annually) of all non-human identities with broad privileges

Default Value:
Salesforce does not restrict the assignment of broad privileges to non-human identities. Administrators can grant any permission to any user type without requiring justification or approval.

SBS-ACS-009: Implement Compensating Controls for Privileged Non-Human Identities

NIST SOC 2 ISO 27001

Control Statement:
Non-human identities with permissions that bypass sharing rules or grant administrative capabilities must have compensating controls implemented to mitigate risk.

Description:
When non-human identities require broad privileges for legitimate business purposes, organizations must implement defense-in-depth protections to reduce the risk of credential compromise or misuse. Compensating controls include IP address restrictions, OAuth scope limitations, activity monitoring and alerting, credential rotation policies, and dedicated identities per integration. Multiple compensating controls should be implemented based on the sensitivity of accessible data and the scope of granted permissions.

Risk: Moderate
Without compensating controls, privileged non-human identities rely solely on credential secrecy for protection—a single point of failure. Unlike human users, these identities typically use persistent credentials (API keys, OAuth tokens, certificates) that do not expire and are not protected by multi-factor authentication. Compensating controls provide defense-in-depth: IP restrictions limit where credentials can be used, monitoring enables detection of compromise, and credential rotation limits the window of exposure. However, other controls (SBS-ACS-008) still govern whether broad privileges are granted; compensating controls reduce blast radius rather than establishing the primary security boundary.

Audit Procedure:

1. Using the results from SBS-ACS-007, identify all non-human identities with broad privileges that have documented business justification

2. For each privileged non-human identity, verify that at least two of the following compensating controls are implemented:

   - **IP Address Restrictions:** Profile or permission set restricts login to specific IP ranges
   - **OAuth Scope Limitations:** Connected app uses minimal OAuth scopes; refresh tokens have expiration
   - **Activity Monitoring:** Automated monitoring alerts on unusual activity (off-hours access, high volume, geographic anomalies)
   - **Credential Rotation:** Credentials are rotated at least every 90 days
   - **Dedicated Identity:** Separate identity per integration (not shared across multiple systems)

3. Verify that monitoring alerts are actively reviewed and responded to

4. Confirm that compensating controls are documented in the justification for the privileged access

Remediation:

1. For each privileged non-human identity, implement IP address restrictions in the assigned profile or permission set to limit access to known integration sources

2. For OAuth-based integrations, configure connected apps with minimal required scopes and enable refresh token expiration

3. Implement automated monitoring for privileged non-human identity activity using Event Monitoring, Shield Event Monitoring, or third-party SSPM tools

4. Establish credential rotation policies requiring API keys, passwords, and certificates to be rotated at least every 90 days

5. Ensure each integration uses a dedicated non-human identity rather than sharing credentials across multiple systems

6. Document all implemented compensating controls in the access justification

Default Value:
Salesforce does not require or enforce compensating controls for privileged non-human identities. IP restrictions, OAuth scopes, monitoring, and credential rotation must be configured manually by administrators.

SBS-ACS-010: Enforce Periodic Access Review and Recertification

HIPAA GDPR NIST SOC 2 ISO 27001

Control Statement:
All user access and configuration influencing permissions and sharing must be formally reviewed and recertified at least annually by designated busines stakeholders, with documented approval and remediation of unauthorized or excessive access.

Description:
The organization must implement a periodic access review process that ensures all active user access (human and non-human) is intentional, necessary, and aligned with current job responsibilities. An access review encompasses all authorization constructs including user profiles, permission set assignments, permission set group memberships, and role hierarchies. A designated business stakeholder (typically a manager, department lead, or data owner) must certify that each user's access is appropriate, with documented evidence of review and approval. Any access identified as excessive, outdated, or no longer required must be documented and remediated within a defined timeframe. The review must include both individual user access and permission construct usage.

The process must establish clear ownership, defined frequency (minimum annual but may be more frequent for sensitive roles or data), and tracked remediation of findings. Organizations may conduct reviews by individual, by business unit, by data classification, or by application—but all access must be reviewed at least annually in aggregate.

Example implementations:

- Annual access reviews conducted by department managers in Q1, with remediation required within 30 days; tracked in a centralized system of record with sign-off by a representative of Business and Security

- Rolling quarterly reviews where each business unit certifies access for their users on a rotating schedule, with all users reviewed within the calendar year

- Role-based reviews where each application owner certifies all users assigned to specific permission sets or profiles, ensuring full coverage across all users annually

- Sensitive role reviews conducted semi-annually (e.g., System Administrator, Finance, HR users) while standard users are reviewed annually

Risk: **Moderate**
Access review is the foundational control for preventing privilege creep, detecting unauthorized access, and remediating excessive permissions. Without periodic review, users accumulate access over time -- permissions granted for past roles remain after job changes, access added for temporary projects becomes permanent, and no formal mechanism ensures access remains least-privilege. Periodic formal recertification by business stakeholders ensures that access governance remains aligned with organizational reality. Documentation of review creates an audit trail and ensures accountability. Regular remediation prevents drift and maintains the integrity of the permission set model defined in SBS-ACS-001.

Audit Procedure:

1. Understand the organization's documented access review policy, including:

   - Defined frequency and review cycle

   - Designated reviewers and escalation path

   - Intended coverage scope and access types included

   - Expected remediation timeframe for findings

   - System of record for tracking review activity and findings.

2. Assess the recency and regularity of access review execution. Locate the most recent completed acccess review cycle and evaluate whether it aligns with the organization's stated review frequency.

3. Examine a representative sample of access review documentation to asssess consistency of execution:

   - Evidence of review and approval by the designated stakeholder
   - Documentation of review date and scope
   - Any findings, exceptions, or questions raised during the review
   - Appropriateness of sample size relative to the organization's user population and complexity

4. For any access identified as excessive, unauthorized, or not recertified:

   - Assess whether the finding was documented
   - Evaluate what remediation action was taken or whether exceptions were formally approved
   - Compare remediation timing against the defined SLA

5. Assess whether the organization maintains a traceable system of record that documents:

   - Who reviewed what access
   - When the review occurred
   - What was approved or questioned
   - What remediation was required and its completion status

6. Evaluate whether the access review process adequately addresses the organization's primary access constructs, which may include the following types of assignment:

   - User profiles
   - Permission sets
   - Permission set groups
   - Role and role hierarchies
   - Public group
   - Queues
   - Sales Territories
   - Delegated administration or elevated permissions

Remediation:

1. If no access review process exists, establish documented policy including frequency, reviewers, scope, and remediation SLAs.

2. Conduct an initial comprehensive access review of all active users, with business unit or department ownership of sign-off.

3. Identify and remediate all access determined excessive, unauthorized, or inappropriate during the initial review.

4. Implement a system of record (spreadsheet, governance tool, or integrated platform) to track reviews, findings, and remediation.

5. Schedule recurring access reviews at minimum annual frequency, with quarterly reviews for sensitive roles or high-risk data.

6. Document the review process, including templates, stakeholder roles, and escalation procedures.

7. Establish accountability for reviewers and tie review completion to performance management or audit requirements.

Default Value: Salesforce does not automatically initiate user access reviews or require stakeholder recertification of access. Organizations must manually track and document access review processes. Without a defined process, access authorization decisions are not systematically validated, and no audit trail of business stakeholder approval exists.

SBS-ACS-011: Enforce Governance of Access and Authorization Changes

HIPAA GDPR NIST SOC 2 ISO 27001

Control Statement:
All changes to Salesforce user access and authorization must be governed through a documented process that requires approval, records business justification, and produces an auditable record of the change.

Description:
Organizations must enforce governance over all changes that grant, modify, or revoke access within Salesforce. Access and authorization changes must be requested, approved prior to implementation, and traceable to a documented business justification.

This control applies to changes including, but not limited to:

- User creation, modification, or deactivation
- Assignment or removal of profiles, permission sets, or permission set groups
- Changes to profile or permission set permissions
- Role hierarchy changes
- Public group, queue, or territory access changes
- Sharing rule or restriction rule changes affecting access

Access changes must be auditable and aligned to the organization's documented access model. Unauthorized or undocumented changes represent control failure.

Risk: High
Without enforced governance over access changes, organizations lose visibility and control over how privileges are granted and modified. Ad hoc access changes increase the risk of excessive privileges, unauthorized access, and violations of least-privilege principles. The absence of approval, justification, or auditability impairs incident investigation, undermines access reviews, and weakens compliance evidence for audits involving identity, access management, and change control.

Audit Procedure:

1. Retrieve evidence of the organization's documented process governing access and authorization changes.
2. Identify access-related changes made during a representative review period.
3. For a sample of changes, verify:
    - An approval record exists prior to implementation
    - Business justification is documented
    - The change is traceable to an identifiable request
    - The implemented change is recorded in available audit or change history records
4. Identify any access changes lacking approval, justification, or auditability as noncompliant.

Remediation:

1. Establish and document a formal governance process for access and authorization changes.
2. Require approval and business justification for all access modifications.
3. Ensure access changes are recorded in an auditable system of record.

Default Value:
Salesforce does not enforce approval workflows or governance for access and authorization changes. Administrators can directly modify users, permissions, roles, and sharing settings without documented approval or justification. While certain changes may appear in audit logs, governance enforcement is dependent on organizational policy and external processes.

SBS-ACS-012: Classify Users for Login Hours Restrictions

NIST SOC 2 ISO 27001

Control Statement:
Organizations must maintain a documented classification of users requiring login hours restrictions or equivalent off-hours authentication monitoring.

Description:
Organizations must perform risk-based classification to identify users for whom off-hours authentication poses elevated security risk. For each classified user, organizations must either configure login hours restrictions on their profile or implement monitoring and alerting for off-hours authentication. Organizations may classify zero users if documented and reviewed periodically.

Risk: **Moderate**
When privileged accounts authenticate without time restrictions or monitoring, compromised credentials can be exploited during off-hours when detection is less likely. Login hours or monitoring provides defense-in-depth by limiting attack windows or enabling investigation. However, this requires credential compromise and does not establish a primary boundary—authentication controls (SBS-AUTH-001) and IP restrictions (SBS-AUTH-003) remain primary protections.

Audit Procedure:

1. Verify the organization maintains a documented classification identifying users requiring login hours restrictions or monitoring.

2. For classified users, verify login hours are configured or off-hours authentication monitoring is implemented.

3. If zero users are classified, verify this decision is documented with justification and reviewed periodically.

Remediation:

1. Perform risk-based user classification based on privileges and data access.

2. For classified users, either configure login hours on profiles or implement off-hours authentication monitoring with alerting.

3. Document classification and implementation decisions in a system of record.

4. Review during periodic access reviews (SBS-ACS-010).

Default Value:
Salesforce does not enforce login hours or monitor off-hours authentication by default; users can authenticate 24x7 unless explicitly configured.

05

AUTHENTICATION

Authentication

This section defines controls related to user authentication in Salesforce production environments. These controls ensure that organizations implement strong identity verification mechanisms, centralize authentication through Single Sign-On, and maintain proper governance over authentication exceptions to reduce the attack surface and enforce consistent identity management practices.

SBS-AUTH-001: Enable Organization-Wide SSO Enforcement Setting

HIPAA GDPR NIST CCPA/CPRA SOC 2 ISO 27001

Control Statement:
Salesforce production orgs must enable the org-level setting that disables Salesforce credential logins for all users.

Description:
Production orgs must enforce SSO authentication at the organizational level by enabling the "Disable login with Salesforce credentials" setting. This setting prevents all users from authenticating with Salesforce usernames and passwords, requiring SSO authentication instead. Individual users can still be exempted from this requirement using the "Is Single Sign-On Enabled" permission (governed by SBS-AUTH-002).

Risk: Critical
Without the org-level SSO enforcement setting enabled, users can authenticate directly to Salesforce using local credentials—creating a parallel authentication path outside centralized identity management. This establishes an uncontrolled security boundary: password-based attacks (credential stuffing, phishing, brute force) can target Salesforce directly, enabling unauthorized access without requiring any other control to fail. Attackers bypass organizational identity controls, MFA policies, and session management enforced at the IdP layer. This setting is the primary technical control that establishes the SSO security boundary.

Audit Procedure:

1. Retrieve `SingleSignOnSettings` (part of `SecuritySettings`) via Metadata API or navigate to Setup → Single Sign-On Settings in the UI.
2. Verify that `isLoginWithSalesforceCredentialsDisabled` is set to `true`.
3. Flag the org if the setting is not enabled.

Remediation:

1. Navigate to Setup → Single Sign-On Settings.
2. Enable **Disable login with Salesforce credentials**.
3. Validate that SSO is properly configured and functional before enabling this setting to prevent lockout.
4. Ensure approved break-glass or administrative accounts have the "Is Single Sign-On Enabled" permission removed via their profiles or permission sets so they can still authenticate if needed.

Default Value:
By default, Salesforce does not enable "Disable login with Salesforce credentials." All users can authenticate with Salesforce usernames and passwords unless this setting is explicitly enabled.

SBS-AUTH-002: Govern and Document All Users Permitted to Bypass Single Sign-On

HIPAA GDPR NIST SOC 2 ISO 27001

Control Statement:
All users who do not have the "Is Single Sign-On Enabled" permission must be explicitly authorized, documented in a system of record, and limited to approved administrative or break-glass use cases.

Description:
When the org-level SSO enforcement setting (SBS-AUTH-001) is enabled, users without the "Is Single Sign-On Enabled" permission can still authenticate using Salesforce credentials—effectively bypassing SSO. Production orgs must maintain an authoritative inventory of all such accounts, documenting their justification, role, and approval. These accounts should be limited to approved administrative or break-glass scenarios only.

Risk: Moderate
Users permitted to bypass SSO represent exceptions to centralized identity governance. Without formal documentation and approval, these accounts can proliferate unnoticed—reducing visibility into access patterns and undermining audit readiness. However, this control provides assurance and governance rather than establishing a security boundary. Undocumented exceptions increase operational risk and reduce audit readiness but require credential compromise for direct security impact.

Audit Procedure:

1. Query all user records to identify users who do **not** have the "Is Single Sign-On Enabled" (`PermissionsIsSsoEnabled`) permission assigned through their profile or permission sets.

2. Verify each identified user appears in the approved system-of-record inventory with documented business justification, owner, and approval.

3. Confirm each exception is authorized for administrative or break-glass purposes only.

4. Validate that these accounts follow strong local authentication controls (e.g., strong password policies, MFA if applicable).

5. Flag any user without documented approval.

6. (Optional) Download API Total Usage logs (EventLogFile - ApiTotalUsage, available in free tier of Event Monitoring) to monitor SSO bypass account activity:

   - Filter API activity by users identified as SSO bypass accounts.

   - Review frequency and timing of API calls to verify usage aligns with documented break-glass purposes.

   - Flag any SSO bypass accounts with regular or unexpected API activity for review against documented justifications.

Remediation:

1. Create or update a formal inventory documenting all SSO-bypass users with their business justification, owner, and approval date.

2. For any undocumented or unjustified users: assign the "Is Single Sign-On Enabled" permission via their profile or permission sets to remove SSO-bypass capability.

3. Ensure all documented exceptions adhere to least-privilege access and strong authentication controls.

4. Establish periodic (e.g., quarterly) review of all SSO-bypass accounts.

Default Value:
By default, no users are assigned the "Is Single Sign-On Enabled" permission, meaning all users can authenticate with Salesforce credentials. Once SBS-AUTH-001 is implemented and the org-level setting is enabled, users must be explicitly assigned this permission to require SSO authentication.

SBS-AUTH-003: Prohibit Broad or Unrestricted Profile Login IP Ranges

NIST SOC 2 ISO 27001

Control Statement:
Profiles in Salesforce production orgs must not contain login IP ranges that effectively permit access from the full public internet or other overly broad ranges that bypass network-based access controls.

Description:
Any profile-level login IP range must reflect explicitly authorized organizational network boundaries. Profiles must not include universally permissive ranges—such as `0.0.0.0–255.255.255.255` or other combinations that allow access from all or nearly all IP addresses—as these configurations disable intended Salesforce network restrictions and undermine authentication controls.

Risk: Moderate
Overly broad login IP ranges effectively disable network-based access controls, allowing authentication from any location on the internet. However, exploitation requires credentials to be compromised first—this control provides defense-in-depth rather than establishing a primary security boundary. When authentication controls (SBS-AUTH-001) are enforced, IP restrictions serve as an additional layer that limits the blast radius of credential compromise.

Audit Procedure:

1. Retrieve all profile login IP ranges via **Setup** → **Profiles** → **Login IP Ranges** or by querying the Profile metadata (`loginIpRanges` field) using the Metadata API.

2. For each profile, enumerate all configured login IP ranges.

3. Identify any ranges that:

   - Cover the entire IPv4 space, or

   - Represent effectively unrestricted access (e.g., `0.0.0.0–255.255.255.255`, `1.1.1.1–255.255.255.255`, or similar patterns).

4. Confirm that all IP ranges align with organizational security policy and defined network boundaries.

5. Flag any profile with an impermissible or overly broad range.

6. Download API Total Usage logs (EventLogFile - ApiTotalUsage, available in free tier of Event Monitoring) to validate IP restrictions are effective:
    - Extract unique `CLIENT_IP` values from recent API activity.
    - Compare against documented approved organizational network ranges.
    - Identify any new or unexpected IP addresses making API calls.
    - Cross-reference unusual IPs with profile assignments to identify potential policy gaps.

Remediation:

1. Remove any profile login IP ranges that effectively grant unrestricted global access.
2. Replace them with IP ranges that correspond to approved corporate networks, office locations, VPN ingress points, or other authorized infrastructure.
3. Validate that updated network restrictions do not block legitimate access paths and that users can authenticate through sanctioned networks.
4. Establish an internal governance process to review and approve all future additions of profile login IP ranges.

Default Value:
Salesforce profiles do not include login IP ranges by default; they must be explicitly configured.

SBS-AUTH-004: Enforce Strong Multi-Factor Authentication for External Users with Substantial Access to Sensitive Data

HIPAA GDPR NIST CCPA/CPRA SOC 2 ISO 27001

Control Statement:
All Salesforce interactive authentication flows for external human users with substantial access to sensitive data must enforce multi-factor authentication that includes at least one strong authentication factor.

Description:
Salesforce must be configured so that every interactive login method available to external human users with substantial access to sensitive data enforces multi-factor authentication using either a strong second factor in addition to a password, or a passwordless flow requiring two or more factors with at least one strong factor, regardless of whether authentication is performed directly by Salesforce or via a single sign-on identity provider.

Organizations must document in their system of record:

1. **Definition of "substantial access to sensitive data"** — The organization's interpretation of what data classifications or access levels constitute substantial access for purposes of this control.
2. **Identification of in-scope users** — Either a list of specific users, or the combination of Salesforce access controls (profiles, permission sets, etc.) that result in substantial access.

Example system-of-record entry:

- **Definition:** "Substantial access to sensitive data" means access to Personally Identifiable Information (as defined under GDPR) relating to individuals other than the user themselves, or access to Special Category Data (as defined under GDPR) relating to any individual.
- **In-scope users:** All users assigned the "Service Channel Partner" profile.

For the purposes of this control, a strong authentication factor is defined as an authentication factor that is resistant to phishing, replay, and credential stuffing attacks. Acceptable strong authentication factors include:

- Push-notification based authenticator app such as Salesforce Authenticator or Okta Verify
- RFC 6238 compliant Time-based One-Time Password Algorithm (TOTP) authenticator app
- FIDO2 hardware key compliant with either WebAuthn or U2F standard
- Biometric authentication such as Touch ID or Windows Hello

Risk: Critical
Without enforced multi-factor authentication, external users with substantial access to sensitive data can authenticate using only a password—establishing a single point of failure for the authentication boundary. External users present elevated credential risk due to weaker identity proofing, less organizational oversight, and exposure to consumer-grade phishing attacks. Attackers who compromise a single password through phishing, credential stuffing, or account takeover gain direct access to sensitive data without requiring any other control to fail. This creates an unprotected authentication path to high-value data that bypasses the defense-in-depth protections applied to internal users.

Audit Procedure:

1. Enumerate all active external human users with substantial access to sensitive data.

2. Validate that in-scope users have the “Multi-Factor Authentication for User Interface Logins” permission through profiles or permission sets.

3. Flag any in-scope users who lack the “Multi-Factor Authentication for User Interface Logins” permission.

Remediation:

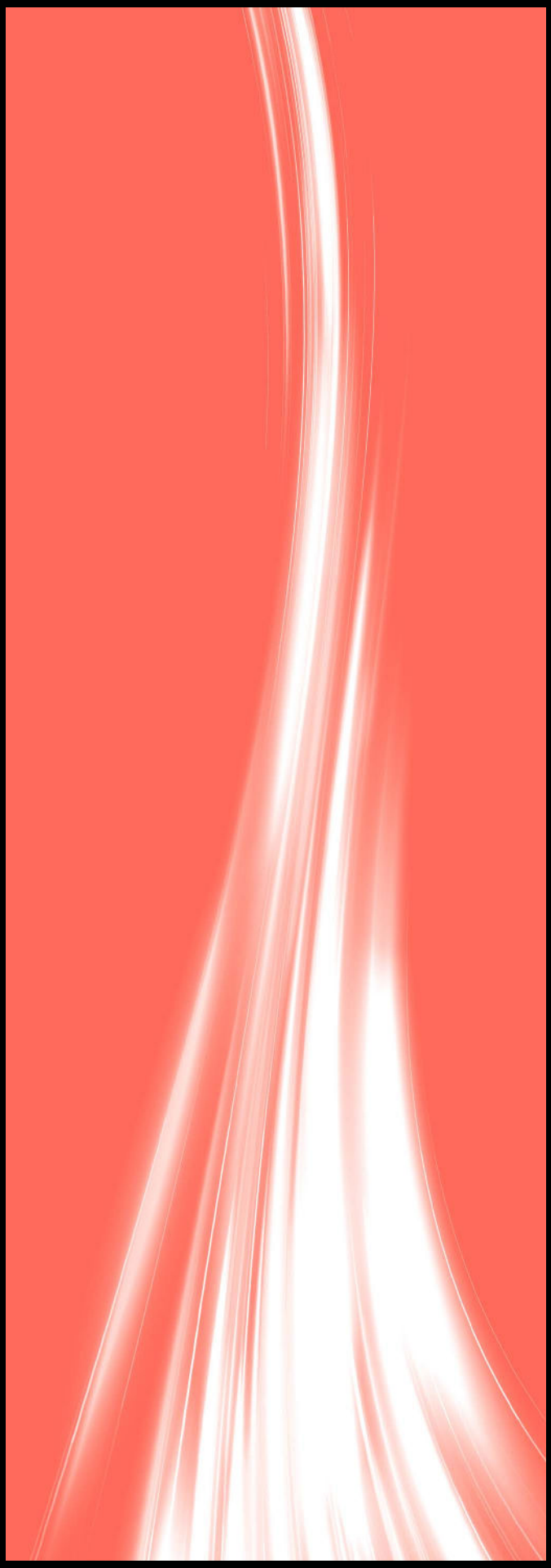
1. Apply the “Multi-Factor Authentication for User Interface Logins” permission through profiles or permission sets for all active external users with substantial access to sensitive data.

2. Configure suitable strong second-factor options in Setup -> Identity -> Identity Verification (e.g., authenticator app, FIDO2 security key).

Default Value:
By default, Salesforce does not enforce strong multi-factor authentication for all external user login flows, and external users may authenticate using single-factor or weak-factor methods unless explicitly restricted by configuration.

References:

- Salesforce Documentation: Multi-Factor Authentication
- NIST SP 800-63B Authentication and Lifecycle Management
- NIST SP 800-53 IA-2



06

CODE SECURITY

Code Security

This section defines controls related to secure development practices for Salesforce code, including Apex, Lightning Web Components, and other programmatic assets. These controls ensure that organizations implement quality gates, peer review, and automated security testing within their development lifecycle to prevent vulnerable or flawed code from entering production environments.

SBS-CODE-001: Mandatory Peer Review for Salesforce Code Changes

ISO 27001

Control Statement:
All Salesforce code changes must undergo peer review and receive approval before merging into any production-bound branch.

Description:
Organizations must configure their source control system to require at least one peer reviewer to approve all changes to Apex, Lightning Web Components, and other programmatic assets before those changes are merged into branches used for production deployments.

Risk: Moderate
Without mandatory peer review, a single developer—whether compromised, malicious, or simply mistaken—can introduce insecure or flawed code directly into the deployment pipeline. This eliminates shared oversight of changes to sensitive business logic, allowing vulnerabilities, backdoors, or destructive changes to reach production without independent human verification before deployment.

Audit Procedure:
1. Inspect source control settings to confirm merge rules require peer review on production-bound branches.

2. Review merge history or representative pull requests to verify peer approvals were recorded.

3. Confirm that peer review processes include security checks such as verifying logging statements do not expose sensitive data

4. Flag any repositories or branches that allow merging without peer approval.

Remediation:
1. Update branch protection rules to require peer review before merge.

2. Train developers on the peer review workflow, including security checks such as identifying sensitive data in logging statements.

3. Block direct commits to production-bound branches.

Default Value:
Salesforce does not enforce code review requirements; these controls depend on the organization's source control configuration.

SBS-CODE-002: Pre-Merge Static Code Analysis for Apex and LWC

ISO 27001

Control Statement:
Static code analysis with security checks for Apex and Lightning Web Components must execute successfully before any code change is merged into a production-bound branch.

Description:
Organizations must implement static application security testing (SAST) in their CI/CD pipeline and configure it to run prior to merge, enforcing security rulesets that detect
vulnerabilities specific to Apex and LWC.

Risk: Moderate
Without enforced static code analysis, known vulnerability patterns in Apex and LWC—such as SOQL injection, insecure data exposure, and improper access control—may enter
production undetected. This increases the likelihood of exploitable flaws persisting in deployed code, creating potential vectors for data breaches or unauthorized access that human reviewers may not catch.

Audit Procedure:
1. Inspect CI/CD pipeline configuration to confirm a static code analysis step runs before merges.

2. Verify the SAST tool includes security rulesets for Apex and Lightning Web Components.

3. Review pipeline logs from representative merges to ensure scans executed and passed.

4. Flag pipelines or branches missing enforced pre-merge scanning.

Remediation:
1. Integrate static code analysis into the CI/CD pipeline for all production-bound branches.

2. Enable Apex and LWC security rulesets within the scanning tool.

3. Configure pipelines to block merges when static analysis fails.

Default Value:
Salesforce does not provide or enforce static code analysis; organizations must implement external SAST tooling.

SBS-CODE-003: Implement Persistent Apex Application Logging

GDPR ISO 27001

Control Statement:
Organizations must implement an Apex-based logging framework that writes application log events to durable Salesforce storage and must not rely on transient Salesforce debug logs for operational or security investigations.

Description:
The organization must deploy a dedicated Apex logging framework—custom-built, open source, or vendor-provided—that programmatically captures application-level log events and stores them in durable Salesforce data structures, such as custom objects, to ensure logs persist beyond the limitations of Salesforce debug logs.

Risk: High
Salesforce debug logs are transient, size-limited, and automatically purged—making them unsuitable for forensic analysis or security investigations. Without persistent application logging, organizations cannot reliably reconstruct access patterns, detect anomalous behavior, or investigate security incidents after the fact. This impairs the ability to identify compromise, attribute malicious activity, or understand the scope of a breach—significantly extending attacker dwell time and reducing accountability for actions taken within the system.

Audit Procedure:
1. Review the Salesforce org for the presence of an Apex logging framework implemented as one or more Apex classes dedicated to log generation and persistence.

2. Verify that the framework writes logs to durable storage, such as a custom object purpose-built for log retention.

3. Confirm that operational and security investigations rely on this persistent logging mechanism rather than Salesforce debug logs.

4. Inspect recent log records to ensure the framework is actively capturing runtime events.

Remediation:
1. Implement or install an Apex logging framework designed for persistent log storage.

2. Create or configure a custom object (or equivalent durable storage) to store log records.

3. Update Apex code to route log events through the framework.

4. Train engineering and security teams to use persistent logs instead of debug logs for investigations.

Default Value:
Salesforce does not provide persistent application-level logging by default; debug logs are transient, size-limited, and automatically purged.

SBS-CODE-004: Prevent Sensitive Data in Application Logs

HIPAA GDPR CCPA/CPRA ISO 27001

Control Statement:
Custom application logging frameworks and Salesforce system logging mechanisms must not capture, store, or transmit credentials, authentication tokens, personally identifiable information (PII), regulated data, or other sensitive values in log messages or structured log fields.

Description:
Organizations must ensure that custom Apex logging frameworks and Salesforce debug logs exclude sensitive data from all log outputs. This applies to:

- Custom logging frameworks writing to custom objects or external systems
- `System.debug()` statements that write to Salesforce debug logs
- Error handling routines that log exception details

Logging implementations must prevent the capture of:

- Authentication credentials including passwords, API keys, OAuth tokens, session identifiers, and client secrets
- Personally identifiable information and regulated data such as SSNs, financial account numbers, credit card details, and protected health information
- Full SOQL query results containing sensitive fields (log record IDs or counts instead)
- Request/response payloads containing authentication headers or authorization tokens
- Unmasked field values from high-sensitivity objects (mask or tokenize before logging)

Organizations must implement mechanisms to prevent sensitive data from being written to logs, such as sanitization functions that developers invoke to remove or mask data before log events are persisted, or establishing code review requirements that check for sensitive data exposure in logging calls.

Risk: Critical
When application logs capture sensitive data, attackers who compromise low-privilege accounts with Read access to log storage can exfiltrate credentials, PII, or regulated data without triggering access controls on the original source objects—transforming a logging framework into a data leakage vector. In regulated industries, a compromised administrator querying log objects can extract thousands of customer records in minutes, with the audit trail showing only "legitimate" queries. During breach investigations, logs become evidence of regulatory violations rather than forensic tools, triggering consent orders and significant financial penalties.

Audit Procedure:

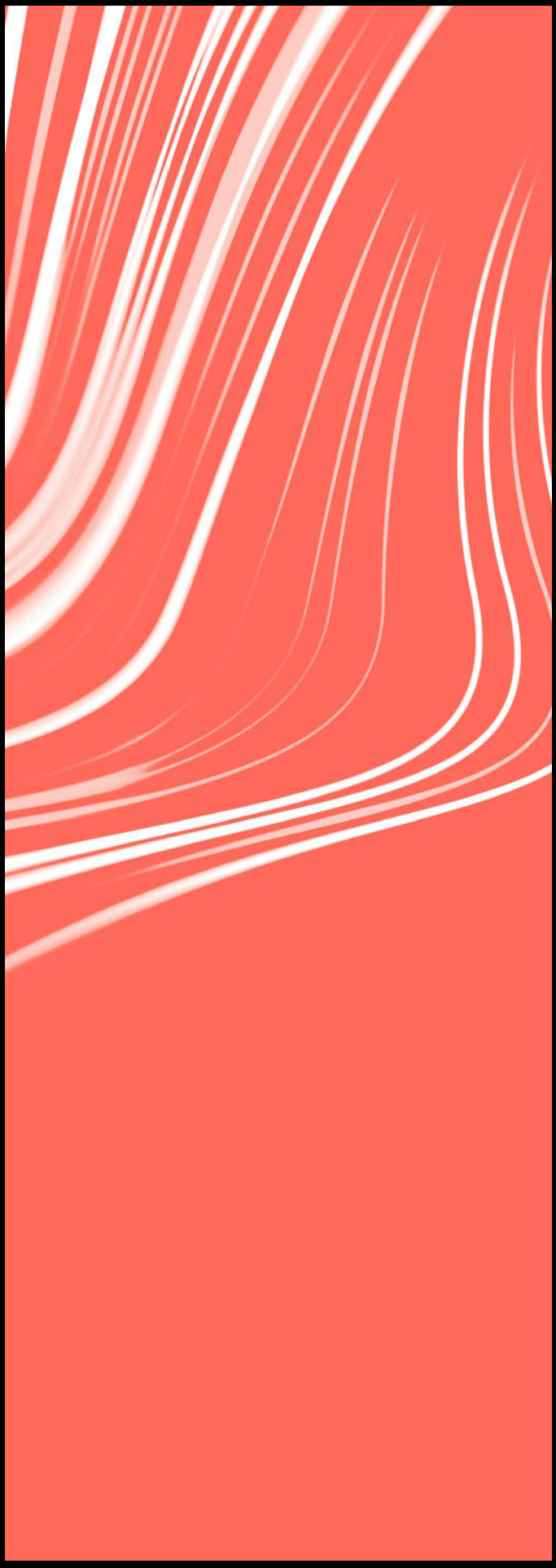
1. Sample representative Apex classes from high-risk areas (customer-facing functionality, payment processing, authentication flows) to identify logging statements in both custom frameworks and `System.debug()` calls.
2. Examine log message construction to detect patterns that may capture the types of sensitive data listed above.
3. Query recent log records stored in custom objects and review Salesforce debug logs to inspect actual log content for sensitive data:
   - Search for patterns matching SSNs, credit card numbers, email addresses, phone numbers
   - Identify authentication tokens, session IDs, or API keys in log messages
   - Flag any log records containing regulated data or PII
4. Verify that mechanisms exist to prevent sensitive data from being logged (such as sanitization functions, code review checks, or automated validation).

Remediation:

1. Implement mechanisms to prevent sensitive data from being written to logs:

```
public class SecureLogger {
    public static void logInfo(String message,
    Map<String, Object> context) {
        // Sanitize context before logging
        Map<String, Object> sanitized =
        sanitizeContext(context);
        Logger.info(message, sanitized);
    }

    private static Map<String, Object>
    sanitizeContext(Map<String, Object> ctx) {
        Map<String, Object> result =
        new Map<String, Object>();
        for (String key : ctx.keySet()) {
            // Mask sensitive fields, log IDs
            instead of full records
            if (key.containsIgnoreCase('password') ||
                key.containsIgnoreCase('token') ||
                key.containsIgnoreCase('ssn')) {
                result.put(key, '***REDACTED***');
            } else if (ctx.get(key) instanceof
            SObject) {
                result.put(key, ((SObject)ctx.
                get(key)).Id);
            } else {
                result.put(key, ctx.get(key));
            }
        }
        return result;
    }
}
```



2. Audit existing log records in custom objects and purge Salesforce debug logs containing sensitive data.

3. Update logging calls to avoid capturing sensitive data:

```
// BAD - logs full account with SSN field
System.debug('Processing: ' + acc);
Logger.info('Processing account', new Map<String,
Object>{'account' => acc});

// GOOD - logs only record ID
System.debug('Processing account: ' + acc.Id);
SecureLogger.logInfo('Processing account', new
Map<String, Object>{
    'accountId' => acc.Id,
    'recordCount' => 1
});

```

4. Consider implementing compensating controls such as automated testing that validates log outputs for sensitive data patterns, code review checks for logging security, or static analysis rules that detect common sensitive data exposure patterns.

Default Value:
Salesforce does not prevent or sanitize sensitive data in custom application logs or system debug logs; developers bear full responsibility for ensuring log content complies with data protection requirements.

07

CUSTOMER PORTALS

Customer Portals

This section defines controls related to secure configuration and development practices for Salesforce customer portals, including Experience Cloud sites, Communities, and other external-facing Salesforce platforms. These controls ensure that organizations implement proper access controls, data isolation, and secure coding practices when exposing Salesforce functionality to external users.

SBS-CPORTAL-001: Prevent Insecure Direct Object Reference (IDOR) in Portal Apex

HIPAA GDPR NIST CCPA/CPRA SOC 2 ISO 27001

Control Statement:
All Apex methods exposed to Experience Cloud or customer portal users must enforce server-side authorization for every record accessed or modified. User-supplied parameters (including record IDs, filters, field names, or relationship references) must not be trusted as the basis for access control and must be validated against the running user's sharing, CRUD, and FLS permissions before use.

Description:
Portal-exposed Apex methods are callable by external users and therefore must not rely on client-provided identifiers or query inputs to determine record access. Accepting record IDs, filter criteria, field lists, or relationship paths without validating the running user's access creates Insecure Direct Object Reference (IDOR) vulnerabilities.

Methods must:

- Run `with sharing` unless explicitly justified.
- Enforce object- and field-level permissions.
- Prevent user-controlled SOQL structure (e.g., dynamic WHERE clauses or field lists).
- Restrict data scope to records the authenticated user is authorized to access according to business rules.

Parameters may be accepted when required for legitimate functionality, but must be validated server-side before querying or performing DML.

Risk: Critical
If portal-exposed Apex trusts user-controlled parameters to determine record access, external users can manipulate inputs to retrieve or modify unauthorized records. This may enable record enumeration, data exfiltration, or data corruption. IDOR vulnerabilities represent a critical authorization boundary failure.

Audit Procedure:

1. Identify all Apex classes exposing `@AuraEnabled`, `@InvocableMethod`, or `@RestResource` methods accessible to portal users.
2. Review method parameters of type `Id`, `String`, collections, or maps that could influence record access.
3. Verify that:
   - Classes run `with sharing` or implement equivalent authorization checks.
   - Record access is validated before query or DML.
   - CRUD and FLS are enforced.
   - Dynamic SOQL does not incorporate unsanitized user input.
4. Attempt to access unauthorized records by manipulating record IDs or filter inputs from a portal user session.
5. Flag any method that relies solely on user-supplied parameters to control record access as noncompliant.

Remediation:

- Enforce `with sharing` on portal-facing classes by default.
- Scope queries using the running user's context where possible.
- If record IDs are accepted as parameters, verify access using sharing or `UserRecordAccess` before returning or modifying data.
- Remove user-controlled query structure and whitelist allowable filter inputs.
- Enforce CRUD and FLS on all returned or modified records.

Default Value:
Salesforce does not validate user-supplied parameters in custom Apex. Developers are responsible for implementing server-side authorization controls in all portal-exposed methods.

SBS-CPORTAL-002: Restrict Guest User Record Access

HIPAA GDPR NIST CCPA/CPRA SOC 2 ISO 27001

Control Statement:
Unauthenticated guest users in customer portals must be restricted to authentication and registration flows only, with no direct access to business objects or custom Apex methods that query organizational data.

Description:
Organizations must configure customer portal guest user profiles to prohibit access to all business-related standard and custom objects, limiting guest user capabilities exclusively to authentication flows (login, registration, password reset, self-service account creation). Guest users must not be granted object-level permissions, field-level access, or the ability to invoke custom Apex methods that return organizational data.

When business requirements necessitate limited guest user access to specific public data (such as knowledge articles, public case submission forms, or product catalogs), organizations must:

- Implement a dedicated service layer architecture where controllers invoke secure service classes that perform explicit access validation
- Use allowlist-based data access (explicitly define queryable records, never accept parameters)
- Apply additional validation using `UserInfo.getUserType() == 'Guest'` to enforce restricted logic paths
- Consider rate limiting and CAPTCHA protection to prevent enumeration attacks

Risk: Critical
Guest users represent the highest-risk trust boundary in Salesforce portals—they are unauthenticated, have zero accountability, generate minimal audit trail, and operate with potential adversarial intent. When guest users are granted object permissions or can invoke custom Apex methods, attackers can systematically enumerate organizational data without even creating an account. Historical Salesforce security updates have repeatedly addressed guest user permission defaults because vendors consistently misconfigure this boundary. A single guest-accessible method that queries user records, cases, accounts, or custom objects creates a public API for data exfiltration accessible to anyone on the internet. This constitutes a Critical boundary violation: unauthenticated attackers access organizational data with no authentication required.

Audit Procedure:

1. Identify all guest user profiles used by customer portal sites (typically named "Site Guest User" or similar).
2. Review object-level permissions for guest user profiles and verify that all business-related standard and custom objects have Read, Create, Edit, Delete permissions set to disabled.
3. Enumerate all custom Apex classes containing `@AuraEnabled` methods and verify that none are accessible to guest users (either by checking profile permissions or testing invocation from guest context).
4. For any guest-accessible functionality beyond authentication flows, verify implementation of service layer architecture with explicit access controls.
5. Test by accessing the portal without authentication and attempting to invoke Apex methods or query objects via built-in Lightning controllers.
6. Flag any guest user object permissions or method access as noncompliant.

Remediation:

1. Remove all object-level permissions from guest user profiles except those explicitly required for authentication flows.
2. Audit and remove guest user access to any custom Apex methods that query or return organizational data.
3. For public data requirements (knowledge articles, case submission), implement service layer pattern:

```
@AuraEnabled
public static List<Knowledge__kav> getPublicArticles() {
    if (UserInfo.getUserType() == 'Guest') {
        // Allowlist-based, no parameters accepted
        return [SELECT Id, Title, Summary FROM
        Knowledge__kav
                WHERE PublicationStatus = 'Online'
                AND IsVisibleInPkb = true
                LIMIT 10];
    }
    throw new AuraHandledException('Access denied');
}

```

4. Implement network-level rate limiting and CAPTCHA for guest-accessible endpoints.
5. Review Salesforce security updates and apply guest user permission restrictions from recent releases.

Default Value:
Salesforce has progressively restricted guest user default permissions in recent releases, but older orgs may retain permissive configurations. Guest user profiles do not prevent object access or Apex invocation by default—administrators must explicitly configure restrictions.

SBS-CPORTAL-003: Inventory Portal-Exposed Apex Classes and Flows

ISO 27001

Control Statement:
Organizations must maintain an authoritative inventory of all Apex classes and Autolaunched Flows exposed to Experience Cloud sites, documenting which components are accessible to external and guest users.

Description:
Organizations must document all Apex classes with `@AuraEnabled` methods and all Autolaunched Flows that can be invoked from Experience Cloud sites. The inventory must include which portal user profiles and permission sets can access each component.

Risk: High
Without a complete inventory of portal-exposed components, organizations cannot assess their external attack surface or enforce security reviews for externally accessible code. Security teams lose visibility into which business logic external users can invoke, preventing effective security testing, incident response, and access governance. This impairs the ability to detect unauthorized exposure of sensitive functionality or identify components requiring security hardening.

Audit Procedure:
1. Request the organization's inventory of portal-exposed Apex classes and Flows from the designated system of record.
2. Query all Apex classes with `@AuraEnabled` methods accessible to portal user profiles.
3. Query all Autolaunched Flows invoked from Experience Cloud pages or components.
4. Verify each component appears in the inventory with documentation of which portal profiles can access it.
5. Flag any portal-exposed component missing from the inventory as noncompliant.

Remediation:
1. Enumerate all Apex classes containing `@AuraEnabled` methods.
2. Enumerate all Autolaunched Flows embedded in Experience Cloud sites.
3. For each component, document which portal user profiles and permission sets have access.
4. Store the inventory in the designated system of record.
5. Establish a process to update the inventory when new components are exposed to portals.

Default Value:
Salesforce does not require or maintain an inventory of portal-exposed components.

SBS-CPORTAL-004: Prevent Parameter-Based Record Access in Portal-Exposed Flows

HIPAA GDPR NIST CCPA/CPRA SOC 2 ISO 27001

Control Statement:
Autolaunched Flows exposed to customer portal users must not accept user-supplied input variables that directly determine which records are accessed.

Description:
Flows invoked from Experience Cloud must not accept input variables for record IDs, object names, or filter criteria. All record access must be derived from the authenticated user's context using $User.ContactId or similar user context resources.

Risk: Critical
Flows accepting user-controlled input variables for record access create IDOR vulnerabilities allowing external users to access any record in the org. Because Autolaunched Flows run in system context without sharing by default, a single flow accepting a record ID input parameter bypasses all permissions and sharing rules. This constitutes a Critical boundary violation: unauthorized users access data they should never see, with no compensating controls required to fail.

Audit Procedure:

1. Using the inventory from SBS-CPORTAL-003, identify all portal-exposed Autolaunched Flows.

2. For each flow, examine input variables for types that could contain record IDs (Text, Record, Text Collection).

3. Review flow logic to determine if input variables influence Get Records, Update Records, or Delete Records elements.

4. Flag any flow accepting user-supplied input variables that control record access as noncompliant.

Remediation:

1. Refactor flows to eliminate input variables controlling record access.

2. Derive accessible records from authenticated user context (e.g., `$User.Id`, `$User.ContactId`, `$User.AccountId`).

3. Configure flows to run in user context ("With Sharing") where available.

Default Value:
Salesforce does not prevent flows from accepting user-supplied input variables. Autolaunched Flows run in system context without sharing by default.

SBS-CPORTAL-005: Conduct Penetration Testing for Portal Security

GDPR ISO 27001

Control Statement:
Organizations with Experience Cloud sites must conduct penetration testing of portal security controls before initial go-live and subsequently after major releases or on a defined cadence.

Description:
Penetration testing validates that authentication boundaries, authorization controls, and data access restrictions function correctly under adversarial conditions. Testing must target portal-exposed Apex classes, Flows, and components, including parameter manipulation, IDOR attempts, and privilege escalation scenarios. Organizations determine ongoing testing frequency based on regulatory requirements and change velocity.

Risk: High
Without regular penetration testing, organizations cannot verify that portal security controls function correctly when adversaries attempt to exploit them. Configuration audits verify settings exist but cannot validate runtime behavior under attack. Undetected vulnerabilities in portal-exposed components allow unauthorized data access.

Audit Procedure:

1. Verify penetration testing was conducted before initial portal go-live.

2. Verify the organization has defined an ongoing testing cadence based on regulatory requirements and change frequency.

3. Request documentation of the most recent portal penetration test.

4. Verify testing occurred according to the defined cadence or after major releases.

5. Confirm test scope included portal-exposed Apex classes and Flows.

6. Review test report for identified vulnerabilities and remediation status.

7. Flag as noncompliant if no go-live testing occurred, ongoing testing does not follow the defined cadence, or if high/critical findings remain unremediated.

Remediation:

1. Conduct penetration testing before initial portal go-live.

2. Define ongoing testing cadence based on regulatory requirements and release frequency.

3. Engage qualified penetration testers with Salesforce Experience Cloud expertise.

4. Define test scope covering all portal-exposed components.

5. Conduct testing according to defined cadence and after major portal changes.

6. Remediate identified vulnerabilities before production deployment.

Default Value:
Salesforce does not require or conduct penetration testing of customer implementations.

08

DATA SECURITY

Data Security

This section defines controls related to the protection, classification, and management of data stored within Salesforce environments. These controls ensure that organizations maintain visibility over sensitive data locations, implement appropriate safeguards for regulated information, and establish processes to support privacy compliance and breach response capabilities.

SBS-DATA-001: Implement Mechanisms to Detect Regulated Data in Long Text Area Fields

GDPR CCPA/CPRA

Control Statement:
The organization must implement a mechanism that continuously or periodically analyzes the contents of all Long Text Area fields to identify the presence of regulated or personal data.

Description:
Salesforce organizations must employ a technical or procedural mechanism to inspect the stored values of all Long Text Area (LTA) fields for regulated data, including PII, PHI, or GDPR-governed attributes. For example, a sales representative may enter customer phone numbers, email addresses, or even payment card details into an Opportunity "Follow-Up Notes" field, creating unstructured storage of regulated data that is difficult to detect without systematic scanning.

Risk: High
Long Text Area fields often contain unstructured, user-entered information that may include sensitive personal data. Without a detection mechanism, regulated data accumulates in unknown locations—obstructing compliance with GDPR Right to Erasure, CCPA deletion requests, and similar privacy obligations. During a security incident, the inability to identify which fields contain personal information makes it impossible to accurately assess exposure, determine the scope of compromised records, or fulfill breach notification requirements. This governance gap significantly impairs incident response and creates ongoing regulatory liability.

Audit Procedure:
1. Identify all Long Text Area fields using Salesforce metadata.
2. Determine whether the organization has a mechanism that scans the contents of each LTA field for regulated data.
3. Confirm that scanning occurs continuously or on a defined recurring schedule.
4. Review scan logs, detection outputs, or configuration details to verify that the mechanism is operational.
5. Validate that all LTA fields across all objects are included in scope.
6. Determine compliance based on whether such a mechanism exists and is functioning.

Remediation:
1. Deploy or configure a tool, script, or process capable of analyzing the contents of LTA fields for regulated data.
2. Ensure scans run continuously or on a recurring schedule.
3. Confirm all applicable fields across all objects are included.
4. Document the scanning process and store execution evidence for audit support.

Default Value:
Salesforce does not natively scan the contents of Long Text Area fields for regulated data.

SBS-DATA-002: Maintain an Inventory of Long Text Area Fields Containing Regulated Data

GDPR CCPA/CPRA

Control Statement:
The organization must maintain an up-to-date inventory of all Long Text Area fields that are known or detected to contain regulated or personal data.

Description:
Organizations must maintain a documented inventory listing each Long Text Area field that contains or is reasonably expected to contain regulated data, based on scanning outputs or operational use. For example, if a Case "Internal Comments" field routinely includes client account numbers or health-related information entered by support agents, that field must appear in the inventory and be tracked accordingly.

Risk: Moderate
Without a current inventory of fields containing regulated data, organizations cannot systematically apply appropriate protection, retention, or access controls to sensitive data locations—and may be unable to fulfill privacy obligations such as GDPR's Right to Erasure or CCPA deletion requests that require knowing all locations where personal data is stored. During audits or breach investigations, the absence of a maintained inventory delays response times and may result in incomplete remediation or missed data locations.

Audit Procedure:

1. Obtain the organization's documented inventory of Long Text Area fields containing regulated data.
2. Compare the inventory against Salesforce metadata to confirm all relevant fields are included.
3. Review scan results or administrative evidence demonstrating how fields were identified.
4. Verify that the inventory includes object name, field API name, data classification, and last review date.
5. Determine whether the inventory is maintained and current; missing, outdated, or incomplete inventories indicate noncompliance.

Remediation:

1. Generate an inventory using scan results, administrative review, and metadata analysis.
2. Document all LTA fields containing regulated data and classify the associated data types.
3. Establish a recurring review cycle to update the inventory.
4. Integrate the inventory into governance functions such as retention, DLP, access reviews, and breach response planning.

Default Value:
Salesforce does not maintain or provide an inventory of Long Text Area fields containing regulated data.

SBS-DATA-003: Maintain Tested Backup and Recovery for Salesforce Data and Metadata

GDPR ISO 27001

Control Statement:
Salesforce production orgs must maintain a documented backup and recovery capability for Salesforce data and metadata, and must test restoration on a defined schedule.

Description:
Organizations must back up Salesforce data and metadata using Salesforce exports, APIs, source control, or third-party tooling, and must document recovery procedures sufficient to restore a known-good state.

Risk: High
Without reliable backups and tested restoration procedures, organizations cannot recover from accidental deletion, malicious data destruction, configuration corruption, or ransomware-like events. This impairs incident response, business continuity, and the ability to validate data integrity after security events or outages.

Audit Procedure:

1. Obtain the documented backup and recovery policy covering Salesforce data and metadata.

2. Verify that backups are performed on a defined schedule and retained per policy.

3. Review evidence of a completed restoration test within the defined testing interval.

4. Confirm that backup storage is protected with appropriate access controls.

Remediation:

1. Implement or configure a backup solution for Salesforce data and metadata.

2. Define backup frequency, retention, and storage protections.

3. Execute and document restoration tests on the defined schedule.

4. Update recovery procedures based on test results.

Default Value:
Salesforce does not provide automatic, comprehensive backup and restore for all data and metadata by default.

SBS-DATA-004: Require Field History Tracking for Sensitive Fields

GDPR ISO 27001

Control Statement:
The organization must maintain a documented list of sensitive fields and ensure Field History Tracking is enabled for each listed field on all in-scope objects.

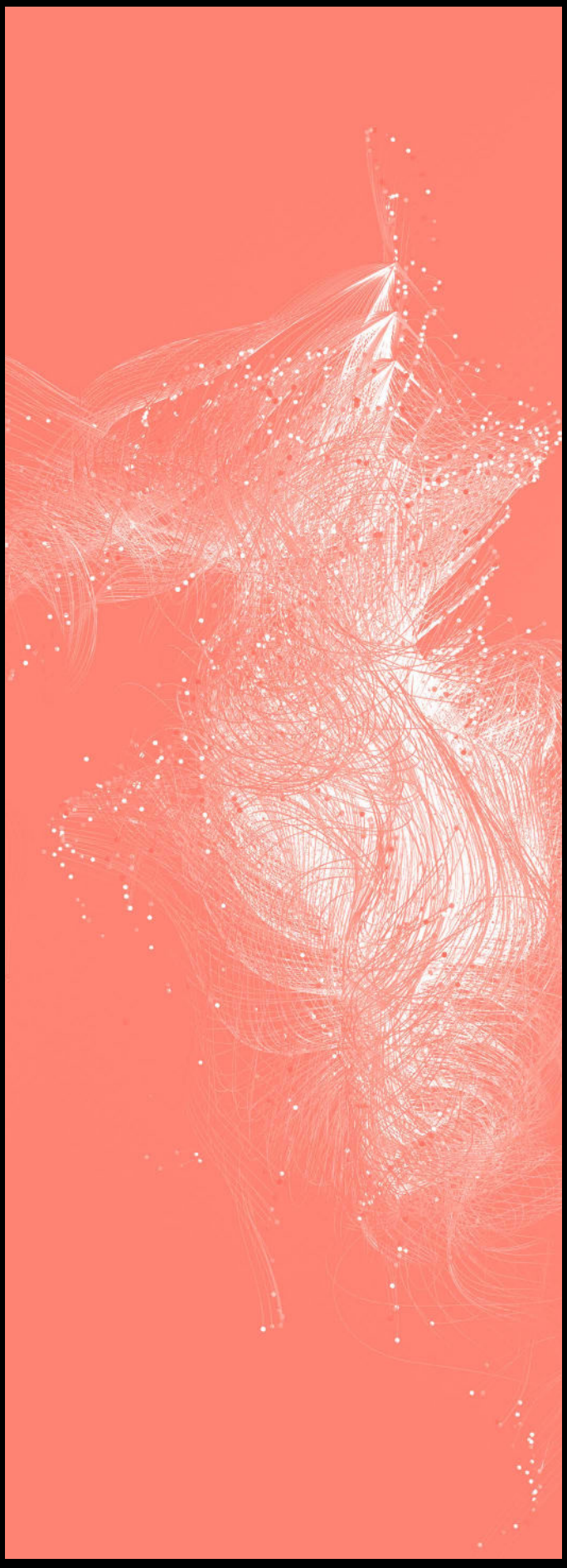
Description:
Organizations must define which fields are sensitive (e.g., regulated data, financial identifiers, or security-relevant attributes) and enable Field History Tracking for those fields so changes can be audited. Any sensitive field without tracking is noncompliant.

Risk: High
Without field history tracking on sensitive fields, unauthorized or accidental changes cannot be reliably detected or investigated. This reduces auditability, impairs incident response, and weakens accountability for changes to regulated or high-impact data.

Audit Procedure:
1. Obtain the organization's documented list of sensitive fields and in-scope objects.
2. Enumerate Field History Tracking settings for those objects.
3. Verify that each listed sensitive field has Field History Tracking enabled.
4. Flag any sensitive field without tracking.

Remediation:
1. Enable Field History Tracking for all listed sensitive fields.
2. Update the sensitive-field list as schemas evolve.
3. Re-verify tracking coverage after changes.

Default Value:
Salesforce does not enable Field History Tracking for sensitive fields by default.



09

DEPLOYMENTS

Deployments

This section defines controls related to metadata deployment practices, configuration change governance, and production environment integrity. These controls ensure that organizations establish clear provenance for production changes, restrict high-risk metadata modifications to controlled deployment processes, and maintain continuous monitoring to detect unauthorized configuration drift.

SBS-DEP-001: Require a Designated Deployment Identity for Metadata Changes

NIST SOC 2 ISO 27001

Control Statement:
Salesforce production orgs must designate a single deployment identity that is exclusively used for all metadata deployments and high-risk configuration changes performed through automated or scripted release processes.

Description:
A dedicated deployment identity (integration user) must be created and used as the sole account for CI/CD, metadata deployments, and automated release tooling. No human user—regardless of administrative privilege—may deploy metadata or execute automated deployment operations using their personal account.

Risk: High
Without a designated deployment identity, organizations cannot reliably attribute production changes—any administrator can deploy metadata, making it impossible to distinguish authorized CI/CD deployments from unauthorized manual changes. This loss of provenance prevents security teams from detecting unauthorized modifications, investigating configuration drift, or determining whether a change was part of an approved release. Attackers or malicious insiders can make direct production changes that blend into legitimate administrative activity, and incident responders cannot reconstruct the timeline of configuration changes during a breach investigation.

Audit Procedure:
1. Identify the user account designated as the deployment identity.
2. Enumerate all recent metadata deployments using tooling such as Deployment Status, Metadata API logs, or audit logs.
3. Verify that all deployments were executed by the designated deployment identity.
4. Flag any metadata deployment performed by a human user or non-deployment identity.

Remediation:
1. Create or identify a dedicated deployment identity.
2. Reconfigure CI/CD pipelines, release management tooling, and automated deployment scripts to authenticate exclusively with the deployment identity.
3. Revoke deployment permissions from all human users.
4. Re-deploy any metadata last deployed by a human user to restore provenance.

Default Value:
Salesforce does not create or enforce a dedicated deployment identity by default.

SBS-DEP-002: Establish and Maintain a List of High-Risk Metadata Types Prohibited from Direct Production Editing

NIST SOC 2 ISO 27001

Control Statement:
Salesforce production orgs must maintain an explicit list of high-risk metadata types that must never be edited directly in production by human users, defaulting at minimum to the SBS baseline list while allowing organizations to extend or refine it as needed.

Description:
Organizations must adopt the SBS baseline list of prohibited direct-in-production changes—which includes Apex Classes, Apex Triggers, LWCs, Aura Components, Profiles, Permission Set definitions, Remote Site Settings, Named Credentials, and core authentication or session security settings—and maintain this list as an internal policy. Organizations may extend this list or define exceptions, but the minimum baseline must be included and documented.

Risk: High
Without an explicit list of high-risk metadata types, organizations cannot define or enforce deployment governance boundaries—leaving critical configuration categories (Apex code, authentication settings, outbound connectivity, permissions) open to uncontrolled direct production editing. Security teams cannot distinguish between metadata that requires strict deployment controls and metadata that can be safely edited manually, resulting in inconsistent governance and gaps in change attribution. The absence of a defined list also prevents effective monitoring (SBS-DEP-003), as there is no baseline to compare against when detecting unauthorized changes.

Audit Procedure:

1. Obtain the organization's documented list of high-risk metadata types prohibited from direct production editing.

2. Confirm that the list, at minimum, includes all SBS baseline categories.

3. Review the list for any documented exceptions and verify they are formally approved.

4. Verify that only the deployment identity has modify permissions for metadata types on the list.

Remediation:

1. Adopt the SBS baseline list of prohibited direct-in-production metadata changes.

2. Add any organization-specific items or exceptions as needed.

3. Remove modify permissions for these metadata types from all human users.

4. Ensure all future changes to listed metadata types are performed exclusively by the deployment identity.

Default Value:
Salesforce does not provide native restrictions or guidance preventing direct production edits to high-risk metadata.

SBS-DEP-003: Monitor and Alert on Unauthorized Modifications to High-Risk Metadata

SOC 2 ISO 27001

Control Statement:
Salesforce production orgs must implement a monitoring capability that detects and reports any modification to high-risk metadata performed by a user other than the designated deployment identity.

Description:
Organizations must maintain a monitoring process—manual or automated—that reviews administrative and metadata changes and identifies when high-risk metadata (as defined in SBS-DEP-002 or extended by the organization) is modified by a human user instead of the designated deployment identity. The monitoring method may use Salesforce APIs, audit logs, export files, CLI tooling, vendor tools, or any combination, provided it reliably detects unauthorized changes within the organization's defined review interval.

Risk: High
Without monitoring for unauthorized metadata changes, organizations cannot detect when high-risk configuration is modified outside the approved deployment process—allowing malicious changes, accidental drift, or insider threats to persist undetected. Security teams lose the ability to identify unauthorized modifications to authentication settings, permission structures, Apex code, or outbound connectivity until a breach or incident reveals the gap. This impairs detection, investigation, and response capabilities for configuration-related security events, extending attacker dwell time and preventing timely remediation of unauthorized changes.

Audit Procedure:

1. Interview system owners to identify the monitoring method(s) used for detecting changes to high-risk metadata.

2. Review documentation describing how the monitoring process works—whether manual log review, automated scripts, API queries, CLI workflows, scheduled exports, or vendor tools.

3. Verify that the monitoring process includes:

   - Coverage of all high-risk metadata types defined by the organization and required by SBS-DEP-002.

   - A review interval appropriate to the organization's change-management expectations (e.g., daily, weekly, or aligned with release cycles).

   - A method for identifying the user who performed each change.

4. Examine historical monitoring records or logs to confirm the process has been performed consistently.

5. Flag noncompliance if no monitoring system exists or if the system cannot detect unauthorized human modifications to high-risk metadata.

Remediation:

1. Implement a monitoring mechanism capable of identifying modifications to high-risk metadata and attributing them to the responsible user. Acceptable approaches include:

   - Manual periodic review of the Salesforce Setup Audit Trail,

   - Exporting audit logs for review,

   - Scheduled API or CLI queries comparing metadata changes,

   - Custom scripts,

   - Vendor-based monitoring tools.

2. Ensure the monitoring method covers all high-risk metadata types listed in the organization's defined prohibited-direct-edit list.

3. Define a repeatable review interval and assign responsibility for conducting the review.

4. Document the monitoring approach and maintain records of reviews and findings.

Default Value:
Salesforce does not provide built-in monitoring or alerting for unauthorized direct-in-production metadata changes; organizations must implement their own processes.

SBS-DEP-004 — Establish Source-Driven Development Process

ISO 27001

Control Statement:
Meaningful Salesforce metadata changes must be deployed through a source-driven, automated, and deterministic deployment process, except where the platform does not provide programmatic deployment support.

Description:
Organizations must track all meaningful metadata changes in a centralized version control system and deploy them using an automated, repeatable, and deterministic process; manual changes in production are permitted only for metadata types that Salesforce does not expose for programmatic deployment.

Risk: High
Without a source-driven deployment process, organizations lose the verifiable audit trail that connects production configuration to approved changes—making it impossible to determine what changed, when, by whom, and whether it was authorized. Security teams cannot investigate configuration-related incidents, restore known-good state during outages, or attribute changes during forensic analysis. Manual production changes bypass code review, testing, and approval workflows, enabling unauthorized, accidental, or malicious modifications to security-sensitive settings without accountability or detection.

Audit Procedure:
1. Identify the organization's standard deployment process and designated deployment identity as defined in SBS-CHG-001.

2. Review recent production metadata changes and their associated deployment records.

3. Verify that changes deployable through Salesforce's programmatic deployment mechanisms originated from centralized version control.

4. Confirm that any manual production changes are limited to metadata types that Salesforce does not support for programmatic deployment.

5. Flag any manually applied changes that could have been deployed through the source-driven process.

Remediation:
1. Establish and maintain a centralized version control repository for Salesforce metadata.

2. Implement or enforce an automated deployment pipeline that deploys changes exclusively from version control.

3. Restrict direct production changes for metadata types that support programmatic deployment.

4. Document and periodically review any required manual production changes for metadata types lacking deployment support.

Default Value:
Salesforce allows direct manual changes to most metadata in production and does not require source control or automated deployments by default.

SBS-DEP-005: Implement Secret Scanning for Salesforce Source Repositories

ISO 27001

Control Statement:
Organizations using source-driven development for Salesforce must implement automated secret scanning on all repositories containing Salesforce metadata, configuration, or deployment scripts to detect and prevent the exposure of credentials, access tokens, and other sensitive authentication material.

Description:
CI/CD pipelines for Salesforce deployments typically require long-lived access tokens, refresh tokens, or other credentials to authenticate as the designated deployment identity. If these credentials are hardcoded in scripts, configuration files, or committed to version control, they become accessible to anyone with repository access—including contractors, consultants, former employees, or attackers who compromise the source control system. Organizations must implement automated secret scanning that runs on every commit and pull request to detect Salesforce-specific secrets (such as access tokens, refresh tokens, consumer secrets, and session IDs) as well as general credential patterns.

Risk: Critical
Exposed Salesforce credentials in source repositories represent a direct path to unauthorized production access—a supply chain attack vector that bypasses all other access controls. Contractors, consultants, or any party with repository access can extract hardcoded tokens and authenticate directly to production orgs with the full permissions of the deployment identity. Attackers who compromise source control systems or CI/CD infrastructure gain immediate access to production Salesforce environments. Unlike other credential exposures, Salesforce access tokens often have broad administrative permissions and long validity periods, making them high-value targets. Organizations cannot detect this exposure through Salesforce audit logs alone—the attacker authenticates with valid credentials, and their activity appears legitimate.

Audit Procedure:
1. Identify all repositories containing Salesforce metadata, SFDX projects, deployment scripts, or CI/CD pipeline configurations.

2. Verify that automated secret scanning is enabled on each repository—either through the source control platform's native capabilities (e.g., GitHub Secret Scanning, GitLab Secret Detection) or through third-party tooling.

3. Confirm that the scanning configuration includes patterns for Salesforce-specific secrets (access tokens, refresh tokens, consumer keys/secrets, session IDs).

4. Review scanning logs or dashboards to verify the tool is actively running and producing results.

5. Verify that detected secrets trigger alerts and block merges or deployments until remediated.

6. Flag noncompliance if any Salesforce-related repository lacks active secret scanning coverage.

Remediation:
1. Enable secret scanning on all repositories containing Salesforce code, metadata, or deployment configurations using platform-native tools or third-party secret scanning solutions.

2. Configure scanning rules to detect Salesforce-specific credential patterns in addition to general secrets.

3. Implement pre-commit hooks or CI checks that block commits containing detected secrets.

4. Immediately rotate any Salesforce access tokens, refresh tokens, or credentials that have been committed to version control—even if subsequently removed, as they persist in git history.

5. Migrate credential storage to secure secrets management solutions (e.g., CI/CD platform secrets, vault systems) and remove all hardcoded credentials from repositories.

6. Establish a periodic rotation schedule for Salesforce deployment credentials to limit the window of exposure if a secret is leaked.

Default Value:
Salesforce does not provide secret scanning capabilities; organizations must implement scanning through their source control platform or third-party tooling. Credentials can be freely committed to repositories without any native detection or prevention.

SBS-DEP-006: Configure Salesforce CLI Connected App with Token Expiration Policies

NIST SOC 2 ISO 27001

Control Statement:
Organizations must configure the Connected App used for Salesforce CLI authentication with refresh token expiration of 90 days or less and access token timeout of 15 minutes or less.

Description:
Salesforce CLI stores OAuth tokens locally on developer workstations to enable command-line access to orgs. The default "Salesforce CLI" Connected App ships with refresh tokens and access tokens set to never expire, meaning stolen or leaked token files provide indefinite access to authorized orgs. Organizations must either create a dedicated Connected App for CLI usage or install and configure the default Connected App with appropriate token expiration policies—refresh tokens must expire within 90 days and access tokens must timeout within 15 minutes. When using a dedicated Connected App, organizations should use the `--client-id` flag with CLI authentication commands.

Risk: High
Salesforce CLI token files stored on local workstations represent a persistent credential exposure risk. If a laptop is stolen, reassigned without proper cleanup, or compromised by malware, attackers can extract token files that provide direct access to Salesforce orgs—including production environments. With the default Connected App configuration, these tokens never expire, giving attackers indefinite access that persists even after the original user's password is changed or their account is deactivated. The attack surface expands with each org a developer authenticates to, as token files accumulate credentials to sandboxes, Dev Hubs, and production orgs. Organizations cannot detect this credential theft through Salesforce audit logs because the attacker authenticates with valid tokens.

Audit Procedure:

1. From Setup, navigate to Connected Apps OAuth Usage (or Apps → Connected Apps → Connected Apps OAuth Usage).
2. Identify the Connected App(s) used for Salesforce CLI authentication—either the default "Salesforce CLI" app or a custom Connected App.
3. Review the OAuth Policies for each CLI-related Connected App:
   - Verify that Refresh Token Policy is set to "Expire refresh token after" with a value of 90 days or less.
   - Verify that Session Policies Timeout Value is set to 15 minutes or less.
4. If a custom Connected App is used, verify that developers are instructed to use the `--client-id` flag when authenticating.
5. Flag noncompliance if any CLI-related Connected App has tokens set to never expire or exceeds the maximum allowed durations.

Remediation:

1. Determine whether to use the default "Salesforce CLI" Connected App or create a dedicated Connected App for CLI authentication.
2. If using the default app:
   - From Setup, navigate to Connected Apps OAuth Usage.
   - Locate "Salesforce CLI" and click Install (if not already installed), then Edit Policies.
   - Set Refresh Token Policy to "Expire refresh token after: 90 Days" (or less).
   - Set Session Policies Timeout Value to "15 minutes" (or less).
3. If creating a dedicated Connected App:
   - Create a new Connected App with OAuth enabled and appropriate callback URL.
   - Configure refresh token expiry to 90 days or less and access token timeout to 15 minutes or less.
   - Distribute the Consumer Key to developers and require use of `--client-id` flag.
4. Communicate to developers that they will need to re-authenticate periodically when refresh tokens expire.
5. Consider implementing compensating controls to protect locally stored token files, such as:
   - Requiring full disk encryption (FileVault, BitLocker) on developer workstations.
   - Enabling remote wipe capability for managed devices.
   - Including Salesforce CLI token file cleanup in device offboarding procedures.
   - Training developers to run `sf org logout --all` before returning or transferring devices.

Default Value:
The default "Salesforce CLI" Connected App is configured with refresh tokens and access tokens that never expire. Organizations must explicitly install and configure the app or create a dedicated Connected App to enforce token expiration policies.

10

SECURITY CONFIGURATION

Security Configuration

This section defines controls related to Salesforce platform security settings and configuration management. These controls ensure that organizations establish clear security baselines, continuously monitor configuration drift, and maintain intentional security postures through systematic review and remediation of platform settings.

SBS-SECCONF-001: Establish a Salesforce Health Check Baseline

ISO 27001

Control Statement:
Salesforce production orgs must define and maintain a Salesforce Health Check baseline—including Salesforce's native baseline XML or an equivalent customized baseline—and ensure it reflects the organization's intentional security configuration posture.

Description:
Organizations must create, upload, and maintain a Salesforce Health Check baseline template in XML format. The baseline must reflect the organization's required security configuration for key platform settings across authentication, session management, content security, and other Health Check parameters. Organizations may use Salesforce's default baseline, an SBS-recommended baseline, or a custom internal baseline, but the baseline must be explicitly defined, documented, and uploaded into Salesforce Health Check.

Risk: High
Without a defined Health Check baseline, organizations have no authoritative reference for what their security configuration should be—making it impossible to detect drift, evaluate deviations, or determine whether current settings reflect intentional decisions or accumulated neglect. Security teams cannot assess configuration-related risk, investigate whether settings were deliberately changed, or demonstrate compliance with security requirements. The absence of a baseline also prevents effective use of Health Check deviation monitoring (SBS-SECCONF-002), as there is no standard to measure against.

Audit Procedure:
1. Navigate to **Setup** → **Health Check** and confirm that a baseline template is uploaded and active.
2. Review the XML baseline directly (via UI or API) to verify that the baseline exists and contains intentional values rather than defaults left unexamined.
3. Interview administrators to confirm the baseline was deliberately chosen or customized and is understood as the organization's configuration standard.
4. If the organization lacks a baseline, flag the control as noncompliant.

Remediation:
1. Create or select a Health Check baseline (Salesforce default, SBS-recommended baseline, or a custom-defined XML).
2. Upload the baseline XML into **Setup** → **Health Check**.
3. Document ownership of the baseline and establish a process for periodic review and updates.
4. Communicate the baseline's purpose and implications to system owners and security stakeholders.

Default Value:
Salesforce provides a default baseline but does not require organizations to review, customize, or maintain it.

SBS-SECCONF-002: Review and Remediate Salesforce Health Check Deviations

ISO 27001

Control Statement:
Salesforce production orgs must periodically review Health Check results against the defined baseline and remediate deviations or formally document approved exceptions.

Description:
Organizations must maintain a repeatable process for reviewing deviations identified in Salesforce Health Check. The process may be manual or automated, and may use Salesforce's native UI, exported Health Check data, API-driven reports, or third-party tooling. The organization must remediate deviations that represent unapproved risk or document and track exceptions when deviations are intentional or operationally necessary.

Risk: High
Without periodic review and remediation of Health Check deviations, configuration drift accumulates undetected—weakening security posture over time as settings diverge from the intended baseline. Security teams cannot identify when critical platform settings (authentication, session management, content security) have been changed or misconfigured, preventing timely response to emerging vulnerabilities. Unaddressed deviations may persist indefinitely, creating exploitable gaps that remain invisible until a breach or audit reveals the exposure.

Audit Procedure:

1. Interview system owners to identify the established Health Check review process and review interval (e.g., monthly, quarterly).

2. Examine evidence of recent Health Check reviews, such as documented review artifacts, exported reports, tickets, changes, or exception records.

3. Verify that deviations were:

   - Remediated within the review window, **or**
   - Documented as exceptions with clear justification and approval.

4. Confirm that the review process is repeatable, assigned to an owner, and actually followed.

5. Flag noncompliance if:

   - No review process exists,
   - No review evidence can be produced, or
   - Deviations occur without remediation or exception documentation.

Remediation:

1. Establish a recurring review process using any reliable method, including:

   - Salesforce Health Check UI,
   - API exports,
   - CLI automation,
   - Scheduled scripts,
   - Vendor tooling.

2. Assign ownership for conducting the review and maintaining documentation.

3. Review current deviations and either remediate them or document exceptions.

4. Implement tracking to ensure deviations are remediated or re-reviewed in future cycles.

Default Value:
Salesforce does not require or track periodic Health Check reviews; deviations may persist indefinitely without administrative action.

11

FILE SECURITY

File Security

This section defines controls related to file and content security in Salesforce environments. These controls ensure that organizations maintain appropriate protections, governance, and lifecycle management over files, documents, and content shared within or outside the organization—reducing the risk of unauthorized access, data leakage, and exposure of sensitive information.

SBS-FILE-001: Require Expiry Dates on Public Content Links

GDPR CCPA/CPRA ISO 27001

Control Statement:
Organizations must ensure that Public Content links have an appropriate expiry date.

Description:
The organization must ensure that any content shared via Public Content links has an appropriate expiry date set dependent upon the classification of the content. The expiry date could be never for non-sensitive content—a PDF of the organization's Privacy Policy as an example—or it could be less than a week for sensitive information.

Risk: Moderate
Without an expiry date, Public Content links remain permanently accessible, extending the window of potential exposure indefinitely. While the link itself must be obtained by an unauthorized party for access to occur, perpetually valid links increase the cumulative risk of data exposure through link leakage, sharing, or discovery. Time-bounded links reduce the blast radius of any single link compromise and support data lifecycle governance.

Audit Procedure:
1. Enumerate all `ContentDistribution` object records via the SOAP/REST API or Apex.
2. Identify all records where `PreferencesExpires = false`.
3. Flag any Public Content links without expiry dates for review.

Remediation:
1. For each flagged content distribution record, determine the sensitivity classification of the associated content.
2. Set an appropriate expiry date on the `ContentDistribution` object based on content classification.
3. Establish organizational policy defining maximum link lifetimes by data classification.

Default Value:
When a user manually creates a Public Content link on a piece of content, Salesforce suggests an expiry date. This can be overridden by the user. In the past, the default was no expiry date.

SBS-FILE-002: Require Passwords on Public Content Links for Sensitive Content

HIPAA GDPR NIST CCPA/CPRA SOC 2 ISO 27001

Control Statement:
Organizations must ensure that Public Content links to sensitive content have a password.

Description:
The organization must ensure that any sensitive content shared via Public Content links has a password set to protect the content if the link is intercepted or inadvertently shared.

Risk: High
Without a password, anyone who obtains an unexpired Public Content link—through interception, accidental sharing, or link harvesting—can immediately access the associated data. For sensitive content, this creates a direct path to data exposure that requires only link acquisition. Password protection adds an authentication layer that prevents opportunistic access and limits the impact of link compromise, supporting breach containment and regulatory compliance for sensitive data handling.

Audit Procedure:
1. Enumerate all `ContentDistribution` object records via the SOAP/REST API or Apex.
2. Identify all records where `Password` is null.
3. Cross-reference with content classification to identify sensitive content lacking password protection.
4. Flag any Public Content links to sensitive content without passwords for review.

Remediation:

1. For each flagged content distribution record, determine the sensitivity classification of the associated content.

2. For sensitive content, set a password on the ContentDistribution record via the Salesforce UI.

3. Communicate the password to intended recipients through a separate, secure channel.

4. Establish organizational policy requiring password protection for all Public Content links to sensitive data.

Default Value:
When a user manually creates a Public Content link on a piece of content, the default is to not have a password.

SBS-FILE-003: Periodic Review and Cleanup of Public Content Links

GDPR CCPA/CPRA ISO 27001

Control Statement:
Organizations must implement a recurring process to review all active Public Content links and remove or remediate links that are no longer required, lack appropriate controls, or were created outside of current policy.

Description:
The organization must establish a defined cadence (e.g., quarterly) to scan all `ContentDistribution` records and review active Public Content links. This review should identify links that are forgotten, no longer needed, were created before current security controls were implemented, resulted from accidental sharing, or otherwise do not comply with organizational policy. Identified links must be remediated by applying appropriate controls (expiry dates, passwords) or deleted if no longer required.

Risk: **Moderate**
Without periodic review, Public Content links accumulate over time—including legacy links created before security policies were established, links that have outlived their business purpose, and links created through accidental or unauthorized sharing. These forgotten links represent persistent exposure that may go undetected indefinitely. While this control does not prevent initial link creation issues, it provides a governance mechanism to identify and remediate accumulated risk, supporting defense-in-depth and reducing the organization's overall exposure footprint.

Audit Procedure:

1. Verify the organization has a documented process for periodic Public Content link review.

2. Confirm the review cadence is defined (e.g., quarterly, monthly) and appropriate for the organization's risk profile.

3. Obtain evidence of recent review execution (e.g., scan results, remediation records, review meeting notes).

4. Verify that reviews include all active `ContentDistribution` records.

5. Confirm that identified issues are tracked through remediation or deletion.

6. Flag organizations without a documented review process or evidence of recent execution.

Remediation:

1. Establish a documented process for periodic review of all ContentDistribution records.

2. Define a review cadence appropriate to organizational risk tolerance (quarterly recommended as a baseline).

3. Create a scanning mechanism (script, report, or tool) to enumerate all active Public Content links.

4. Define review criteria to identify links requiring remediation: missing expiry dates, missing passwords on sensitive content, links older than a defined threshold, or links to content no longer requiring external sharing.

5. Assign ownership for the review process and remediation actions.

6. Maintain records of each review cycle for audit purposes.

Default Value:
Salesforce does not provide a built-in mechanism for periodic review of Public Content links; organizations must implement this process manually or through custom tooling.

12

SALESFORCE LOGGING ARCHITECTURE

Salesforce provides multiple categories of monitoring and logging data with different levels of detail and retention. These logging mechanisms fall into three primary categories.

1. Built-in Audit Logs (Available in All Salesforce Orgs)

Every Salesforce organization includes several standard audit data sources that record authentication activity and administrative changes. These logs are automatically generated and cannot be disabled.

Examples include:

- **Login History (`LoginHistory`)** – records login attempts, including source IP address, login status, and authentication method.
- **Setup Audit Trail (`SetupAuditTrail`)** – records administrative configuration changes within the environment.
- **Field History Tracking** – optional feature that records historical field value changes for selected objects.
- **API Usage Metrics** – aggregate counts of API usage available through the Salesforce limits APIs and administrative dashboards.

These logs provide baseline visibility but typically do not contain sufficient detail for advanced security monitoring or forensic investigations.

2. Event Monitoring Logs (EventLogFile Dataset)

Salesforce provides enhanced activity logging through **Event Monitoring**, which generates detailed event logs stored in the `EventLogFile` dataset.

A limited set of Event Monitoring logs is available **by default** in Enterprise, Unlimited, and Performance editions with **1-day retention**. These include:

- Login
- Logout
- API Total Usage
- Apex Unexpected Exception
- CORS Violation
- CSP Violation
- Hostname Redirect events

The **Event Monitoring add-on** expands this capability by providing dozens of additional event types and extending retention up to **one year**, depending on licensing.

Examples of additional event types include:

- API request logs
- Report access
- Dashboard usage
- File downloads
- Lightning page navigation
- Apex execution
- URI activity

These logs provide detailed telemetry required for advanced security monitoring, anomaly detection, and forensic investigation.

3. Real-Time Event Streams (Streaming Events)

Salesforce also exposes certain monitoring events as **real-time event streams** through the Streaming API and Event Bus.

Examples include:

- API anomaly detection events
- credential stuffing detection events
- file activity events
- Lightning URI activity

These streaming events allow external monitoring systems or SIEM platforms to receive near real-time security telemetry. They are separate from the historical EventLogFile dataset and are primarily used for active threat detection rather than retrospective analysis.

SBS-MON-001: Enable Event Monitoring Log Storage

GDPR ISO 27001

Control Statement:
Organizations using Salesforce Event Monitoring must ensure that storage of required Event Monitoring logs is enabled for all event types necessary to support the organization's security monitoring and compliance policies.

Description:
Salesforce Event Monitoring generates detailed activity logs in the EventLogFile dataset. These logs provide visibility into user behavior, API activity, report usage, file access, and other system interactions.

Salesforce automatically generates many event logs; however, organizations must ensure that **storage is enabled for the event types they intend to retain and analyze**.

Event Monitoring log storage determines whether Salesforce retains generated event logs so they can be accessed through the platform or exported to external monitoring systems.

Examples of Event Monitoring log types include:

- API activity logs
- API Total Usage logs
- Report access logs
- Dashboard access logs
- URI navigation logs
- Lightning page view logs
- Apex execution logs
- File download logs

If storage is not enabled for a given event type, Salesforce may not retain the corresponding event data, preventing organizations from analyzing that activity.

Organizations should ensure storage is enabled for all event types necessary to support security monitoring, incident response, and compliance requirements.

Risk: High
Failure to ensure Event Monitoring log storage is enabled creates a significant monitoring gap.

Without these logs:

- Security teams cannot reconstruct user activity during a suspected breach.
- Data exfiltration events (such as large file downloads or abnormal API usage) may go undetected.
- Detailed forensic investigations become impossible due to lack of historical telemetry.
- Organizations may fail to meet regulatory or internal security logging requirements.

Because Salesforce logs cannot be retroactively generated, failing to retain required event logs results in permanent loss of security telemetry.

Audit Procedure:
1. Navigate to **Setup**
2. Open **Event Monitoring Settings**
3. Verify that **Generate Event Log Files** is enabled
4. Open **Event Manager**
5. Review each event type and verify that **Enable Storage** is enabled for all log types required by the organization's security monitoring policy

Remediation:
1. Navigate to **Setup**
2. Open **Event Monitoring Setting**s
3. Enable **Generate Event Log Files**
4. Open **Event Manager**
5. For each required event type:
   - Select the event
   - Enable **Storage**

Organizations should also implement automated export or integration with external monitoring platforms where appropriate.

Default Value:
Salesforce provides a limited set of Event Monitoring logs by default in Enterprise, Unlimited, and Performance editions. These logs include Login, Logout, and API Total Usage events with **1-day retention**.

Additional event types and extended retention (up to one year) require **the Event Monitoring add-on**.

SBS-MON-002: Retaining Event Logs

GDPR ISO 27001

Control Statement:
Organizations must retain security event logs for the defined retention period and implement measures to protect the logs from tampering and unauthorized deletion to ensure forensic availability.

Description:
The retention of security event logs is critical for effective incident investigation and forensic analysis. Security incidents often remain undetected for extended periods of time, and investigators must be able to reconstruct historical activity to determine the origin, scope, and impact of an attack.

Salesforce provides multiple categories of logs with different native retention periods.

Built-in audit logs include:

- **LoginHistory** – records login attempts and authentication activity. Retention is up to approximately 6 months, though high-volume environments may experience shorter effective retention.
- **SetupAuditTrail** – records administrative configuration changes. Retention is 180 days.

These built-in logs provide baseline visibility but typically lack the detail required for advanced security investigations.

Organizations requiring detailed user activity telemetry must use **Event Monitoring**, which provides the `EventLogFile` dataset. These logs include detailed activity such as:

- API activity
- report access
- dashboard usage
- file downloads
- Lightning page navigation
- Apex execution
- URI activity

Retention of Event Monitoring logs depends on licensing:

- **Without the Event Monitoring add-on**: Salesforce provides a limited set of Event Monitoring logs (including Login, Logout, and API Total Usage) with **1-day retention**.
- **With the Event Monitoring add-on**: many additional event types become available and retention is extended, typically ranging from **30 days up to one year**, depending on the license.

Because Salesforce automatically purges logs once their retention period expires, organizations with longer retention requirements must export these logs to an external system.

Common approaches include forwarding logs to:

- a Security Information and Event Management (SIEM) platform
- centralized log aggregation systems
- secure long-term archival storage

External retention ensures that forensic data remains available even after Salesforce purges the logs from the platform.

Salesforce also provides administrative capabilities that allow authorized users to delete stored Event Monitoring data. To reduce the risk of log tampering, organizations must strictly control and monitor permissions related to Event Monitoring data deletion.

Changes to the **"Delete Event Monitoring Data"** permission and related configuration settings should be monitored through the `SetupAuditTrail` log.

Risk: High
Failure to implement adequate log retention and protection creates significant risks for incident response and regulatory compliance.

If logs are not retained for a sufficient period:

- Security teams may be unable to reconstruct the timeline of an attack.
- Indicators of compromise may be permanently lost before detection occurs.
- The organization may be unable to determine the scope of data exposure.

In addition, attackers who obtain administrative privileges may attempt to delete stored event logs to conceal malicious activity. Without secure external copies stored in a separate system, this could permanently destroy critical forensic evidence.

Because Salesforce logs cannot be retroactively generated, failure to retain logs results in permanent loss of investigative data.

Audit Procedure:

1. Obtain the organization's security policy defining required log retention periods.

2. Review the Salesforce organization's licensing and Event Monitoring configuration to determine the native retention period of logs.

3. Verify whether Event Monitoring logs are exported to an external system such as a SIEM or centralized logging platform.

4. Confirm that the combined retention period (Salesforce native retention plus external storage retention) meets or exceeds the required organizational or regulatory retention policy.

5. Review the SetupAuditTrail log to confirm that changes to Event Monitoring configuration and permissions are monitored, including assignment of the **Delete Event Monitoring Data** permission.

Remediation:

1. If longer retention is required, purchase Salesforce Event Monitoring or export logs to an external secure storage system.

2. Implement automated log export or forwarding to a SIEM or centralized monitoring platform.

3. Restrict the **Delete Event Monitoring Data** permission to a minimal set of trusted administrators.

4. Configure monitoring alerts to detect any assignment of this permission or modification of Event Monitoring settings.

5. Periodically verify that log exports are functioning correctly and that required log types are being successfully captured.

Default Value:
Typical native Salesforce retention periods include:

- **LoginHistory**: up to approximately 6 months (may be shorter in high-volume environments)
- **SetupAuditTrail**: 180 days
- **Event Monitoring logs (`EventLogFile` dataset)**: a limited set of logs is available by default with **1-day retention**; extended retention and additional log types require the Event Monitoring add-on

SBS-MON-003: Monitor for Suspicious Logins

GDPR NIST CCPA/CPRA SOC 2 ISO 27001

Control Statement:
Organizations must continuously monitor and alert on anomalous login patterns to promptly detect and mitigate compromised accounts and application credentials.

Description:
Continuously monitoring user and application login activity is paramount for maintaining the security of a Salesforce environment, especially for Orgs containing sensitive data. The primary objective is the early detection of compromised credentials, which are often the initial vector for sophisticated data breaches. Effective monitoring requires organizations to move beyond simple success/failure log tracking and implement sophisticated anomaly detection techniques.

Organizations must establish a baseline of normal login behavior for all users and connected applications. Any deviation from this baseline should trigger an alert for investigation. Key anomalous patterns to continuously monitor include:

1. Geographic and Travel Anomalies: Logins originating from an unexpected or uncommon geolocation, particularly from high-risk regions. A high-severity alert should be generated for "impossible travel" scenarios, where a user's successive login attempts originate from two distant locations within a time frame that makes physical travel unfeasible.

2. Suspicious Network Origins: Logins from IP addresses associated with known malicious or anonymity-enabling networks. This includes TOR exit nodes, commercial/consumer VPNs, and proxies, which attackers use to mask their true location and identity.

3. Time-based Anomalies: Logins occurring outside of a user's typical working hours (e.g. in the middle of the night) or during a period when the user is known to be on vacation.

4. High-Volume Failed Logins: A large number of failed login attempts followed by a successful one (a common pattern for brute-force or credential stuffing attacks), particularly if the successful login originates from a previously unseen IP address.

5. Client and User-Agent Changes: Unexplained shifts in the login application (e.g. logging in via a mobile app when only web access is typical) or a suspicious, malformed, or blacklisted user-agent string.

Due to the volume and complexity of login data, this monitoring is typically performed by processing Salesforce `LoginHistory` and relevant Event Monitoring logs in a dedicated monitoring system.

Risk: High
Failure to continuously monitor and rapidly respond to suspicious login activity creates an immediate and severe risk of a full system compromise. This control's absence exposes the organization to three critical, cascading risks:

1. The primary risk is that a successful suspicious login is the initial foothold for a sophisticated attack. Without automated, anomaly-based alerting, the attacker's presence goes unnoticed, leading to a prolonged "dwell time" (the period between compromise and detection). A longer dwell time allows the attacker to extensively reconnoiter the environment, escalate privileges, and establish persistence. The longer the intruders have to prepare, the harder it is to evict them from the Org.

2. Compromised credentials grant attackers direct access to sensitive customer data, intellectual property, and internal records. The lack of timely detection enables the attacker to perform mass data exfiltration via API, reporting, or export functionality before the account can be disabled, leading to financial, regulatory, and reputational damage.

3. An attacker with elevated access can modify or delete critical configuration settings, business-critical data, or code, leading to system downtime, operational failures, and a loss of data integrity.

Audit Procedure:
1. Verify the organization utilizes a continuous analytics solution that looks for anomalies in Salesforce login data.

2. Confirm that the analytics solution's monitoring scope includes all user logins and all non-human integration or application user logins, regardless of whether the authentication method is Single Sign-On (SSO) or direct Salesforce login.

3. Confirm that there is a documented, mandatory procedure to periodically (e.g. quarterly) review, test, and update the login anomaly detection rules and baselines to ensure they remain effective and minimize false positives. Review should also include an analysis of investigated suspicious login alerts during the review period – no investigations can mean the rules are not firing.

4. Review the process and history of investigating and responding to suspicious login alerts.

Remediation:
1. Take into use a dedicated Security Information and Event Management (SIEM) or a specialized login anomaly detection platform. If a platform is already in place, ensure all Salesforce authentication and event logs, particularly LoginHistory and relevant Event Monitoring logs, are being successfully exported, ingested, and correctly parsed.
2. Configure the monitoring solution to explicitly analyze login events from all sources, including:
   - Human Users: Both Single Sign-On (SSO) and direct Salesforce credentials.
   - Non-Human/Integration Accounts: Any connected systems, APIs, or automated processes.
3. Periodically review and test the effectiveness of the deployed detection rules, minimizing false positives and updating baselines as organizational travel patterns or integration accounts change.
4. Define clear, high-priority, and time-bound Standard Operating Procedures (SOPs) to investigate and respond to all generated suspicious login alerts.

Default Value:
Salesforce natively provides the raw login data in the LoginHistory object. However, without additional paid features or products or custom in-house development, Salesforce does not automatically detect or alert on the sophisticated, anomalous login patterns described in this control. Detection is entirely manual, relying on an administrator running reports against the raw LoginHistory data.

SBS-MON-004: Monitor for Suspicious API Activity

GDPR NIST CCPA/CPRA SOC 2 ISO 27001

Control Statement:
Organizations must continuously monitor and alert on all API activity to establish a baseline, detect anomalous and malicious activity, and identify potential application and integration abuse in a timely manner.

Description:
Monitoring of API activity is a critical security layer that is necessary even after robust login monitoring is in place. Login monitoring alone is insufficient because many advanced threats — including compromised services, session hijacking, stolen access tokens, and malicious insider activity — can bypass the initial authentication step. To effectively detect these post-authentication attacks, organizations must continuously analyze access patterns against established baselines of normal behavior.

This requires ingesting and analyzing granular API event logs, which are primarily available through the Salesforce Event Monitoring add-on. Key anomalous patterns that must be continuously monitored, baselined, and alerted upon include:

1. Unexpected Data Access/Objects: An application or user suddenly querying, modifying, or deleting objects they have never interacted with before (e.g. a Sales user starting to query security configurations or an integration account exporting massive volumes of a sensitive object like Case or Account).

2. Anomalous Volume of API Calls: Any sharp, unexplained spike or drop in API request volume from a specific user or application, can be an indicator of mass data exfiltration or other malicious activity.

3. Uncharacteristic Action Types: A pattern shift from read-heavy operations to write/delete-heavy operations, such as a reporting application suddenly creating or deleting users, which points to application abuse or compromise.

4. Suspicious Network Origins (Post-Login): API activity originating from an IP address or geolocation that is unusual for the user or application, especially when the access token may have been stolen and is being used from a suspicious location.

5. Use of Unapproved/Deprecated API Versions: Calls being made using API endpoints or versions that the organization has explicitly phased out or restricted for security reasons.

6. Unexpected applications: Application use by users should be monitored for anomalies. It should be noted that the attackers can use common applications and try to hide in plain sight in the logs. They can also use “evil twin” attacks pretending to be a service used by the company but in reality using a different account or even a completely fake tool.

Risk: High
Failure to monitor API activity creates a critical blind spot for post-authentication attacks, leading to severe and cascading consequences:

1. The primary risk is that a compromised application, integration account, or stolen session token can be used to perform mass, high-speed data exfiltration.

2. Attackers can leverage API access to perform unauthorized and undetectable manipulation of the Salesforce Org.

Audit Procedure:

1. Verify the organization utilizes a continuous analytics solution (e.g. SIEM, log aggregation platform) that is verifiably integrated with and ingesting all required API activity data, specifically the detailed Event Monitoring logs from Salesforce.

2. Confirm that the monitoring scope includes all API access, including user-initiated, non-human/integration accounts, and all API methods (e.g. REST, SOAP, Bulk). Verify the logs ingested are granular enough to track specific API calls (CRUD operations) against specific objects.

3. Request evidence (e.g. SIEM rule configurations, simulated alert outputs) to verify that the following specific, high-severity anomaly detection rules are active and tuned:

   - Mass Data Exfiltration: Alerts on an anomalous spike in data retrieval volume for a specific user or application, especially for sensitive objects.

   - Unauthorized Object Access: Alerts on a user or integration account attempting CRUD operations on objects they have no historical precedent for interacting with (e.g. Sales user accessing Security Settings).

   - Suspicious Action Shift: Alerts on a shift from a baseline of read-only operations to a high volume of write or delete operations by a user or application.

   - Suspicious Network Origins: Alerts on API calls originating from a suspicious or high-risk geolocation/IP address that is unusual for the accessing entity.

4. Process Review - Triage & Response: Examine the documented procedures for triage, investigation, and response to API anomaly alerts. Review a sample of anomaly findings and the actions taken on them for e.g. the past 6 months, to confirm that alerts are being acted upon within the defined parameter.

Remediation:

1. Deploy or build a dedicated analytics solution for granular API log analysis. Ensure all relevant Salesforce Event Monitoring logs for API activity are being successfully exported, ingested, and correctly parsed into the monitoring platform.

2. Configure the analytics solution to first establish a baseline of normal API behavior for every user and integration account. Following the baseline period, configure and tune high-severity anomaly detection rules to detect deviations, specifically rules that:

   - Alert on an anomalous spike in data retrieval (read/query) volume.
   - Monitor for CRUD (Create, Read, Update, Delete) operations on objects or security settings outside of an entity's established historical interaction baseline.
   - Flag a change from read-heavy activity to an unusual volume of write or delete operations.
   - Alert on API calls originating from a suspicious or atypical geolocation/IP address for the accessing entity.

3. Define clear, high-priority, and time-bound Standard Operating Procedures (SOPs) to investigate and respond to all generated API anomaly alerts. The procedure must include immediate steps to lock or deactivate the compromised user or application account and revoke stolen access tokens to prevent further data loss or system manipulation.

4. Establish a mandatory, recurring (e.g. quarterly) review process to:

   - Periodically test the effectiveness of the deployed detection rules using simulated incidents (e.g. table-top exercises).
   - Review and update the established baselines as new integrations are deployed or organizational usage patterns change.

Default Value:
Salesforce provides some raw API usage data (APITotalUsage), but it does not automatically detect, baseline, or alert on sophisticated API anomalies. Automated anomaly detection requires paid add-ons (like Event Monitoring), third-party solutions, or custom in-house development.

SBS-MON-005: Monitor API Usage Against Limits

ISO 27001

Control Statement:
Organizations must implement continuous, real-time monitoring and alerting on current API usage against defined Salesforce limits to proactively prevent service disruptions.

Description:
Monitoring API usage against defined organizational and Salesforce-imposed limits is a critical operational security measure to ensure service availability. These limits, which operate on a rolling 24-hour window and represent a total, shared quota across all users and integrated applications in the Org, are designed to ensure system stability and fair resource allocation.

When the organizational API call quota is exceeded, the Salesforce platform begins to block all further inbound API requests until the rolling 24-hour count drops below the threshold. This results in immediate and severe business process disruptions, as core integrations are effectively disabled. Examples of process failures include:

- Inability to sync critical data (e.g. financial or delivery information) between ERP/external systems and Salesforce.
- Interruption of vital business processes (e.g. orders or customer service tickets failing to be created or updated).
- Failure of marketing automation or data warehousing tools to ingest or extract necessary information.

Limit breaches can occur for two primary reasons:

1. Gradual Overconsumption: Poor architectural design or uncontrolled integration growth, where the cumulative API usage of all systems slowly exceeds the Org's capacity.
2. Sudden Spike: A sudden, catastrophic spike in calls caused by a misconfigured internal or third-party application e.g. getting stuck in a loop.

To proactively mitigate these risks, organizations must implement a continuous, real-time monitoring solution. This solution must track current API consumption against the official limits (which are tied to the Salesforce Edition and user license count) and provide immediate alerts before the critical threshold is breached. Limits and current usage can be reviewed in the "System Overview" section under Setup. It is also important to monitor sub-quotas for specific services like Bulk API, Streaming API, and Platform events.

When the API limit is at risk (e.g. above a defined threshold, for example, 80% or 90% utilization), the organization must execute a defined, high-priority incident response plan. This plan must first focus on immediate triage to identify the source of the consumption spike — whether it is a malicious actor, a runaway integration, or unexpected growth. While root cause is still under investigation immediate mitigation steps include:

1. Traffic Reduction: Temporarily reducing API calls by disabling non-critical integrations or their respective system users to conserve remaining quota.

2. Quota Increase: Proactively contacting Salesforce to purchase additional, temporary or permanent API quota if architectural changes are not immediately feasible.

Risk: **Moderate**
Failure to stay within API limits creates an immediate and severe risk to the availability of critical business operations. Exceeding the rolling 24-hour API quota blocks all further inbound requests, which effectively disables core integrations (e.g. with ERP), leading to catastrophic failures of vital business processes like order placement and resulting in direct financial loss.

Audit Procedure:

1. Review the organization's daily API limit and current 24-hour usage in the Salesforce System Overview in Setup.

2. Verify that a continuous monitoring solution (at minimum Salesforce's native API Usage Notifications) is implemented and active.

3. Confirm that the monitoring solution is configured to trigger immediate, high-priority alerts at a proactive threshold (e.g. 80% or 90% utilization) before the hard limit is breached. Request evidence (e.g. rule configurations, test alerts) to support this.

4. Examine the formal, documented incident response plan for API limit breaches. Verify the plan clearly defines:

   - The immediate triage steps to identify the runaway process or application causing the spike.

   - The mandatory, temporary mitigation steps (e.g. disabling non-critical integrations).

   - The process for escalating and requesting a temporary or permanent API quota increase from Salesforce.

5. Review the history (e.g. for the past 12 months) of any API limit breach or near-breach incidents. Confirm that the response procedure was followed and was effective in mitigating the service disruption.

Remediation:

1. Configure Salesforce's native "API Usage Notifications" in Setup and, more critically, integrate API consumption data into an external monitoring solution. Configure high-priority alerts to trigger at a proactive utilization threshold (e.g. 80-90%) before the hard limit is reached.

2. Establish a formal, high-priority Standard Operating Procedure (SOP) to be executed upon a high-utilization alert. This SOP must clearly define and be rehearsed to:

   - Immediately identify the user, application, or integration responsible for the spike in API consumption.

   - Mandate the immediate, temporary disabling of non-critical, high-volume integrations to conserve remaining quota.

   - Outline the process for proactively requesting a temporary or permanent API quota increase from Salesforce.

3. Following any API limit breach or near-breach, mandate a post-incident review to analyze the root cause. The outcome should be a plan to re-engineer high-volume integrations for more efficient usage (e.g. using Bulk API or other low-volume methods) to prevent recurrence.

4. Continuous Monitoring Maintenance: Establish a mandatory, recurring (e.g. quarterly) review process to confirm the ongoing integrity and functionality of the API usage data export to the monitoring platform.

Default Value:
Salesforce provides native API limit and usage data, including basic API Usage Notifications. However, it does not automatically implement the proactive, high-priority alerting (e.g. at 80-90% utilization) required to prevent service disruption. Effective control requires manual configuration of native features and/or integration with a monitoring solution.

13

CONCLUSION

Conclusion: Engineering for Inevitability

Traditional security models often focus exclusively on prevention. However, as modern attack patterns have shown, even the most robust perimeters can be bypassed through overprivileged identities and ungoverned integrations.

The SBS framework is built on the principle of **assumed compromise**. By enforcing documented permission models, restricting non-human identities, and ensuring durable event logging, the benchmark ensures that when an incident occurs, its "blast radius" is contained and its footprints are visible. Implementing SBS is an investment in forensic readiness and operational resilience. In an era where Salesforce houses an organization's most sensitive intellectual property and customer data, an objective, independent benchmark is no longer optional—it is a fundamental requirement for the modern enterprise.

14

AUTHORS

Authors

The Security Benchmark for Salesforce is developed by the following authors. Editorial direction is centralized under a Chief Editor to ensure consistency, coherence, and long-term integrity of the benchmark.

Participation as an author is in an individual, professional capacity. The Security Benchmark for Salesforce is an independent initiative and is not owned, controlled, sponsored, or endorsed by the employers of its authors. The views expressed reflect the professional judgment of the individual authors and not those of their respective organizations.

Pablo Gonzalez
Director of Product, AutoRABIT
Chief Editor. Product management and security architecture experience in DevSecOps and Salesforce platform security.

Jannis Schreiber
Technical Architect, The Mobility House
IT management and software engineering experience with focus on enterprise Salesforce implementations and sustainable technology solutions.

Jakub Stefaniak
Salesforce CTA & CTO, Aquiva Labs
Technical architecture and executive leadership experience with expertise in AppExchange product development and platform security.

Anderson Anthony
Salesforce Security Consultant
Salesforce security administration experience specializing in governance, risk management, and compliance for regulated environments.

Justin Hazard
CISO, AutoRABIT
15+ years of information security experience from SOC operations to executive security leadership, with focus on building security teams and developing security professionals.

John Crimmings
Senior Principle, Slalom
Seasoned Salesforce architect with deep expertise in enterprise security design, data protection strategies, and building secure, scalable platform solutions.

Scott Covert
Founder, Tython
14+ years of Salesforce platform development experience, specializing in cloud computing solutions, application development, and product leadership on the Force.com platform.

Lawrence Newcombe
Salesforce CTA, GiveClarity
Technical and solution architecture experience specializing in nonprofit sector transformation programs, AppExchange product development, and complex security implementations including identity and integration.

Salah Mansour Akridiss
Salesforce Architect,
Banque Internationale à Luxembourg
Salesforce technical architecture experience with focus on platform security, enterprise implementations, and secure solution design for regulated financial services.

Doug Merrett
Founder, Platinum7
13 years of experience at Salesforce with deep expertise in platform security, compliance, and resilience across enterprise environments.

Mika Ståhlberg
Co-founder & CTO, Valo
20+ years of cyber security and software engineering leadership, with expertise spanning AI-driven cloud solutions, low-level forensics, security governance, and Salesforce integration architecture.