
TSP
Accelerator
Transaction Security Policies, Made Easy

Table of Contents

⚠️ Important: Testing and Deployment Notice

Before enabling any Transaction Security Policy (TSP) provided in the TSP Accelerator managed package within a production environment, it is critical to conduct comprehensive testing in a non-production environment, such as a Sandbox or Scratch Org. Because these policies evaluate events in real-time–including logins, API queries, and report exports –improperly configured logic or missing metadata allowlists can result in immediate and unintended service disruptions or "lock-out" scenarios for your users. Always validate that policy triggers and configuration dependencies align with your specific organizational workflows and security requirements before final activation.

Background and Repository

This guide serves as a comprehensive technical reference for the security policies within TSP Accelerator. It details the underlying logic, event triggers, and configuration requirements for each policy, organized by their respective Security Lenses.

Additionally, this open source repository holds all of the actual Apex code for the security policies within TSP Accelerator. Use this repository to gain a deeper understanding of the logic used in each policy, or as a starting point for customizing your own policies to fit your specific needs.

Data Protection & Classification

Detect High Risk Data in API Queries

- Event: ApiEvent
- What it does: Detects API calls where the query includes fields classified as 'High-Risk' through Salesforce Data Classification.
- Logic: Returns true if any fields in event.Query are categorized with a Data Sensitivity Level marked as High-Risk.
- Configuration dependencies:
    - The appropriate Data Sensitivity Levels should be marked as a High-Risk Level
    - The appropriate fields on the appropriate objects should be categorized with the appropriate Data Sensitivity Levels
    - Salesforce Help for these items

Detect High Risk Data in Bulk API Result Downloads

- Event: BulkApiResultEventStore
- What it does: Detects when the results of the Bulk API call are downloaded where the query includes fields classified as 'High-Risk' through Salesforce Data Classification.
- Logic: * Logic: Returns true if any fields in event.Query are categorized with a Data Sensitivity Level marked as High-Risk. Every entity the query reads is evaluated: the top-level FROM clause plus the FROM clause of each subquery, with child relationship names resolved to their underlying object. Each entity's High-Risk fields are matched against the full text of the query, so a field name shared across multiple entities errs toward detection.
- Configuration dependencies:
    - The appropriate Data Sensitivity Levels should be marked as a High-Risk Level
    - The appropriate fields on the appropriate objects should be categorized with the appropriate Data Sensitivity Levels
    - Salesforce Help for these items

Detect High Risk Data in List View

- Event: ListViewEvent
- What it does: Detects when users access data with list views where the column headers include fields classified as 'High-Risk' through Salesforce Data Classification.
- Logic: Returns true if any fields in event.ColumnHeaders are categorized with a Data Sensitivity Level marked as High-Risk.
- Configuration dependencies:
    - The appropriate Data Sensitivity Levels should be marked as a High-Risk Level
    - The appropriate fields on the appropriate objects should be categorized with the appropriate Data Sensitivity Levels
    - Salesforce Help for these items

Detect High Risk Data in Report Export

- Event: ReportEvent
- What it does: Detects when reports are exported actions where the column headers include fields classified as 'High-Risk' through Salesforce Data Classification.
- Logic: Returns true if any fields in event.ColumnHeaders are categorized with a Data Sensitivity Level marked as High-Risk and event.Operation begins with 'ReportExported'. This covers all three export operations: ReportExported, ReportExportedAsynchronously and ReportExportedUsingExcelConnector.
- Configuration dependencies:
    - The appropriate Data Sensitivity Levels should be marked as a High-Risk Level
    - The appropriate fields on the appropriate objects should be categorized with the appropriate Data Sensitivity Levels
    - Salesforce Help for these items

Access Control (Authentication)

Detect Impossible Travel Scenarios

- Event: LoginEvent
- What it does: Detects when a user logs in from a different city or country within a 30-minute window compared to their most recent prior login.
- Logic: Looks up the user's most recent prior login in LoginHistory, then returns true when the time difference is < 30 minutes and either event.City or event.CountryIso has changed.
- Configuration dependencies:
    - n/a

Detect Login As Events

- Event: LoginAsEvent
- What it does: Detects when an administrator uses the "Login As" feature to access Salesforce as another user.
- Logic: Returns true for every LoginAsEvent received (no additional filtering is implemented in the Apex condition).
- Configuration dependencies:
    - n/a

Detect Logins Based on TLS Cipher Suite

- Event: LoginEvent
- What it does: Detects logins negotiated with a TLS cipher suite that is not in the approved allowlist.
- Logic: Returns true if event.CipherSuite is populated and is not found in the allowlist stored in Allowed_Cipher_Suite__mdt.Cipher_Suite__c.
- Configuration dependencies:
    - Add custom metadata records to the Allowed_Cipher_Suite__mdt to maintain an allowlist of approved TLS cipher suites.
        - Salesforce Help for adding custom metadata records

Detect Logins from Countries Not on Allowed List

- Event: LoginEvent
- What it does: Detects logins where the login country is not in the allowlist of countries.
- Logic: Returns true if event.CountryIso is not found in the allowlist stored in Approved_Login_Country__mdt.Alpha_2_Country_Code__c.
- Configuration dependencies:
    - Add custom metadata records to the Approved_Login_Country__mdt to maintain an allowlist of approved Alpha-2 country codes.
        - Salesforce Help for adding custom metadata records
        - Country Codes - ISO 3166

Detect Logins from Countries on Blocked List

- Event: LoginEvent
- What it does: Detects logins where the login country is in the blocklist of countries.
- Logic: Returns true if event.CountryIso is found in the block stored in Blocked_Login_Country__mdt.Alpha_2_Country_Code__c.
- Configuration dependencies:
    - Add custom metadata records to the Blocked_Login_Country__mdt to maintain an allowlist of approved Alpha-2 country codes.
        - Salesforce Help for adding custom metadata records
        - Country Codes - ISO 3166

Monitor Internal Logins that Bypass SSO

- Event: LoginEvent
- What it does: Detects internal standard users who log in directly instead of authenticating via SSO.
- Logic: Returns true when event.LoginType = 'Application' and event.UserType = 'Standard'
- Configuration dependencies:
    - n/a

Security Model (Authorization)

Detect Large API Queries by a Guest User

- Event: ApiEvent
- What it does: Detects high-volume API queries executed by Guest User profiles.
- Logic: Returns true when the ProfileId of the event.UserId is found in the Guest_Profile_Id__mdt.Guest_Profile_Id__c and event.RowsProcessed > 1000.
- Configuration dependencies:
    - n/a

Detect Privilege Escalation

- Event: PermissionSetEventStore
- What it does: Detects changes where there are changes to critical permission in permission sets and permission set groups.
- Logic: Returns true for every PermissionSetEventStore event received (no additional filtering is implemented in the Apex condition).
- Configuration dependencies:
    - n/a

Detect Report Exports by Unapproved Profiles

- Event: ReportEvent
- What it does: Detects report exports performed by users whose Profile is not in the approved allowlist.
- Logic: Returns true when event.Operation begins with 'ReportExported' and the Profile.Name of the event.UserId is not found in the allowlist stored in the Approved_Profile custom metadata object. All three export operations are evaluated: ReportExported, ReportExportedAsynchronously and ReportExportedUsingExcelConnector.
- Configuration dependencies:
    - Add custom metadata records to Approved_Profile__mdt to maintain an allowlist of approved Profile names.
        - Salesforce Help for adding custom metadata records

Integration

Detect API Access from Unapproved Third Party Apps

- Event: ApiEvent
- What it does: Detects API calls made by client applications that are not in the approved allowlist.
- Logic: Returns true when event.Client is not found in Approved_API_Client__mdt.Client_Name__c.
- Configuration dependencies:
    - Add custom metadata records to Approved_API_Client__mdt to maintain an allowlist of approved Client names.
        - Salesforce Help for adding custom metadata record

Data Loss Prevention (DLP)

Detect API File Downloads

- Event: FileEvent
- What it does: Detects when a file is downloaded via API.
- Logic: Returns true when FileAction = 'API_DOWNLOAD'.
- Configuration dependencies:
    - n/a

Total Record Detection for API Queries

- Event: ApiEvent
- What it does: Detects users who cumulatively access an unusually high number of unique records via List Views, Report Exports and API queries over a 24-hour period.
- Logic: Returns true where event.UserId = trxsecurity__RecordAccessLog__c.User__c and the sum oftrxsecurity__ RecordAccessLog__c.Count_of_Records__c values over the last 24 hours, plus the number of records accessed by the event currently being evaluated, exceeds 10,000.
- Configuration dependencies:
    - Activation of the TSP - Process Record Access Event flow, which populates RecordAccessLog__c based on ApiEvent
    - Recommended - Activation of the TSP - Delete Record Access Records flow, which deletes trxsecurity__RecordAccessLog__c records that were created more than 2 days ago and are thus no longer needed for this use case.
        - The batch apex that handles the deletion is also an invocable action should you want to call this from your own logic (e.g. trxsecurity__DeleteOldRecordAccessLogs.deleteOldRecords(...))
        - As of v1.6 the running user must have Delete permission on trxsecurity__RecordAccessLog__c. The records are queried and deleted in user mode, and a request from a user without that permission is rejected rather than partially processed. The included TSP Accelerator permission set grants this access.
        - As of v1.6 the days threshold must be a positive integer (1 or greater). A threshold of 0 is rejected, because it would delete every record created before today.
    - Salesforce Help for activating a flow
- *Note: Data management jobs, such as integrations or archiving, may be impacted by this policy based on how they interact with records, and the volume of records they interact with.*
- *Note: This policy was labeled "Total Access and Export Detection for API Queries" in versions prior to v1.4.1.*
- *Note: In versions prior to v1.6 only previously logged* trxsecurity__*RecordAccessLog__c records were counted. Because the current event is logged asynchronously, its records were not yet persisted when the policy was evaluated, so a single query large enough to exceed the*

threshold on its own was never detected. The current event's record count is now included in the comparison.

Total Record Detection for List Views

- Event: ListViewEvent
- What it does: Detects users who cumulatively access an unusually high number of unique records via List Views, Report Exports and API queries over a 24-hour period.
- Logic: Returns true where event.UserId = trxsecurity__RecordAccessLog__c.User__c and the sum of trxsecurity__RecordAccessLog__c.Count_of_Records__c values over the last 24 hours exceeds 10,000.
- Configuration dependencies:
    - Activation of the TSP - Process Record Access Event flow, which populates trxsecurity__RecordAccessLog__c based on ListViewEvent
    - Recommended - Activation of the TSP - Delete Record Access Records flow, which deletes trxsecurity__RecordAccessLog__c records that were created more than 2 days ago and are thus no longer needed for this use case.
        - The batch apex that handles the deletion is also an invocable action should you want to call this from your own logic (e.g. trxsecurity__DeleteOldRecordAccessLogs.deleteOldRecords(...))
        - As of v1.6 the running user must have Delete permission on trxsecurity__RecordAccessLog__c. The records are queried and deleted in user mode, and a request from a user without that permission is rejected rather than partially processed. The included TSP Accelerator permission set grants this access.
        - As of v1.6 the days threshold must be a positive integer (1 or greater). A threshold of 0 is rejected, because it would delete every record created before today.
    - Salesforce Help for activating a flow
- *Note: This policy was labeled "Total Access and Export Detection for List Views" in versions prior to v1.4.1.*
- *Note: In versions prior to v1.6 only previously logged* trxsecurity__*RecordAccessLog__c records were counted. Because the current event is logged asynchronously, its records were not yet persisted when the policy was evaluated, so a single query large enough to exceed the threshold on its own was never detected. The current event's record count is now included in the comparison.*

Total Record Detection for Report Exports

- Event: ReportEvent
- What it does: Detects users who cumulatively access an unusually high number of unique records via List Views, Report Exports and API queries over a 24-hour period.
- Logic: Returns true where event.Operation beings with 'ReportExported' and event.UserId = trxsecurity__RecordAccessLog__c.User__c and the sum of trxsecurity__RecordAccessLog__c.Count_of_Records__c values over the last 24 hours exceeds 10,000.

- Configuration dependencies:
    - Activation of the TSP - Process Record Access Event flow, which populates trxsecurity__RecordAccessLog__c based on ReportEvent and ReportEvent.Operation begins with 'ReportExported'
    - Recommended - Activation of the TSP - Delete Record Access Records flow, which deletes trxsecurity__RecordAccessLog__c records that were created more than 2 days ago and are thus no longer needed for this use case.
        - The batch apex that handles the deletion is also an invocable action should you want to call this from your own logic (e.g. trxsecurity__DeleteOldRecordAccessLogs.deleteOldRecords(...))
        - As of v1.6 the running user must have Delete permission on trxsecurity__RecordAccessLog__c. The records are queried and deleted in user mode, and a request from a user without that permission is rejected rather than partially processed. The included TSP Accelerator permission set grants this access.
        - As of v1.6 the days threshold must be a positive integer (1 or greater). A threshold of 0 is rejected, because it would delete every record created before today.
    - Salesforce Help for activating a flow
- *Note: This policy was labeled "Total Access and Export Detection for Report Exports" in versions prior to v1.4.1.*
- *Note: In versions prior to v1.6 only the exact Operation value 'ReportExported' was logged and counted. Asynchronous exports – the path Salesforce uses for large exports – never reached trxsecurity__RecordAccessLog__c, so the 24-hour total that all three Total Record Detection policies read was under-counted. Additionally, only previously logged records were counted; because the current event is logged asynchronously, a single export large enough to exceed the threshold on its own was never detected. The current export's record count is now included in the comparison, and where a report execution is chunked each chunk is counted once.*

Monitoring

Detect API Anomaly Threat Detection Event

- Event: ApiAnomalyEventStore
- What it does: Detects the Threat Detection event for potential API anomaly activity.
- Logic: Returns true for every ApiAnomalyEventStore event received (no additional filtering is implemented in the Apex condition).
- Configuration dependencies:
    - n/a

Detect Credential Stuffing Threat Detection Event

- Event: CredentialStuffingEventStore
- What it does: Detects the Threat Detection event for potential credential stuffing activity.
- Logic: Returns true for every CredentialStuffingEventStore event received (no additional filtering is implemented in the Apex condition).
- Configuration dependencies:
    - n/a

Detect Dormant User

- Event: LoginEvent
- What it does: Detects logins by users who have not had a successful login within the last 90 days.
- Logic: Returns true when LoginHistory contains no successful login for event.UserId in the 90 days before event.EventDate. Users with no LoginHistory records at all are treated as new accounts rather than dormant ones, so the policy does not apply to them.
- Configuration dependencies:
    - n/a
- *Note: In versions prior to v1.6 this policy examined only the 10 most recent login attempts, so a dormant account whose window contained more than 10 failed login attempts was not detected. The full 90-day window is now evaluated.*

Detect Guest User Anomaly Threat Detection Event

- Event: GuestUserAnomalyEventStore
- What it does: Detects the Threat Detection event for potential guest user anomaly activity.
- Logic: Returns true for every GuestUserAnomalyEventStore event received (no additional filtering is implemented in the Apex condition).
- Configuration dependencies:
    - n/a

Detect Login Anomaly Threat Detection Event

- Event: LoginAnomalyEventStore
- What it does: Detects the Threat Detection event for potential login anomaly activity.
- Logic: Returns true for every LoginAnomalyEventStore event received (no additional filtering is implemented in the Apex condition).
- Configuration dependencies:
    - n/a

Detect Report Anomaly Threat Detection Event

- Event: ReportAnomalyEventStore
- What it does: Detects the Threat Detection event for potential report anomaly activity.
- Logic: Returns true for every ReportAnomalyEventStore event received (no additional filtering is implemented in the Apex condition).
- Configuration dependencies:
    - n/a

Detect Session Hijacking Threat Detection Event

- Event: SessionHijackingEventStore
- What it does: Detects the Threat Detection event for potential session hijacking activity.
- Logic: Returns true for every SessionHijackingEventStore event received (no additional filtering is implemented in the Apex condition).
- Configuration dependencies:
    - n/a