On this page

Catalog

Event Types

Event types are the primary method of categorization within the Okta eventing platform. They allow consumers to easily group notable system occurrences based on behavior. This resource contains the complete event type catalog of this platform.

Catalog

The following is a full list of event types used in the System Log API with associated descriptions and related metadata.

Download a CSV file with all event types: Okta Event Types.

Note: Certain tags on the event type indicate the specific behavior of the associated System Log events:

- `oie-only`: This event type is only available in Okta Identity Engine enabled orgs
- `event-hook-eligible`: This event type is eligible for use with an event hook
- `changeDetails`: This event type may include the `changeDetails` object within an associated target

Search event types for...

All Releases

automation that depends on request cancellation. Okta Identity Governance API can be used to get more details about the canceled request.

`access` `event-hook-eligible`

Since: 2023.06.0

access.request.condition.activate

Access request condition activated. Can be used to audit access request condition to an Okta resource or to trigger downstream automation that depends on access request condition activation. Okta Identity Governance API can be used to get more details about the activated access request condition.

`access` `event-hook-eligible`

Since: 2024.03.0

access.request.condition.create

Access request condition created. Can be used to audit access request condition to an Okta resource or to trigger downstream automation that depends on access request condition creation. Okta Identity Governance API can be used to get more details about the created access request condition.

`access` `event-hook-eligible`

Since: 2024.03.0

access.request.condition.deactivate

Access request condition deactivated. Can be used to audit access request condition to an Okta resource or to trigger downstream automation that depends on access request condition deactivation. Okta Identity Governance API can be used to get more details about the deactivated access request condition.

`access` `event-hook-eligible`

Since: 2024.03.0

access.request.condition.delete

Access request condition deleted. Can be used to audit access request condition to an Okta resource or to trigger downstream automation that depends on access request condition deletion. Okta Identity

access.request.condition.invalidate

Access request condition invalidated. Can be used to audit access request condition to an Okta resource or to trigger downstream automation that depends on access request condition invalidation. Okta Identity Governance API can be used to get more details about the invalidated access request condition.

`access` `event-hook-eligible`

Since: 2024.03.0

access.request.condition.update

Access request condition updated. Can be used to audit access request condition to an Okta resource or to trigger downstream automation that depends on access request condition update. Okta Identity Governance API can be used to get more details about the updated access request condition.

`access` `event-hook-eligible` `changeDetails`

Since: 2024.03.0

access.request.create

Access request created. Can be used to audit access to an Okta resource or to trigger downstream automation that depends on request creation. Okta Identity Governance API can be used to get more details about the created request.

`access` `event-hook-eligible`

Since: 2022.11.0

access.request.reject

Access request rejected. Can be used to audit access to an Okta resource or to trigger downstream automation that depends on request rejection. Okta Identity Governance API can be used to get more details about the rejected request.

`access` `event-hook-eligible`

Since: 2024.05.0

access.request.resolve

access.request.sequence.create

Access request sequence created. Can be used to audit the approval sequence and when it was created and what was defined within the sequence to verify the approvals required. Okta Identity Governance API can be used to get more details about the created access request sequence.

`access` `event-hook-eligible` `changeDetails`

Since: 2024.10.0

access.request.sequence.delete

Access request sequence deleted. Can be used to audit the approval sequence and when it was deleted and what was defined within the sequence to verify the approvals required. Okta Identity Governance API can be used to get more details about the deleted access request sequence.

`access` `event-hook-eligible` `changeDetails`

Since: 2024.10.0

access.request.sequence.update

Access request sequence updated. Can be used to audit the approval sequence and when it was updated and what was defined within the sequence to verify the approvals required. Okta Identity Governance API can be used to get more details about the updated access request sequence.

`access` `event-hook-eligible` `changeDetails`

Since: 2024.10.0

account.org.add

Org is added to an Aerial account. Triggered when an org is added to an Aerial account. This event is fired in both the Aerial org and the added target org.

`account-service`

Since: 2023.10.1

Since: 2024.02.2

account.org.delete.request

Org is requested to be deleted. Triggered when a org is requested to be deleted. This event is fired in the Aerial org. The user or API client who requested the delete will be the actor and the org to be deleted will be in the target.

`account-service`

Since: 2024.02.2

account.org.product.update

Products are updated on an org. Triggered when Products are updated on an org. This event is fired only in the Aerial org.

`account-service`

Since: 2023.10.1

account.org.status.update

Org status is updated. Triggered when the status of an org is updated. This event is fired only in the Aerial org.

`account-service`

Since: 2023.10.1

analytics.feedback.provide

An admin has provided feedback on a detection Okta provided which indicated a change in user or sessio risk. This can be used to monitor feedback provided by admins in response to Okta determined changes in risk. This event is fired when an admin chooses to provide feedback on a detection event in the admin console.

See also: Identity Threat Protection with Okta AI Event Types

`risk` `security`

This event may be used to identify access by a user to a report data set from Okta. This may be useful to audit access to report data for security investigations, compliance audits, and evaluation of the utility of a report within the Org. This event only indicates that a user has downloaded the export file. The user that downloaded it may not be the user that requested generation of the export file. See analytics.reports.export.request and analytics.reports.export.generate for related actions.

Since: 2022.05.1

analytics.reports.export.generate

Okta has generated an export file for a report available in the admin console. This event may be used to identify whether Okta successfully generated the export file that a user requested for a report. This event is primarily useful for troubleshooting if a report fails to generate. This event does not indicate whether a user downloaded the report file. See analytics.reports.export.request and analytics.reports.export.download for related actions.

Since: 2022.05.1

analytics.reports.export.request

A user has requested that Okta generate an export file for a report available in the admin console. This event may be used to identify a request by a user to export a report data set from Okta. This may be useful to audit access to report data for security investigations, compliance audits, and evaluation of the utility of a report within the Org. This event only indicates that a user requested the export. It does not indicate that an export file was successfully generated by Okta nor that the export file was accessed by a user. See analytics.reports.export.generate and analytics.reports.export.download for those actions.

Since: 2022.05.1

app.access_request.approver.approve

Request to access an app was approved by an administrator-defined approver.

`app-instance-request` `event-hook-eligible`

Since: 2017.43

app.access_request.approver.deny

app.access_request.delete

Request to access an app was deleted by an administrator.

app-instance-request event-hook-eligible

Since: 2017.43

app.access_request.deny

Request to access an app was denied after at least one approver denied the request.

app-instance-request event-hook-eligible

Since: 2017.43

app.access_request.expire

Request to access an app expired by the system due to lack of approver action.

app-instance-request event-hook-eligible

Since: 2017.43

app.access_request.grant

Request to access an app was granted after all approvers approved the request.

app-instance-request event-hook-eligible

Since: 2017.43

app.access_request.request

Request to access an app was performed by a user.

app-instance-request event-hook-eligible

Since: 2017.43

app.ad.api.user_import.account_locked

Active Directory user account set to locked following profile update: user is locked in active directory.

ad-app

believe this contact should be imported.

ad-app

Since: 2015.47

app.ad.api.user_import.warn.skipped_user.attribute_invalid_value

Skipping import of user due to an invalid AD attribute.

ad-app

Since: 2015.47

app.ad.api.user_import.warn.skipped_user.missing_required_attribute

Skipping import of user due to a required AD attribute being null.

ad-app

Since: 2011.01

app.app_instance.csr.generate

Certificate signing request (CSR) generated.

app

Since: 2017.15

app.app_instance.csr.publish

Certificate signing request (CSR) published.

app

Since: 2017.15

app.app_instance.csr.revoke

Certificate signing request (CSR) revoked.

app

Since: 2017.15

app.app_instance.provision_sync_job.started and app.app_instance.provision_sync_job.failed.

admin app user-provision

Since: 2019.08.3

app.app_instance.provision_sync_job.failed

Fired when a provision sync job has failed. This can be used to identify when a provision sync job has failed. When fired, this event contains information about the reason the provision sync job failed. Related events include app.app_instance.provision_sync_job.started and app.app_instance.provision_sync_job.completed.

admin app user-provision

Since: 2019.08.3

app.app_instance.provision_sync_job.started

Fired when a provision sync job has successfully started. This can be used to confirm that a provision sync job has successfully started. Related events include app.app_instance.provision_sync_job.completed and app.app_instance.provision_sync_job.failed.

admin app user-provision

Since: 2019.08.3

app.audit_report.download

Application access report downloaded.

app

Since: 2017.52

app.audit_report.download.local.active

Application access report downloaded.

app

Since: 2017.52

app.audit_report.download.rogue.report

Rogue report downloaded.

app

Since: 2017.52

app.generic.unauth_app_access_attempt

User attempted unauthorized access to app.

app

Since: 2016.06

app.inbound_del_auth.login_success

Successful inbound delegated authentication request for user.

delegated-auth

Since: 2016.18

app.kerberos_rich_client.account_not_found

Kerberos based rich client authentication failed: Could not find Office 365 app user for the AD user with principal id.

app kerberos-rich-client

Since: 2017.50

app.kerberos_rich_client.instance_not_found

Kerberos based rich client authentication failed: Unknown app instance id.

app kerberos-rich-client

Since: 2017.50

app.kerberos_rich_client.multiple_accounts_found

app.kerberos_rich_client.user_authentication_successful

Kerberos based rich client authentication successful for Office 365 user.

app kerberos-rich-client

Since: 2017.52

app.keys.clone

Application signing key cloned.

app

Since: 2017.25

app.keys.generate

New signing key generated.

app

Since: 2017.25

app.keys.rotate

Application signing key rotated.

app

Since: 2017.25

app.ldap.password.change.failed

Password change failed.

ldap-app

Since: 2014.18

app.oauth2.admin.consent.grant

Administrator consent granted for scope. This event can be used to track when an administrator grants consent to a client to request a specific scope. This event is fired when an admin grants consent.

Administrator consent revoked for scope. This event can be used to track when an administrator revokes consent to a client to request a specific scope. This event is fired when an admin revokes consent.

`oauth2` `oauth2-as-runtime` `oauth2-org-as`

Since: 2019.12.0

app.oauth2.api_resource.create

OAuth2 API Resource is created. Manage and audit lifecycle events of API resources. Administrators are made aware that a new API resource is getting created under Authorization servers.

`oauth2` `oauth2-api-resource`

Since: 2022.04.2

app.oauth2.api_resource.delete

OAuth2 API Resource is deleted. Manage and audit lifecycle events of API resources. Administrators are made aware that a new API resource is getting deleted under Authorization servers.

`oauth2` `oauth2-api-resource`

Since: 2022.04.2

app.oauth2.api_resource.update

OAuth2 API Resource is updated. Manage and audit lifecycle events of API resources. Administrators are made aware that a new API resource is getting updated under Authorization servers.

`oauth2` `oauth2-api-resource`

Since: 2022.04.2

app.oauth2.as.authorize

OAuth2 authorization request.

`oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2016.14

app.oauth2.as.authorize.code

app.oauth2.as.authorize.implicit.access_token

OAuth2 authorization implicit access token request.

`oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2016.14

app.oauth2.as.authorize.implicit.id_token

OAuth2 authorization implicit ID token request.

`oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2016.14

app.oauth2.as.authorize.scope_denied

Some of the requested scopes were denied by the policy.

`oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2016.14

app.oauth2.as.consent.grant

User granted consent to app.

`event-hook-eligible` `oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2016.14

app.oauth2.as.consent.revoke

Consent revoked.

`event-hook-eligible` `oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2016.14

app.oauth2.as.consent.revoke.implicit.as

All consent revoked for authorization server.

`event-hook-eligible` `oauth2` `oauth2-as-runtime` `oauth2-custom-as`

`event-hook-eligible` `oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2016.14

app.oauth2.as.consent.revoke.implicit.scope

All consent revoked for scope.

`event-hook-eligible` `oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2016.14

app.oauth2.as.consent.revoke.implicit.user

Consent for all scopes revoked for user.

`event-hook-eligible` `oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2016.14

app.oauth2.as.consent.revoke.user

All consent revoked for user.

`event-hook-eligible` `oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2016.14

app.oauth2.as.consent.revoke.user.client

User consent revoked for client.

`event-hook-eligible` `oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2016.14

app.oauth2.as.evaluate.claim

Claim evaluation for OAuth 2.0 token. This event is triggered when the OAuth 2.0 authorization server's claim evaluation process can't be completed and fails. This event is useful when detecting misconfigured claims. Recorded details include the requester's ID, the client ID, the user ID, and the claims that couldn't be

app.oauth2.as.interact.interaction_code

Fired when interaction code is generated by OIE. This event can be used by administrators to audit interaction_code generation, and troubleshoot why the IdX transaction has failed. When fired, this event contains hashed values of the interaction_code and interaction_handle, as well as information about the client to which they were issued.

`oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2021.01.0

app.oauth2.as.interact.interaction_handle

Fired when interaction handle is generated by OIE. This event can be used by administrators to detect if additional interaction is required and an interaction handle has been issued. When fired this event contains interaction handle hash and the client to which it was issued.

`oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2021.01.0

app.oauth2.as.key.rollover

Custom Authorization Server token signing key rolled over.

`oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2016.14

app.oauth2.as.token.detect_reuse

Detect one-time refresh token attempted reuse. This event can be used by administrators to detect and audit attempted reuse of one-time refresh tokens. When fired this event contains information about the user, client to which the refresh token was minted, and the hash of the refresh tokens.

`oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2020.09.3

app.oauth2.as.token.grant

app.oauth2.as.token.grant.access_token

OAuth 2.0 access token is granted. This event is triggered within OAuth 2.0 frameworks when an app successfully grants an access token to a user or service. The event occurs post-authentication and authorization, marking the final step in accessing protected resources. Use this event as a comprehensive audit trail for issued tokens. The event captures details such as the client ID, subject ID, token attributes (for example: scope, validity period), and the grant type used. This information helps with security audits, ensuring compliance with access policies and troubleshooting authorization flows. Specifically, variations in token attributes and grant type offer insights into the security posture and operational efficiency of OAuth 2.0 implementations. While this event primarily signifies successfully issued tokens, the event details are helpful in many areas. They help flag potential misuse of token grants or anomalies in token attributes. The event details also help facilitate a prompt response to deviations from established security practices.

`oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2016.14

app.oauth2.as.token.grant.device_secret

Grant an OAuth2 device_secret for the Native SSO flow. This event adds tracking to let admins know when Native SSO is being used to protect desktop or mobile apps. When fired this event contains the device secret id which administrators can use to correlate with single logout events across native desktop apps.

`oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2021.04.2

app.oauth2.as.token.grant.id_token

OAuth 2.0 ID token is granted. This event occurs when an OAuth 2.0 authorization server grants an ID token to a client after successful authentication. The ID token, which encapsulates the user's identity information, verifies the user's identity to the client app. Recorded details include the client ID, user ID, token issuance time, and claims associated with the user's identity. You can use this data for security audits, enabling precise tracking of user identity verification across apps. The issuance of an ID token follows established protocols for secure authentication. This ensures that sensitive user information is transmitted securely between the authorization server and the client.

`oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2016.14

app.oauth2.as.token.revoke

OAuth2 token revocation request.

`oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2016.14

app.oauth2.authorize

OIDC authorization request.

`oauth2` `oauth2-as-runtime` `oauth2-org-as`

Since: 2016.14

app.oauth2.authorize.code

OIDC authorization code request.

`oauth2` `oauth2-as-runtime` `oauth2-org-as`

Since: 2016.14

app.oauth2.authorize.implicit.access_token

OIDC authorization implicit access token request.

`oauth2` `oauth2-as-runtime` `oauth2-org-as`

Since: 2016.14

app.oauth2.authorize.implicit.id_token

OIDC authorization implicit ID token request.

`oauth2` `oauth2-as-runtime` `oauth2-org-as`

Since: 2016.14

app.oauth2.client.lifecycle.activate

app.oauth2.client.lifecycle.create

Create OAuth client.

`oauth2` `oauth2-client` `oauth2-client-lifecycle`

Since: 2017.24

app.oauth2.client.lifecycle.deactivate

Deactivate OAuth client.

`oauth2` `oauth2-client` `oauth2-client-lifecycle`

Since: 2017.24

app.oauth2.client.lifecycle.delete

Delete OAuth client.

`oauth2` `oauth2-client` `oauth2-client-lifecycle`

Since: 2017.24

app.oauth2.client.lifecycle.update

Update OAuth client.

`oauth2` `oauth2-client` `oauth2-client-lifecycle`

Since: 2017.24

app.oauth2.client.privilege.grant

An OAuth 2.0 client app's admin privileges changed. This can be used to audit the provisioning of admin privileges for OAuth 2.0 client apps. When fired, this event contains information about the type of admin privileges the OAuth 2.0 client app currently has. Related events include: APP_OAUTH2_CLIENT_PRIVILEGE_REVOKE.

`event-hook-eligible` `oauth2` `oauth2-client`

Since: 2023.04.1

APP_OAUTH2_CLIENT_PRIVILEGE_GRANT.

`event-hook-eligible` `oauth2` `oauth2-client`

Since: 2023.04.1

app.oauth2.client.read_client_secret

Read OAuth client's secret(s). Use this event to verify that an OAuth client's secret(s) have been read when the client is returned in certain API responses. For example, an admin might use this event to audit if a client's secrets were read when using the client credentials management API. When fired, this event indicates that an OAuth client's secrets were read. The targets array may include references to multiple client secrets.

`oauth2` `oauth2-client`

Since: 2024.08.1

app.oauth2.client_id_rate_limit_warning

Fired when requests from a single client id has consumed majority of an org's rate limit on the OAuth2 endpoint. This event can be used by admins to discover and deactivate a rogue client. The admin is able to manage the client via the Syslog UI. When fired, this event contains information about the responsible client id. As of release, this event is fired when a single client id consumes 90% of an org's OAuth2 rate limit; this threshold is subject to change.

`oauth2` `oauth2-client`

Since: 2019.04.2

app.oauth2.consent.grant

User granted consent to app. This event can be used to identify the org AS consent grant. When fired, the event contains information about the successful consent grant by org AS.

`oauth2` `oauth2-as-runtime` `oauth2-org-as`

Since: 2021.10.2

app.oauth2.credentials.lifecycle.activate

app.oauth2.credentials.lifecycle.create

OAuth client credentials (either client secret or JWK) is activated for an application. Use this event to find out if an application has activated a new client secret or private/public key. This could be used to audit changes made to client credentials.

`oauth2` `oauth2-client` `oauth2-client-credentials-lifecycle`

Since: 2022.05.3

app.oauth2.credentials.lifecycle.deactivate

OAuth client credentials (either client secret or JWK) is deactivated for an application. Use this event to find out if an application has an existing client secret or private/public key that has been deactivated. This could be used to audit changes made to client credentials.

`oauth2` `oauth2-client` `oauth2-client-credentials-lifecycle`

Since: 2022.05.3

app.oauth2.credentials.lifecycle.delete

OAuth client credentials (either client secret or JWK) is deleted for an application. Use this event to find out if an application has an existing client secret or private/public key that has been deleted. This could be used to audit changes made to client credentials.

`oauth2` `oauth2-client` `oauth2-client-credentials-lifecycle`

Since: 2022.05.3

app.oauth2.interact.interaction_code

Fired when interaction code is generated by OIE. This event can be used by administrators to audit interaction_code generation, and troubleshoot why the IdX transaction has failed. When fired, this event contains hashed values of the interaction_code and interaction_handle, as well as information about the client to which they were issued.

`oauth2` `oauth2-as-runtime` `oauth2-org-as`

Since: 2021.01.0

oauth2 oauth2-as-runtime oauth2-org-as

Since: 2021.01.0

app.oauth2.invalid_client_credentials

Multiple requests with invalid client credentials for client id.

oauth2 oauth2-as-runtime oauth2-org-as

Since: 2016.14

app.oauth2.key.rollover

Org Authorization Server token signing key rolled over.

oauth2 oauth2-as-runtime oauth2-org-as

Since: 2016.14

app.oauth2.signon

User performed OIDC single sign on to app.

oauth2 oauth2-client

Since: 2016.14

app.oauth2.token.detect_reuse

Detect one-time refresh token attempted reuse. This event can be used by administrators to detect and audit attempted reuse of one-time refresh tokens. When fired this event contains information about the use client to which the refresh token was minted, and the hash of the refresh tokens.

oauth2 oauth2-as-runtime oauth2-org-as

Since: 2020.09.3

app.oauth2.token.grant

OIDC token request.

oauth2 oauth2-as-runtime oauth2-org-as

`oauth2` `oauth2-as-runtime` `oauth2-org-as`

Since: 2016.14

app.oauth2.token.grant.id_token

OIDC id token is granted.

`oauth2` `oauth2-as-runtime` `oauth2-org-as`

Since: 2016.14

app.oauth2.token.grant.refresh_token

OIDC refresh token is granted.

`oauth2` `oauth2-as-runtime` `oauth2-org-as`

Since: 2016.14

app.oauth2.token.revoke

OIDC token revocation request.

`oauth2` `oauth2-as-runtime` `oauth2-org-as`

Since: 2016.14

app.oauth2.token.revoke.implicit.as

Tokens revoked for authorization server.

`oauth2` `oauth2-as-runtime` `oauth2-org-as`

Since: 2016.14

app.oauth2.token.revoke.implicit.client

Tokens revoked for client.

`oauth2` `oauth2-as-runtime` `oauth2-org-as`

Since: 2016.14

app.oauth2.trusted_server.add

Trusted authorization server is added. Administrators can use this event to debug and audit trusted authorization server operations. When fired, this event contains the authorization server IDs of the servers involved.

`event-hook-eligible` `oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2023.02.0

app.oauth2.trusted_server.delete

Trusted authorization server is removed. Administrators can use this event to debug and audit trusted authorization server operations. When fired, this event contains the authorization server IDs of the servers involved.

`event-hook-eligible` `oauth2` `oauth2-as-runtime` `oauth2-custom-as`

Since: 2023.02.0

app.office365.api.change.domain.federation.success

Successfully updated the domain federation from old settings to new settings.

`app` `office365-app`

Since: 2017.01

app.office365.api.error.ad.user

User is assigned to more than one instance of Active Directory, could not set Immutable ID.

`app` `office365-app`

Since: 2017.01

app.office365.api.error.check.user.exists

Could not determine status of Office 365 user, received error.

`app` `office365-app`

app office365-app

Since: 2017.01

app.office365.api.error.deactivate.user

Could not deactivate Office 365 user, received error.

app office365-app

Since: 2017.01

app.office365.api.error.download.custom.objects

Could not download group/role/license data for your Office 365 instance, received error.

app office365-app

Since: 2017.01

app.office365.api.error.download.groups

Could not download all groups from your Office 365 instance, received error.

app office365-app

Since: 2017.01

app.office365.api.error.download.users

Could not download all users from your Office 365 instance, received error.

app office365-app

Since: 2017.01

app.office365.api.error.endpoint.unavailable

Unable to reach the Office 365 endpoint.

app office365-app

Since: 2017.01

app.office365.api.error.get.company.dirsync.status.failure

Unable to provision user to Office 365, because 'Directory Sync' value in Azure Active Directory is unsupported. Please visit the Azure Active Directory portal and set 'Directory Sync' state to Activated and retry.

app office365-app

Since: 2017.01

app.office365.api.error.get.company.dirsync.status.pending

Unable to provision user to Office 365, because 'Directory Sync' value in Azure Active Directory not yet in Activated state. This may take up to 72 hours. Please visit the Azure Active Directory portal and retry when in Activated state.

app office365-app

Since: 2017.01

app.office365.api.error.get.object.ids.by.group.id

Could not get users by group id from your Office 365 instance, received error.

office365-app

Since: 2018.37

app.office365.api.error.group.create.failure

Could not create Office 365 group, received error.

app office365-app

Since: 2017.01

app.office365.api.error.group.create.failure.name.in.use

Could not create Office 365 group because the name is already in use, received error.

app office365-app

app office365-app

Since: 2017.01

app.office365.api.error.group.membership.update.assignment.failure

Could not update the Office 365 group membership because of an error assigning a user to the group, received error.

app office365-app

Since: 2017.01

app.office365.api.error.group.membership.update.failure

Could not update the Office 365 group membership, received error.

app office365-app

Since: 2017.01

app.office365.api.error.group.membership.update.group.not.found.failure

Could not update the Office 365 group membership because the group could not be found, received error.

app office365-app

Since: 2017.01

app.office365.api.error.group.membership.update.removal.failure

Could not update the Office 365 group membership because of an error removing a user from the group, received error.

app office365-app

Since: 2017.01

app.office365.api.error.group.update.failure

Could not update Office 365 group, received error.

app office365-app

app office365-app

Since: 2017.01

app.office365.api.error.import.profile

Could not import profile for Office 365 user, received error.

app office365-app

Since: 2017.01

app.office365.api.error.no.endpoints.found

No Office 365 endpoint found to send our request.

app office365-app

Since: 2017.01

app.office365.api.error.push.password

Could not push password for Office 365 user, received error.

app office365-app

Since: 2017.01

app.office365.api.error.push.profile

Could not push profile for Office 365 user, received error.

app office365-app

Since: 2017.01

app.office365.api.error.reactivate.user

Could not reactivate Office 365 user, received error.

app office365-app

Since: 2017.01

app.office365.api.error.remove.domain.federation.failure.access.denied

Unable to remove the domain federation because the admin user is not authorized to perform the task.

app office365-app

Since: 2017.01

app.office365.api.error.remove.domain.federation.failure.domain.not.found

Unable to remove the domain federation because the specified domain was not found.

app office365-app

Since: 2017.01

app.office365.api.error.revoke.refresh.token

Failed to revoke refresh tokens for user.

app office365-app

Since: 2017.01

app.office365.api.error.set.company.dirsync.failure

Unable to enable Office 365 directory sync for the company, received error.

app office365-app

Since: 2017.01

app.office365.api.error.set.company.dirsync.status.failure

Unable to enable Office 365 directory sync for the company, because 'Directory Sync' value in Azure Active Directory is unsupported. Please visit the Azure Active Directory portal and set 'Directory Sync' state to Activated.

app office365-app

Since: 2017.01

app.office365.api.error.set.domain.federation.failure.access.denied

Unable to setup the domain federation because the admin user is not authorized to perform the task.

app office365-app

Since: 2017.01

app.office365.api.error.set.domain.federation.failure.domain.default

Unable to setup the domain federation because the specified domain is the default domain.

app office365-app

Since: 2017.01

app.office365.api.error.set.domain.federation.failure.domain.not.found

Unable to setup the domain federation because the specified domain was not found.

app office365-app

Since: 2017.01

app.office365.api.error.sync.contact

Failed to sync contact, received error.

app office365-app

Since: 2017.01

app.office365.api.error.sync.finalize

Failed to finalize export to Office 365, received error.

app office365-app

Since: 2017.01

app.office365.api.error.sync.group

app.office365.api.error.sync.not.activated

Sync could not execute because Office 365 directory sync for the company not yet Activated. Sync will retry after a period of time.

app office365-app

Since: 2017.01

app.office365.api.error.sync.set.attribute

Failed to set attribute, received error.

app office365-app

Since: 2017.01

app.office365.api.error.sync.user

Failed to sync user, received error.

app office365-app

Since: 2017.01

app.office365.api.error.unable.to.create.graph.client

An error occurred while creating the Azure Active Directory Graph API client. Please try the last operation again. If this error persists, please contact Okta support.

app office365-app

Since: 2017.01

app.office365.api.error.validate.admin.creds

User does not have the Company Administrator role. Please try again with a user which has this role.

app office365-app

Since: 2017.01

app.office365.api.error.validate.creds

app.office365.api.error.validate.creds.unknown.exception

Could not communicate with Office 365 to validate your credentials, received error.

app office365-app

Since: 2017.01

app.office365.api.error.x-ms-forwarded-client-ip-header.absent

X-MS-Forwarded-Client-IP header either empty or not found in the request.

app office365-app

Since: 2017.01

app.office365.api.remove.domain.federation.success

Successfully removed the domain federation.

app office365-app

Since: 2017.01

app.office365.api.set.domain.federation.success

Successfully set up the domain federation with new settings.

app office365-app

Since: 2017.01

app.office365.api.sync.complete

User sync completed.

app office365-app

Since: 2017.01

app.office365.api.sync.heartbeat.sent

Heartbeat sent to Microsoft Azure Active Directory.

app office365-app

app office365-app

Since: 2017.01

app.office365.api.sync.job.complete.contact

Sync job completed.

app office365-app

Since: 2017.01

app.office365.api.sync.job.complete.group

Sync job completed.

app office365-app

Since: 2017.01

app.office365.api.sync.job.complete.user

Sync job completed.

app office365-app

Since: 2017.01

app.office365.clientplatform.conversion.job.processing.app.instance

Begin processing client access conversion for app instance.

app office365-app

Since: 2017.01

app.office365.clientplatform.conversion.job.skipping.migration

Skipping migration of client access rules for app instance.

app office365-app

Since: 2017.01

app.office365.dirsync.skipping.critical-system-object

Skipping sync of critical system object.

app office365-app

Since: 2017.01

app.office365.dirsync.skipping.non-security-group-invalid-mail

Skipping sync of non security object with invalid mail.

app office365-app

Since: 2017.01

app.office365.dirsync.skipping.reserved-attribute-value

Skipping sync of object with reserved attribute value.

app office365-app

Since: 2017.01

app.office365.dirsync.skipping.systemmailbox

Skipping sync of system mailbox object.

app office365-app

Since: 2017.01

app.office365.dirsync.skipping.without-name-and-displayname

Skipping sync of non security object without name and display name.

app office365-app

Since: 2017.01

app.office365.error.importing.user

app.office365.graph.api.error.no.mailbox.found

No MailBox found for Office 365 user.

app office365-app

Since: 2017.01

app.office365.graph.api.error.rate-limit.exceeded

Rate limit exceeded for Microsoft Graph.

app office365-app

Since: 2017.01

app.office365.graph.api.error.service.principal.creation.failed

Failure while trying to create service principal.

office365-app

Since: 2017.01

app.office365.graph.api.error.service.principal.msgraph.authentication.failure

Failure while trying to create service principal due to a Mircrosoft Graph authentication issue.

office365-app

Since: 2017.01

app.office365.service.principal.cleanup.job.complete

End processing Office 365 service principal cleanup.

app office365-app

Since: 2017.01

app.office365.service.principal.cleanup.job.invalid.credentials

The admin username or password is invalid. Please use the Azure Active Directory cmdlets to execute the command 'Remove-MsolServicePrincipal -AppPrincipalId' to manually cleanup the service principal.

Begin performing Office 365 service principal cleanup.

app office365-app

Since: 2017.01

app.office365.service.principal.cleanup.job.skipping.missing.creds

Skipping app instance during Office 365 service principal cleanup as it does not contain Office 365 admin user credentials. Please use the Azure Active Directory cmdlets to execute the command 'Remove-MsolServicePrincipal -AppPrincipalId' to manually cleanup the service principal.

app office365-app

Since: 2017.01

app.office365.service.principal.cleanup.job.skipping.no.service.principal

Skipping app instance during Office 365 service principal cleanup as it does not have a service principal.

app office365-app

Since: 2017.01

app.office365.service.principal.cleanup.job.unable.to.delete.service.principal

Unable to automatically delete the Office 365 service principal. Please use the Azure Active Directory cmdlets to execute the command 'Remove-MsolServicePrincipal -AppPrincipalId' to manually cleanup the service principal.

app office365-app

Since: 2017.01

app.office365.user.delete.success

Successfully deleted the Office 365 user.

app office365-app

Since: 2017.01

app.office365.user.lifecycle.action.failed

app.office365.user.remove.licenses.success

Successfully removed all the licenses for the Office 365 user.

app office365-app

Since: 2017.01

app.policy.sign_on.update

Update app sign on policy. This event is used to audit when an app sign on policy is updated. This event is fired when an admin updates an app's sign on policy and logs what was changed.

policy

Since: 2022.08.0

app.radius.agent.listener.failed

Radius agent listener failed.

app radius

Since: 2018.13

app.radius.agent.listener.succeeded

Radius agent listener succeeded.

app radius

Since: 2018.13

app.radius.agent.port_inaccessible

Radius agent failed to listen on port.

app radius

Since: 2018.13

app.radius.agent.port_reaccessible

Radius agent was able to listen on port again.

No permission accessing any Radius app info. This event can be used to monitor and notify admins when some users who access radius app info have no permission. Fired when users who access radius app info have no permission.

app radius

Since: 2020.08.0

app.radius.info_access.partial_permission

No permission accessing info for part of Radius apps. This event can be used to monitor and notify admins when some users who access radius app info have only partial permission. Fired when users who access radius app info have partial permission.

app radius

Since: 2020.08.0

app.realtimesync.import.details.add_user

Real time sync added new User.

app

Since: 2014.25

app.realtimesync.import.details.delete_user

Real time sync removed existing User.

app

Since: 2014.25

app.realtimesync.import.details.update_user

Fired when a real time import includes an update to an existing user. This can be used to see details about the user updates included in a real time sync import. When fired, this event contains information about the type of update made, including whether or not a user was suspend or unsuspended. Related events include: app.realtimesync.import.details_add_user and app.realtimesync.import.details_delete_user.

app

Enduser Dashboard. The application request attempts to send an email to an admin with the user's request. This event only indicates that the request was made, not necessarily that the email was successfully delivered.

Since: 2024.09.0

app.rum.config.validation.error

Error validating instance configuration. Can be used to identify configuration issues with remote user management.

rum

Since: 2018.42

app.rum.is.api.account.error

RUM API account is not configured or empty. Can be used to identify RUM API account configuration issues.

rum

Since: 2018.42

app.rum.package.thrown.error

Errors during execution. Can be used to identify any errors during execution of remote user management.

rum

Since: 2018.42

app.rum.validation.error

Error during package validation. Can be used to identify validation issues with remote user management packages.

rum

Since: 2018.42

app.saml.sensitive.attribute.update

app cvd

Since: 2019.01.1

app.user_management

Imported new or deleted existing member of an application group.

app-user-management

Since: 2016.04

app.user_management.grouppush.mapping.created.from.rule

A Group Push mapping to the group has been created from the rule.

app

Since: 2017.51

app.user_management.grouppush.mapping.created.from.rule.error.duplicate

A Group Push mapping to the group did not get created from rule because an existing mapping already existed.

app

Since: 2017.51

app.user_management.grouppush.mapping.created.from.rule.error.validation

A Group Push mapping to the group did not get created from rule because of the validation error.

app

Since: 2017.51

app.user_management.grouppush.mapping.created.from.rule.errors

A Group Push mapping to the group did not get created from rule.

app

Since: 2017.51

app.user_management.import.csv.line.error

Error reading line from CSV.

app

Since: 2017.51

app.user_management.push_new_user_success

Successfully pushed new user account to app.

app

Since: 2017.51

app.user_management.update_from_master_failed

Could not apply import.

app

Since: 2017.51

app.user_management.user_group_import.create_failure

Failed to create group from app.

app app-user-management

Since: 2018.03

app.user_management.user_group_import.delete_success

Deleted the group from app.

app app-user-management

Since: 2018.03

app.user_management.user_group_import.update_failure

app.user_management.user_group_import.upsert_fail

Failed to import the group from app. This event helps identify when a group is failed to be imported. Fired when we skip processing an import of a group.

app app-user-management

Since: 2020.07.1

app.user_management.user_group_import.upsert_success

Imported the group from app.

app app-user-management

Since: 2018.03

application.appuser.mapping.invalid.expression

App user property mapping has invalid expressions. Can be used to identify invalid expressions. Note that a single event is fired for all invalid expressions.

app

Since: 2018.47

application.cache.invalidate

Event fired when a app list cache is invalidated because a new app is created. Can be used to make sure App List cache is invalidated after a new app is created.

invalidate-app-list-cache

Since: 2018.42

application.configuration.detect_error

Application configuration error detected.

app

Since: 2016.13

application.configuration.disable_fed_broker_mode

Disable Federation Broker Mode for app.

app

Since: 2017.24

application.configuration.enable_delauth_outbound

Enable delegated authentication for app.

app

Since: 2016.13

application.configuration.enable_fed_broker_mode

Enable Federation Broker Mode for app.

app

Since: 2017.24

application.configuration.import_schema

Okta couldn't download application configuration. Can be used to identify when an app schema couldn't b downloaded from a remote application. Event fired when Okta couldn't download application-specific data from a remote app. This may happen when admin updates provisioning details.

app-api

Since: 2017.33

application.configuration.read_client_secret

A client secret in an MFA-only app has been read. Verify that a client secret in an MFA-only app has been read. This events indicates that a client secret in an MFA-only app has been read.

agent app

Since: 2024.03.2

application.configuration.update

Okta couldn't verify api credentials. Can be used when Okta couldn't check the credentials by execution some custom, application dependent, set of requests. Okta fires this event to notify issues with credentials validation. Could be issues with proper permissions as well.

app-api

Since: 2017.33

application.configuration.update_api_credentials_for_pass_change

Update API credentials due to user updating password.

app

Since: 2016.13

application.configuration.update_logo

Change app logo.

app

Since: 2016.13

application.configuration.update_rate_limits

Update rate limits for an OAuth App. This can be used to track the updates to rate limits for an OAuth application. When fired, this event contains details about the actor, who triggered the event, the OAuth app for which the rate limit was updated, etc. Actual value change details can be found in debug data such as the old and new values.

app

Since: 2023.01.0

application.integration.api_query

application.integration.authentication_failure

Error authenticating. Can be used when Okta couldn't authenticate with the provided credentials to a remote api. Okta fires this event when it couldn't access a remote api with provided credentials.

app-api

Since: 2017.33

application.integration.general_failure

Generic error occured. Can be used when there is some uncategorized error occurs. Okta fires this event for different unhandled exceptions.

app-api

Since: 2017.33

application.integration.rate_limit_exceeded

API rate limit exceeded. Can be used when Okta reaches api calls/minute rate limit. Okta fires this event when there are too many requests for a specific customer.

app-api

Since: 2017.33

application.integration.transfer_files

Unable to transfer files. Can be used when Okta fails to transfer files from one user to another. Okta fires this event when it fails to process user-to-user file transfers.

app-api

Since: 2017.33

application.lifecycle.activate

Activate application.

app event-hook-eligible

Since: 2016.13

application.lifecycle.deactivate

Deactivate application.

app event-hook-eligible

Since: 2016.13

application.lifecycle.delete

Delete application.

app event-hook-eligible

Since: 2016.13

application.lifecycle.update

Update application.

app event-hook-eligible changeDetails

Since: 2016.13

application.policy.sign_on.deny_access

Deny user access due to app sign on policy. When fired due to app assurance being evaluated as unsatisfiable (the policy requirements could not be satisfied by the users' current set of available authenticator enrollments), this event contains information about the user and the app that the user is trying to authenticate into.

app event-hook-eligible

Since: 2016.13

application.policy.sign_on.rule.create

Create rule for app sign on policy.

app

Since: 2016.13

application.policy.sign_on.update

Update app sign on policy.

`app` `changeDetails`

Since: 2016.13

application.provision.field_mapping_rule.change

Event fired when field mapping rules modified. Can be used to make sure when custom mapping rules are modified.

`field-mapping-rule-modification`

Since: 2018.42

application.provision.group.add

Fired when Okta provisions a new group on a remote application. Can be used to identify when Okta provisions a group on a remote application. Event fired when the group provisioning failed for any reason.

`app-api`

Since: 2017.33

application.provision.group.import

Fired when Okta downloads a remote group. Can be used to identify when Okta tries to download remote group details. Event fired when Okta fails to reach the group detail from a remote application.

`app-api`

Since: 2017.33

application.provision.group.remove

Fired when Okta removes a remote group. Can be used to identify when a group has been unassigned. Event fired when Okta failed to delete group from remote application.

`app-api`

fired when Okta fails to update a remote group for any reason.

app-api

Since: 2017.33

application.provision.group.verify_exists

Fired when group no longer exists on a remote application. Can be used to identify when a group no longer exists on a remote application. Event fired when group push enhancement enabled and there is no group found on update or delete.

app-api

Since: 2017.33

application.provision.group_membership.add

Failed to assign a user to a group. Can be used when Okta failed to assign user to a group on remote application. Okta fires this event if there are any issues while provision a membership to a remote application.

app-api

Since: 2017.33

application.provision.group_membership.import

Error while downloading memberships. Can be used when Okta failed to download users and groups relationships. Okta fires this event if there are any issues while importing a membership from a remote application.

app-api

Since: 2017.33

application.provision.group_membership.remove

Fired when there is an error while removing user(s) from group. Can be used when Okta failed to unassign user from a group on remote application. Okta fires this event when there are any issues while provision a membership to a remote application.

app-api

to push updated memberships to a remote application. Okta fires this event when couldn't update memberships on a remote application. Could be user removal/addition.

app-api

Since: 2017.33

application.provision.group_push.activate_mapping

Group push activated mappings.

app

Since: 2017.29

application.provision.group_push.deactivate_mapping

Group push deactivated mappings. Can be used to audit when a group push mapping is deactivated or to trigger downstream automation. The corresponding event type for activating a group push mapping is application.provision.group_push.activate_mapping.

app

Since: 2024.07.2

application.provision.group_push.delete_appgroup

Group push deleted application group.

app

Since: 2017.29

application.provision.group_push.mapping.and.groups.deleted.rule.deleted

An existing mapping and its target groups have been deleted because a mapping rule was deleted.

app

Since: 2017.29

application.provision.group_push.mapping.app.group.renamed

application.provision.group_push.mapping.app.group.renamed.failed

A mapped app group couldn't be renamed when the source group was renamed.

app

Since: 2017.29

application.provision.group_push.mapping.created

A new mapping has been created.

app

Since: 2017.29

application.provision.group_push.mapping.created.from.rule.warning.duplicate.n...

A new mapping from a rule was not created due to a duplicate group name.

app

Since: 2017.29

application.provision.group_push.mapping.created.from.rule.warning.duplicate.n...

A new mapping from a rule was not created due to another mapping will be created that has the same user group name.

app

Since: 2017.29

application.provision.group_push.mapping.created.from.rule.warning.upsertGrou...

An upsert to a group caused group push rule re-evaluation. A new mapping from a rule was not created d to a duplicate group name.

app

Since: 2017.29

application.provision.group_push.mapping.deactivated.source.group.renamed

application.provision.group_push.mapping.deactivated.source.group.renamed.fai...

An existing mapping couldn't be deactivated when the source group was renamed.

app

Since: 2017.29

application.provision.group_push.mapping.update.or.delete.failed

Group push mapping change failed and will be retried. Can be used to identify transient errors that may temporarily impact the group push mapping but likely do not require admin intervention. This event typically requires no action as the corresponding operation will be retried. Refer to application.provision.group_push.mapping.update.or.delete.failed for events that may require intervention.

app

Since: 2017.29

application.provision.group_push.mapping.update.or.delete.failed.with.error

Group push mapping change failed and cannot be retried. Can be used to identify group push mapping errors which may require admin intervention to address. Unlike the similarly named event, application.provision.group_push.mapping.update.or.delete.failed, when this event is fired the corresponding action that triggered it will not be retried by Okta and may indicate a configuration problem. For example, invalid authorization credentials with the target application due to an expired password or invalid access token.

app event-hook-eligible

Since: 2017.29

application.provision.group_push.push_memberships

Group push pushed memberships.

app

Since: 2017.29

application.provision.group_push.pushed

application.provision.group_push.removed

A group was removed from an app.

app

Since: 2017.29

application.provision.group_push.updated

A group was updated in an app.

app

Since: 2017.29

application.provision.integration.call_api

Application integration API called.

app-api

Since: 2016.15

application.provision.user.activate

Activate user's application membership.

app-api

Since: 2016.14

application.provision.user.deactivate

Push user deactivation to external application.

app-api

Since: 2016.14

application.provision.user.deprovision

Deprovision user from external application.

app

app-api

Since: 2017.33

application.provision.user.import_profile

Import profile from external application.

app-api

Since: 2017.33

application.provision.user.password

Issue pushing user password to external application.

app-api

Since: 2017.33

application.provision.user.push

Push new user to external application.

app-api

Since: 2016.14

application.provision.user.push_okta_password

Push user's Okta password to application.

app

Since: 2016.14

application.provision.user.push_password

Push user's password to application.

app

Since: 2016.14

application.provision.user.reactivate

Push user reactivation in external application.

app-api

Since: 2016.14

application.provision.user.sync

Sync user in external application.

app event-hook-eligible

Since: 2016.14

application.provision.user.verify_exists

Verify user exists in external application.

app-api

Since: 2016.14

application.registration_policy.lifecycle.create

Create registration policy.

app

Since: 2017.52

application.registration_policy.lifecycle.update

Update registration policy.

app

Since: 2017.52

application.user_membership.add

application.user_membership.approve

User approved for application (assigned by not provisioned).

`user-provision`

Since: 2016.33

application.user_membership.change_password

Change application password for user.

`app` `event-hook-eligible`

Since: 2016.11

application.user_membership.change_username

Change user's application username.

`app`

Since: 2016.02

application.user_membership.deprovision

User deprovisioned from application (was previously revoked).

`user-provision`

Since: 2016.33

application.user_membership.provision

User provisioned to application (was previously approved).

`user-provision`

Since: 2016.33

application.user_membership.remove

Remove user's application membership.

`event-hook-eligible` `user-provision`

app

Since: 2016.02

application.user_membership.restore_password

Restore user's password for an application.

app

Since: 2016.02

application.user_membership.revoke

User revoked from application (unassigned but not yet deprovisioned).

user-provision

Since: 2016.33

application.user_membership.show_password

Show user's password for application.

app

Since: 2016.02

application.user_membership.update

Updated user application property.

app event-hook-eligible

Since: 2016.02

certification.campaign.close

Triggered when a campaign is closed. This event can be used by admins to audit Access Certification Review activity to understand when a Campaign Instance has transitioned into the closed state. This event is triggered when a Campaign is closed either by an admin or on the configured campaign end date.

certification event-hook-eligible

customizable context settings changes in the system log as well as reports. This is triggered when the customizable context settings are updated at the org level.

`certification` `event-hook-eligible`

Since: 2024.06.1

certification.campaign.create

Triggered when a new campaign is created. Can be used to audit campaign activity in the system log as well as reports. This is triggered by creating a new campaign.

`certification`

Since: 2021.07.1

certification.campaign.delete

Triggered when a campaign is deleted. Can be used to audit campaign activity in the system log as well as reports. This is triggered by deleting a campaign.

`certification`

Since: 2021.07.1

certification.campaign.item.decide

Triggered when a decision on the access to a resource is made. Can be used to audit the decision activity related to an item in a certification campaign, such as the access of a user to an application. The outcome.result field will be SUCCESS for a decision to approve or revoke and will be SKIPPED for a decision to delegate. This is triggered when a reviewer makes a decision on a campaign item, or at the end of a campaign if an item has not been reviewed. The result of the decision is included in the debugData (APPROVE, REVOKE, DELEGATE, NORESPONSE).

`certification` `event-hook-eligible`

Since: 2021.10.1

certification.campaign.item.remediate

certification.campaign.launch

Triggered when a campaign is launched. This event can be used by admins to audit Access Certification Review activity to understand when a Campaign Instance has transitioned into the Active state. This event is triggered when a Campaign starts and moves from scheduled to active.

`certification` `event-hook-eligible`

Since: 2022.02.1

certification.campaign.update

Triggered when a campaign is updated. Can be used to audit campaign activity in the system log as well as reports. This is triggered by updating a campaign.

`certification`

Since: 2021.07.1

certification.remediation.open

Triggered when the remediation state is open. Can be used to audit remediation activity in the system log as well as reports. This is triggered when the remediation state is open.

`certification`

Since: 2021.07.1

core.concurrency.org.limit.violation

Too many requests in flight.

`concurrency-limit`

Since: 2017.39

core.el.evaluate

Evaluate Expression Language.

`okta-el`

Since: 2017.20

credential.register

Fired when a credential is registered. This event fires when the registration of a credential is successful or fails. This can be used to audit that a credential has been successfully registered, and troubleshoot why a credential registration attempt has failed.

`user-factor`

Since: 2019.02.3

credential.revoke

Fired when a credential is revoked. This event fires when the revocation of a credential is successful or fails. This can be used to audit that a credential has been successfully revoked, and troubleshoot why a credential revocation attempt has failed.

`user-factor`

Since: 2019.02.3

device.assurance.policy.add

Add device assurance policy. Use this event to monitor when a device assurance policy is created. The name and platform of the new policy are included in the event.

`device-identity` `oie-only`

Since: 2024.08.3

device.assurance.policy.delete

Delete device assurance policy. Use this event to monitor when a device assurance policy is deleted. The name of the deleted policy is included in the event.

`device-identity` `oie-only`

Since: 2024.08.3

device.assurance.policy.update

device.check.add

Add device check. Use this event to monitor when a custom device check is created. The platform, name, variable name, description, and query of the new device check are included in the event.

`device-identity` `oie-only`

Since: 2024.09.0

device.check.delete

Delete device check. Use this event to monitor when a device check is deleted. The name of the deleted device check is included in the event.

`device-identity` `oie-only`

Since: 2024.09.0

device.check.update

Update device check. Use this event to monitor when a device check is updated, and what changed. The details of what is changed in the device check are included in the event.

`device-identity` `oie-only` `changeDetails`

Since: 2024.09.0

device.custom_push.send_notification

Fired when a Push notification sent to a device for custom app. Used to log success and failure for the push notifications with relevant information to allow org developers to troubleshoot push configurations for custo push authenticator. Note that this event is fired whenever a Push is sent.

`custom-push`

Since: 2022.05.3

device.desktop_mfa.configuration.update

Fired when a Desktop MFA configuration is updated by an admin. Admin can monitor who update the Desktop MFA configuration value. More details of configuration update in Target.changeDetails.

Fired when a Desktop MFA enrollment is registered to Okta Server. Admin can monitor which Okta user and device has enrolled with Desktop MFA. The registration happens after a user logs in a device with an online factor.

`device-mfa` `oie-only`

Since: 2024.08.0

device.desktop_mfa.recovery_pin.generate

Fired when a device recovery PIN is generated by an admin. Admin can monitor who generates a device recovery PIN for which user and device. The event is fired even when the generation fails.

`device-mfa` `oie-only`

Since: 2024.08.0

device.desktop_mfa.recovery_pin.rotate_secret

Fired when a device rotates the device recovery PIN secret for Desktop MFA to Okta server. Admin can monitor if a rotation happens for the device recovery PIN secret of a user on a device. The rotation is supposed to happen every 7 days for each user on each device.

`device-mfa` `oie-only`

Since: 2024.08.0

device.enrollment.create

Enroll new device. This can be used by any admin to monitor when a new device is registered successfully for Okta Verify. The user must have below the max allowed devices and a valid device status (not suspended or deactivated).The targets field contains key details of the enrolled device including name, status, serialNumber, imei, meid, osVersion, osPlatform. which may be useful for identifying the device, tracking which device platforms and OS versions that enrolled in Okta Device Authenticator.

`device-identity` `event-hook-eligible` `oie-only` `user`

Since: 2020.10.4

device.integration.endpoint_security.activate

Since: 2024.08.0

device.integration.endpoint_security.deactivate

Triggered when an admin deactivates an endpoint security device integration configuration. You can use the event to audit endpoint security device integration configuration status change. When triggered, the endpoint security device integration configuration has been deactivated for a device platform and the endpoint security device integration signals will be not be requested from devices.

`device-identity` `oie-only` `user`

Since: 2024.08.0

device.lifecycle.activate

Activate device. You can use the event to audit device status change. When triggered, the device can be suspended or deactivated. Also, a user can access protected resources from an active device if permitted by the App Sign-On policies applied to the resources.

`device-identity` `event-hook-eligible` `oie-only` `user`

Since: 2021.07.1

device.lifecycle.deactivate

Deactivate device. You can use the event to audit device status change. When a device is deactivated, it ca not be associated with any Okta Verify factor in the future.

`device-identity` `event-hook-eligible` `oie-only` `user`

Since: 2021.07.1

device.lifecycle.delete

Delete device. You can use the event to audit device status change. When triggered, the device no longer appears in the Admin Console.

`device-identity` `event-hook-eligible` `oie-only` `user`

Since: 2021.07.1

and therefore cannot be suspended.

`device-identity` `event-hook-eligible` `oie-only` `user`

Since: 2021.07.1

device.lifecycle.unsuspend

Unsuspend device. You can use the event to audit device status change. When triggered, all Okta Verify factors associated with the device are unsuspended, and users can access protected resources from the device.

`device-identity` `event-hook-eligible` `oie-only` `user`

Since: 2021.07.1

device.local_account.create

Fired when a user creates a local account on a device backed by Okta credentials. Will allow an admin to identify and audit which Okta users are creating local accounts on registered Okta device using Just-In-Time (JIT) local account creation feature. Note that the event is fired even when the account creation is unsuccessful.

`device-sso` `oie-only`

Since: 2024.06.2

device.password_sync.authentication

Fired when the OS tries to sync a local account password with an Okta password. Can be used to audit tha a credential has been successfully registered, and troubleshoot why a credential registration attempt has failed. Note that the event is fired even when the password sync is unsuccessful.

`device-sso` `oie-only`

Since: 2023.06.1

device.password_sync.enrollment.create

This event fires when Desktop Password Sync enrollment is successful or fails. Can be used to audit which users enrolled in Desktop Password Sync or troubleshoot why enrollment failed. Note that the event is fired

device.platform.add

Triggered when an admin adds a device management platform. You can use the event to audit device management platform status change. When triggered, the device management platform will be available to the org.

`device-identity` `oie-only` `user`

Since: 2021.07.1

device.platform.delete

Triggered when an admin deletes a device management platform. You can use the event to audit device management platform status change. When triggered, the device management platform no longer appears in the Admin Console.

`device-identity` `oie-only` `user`

Since: 2021.07.1

device.platform.renew

Triggered when a component of the device management platform is renewed, such as a registration authority used during SCEP flows. You can use the event to audit device management platform renewals. For example, auditing if and when a registration authority was renewed in order to continue being used during SCEP flows. This can be triggered automatically by our automated renewal systems when the device management platform component is within the renewal period. The renewed component will appear in the Admin Console.

`device-identity` `oie-only`

Since: 2021.07.1

device.platform.secret_key.reset

Triggered when an admin resets the secret key for a device management platform. You can use the event to audit device management platform secret key change. When triggered, the previous device management platform secret key is no longer valid.

`device-identity` `oie-only` `user`

Since: 2024.08.0

admin can update some fields in the device management platform configuration. Additionally the CA Renewalactivation framework can update RA Configurations or SCEP Challenges.

`device-identity` `oie-only` `user`

Since: 2021.07.1

device.platform_sso.keys.register

A device registered public keys for Platform Single Sign-On (SSO). May be useful to troubleshoot failed PlatformSSO authentications or to identify unexpected key rotations. This event typically occurs as the result of an action in taken in an MDM profile. When new Device PlatformSSO keys are registered, a user must re-enroll into PlatformSSO.

`device-sso` `oie-only`

Since: 2024.08.0

device.push.provider.create

Indicates that a new push notification service has been successfully created. The notification service enables push notification as an authentication option through Okta to a push provider such as the Apple Push Notification service or the Google Firebase Cloud Messaging service. You can use this event to verify when a notification service was created for a custom app. When triggered, a new push notification service appears in the Admin Console.

`oie-only` `push-provider`

Since: 2022.04.3

device.push.provider.delete

Indicates that a push notification service has been deleted. The notification service enables push notificatio as an authentication option through Okta to a push provider such as the Apple Push Notification service or the Google Firebase Cloud Messaging service. You can use this event to verify when a notification service was deleted for a custom app. When triggered, a push notification service is removed from the Admin Console.

`oie-only` `push-provider`

Since: 2022.04.3

service was updated for a custom app. When triggered, a push notification service is updated in the Admin Console.

`oie-only` `push-provider`

Since: 2022.04.3

device.signals.status.timeout

A registered device associated with at least one user session hasn't communicated with Okta within the required time interval. Use this event to find registered devices that have lost communication with Okta. This event contains the device unique identifier in the System Log actor object. You can use this information to find other related events.

See also: Identity Threat Protection with Okta AI Event Types

`device-identity` `oie-only`

Since: 2023.11.0

device.user.add

Add device to user. You can use the event to audit device user association activity. The event is triggered when a user adds a new account in Okta Verify.

`device-identity` `event-hook-eligible` `oie-only` `user`

Since: 2021.07.1

device.user.remove

Remove device from user. You can use the event to audit device user association activity. The device remains in the Universal Directory after the user is removed.

`device-identity` `event-hook-eligible` `oie-only` `user`

Since: 2021.07.1

directory.app_user_profile.bootstrap

Bootstrap application user profile.

Update application user profile.

cvd directory

Since: 2016.12

directory.external.group.membership.add

External API call to add user group membership in a directory. This event audits the directory integration API when it adds a user to a group in a directory. Note that the event is fired even when the API call is unsuccessful.

ad-agent group user

Since: 2024.07.0

directory.external.group.membership.remove

External API call to remove user group membership in a directory. This event audits the directory integration API when it removes a user from a group in a directory. Note that the event is fired even when the API call is unsuccessful.

ad-agent group user

Since: 2024.07.0

directory.linked_object.create

An admin can create a linked object that is related to user profiles. This event may be used to identify when a linked object is created, and who created the linked object. This may be useful for admins to validate why a change in the user profile has happened. While linked object creation does not trigger or happen as a result of another event, it is overall related to custom property update, creation and deletion. This event only indicates the creation of a linked object. See directory.linked_object.delete for deletion of linked objects.

cvd directory

Since: 2022.11.1

directory.linked_object.delete

cvd directory

Since: 2022.11.1

directory.mapping.update

Update universal directory mappings.

cvd directory

Since: 2016.12

directory.non_default_user_profile.create

Create non-default universal directory user profile. This can be used to audit that a new non-default universal directory user profile has been created. When fired, this event contains the name and id of the newly created user profile.

cvd directory

Since: 2019.04.2

directory.user_profile.bootstrap

Bootstrap universal directory user profile.

cvd directory

Since: 2016.12

directory.user_profile.update

Update universal directory user profile directory.user_profile.update.

cvd directory

Since: 2016.12

event_hook.activated

Triggered when an event hook has been activated. Used to notify admins that an event hook has been activated. When triggered, this events contains information about the activated event hook.

Triggered when an event hook has been created. Used to notify admins that an event hook has been created. When triggered, this events contains information about the created event hook.

event-hook

Since: 2019.03.4

event_hook.deactivated

Triggered when an event hook has been deactivated. Used to notify admins that an event hook has been deactivated. When triggered, this events contains information about the deactivated event hook.

event-hook

Since: 2019.03.4

event_hook.deleted

Triggered when an event hook has been deleted. Used to notify admins that an event hook has been deleted. When triggered, this events contains information about the deleted event hook.

event-hook

Since: 2019.03.4

event_hook.delivery

Triggered when an event hook delivery fails. Used to identify when an event hook from Okta is not successfully delivered to the configured endpoint. Note that the event is triggered only when the delivery is unsuccessful.

event-hook

Since: 2019.04.0

event_hook.updated

Triggered when an event hook has been updated. Used to notify admins that an event hook has been updated. When triggered, this events contains information about the updated event hook.

event-hook

Since: 2019.03.4

Since: 2019.03.4

group.application_assignment.add

Add assigned application to group.

`event-hook-eligible` `group`

Since: 2016.06

group.application_assignment.remove

Remove assigned application from group.

`event-hook-eligible` `group`

Since: 2016.05

group.application_assignment.skip_assignment_reconcile

No Description

`group`

Since: 2017.51

group.application_assignment.update

Update assigned application in group.

`event-hook-eligible` `group`

Since: 2016.13

group.lifecycle.create

Create Okta group. This can be used to make sure an Okta group is successfully created. Event fired when an Okta group is successfully created.

`event-hook-eligible` `group`

Since: 2019.11.0

Since: 2019.11.0

group.privilege.grant

Group's admin privilege granted. This can be used to audit the provisioning of admin privileges for groups. When fired, this event contains information about the type of admin privileges the group currently has, and what entity sources the group. The group granted privileges can be an Okta sourced group, and AD-sourced group, or an LDAP-sourced group Related events include: GROUP_PRIVILEGE_REVOKE.

`event-hook-eligible` `group`

Since: 2019.03.0

group.privilege.revoke

Group's admin privilege revoked. This can be used to audit the deprovisioning of admin privileges from groups. When fired, this event indicates the group has no more admin privileges. All of group's privileges were revoked. Related events include: GROUP_PRIVILEGE_GRANT.

`event-hook-eligible` `group`

Since: 2019.03.0

group.profile.update

Okta group profile updated. Events of this type can be used by an IT administrator who wants to trigger an Okta Workflow to provision groups into downstream systems. The utility of the Event type is for Provisioning use cases to downstream systems.A classic example of this is a customer who uses Okta for Office 365 LCM, and wants to push a distribution list from Okta to Office 365.

`event-hook-eligible` `group`

Since: 2021.03.2

group.user_membership.add

Add user to group membership.

`event-hook-eligible` `group`

Since: 2016.02

group.user_membership.rule.add_exclusion

Add user to group membership exclusion rule.

group

Since: 2017.51

group.user_membership.rule.deactivated

No Description

group

Since: 2017.51

group.user_membership.rule.error

group membership rule is in error state.

group

Since: 2017.51

group.user_membership.rule.evaluation

No Description

group

Since: 2017.51

group.user_membership.rule.invalidate

Invalidate group membership rule.

group

Since: 2017.51

group.user_membership.rule.trigger

iam.resourceset.bindings.add

Admin role assignment is created. This event can be used to track and audit when a new admin role assignment is created. When fired this event contains information about the new user or group admin assignments for roles associated with the resource set.

admin-role event-hook-eligible

Since: 2021.02.2

iam.resourceset.bindings.delete

Admin assignment is deleted. This event can be used to track and audit when an admin role assignment is deleted. When fired this event contains information about the deleted user or group admin assignments for roles associated with the resource set.

admin-role event-hook-eligible

Since: 2021.02.2

iam.resourceset.create

Resource set is created. This event can be used to track and audit when a resource set is created. When fired this event contains information about the resources contained in the resource set that is created.

admin-role event-hook-eligible

Since: 2021.02.2

iam.resourceset.delete

Resource set is deleted. This event can be used to track and audit when a resource set is deleted. When fired this event contains information about the resources contained in the resource set that is deleted.

admin-role event-hook-eligible

Since: 2021.02.2

iam.resourceset.resources.add

Resources are added to a resource set. This event can be used to audit the resources added to a resource set. When fired this event contains information about the resources added to the resource set.

Resources are deleted from a resource set. This event can be used to audit the resources deleted from a resource set. When fired this event contains information about the resources deleted from the resource set.

`admin-role` `event-hook-eligible`

Since: 2021.02.2

iam.resourceset.update

Resource set update. Use this event to track and audit when a resource set was updated. This event contains information about the updated name and description of the resource set.

`admin-role` `event-hook-eligible` `changeDetails`

Since: 2024.09.3

iam.role.create

Custom admin role is created. This event can be used to track and audit when a custom admin role is created. When fired this event contains information about the permissions contained in the role that is created.

`admin-role` `event-hook-eligible`

Since: 2021.02.2

iam.role.delete

Custom admin role is deleted. This event can be used to track and audit when a custom admin role is deleted. When fired this event contains information about the permissions contained in the role that is deleted.

`admin-role` `event-hook-eligible`

Since: 2021.02.2

iam.role.permission.conditions.add

Conditions added to a permission in Okta. Use this event to evaluate impact on admin privileges as their scope might be impacted. This event is triggered when a condition is added to a role-based permission in Okta. A condition on a permission allows super admins to implement finer grained authorizations for stricter

iam.role.permission.conditions.delete

Conditions deleted from a permission in Okta. Use this event to evaluate impact on admin privileges as their scope might be impacted. This event is triggered when a condition is deleted from a role-based permission in Okta. A condition on a permission allows super admins to implement finer grained authorizations for stricter security postures. The event can be accompanied with other events for permissions such as iam.role.permissions.delete.

`admin-role` `event-hook-eligible`

Since: 2022.12.0

iam.role.permissions.add

Permissions are added to a custom admin role. This event can be used to audit the permissions added to a custom admin role. When fired this event contains information about the permissions added to the role.

`admin-role` `event-hook-eligible`

Since: 2021.02.2

iam.role.permissions.delete

Permissions are deleted from a custom admin role. This event can be used to audit the permissions deleted from a custom admin role. When fired this event contains information about the permissions deleted from the role.

`admin-role` `event-hook-eligible`

Since: 2021.02.2

iam.role.update

Custom admin role update. Use this event to track and audit when a custom admin role was updated. This event contains information about the updated name and description of the role.

`admin-role` `event-hook-eligible` `changeDetails`

Since: 2024.09.3

Since: 2019.01.2

inline_hook.created

Triggered when an inline hook has been created. Used to notify admins that an inline hook has been created. When triggered, this events contains information about the created inline hook.

`inline-hook`

Since: 2019.01.2

inline_hook.deactivated

Triggered when an inline hook is deactivated. Used to identify when an inline hook lifecycle status was changed to deactivated. When triggered, this events contains information about the deactivated inline hook.

`inline-hook`

Since: 2019.01.2

inline_hook.deleted

Triggered when an inline hook has been deleted. Used to notify admins that an inline hook has been deleted. When triggered, this events contains information about the deleted inline hook.

`inline-hook`

Since: 2019.01.2

inline_hook.executed

Triggered when an inline hook has been executed. Used to notify admins about the outcome of execution of an inline hook. Note that the event is fired when the execution is unsuccessful.

`event-hook-eligible` `inline-hook`

Since: 2019.01.2

inline_hook.response.processed

inline_hook.updated

Triggered when an inline hook has been modified. Used to notify admins that an inline hook has been updated. When triggered, this events contains information about the updated inline hook.

`inline-hook`

Since: 2019.01.2

inline_hook.verified

Triggered when attempting to verify an inline hook. Used to notify admins about the outcome of inline hook endpoint URL verification. Note that the event is fired even when the verification is unsuccessful.

`inline-hook`

Since: 2019.01.2

integration.api_service.lifecycle.authorize

Authorize API service integration. This event is triggered when an admin authorized an OAuth 2.0 service app from the Okta Integration Network (OIN) to access the Okta org (tenant) using Okta management APIs. An API service integration is an integration to an OAuth 2.0 service app available from the Okta Integration Network (OIN).

`app` `integration`

Since: 2023.02.2

integration.api_service.lifecycle.revoke

Revoke API service integration. This event is triggered when an admin revoked API access from an OAuth 2.0 service app to the Okta org. An API service integration is an integration to an OAuth 2.0 service app available from the Okta Integration Network (OIN).

`app` `integration`

Since: 2023.02.2

mim.command.generic.acknowledged

No Description

`mim`

Since: 2016.13

mim.command.generic.cancelled

No Description

`mim`

Since: 2016.13

mim.command.generic.delegated

No Description

`mim`

Since: 2016.13

mim.command.generic.error

No Description

`mim`

Since: 2016.13

mim.command.generic.new

No Description

`mim`

Since: 2016.13

mim.command.generic.notnow

mim.command.ios.acknowledged

No Description

mim

Since: 2016.13

mim.command.ios.cancelled

No Description

mim

Since: 2016.13

mim.command.ios.error

No Description

mim

Since: 2016.13

mim.command.ios.formaterror

No Description

mim

Since: 2016.13

mim.command.ios.new

No Description

mim

Since: 2016.13

mim.createEnrollment.ANDROID

No Description

mim

mim

Since: 2016.39

mim.createEnrollment.OSX

No Description

mim

Since: 2016.39

mim.createEnrollment.UNKNOWN

No Description

mim

Since: 2016.39

mim.createEnrollment.WINDOWS

No Description

mim

Since: 2016.39

mim.streamDevicesAppListCSVDownload

No Description

mim

Since: 2016.39

mim.streamDevicesCSVDownload

No Description

mim

Since: 2016.39

oauth2.as.activated

Authorization server is activated.

oauth2 oauth2-as-lifecycle

Since: 2017.22

oauth2.as.created

Authorization server is created.

oauth2 oauth2-as-lifecycle

Since: 2016.50

oauth2.as.deactivated

Authorization server is deactivated.

oauth2 oauth2-as-lifecycle

Since: 2017.22

oauth2.as.deleted

Authorization server is deleted.

oauth2 oauth2-as-lifecycle

Since: 2016.50

oauth2.as.updated

Authorization server is updated.

oauth2 oauth2-as-lifecycle

Since: 2016.50

oauth2.claim.created

oauth2.claim.deleted

OAuth2 claim is deleted.

`oauth2` `oauth2-claim`

Since: 2016.50

oauth2.claim.updated

OAuth2 claim is updated.

`oauth2` `oauth2-claim`

Since: 2016.50

oauth2.scope.created

OAuth2 scope is created.

`oauth2` `oauth2-scope`

Since: 2016.50

oauth2.scope.deleted

OAuth2 scope is deleted.

`oauth2` `oauth2-scope`

Since: 2016.50

oauth2.scope.updated

OAuth2 scope is updated.

`oauth2` `oauth2-scope`

Since: 2016.50

omm.app.VPN.settings.changed

No Description

`omm`

omm

Since: 2018.01

omm.app.eas.cert_based.settings.changed

No Description

omm

Since: 2018.01

omm.app.eas.disabled

No Description

omm

Since: 2018.01

omm.app.eas.settings.changed

No Description

omm

Since: 2018.01

omm.cma.created

No Description

omm

Since: 2018.01

omm.cma.deleted

No Description

omm

Since: 2018.01

omm.enrollment.changed

No Description

`omm`

Since: 2018.01

org.not_configured_origin.redirection.usage

Using untrusted origin for redirection.

`admin` `org`

Since: 2017.44

pam.ad_connection.create

This event is triggered after an Active Directory Connection is created for discovering servers.

`pam`

Since: 2022.02.0

pam.ad_connection.delete

This event is triggered after an Active Directory Connection is deleted.

`pam`

Since: 2022.02.0

pam.ad_connection.update

This event is triggered after an Active Directory Connection is updated.

`pam`

Since: 2022.02.0

pam.ad_task_settings.create

pam.ad_task_settings.delete

This event is triggered after settings that are related to discovering servers in an Active Directory connection are deleted.

pam

Since: 2022.02.0

pam.ad_task_settings.update

This event is triggered after settings that are related to discovering servers in an Active Directory connection are updated.

pam

Since: 2022.02.0

pam.ad_task_settings.update_schedule

This event is triggered after the schedule for discovering Active Directory servers is updated.

pam

Since: 2022.02.0

pam.ad_user_sync_task_settings.activate

This event is triggered after the settings for discovering Active Directory users in an Active Directory connection is activated. Use this event to monitor activation of AD User Sync Task Settings objects. This event contains reference to an AD User Sync Task Settings object.

pam

Since: 2023.06.1

pam.ad_user_sync_task_settings.create

This event is triggered after settings that are related to discovering users in an Active Directory connection are created. Use this event to monitor creation of AD User Sync Task Settings objects. This event contains reference to an AD User Sync Task Settings object.

This event is triggered after the settings for discovering Active Directory users in an Active Directory connection is deactivated. Use this event to monitor deactivation of AD User Sync Task Settings objects. This event contains reference to an AD User Sync Task Settings object.

pam

Since: 2023.06.1

pam.ad_user_sync_task_settings.delete

This event is triggered after the settings for discovering Active Directory users in an Active Directory connection is deleted. Use this event to monitor deletion of AD User Sync Task Settings objects. This event contains reference to an AD User Sync Task Settings object.

pam

Since: 2023.06.1

pam.ad_user_sync_task_settings.update

This event is triggered after settings that are related to discovering users in an Active Directory connection are updated. Use this event to monitor update of AD User Sync Task Settings objects. This event contains reference to an AD User Sync Task Settings object.

pam

Since: 2023.06.1

pam.ad_user_sync_task_settings.update_schedule

This event is triggered after the schedule for discovering Active Directory users in an Active Directory connection is updated. Use this event to monitor schedule update of AD User Sync Task Settings objects. This event contains reference to an AD User Sync Task Settings object.

pam

Since: 2023.06.1

pam.apikey.delete

This event is triggered after a service user's API key is deleted.

This event is triggered after a service user's API key is rotated.

`pam`

Since: 2022.02.0

pam.auth_token.issue

This event is triggered when an ASA client has been authenticated and is issued an authentication token with elevated capabilities.

`pam`

Since: 2022.02.0

pam.billing_contact.create

This event is triggered after a billing contact is created for an ASA team. This event is only applicable to legacy ASA customers.

`pam`

Since: 2022.02.0

pam.client.assign

This event is triggered after an ASA client is assigned to an ASA user.

`pam`

Since: 2022.02.0

pam.client.enroll

This event is triggered after an ASA client is enrolled with ASA.

`pam`

Since: 2022.02.0

pam.client.remove

This event is triggered after an ASA client is removed from a team.

This event is triggered after the state of an ASA client is updated.

pam

Since: 2022.02.0

pam.client_enrollment_policies.create

This event is triggered after an ASA client enrollment policy is created.

pam

Since: 2022.02.0

pam.client_enrollment_policies.delete

This event is triggered after an ASA client enrollment policy is deleted.

pam

Since: 2022.02.0

pam.client_enrollment_policies.update

This event is triggered after an ASA client enrollment policy is updated.

pam

Since: 2022.02.0

pam.client_enrollment_policy_token.delete

This event is triggered after an ASA client enrollment token is deleted.

pam

Since: 2022.02.0

pam.client_enrollment_policy_token.rotate

This event is triggered after an ASA client enrollment token is rotated.

pam

Since: 2022.02.0

pam.cloud_account.delete

This event is triggered after a cloud account has been removed from a project.

pam

Since: 2022.02.0

pam.cloud_account.update

This event is triggered after a cloud account, which is used for importing servers into ASA, is updated.

pam

Since: 2022.02.0

pam.entitlement_sudo.add_to_project

This event is triggered after a sudo entitlement object is added to a project.

pam

Since: 2022.02.0

pam.entitlement_sudo.create

This event is triggered after a sudo entitlement object is created.

pam

Since: 2022.02.0

pam.entitlement_sudo.remove

This event is triggered after a sudo entitlement object is removed.

pam

Since: 2022.02.0

pam.entitlement_sudo.remove_from_project

pam.entitlement_sudo.update

This event is triggered after a sudo entitlement object is updated.

pam

Since: 2022.02.0

pam.gateway.create

This event is triggered after an ASA gateway is created.

pam

Since: 2022.02.0

pam.gateway.delete

This event is triggered after an ASA gateway is deleted.

pam

Since: 2022.02.0

pam.gateway.setup_token.create

This event is triggered after a gateway setup token is created.

pam

Since: 2022.02.0

pam.gateway.setup_token.delete

This event is triggered after a gateway setup token is deleted.

pam

Since: 2022.02.0

pam.gateway.setup_token.update

This event is triggered after a gateway setup token is updated.

pam

pam

Since: 2022.02.0

pam.gateway_creds.issue

This event is triggered after the gateway issues credentials for a server.

pam

Since: 2022.02.0

pam.group.bulk_membership_change

This event is triggered after the members belonging to an ASA group were updated in bulk by a SCIM driver.

pam

Since: 2022.02.0

pam.group.create

This event is triggered after an ASA group is created.

pam

Since: 2022.02.0

pam.group.delete

This event is triggered after an ASA group is deleted.

pam

Since: 2022.02.0

pam.incoming_federation.approve

This event is triggered when an ASA team admin from another team has approved a request to federate identities identities from their team to this team. Only applicable to legacy ASA customers.

pam

Since: 2022.02.0

Since: 2022.02.0

pam.member.add

This event is triggered after a user is added to an ASA group.

pam

Since: 2022.02.0

pam.member.remove

This event is triggered after a user is removed from an ASA group.

pam

Since: 2022.02.0

pam.offline_disabled_event

This event is triggered after disconnected mode is disabled for a group.

pam

Since: 2022.02.0

pam.offline_enabled_event

This event is triggered after disconnected mode is enabled for a group.

pam

Since: 2022.02.0

pam.offline_group.secrets.rotate

This event is triggered after disconnected mode credentials are rotated for a group.

pam

Since: 2022.02.0

pam.outgoing_federation.approve

pam.password.change

This event is triggered after a user password changed. This event is only applicable to legacy ASA customers.

pam

Since: 2022.02.0

pam.password.reset

This event is triggered after a user password reset request is submitted. This event is only applicable to legacy ASA customers.

pam

Since: 2022.02.0

pam.permission.change

This event is triggered after group permissions are updated.

pam

Since: 2022.02.0

pam.preauthorization.create

This event is triggered after a preauthorization is created.

pam

Since: 2022.02.0

pam.preauthorization.update

This event is triggered after a preauthorization is updated.

pam

Since: 2022.02.0

pam

Since: 2024.10.1

pam.privileged_accounts.create

Create a privileged account in Okta Privileged Access. Use this event to verify when a privileged account has successfully created in Okta Privileged Access. The creation request can only be initiated via Universal Directory. For Universal Directory accounts, the outcome result can be SUCCESS or FAILURE. For third-party app privileged accounts, the outcome result can be SUCCESS, FAILURE, or DEFERRED.

pam

Since: 2024.10.1

pam.privileged_accounts.delete

Delete a privileged account in Okta Privileged Access. Use this event to verify when a privileged account has successfully deleted in Okta Privileged Access. The deletion request can only be initiated via Universal Directory. The outcome result can be SUCCESS or FAILURE.

pam

Since: 2024.10.1

pam.privileged_accounts.password.reveal

Reveal password for a privileged account in Okta Privileged Access. Use this event to identify for which account the Okta Privileged Access user revealed the password. Contains the details on the user access method the user used to reveal the password.

pam

Since: 2024.10.1

pam.privileged_accounts.password.update

Update password for a privileged account in Okta Privileged Access. Use this event to identify for which account the Okta Privileged Access user updated the password. Contains the details on the user access method the user used to update the password.

Indicates an automatic password rotation completion event. Use this event to determine the final status of a password rotation for a given privileged account. The outcome result can be SUCCESS, FAILURE or DEFERRED, based on settings and retry mechanisms in Okta Privileged Access.

pam

Since: 2024.10.1

pam.privileged_accounts.password_rotation.start

Initiate an automatic password rotation based on Okta Privileged Access's Resource Group Project settings. Use this event to determine when an automatic password rotation for a given privileged account has begun. The outcome result can be SUCCESS or FAILURE.

pam

Since: 2024.10.1

pam.privileged_accounts.update

Update a privileged account's details in Okta Privileged Access. Use this event to verify when a privileged account has successfully updated in Okta Privileged Access. The update request can only be initiated via Universal Directory. The outcome result can be SUCCESS or FAILURE.

pam

Since: 2024.10.1

pam.project.add_group

This event is triggered after a group is added to a project.

pam

Since: 2022.02.0

pam.project.create

This event is triggered after a Project is created. For ASA, this event only contains the Project name. For Okta Privileged Access, this event contains the Project name and the associated Resource Group.

pam

Privileged Access, this event contains the Project name and the associated Resource Group.

pam

Since: 2022.02.0

pam.project.remove_group

This event is triggered after a group is removed from a project.

pam

Since: 2022.02.0

pam.project.update

This event is triggered after a Project is updated. Only applicable for Okta Privileged Access. This event contains the Project name and the associated Resource Group.

pam

Since: 2023.04.0

pam.project_group_selector.update

This event is triggered after server selectors for a group assigned to a project are updated.

pam

Since: 2022.02.0

pam.resource.checkin.end

This event is triggered when a resource's checkin process completes or fails to complete. Monitor 'FAILED' outcomes of this event to identify resources that may be unavailable for checkout due to an incomplete checkin. This event contains details of the original checkout and, if a failure occurred, the reason why the checkin failed.

pam

Since: 2024.05.1

pam.resource.checkin.start

pam.resource.checkout

This event is triggered when a resource is checked out. Use this event to identify the exclusive access to resources. This event contains details of the resource and the user who checked it out.

`pam`

Since: 2024.05.1

pam.resource_group.create

This event is triggered after a Resource Group is created. Monitor this event to be notified when new teams in your Okta org begin using Okta Privileged Access. Only applicable for Okta Privileged Access. This event defines when a Resource Administrator has created a new Resource Group to manage resources.

`pam`

Since: 2023.04.0

pam.resource_group.delete

This event is triggered after a Resource Group is deleted. Monitor this event to be notified when a team in your Okta org stops managing access to a resource. Only applicable for Okta Privileged Access. This event defines when a Resource Administrator deleted a Resource Group.

`pam`

Since: 2023.04.0

pam.resource_group.update

This event is triggered after a Resource Group is updated. Monitor this event to be notified when Resource Group settings change. Only applicable for Okta Privileged Access. This event defines when a Resource Administrator has modified the settings for a Resource Group.

`pam`

Since: 2023.04.0

the name of the related Secret and an Actor. The Actor is the User that created the Secret.

pam

Since: 2023.12.0

pam.secret.delete

This event is triggered when a Secret, such as a password, stored in the Okta Privileged Access Vault is deleted. Use this event to identify the deletion of an existing Secret. For example, deleting a Secret at an unusual time or outside of a standard process may be of interest to security analysts. Each event of this type references the name of the related Secret and an Actor. The Actor is the User that deleted the Secret.

pam

Since: 2023.12.0

pam.secret.reveal

This event is triggered when the contents of a Secret, such as a password, is revealed to a user. Use this event to identify the access of a Secret. For example, accessing a Secret at an unusual time or outside of a standard process may be of interest to security analysts. Each event of this type references the name of the related Secret and an Actor. The Actor is the User that revealed the Secret.

pam

Since: 2023.12.0

pam.secret.update

This event is triggered when a Secret, such as a password, stored in the Okta Privileged Access Vault is updated. Use this event to identify an update to an existing Secret. For example, updating a Secret at an unusual time or outside of a standard process may be of interest to security analysts. Each event of this type references the name of the related Secret and an Actor. The Actor is the User that updated the Secret.

pam

Since: 2023.12.0

pam.secret_folder.create

pam

Since: 2023.12.0

pam.secret_folder.delete

This event is triggered after a Secret Folder is deleted. Secret Folders are containers used to organize and store Secrets. Use this event to identify the deletion of an existing Secret Folder. For example, deleting a Secret Folder at an unusual time or outside of a standard process may be of interest to security analysts. Each event of this type references the name of the related Secret Folder and an Actor. The Actor is the User that deleted the Secret Folder.

pam

Since: 2023.12.0

pam.secret_folder.update

This event is triggered after a Secret Folder is updated. Secret Folders are containers used to organize and store Secrets. Use this event to identify an update to an existing Secret Folder. For example, updating a Secret Folder at an unusual time or outside of a standard process may be of interest to security analysts. Each event of this type references the name of the related Secret Folder and an Actor. The Actor is the User that updated the Secret Folder.

pam

Since: 2023.12.0

pam.security_policy.create

This event is triggered after a Security Policy is created. Use this event to determine when Security Administrators create new Security Policies. Only applicable for Okta Privileged Access. This event contains the Principals associated with the Security Policy and the number of rules in the policy.

pam

Since: 2023.04.0

pam.security_policy.delete

Since: 2023.04.0

pam.security_policy.evaluate

This event is triggered when an operation requires a Security Policy evaluation. Use this event to understand how Security Policies are utilized to control access to resources. Currently, this event is only triggered when a user isn't authorized to perform an operation due to existing Security Policies.

pam

Since: 2023.12.0

pam.security_policy.update

This event is triggered after a Security Policy is updated. Use this event to determine when Security Administrators update Security Policies and to identify important changes made to policies. Only applicable for Okta Privileged Access. This event contains the Principals associated with the Security Policy and the number of rules in the policy.

pam

Since: 2023.04.0

pam.server.enroll

This event is triggered after a server running the Okta ASA agent has enrolled with ASA.

pam

Since: 2022.02.0

pam.server.reassign

This event is triggered after a server is reassigned from one project to another.

pam

Since: 2022.02.0

pam.server.remove

pam.server.ssh_login

This event is triggered after a user performs an SSH login to a server.

pam

Since: 2022.02.0

pam.server_account.discovered

This event is triggered after a server account is first discovered by the Server Agent. Only applicable for Okta Privileged Access. This event contains the name of the discovered account and the associated server.

pam

Since: 2023.04.0

pam.server_account.password_change.initiated

This event is triggered after a password rotation is requested for a local server account. Use this event to verify that the password settings are being correctly applied to your servers. This event contains the name of the local server account being modified and the associated server.

pam

Since: 2023.04.0

pam.server_account.password_change.out_of_band

This event is triggered after a server account password is altered via a method other than scheduled rotation. You MUST monitor this event to ensure that unauthorized users are not attempting to reset local server account passwords in an attempt to gain access to servers. Only applicable for Okta Privileged Access. This event contains the modified server account and the associated server.

pam

Since: 2023.04.0

pam.server_account.password_change.update

This event is triggered after a server reports an attempt to perform a password rotation. The outcome.result field contains either 'SUCCESS' or 'FAILURE' and should be monitored to detect any password rotation errors.

pam.server_account.update

This event is triggered after a discovered server account is updated. Use this event to observe how often the system updates server accounts. Only applicable for Okta Privileged Access. This event contains the name of the updated account and the associated server.

`pam`

Since: 2023.04.0

pam.server_labels.update

This event is triggered after server labels are updated.

`pam`

Since: 2022.02.0

pam.service.create

This event is triggered after a service bound to a service user is created on a server.

`pam`

Since: 2022.02.0

pam.service.remove

This event is triggered after a service is removed from a server.

`pam`

Since: 2022.02.0

pam.sudo_command_bundle.create

This event is triggered after a sudo command bundle is created. Use this event to determine when Resource Administrators create a new Sudo Command Bundle. Only applicable for Okta Privileged Access. This event defines when a Resource Administrator has created a new sudo command bundle.

`pam`

Since: 2023.06.1

pam

Since: 2023.06.1

pam.sudo_command_bundle.update

This event is triggered after a sudo command bundle is updated. Use this event to determine when Resource Administrators update an existing Sudo Command Bundle. Only applicable for Okta Privileged Access. This event defines when a Resource Administrator has updated a new sudo command bundle.

pam

Since: 2023.06.1

pam.team.create

This event is triggered after a team is created in ASA.

pam

Since: 2022.02.0

pam.team_group_attribute.create

This event is triggered after team-level group attributes are created.

pam

Since: 2022.02.0

pam.team_group_attribute.delete

This event is triggered after team-level group attributes are deleted.

pam

Since: 2022.02.0

pam.team_group_attribute.update

This event is triggered after team-level group attributes are updated.

pam

customers.

pam

Since: 2022.02.0

pam.team_project_group_attribute.create

This event is triggered after project-level group attribute overrides are created.

pam

Since: 2022.02.0

pam.team_project_group_attribute.delete

This event is triggered after project-level group attribute overrides are deleted.

pam

Since: 2022.02.0

pam.team_project_group_attribute.update

This event is triggered after project-level group attribute overrides are updated.

pam

Since: 2022.02.0

pam.team_project_user_attribute.create

This event is triggered after project-level user attribute overrides are created.

pam

Since: 2022.02.0

pam.team_project_user_attribute.delete

This event is triggered after project-level user attribute overrides are deleted.

pam

Since: 2022.02.0

pam.team_settings.update

This event is triggered after team settings are updated.

pam

Since: 2022.02.0

pam.team_user_attribute.create

This event is triggered after team-level user attributes are created.

pam

Since: 2022.02.0

pam.team_user_attribute.delete

This event is triggered after team-level user attributes are deleted.

pam

Since: 2022.02.0

pam.team_user_attribute.update

This event is triggered after team-level user attributes are updated.

pam

Since: 2022.02.0

pam.unbound_client.enroll

This event is triggered after an ASA client is enrolled by using the 'sft fleet enroll' command.

pam

Since: 2022.02.0

pam.unmanaged_server.create

pam.user.create

This event is triggered after a user is created in ASA.

`pam`

Since: 2022.02.0

pam.user.remove

This event is triggered after a user is removed from ASA.

`pam`

Since: 2022.02.0

pam.user.update

This event is triggered after a user is updated in ASA.

`pam`

Since: 2022.02.0

pam.user_creds.issue

This event is triggered when an Okta user is authorized and initiates a connection to a server protected by Okta.

`pam`

Since: 2022.02.0

personal.admin.configuration.update

Okta personal app migration flag is updated. Triggered when app migration is updated on an Org. This event is fired when admin toggle app migration in org.

`okta-personal`

Since: 2024.02.8

Since: 2024.02.8

pki.ca.add

Triggered when an admin creates an Okta CA (ROOT or Intermediate certs) or uploads a 3rd party certificate chain. You can use the event to audit the Okta CA or 3rd party certificate authority status change. When triggered, the Okta CA or 3rd party certificate authority will appear in the Admin Console.

`device-identity` `oie-only` `user`

Since: 2021.07.1

pki.ca.delete

Triggered when an admin deletes a 3rd party certificate chain. You can use the event to audit the 3rd party certificate authority status change. When triggered, the 3rd party certificate authority is no longer available to the org.

`device-identity` `oie-only` `user`

Since: 2021.07.1

pki.ca.renew

Triggered when one or more certificates that belong to a certificate authority are renewed. Use to audit certificate renewals that belong to a certificate authority. You can also use it as a notification to download th renewed certificates. When triggered, this event includes the old certificates and the new certificate replacements.

`device-identity` `oie-only`

Since: 2024.05.1

pki.cert.bind

Triggered when a certificate is bound to a device. You can use the event to audit certificate device binding relationship. When triggered, the device appears in the Admin Console as managed device.

`device-identity` `oie-only` `user`

Since: 2021.09.1

is up and running properly and has not been changed by the issuing Certificate Authority (CA). When fired, this event will include the URL of the CRL that is having an issue along with a corresponding HTTP error code.

`device-identity` `oie-only`

Since: 2023.07.2

pki.cert.issue

Device Trust certificate issuance.

`device-trust-cert-distribution-and-binding`

Since: 2017.45

pki.cert.lifecycle.activate

Triggered when a certificate marked as hold is removed from the CRL or when renewed Okta CA certificates marked as inactive are activated. You can use the event to audit certificate lifecycle change. When an admin activates/unsuspends a device, the certificate associated with the device is activated when used in the next Okta Verify flow. Additionally when an admin or activation job activates an inactive certificate it can then be used to issue client certificates in SCEP.

`device-identity` `oie-only` `user`

Since: 2021.09.1

pki.cert.lifecycle.delete

Triggered when a certificate is deleted as a result of an admin deleting the binding device. You can use the event to audit certificate lifecycle change. When triggered, the certificate no longer appears in the Admin Console.

`device-identity` `oie-only` `user`

Since: 2021.09.1

pki.cert.lifecycle.hold

Triggered when a certificate is temporarily on hold and appears on CRL. You can use the event to audit certificate lifecycle change. A certificate on hold can be activated after it is removed from CRL.

Triggered when a certificate is revoked and appears on CRL. You can use the event to audit certificate lifecycle change. Once revoked, a certificates can not be activated.

`device-identity` `oie-only` `user`

Since: 2021.09.1

pki.cert.lifecycle.suspend

Triggered when a certificate is suspended as a result of an admin deactivating the binding device. You can use the event to audit certificate lifecycle change. When triggered, the certificate can not be used to send the management hint.

`device-identity` `oie-only` `user`

Since: 2021.09.1

pki.cert.renew

Triggered when a Device Trust certificate is renewed.

`device-trust-cert-distribution-and-binding`

Since: 2017.45

pki.cert.revoke

Device Trust certificate revocation.

`device-trust-cert-distribution-and-binding`

Since: 2017.45

plugin.downloaded

Plugin downloaded.

`plugin`

Since: 2016.48

plugin.script_status

policy.auth_reevaluate.action

Invocation of a post auth session action. This event is triggered when Okta logs a user out of their configured apps or runs a Workflow in response to an authentication or global session policy violation. This event is triggered when Okta logs a user out of their configured apps or runs a Workflow in response to an authentication or global session policy violation.

See also: Identity Threat Protection with Okta AI Event Types

`policy` `security` `session`

Since: 2024.07.2

policy.auth_reevaluate.enforce

Evaluation of a post auth session. This event is triggered when a post auth session evaluation occurs. This event is triggered when a post auth session evaluation occurs.

See also: Identity Threat Protection with Okta AI Event Types

`policy` `security` `session`

Since: 2024.07.2

policy.auth_reevaluate.fail

Auth policy re-evaluation has occurred and has resulted in a policy violation. Can be used to identify which user, apps, and session were involved in a policy violation event. Event fired when continuing access evaluation results in failure.

See also: Identity Threat Protection with Okta AI Event Types

`event-hook-eligible` `policy` `security` `session`

Since: 2023.09.0

policy.continuous_access.action

Deprecated: Continuous Access policy action invocation. Signal that an action associated with a continuous access policy evaluation has been invoked. Event fired when an action associated with a continuous access

Since: 2023.09.0

policy.continuous_access.evaluate

Deprecated: Evaluation of Continuous Access Policy. Signal that continuous access policy has been evaluated for a session which has failed CAE. Event fired when continuous access policy has been evaluated for a session which has failed CAE. See the event type policy.auth_reevaluate.enforce that replaces this deprecated event.

See also: Identity Threat Protection with Okta AI Event Types

`policy` `security` `session`

Since: 2023.09.0

policy.entity_risk.action

Entity Risk policy action invocation. Signal that an action associated with an entity risk policy evaluation has been invoked. Event fired when an action associated with an entity risk policy evaluation has been invoked.

See also: Identity Threat Protection with Okta AI Event Types

`policy` `security` `session`

Since: 2023.09.0

policy.entity_risk.evaluate

Evaluation of Entity Risk policy. Signal that entity risk policy has been evaluated for an entity for which we have received a risk change event. Event fired when entity risk policy has been evaluated for an entity for which a risk change event was generated.

See also: Identity Threat Protection with Okta AI Event Types

`policy` `security` `session`

Since: 2023.09.0

policy.evaluate_sign_on

resource), and DENY(user is denied from accessing the resource). For Okta Identity Engine (OIE), a single policy.evaluate_sign_on event may include the evaluation result of Okta global session policy and authentication policy. For Okta Classic Engine, the evaluation result of Okta sign-on policy and app sign-on policy will be recorded in individual policy.evaluate_sign_on events.

`policy`

Since: 2017.11

policy.execute.user.start

Start execution of policy for user.

`policy`

Since: 2018.15

policy.lifecycle.activate

Activate policy.

`event-hook-eligible` `policy`

Since: 2016.14

policy.lifecycle.create

Create policy.

`policy`

Since: 2016.14

policy.lifecycle.deactivate

Deactivate policy.

`event-hook-eligible` `policy`

Since: 2016.14

policy.lifecycle.delete

policy.lifecycle.overwrite

Overwrite policy.

`policy`

Since: 2017.45

policy.lifecycle.update

Update policy.

`event-hook-eligible` `policy` `changeDetails`

Since: 2016.14

policy.mapping.create

Create policy mapping. This event is used to audit when a policy is mapped to a resource. This event is fired when a policy is mapped to a resource. The isPreviousPolicy attribute within the Policy Targets' Details denotes whether or not it was the previous or new policy being mapped.

`policy`

Since: 2021.12.0

policy.rule.action.execute

Scheduled execution of policy rule action.

`policy`

Since: 2018.15

policy.rule.activate

Activate policy rule.

`event-hook-eligible` `policy`

Since: 2016.14

policy.rule.add

policy.rule.deactivate

Deactivate policy rule.

`event-hook-eligible` `policy`

Since: 2016.14

policy.rule.delete

Delete policy rule.

`event-hook-eligible` `policy`

Since: 2016.14

policy.rule.invalidate

Invalidate policy rule.

`policy`

Since: 2016.14

policy.rule.update

Update policy rule.

`event-hook-eligible` `policy`

Since: 2016.14

policy.scheduled.execute

Scheduled execution of policy.

`policy`

Since: 2018.15

scheduled_action.user_suspension.canceled

Canceled scheduled user suspension.

`uncategorized`

uncategorized

Since: 2017.32

scheduled_action.user_suspension.scheduled

Scheduled user suspension.

uncategorized

Since: 2017.32

scheduled_action.user_suspension.updated

Updated scheduled user suspension.

uncategorized

Since: 2017.32

security.attack_protection.settings.update

Triggered when settings to protect against password-based attacks are updated. Useful for monitoring potential intrusion if the change was not planned. Covered features include Require possession factor before password during MFA and Block suspicious password attempts from unknown devices.

mfa security changeDetails

Since: 2024.08.0

security.authenticator.lifecycle.activate

Fired when an admin activates an authenticator for the org. This event can be used to identify who activate an authenticator and which authenticator was activated. When fired, this event contains information about the authenticator type that was activated and the actor who activated the authenticator. Authenticator activation occurs when an authenticator is added. Related events include security.authenticator.lifecycle.deactivate.

authenticator event-hook-eligible oie-only

Since: 2020.06.3

authenticator specific information. Authenticator creation occurs when an authenticator is added. Related events include security.authenticator.lifecycle.update.

`authenticator` `event-hook-eligible` `oie-only`

Since: 2022.06.0

security.authenticator.lifecycle.deactivate

Fired when an admin deactivates an authenticator for the org. This event can be used to identify who deactivated an authenticator and which authenticator was deactivated. When fired, this event contains information about the authenticator type that was deactivated and the actor who deactivated the authenticator. Authenticator deactivation occurs when an authenticator is removed. Related events include security.authenticator.lifecycle.activate.

`authenticator` `event-hook-eligible` `oie-only`

Since: 2020.06.3

security.authenticator.lifecycle.update

Fired when an admin updates an authenticator in the org. This event can be used to identify who updated an authenticator and which authenticator was updated. The actor specifies the user that updated the authenticator and the target specifies the authenticator name and the ID. There may be a second target with details of any authenticator method updates. This event could also contain authenticator specific information. Authenticator update occurs when an authenticator is edited. Related events include security.authenticator.lifecycle.create.

`authenticator` `event-hook-eligible` `oie-only`

Since: 2022.06.0

security.behavior.settings.create

Behavior settings create. This can also be used to identify when a behavior setting is created. When fired, this event contains information about a created setting.

`behavior-settings`

Since: 2019.07.0

Since: 2019.07.0

security.behavior.settings.update

Behavior settings update. This can also be used to identify when a behavior setting has been changed. When fired, this event contains information about a updated setting.

`behavior-settings`

Since: 2019.07.0

security.breached_credential.detected

A credential, such as a password, which is associated with a known breach was used during an authentication flow. Used to identify users for whom credential rotation or other risk mitigation is necessary. The actor is the user with the breached credential. For Identity Engine, a target will indicate the specific credential associated with the breach. The outcome for this event will always be SUCCESS with a severity level of WARN. If breached credential protection is enabled, auser.session.clear will also be fired. These two events can be correlated by the Request ID.

`account` `event-hook-eligible` `security` `user`

Since: 2024.04.2

security.device.add_request_blacklist_policy

Added request blacklist to request blacklist policies.

`device` `security`

Since: 2018.08

security.device.remove_request_blacklist_policy

Removed request blacklist from request blacklist policies.

`device` `security`

Since: 2018.08

security.events.provider.activate

Activate a security events provider. Appears when an authorized security events provider, such as the Shared Signals Framework (SSF) transmitter, is activated.This event helps admins troubleshoot issues with the delivery of security events to Okta. When fired, this event contains information about the activated security events provider.

`security`

Since: 2024.08.0

security.events.provider.create

Create a security events provider. Appears when an authorized security events provider, such as the Shared Signals Framework (SSF) transmitter, is created.This event helps admins troubleshoot issues with the delivery of security events to Okta. When fired, this event contains information about the created security events provider.

`security`

Since: 2024.08.0

security.events.provider.deactivate

Deactivate a security events provider. Appears when an authorized security events provider, such as the Shared Signals Framework (SSF) transmitter, is deactivated.This event helps admins troubleshoot issues wi the delivery of security events to Okta. When fired, this event contains information about the deactivated security events provider.

`security`

Since: 2024.08.0

security.events.provider.delete

Delete a security events provider. Appears when an authorized security events provider, such as the Shared Signals Framework (SSF) transmitter, is deleted.This event helps admins troubleshoot issues with the delivery

security.events.provider.receive_event

Appears when a security events provider submits a valid event for each known detection. The event helps admins debug or monitor SSF provider submissions. The event contains debug context data about the provider's risk report.

See also: Identity Threat Protection with Okta AI Event Types

Since: 2022.10.0

security.events.provider.update

Update a security events provider. Appears when an update is made to an authorized security events provider,such as the Shared Signals Framework (SSF) transmitter.This event helps admins troubleshoot issues with the delivery of security events to Okta. When fired, this event contains information about the updated security events provider.

`security`

Since: 2024.08.0

security.events.transmitter.create

Create security events transmitter. Appears when a specific security events transmitter, such as the Shared Signals Framework (SSF) transmitter, is created. This event helps admins troubleshoot issues with event delivery to security event receivers. This event contains configuration details of the created security events transmitter.

`security`

Since: 2024.08.0

security.events.transmitter.delete

Delete security events transmitter. Appears when a specific security events transmitter, such as the Shared Signals Framework (SSF) transmitter, is deleted. This event helps admins troubleshoot issues with event delivery to security events receivers. This event contains configuration details of the deleted security events transmitter.

Update security events transmitter. Appears when there is an update to a specific security events transmitter, such as the Shared Signals Framework (SSF) transmitter. This event helps admins troubleshoot issues with event delivery to security events receivers. This event contains configuration details of the updated security events transmitter.

`security`

Since: 2024.08.0

security.request.blocked

Security request blocked.

`security`

Since: 2018.32

security.session.detect_client_roaming

Roaming session detected for user.

`security` `session`

Since: 2017.28

security.threat.configuration.update

Fired when a ThreatInsight configuration has been updated. This can be used to identify when an existing ThreatInsight configuration has been updated. An update can be updating the action or the excluded zone. When fired, this event contains information about who made the update to the configuration.

`threat-insight-configuration`

Since: 2019.07.0

security.threat.detected

Request from an IP identified as malicious by Okta ThreatInsight. This can be used to monitor and act on credential based attacks (such as Brute Force, Password Spray) on your organization. The reasons why the request was classified as malicious can be found in the outcome.reason field. The outcome.result field will be 'ALLOW', 'DENY' or 'RATE_LIMIT' based on whether Okta Threat Insight is configured in log mode or log and

security.trusted_origin.activate

A trusted origin is activated. When an event is emitted upon the activation of a trusted origin, customers can monitor these events and take remedial action. Event is triggered when a trusted origin is activated.

`event-hook-eligible` `trusted-origins` `changeDetails`

Since: 2024.05.0

security.trusted_origin.create

A trusted origin is created. When an event is emitted upon the creation of a trusted origin, customers can monitor these events and take remedial action. Event is triggered when a trusted origin is created.

`event-hook-eligible` `trusted-origins`

Since: 2024.05.0

security.trusted_origin.deactivate

A trusted origin is deactivated. When an event is emitted upon the deactivation of a trusted origin, customers can monitor these events and take remedial action. Event is triggered when a trusted origin is deactivated.

`event-hook-eligible` `trusted-origins` `changeDetails`

Since: 2024.05.0

security.trusted_origin.delete

A trusted origin is deleted. When an event is emitted upon the deletion of a trusted origin, customers can monitor these events and take remedial action. Event is triggered when a trusted origin is deleted.

`event-hook-eligible` `trusted-origins`

Since: 2024.05.0

security.trusted_origin.update

A trusted origin is updated. When an event is emitted upon the modification of a trusted origin, customers can monitor these events and take remedial action. Event is triggered when a trusted origin is updated.

Fired when a country has been added to the voice call blacklist. This can be used to identify when a country has been blacklisted for voice call. When fired, this event contains information about the country that was added to the blacklist.Related events include security.voice.remove_country_blacklist.

`security` `voice`

Since: 2019.03.3

security.voice.remove_country_blacklist

Fired when a country has been removed from the voice call blacklist. This can be used to identify when a country has been removed from voice call blacklist. When fired, this event contains information about the country that was removed from the blacklist.Related events include security.voice.add_country_blacklist.

`security` `voice`

Since: 2019.03.3

security.zone.make_blacklist

Added IPs to blacklist zone.

`network-zone` `security`

Since: 2017.06

security.zone.remove_blacklist

Removed IPs from blacklist zone.

`network-zone` `security`

Since: 2017.06

self_service.disabled

Self-service disabled for app.

`self-service`

Since: 2017.48

self_service.enabled

support.org.update

Okta has updated the configuration or data within the Org. This can be used to identify modifications to an Org which are the result of an action by an Okta staff member. Such actions are typically taken in response to a customer request, such as request to enable an Early Access feature. In some cases, these actions may be the result of a review initiated by Okta, such as a review in response to a production service alert. See the supportAction object within the debugContext.debugData objection for more information about the type of update.

`support-audit`

Since: 2022.06.2

support.org.view

Okta has viewed a page which contains customer data. This can be used to identify an action taken by an Okta staff member in the support tool which resulted in a view of customer data. Such actions are typically taken in response to a customer request, such as in the process of investigating an issue raised through a support case. In some cases, these actions may be the result of a review initiated by Okta, such as a review in response to a production service alert. See the supportAction object within the debugContext.debugData objection for more information about the type of update.

`support-audit`

Since: 2022.06.2

system.agent.ad.config_change_detected

A monitored variable in an AD agent configuration file has changed. This can be used to audit that a customer's AD agent configuration file has changed. This event occurs when a monitored variable in an AD agent configuration file has changed.

`ad-agent` `changeDetails`

Since: 2024.09.0

system.agent.ad.connect

Connect AD agent to Okta.

`ad-agent`

ad-agent

Since: 2016.20

system.agent.ad.deactivate

Deactivate AD agent.

ad-agent

Since: 2016.20

system.agent.ad.delete

Delete AD agent.

ad-agent

Since: 2016.20

system.agent.ad.import_ou

Perform import OU by AD agent.

ad-agent

Since: 2016.20

system.agent.ad.import_user

Perform import user by AD agent.

ad-agent

Since: 2016.20

system.agent.ad.invoke_dir

Perform directory invoke command by AD agent.

ad-agent

Since: 2016.20

system.agent.ad.read_config

Perform config read by AD agent.

ad-agent

Since: 2016.20

system.agent.ad.read_dirsync

Perform dirsync read by AD agent.

ad-agent

Since: 2016.20

system.agent.ad.read_ldap

Perform LDAP read by AD agent.

ad-agent

Since: 2016.20

system.agent.ad.read_schema

Perform schema read by AD agent.

ad-agent

Since: 2016.20

system.agent.ad.read_topology

Directory agent performed topology import operation.

ad-agent

Since: 2016.20

system.agent.ad.realtimesync

system.agent.ad.reset_user_password

Perform user password reset by AD agent.

ad-agent

Since: 2016.20

system.agent.ad.start

Start AD agent.

ad-agent

Since: 2016.20

system.agent.ad.unlock_user_account

Perform unlock user account by AD agent.

ad-agent

Since: 2016.20

system.agent.ad.update

Update AD agent configuration.

ad-agent

Since: 2016.20

system.agent.ad.update_user

User Auth and Update.

ad-agent

Since: 2016.20

system.agent.ad.upgrade

Upgrade AD agent.

ad-agent

upload is successful or fails. This can be used to audit that logs files are being fetched successfully, have been uploaded successfully, and troubleshoot why an IWA log upload has failed. When fired, this event indicates whether a log file upload has been successful or failed. This event also indicates whether the event was initiated by the Okta system or a user. Related events: none, all debugging context is included in this event.

`ad-agent`

Since: 2019.02.1

system.agent.ad.upload_log

Upload AD agent log.

`ad-agent`

Since: 2016.20

system.agent.ad.write_ldap

Perform LDAP write by AD agent.

`ad-agent`

Since: 2016.20

system.agent.auto_update

Fired when an individual agent auto-update succeeds or fails. Confirms a successful agent auto-update, o provides troubleshooting information when the agent auto-update is unsuccessful. Indicates when an age auto-update is successful or unsuccessful.

`ad-agent` `agent-pool`

Since: 2021.10.0

system.agent.connector.connect

Connect connector agent to Okta.

`connector-agent`

Since: 2016.20

system.agent.connector.delete

Delete connector agent.

`connector-agent`

Since: 2016.20

system.agent.connector.reactivate

Reactivate connector agent.

`connector-agent`

Since: 2016.20

system.agent.ldap.change_user_password

Perform change user password by LDAP agent.

`ldap-app`

Since: 2016.20

system.agent.ldap.create_user_JIT

Perform create user JIT by LDAP agent.

`ldap-app`

Since: 2016.20

system.agent.ldap.disconnect

Disconnect LDAP agent from Okta.

`ldap-app`

Since: 2016.20

system.agent.ldap.realtimesync

system.agent.ldap.reconnect

Reconnect LDAP agent to Okta.

`ldap-app`

Since: 2016.20

system.agent.ldap.reset_user_password

LDAP agent performed a password reset.

`ldap-app`

Since: 2016.20

system.agent.ldap.unlock_user_account

LDAP agent performed account unlock for User.

`ldap-app`

Since: 2016.45

system.agent.ldap.update_user

Fired when LDAP Delegated Authentication is used to sign in and a user profile is updated. Can be used by admins to identify user profile changes resulting from corresponding changes in the LDAP directory. The previous name for this event was system.agent.ad.update_user.

`ldap-app`

Since: 2021.10.0

system.agent.ldap.update_user_password

Perform update user password by LDAP agent.

`ldap-app`

Since: 2016.20

event can be used to track the deployment and integration of Okta agents across an org's infrastructure. This information can be useful for security audits, compliance reporting, and managing the overall Okta ecosystem.

`agent`

Since: 2024.07.2

system.agent_pools.auto_update

Fired when the status of an agent pool auto-update is changed. Confirms an agent pool auto-update status change and provides troubleshooting information. Indicates when the status of an agent pool auto-update is changed.

`ad-agent` `agent-pool`

Since: 2021.10.0

system.api_token.create

Create API token. This event occurs when a new unscoped API token is generated within the system. The unscoped API token grants authenticated access to the system's API for automated tasks or integration purposes. Event log details include the token ID, the user, or service it was created for, and the time of creation. This information helps maintain a secure API access framework by allowing administrators to track token issuance. Administrators can also enforce least privilege access and promptly identify any unauthorized token creation.

`event-hook-eligible` `token`

Since: 2016.12

system.api_token.enable

Enable API token.

`token`

Since: 2016.12

system.api_token.revoke

system.api_token.update

An API token has been updated. This event can be used to identify a change to an existing API token, such as a change to the applicable rate limits for the token. Details of the change can be found in the debugData. This event does not change whether the token is valid for use, for actions that impact validity see system.api_token.enable and system.api_token.revoke.

`token`

Since: 2022.07.0

system.beta.feature.enable

Fired when an admin has enabled a BETA feature. This can be used to understand the status of the BETA Feature and identify who has enabled it for an org. When fired, this event contains information about the enabled BETA Feature, as well as the admin who enabled it.

`admin` `self-service-feature-management` `system`

Since: 2019.07.1

system.billing.sms_usage_sent

Indicates that a report for SMS usage was sent to the billing system.

`admin` `billing`

Since: 2018.36

system.brand.create

This event is fired when the brand resource is created. Developer and org admins can use this event to identify when the brand resource was created. The event contains information about the created brand.

`admin`

Since: 2023.01.0

system.brand.delete

This event is fired when a brand resource is deleted. Developer and org admins can use this event to identify when a brand resource was deleted. The event contains information about a deleted brand.

This event is fired when the brand resource is updated. Developer and org admins can use this event to identify when the brand resource was updated. The event contains information regarding specific updates made to brand like "customPrivacyPolicyUrl".

`admin`

Since: 2021.08.0

system.captcha.create

A captcha instance is created for Sign-in Widget. Indicates when a captcha instance was created. This event is fired when org admin creates a captcha instance.

`captcha` `system`

Since: 2021.05.1

system.captcha.delete

A captcha instance is deleted. Indicates when a captcha instance was deleted. This event is fired when org admin deletes a captcha instance.

`captcha` `system`

Since: 2021.05.1

system.captcha.update

A captcha instance is updated. Indicates when a captcha instance was updated. This event is fired when org admin updates a captcha instance.

`captcha` `system`

Since: 2021.05.1

system.client.concurrency_rate_limit.notification

Notify when too many requests in flight for client. This can be used to notify whenever there are too many concurrent requests from a client without enforcing any violation. When fired, this event contains information about the request such as client, device and ip details.

`system`

from a client. When fired, this event contains information about the request such as client, device and ip details.

`system`

Since: 2020.06.1

system.client.rate_limit.notification

Notify when client rate limits are exceeded. This can be used to notify whenever a client is exceeding its rate limit without enforcing any violation. When fired, this event contains information about the request such as client, device and ip details.

`system`

Since: 2020.09.4

system.client.rate_limit.violation

Client rate limit violation. This can be used to track if a client is exceeding its rate limit. When fired, this event contains information about the request such as client, device and ip details.

`system`

Since: 2020.06.1

system.csv.import_user

Import of user from CSV is skipped. Informs when import of a user from CSV has been skipped due to reasons such as missing required attributes or unknown unique identifier. This event is logged when import of a user is skipped during CSV directory import workflow for on-premises systems using Okta provisioning agent.

`system`

Since: 2018.28

system.custom_error.delete

Custom error page is deleted. Can be used to identify when an admin has deleted the custom error page. Event fired when the custom error page is deleted.

Custom error page is updated. Can be used to identify when an admin has customized the error page. Event fired when the error page is successfully updated.

`admin`

Since: 2020.12.0

system.custom_signin.delete

Custom sign-in page is deleted. Can be used to identify when an admin has deleted the custom sign-in page. Event fired when custom sign-in page is deleted.

`admin`

Since: 2023.01.0

system.custom_signin.update

Custom sign-in page is updated. Can be used to identify when an admin has customized the sign-in page. Event fired when custom sign-in page is updated.

`admin`

Since: 2020.12.0

system.custom_signout.update

Custom sign-out page is updated. Admin has updated the custom sign-out page. Event fired when custom sign-out page is updated.

`admin`

Since: 2023.01.0

system.custom_url_domain.cert_renew

Okta managed certificates for custom domain are renewed. Can be used to identify when okta managed certificate renewal batch job has renewed certificates for custom domain. When fired, the event contains information about the domain name and certificate source type.

`system`

Since: 2021.11.0

admin system

Since: 2020.12.0

system.custom_url_domain.delete

Custom domain is deleted. Can be used to identify when an admin has deleted their custom domain. When fired, the event contains information about the domain name that was deleted.

admin

Since: 2021.11.0

system.custom_url_domain.initiate

Custom domain setup is initiated. Admin has initiated custom domain setup by inputting their custom domain for DNS verification. When fired, the event contains information about the domain name, certificate source type and domain validation status.

admin

Since: 2020.12.0

system.custom_url_domain.update

Custom domain brand association is updated. Admin has updated the custom domain association with the brand. When fired, the event contains the domain name, certificate source type, domain validation status a information about the brand it is associated with.

admin

Since: 2023.01.0

system.custom_url_domain.verify

Verify custom domain ownership. Identifies whether an admin has succeeded or failed to verify the ownership of the domain name. When fired, the event contains information about the domain name, certificate source type and domain validation status.

admin

Since: 2020.12.0

Since: 2019.09.0

system.directory.debugger.grant

Grant Directory Debugger access for Okta support. This can be used to audit the Directory Debugger access grants to Okta support. When fired, this event contains information about Directory Debugger access grant.

agent

Since: 2019.09.0

system.directory.debugger.query_executed

A read-only query executed against AD/LDAP instance by Okta support using the Directory Debugger tool. This can be used to audit the queries executed by Okta support using Directory Debugger. When fired, this event contains information about Directory Debugger query.

agent

Since: 2019.09.0

system.directory.debugger.revoke

Revoke Directory Debugger access for Okta support. This can be used to audit the Directory Debugger access revoke. When fired, this event contains information about Directory Debugger access revoke.

agent

Since: 2019.09.0

system.dr.failback

The Enhanced Disaster Recovery (EDR) failback operation for the org domains were initiated. Triggered when the Enhanced Disaster Recovery (EDR) failback operation for the org domains were initiated. This event is fired when the Enhanced Disaster Recovery (EDR) failback operation for the org domains were initiated. If failback is successful, the outcome for this event will be SUCCESS. If failback is not successful, the outcome for this event will be FAILURE.

dr

the Enhanced Disaster Recovery (EDR) failover operation for the org domains were initiated. This event is fired when the Enhanced Disaster Recovery (EDR) failover operation for the org domains were initiated. If failover is successful, the outcome for this event will be SUCCESS. If failover is not successful, the outcome for this event will be FAILURE.

`dr`

Since: 2024.09.0

system.email.account_unlock.sent_message

Send self-service account unlock email.

`email`

Since: 2016.13

system.email.challenge_factor_redeemed

This event indicates that a user completed an email factor challenge. This can be used to identify when a credential sent in an email to a user has been redeemed (the link was clicked or the code was entered). When fired, this event contains information about the result. Success if successful or error reasons should be present for failure cases (e.g. incorrect code, timeout, expired, etc.). The event also contains a debugData with the action (the link was clicked or the code was entered).

`email`

Since: 2019.07.0

system.email.delivery

An email's delivery status was updated. Used to notify admins of a bounced or dropped email. For certain bounce events, the context information may be lost by the email provider(s) due to email server communication delays. Such delayed bounce events will not appear in syslog. As of the 2022.08.0 release, this is also used to identify other email events e.g. delivered, deferred. See the event debugData for help identifying a remediation, such as updating an incorrect email address.

`email` `event-hook-eligible`

Since: 2022.05.0

Since: 2019.01.1

system.email.mfa_reset_notification.sent_message

MFA reset notification email sent. Used to notify admins MFA reset notification email has been sent.

`email`

Since: 2019.01.1

system.email.new_device_notification.sent_message

New device signin notification email sent.

`email`

Since: 2016.13

system.email.password_reset.sent_message

Send self-service password reset email.

`email`

Since: 2016.13

system.email.send_factor_verify_message

An email was sent to a user for verification. Used to notify admins that an email was sent to a user for verification. When fired, this event contains information about the token lifetime in the debugData.

`email`

Since: 2019.07.0

system.email.template.create

This event is fired when a custom email template is created. Developers and Org Admins can use this to identify when a default email template has been overridden with a new template. The event details can be used to identify the template type and template engine. Usually this event will precede "system.email.template.update" or "system.email.template.delete" events.

This event is fired when a custom email template is deleted. Developers and Org Admins can use this to identify when a custom email template has been deleted to fall back to default template. The event details can be used to identify the template type and template engine. Usually this event will follow "system.email.template.create" or "system.email.template.update" events.

admin email

Since: 2021.07.0

system.email.template.settings_changed

This event is fired when the settings for an email template is changed. Developers and Org Admins can use this to identify when an email template setting has been changed. When fired, this event contains information about the email template and settings that were changed.

admin email

Since: 2022.05.0

system.email.template.update

This event is fired when a custom email template has been updated. Developers and Org Admins can use this to identify when a custom email template has been updated. The event details can be used to identify the template type and template engine. Usually this event will follow "system.email.template.create" and precede "system.email.template.delete" events.

admin email

Since: 2020.03.0

system.email_domain.create

Email domain is created. Admin has initiated email domain setup by inputting their domain details for DNS verification. When fired, the event contains information about the domain name, display name, user name, brand id and validation status.

admin

Since: 2023.01.0

system.email_domain.delete

system.email_domain.update

Email domain is updated. Admin has updated the email domain. When fired, the event contains information about the email domain that was updated.

admin

Since: 2023.01.0

system.email_domain.verify

Verify email domain. Identifies whether an admin has succeeded or failed to verify the email domain. When fired, the event contains information about the email domain that is being verified.

admin

Since: 2023.01.0

system.feature.disable

Fired when self service features are requested to be disabled by admins. Use to determine who enabled the features and any limitations the features have. When fired, this event contains information about the requested features, their names and lifecycle state, the admin who made the change, and any possible limitations associated with the features. Related events include 'system.feature.enable'.

admin self-service-feature-management system

Since: 2019.05.0

system.feature.ea_auto_enroll

Fired when an org has subscribed to or unsubscribed from EA Feature Auto Enroll. This can be used to understand the status of EA Feature Auto Enroll subscription and identify who has made changes to the subscription. When fired, this event contains information about the status of EA Feature Auto enroll subscription, as well as the admin who made any subscription changes.

admin self-service-feature-management system

Since: 2019.03.1

limitations associated with the features. Related events include 'system.feature.disable'.

`admin` `self-service-feature-management` `system`

Since: 2019.05.0

system.hook.key.created

Create a new hook key. This event can be used to identify when an admin created a new hook key. When triggered, this events contains information about the created hook key.

`hook-key`

Since: 2022.10.2

system.hook.key.deleted

Delete a hook key. This event can be used to identify when an admin deleted a hook key. When triggered, this events contains information about the deleted hook key.

`hook-key`

Since: 2022.10.2

system.hook.key.updated

Update a hook key. This event can be used to identify when an admin updated a hook key. When triggered, this events contains information about the updated hook key.

`hook-key`

Since: 2022.10.2

system.identity_sources.bulk_delete

Upload bulk delete data. Loads bulk data into an Identity Source Session for deactivation in Okta for an identity source. This event can be used to track the deactivations of user profiles in Okta from the custom identity source.

`identity-sources`

Since: 2024.08.0

`identity-sources`

Since: 2024.08.0

system.idp.key.create

Fired when a new Identity provider key credential is created. This can be used to audit that a new identity provider key credential has been created. When fired, this event indicates a new X.509 certificate credential is added to the IdP key store.

`event-hook-eligible` `idp`

Since: 2023.12.2

system.idp.key.delete

Fired when an Identity provider key credential is deleted. This can be used to audit that an identity provider key credential has been deleted. When fired, this event indicates a X.509 certificate credential by kid is deleted if it isn't currently being used by an active or inactive IdP.

`event-hook-eligible` `idp`

Since: 2023.12.2

system.idp.key.update

Fired when an Identity provider key credential is updated. This can be used to audit that an identity provide key credential has been updated. When fired, this event indicates a X.509 certificate credential is updated the IdP key store.

`event-hook-eligible` `idp`

Since: 2023.12.2

system.idp.lifecycle.activate

Fired when an Identity provider is activated. This can be used to audit that an identity provider has been activated. When fired, this event indicates an Identity provider was activated. This event also indicates the type of the identity provider that was activated.

`event-hook-eligible` `idp`

been created. When fired, this event indicates an Identity provider was successfully created. This event also indicates the type of the identity provider that was created.

`event-hook-eligible` `idp`

Since: 2020.09.1

system.idp.lifecycle.deactivate

Fired when an Identity provider is deactivated. This can be used to audit that an identity provider has been deactivated. When fired, this event indicates an Identity provider has been deactivated. This event also indicates the type of the identity provider that was deactivated.

`event-hook-eligible` `idp`

Since: 2020.09.1

system.idp.lifecycle.delete

Fired when an Identity provider is deleted. This can be used to audit that an identity provider has been deleted. When fired, this event indicates an Identity provider was deleted. This event also indicates the type of the identity provider that was deleted.

`event-hook-eligible` `idp`

Since: 2020.09.1

system.idp.lifecycle.read_client_secret

Fired when Identity provider(s) with a client secret is read. This can be used to audit that identity provider(s) with a client secret has been read. When fired, this event indicates one or more Identity providers with a client secret was read.

`event-hook-eligible` `idp`

Since: 2020.12.2

system.idp.lifecycle.update

Fired when an Identity provider is updated. This can be used to audit that an identity provider configuration has been updated. When fired, this event indicates an Identity provider configuration was updated. This

system.import.clear.unconfirmed.users.summary

Clear Unconfirmed Imported Users. Can be used for clearing unconfirmed imported users from last import result. Note that a single event is fired for clearing unconfirmed imported users instead of fire delete event on each user.

app

Since: 2019.01.1

system.import.complete

Import process complete.

event-hook-eligible import system

Since: 2016.14

system.import.complete_batch

Batch import process complete.

import system

Since: 2016.14

system.import.custom_object.complete

Import of custom objects completed.

import system

Since: 2016.14

system.import.custom_object.create

Create custom object triggered by import process.

import system

Since: 2016.14

system.import.custom_object.delete

system.import.custom_object.update

Update custom object triggered by import process.

import system

Since: 2016.14

system.import.download.complete

Fired at the completion of the download objects phase, when the objects (users, groups, devices) to be imported have been downloaded from the system of record. This can be used to determine the progress of an import, as well as to monitor to trigger processes that should run concurrently with the import. Fired at the completion of the download objects phase, when the objects (users, groups, devices) to be imported have been downloaded from the system of record.

import system

Since: 2020.01.0

system.import.download.start

Fired at the start of the download objects phase, when the objects (users, groups, devices) to be imported are being downloaded from the system of record. This can be used to determine when an import has started, as well as to monitor to trigger processes that should run concurrently with the import. Fired at the start of the download objects phase, when the objects (users, groups, devices) to be imported are being downloaded from the system of record.

import system

Since: 2020.01.0

system.import.group.complete

Import of groups completed.

import system

Since: 2016.14

system.import.group.create

system.import.group.delete

Remove group triggered by import process.

event-hook-eligible import system

Since: 2016.14

system.import.group.start

Start importing groups from refreshing AppGroups.

import system

Since: 2016.14

system.import.group.update

Update group triggered from import process.

import system

Since: 2016.14

system.import.group_membership.complete

Import of application group members completed.

import system

Since: 2016.14

system.import.implicit_deletion.complete

Fired upon completion of the implicit deletion phase, when Okta checks for the deletion of users, groups, ar custom objects. This can be used to determine the progress of an import, as well as to monitor to trigger processes that should run concurrently with the import. Fired upon completion of the implicit deletion phase, when Okta checks for the deletion of users, groups, and custom objects.

import system

Since: 2020.01.0

Okta checks for the deletion of users, groups, and custom objects.

`import` `system`

Since: 2020.01.0

system.import.import_profile

Import user profile triggered by import process.

`import` `system`

Since: 2016.14

system.import.import_provisioning_info

Import provisioning info triggered by import process.

`import` `system`

Since: 2016.14

system.import.membership_processing.complete

Fired upon completion of the membership processing phase, when Okta checks which groups users being imported into Okta should be added to/removed from. This can be used to determine the progress of an import, as well as to monitor to trigger processes that should run concurrently with the import. Fired upon completion of the membership processing phase, when Okta checks which groups users being imported into Okta should be added to/removed from.

`import` `system`

Since: 2020.01.0

system.import.membership_processing.start

Fired at the start of the membership processing phase, when Okta checks which groups users being imported into Okta should be added to/removed from. This can be used to determine the progress of an import, as well as to monitor to trigger processes that should run concurrently with the import. Fired at the start of the membership processing phase, when Okta checks which groups users being imported into Okta should be added to/removed from.

Fired upon completion of the object creation phase, when the first batch of objects is created/updated. This can be used to determine the progress of an import, as well as to monitor to trigger processes that should run concurrently with the import. Fired upon completion of the object creation phase, when the first batch of objects is created/updated.

`import` `system`

Since: 2020.01.0

system.import.object_creation.start

Fired at the completion of the download objects phase, when the objects (users, groups, devices) to be imported have been downloaded from the system of record. This can be used to determine the progress of an import, as well as to monitor to trigger processes that should run concurrently with the import. Fired at the completion of the download objects phase, when the objects (users, groups, devices) to be imported have been downloaded from the system of record.

`import` `system`

Since: 2020.01.0

system.import.roadblock

Import roadblock triggered due to exceeded threshold.

`event-hook-eligible` `import` `system`

Since: 2016.14

system.import.roadblock.reschedule_and_resume

The affected import from AppInstance has been rescheduled. All other imports will resume.

`import` `system`

Since: 2017.19

system.import.roadblock.resume

The affected import from AppInstance has been canceled. All other imports will resume.

`import` `system`

identify when an admin updated the Max Import Unassignment roadblock setting, and what the setting was updated to. This event includes details on what the roadblock was updated to and who made the change.

`import` `system`

Since: 2019.11.0

system.import.schedule

Import process was scheduled. This event can be used to track when import jobs were triggered, which helps with audit trails. This event may also be useful when troubleshooting a failed import, as it indicates the time at which the process was first triggered and the user or application that invoked the import. Import is a multi-stage process which may import users, groups, and group memberships. Each stage has corresponding events in the system log. For example 'system.import.user.start' indicates beginning of user import process.

`app`

Since: 2024.07.2

system.import.session.cancelled

Cancel an import session. This event can be used to identify when an admin cancel import session. This event includes details when the import session be canceled.

`import` `system`

Since: 2022.10.0

system.import.session.created

Create a new import session. This event can be used to identify when an admin start new import session. This event includes details when the import process be created.

`import` `system`

Since: 2022.10.0

system.import.session.expired

system.import.session.triggered

Triggered an import session to start importing. This event can be used to identify when an admin trigger the import job from an open session. This event includes details when the import process be triggered.

`import` `system`

Since: 2022.10.0

system.import.start

import started.

`event-hook-eligible` `import` `system`

Since: 2016.14

system.import.user.complete

Import of user completed.

`import` `system`

Since: 2016.14

system.import.user.create

Create user triggered by import process.

`import` `system`

Since: 2016.14

system.import.user.delete

Delete user triggered by import process.

`import` `system`

Since: 2016.14

system.import.user.match

system.import.user.start

Start importing users triggered import process.

`import` `system`

Since: 2016.14

system.import.user.suspend

Suspend user triggered by import process.

`import` `system`

Since: 2016.24

system.import.user.unsuspend

Unsuspend user triggered by import process.

`import` `system`

Since: 2016.24

system.import.user.unsuspend_after_confirm

No Description

`import` `system`

Since: 2016.24

system.import.user.update

Update user triggered by import process.

`import` `system`

Since: 2016.14

system.import.user.update_user_lifecycle_from_master

Update user status triggered by import process.

Bulk Import users from CSV is completed. Informs when bulk user import from CSV has been completed. This event is logged when bulk user import from CSV has completed with the outcome as success or failure. When fired, this event also contains debug context about the number of users added/updated/unchanged or with errors.

`admin` `csv-upload` `user-import`

Since: 2021.01.2

system.import.user_csv.start

Bulk Import of users from CSV is started. Informs when bulk import of users from CSV has been attempted to be uploaded. This event is logged when bulk user import from CSV has started and is a precursor to user.lifecycle.create; user.lifecycle.activate events.

`admin` `csv-upload` `user-import`

Since: 2021.01.2

system.import.user_match.confirm

Import user matching assignment confirmed. This event can be used to track when the confirmation of user matching assignments was triggered on the Import page, which helps with audit trails. This event may also be useful when troubleshooting incorrect user matches. After users are imported from the app, they're matched and assigned with existing Okta users on the basis of Name, Username, and Email. The assignment confirmation is a manual step, needing admin intervention.

`app`

Since: 2024.07.2

system.import.user_match.unignore

Assignment was unignored. This event indicates that a user match, which was previously marked to be ignored during imports, has been reactivated for consideration. It's important for tracking changes in user matching policies and decisions during the import process. This event can be of critical importance for auditing purposes, especially when investigating why certain user accounts were matched or updated after being ignored in previous imports. It helps maintain the accuracy and integrity of user data by ensuring that valid matches are not permanently overlooked.

Assignment was modified. This event can be used to track when an assignment was modified. This may also be useful when troubleshooting incorrect user assignments. After users are imported from the app, they're matched and assigned with existing Okta users on the basis of Name, Username, and Email. Assignments can be modified by the admin through a manual intervention.

app

Since: 2024.07.2

system.import.user_matching.complete

Fired upon completion of the user matching phase, when Okta attempts to match imported users to existing Okta users. This can be used to determine the progress of an import, as well as to monitor to trigger processes that should run concurrently with the import. Fired upon completion of the user matching phase, when Okta attempts to match imported users to existing Okta users.

import system

Since: 2020.01.0

system.import.user_matching.start

Fired at the start of the user matching phase, when Okta attempts to match imported users to existing Okta users. This can be used to determine the progress of an import, as well as to monitor to trigger processes that should run concurrently with the import. Fired at the start of the user matching phase, when Okta attempts to match imported users to existing Okta users.

import system

Since: 2020.01.0

system.iwa.create

Create IWA agent.

iwa system

Since: 2016.13

system.iwa.go_offline

system.iwa.go_online

IWA going online.

iwa system

Since: 2016.13

system.iwa.promote_primary

Promote IWA agent to primary.

iwa system

Since: 2016.13

system.iwa.remove

Remove IWA agent.

iwa system

Since: 2016.13

system.iwa.update

Update IWA agent.

iwa system

Since: 2016.13

system.iwa.use_default

No primary IWA app found. Using default login.

iwa system

Since: 2016.13

system.iwa_agentless.auth

Agentless IWA authentication.

iwa system

authentication request after Agentless DSSO fails. This can also be used for end-to-end tracking of an ADSSO failure to the subsequent authentication it is redirected to by searching for the common stateTokenHash. When fired, this event contains the stateTokenHash which will be common before and after the redirection occurs.

iwa system

Since: 2022.11.2

system.iwa_agentless.redirect

Fired when an Agentless DSSO authentication request is redirected to an onprem IWA authentication or the default login page. This can be used to identify when an agentless authentication request resulted in a redirect to an onprem IWA or default login page. This can also be used to identify the potential cause of the redirect. When fired, this event identifies the cause of the redirection. When a custom error page is defined, a redirect event is not always generated when a redirection occurs.

iwa system

Since: 2019.05.4

system.iwa_agentless.update

Update to agentless IWA.

iwa system

Since: 2018.22

system.iwa_agentless.user.not_found

Fired when a user could not be found during Agentless DSSO authentication, resulting in an authentication failure. This can be used to identify when an agentless authentication request resulted in a failure. The failur could be due to the user not being found in Okta, Okta not being able to connect to AD, or the user not being found in AD. This can also be used to identify the potential cause of the failure. When fired, this event contains information about the potential cause of the failure.

iwa system

Since: 2019.08.0

This event also indicates the initiator of the event and the current setting for Kerberos Realm. Related events: none, all debugging context is included in this event.

`iwa` `system`

Since: 2019.05.4

system.ldapi.admin_limit_exceeded

This event indicates that an administrative limit was exceeded when processing an LDAP interface operation. It can be used to audit and debug failures caused by exceeding an administrative limit. This event may occur periodically when an LDAP operation results in a large number of corresponding actions in the Okta directory. These errors are often temporary and will subside when Okta has processed the actions. Contact Okta support if you see such errors consistently over the course of a day or more.

`ldapi`

Since: 2023.03.0

system.ldapi.bind

Fired when a user performs a BIND to LDAP Interface. Can be used to identify when a user attempted to perform an LDAP authentication for audit or debugging purposes. Note that the firing of this event is subject to LDAPi event filtering rules.

`ldapi`

Since: 2018.10

system.ldapi.search

Fired when a user performs a SEARCH to LDAP Interface. Can be used to identify when a user attempted to perform a search on LDAP Interface for audit or debugging purposes. Note that the firing of this event is subject to LDAPi event filtering rules.

`ldapi`

Since: 2018.10

system.ldapi.unbind

system.log_stream.lifecycle.activate

Log stream activated. This event can be used to track and audit when a user activates a log stream. When fired, this event indicates that a user activated a log stream configuration.

`event-hook-eligible` `log-stream`

Since: 2021.09.1

system.log_stream.lifecycle.create

Log stream created. This event can be used to track and audit when a user creates a log stream. When fired, this event indicates that a user created a log stream configuration.

`event-hook-eligible` `log-stream`

Since: 2021.09.1

system.log_stream.lifecycle.deactivate

Log stream deactivated. This event can be used to track and audit when a user or Okta deactivates a log stream. When fired, this event indicates that a user or Okta deactivated a log stream configuration.

`event-hook-eligible` `log-stream`

Since: 2021.09.1

system.log_stream.lifecycle.delete

Log stream deleted. This event can be used to track and audit when a user deletes a log stream. When fire this event indicates that a user deleted a log stream configuration.

`event-hook-eligible` `log-stream`

Since: 2021.09.1

system.log_stream.lifecycle.update

Log stream updated. This event can be used to track and audit when a user updates a log stream. When fired, this event indicates that a user updated a log stream configuration.

Activate a new authentication factor. Can be used to identify when an admin has enabled a new factor for authentication. When fired the event will contain details of which factor is enabled.

admin mfa

Since: 2021.01.1

system.mfa.factor.deactivate

Deactivate MFA factor. Can be used to identify when an admin has disabled a factor for MFA. When fired the event will contain details of which factor is disabled.

admin mfa

Since: 2021.01.1

system.operation.concurrency_limit.violation

Operation concurrency limit violation. This can be used to track if there are too many concurrent operations of the given type. The operation type information is available in debugData. When fired, this event contains information about the operation such as its actor, type, scope and threshold details. OperationRateLimitType in debugData will indicate the category to which the concurrency limit is being applied (e.g. web_request), OperationRateLimitSubtype defines specific subtypes (e.g. ssws_token) and OperationRateLimitScope will indicate the scope of the rate limit (e.g. token).

system

Since: 2022.07.0

system.operation.rate_limit.violation

Operation rate limit violation. This can be used to track if an operation is exceeding its rate limit. When fired, this event contains information about the operation such as actor, type, scope and threshold details. OperationRateLimitType in debugData will indicate the category to which the rate limit is being applied (e.g. authenticator_otp_verification), OperationRateLimitSubtype defines specific subtypes (e.g. Email Factor for authenticator_otp_verification) and OperationRateLimitScope will indicate the scope of the rate limit (e.g. user or org level). Formerly, this event was used to indicate blocked SMS/Call transactions, please see system.sms.send*/system.voice.send* for blocked transactions.

system

this event contains information about the operation such as actor, type, scope and threshold details. OperationRateLimitType in debugData will indicate the category to which the rate limit is being applied (e.g. authenticator_otp_verification), OperationRateLimitSubtype defines specific subtypes (e.g. Email, SMS or Voice call for authenticator_otp_verification type) and OperationRateLimitScope will indicate the scope of the rate limit (e.g. user or org level).

`system`

Since: 2021.01.2

system.org.captcha.activate

Enable org-wide captcha support. Indicates when org-wide captcha support is enabled, for which pages and using which captcha instance. This event is fired when org admin enables org-wide captcha for any supported pages.

`captcha` `system`

Since: 2021.05.1

system.org.captcha.deactivate

Disable org-wide captcha support. Indicates when org-wide captcha support is disabled. This event is fired when org admin disables org-wide captcha support for all pages.

`captcha` `system`

Since: 2021.05.1

system.org.lifecycle.create

Org creation.

`system`

Since: 2016.51

system.org.rate_limit.burst

Fired when burst rate limit capacity is activated. This can be used to identify when an API in the Org exceeds standard rate limits and the frequency with which the activities occur. This event is fired after a

system.org.rate_limit.expiration.warning

Rate limit approaching expiration date.

`system`

Since: 2018.35

system.org.rate_limit.violation

Rate limit violation.

`event-hook-eligible` `system`

Since: 2017.02

system.org.rate_limit.warning

Rate limit warning.

`event-hook-eligible` `system`

Since: 2017.02

system.org.task.remove

Tasks removed.

`system`

Since: 2017.33

system.push.send_factor_verify_push

Fired when a Push notification is sent to a device. Used to notify admins when a push was sent to a user for verification. Note that this event is fired whenever a Push is sent.

`push`

Since: 2020.06.3

access.request for events relevant to OIG Access requests.

`self-service` `changeDetails`

Since: 2024.08.0

system.sms.receive_status

Fired when receiving a status update on SMS message from provider. This event can be used by Org Admins to identify users that are/aren't getting one-time passcodes delivered successfully via SMS, provider status can be obtained from status field in debug data. For any system.sms.send_* event, there should be exactly one of this event.

`sms`

Since: 2020.08.4

system.sms.send_account_unlock_message

Send self-service account unlock SMS message. As of the 2022.06.0 release this event is also used to identify transactions blocked by Okta, which is indicated by a "deny" outcome. Previously, the system.operation.rate_limit.violation was used to identify blocked transactions. Additionally, the method of generating the MobilePhone ID in the event has changed for Okta Classic. It has not changed for Okta Identity Engine.

`sms` `system`

Since: 2016.12

system.sms.send_factor_verify_message

Send second factor auth SMS. As of the 2022.06.0 release this event is also used to identify transactions blocked by Okta, which is indicated by a "deny" outcome. Previously, the system.operation.rate_limit.violation was used to identify blocked transactions. Additionally, the method of generating the MobilePhone ID in the event has changed for Okta Classic. It has not changed for Okta Identity Engine.

`sms` `system`

Since: 2016.12

generating the MobilePhone ID in the event has changed for Okta Classic. It has not changed for Okta Identity Engine.

`sms` `system`

Since: 2016.12

system.sms.send_password_reset_message

Send self-service password reset SMS message. As of the 2022.06.0 release this event is also used to identify transactions blocked by Okta, which is indicated by a "deny" outcome. Previously, the system.operation.rate_limit.violation was used to identify blocked transactions. Additionally, the method of generating the MobilePhone ID in the event has changed for Okta Classic. It has not changed for Okta Identity Engine.

`sms` `system`

Since: 2016.12

system.sms.send_phone_verification_message

Send phone verification SMS message. As of the 2022.06.0 release this event is also used to identify transactions blocked by Okta, which is indicated by a "deny" outcome. Previously, the system.operation.rate_limit.violation was used to identify blocked transactions. Additionally, the method of generating the MobilePhone ID in the event has changed for Okta Classic. It has not changed for Okta Identity Engine.

`event-hook-eligible` `sms` `system`

Since: 2016.12

system.theme.update

This event is fired when the theme resource is updated. Developer and org admins can use this event to identify when and how the theme resource was updated. Event details can be used to identify changes made to theme assets including updates to theme hex codes, logo, background image, and favicon. This event also tracks which combination of theme assets was applied to end users pages such as the sign-in page, error pages, and email templates.

`admin`

identify users that are/aren't getting one-time passcodes delivered successfully via voice call, provider status can be obtained from status field in debug data. For any system.voice.send_* event, there should be exactly one of this event.

voice

Since: 2020.08.4

system.voice.send_account_unlock_call

Send self-service account unlock call. As of the 2022.06.0 release this event is also used to identify transactions blocked by Okta, which is indicated by a "deny" outcome. Previously, the system.operation.rate_limit.violation was used to identify blocked transactions. Additionally, the method of generating the MobilePhone ID in the event has changed for Okta Classic. It has not changed for Okta Identity Engine.

voice

Since: 2017.44

system.voice.send_call

Send phone call.

voice

Since: 2017.44

system.voice.send_mfa_challenge_call

Send second factor auth call. As of the 2022.06.0 release this event is also used to identify transactions blocked by Okta, which is indicated by a "deny" outcome. Previously, the system.operation.rate_limit.violation was used to identify blocked transactions. Additionally, the method of generating the MobilePhone ID in the event has changed for Okta Classic. It has not changed for Okta Identity Engine.

voice

Since: 2017.44

system.voice.send_password_reset_call

`voice`

Since: 2017.44

system.voice.send_phone_verification_call

Send phone verification call. As of the 2022.06.0 release this event is also used to identify transactions blocked by Okta, which is indicated by a "deny" outcome. Previously, the system.operation.rate_limit.violation was used to identify blocked transactions. Additionally, the method of generating the MobilePhone ID in the event has changed for Okta Classic. It has not changed for Okta Identity Engine.

`event-hook-eligible` `voice`

Since: 2017.44

task.lifecycle.activate

Activated system task.

`task`

Since: 2018.15

task.lifecycle.create

Created system task.

`task`

Since: 2018.15

task.lifecycle.deactivate

Deactivated system task.

`task`

Since: 2018.15

task.lifecycle.delete

Deleted system task.

Updated system task.

`task`

Since: 2018.15

user.account.expire_password

Fired when the user's Okta password is expired. This can be used to audit cases where a user's password is expired by an administrator. When fired, this event contains information about the user whose password was expired, whether a temporary password was created for the user, or if the user's sessions were revoked.

`account` `user`

Since: 2024.01.2

user.account.lock

Auto-lock user account for Okta.

`account` `event-hook-eligible` `user`

Since: 2016.02

user.account.lock.limit

This event is fired when a user account has reached the lockout limit. The account will not auto-unlock and a user or client cannot gain access to the account. This event indicates an account that will not be able to log in until remedial action is taken by the account admin. This event can be used to understand the specifics of an account lockout. Often this indicates a client application that is repeatedly attempting to authenticate with invalid credentials such as an old password.

`account` `user`

Since: 2019.05.0

user.account.preference_update

User preferences updated. This can be used for debugging and auditing purposes. These preferences live outside the user profile.

`account` `user`

When fired, this event contains information about the type of admin privileges the user currently has. The list of current privileges contain both individually assigned roles as well as the ones granted to the user through their group membership. Related events include: USER_ACCOUNT_PRIVILEGE_REVOKE.

`event-hook-eligible` `user`

Since: 2016.15

user.account.privilege.revoke

All of user's admin privilege revoked. This can be used to audit the deprovisioning of admin privileges from users. When fired, this event indicates the user has no more admin privileges. All of user's privileges were revoked including individually assigned roles as well as the ones granted to the user through their group membership. Related events include: USER_ACCOUNT_PRIVILEGE_GRANT.

`event-hook-eligible` `user`

Since: 2016.15

user.account.report_suspicious_activity_by_enduser

User reported suspicious activity. This event is used to identify user account suspicious activity.

`event-based-trigger-eligible` `event-hook-eligible` `user`

Since: 2019.01.1

user.account.reset_password

Fired when the user's Okta password is reset.

`account` `event-hook-eligible` `user`

Since: 2016.15

user.account.unlock

Auto-unlock user account for Okta.

`account` `event-hook-eligible` `user`

Since: 2016.15

user.account.unlock_failure

Failed to schedule unlock job for user.

account user

Since: 2018.23

user.account.unlock_token

Issued recovery token for self-service account unlock.

account user

Since: 2017.47

user.account.update_password

User update password for Okta.

account end-user-visible event-hook-eligible user

Since: 2016.15

user.account.update_primary_email

User primary email updated.

account end-user-visible user user-config

Since: 2018.05

user.account.update_profile

Update user profile for Okta.

account event-hook-eligible user user-config

Since: 2016.02

user.account.update_secondary_email

user.account.update_user_type

Fires when a user changes from one type to another. Can be used to audit when a user gets converted from a contractor to a full-time employee, for example. Data includes the old and new type ids. There may be an accompanying update_profile event if values were changed.

`account` `user` `user-config`

Since: 2020.02.0

user.account.use_token

Invalid self service recovery token used by user.

`account` `user`

Since: 2016.15

user.authentication.auth

Authenticate user.

`user`

Since: 2016.02

user.authentication.auth_unconfigured_identifier

Fired after a user authenticates via a directory instance that is not the highest priority profile source for the user. This can be used to track users that are using an identifier to login which is different from the admin configured identifier for that user which might result in unexpected login results. When fired, this event will contain useful information about the user, the directory instance that was used to login the user, and the directory instance that should have been used instead.

`directory` `user`

Since: 2023.01.2

user.authentication.auth_via_AD_agent

Authenticate user with AD agent.

`directory` `user`

`event-hook-eligible` `user`

Since: 2016.18

user.authentication.auth_via_LDAP_agent

Authenticate user via LDAP agent.

`directory` `user`

Since: 2016.18

user.authentication.auth_via_inbound_SAML

Authenticate user via inbound SAML.

`user`

Since: 2016.27

user.authentication.auth_via_inbound_delauth

Authenticate user via inbound delauth.

`user`

Since: 2016.02

user.authentication.auth_via_iwa

Authenticate user via IWA.

`user`

Since: 2016.02

user.authentication.auth_via_mfa

Authentication of user via MFA. For Okta Classic orgs, this event will only fire for second factor verifications, whereas for Identity Engine orgs, this event will fire for both primary and second factor verifications.

`event-hook-eligible` `mfa`

Since: 2016.02

user.authentication.auth_via_richclient

Authentication of a user via Rich Client.

`user`

Since: 2016.18

user.authentication.auth_via_social

Authenticate user with social login.

`event-hook-eligible` `user`

Since: 2016.18

user.authentication.authenticate

Authentication via device trust certificate.

`device-trust-authentication` `event-hook-eligible` `user`

Since: 2017.44

user.authentication.dsso_via_non_priority_source

Desktop Single Sign On (DSSO) authentication has been attempted using a profile source that is not the highest priority profile source for the given Okta user. This event may indicate a potential security risk as the highest priority profile source is often expected to be used in this flow. The presence of this event may be benign, or it may indicate an attempt to authenticate the user from a compromised Active Directory domain. The debugContext object in this event contains useful information regarding the Okta user, the prioritized profile source, and the profile source that was used in the DSSO attempt.

`directory` `user`

Since: 2024.03.0

user.authentication.slo

User single logout out (SLO) from app.

Fired when a user performs a single sign-on (SSO) to an app instance and contains the client details of the user. Can be used to identify when a user attempted to sign into an application for audit or debugging purposes. Note that the event is fired even when the sign-on is unsuccessful.

`event-hook-eligible` `user`

Since: 2016.11

user.authentication.universal_logout

This event is fired when an admin or system account triggers Universal Logout against an app instance. It contains the app instance details for which the Universal Logout API was fired. This event identifies when applications have had Universal Logout triggered for audit or debugging purposes. This event is only fired once. It's only fired for applications that have been configured for Universal Logout. You can configure it under Risk policy, Post Auth Session policy, or in an admin-initiated Clear User Session.

See also: Identity Threat Protection with Okta AI Event Types

`event-hook-eligible` `session` `user`

Since: 2024.02.2

user.authentication.universal_logout.scheduled

This event is fired when an admin manually triggers Universal Logout for a user. It contains context about the initiating request, such as where the request originated and how the Universal Logout endpoint was invoked. After Universal Logout is complete, the user.authentication.universal_logout event is fired, and you can correlate both events using the traceID. This event identifies the geolocation, IP address, and IP chain of the requesting entity. This event is only fired once. You can correlate this event with the user.authentication.universal_logout event using traceID.

`event-hook-eligible` `session` `user`

Since: 2024.06.0

user.authentication.verify

Verify user identity.

`end-user-visible` `user`

be helpful when troubleshooting unexpected behavior detection evaluations. This event is triggered when an administrator manually resets a user's behavior profile in the Admin Console.

`behavior-profile` `event-hook-eligible`

Since: 2024.08.0

user.credential.enroll

Device Trust certificate enrollment.

`device-trust-cert-distribution-and-binding` `event-hook-eligible` `user`

Since: 2017.45

user.identity_snapshot.attestation.create

Create identity snapshot attestation for a user. This event can be used by administrators to audit identity snapshot attestations minted for a user. The user and the application are in the event, signifying which user the attestation token is being minted for, and which application is requesting it.

`attestation` `user`

Since: 2020.09.3

user.import.password

Fired when a user has successfully logged in to Okta and an attempt to import their Password has been made. This can be used to understand if a user password import attempt was successful or if it failed. If the attempt failed, the password import will be tried again on a subsequent successful login. When fired, this event contains information about the import type, and whether or not the password import was successful. the import is successful, it is safe to "clean up" that user from an external system. If the import failed, Okta w continue retrying the import during every successful authentication attempt until the password is successfully imported. Check the failure reason for details about whether any action is needed for the import to succeed.

`credential` `event-hook-eligible` `import` `user`

Since: 2020.05.1

user.lifecycle.activate

user.lifecycle.create

Create Okta user.

`event-hook-eligible` `user`

Since: 2016.02

user.lifecycle.deactivate

Deactivate Okta user.

`event-hook-eligible` `user`

Since: 2016.02

user.lifecycle.delete.completed

Delete Okta user completed.

`user`

Since: 2016.29

user.lifecycle.delete.initiated

Delete Okta user initiated.

`event-hook-eligible` `user`

Since: 2016.29

user.lifecycle.jit.error.read_only

Failed to JIT create user.

`user`

Since: 2018.06

user.lifecycle.password_mass_expiry

Mass expire all users' passwords initiated.

`user`

`event-hook-eligible` `user`

Since: 2016.13

user.lifecycle.suspend

Suspend Okta user.

`event-hook-eligible` `user`

Since: 2016.13

user.lifecycle.unsuspend

Unsuspend Okta user.

`event-hook-eligible` `user`

Since: 2016.13

user.mfa.attempt_bypass

Attempt bypass of factor.

`mfa`

Since: 2016.11

user.mfa.factor.activate

Activate factor or authenticator enrollment method for user. Provides org admins with audit log and oversig utility for an MFA factor when it is activated. When fired, the event contains information about the MFA facto that has been activated, as well as the target user and the user activating the factor. For Identity Engine org this event will fire when an authentication method is enrolled.

`end-user-visible` `event-hook-eligible` `mfa`

Since: 2016.11

user.mfa.factor.deactivate

`end-user-visible` `event-hook-eligible` `mfa`

Since: 2016.11

user.mfa.factor.reset_all

Reset all factors or authenticator enrollments for user. Provides org admins with audit log and oversight utility for the change in MFA factor lifecycle statuses when all MFA factors for a user are permanently deactivated. When fired, the event contains information about the target user for whom all factors have been deactivated, as well as the user resetting the factors. For Identity Engine orgs, this event contains information about a target user for whom all authenticator enrollments have been reset.

`event-hook-eligible` `mfa`

Since: 2016.11

user.mfa.factor.suspend

Suspend factor or authenticator enrollment method for user. Provides org admins with audit log and oversight utility for the change in MFA factor lifecycle status when a factor is suspended, usually as a result of suspected compromise. When fired, the event contains information about the MFA factor that has been suspended, as well as the target user and the user suspending the factor. When unsuspended, related event user.mfa.factor.unsuspend will be fired.

`event-hook-eligible` `mfa` `oie-only`

Since: 2020.09.4

user.mfa.factor.unsuspend

Unsuspend factor or authenticator enrollment method for user. Provides org admins with audit log and oversight utility for the change in MFA factor lifecycle status when a factor is reactivated from a state of suspension, after it has been determined that the authenticator is secure. When fired, the event contains information about the MFA factor that has been unsuspended, as well as the target user and the user reactivating the suspended factor. Before suspension, related event user.mfa.factor.suspend would have been fired.

`event-hook-eligible` `mfa` `oie-only`

Since: 2020.09.4

user.mfa.okta_verify

Verify user with Okta verify.

`mfa`

Since: 2016.11

user.mfa.okta_verify.deny_push

User rejected Okta push verify. This event is triggered in classic V1 API calls. In OIE we use a generic event for factor verification failure: user.authentication.auth_via_mfa with reason INVALID_CREDENTIALS.

`mfa`

Since: 2018.03

user.mfa.okta_verify.deny_push_upgrade_needed

Rejected Okta push verify as Upgrade Needed. This can be used to audit events where Okta push verify was rejected as the app needed upgrade. Note that the event is fired when Okta Verify push is rejected. It is possible that the user might have chosen another factor and made successful login as well.

`mfa`

Since: 2020.05.0

user.risk.change

Indicates a user's risk level has changed. This event can be used to monitor risk level changes for users. Th event triggers when Okta determines that a user is associated with a change in risk activity or context.

See also: Identity Threat Protection with Okta AI Event Types

`event-hook-eligible` `risk` `security`

Since: 2023.01.2

user.risk.detect

Since: 2024.06.1

user.session.access_admin_app

User accessing Okta admin app.

`admin` `app` `session` `user`

Since: 2016.14

user.session.clear

Clear user session.

See also: Identity Threat Protection with Okta AI Event Types

`event-hook-eligible` `session` `user`

Since: 2016.15

user.session.context.change

User session context changed. This event indicates that the context in which the session is being used has changed significantly enough from the context in which the event was created, that re-evaluation of policy may be required. Often this indicates a security issue related to the session.

`event-hook-eligible` `session` `user`

Since: 2023.01.0

user.session.end

User logout from Okta.

See also: Identity Threat Protection with Okta AI Event Types

`event-hook-eligible` `session` `user`

Since: 2016.02

user.session.expire

user.session.impersonation.end

End impersonation session.

session user

Since: 2016.09

user.session.impersonation.extend

Extend impersonation session.

session user

Since: 2016.09

user.session.impersonation.grant

Enable impersonation grant.

session user

Since: 2016.09

user.session.impersonation.initiate

Initiate impersonation session.

session user

Since: 2016.09

user.session.impersonation.revoke

Revoke impersonation grant.

session user

Since: 2016.09

user.session.start

User login to Okta.

This event can be used by any admin or security team member to monitor the creation of new connections for Workflows connectors. The target fields provide information on the user that created the connection, the application for which the connection was created, and the display name the user provided for the connection. Other connection lifecycle events include: workflows.user.connection.revoke, workflows.user.connection.reauthorize, and workflows.user.connection.delete. Note that this event only indicates if a connection was successfully added to the database, and does not distinguish whether or not that connection is valid.

`workflows`

Since: 2021.02.1

workflows.user.connection.delete

This event can be used by any admin or security team member to monitor the deletion of existing Workflows connections. The target fields provide information on the user that deleted the connection, the application for which the connection was deleted, and the display name originally provided for the connection. Other connection lifecycle events include: workflows.user.connection.create, workflows.user.connection.reauthorize, and workflows.user.connection.revoke. Note that for OAuth connections this will often fire with the workflows.user.connection.revoke event.

`workflows`

Since: 2021.02.1

workflows.user.connection.reauthorize

This event can be used by any admin or security team member to monitor the reauthorization of existing connections for Workflows connectors. Reauthorization can be used to retrieve a new access token or to change the credentials used by a connection. The target fields provide information on the user that reauthorized the connection, the application for which the connection was reauthorized, and the display name originally provided for the connection. Other connection lifecycle events include: workflows.user.connection.create, workflows.user.connection.revoke, and workflows.user.connection.delete. Note that this event only indicates if a user attempted to reauthorize a connection, and does not distinguish whether or not that reauthorization was successful.

`workflows`

Since: 2021.02.1

connection, the application for which the connection was revoked, and the display name originally provided for the connection. Other connection lifecycle events include: workflows.user.connection.create, workflows.user.connection.reauthorize, and workflows.user.connection.delete. Note that this event only fires for connections where the service supplies an API endpoint for revoking tokens. Tokens that cannot be revoked via API must be managed manually in the third party application.

`workflows`

Since: 2021.02.1

workflows.user.delegatedflow.run

This event can be used by admins or security team members to monitor the execution of delegated flows in the Workflows platform from the Admin application. The actor field provides the Okta User ID of the user that ran the flow. The target fields provide context on the Workflows instance as well as the name and flow id of the executed flow. This event only indicates if the flow was successfully triggered and does not provide information about whether the flow encountered an error.

See also: Identity Threat Protection with Okta AI Event Types

`workflows`

Since: 2022.06.0

workflows.user.execution_log_stream_connection.activate

Workflows admin activated execution log streaming for their org. Connections to a downstream HTTP endpoint (e.g. ingestion point to a SIEM) may be configured to stream execution logs for a Workflows org. Note that these logs only contain metadata about flow executions and not the I/O data processed in each execution.

`workflows`

Since: 2024.03.0

workflows.user.execution_log_stream_connection.deactivate

Workflows admin deactivated execution log streaming for their org. Connections to stream Workflows execution logs may be shut off by a Workflows admin at any time. Note that deactivating an execution log

workflows.user.execution_log_stream_connection.update

Workflows admin updated the configuration of their org's execution log streaming connection. These changes may be related to the connection's destination URL, event subscriptions, or the headers and message body of each request. Note that the detailEntry field in this event's target object contains an array of fieldsUpdated, with the following possible values: DESTINATION_URL, EVENT_SUBSCRIPTIONS, HEADERS, or BODY.

`workflows`

Since: 2024.03.0

workflows.user.flow.activate

Triggered when a user activates a flow in Workflows. Can be used to audit user activity in Workflows. Event is fired when a user toggles a flow on.

`workflows`

Since: 2021.02.1

workflows.user.flow.create

Triggered when a user creates a new flow in Workflows. Can be used to audit user activity in Workflows. Event is fired when a user creates and saves a new flow.

`workflows`

Since: 2021.02.1

workflows.user.flow.deactivate

Triggered when a user deactivates a flow in Workflows. Can be used to audit user activity in Workflows. This is triggered by deactivating a flow.

`workflows`

Since: 2021.02.1

workflows.user.flow.delete

workflows.user.flow.execution.cancel

Workflows user requested to cancel flow execution. These requests attempt cancellation of in-progress flow executions which are infinitely looping, stalled, or accidentally triggered. Canceling a flow execution cancels the execution of all remaining steps in the flow as well as all parent and helper flow executions associated with that execution. These cancellation requests are best-effort meaning that at the time of request some execution processes may be past the point of no return and will still complete.

`workflows`

Since: 2023.08.0

workflows.user.flow.execution_history.activate

Workflows user activated saving execution history for a given flow. Flows may save recent execution history for the purposes of testing, debugging, or auditing a flow's activity in the Workflows console. Note that in-product flow execution history is retained for 30 days.

`workflows`

Since: 2024.03.0

workflows.user.flow.execution_history.deactivate

Workflows user deactivated saving execution history for a given flow. Flows may be opted out of saving recent execution history for any reason (e.g. handling extremely sensitive data). Note that this setting is managed per individual flow, so helper flows invoked by a flow which has this setting deactivated will continue to write history unless switched off themselves.

`workflows`

Since: 2024.03.0

workflows.user.flow.execution_history.delete

Workflows user deleted all or part of a flow's execution history. Either in the course of testing / debugging or for data sensitivity / compliance reasons, a Workflows user may elect to delete recent execution history for a given flow. Note that the detailEntry field in this event's target object contains an executionHistoryType, which may be IO_DATA_ONLY or ALL depending on which option was selected in the UI.

Workflows user activated execution log streaming for a given flow. A flow with execution log streaming deactivated may be reactivated by an authorized Workflow user at any time. This flow-level setting is enabled by default and so this activation event will only fire in the case of an individual flow having execution log streaming deactivated then reactivated.

`workflows`

Since: 2024.03.0

workflows.user.flow.execution_log_stream.deactivate

Workflows user deactivated execution log streaming for a given flow. Individual flows may have execution log streaming deactivated by an authorized Workflow user at any time to remain within monthly execution log limits per org or to reduce log volume/noise in downstream systems. This flow-level setting is enabled by default and must be manually deactivated on individual flows for which execution log streaming is undesired while an org's execution log streaming connection remains active.

`workflows`

Since: 2024.03.0

workflows.user.flow.export

Triggered when a user exports a flow from Workflows. Can be used to audit user activity in Workflows. Event is fired when a user exports one or more flows as a flowpack.

`workflows`

Since: 2021.02.1

workflows.user.flow.import

Triggered when a user imports a flow into Workflows. Can be used to audit user activity in Workflows. Event fired when a user imports one or more flows as a flowpack.

`workflows`

Since: 2021.02.1

workflows.user.flow.move

context field.

`workflows`

Since: 2024.07.0

workflows.user.flow.save

Triggered when a user saves a flow in Workflows. Can be used to audit user activity in Workflows. Event is fired when a user saves a flow.

`workflows`

Since: 2021.02.1

workflows.user.folder.create

This event can be used by any admin or security team member to monitor the creation of new folders in the Workflows platform. The payload provides information about the user that created the folder and the folder that was created. Other folder lifecycle events include: workflows.user.folder.delete, workflows.user.folder.import, workflows.user.folder.export, and workflows.user.folder.rename. Note that this event doesn't fire when a folder is imported. For that, users can reference workflows.user.folder.import.

`workflows`

Since: 2023.06.1

workflows.user.folder.delete

This event can be used by any admin or security team member to monitor the deletion of folders in the Workflows platform. The payload provides information on the user that deleted the folder and which folder was deleted. Other folder lifecycle events include: workflows.user.folder.create, workflows.user.folder.import, workflows.user.folder.export, and workflows.user.folder.rename. Note that this event fires when a user manually deletes a folder and recursively for each subfolder contained within the deleted folder. Subsequent workflows.user.flow.delete and workflows.user.table.delete events will fire for each flow and table deleted within each folder.

`workflows`

Since: 2022.11.1

workflows.user.folder.delete, workflows.user.folder.import, and workflows.user.folder.rename. Note that this event fires for the exported folder and recursively for each subfolder contained within the exported folder depending on the user's selection. Subsequent workflows.user.flow.export and workflows.user.table.schema.export events will fire for each flow and table exported within each exported folder. Additional folder information can be found in the debug context field.

`workflows`

Since: 2023.06.1

workflows.user.folder.import

This event can be used by any admin or security team member to monitor when a user imports a folder to the Workflows platform. The payload provides information on the user that imported the folder and the folder that was imported. Other folder lifecycle events include: workflows.user.folder.create, workflows.user.folder.delete, workflows.user.folder.export, and workflows.user.folder.rename. Note that this event fires for the imported folder and recursively for each subfolder contained within the imported folder. Subsequent workflows.user.flow.import and workflows.user.table.schema.import events will fire for each flow and table imported within each imported folder. Additional folder information can be found in the debug context field.

`workflows`

Since: 2023.06.1

workflows.user.folder.move

This event can be used by any admin or security team member to monitor when a user moves a folder in th Workflows platform. The payload provides information on the user that moved the folder and the folder that was moved. Other folder lifecycle events include workflows.user.folder.create, workflows.user.folder.delete, workflows.user.folder.import, workflows.user.folder.export, workflows.user.folder.rename, and workflows.user.folder.duplicate. Note that this event fires for the moved folder and recursively for each subfolder contained within the moved folder. Additional information including old and new folder locations can be found in the debug context field.

`workflows`

Since: 2024.07.0

workflows.user.folder.delete, workflows.user.folder.import, and workflows.user.folder.export. Additional information including old and new folder names can be found in the debug context field.

`workflows`

Since: 2023.06.1

workflows.user.role.group.add

This event can be used by any admin or security team member to monitor the addition of Workflows roles to an Okta group. The payload provides information about both the group to which the role was added and the role that was added. Related events include workflows.user.role.group.remove, workflows.user.role.user.add, workflows.user.role.user.remove, application.user_membership.add, and application.user_membership.remove. The event fires when an admin manually adds a role to an Okta group in the Workflows console. Adding multiple roles in a single action triggers multiple system log events.

`workflows`

Since: 2024.02.1

workflows.user.role.group.remove

This event can be used by any admin or security team member to monitor the removal of Workflows roles from an Okta group. The payload provides information about both the group from which the role was removed and the role that was removed. Related events include workflows.user.role.group.add, workflows.user.role.user.add, workflows.user.role.user.remove, application.user_membership.add, and application.user_membership.remove. The event fires when an admin manually removes a role from an Okta group in the Workflows console. Removing roles in a single action triggers multiple system log events.

`workflows`

Since: 2024.02.1

workflows.user.role.user.add

This event can be used by any admin or security team member to monitor the addition of Workflows roles to an Okta user. The payload provides information about both the user to whom the role was added and the role that was added. Related events include workflows.user.role.user.remove, workflows.user.role.group.add, workflows.user.role.group.remove, application.user_membership.add, and

workflows.user.role.user.remove

This event can be used by any admin or security team member to monitor the removal of Workflows roles from an Okta user. The payload provides information about both the user from whom the role was removed and the role that was removed. Related events include workflows.user.role.user.add, workflows.user.role.group.add, workflows.user.role.group.remove, application.user_membership.add, and application.user_membership.remove. The event fires when an admin manually removes a role from a user in the Workflows console. Removing multiple in a single action triggers multiple system log events.

`workflows`

Since: 2024.02.1

workflows.user.table.create

This event can be used by any admin or security team member to monitor the creation of new tables in the Workflows platform. The target fields provide information on the user that created the table and the new table. Other table lifecycle events include: workflows.user.table.view, workflows.user.table.update, and workflows.user.table.delete. Note that this event doesn't fire when a table is imported. For that, users can reference workflows.user.table.import or workflows.user.folder.import.

`workflows`

Since: 2021.02.1

workflows.user.table.delete

This event can be used by any admin or security team member to monitor when a user deletes a table from the Workflows platform. The target fields provide information on the user that deleted the table and the tabl itself. Other table lifecycle events include: workflows.user.table.view, workflows.user.table.update, and workflows.user.table.create.

`workflows`

Since: 2021.02.1

workflows.user.table.export

`workflows`

Since: 2021.02.1

workflows.user.table.import

This event can be used by any admin or security team member to monitor when a user imports table data into the Workflows platform using the Tables interface. The target fields provide information on the user that imported the table and the table itself. Related events include: workflows.user.table.export, workflows.user.folder.export, and workflows.user.folder.import. Note that importing through the table interface requires an existing schema and is used to import the data from a .csv file. This event does not fire as part of workflows.user.folder.import.

`workflows`

Since: 2021.02.1

workflows.user.table.move

This event can be used by any admin or security team member to monitor users moving tables between folders on the Workflows platform. The payload provides information on the user that moved the table and the table that was moved. Other Workflows resource move events include workflows.user.folder.move and workflows.user.flow.move. Note that this event fires when a user manually drags a table from one folder to another folder. Additional information including old and new folder locations can be found in the debug context field.

`workflows`

Since: 2024.07.0

workflows.user.table.schema.export

This event can be used by any admin or security team member to monitor when a user exported a table schema from the Workflows platform. The payload provides information on the user that exported the table schema and the table that was exported. Other related table events include: workflows.user.table.create, workflows.user.table.delete, workflows.user.table.update, workflows.user.table.view, workflows.user.table.import, workflows.user.table.export, and workflows.user.table.schema.import. This event fires when a user exports a folder that contains a table.

This event can be used by any admin or security team member to monitor when a user has imported a table schema into the Workflows platform. The payload provides information on the user that imported the schema and the table that was created from that schema. Other related table events include: workflows.user.table.create, workflows.user.table.delete, workflows.user.table.update, workflows.user.table.view, workflows.user.table.import, workflows.user.table.export, and workflows.user.table.schema.export. This event fires when a user imports a folder that contains a table.

`workflows`

Since: 2023.06.1

workflows.user.table.update

This event can be used by any admin or security team member to monitor when a user updates a table's schema on the Workflows platform. The target fields provide information on the user that updated the table and the table itself. Other table lifecycle events include workflows.user.table.view, workflows.user.table.create, and workflows.user.table.delete. Note that this event does not include information about what was updated, only that the table name or columns were modified. It does not fire when the table data itself is updated.

`workflows`

Since: 2021.02.1

workflows.user.table.view

This event can be used by any admin or security team member to monitor the viewing of table data in the Workflows platform. The target fields provide information on the user that viewed the table and which table was viewed. Other table lifecycle events include: workflows.user.table.create, workflows.user.table.update, and workflows.user.table.delete. Note that this event only fires when a user manually accesses a table. It does not fire when table data is accessed using the Workflows Table functions.

`workflows`

Since: 2021.02.1

zone.activate

Network zone activate.

`network-zone`

network-zone

Since: 2017.49

zone.deactivate

Network zone deactivate.

network-zone

Since: 2017.49

zone.delete

Network zone delete.

network-zone

Since: 2017.49

zone.make_blacklist

Network zone mark as blacklist.

network-zone

Since: 2017.49

zone.remove_blacklist

Network zone unmark as blacklist.

network-zone

Since: 2017.49

zone.update

Network zone update.

network-zone

Since: 2017.49

okta Developer

Questions? Ask us on the forum.

OKTA.COM >
Products, case studies, resources

HELP CENTER >
Knowledgebase, roadmaps, and more

TRUST >
System status, security, compliance

Contact & Legal

Contact our team
Contact sales

More information

Integrate with Okta
Pricing

Feedback