Security Analytics Integrations with Okta

Okta Inc.
301 Brannan Street, 3rd Floor
San Francisco, CA, 94107

info@okta.com
1-888-722-7871

v1.0
May 2017

Table of Contents

What is Okta

Okta is the secure foundation for connections between people and technology. With offerings like Single Sign-on (SSO), Lifecycle Management (LCM), Adaptive Multi-Factor Authentication (MFA), Universal Directory (UD) and API Access Management, Okta is a cloud enabling platform that is paving the way for fast and wide adoption of cloud services in the enterprise. The power of Okta's core identity services are also available to software developers and integrators through our developer platform product (http://developer.okta.com/).

What are common uses case where my customers would benefit from identity context?

Using our logs, you can ingest activities flowing through Okta for the purposes of:

- Displaying within your dashboards
- Understanding user behavior in the cloud
- Creating incidents or alerts based on observations

Using our other endpoints, you can discover additional information about users to:

- Cross reference different identity expressions (e.g., an application referring to John Doe, Jonathan Doe, Jdoe, john.doe etc.)
- Identify relationships between users
- Discover any number of user profile attributes

It is also possible to write back to Okta for the purposes of adapting access based on risk scores:

- Suspend user access
- Enforce multi-factor authentication
- Remove privileged applications
- Leverage our Multi-Factor authentication

What do I do now?

Build	Build an integration using these guidelines and following the requirements below.
Get Approved	Submit a request to Okta's partner team at bd-isv@okta.com to have your integration reviewed. Please provide the following: • Configuration guide • Datasheet • Technical and business contact information • Support escalation plan
Market	Engage in joint go-to-market opportunities, potentially including: • Being listed on okta.com • Field and channel enablement • Referral fees • Joint webinars • Joint events

The Solution

How do I interact with Okta?

Okta has well documented public API endpoints (http://developer.okta.com/documentation/). In this document, we will discuss partner integration guidance that builds on that documentation.

Common Guidance and Requirements

Environment Setup

All configurations will need to provide a customer the ability to define their base URL and API key. The base URL will be used as the basis for building organization specific RESTful URL's and the API key is included in the Authorization header of requests to authenticate the interaction. The API key is to be considered extremely sensitive and controls should be put in place to protect it in the same manner that a password would be protected.

More on getting setup

http://developer.okta.com/docs/api/getting_started/api_test_client.html
http://developer.okta.com/docs/api/getting_started/design_principles.html#authentication

User-Agent

To provide for visibility into usage patterns and adoption of integrations we **require** partners to use a distinct and agreed upon User-Agent string. Generally, this would look something like:

> <company>/<version> (e.g. *Acme/1.0*)
> or
> <product>/<version> (*e.g. SuperSIEMNexGen/2.0*)

Please work with as at bd-isv@okta.com to register and track the integration.

More on User-Agents

http://developer.okta.com/docs/api/getting_started/design_principles.html#user-agent

Pagination

Most queries to endpoints that returns lists will require support for pagination. This **must** be incorporated into all development. Different endpoints will have different suggested page sizes. Please refer to the endpoint specific documentation below for that guidance.

More on Pagination

http://developer.okta.com/docs/api/getting_started/design_principles.html#pagination

Rate Limiting

Variable rate limits are applied to all requests. Care should be taken in all development to ensure that rate limits are observed and handled. Rate limits are applied on an Okta Org level meaning queries coming from clients outside of your purview will affect you. Throttling and error handling should be incorporated, as error codes are returned when rate limits are exhausted.

More on Rate Limiting

http://developer.okta.com/docs/api/getting_started/design_principles.html#rate-limiting

Intervals and Filters

To protect both parties from wasting resources we have recommended guidelines for polling intervals and default filters to apply. Please refer to the blue Best Practices sections for each endpoint to review the guidance specific to that endpoint.

	User Agent	Page Size	Interval	Delta Polling	Rate Limited	More Resources
Logs	Required	100	300	Yes	Yes *(60/Minute)*	System Log API
Users	Required	200	86400	Yes	Yes	Users API
Groups	Required	1000	86400	Yes	Yes	Groups API
Apps	Required	20	86400	No	Yes	Apps API
AppUser	Required	20	-	No	Yes	
AppGroup	Required	20	-	No	Yes	

Endpoint Specific Details

When interacting with Okta there are a variety of different types of data you can retrieve and interact with. This document is structured in such a way that each different data type is described individually by the endpoint (URI) that is used to interact with it.

Logs

Our System Log API provides more functionality than the Events API, including more flexible query parameters and more context provided in the log object returned. See the reference section at the end of the document for information on migrating from the Event API to the System Log API.

Authentication events, user profile updates, user state changes, application and group assignment, Okta platform changes and a host of additional information is available through the logs, which clearly describe the actor, action and targets.

This endpoint is most relevant for SIEM, UEBA or a CASB ingesting all log entries from Okta. Other specialized cases exist where tightly scoped queries are made to detect a specific condition or an ad-hoc query to provide enriched context to an incident (security or operational) investigation.

Best Practices

In addition to the Common Guidance offered above the System Log API carries these unique guidelines.

The page size (*limit parameter*) should be a configurable value with a range between 10 and 100, the default value should be 100.

The Interval of ongoing polling should be configurable. Frequent polling of new logs is preferred and a suggested of interval between 5 and 300 seconds is encouraged. A maximum interval of 1 day should be enforced.

Tightly tied to the interval is the idea of doing delta and date bound queries. Ongoing queries should use the events previously returned as a “cursor” and filter for events published after the newest timestamp already collected. To build an event collector that will function well with backlogs, historical data and current data, please see the following guidance:

For polling logs all queries should be date bookended queries using these parameters:

- ?since=StartDate&until=EndDate
- No greater than 1 day gap between StartDate and EndDate

The computation of StartDate should be one of:

- A configurable or computed date no greater than 180 days in the past
- The largest/newest timestamp already present in the system

The computation of EndDate should be

- 1 day beyond that of StartDate.

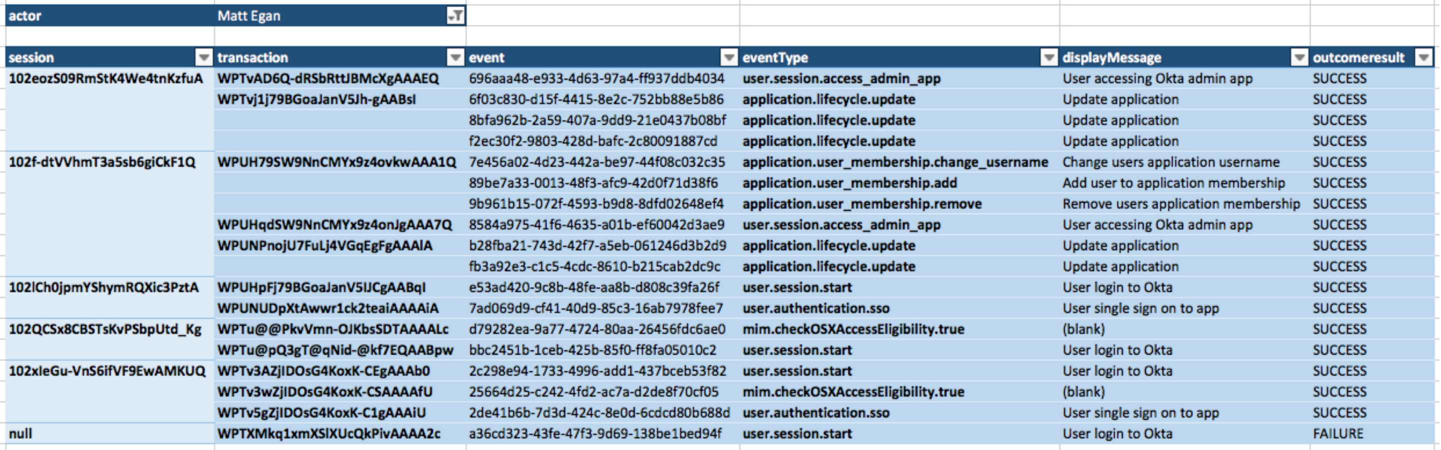
eventType Namespace

The log eventType property is a hierarchical string representing a consistent parent.sublevel.action allowing for flexible queries and human readability. Using SCIM filter syntax Searching for eventType co ".create" will return creation events for all object types while searching for eventType sw "user.authentication." This will return all activities relating to user authentication.

Event Stitching

Throughout a session many requests can occur, and within a given request many events may be logged.

To illustrate this principle, the table below shows 18 events produced from 13 transactions over 6 different sessions, all performed by one user.



actor	Matt Egan

session	transaction	event	eventType	displayMessage	outcomeresult
102eozS09RmStK4We4tnKzfuA	WPTvAD6Q-dRSbRttJBMcXgAAAEQ	696aaa48-e933-4d63-97a4-ff937ddb4034	user.session.access_admin_app	User accessing Okta admin app	SUCCESS
	WPTvj1j79BGoaJanV5Jh-gAABsI	6f03c830-d15f-4415-8e2c-752bb88e5b86	application.lifecycle.update	Update application	SUCCESS
		8bfa962b-2a59-407a-9dd9-21e0437b08bf	application.lifecycle.update	Update application	SUCCESS
		f2ec30f2-9803-428d-bafc-2c80091887cd	application.lifecycle.update	Update application	SUCCESS
102f-dtVVhmT3a5sb6giCkF1Q	WPUH79SW9NnCMYx9z4ovkwAAA1Q	7e456a02-4d23-442a-be97-44f08c032c35	application.user_membership.change_username	Change users application username	SUCCESS
		89be7a33-0013-48f3-afc9-42d0f71d38f6	application.user_membership.add	Add user to application membership	SUCCESS
		9b961b15-072f-4593-b9d8-8dfd02648ef4	application.user_membership.remove	Remove users application membership	SUCCESS
	WPUHqdSW9NnCMYx9z4onJgAAA7Q	8584a975-41f6-4635-a01b-ef60042d3ae9	user.session.access_admin_app	User accessing Okta admin app	SUCCESS
	WPUNPnojU7FuLj4VGqEgFgAAAlA	b28fba21-743d-42f7-a5eb-061246d3b2d9	application.lifecycle.update	Update application	SUCCESS
		fb3a92e3-c1c5-4cdc-8610-b215cab2dc9c	application.lifecycle.update	Update application	SUCCESS
102lCh0jpmYShymRQXic3PztA	WPUHpFj79BGoaJanV5IJCgAABqI	e53ad420-9c8b-48fe-aa8b-d808c39fa26f	user.session.start	User login to Okta	SUCCESS
	WPUNUDpXtAwwr1ck2teaiAAAAiA	7ad069d9-cf41-40d9-85c3-16ab7978fee7	user.authentication.sso	User single sign on to app	SUCCESS
102QCSx8CBSTsKvPSbpUtd_Kg	WPTu@@PkvVmn-OJKbsSDTAAAALc	d79282ea-9a77-4724-80aa-26456fdc6ae0	mim.checkOSXAccessEligibility.true	(blank)	SUCCESS
	WPTu@pQ3gT@qNid-@kf7EQAABpw	bbc2451b-1ceb-425b-85f0-ff8fa05010c2	user.session.start	User login to Okta	SUCCESS
102xIeGu-VnS6ifVF9EwAMKUQ	WPTv3AZjIDOsG4KoxK-CEgAAAb0	2c298e94-1733-4996-add1-437bceb53f82	user.session.start	User login to Okta	SUCCESS
	WPTv3wZjIDOsG4KoxK-CSAAAAfU	25664d25-c242-4fd2-ac7a-d2de8f70cf05	mim.checkOSXAccessEligibility.true	(blank)	SUCCESS
	WPTv5gZjIDOsG4KoxK-C1gAAAiU	2de41b6b-7d3d-424c-8e0d-6cdcd80b688d	user.authentication.sso	User single sign on to app	SUCCESS
null	WPTXMkq1xmXSlXUcQkPivAAAA2c	a36cd323-43fe-47f3-9d69-138be1bed94f	user.session.start	User login to Okta	FAILURE

In a Log data object the following mappings apply:

	Log Data Model Location	Description
session	authenticationContext.externalSessionId	External Session identifier of the request
transaction	transaction.id	ID of the transaction object
event	uuid	Randomly Generated Unique Identifier for event

Data

Please refer to our online documentation for a detailed description of the Log Data Model (http://developer.okta.com/docs/api/resources/system_log.html#log-model)

Events for failed authentication attempts and failed multifactor verification attempts are potential indicators of abuse. Additional context provided in the log will allow for pivoting this information based on things like Target User, Client IP address, Geography, User-Agent and more.

Below are truncated examples of specific logs to illustrate the log object structure and the data it contains. In these examples, we will use the following semantics:

- An Okta compatible filter that would return the example event will be included
- An ellipsis ("…") is used to signify truncated data
- Object properties will be referenced using Dot notation

Authentication Failure

`filter=(eventType eq "user.session.start" AND outcome.result eq "FAILURE")`

```
{
 "uuid": "91c25327-2d83-46d8-8e25-5ef7d9c615db",
 "published": "2017-04-12T18:56:31.884Z",
 "eventType": "user.session.start",
 "displayMessage": "User login to Okta",
 "severity": "WARN",
 "version": "0",
 "outcome": {"result": "FAILURE", "reason": "INVALID_CREDENTIALS"},
 "actor": {"id": "00u1785fc892YN6Tb1d8", "type": "User",
           "alternateId": "some.user@oktaprise.com",
           "displayName": "Some User", "detailEntry": null
          },
 "client": {"userAgent": {"rawUserAgent": "Mozilla/5.0 ...", "...": "..."},
            "ipAddress": "208.223.254.2", "geographicalContext": {"...": "..."},
            "...": "..."
           },
 "authenticationContext": {"externalSessionId": "unknown", "...": "..."},
 "securityContext": {"...": "..."},
 "debugContext": {"debugData": {"requestUri": "/auth/saml20/0oazgLZgct51d8"}},
 "transaction": {"type": "WEB", "id": "WO54X5uSLmaaPH7DtL7wAAzY", "detail": null},
 "request": {"ipChain": [{"...": "..."}]},
 "target": null
}
```

Note that there is no `authenticationContext.externalSessionId` *set. This value is only populated on a successful login.*

From here you can easily start to pivot off the various pieces of information contained.

- Is this an isolated event for the user (actor.id)?
- Are we seeing this same client (`client.ipAddress`) fail for many users?
- Are we seeing morphing client details (`client.*`) for a single user?

Single Sign on to app

`filter=(eventType eq "user.authentication.sso" AND outcome.result eq "SUCCESS")`

```
{
 "uuid": "fdbfe325-dcbb-4380-a778-eb85d0a2c182",
 "published": "2017-04-12T20:55:53.515Z",
 "eventType": "user.authentication.sso",
 "displayMessage": "User single sign on to app",
 "severity": "INFO",
 "version": "0",
 "outcome": {"result": "SUCCESS", "reason": null},
 "actor": {"id": "00u1ae58uup0y5Qkg1d8", "type": "User",
 "alternateId": "some.user@oktaprise.com", "displayName": "Some User",
 "detailEntry": null},
 "client": {"...": "...", "ipAddress": "208.223.254.3"},
 "authenticationContext": {"...": "...", "externalSessionId": "102-EXKTC7eQGeB2NoV7pdHmw"},
 "securityContext": {"...": "..."},
 "debugContext": {"debugData": {"initiationType": "IDP_INITIATED", "...": "..."}},
 "transaction": {"type": "WEB", "id": "WO6UWSeI@4LKXKr0eDMlWQAAA84", "detail": {}},
 "request": {"ipChain": [{"...": "..."}]},
 "target": [
   {"id": "0oa1a35nndfI7w1yp1d8", "type": "AppInstance",
    "alternateId": "Work by FB", "...": "...", "detailEntry": {"signOnModeType": "SAML_2_0"}
   },
   {"id": "0ua1ae5gw9d37jzgJ1d8", "type": "AppUser", "alternateId": "suser@newoktaprise.com",
    "displayName": "Some User", "detailEntry": null
   }
  ]
}
```

A successful IdP-initiated Sign On into an app called “Work by FB”:

- Note the `target[1].alternateId` is different than the `actor.alternateId`
  - The `target[1].alternateId` is the username in the downstream system, is this abuse?

- Pivoting linking other logs with the same `authenticationContext.externalSessionId` can provide:
  - Context about how and when the user was initially authenticated
  - Insight into potential client roaming
  - Insight into other applications used during the same session

Over time you can paint a rich picture of user and endpoint behavior on and off your network.

Some less obvious events to utilize are listed here:

- User state changes (`eventType sw "user.lifecycle."`)
- Group membership changes (`eventType sw "group.user_membership."`)
- Directory agent events (`eventType sw "system.agent."`)
- Application provisioning events (`eventType sw "application.provision."`)
- Application username changes (`eventType eq "application.user_membership.change_username"`)
- Password Reset events (`eventType eq "user.account.reset_password"`)
  - An eventType of (system.email.password_reset.sent_message or system.sms.send_password_reset_message) without a subsequent user.account.reset_password for that user is an indicator of an abandoned password reset attempt and potential abuse.
- MFA lifecycle events (`eventType sw "user.mfa.factor."`)

Tracking activities like these can identify both operational and security risks.

More

http://developer.okta.com/docs/api/resources/system_log.html

Users

At the heart of the Okta Identity Cloud is the User object. This object exhaustively describes the user including:

- Dates related to various updates,
- Credential information,
- Current state, and
- An extensible user schema.

This information can provide value to any integration seeking to provide user context.

Best Practices

In addition to the Common Guidance offered above the Users API carries these unique guidelines.

The page size (*limit parameter*) should be a configurable value with a range between 10 and 200, the default value should be 200.

The interval of ongoing polling should be configurable. Frequent polling of user objects is generally discouraged and only warranted with strict stipulations described below. While user objects and associated profiles are volatile they are not fluid. Consider the cost/benefit associated with queries you perform.

If your goal is to populate and synchronize an external system with Okta identities a SCIM integration might be warranted. Please review our online resource for more information (http://developer.okta.com/standards/SCIM).

When using the API to sync user data with an external system keep in mind the desired outcome of the integration and perform delta queries using the most appropriate date filter, or query the Logs API to watch for user authentication, lifecycle and profile events.

field	associated eventType	Note
created	user.lifecyle.create	When the user was created
activated	user.lifecycle.activate	When the user was last activated
statusChanged	user.lifecycle.*	the timestamp of the most recent state change
passwordChanged	user.account.update_password	the timestamp of the most recent password
lastUpdated	user.account.update_profile or any from above	the timestamp of the most recent profile, password or state change
lastLogin	user.authentication.auth*	the timestamp of the most recent

When polling for users, queries should be date driven using the search and filter capabilities

- ?filter=lastUpdated gt StartDate
- StartDate being the time of the last polling interval
- lastUpdated being the most granular date value from the list above to suite your needs

With proper filtering the interval used becomes less of an issue, as an integration only interested in credential changes or login activity should filter accordingly and ignore irrelevant churn.

Synchronization jobs should, at a minimum, introduce filters on lastUpdated to ongoing queries to minimize needless sifting.

Data

State

In addition to the attributes discussed in the filtering guidelines above the User object has a status attribute. Refer to the online documentation describing the controlling state machine. (http://developer.okta.com/docs/api/resources/users.html#user-status).

Profile

A Universal Directory enabled Okta Org features an extensible schema with the ability to source and master data from many sources including Applications and Directories. Information related to the user's organizational role, hierarchy, geographic location and more can be found in the user profile. The schema is extensible and the level of detail contained is based entirely on the customer's implementation.

The default attributes of a user are aligned with core SCIM attributes and listed here: (http://developer.okta.com/docs/api/resources/users.html#default-profile-properties)

Retrieve a user (or collection of users) using one of the following methods:

Ambiguous Search

Use the q (query) parameter to search across multiple attributes to find users (http://developer.okta.com/docs/api/resources/users.html#find-users)

```
Request
GET {base_url}/api/v1/users/?q=John

Response
HTTP/1.1 200 OK
[
 {
        "id": "00u1aq5mpenI88ZEn1d8",
        "status": "ACTIVE",
        "lastUpdated": "2017-04-17T23:16:50.000Z",
        "…",
        "profile":
        {
                "firstName": "John",
                "lastName": "Doe",
                "login": "jdoe@domain.tld",
                "email": "Joh.Doe@company.tld",
                "…"
        },
        "…"
 },
 {"…"}
]
```

Targeted Search

Use our filter and search capabilities to locate users with greater accuracy and flexibility (http://developer.okta.com/docs/api/resources/users.html#list-users-with-a-filter)

```
Request
GET {base_url}/api/v1/users/?filter=(profile.firstName eq "John" AND profile.lastName eq "Doe")

Response
HTTP/1.1 200 OK
[
 {
        "id": "00u1aq5mpenI88ZEn1d8",
        "status": "ACTIVE",
        "lastUpdated": "2017-04-17T23:16:50.000Z",
        "…",
        "profile":
        {
                "firstName": "John",
                "lastName": "Doe",
                "login": "jdoe@domain.tld",
                "email": "Joh.Doe@company.tld",
                "…"
        },
        "…"
 }
]
```

Get Single User

Retrieve a single user based on the user's:

- username (*login*) (*jdoe@domain.tld*)
- Short username (*jdoe*)
- Okta id (*00u1aq5mpenI88ZEn1d8*)

Using Get User (http://developer.okta.com/docs/api/resources/users.html#get-user)

```
Request
GET {base_url}/api/v1/users/jdoe
Or
GET {base_url}/api/v1/users/jdoe@domain.tld
Or
GET {base_url}/api/v1/users/00u1aq5mpenI88ZEn1d8

Response
HTTP/1.1 200 OK
 {
        "id": "00u1aq5mpenI88ZEn1d8",
        "status": "ACTIVE",
        "lastUpdated": "2017-04-17T23:16:50.000Z",
        "…",
        "profile":
        {
                "firstName": "John",
                "lastName": "Doe",
                "login": "jdoe@domain.tld",
                "email": "Joh.Doe@company.tld",
                "…"
        },
        ""…""
 }
```

Methods of, and reasons to, manipulate data in a user profile are discussed in our “Write back to enforce policy in Okta” Users section below.

More

http://developer.okta.com/docs/api/resources/users.html

Groups

Groups are a first-class citizen in the Okta environment. All the standard uses of Groups are leveraged within Okta and subsequently extended to orbiting applications and directories. They serve purposes including but not limited to:

- Application Assignment,
- Application Role/License/Entitlement and
- Policy Assignment.

Best Practices

In addition to the Common Guidance offered above the Groups API carries these unique guidelines.

The page size (*limit parameter*) should be a configurable value with a range between 100 and 10000, the default value should be 10000.

The interval of ongoing polling should be configurable. Frequent polling of group objects is generally discouraged and only warranted with strict stipulations described below.

When using the API to sync group data and group membership information with an external system keep in mind the desired outcome of the integration.

Tightly tied to the interval used is the idea of doing date bound queries to retrieve delta datasets. There are two Date fields available to determine changes to a group.

- *lastUpdated* is the timestamp when a group's profile was last updated
- *lastMembershipUpdated* is the timestamp when a user was last added to or removed from that group

These values change independently. Membership changes will not modify the lastUpdated timestamp.

When polling for groups and group changes, queries should be date driven using the search and filter capabilities:

- ?filter=lastMembershipUpdated gt StartDate
- StartDate being the time of the last polling interval
- lastMembershipUpdated being the granular date value from the list above to suite your needs

With proper filtering the interval used becomes less of an issue, as an integration only interested in group membership changes would filter accordingly and ignore irrelevant churn.

Synchronization jobs should, at a minimum, introduce filters on lastMembershipUpdated.

Data

In addition to hosting native groups, Okta can source and replicate group membership between directories and apps. Every group object in Okta will contain a Type property that describes the source of the group. The profile of a group will vary based on the source of the group. Different app groups have different profiles.

Groups will always have a name, description and sufficient context to identify and associate back to their source. For example, an Active Directory group has an externalId attribute that is the AD groups Object-Guid.

Group membership manipulation is the de facto standard for affecting user entitlements and restrictions including but not limited to:

- Application Assignment
- Application Role
- Application License
- Authentication Policy
- Password Policy

Retrieve a Group with stats and app details

Using Get Group (http://developer.okta.com/docs/api/resources/groups.html#get-group) you can also add a query parameter expand with a value of app and or stats. The result is a single call with additional details about the group. This method works when getting a singular group by id and when listing groups with or without a filter or query applied.

```
Request
GET {base_url}/api/v1/groups/?limit=100&expand=app,stats
or
GET {base_url}/api/v1/groups/?filter=lastMembershipUpdated gt 2017-04-17T23:16:50.000Z
&expand=app,stats
or
GET {base_url}/api/v1/groups/00gwy337uaRYJVHTHACG?expand=app,stats

Response
HTTP/1.1 200 OK
[
 {
        "id": "00gwy337uaRYJVHTHACG",
        "lastUpdated": "2017-03-31T23:16:50.000Z",
        "lastMembershipUpdated": "2017-04-17T23:16:50.000Z",
        "…": "…"
        "type": "APP_GROUP",
        "…",
        "profile": {
                "name": "Domain Group 1",
                "…": "…",
                "externalId": "nYGCoOeiW0uRVHW3MQcB1Q=="
        },
        "_embedded": {
                "app": {
                        "id": "0oarja7d8gWSEGZBPZVB",
                        "name": "active_directory",
                        "…": "…"
                },
                "stats": {
                        "usersCount": 301,
                        "appCount": 0,
                        "groupPushMappingsCount": 0,
                        "…": …"
                }
        },
        "…"
 },
 {…}
]
```

Retrieve Group Members

Using the same logic described online in List Group Members (http://developer.okta.com/docs/api/resources/groups.html#list-group-members) you can retrieve a list of users in each group. *Hint: use the group._embedded.stats.usersCount value to know if ANY users are assigned*

If your integration doesn't need credential and credential provider related details when listing group members use the "skinny_users" endpoint, it has the following differences from the full "users" endpoint:

- credentials.provider object missing
- _links object only contains .self reference

```
Request
GET {base_url}/api/v1/groups/00gwy337uaRYJVHTHACG/skinny_users
or
GET {base_url}/api/v1/groups/00gwy337uaRYJVHTHACG/users

Response
HTTP/1.1 200 OK
[
  {
    "id": "00u10eqzrjiGORRUNTBM",
    "status": "ACTIVE",
    "created": "2015-01-07T01:52:51.000Z",
    "…": "…",
    "profile": {
      "login": "jsmith@oktaprise.com",
      "email": "jane.smith@oktaprise.com",
      "…": "…"
    },
    "credentials": {
      "recovery_question": {
        "question": "What is the name of your first stuffed animal?"
      }
    },
    "_links": {
      "self": {
        "href": "https://oktaprise.okta.com/api/v1/users/00u10eqzrjiGORRUNTBM"
      }
    }
  },
  {
    "id": "00u10h0t5suAYARMBTGF",
    "status": "ACTIVE",
    "created": "2015-01-09T05:28:55.000Z",
    "…": "…",
    "profile": {
      "login": "jdoe@oktaprise.com",
      "email": "joshua.kroeze@oktaprise.com",
      "…": "…"
    },
    "credentials": {
      "recovery_question": {
        "question": "What was the first thing you learned to cook?"
      }
    },
    "_links": {
      "self": {
        "href": "https://oktaprise.okta.com/api/v1/users/00u10h0t5suAYARMBTGF"
      }
    }
  }
]
```

Retrieve Apps Assigned by a Group

Using the logic described online with List Assigned Applications (http://developer.okta.com/docs/api/resources/groups.html#list-assigned-applications) you can retrieve a collection of applications that are assigned based on membership of that group. *Hint: use the group._embedded.stats.appsCount value to know if ANY apps are assigned*

```
Request
GET {base_url}/api/v1/groups/00gwy337uaRYJVHTHACG/apps

Response
HTTP/1.1 200 OK
[
  {
    "id": "0oa9lv1b0tRF7p34K0h7",
    "name": "scim2headerauth",
    "label": "SCIM 2.0 Test App (Header Auth)",
    "status": "ACTIVE",
    "…": "…",
  },
  {
    "id": "0oa9sieay0Tju66dM0h7",
    "name": "github_enterprise",
    "label": "GitHub Business2",
    "status": "ACTIVE",
    "…": "…",
  },
  {
    "id": "0oaa097p4wlH2q8To0h7",
    "name": "servicenow_ud",
    "label": "ServiceNow UD",
    "status": "ACTIVE",
    "…": "…",
  }
]
```

Methods of, and reasons to, manipulate Groups and group membership are discussed in our “Write back to enforce policy in Okta” Groups section below.

More

http://developer.okta.com/docs/api/resources/groups.html

Apps

Apps are the representation of an application or directory source or target in Okta. Specific application attributes are defined on an app by app basis. Through the Apps endpoint you gain additional insight into who a user is across the ecosystem. We also see the meaning behind a group and the roles and entitlements it describes.

Best Practices

In addition to the Common Guidance offered above, the Apps API carries these unique guidelines.

The page size (*limit parameter*) should be a configurable value with a range between 10 and 100. The default value should be 20.

The interval of ongoing polling should be configurable. Frequent polling of apps can provide little value on its own. While not resource intensive an interval minimum of daily would be acceptable.

Further guidance about resources inside of the Apps endpoint are discussed in the AppUsers and AppGroups sections below.

Data

The application data model is described in detail online (http://developer.okta.com/docs/api/resources/apps.html#application-properties)

Here we will draw attention to a few App attributes and their meaning in the context of a security analytics integration:

- features
  - This collection will describe the provisioning capabilities that are in effect for the application
  - http://developer.okta.com/docs/api/resources/apps.html#features

- signOnMode
  - This object will describe what method of authentication (if any) is in place for the application
  - http://developer.okta.com/docs/api/resources/apps.html#signon-modes

- credentials.scheme
  - For applications where Okta is vaulting credentials for application, using this attribute describes how they are managed
  - http://developer.okta.com/docs/api/resources/apps.html#authentication-schemes

Retrieve a list of Active Applications

Using a simple status filter to list the active Applications (http://developer.okta.com/docs/api/resources/apps.html#list-applications)

```
Request
GET {base_url}/api/v1/apps?filter=status eq "ACTIVE"

Response
HTTP/1.1 200 OK
[
  {
    "id": "0oa8tvgiuh93NlQ0W0h7",
    "name": "zendesk",
    "label": "Zendesk",
    "status": "ACTIVE",
    "lastUpdated": "2016-11-20T04:01:19.000Z",
    "…": "…",
    "features": [
      "PUSH_NEW_USERS",
      "PUSH_USER_DEACTIVATION",
      "IMPORT_USER_SCHEMA",
      "REACTIVATE_USERS",
      "PUSH_PROFILE_UPDATES",
      "IMPORT_NEW_USERS"
    ],
    "signOnMode": "SAML_2_0",
    "..": "…"
  },
  {
    "id": "0oa11hz4wq4dioDw91e8",
    "name": "bluejeans",
    "label": "BlueJeans",
    "status": "ACTIVE",
    "lastUpdated": "2015-04-17T19:27:36.000Z",
    "…": "…",
    "signOnMode": "BROWSER_PLUGIN",
    "credentials": {
      "scheme": "EDIT_USERNAME_AND_PASSWORD",
      "userNameTemplate": {
        "template": "${source.login}",
        "type": "BUILT_IN"
      },
      "revealPassword": true,
      "signing": {
        "kid": "…"
      }
    },
    "…": "…"
  }
]
```

More

http://developer.okta.com/docs/api/resources/apps.html

AppUsers

Like the User object discussed previously the AppUser object is an extensible and information rich data source. The AppUser object is a representation of a user’s profile specific to the associated application or directory and can include profile data, role, license information and more depending on the type of integration.

Best Practices

The page size (*limit parameter*) should be a configurable value with a range between 10 and 100, the default value should be 20.

A specific need should be present before polling the information present. The data models of AppUsers varies by application and implementation.

If full profile data isn't required please use the "skinny_user" variant. This endpoint is optimized for speed and efficiency but contains less data.

Data

Refer to the Application User data model (http://developer.okta.com/docs/api/resources/apps.html#application-user-properties)

Retrieve a list of AppUsers for an app

Using this method you can list the user assigned to an app which will include their AppUser profile (http://developer.okta.com/docs/api/resources/apps.html#list-users-assigned-to-application)

```
Request
GET {base_url}/api/v1/apps/{app_id}/skinny_users
or
GET {base_url}/api/v1/apps/{app_id}/users
or
GET {base_url}/api/v1/apps/{app_id}/users/{user_id}

Response
HTTP/1.1 200 OK
[
  {
    "id": "00u8tvvbi3eihn3Ur0h7",
    "externalId": "15467658348",
    "created": "2016-11-20T04:04:42.000Z",
    "lastUpdated": "2016-11-20T23:09:40.000Z",
    "scope": "GROUP",
    "status": "PROVISIONED",
    "statusChanged": "2016-11-20T04:04:44.000Z",
    "syncState": "SYNCHRONIZED",
    "lastSync": "2016-11-20T23:09:40.000Z",
    "credentials": {
      "userName": "nick@mytest.oktapreview.com"
    },
    "profile": {
      "Role": "agent",
      "phone": "555-789-1231",
      "Groups": [
        "Support"
      ],
      "alias": "Nick",
      "Organization": "mattegantest",
      "locale": "English",
      "RestrictionId": "all",
      "timeZone": "Pacific Time (US & Canada)",
      "firstName": "Joeseph"
    }
  },
  {…}
]
```

Retrieve Apps and AppUser objects for a user

Using this Method, you can make a single call to retrieve a collection of all application objects with an AppUser object (like shown above) nested within each Application in the _embedded object. This approach will reduce the need to make iterative calls to fully elaborate a user's application footprint (http://developer.okta.com/docs/api/resources/apps.html#list-applications-assigned-to-user).

```
Request
GET {base_url}/api/v1/apps?filter=user.id eq "00u1a7q3KgTkZE1d8"&expand=user/00u1a7q3KgTkZE1d8

Response
HTTP/1.1 200 OK
[
  {
    "id": "0oa8tvgiuh93NlQ0W0h7",
    "name": "zendesk",
    "label": "Zendesk",
    "status": "ACTIVE",
    "..": "…"
    "_embedded": {
      "user": {
        "id": "00u1a7q3KgTkZE1d8",
        "externalId": "Matt.Egan@oktaprise.com",
        "…": "…",
        "profile": {
          "appProperty": "Value",
          "appRole": [ role1,role2 ]
        }
  },
  {…},
  {…}
]
```

More

http://developer.okta.com/docs/api/resources/apps.html#application-user-model

AppGroups

The AppGroups endpoint is useful for gaining a better understanding of the meaning of groups present in Okta. For example, risk scores could be inferred based on access granted through a given group.

Best Practices

The page size (*limit parameter*) should be a configurable value with a range between 10 and 100, the default value should be 20.

Like the guidance given for polling applications, the volatility of groups used to assign applications is low. Daily intervals should suffice in most cases.

Data

Refer to the Application Group data model (http://developer.okta.com/docs/api/resources/apps.html#application-group-model)

Retrieve a list of AppUsers for an app

Using this method you can list the group assignments for an app (http://developer.okta.com/docs/api/resources/apps.html#list-groups-assigned-to-application)

```
Request
GET {base_url}/api/v1/apps/{app_id}/users

Response
HTTP/1.1 200 OK
[
  {
    "id": "00g8tvrpgdnXLHtzi0h7",
    "lastUpdated": "2016-11-20T04:04:03.000Z",
    "priority": 0,
    "profile": {
      "tags": null,
      "CustomRole": "No Custom Role",
      "Role": "agent",
      "Groups": [
        "Support"
      ],
      "timeZone": "Pacific Time (US & Canada)"
    },
    "…": "…"
  },
  {}
]
```

More

http://developer.okta.com/docs/api/resources/apps.html#application-group-model

Write back to enforce policy in Okta ("Close the Loop")

To resolve an incident or mitigate a perceived threat, an external system may want to affect a user's state, modify authentication policies or reduce application availability. This section will describe how and when a system might take these actions.

Information about examples

Simplified Examples

For readability simplified versions of the API transactions will be illustrated below. We use the following modifications:

- Repetitive details like content-type and authorization headers will be excluded
- Request and response objects will be truncated to focus on the most relevant information
  - Truncated sections will be denoted by an ellipsis ("…")
- Replacement values will be noted in italicized braces *{}*
- Examples:
  - *{base_url}* = Base URL of the Okta org (e.g. https://acme.okta.com/)
  - *{user_id}* = Opaque and Immutable Okta ID for a user (e.g. 00u1ae58uup0y5Qkg1d8)
  - *{group_id}* = Opaque and Immutable Okta ID for a group (e.g. 00g1at1k0dzmV839P1d8)

Resolving Okta identifiers

When using logs from Okta to write back to Okta, the values for user_id, group_id and others will be present in the log. They will be found in the actor or target object, and it will describe the type of object and the id of the object.

When using logs from outside of Okta to trigger events, use the lookup and search functions described in the Users and Groups sections above to retrieve the Okta identifier for those objects.

Users

An individual user's state, sessions, credentials and profile can be updated depending on the specific use case.

State

A user's state can be toggled according to a prescribed state machine (http://developer.okta.com/docs/api/resources/users.html#user-status)

Suspending a user is a non-destructive operation that will leave a user profile, credentials, enrolled factors, groups membership and assigned applications intact while still preventing the user from signing into Okta or any subordinate application. **The act of suspending a user will destroy any existing Okta session for that user.**

Suspend a User (http://developer.okta.com/docs/api/resources/users.html#suspend-user)

```
Request
POST {base_url}/api/v1/users/{user_id}/lifecycle/suspend

Response
HTTP/1.1 200 OK
```

To reverse this operation after a threat has been cleared or an internal timer has elapsed you can return a suspended user back to an active state:

Unsuspend a User (http://developer.okta.com/docs/api/resources/users.html#unsuspend-user)

```
Request
POST {base_url}/api/v1/users/{user_id}/lifecycle/unsuspend

Response
HTTP/1.1 200 OK
```

There are other user state operations that can be changed. **Be aware of the full effect of such changes before implementing this feature.** Read our online documentation for more details about user lifecycle operations (http://developer.okta.com/docs/api/resources/users.html#lifecycle-operations)

Sessions

You can clear existing user sessions, forcing a user to authenticate on the next operation. This action may be taken alone to clear 'suspect' sessions or in conjunction with other actions – like changing authentication policies to enforce MFA or expiring a password – to accelerate the enforcement of that change:

Clear User Session (http://developer.okta.com/docs/api/resources/users.html#clear-user-sessions)

```
Request
DELETE {base_url}/api/v1/users/{user_id}/sessions

Response
HTTP/1.1 204 NO CONTENT
```

Credentials

In certain situations, expiring or changing a user's password might also be prudent. Okta provides a consistent mechanism for expiring and changing passwords for users regardless of password authority. For example, when integrated with an on premise Active Directory (AD) expiring or resetting passwords will propagate through Okta to AD with no changes required to the calling client.

While there are more operations you can perform against a user's credentials we will focus on expiring and changing passwords. **Changing or expiring a password does not clear existing sessions for that user.**

To expire a password, forcing a user to change their **current password** the next time they login to a connected system, make this call:

Expire Password (http://developer.okta.com/docs/api/resources/users.html#expire-password)

```
Request
POST {base_url}/api/v1/users/{user_id}/lifecycle/expire_password

Response
HTTP/1.1 200 OK
{
        "id": "{user_id}",
         "…"
}
```

One method of dealing with suspected leaked credentials is to expire and change a password at the same time. As a result, the user is forced to change their password the next time they login using a **randomly generated password**. Depending on the desired behavior there are two ways to perform this action:

Expire a password (http://developer.okta.com/docs/api/resources/users.html#expire-password)

Have Okta generate a random, policy compliant password and communicate it to the user through whatever means are available.

```
Request
POST
{base_url}/api/v1/users/{user_id}/lifecycle/expire_password?tempPassword=True

Response
HTTP/1.1 200 OK
{
        "tempPassword": "HR076g21"
}
```

Reset a password (http://developer.okta.com/docs/api/resources/users.html#reset-password)

Using an Okta password reset link, require the user to proceed through an Okta password reset sequence. You can choose to have Okta send the link to the user via email or collect and provide it to the user through other means.

```
Request
POST {base_url}/api/v1/users/{user_id}/lifecycle/reset_password?sendEmail=False

Response
HTTP/1.1 200 OK
{
        "resetPasswordUrl": "{base_url}/reset_password/XE6wE17zmphl3KqAPFxO"
}
```

Profile

With the backing of Universal Directory, Okta Expression Language and rules based groups, a user profile is also a prime mechanism to enforce policy in Okta:

- Profile mappings can change user state in downstream applications based on an attribute value.
- Similarly, our rules based group memberships can be driven by a variety of inputs, one of which is user attribute values.

The following is one way to benefit from these capabilities:

- You could extend a user profile, for example by adding a health indicator attribute,
- You could then configure policies in Okta based on that attribute value,
- Finally, your system could manipulate that value in Okta based on your system's observations

User attributes are:

- Extensible
- Strongly Typed (string, number, Boolean, integer, etc.)
- Capable of being mapped downstream or leveraged by rules based groups
- Discoverable through our Schemas API (http://developer.okta.com/docs/api/resources/schemas.html#schemas-api)

Even with all this power and flexibility a partial update of a user profile to a known or set of known attributes is a simple operation.

Update Profile (http://developer.okta.com/docs/api/resources/users.html#update-profile-1)

```
Request
POST {base_url}/api/v1/users/{user_id}
{
        "profile":
        {
                "health": "good",
                "other": "value"
        }

}

Response
HTTP/1.1 200 OK
{
        "id": "{user_id}",
        "…",
        "profile":
        {
                "…",
                "health": "good",
                "other": "value",
                "…"
        },
        "…"
}
```

Groups

Groups are a first-class citizen in the Okta environment. They serve purposes including but not limited to:

- Application Assignment,
- Application Role, and
- Policy Assignment.

With that in mind, the simple manipulation of group membership can have far reaching effects to strengthen your customer's security posture. For example:

- Adding a user to a group could enforce a restrictive authentication policy with short session lifetimes that always require MFA policy;
- Removing a user from a group could un-assign a sensitive application or remove a permissive role in a downstream application.

As previously mentioned, the *{group_id}* referenced below would need to be discovered or configured in some way. Refer to the Groups section above for ideas and our online documentation for more details related to working with groups (http://developer.okta.com/docs/api/resources/groups.html)

Add Member

```
Request
PUT {base_url}/api/v1/groups/{group_id}/users/{user_id}

Response
HTTP/1.1 204 NO CONTENT
```

Remove Member

```
Request
DELETE {base_url}/api/v1/groups/{group_id}/users/{user_id}

Response
HTTP/1.1 204 NO CONTENT
```

References

Event API to System Log API

Use the following table for reference when moving from the Events API to the System Log API

Event Property	Log Property
event.eventId	log.uuid
event.sessionId	log.authenticationContext.externalSessionId
event.requestId	log.transaction.id
event.published	log.published
event.action.message	log.displayMessage
event.action.categories[0]	n/a, log.severity and log.outcome contain similar
event.action.categories[1]	n/a, log.severity and log.outcome contain similar
event.action.objectType	log.legacyEventType
event.action.requestUri	log.debugContext.debugData.requestUri
event.actors[0].id	log.client.userAgent.rawUserAgent
event.actors[0].displayName	log.client.userAgent.browser
event.actors[0].ipAddress	log.client.ipAddress
event.actors[0].objectType	log.client.device
event.targets[0].id	log.actor.id
event.targets[0].displayName	log.actor.displayName
event.targets[0].login	log.actor.alternateId
event.targets[0].objectType	log.actor.type

Common Successful Events (eventType, message and Percentage of observation)

eventType	displayMessage	Pct (%)
application.user_membership.add	Add user to application membership	19.19%
user.authentication.sso	User single sign on to app	6.09%
application.user_membership.remove	Remove users application membership	4.99%
user.session.start	User login to Okta	4.56%
application.provision.user.push_profile	Push users profile to external application	4.08%
app.user_management	Successfully imported new member to an app group	3.57%
application.provision.user.sync	Sync user in external application	3.52%
application.user_membership.update	Updated user application property	3.40%
system.agent.ad.realtimesync	Perform RealTimeSync by AD agent	3.34%
user.authentication.auth_via_AD_agent	Authenticate user with AD agent	3.05%

user.session.access_admin_app	User accessing Okta admin app	2.47%
group.user_membership.add	Add user to group membership	1.99%
application.provision.user.push	Push new user to external application	1.86%
user.account.update_profile	Update user profile for Okta	1.49%
user.session.end	User logout from Okta	1.44%
application.lifecycle.update	Update application	1.20%
application.provision.user.verify_exists	Verify user exists in external application	1.17%
user.authentication.auth_via_mfa	Authentication of user via MFA	1.01%

Common Failure Events (eventType, message and Percentage of observation)

eventType	displayMessage	Pct (%)
user.session.start	User login to Okta	1.75%
system.agent.ad.realtimesync	Perform RealTimeSync by AD agent	1.63%
user.authentication.auth_via_AD_agent	Authenticate user with AD agent	0.89%
user.authentication.auth_via_radius	Authentication of user via Radius	0.25%
user.account.reset_password	User reset password for Okta (by admin)	0.21%
app.generic.unauth_app_access_attempt	User attempted unauthorized access to app	0.07%
system.agent.ad.connect	Connect AD agent to Okta	0.03%
user.authentication.auth_via_mfa	Authentication of user via MFA	0.03%
system.agent.ad.reset_user_password	Perform user password reset by AD agent	0.03%
application.provision.user.push_profile	Push users profile to external application	0.02%
system.agent.ad.invoke_dir	Perform directory invoke command by AD agent	0.02%
application.provision.user.sync	Sync user in external application	0.02%
app.oauth2.as.authorize	OAuth2 authorization request	0.01%
app.oauth2.authorize	OIDC authorization request	0.01%
user.lifecycle.create	Create okta user	0.01%
user.authentication.auth_via_IDP	Authenticate user via IDP	0.01%
application.provision.integration.call_api	Application integration API called	0.00%
system.agent.ad.write_ldap	Perform LDAP write by AD agent	0.00%
app.oauth2.authorize.invalid_client_id	OIDC authorization request	0.00%
app.oauth2.authorize.user_not_assigned	OIDC authorization request	0.00%
system.agent.ad.read_ldap	Perform LDAP read by AD agent	0.00%
system.agent.ad.read_toplogy	(blank)	0.00%
user.account.lock	Max sign in attempts exceeded	0.00%
user.account.update_password	User update password for Okta	0.00%