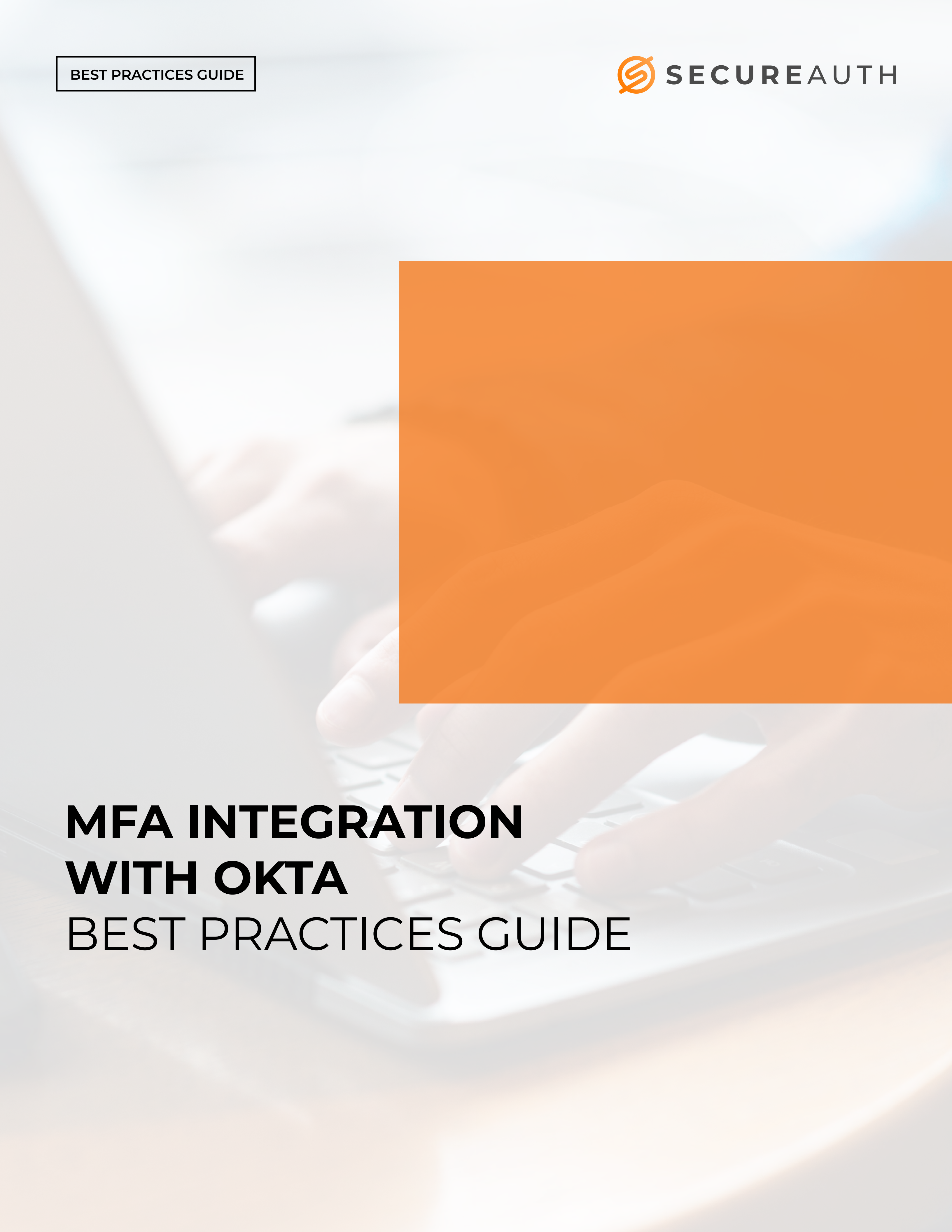
BEST PRACTICES GUIDE

SECUREAUTH



MFA INTEGRATION WITH OKTA

BEST PRACTICES GUIDE

TABLE OF CONTENTS

Introduction

Running SecureAuth IdP multi-factor authentication (MFA) with Okta is seamless and easily implemented. Okta has the ability to act as a service provider for an external IdP and can consume users from active directory if required via Just-in-time (JIT) provisioning . Okta provides an SP init saml for its main authentication page. This means you can configure Okta to redirect to an outside IdP for initial authentication for any user.
Okta has one limitation for this integration that would need to be rectified before moving forward with this integration. Currently, Okta still allows for authentication to the main Okta forms authentication page; Okta would need to block authentication for any given sub-domain and redirect users after the Submit button is pressed.

Okta has APIs to access profile data and validate user information that would enable SecureAuth to utilize Okta-specific profile and membership data when authenticating an on-premise user to the customer's environment during an SP init saml transaction for all Okta- specific users. SecureAuth has performed a similar integration with Azure enabling access to Azure-specific data for on-premise user validation of off-premise data. Alternatively, SecureAuth offers a suite of APIs that can be accessed from the Okta interface, using either local Okta membership and profile data, or on-premise Active Directory data, if needed. This second option can be used if a tighter integration with the Okta cloud application is required. Okta currently has integrations with on-premise RSA tokens and other vendors with a similar function. SecureAuth IdP can be integrated in a like manner using our adaptive, authentication, and IDM APIs.

What follows are a series of use cases, indicating how each scenario would be implemented together with Okta / SecureAuth IdP customizations that must be implemented in order to make each use case functional.

Use Case 1: Okta Inbound SP SAML with Active Directory User Store

This use case describes the preferred and best practices for integration of SecureAuth IdP Login and MFA with Okta SSO using Active Directory credentials.

This use case could be utilized if a customer wanted to use all SecureAuth IdP multi-factor options, adaptive authentication features, APIs, IDM features, authentication protocols, SSO integration capabilities, and Okta SSO function, on-premise directory data such as Active Directory, Okta Group Management, and Okta APIs.

For an illustration of this process, refer to Figure 1, “Internal Active Directory Flow,” on page 5 .

Two possible flows are described together with the Okta customization required to implement them .

Flow 1: Standard SP-Initiated Login URL

1. User attempts to access Okta login via a typical well-know Okta sub-domain URL.
2. User is redirected to SecureAuth with a valid SAML request.
3. User performs a standard userid password and MFA at the SecureAuth login.
4. Data is pulled and validated from Active Directory.
5. User is redirected back to Okta with a valid SAML assertion and allowed access to the Okta portal with the relevant applications.

A further discussion of this process is provided in https://vootsy.oktapreview.com

Flow 2: Forms Login URL

This use case describes the preferred and best practices for integration of SecureAuth Login and MFA with Okta SSO using Active Directory credentials .

This use case could be utilized if a customer wanted to use all SecureAuth multi-factor options, SecureAuth adaptive authentication features, SecureAuth APIs, SecureAuth IDM features, SecureAuth authentication protocols, SecureAuth SSO integration capabilities, Okta SSO function, on-premise directory data such as Active Directory, Okta Group Management, and Okta APIs .

For an illustration of this process, refer to Figure 1, "Internal Active Directory Flow," on page 5 .

Two possible flows are described together with the Okta customization required to implement them .

Flow 1: Standard SP-Initiated Login URL

1. User attempts to access direct forms login via a well-known link.
2. User enters the userid into email address field of Okta login.
3. Okta detects that the user belongs to an SP init SAML-enabled sub-domain in a proper SAML group.
4. User is redirected to SecureAuth IdP with valid SAML request.
5. User performs a standard userid password and MFA at the SecureAuth IdP login.
6. Data is pulled and validated from the Active Directory.
7. User is redirected back to Okta with a valid SAML assertion and allowed access to an Okta portal with available applications

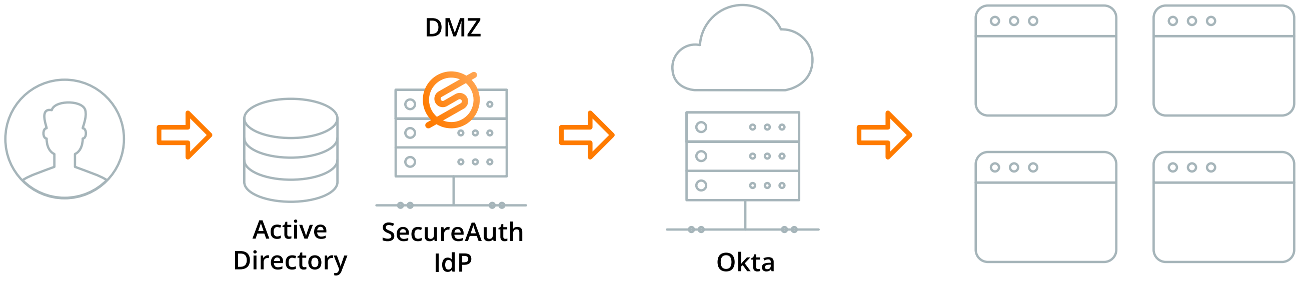
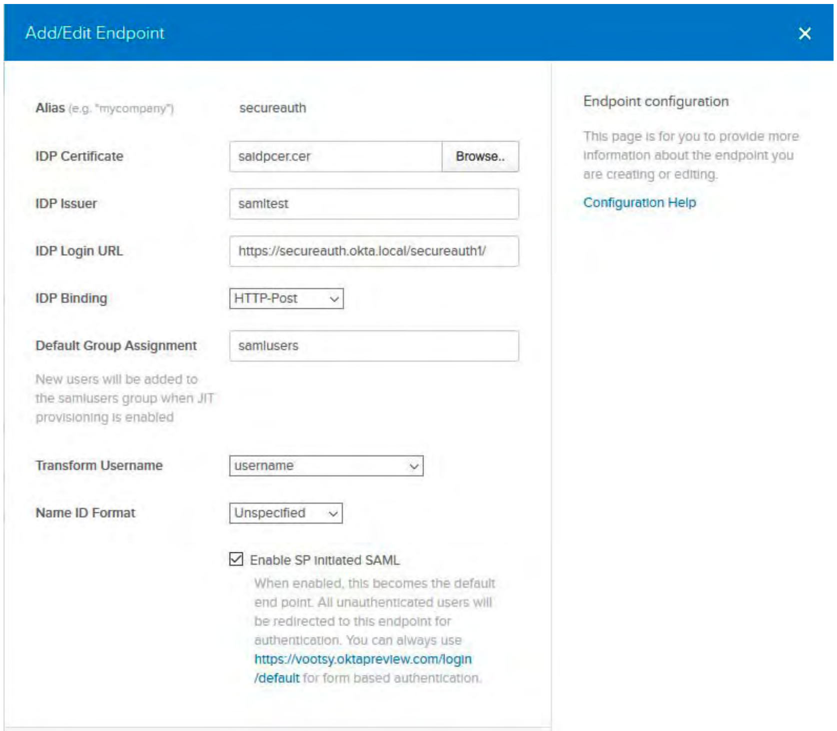


Figure 1. Internal Active Directory Flow

A further discussion of this process is provided in https://vootsy.oktapreview.com/login/default

Required Okta Customizations

Okta would need to customize the outbound SAML configuration to add a check box that disables login for specific groups as well as a field to indicate the groups (this field should support multiple groups) . This would allow Okta to still maintain the default forms logins for administrators and other groups as well (refer to Figure 2, “Okta Configuration Modification (Outbound SAML),” on page 6) .



Add/Edit Endpoint ×

Alias (e.g. "mycompany") secureauth

IDP Certificate saldpcer.cer Browse..

IDP Issuer samltest

IDP Login URL https://secureauth.okta.local/secureauth1/

IDP Binding HTTP-Post

Default Group Assignment samlusers

New users will be added to the samlusers group when JIT provisioning is enabled

Transform Username username

Name ID Format Unspecified

☑ Enable SP initiated SAML

When enabled, this becomes the default end point. All unauthenticated users will be redirected to this endpoint for authentication. You can always use https://vootsy.oktapreview.com/login/default for form based authentication.

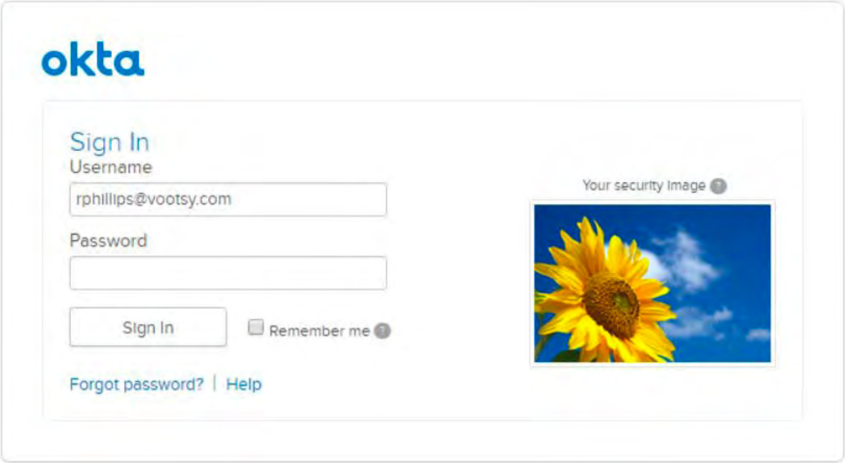
Endpoint configuration

This page is for you to provide more information about the endpoint you are creating or editing.

Configuration Help

Figure 2. Okta Configuration Modification (Outbound SAML)

Okta would also need to customize the main default login to search for users of a specific sub- domain in a specific group, and, if detected, redirect to the selected IdP as shown in Figure 3 . This is standard practice for Salesforce SP, Google SP, Microsoft Azure, and SecureAuth SAML Consumer SP, and is a widely-used procedure when utilizing SP initiation to avoid allowing users to go around a selected IdP .



okta

Sign In

Username rphillips@vootsy.com

Password

Sign In

□ Remember me

Forgot password? | Help

Your security image

Figure 3. Okta Login Modification

Use Case 2: Okta Inbound SP SAML with Okta UserStore

This use case details the preferred and best practice for combining SecureAuth Login and MFA with Okta using Okta data store-specific credentials. For an illustration of this process, refer to Figure 4, "External Okta Data Store Flow via API Connection," on page 8.

This use case would be used if the customer wanted to use all SecureAuth multi-factor options, SecureAuth adaptive authentication features, SecureAuth APIs, SecureAuth IDM features, SecureAuth authentication protocols, SecureAuth SSO integration capabilities, Okta SSO function, Okta SSO user data, Okta group management, and Okta APIs.

Flow 1: Standard SP-Initiated Login URL

1. User attempts to access Okta login via a typical well-known Okta sub-domain URL.
2. User is redirected to SecureAuth with a valid SAML request.
3. User performs a standard userid password and MFA at the SecureAuth login.
4. User data and multi-factor data is validated and retrieved via API from the specified Okta sub-domain.
5. User is redirected back to Okta with a valid SAML assertion and allowed access to the Okta portal with one or more applications.

A further discussion of this process is provided in https://vootsy.oktapreview.com/login/default

Flow 2: Forms Login URL

1. User attempts to access direct forms login via a well-known link.
2. User enters userid into the email address field of the Okta login.
3. Okta detects that the user belongs to SP init SAML-enabled sub-domain and a proper SAML group.
4. User is redirected to SecureAuth IdP with a valid SAML request.
5. User performs standard userid password and MFA at the SecureAuth IdP login.
6. User data and multi-factor data is validated and retrieved via API from the designated Okta sub-domain.
7. User is redirected back to Okta with a valid SAML assertion and allowed access to the Okta portal with the required applications.

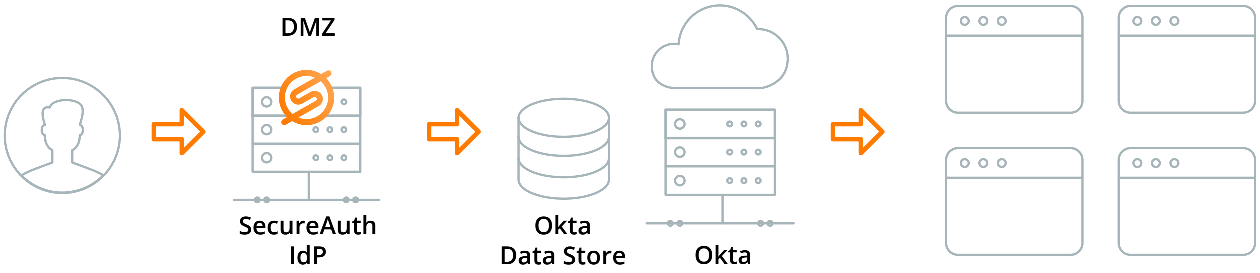
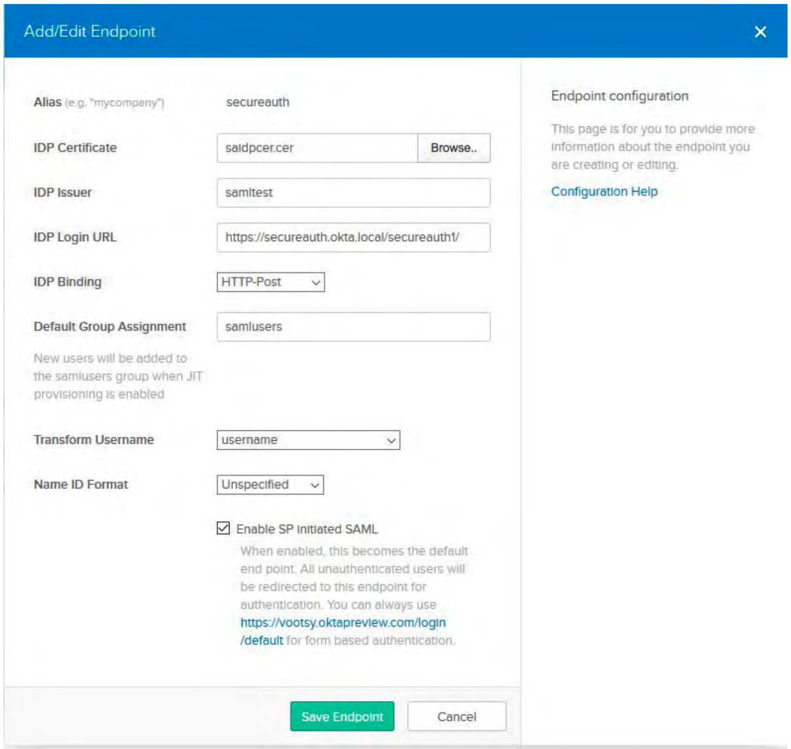


Figure 4. External Data Store Flow via APA Connection

A further discussion of this process is provided in https://vootsy.oktapreview.com/login/default

Required Okta Customizations

In the preceding use case, Okta would need to customize the outbound SAML configuration to include a check box to disable login for specific groups and also add a field to specify one or more required groups (this field should allow for multiple groups) . This would enable Okta to maintain the default forms logins for administrators and other groups as well as shown in Figure 5, “Okta Configuration Modification (Outbound SAML),” on page 9 .



Add/Edit Endpoint

Alias (e.g. "mycompany") secureauth

IDP Certificate: saldpcer.cer

IDP Issuer: samltest

IDP Login URL: https://secureauth.okta.local/secureauth1/

IDP Binding: HTTP-Post

Default Group Assignment: samlusers

New users will be added to the samlusers group when JIT provisioning is enabled

Transform Username: username

Name ID Format: Unspecified

☑ Enable SP Initiated SAML

When enabled, this becomes the default end point. All unauthenticated users will be redirected to this endpoint for authentication. You can always use https://vootsy.oktapreview.com/login/default for form based authentication.

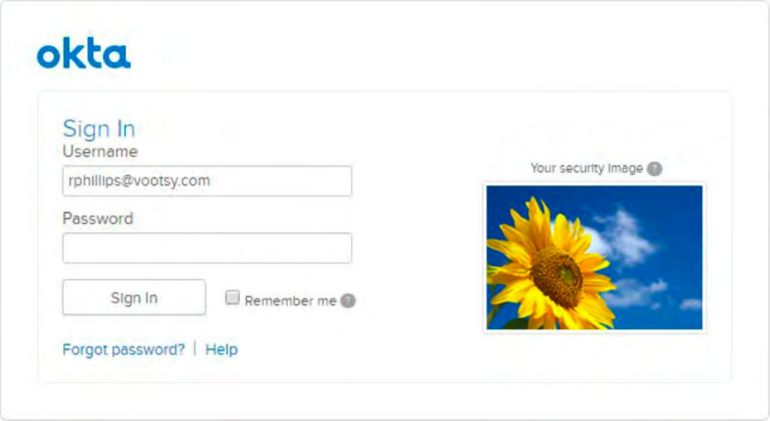
Endpoint configuration

This page is for you to provide more information about the endpoint you are creating or editing.

Configuration Help

Figure 5. Okta Configuration Modification (Outbound SAML)

Okta would also need to customize the main default login to look for users of a specific sub- domain in a specific group, and if detected, the user could be redirected to the selected IdP . This is standard practice for Salesforce SP, Google SP, Microsoft Azure, and SecureAuth SAML Consumer SP, and is a widely-used procedure when utilizing SP initiation to avoid allowing users to go around a selected IdP.



Sign In

Username: rphillips@vootsy.com

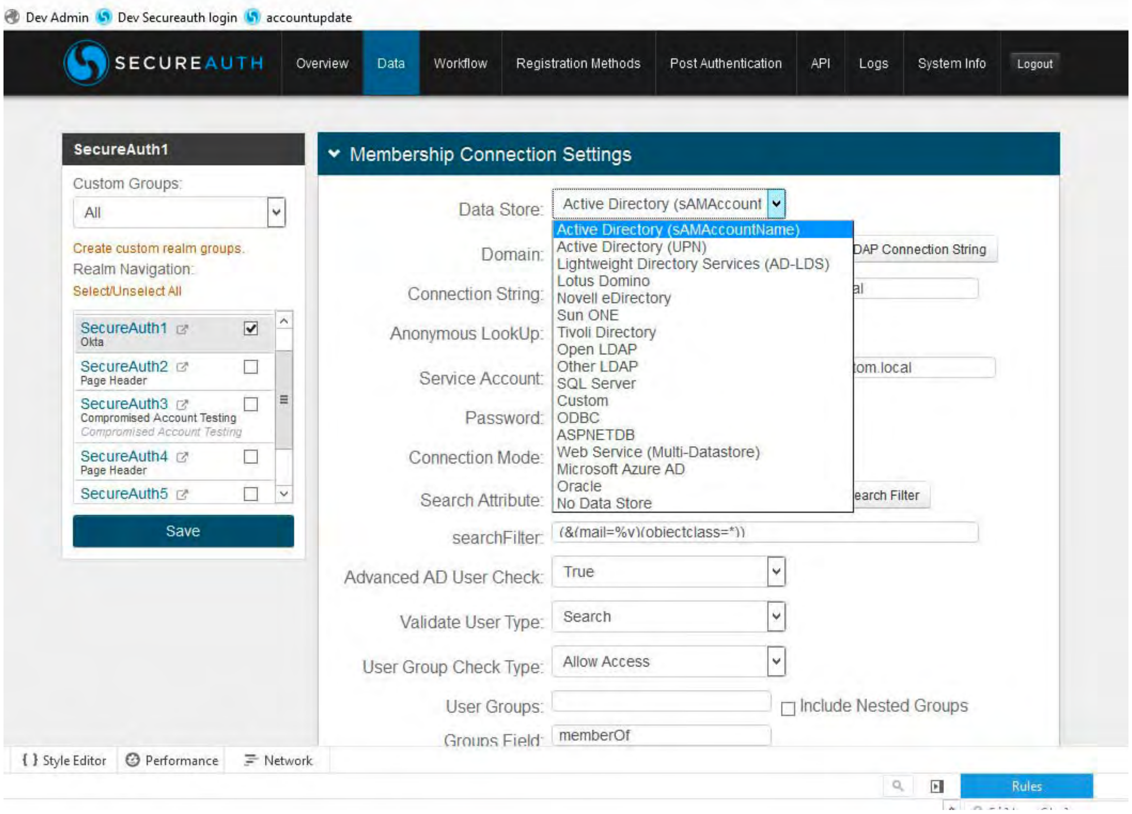
Password:

□ Remember me

Figure 6. Okta Login Modification

Required SecureAuth IdP Customizations

SecureAuth IdP would need to create a custom Membership and Profile provider that could access Okta's user data via API . For an example of this, refer to Figure 7, "SecureAuth Directory Membership and Profile Modification," on page 10 . This is a standard modification and would take roughly 40 hours to complete .



Custom Groups: All

Realm Navigation:

- SecureAuth1 Okta ☑
- SecureAuth2 Page Header □
- SecureAuth3 Compromised Account Testing *Compromised Account Testing* □
- SecureAuth4 Page Header □
- SecureAuth5 □

Membership Connection Settings

Data Store: Active Directory (sAMAccount

Domain:

Connection String: al

Anonymous LookUp:

Service Account: om.local

Password:

Connection Mode:

Search Attribute:

searchFilter: (&(mail=%v)(objectclass=*))

Advanced AD User Check: True

Validate User Type: Search

User Group Check Type: Allow Access

User Groups: □ Include Nested Groups

Groups Field: memberOf

Figure 7. SecureAuth IdP Directory Membership and Profile Modification

Use Case 3: Okta API Integration with SecureAuth

This use case discusses the preferred and best practice for Okta logins that use SecureAuth Adaptive, Authentication, and IDM APIs . This integration would be used if the customer wanted all configuration and authentication options in Okta but still wanted to use SecureAuth MFA, adaptive authentication, and fingerprinting features .

Standard Okta Login URL

This process considers a standard Okta login URL .

1. User attempts to access an Okta login via a typical well-know Okta sub-domain URL .
2. User is prompted for the Okta userid and password .
3. User is prompted for SecureAuth API-driven 2-factor authentication via the Okta interface .
4. User data and multi-factor data is validated and retrieved via API from the specified Okta sub-domain or from an on-premise Active Directory .
5. User is allowed access to Okta and its portal applications .

An illustration of this process is shown in Figure 8.

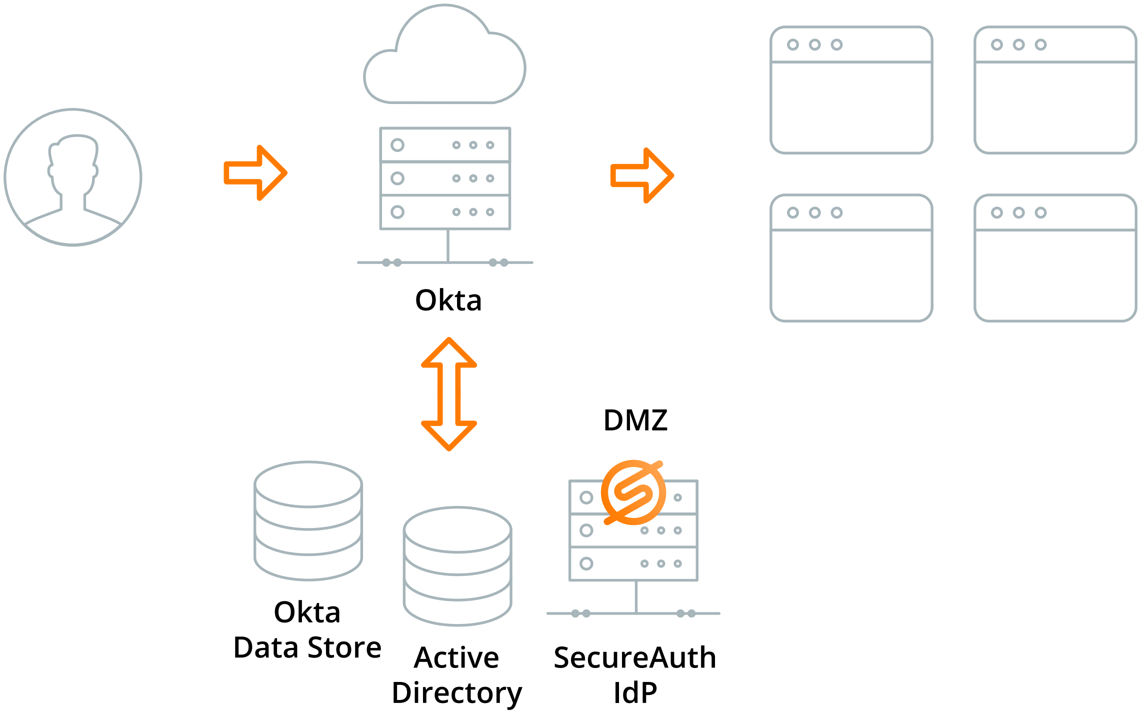


Figure 8. Okta authentication Utilizing SecureAuth APIs

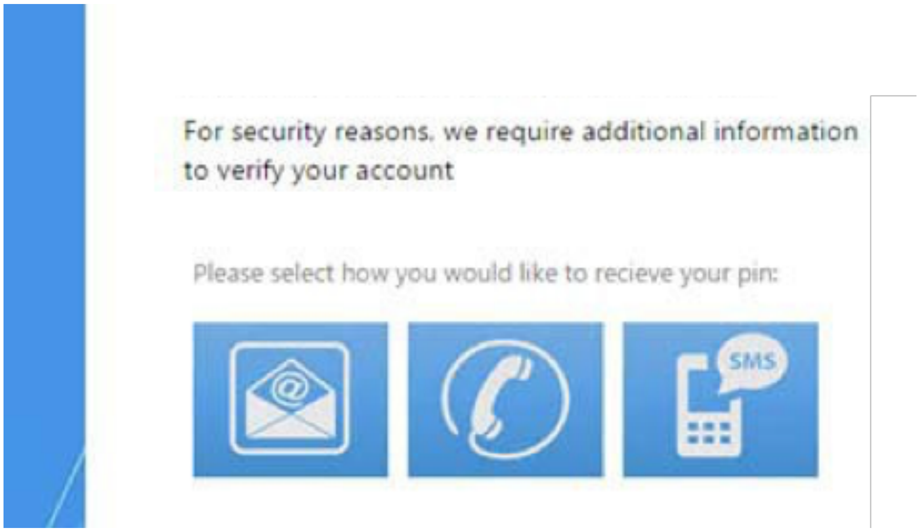
A further discussion of this process is provided in https://vootsy.oktapreview.com/login/default

Required Okta Customizations

Due to the number of modifications required for this use case, this customization should be left up to Okta. However, several possible settings are needed, as indicated in the following code.

```
{
        “Enablelogs”: “true”,
        “ValidationURL1”: “https://localhost/secureauth21”,
        “ValidationURL2”: “https://localhost/secureauth21”,
        “ManagementUIFriendlyName”: “SecureAuthAdapter”,
        “LogsPath”: “C:\SecureAuthAdapter\Logs”,
        “AppID”: “566adfff8ac84fef9c5c5704e2cc41dc”,
        “AppKey”: “a2f5b0f61e8c77872d9f71733e6e156a703aa9be57711ca5d5b5163f5be72246”,
        “SecureAuthRealmUrl”: “https://172.16.19.25/SecureAuth10/”,
        “phoneimageurl”: “http://vm-oc1-cd0505.sacustom.local/adfsimages/phoneadapter.jpg”,
        “smsimageurl”: “http://vm-oc1-cd0505.sacustom.local/adfsimages/smsadapter.jpg”,
        “emailimageurl”: “http://vm-oc1-cd0505.sacustom.local/adfsimages/emailadapter.jpg”,
        “progressgifurl”: “http://vm-oc1-cd0505.sacustom.local/adfsimages/301.GIF”,
        “emailprop”: “Email1”,
        “smsprop”: “Phone1”,
        “phoneprop”: “Phone1”,
        “enablephone”: “true”,
        “enablesms”: “true”,
        “enableemail”: “true”,
        “disablessl” : “true”
}
```

An example of one possible interface derived from this code is shown in Figure 9 .



For security reasons, we require additional information to verify your account

Please select how you would like to recieve your pin:

Figure 9. Possible User Interface for SecureAuth API Multi-Factor

Use Case 4: Okta On-Premise Agent Multi-Factor Integration with SecureAuth

This use case discusses an Okta login using on-premise MFA RADIUS connection to SecureAuth IdP for soft token validation . Customers would use this feature if it were required to validate SecureAuth IdP hard tokens with RADIUS .

Standard Okta Login URL Flow

The following steps describe the flow that would be required for a standard Okta Login using a RADIUS server . See an illustration of this flow in Figure 10 .

1. User attempts to access Okta login via a typical well-known Okta sub-domain URL.
2. User is prompted for Okta userid and password.
3. User is prompted to enter SecureAuth IdP soft token as a second factor.
4. The soft token is validated via SecureAuth IdP RADIUS server.
5. User is allowed access to Okta and portal applications.

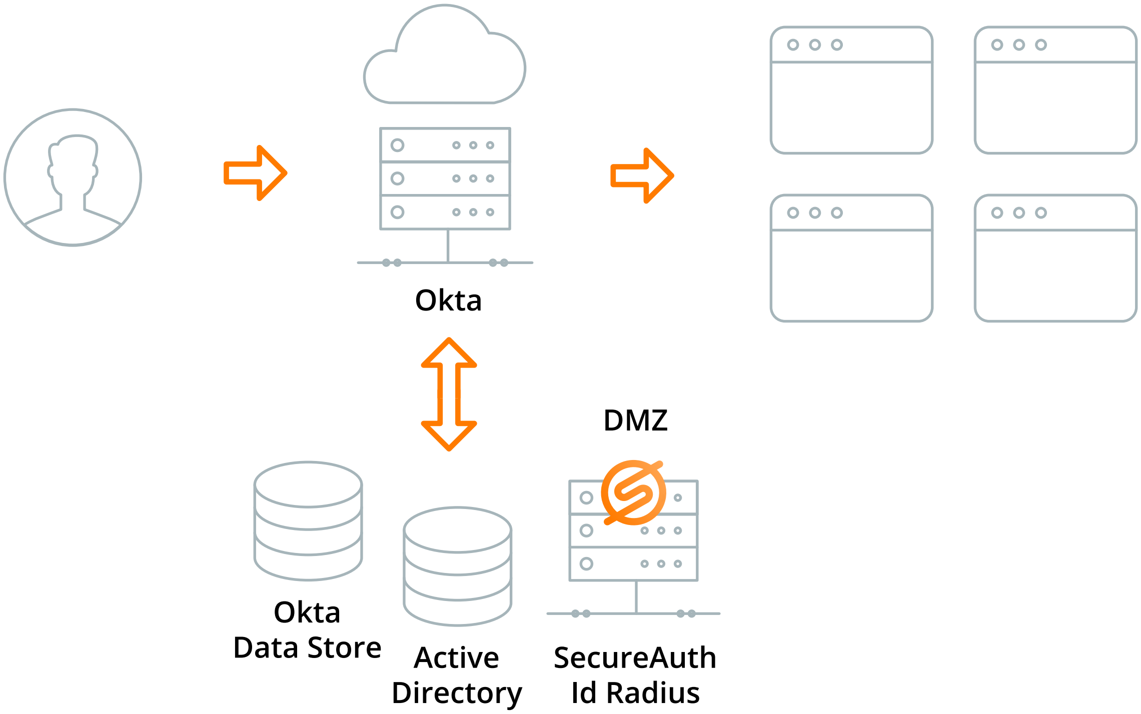
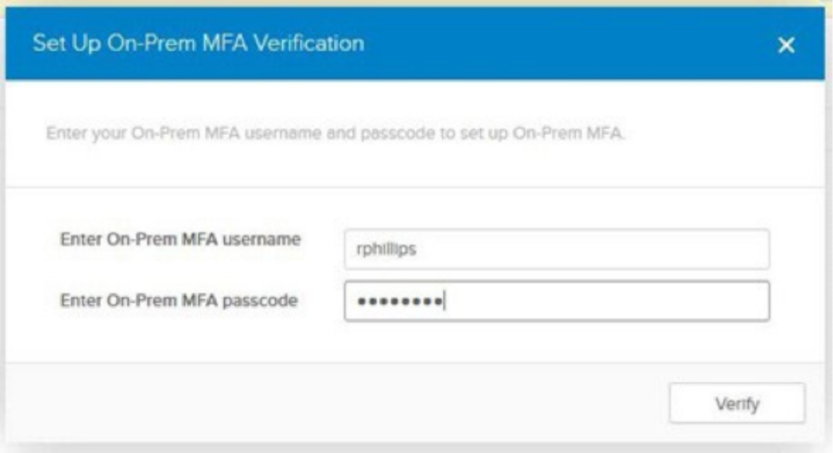


Figure 10. Okta On-Premise MFA Agent RADIUS

A further discussion of this process is provided in https://vootsy.oktapreview.com/login/default

Required Okta Customizations

Okta's On-Premise MFA Agent should support standard RADIUS User + Passcode as shown in Figure 11 .



Set Up On-Prem MFA Verification

Enter your On-Prem MFA username and passcode to set up On-Prem MFA.

Enter On-Prem MFA username: rphillips

Enter On-Prem MFA passcode: ••••••••

Figure 11. Sample Okta Setup On-Prem MFA Verification Screen

Summary

Using the strategies detailed in this guide, customers who already employ Okta can integrate SecureAuth IdP into their existing security environment, providing an even richer set of tools for creating secure, state-of-the-art MFA flows.

SECUREAUTH