SafeNet Authentication Service (SAS)

SAML Authentication Quick Start Guide

gemalto
security to be free

Document Part Number: 007-012471-001, Rev. G

Release Date: August 2016

Contents

Preface

Applicability

The information in this document applies to:

- **SafeNet Authentication Service (SAS)**—A cloud service of SafeNet, Inc.
- **SafeNet Authentication Service – Service Provider Edition (SAS-SPE)**—The software used to build an authentication service.
- **SafeNet Authentication Service – Private Cloud (SAS-PCE)**—A term used to describe the implementation of SPE/PCE.

Introduction

This guide describes the application, configuration and use of SafeNet Authentication Service as a SAML Identity Provider ("IdP") to relying SAML Service Providers ("SP"). It describes:

- How to configure a Virtual Server to be an IdP
- How to use the SAML Provisioning Rules Module introduced in SafeNet Authentication Service and LDAP to automate the configuration of individual user accounts to permit authentication for designated SPs, such as Google Apps
- How to customize logon and other pages presented to the user during SAML authentication
- Provides examples of SAML configurations for Google Apps and Salesforce

Readers are encouraged to read this guide in the order in which information is presented, as successive chapters often rely on information and concepts presented in prior chapters.

Audience

This guide is intended for SafeNet Authentication Service administrators responsible for how managed authentication services are delivered and responsible for configuring SAS to reflect the internal business processes, service level agreements, and management hierarchy.

Support Contacts

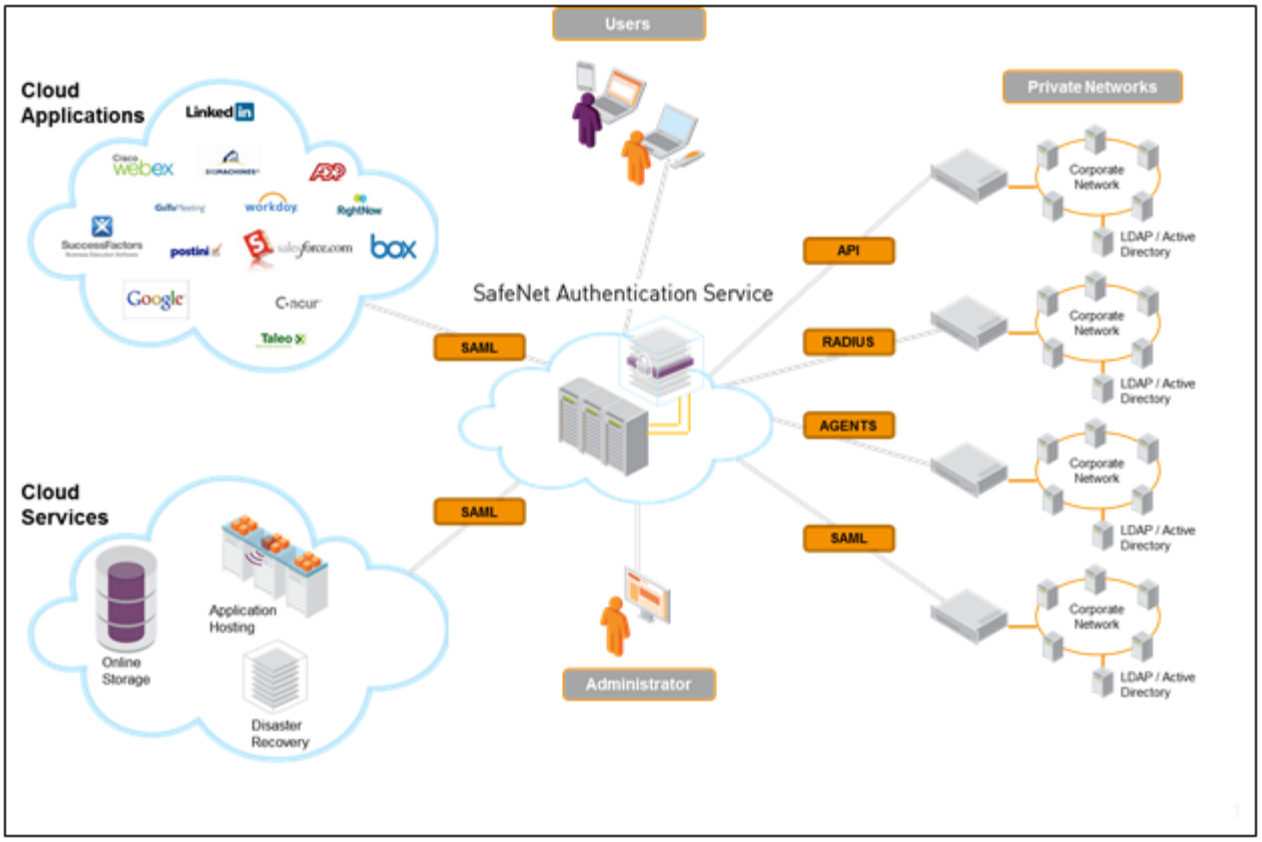
If you encounter a problem while installing, registering or operating this product, please make sure that you have read the documentation. If you cannot resolve the issue, contact your supplier or Gemalto Customer Support. Gemalto Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between Gemalto and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Contact Method	Contact Information	
Address	Gemalto 4690 Millennium Drive Belcamp, Maryland 21017, USA	
Phone	US	1-800-545-6608
	International	1-410-931-7520
Technical Support Customer Portal	https://serviceportal.safenet-inc.com Existing customers with a Technical Support Customer Portal account can log in to manage incidents, get the latest software upgrades, and access the Gemalto Knowledge Base.	

1
SAS and SAML

SafeNet Authentication Service adds SAML to the range of authentication options available to cloud subscribers.

Figure 1: SafeNet Authentication Service



This means that enterprises can:

- Extend strong authentication beyond the enterprise perimeter to include cloud apps such as Salesforce and Google Apps.
- Use SafeNet Authentication Service to protect internal applications such as SAP and others that support SAML authentication.
- Use SafeNet Authentication Service with perimeter devices, such as SSL VPNs, that support SAML authentication.
- Enable authorized users to authenticate into cloud apps in a simple, familiar, and consistent manner using the same token/authentication method they use for VPN and other traditional access.

- Automate cloud app authorization.
- Use SafeNet Authentication Service reporting to audit all user authentication activity, including authentication into cloud apps.

> **NOTE:** This document describes Service Provider (SP) initiated workflows only.

A Brief Introduction to SAML

SAML (“Security Assertion Markup Language”) is an Extensible Markup Language (XML) standard for exchanging authentication and authorization data between security domains; that is, between an identity provider (“IdP”) such as SafeNet Authentication Service and a service provider (“SP”), typically a web application such as Google Apps. It allows a user to log on once for affiliated, but separate, web sites or web applications.

SAML specifies three components—assertions, protocol, and binding. There are three assertions—authentication, attribute, and authorization.

- Authentication assertion validates the user's identity.
- Attribute assertion contains specific information about the user.
- Authorization assertion identifies what the user is authorized to do.

Protocol defines how SAML asks for and receives assertions. Binding defines how SAML message exchanges are mapped to Simple Object Access Protocol (SOAP) exchanges. SAML works with multiple protocols, including Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), and File Transfer Protocol (FTP), and also supports SOAP, BizTalk, and Electronic Business XML (ebXML).

While generally considered an authentication protocol for web apps, and in particular cloud computing, SAML is in fact supported by a range of applications and devices, including SAP, and perimeter devices such as SSL VPNs.

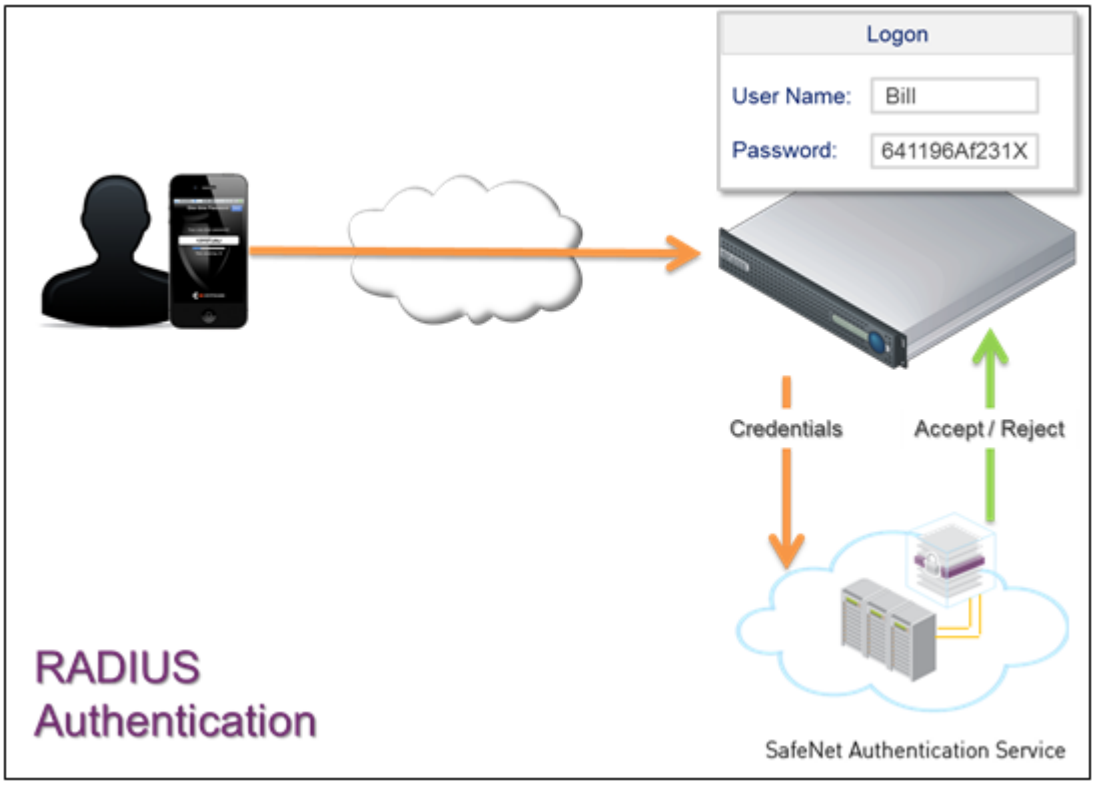
RADIUS on Steroids?

Those familiar with RADIUS might want to think of SAML as “RADIUS on steroids,” though the protocol is vastly different and substantially more flexible.

In a traditional RADIUS scenario, a user is prompted to provide authentication credentials (user ID and password) by an access point such as a VPN. The VPN uses the RADIUS protocol to pass the credentials to the authentication service for validation which in turn sends an “accept” or “reject” message via RADIUS back to the VPN (Figure 2: RADIUS Authentication User Experience on page 8).

By standardizing on RADIUS, an organization gains the freedom to choose any vendor’s RADIUS client (for example, a VPN) and be assured that it could use any other vendor’s RADIUS Server (for example, SafeNet Authentication Service).

Figure 2: RADIUS Authentication User Experience



However, RADIUS has rarely been adopted outside of network perimeter devices. Much like the days before the adoption of RADIUS, applications have each tended to have their own authentication mechanism. As a result, users tend to have many passwords and had to log into individual applications.

With the growth in web apps, and in particular cloud computing, this quickly becomes unmanageable for users and administrators alike. Obviously, a new authentication standard is required that can be adopted by application developers with ease, without requiring specific knowledge of how or what the authentication method will be. And equally important, the standard must provide a way to federate identity so that users do not require many passwords or a separate logon to individual applications.

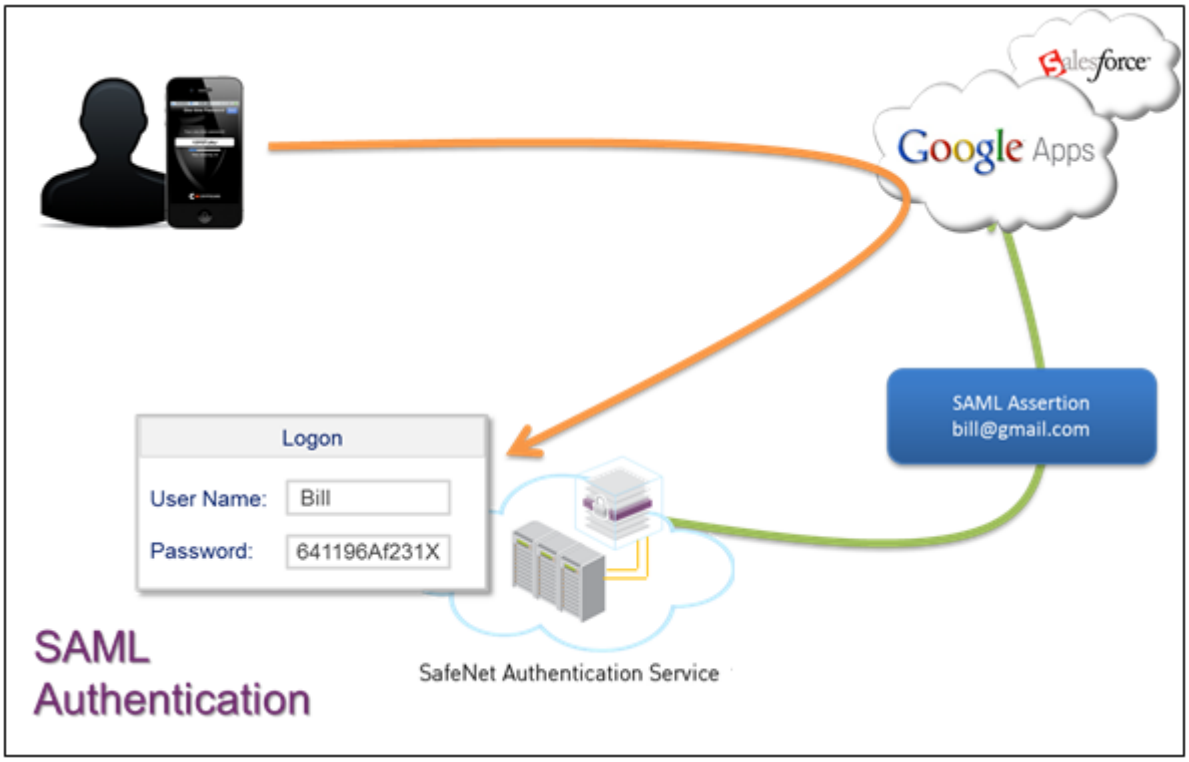
SAML, and in particular SAML 2.0, is the standard that makes this possible.

How Does SAML Work?

A SAML Service Provider (such as Google Apps, Salesforce, and SSL VPNs) “relies” on a SAML IdP (such as SafeNet Authentication Service) to present the logon page and authenticate users.

When a user logs in to an application that supports SAML, they are redirected to SafeNet Authentication Service where they must authenticate. If the authentication is successful, the user is redirected to their cloud app where access is granted. The SAML “assertion” generated by the IdP in response to a successful authentication is used by the Service Provider to grant the user access to the application.

Figure 3: SAML Authentication User Experience



Web Application SSO

Where an affiliation exists between separate web sites or applications, the successful SAML authentication will result in user access to the affiliate without imposing additional user logon – essentially web SSO. Figure 3: SAML Authentication User Experience illustrates a possible affiliation between Google Apps and Salesforce that would permit a user authenticated into one of these services to be able to use the other service without additional authentication.

Managing Cloud Identities

It's not uncommon for individual cloud applications to impose specific requirements with respect to user IDs. For example, a user may require a Gmail account (for example, bill@gmail.com) to log in to Google Apps, whereas Salesforce may require a domain-specific email address (for example, bill@company.com). If there's no affiliation between the web apps, the user may be required to log on separately to each application using different credentials. These, of course, may be in addition to the user ID required for logon through the corporate VPN (for example, bill).

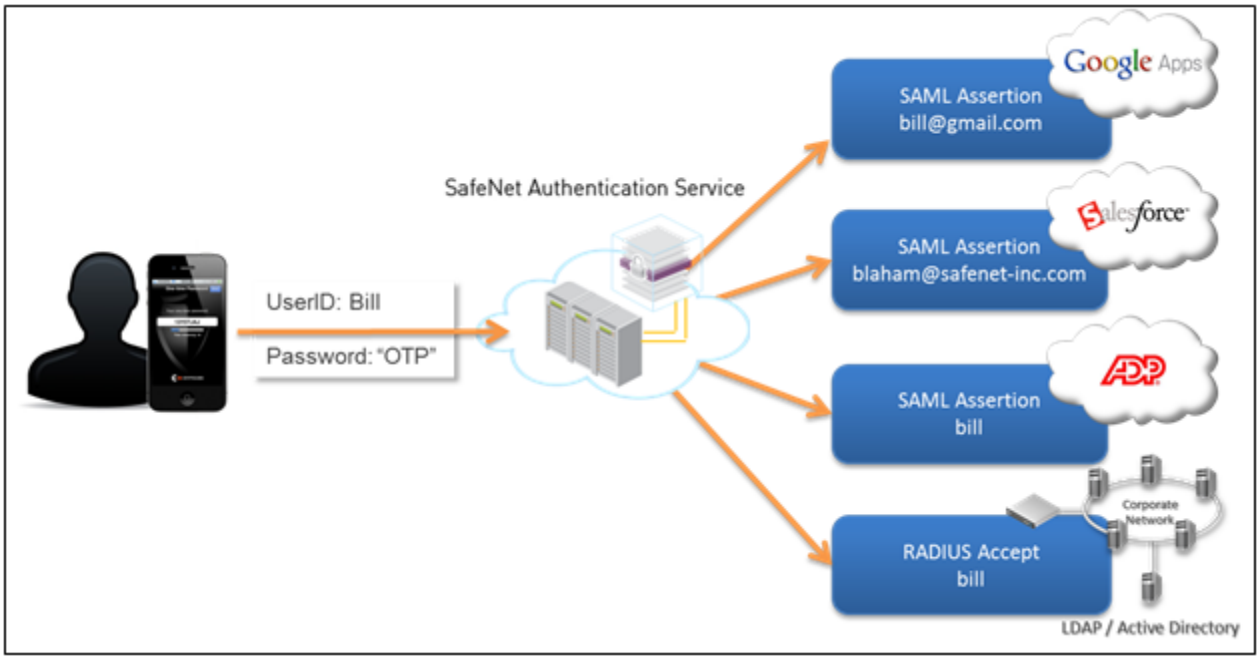
This can quickly become confusing and unmanageable for users and administrators. Fortunately, there are several remedies:

- Use SafeNet Authentication Service to normalize the user's logon credentials across corporate and cloud applications and services.
- Use SafeNet Authentication Service in conjunction with a cloud single sign-on (SSO) service.

Normalizing User Credentials Using SafeNet Authentication Service

One of the capabilities of SafeNet Authentication Service is to authenticate a user with a single credential—their user ID and one-time password (OTP)—but provide a different, specific credential required by the cloud app service. Effectively, SAS, on successful authentication, replaces the user ID provided during authentication with the user ID required by the cloud application in the SAML assertion. For the user, this delivers a consistent logon methodology (for example, UserID: Bill, Password: OTP) and insulates the user from any other credential management requirements.

Figure 4: User Credentials Using SafeNet Authentication Service

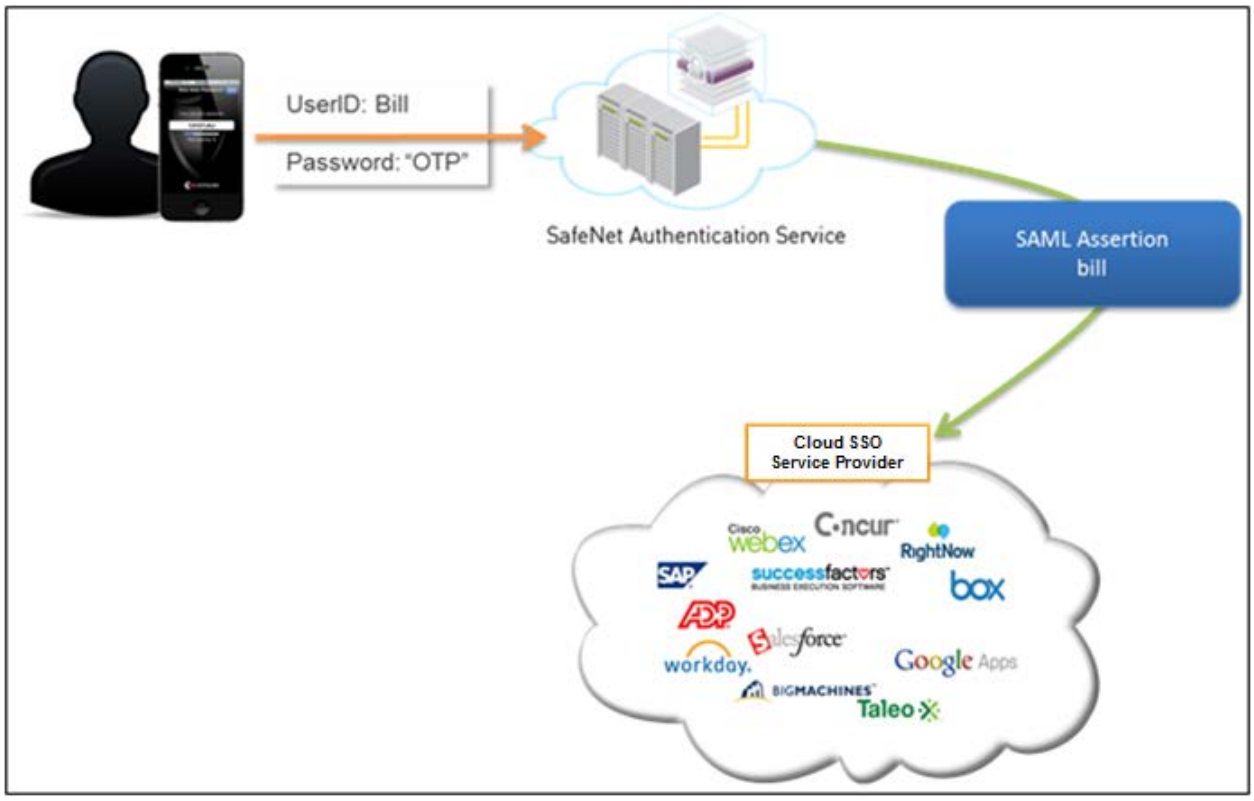


SafeNet Authentication Service with Cloud SSO Service Providers

Cloud SSO service providers provide a front-end for managing multiple cloud service providers and applications. Typically, these front-ends support SAML authentication and can therefore use SAS as the IdP.

The cloud SSO can be configured as a SAML service provider, relying on SAS to authenticate the user. Once authenticated, the user has access to cloud applications and services configured for their personal cloud SSO account.

Figure 5: SafeNet Authentication Service with Cloud SSO Service Providers



Automating Cloud App Authorization

One of the challenges facing administrators of large user populations is efficient and timely activation of SAML authentication. As the number of users and cloud apps grow, so too does the challenge of timely activation/deactivation.

SafeNet Authentication Service offers an elegant solution to the problem in the form of SAML Provisioning Rules. Generally, these rules are triggered on the addition or removal of a user from an LDAP security group and/or SafeNet Authentication Service internal group, and allow/deny authentication for users authenticating at the specified SAML service providers respectively.

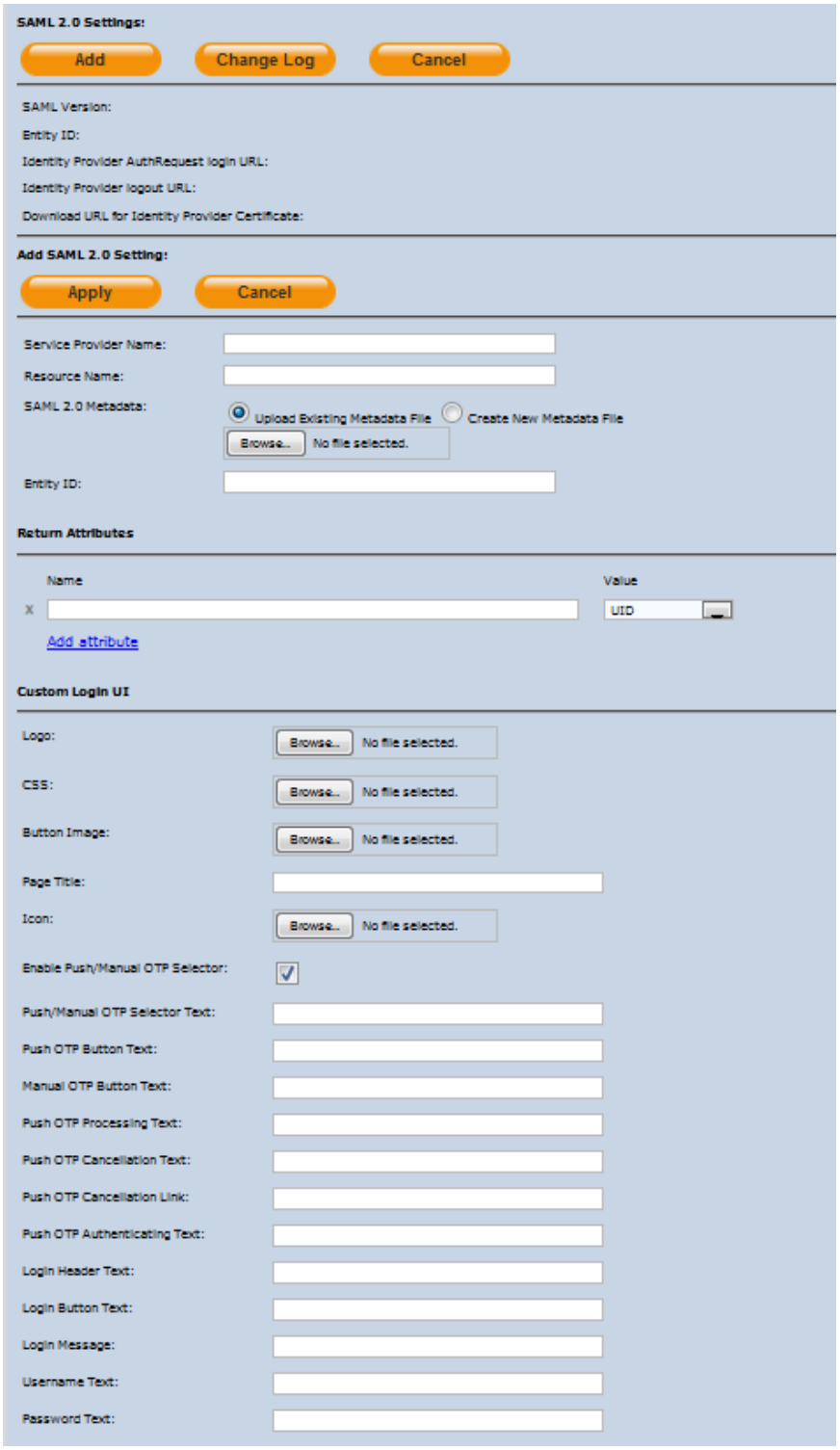
2
Configuring SAML Authentication in SafeNet Authentication Service

There are three steps to configure SAML authentication:

1. **Configure SAML Service Providers**—Use this step to configure the virtual server to process authentication requests received from a specific SAML service provider.
2. **Configure SAML Services**—Use this step to manually enable SAML authentication for individual users to one or more of the SAML service providers created in step 1.
3. **Configure SAML Provisioning Rules**—Use this step to automatically enable SAML authentication for individual users to one or more of the SAML service providers created in step 1. SAML Provisioning Rules can be used instead of manual configuration of services (Step 2) or in addition to manual configuration.

Step 1: Configure SAML Service Providers

In SAS, start the configuration of SAML service providers from the **SAML Service Providers** module on the **VIRTUAL SERVERS > COMMS** tab.



The information displayed below the **Add** button will be required by your service provider.

Click **Add** to insert a new provider into the list where:

- **Relying Party ID**—This is the “Entity ID” of the SAML Service Provider, typically (but not always) in the form of a URL. This value will be provided by the SAML Service Provider or can be extracted from the metadata (XML file) provided by the SAML Service Provider.

  For example:

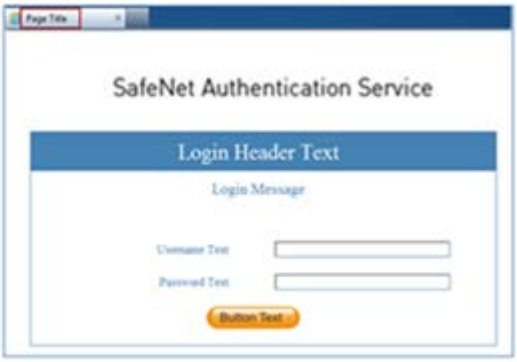
  <?xml version="1.0" encoding="UTF-8"?>

  <md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" entityID=https://mycompany.salesforce.com

- **Service Provider Name**—This is a name you assign to the Relying Party for easy identification. This name will appear in SAML Services lists under **Assignment > SAML Services**, and under **Policies > Automation Policies > SAML Provisioning Rules**.
- **Resource Name**—Push notifications include the SAML service where a push was initiated. To ensure users can easily recognize the source of the push notification, this field allows Operators to customize the SAML service name that will display. This field defaults to the SAML **Service Provider Name** when a new SAML service is added.
- **Return Attributes**—Return attributes are used to enable SAML applications to integrate with SAS, and to authorize the user based on the attribute values. For newly created SAML services, the SAS Operator will need to add all required attributes (there are no default attributes). If there are no attributes added, the agent will not send any. To add a return attribute:
  - Under Return Attributes, click **Add attribute**.
  - Define the **Name** for the attribute, and then select the attribute type from the **Value** menu. If a custom value is needed, select **Custom** from the **Value** menu, and then enter the text to define the value.
  - To delete a return attribute, click the **X** adjacent to the attribute.
- **SAML 2.0 Metadata**
  - **Upload Existing Metadata File**—This is an XML file that is generated by your SAML Service Provider.
  - **Create New Metadata File**—Some SAML Service Providers do not provide a metadata file, but instead provide only their Entity ID and Location (essentially the resource being accessed). Use this option to have the virtual server create and add a metadata file based on this information.

    The remaining options are used to customize the appearance of the **Login** page presented to the user:

SAS Cloud v3.5.1 (or later)



SAS PCE

- **Logo**—This is the logo you want to appear on the logon form presented to your users during authentication.
- **CSS**—Modify the default CSS then upload to modify the appearance of the page. The following is the default CSS:

```
.tableBanner
{
   width: 600px;
   border-width: 0px;
   border-spacing: 0px;
   background-color: white;
}

.tableMain
{
   width: 600px;
   border-width: 1px;
   border-spacing: 0px;
   border-style: solid;
   border-color: #4682B4;
   border-collapse: separate;
   background-color: white;
   padding: 0px;
}

.tdTopSpaceAboveBanner
{
   height: 50px; text-align: center;
}

.tdBanner
{
   height: 100px; text-align: center;
}

.tdSpaceBelowBanner
{
   height: 50px; text-align: center;
}
```

```
.tdLoginHeader
{
   height: 50px; text-align: center; font-size: 28px; color: white; background-color: #4682B4; padding-left: 0px; padding-right: 0px;
}

.tdLoginMessage
{
   height: 50px; text-align: center; font-size:20px; color: #4682B4;
}

.tdUserNameLabel
{
   text-align: right;
   font-size: 15px;
   color: #4682B4;
   padding-left: 70px;
}

.textUserName
{
   width: 225px; height: 20px; text-align: left; border-color: #4682B4; border-width: 1px;
}

.tdPasswordLabel
{
   text-align: right;
   font-size: 15px;
   color: #4682B4;
   padding-left: 70px;
}

.textPassword
{
   width: 225px; height: 20px; text-align: left; border-color: #4682B4; border-width: 1px;
}
.tdUserName
{
   padding-left: 60px;
```

```
}
.tdPassword
{
   padding-left: 60px;
}
.td20PxSpace
{
   height: 20px;
}
.td40PxSpace
{
   height: 40px;
}
.tdUserErrorMessage
{
   height: 40px; color: red; text-align: center; font-size: 14px;
}
.tdSubmit
{
   text-align: center; height: 30px;
}
.buttonSubmit
{
   background-color: white; background-repeat:no-repeat; border-width: 0px; width: 120px; height:
28px; text-align: center; font-size: 14px; color: white;
}

.tdSpaceBelowLoginWindow
{
   height: 80px;
}

.relayingParty
{
   text-align: center; font-size: 10px; color:darkblue; height: 20px;
}
.sessionTimeout
{
   text-align: center; font-size: 12px; color:blue;
```

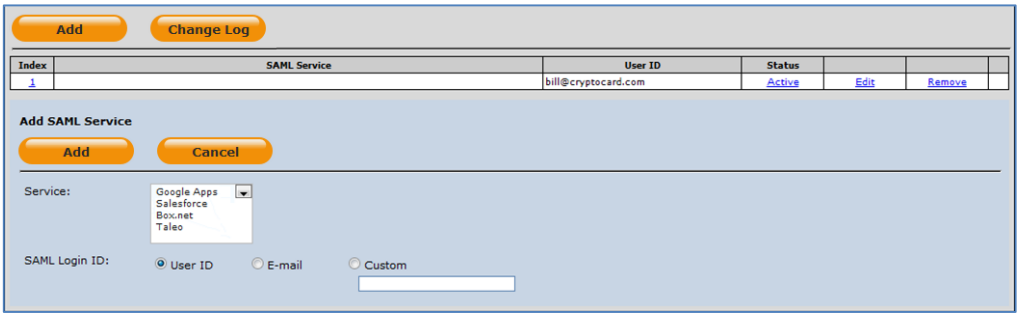
```
}
.sessionWarning
{
   text-align: center; font-size: 14px; color:crimson;
}
.copyRight
{
   text-align: center; font-size: 8px; color: darkblue; height: 20px;
}

.td404Error
{
   height: 40px; color: red; text-align: left; font-size: 28px;
}
.tdError
{
   height: 40px; color: red; text-align: left; font-size: 28px;
}
.tdWarning
{
   height: 40px; color: brown; text-align: left; font-size: 28px;
}
.tdInformation
{
   height: 40px; color: darkblue; text-align: left; font-size: 28px;
}
.tdSignoutMessage
{
   height: 40px; color: red; text-align: left; font-size: 18px;
}
.tdErrorMessage
{
   height: 40px; color: red; text-align: left; font-size: 14px;
}
```

- **Button Image**—This is the image used for the logon button.
- **Page Title**—This is the page title displayed on the browser tab.
- **Icon**—This is the icon displayed on the browser tab.
- **Enable Push/Manual OTP Selector**—(not available with SafeNet Authentication Service – PCE/SPE editions) This check box displays controls on the SAML **Login** page for selecting between Push OTP and manually entering the OTP.
- **Push/Manual OTP Selector Text**—(not available with SafeNet Authentication Service – PCE/SPE editions) This field is displayed if the **Enable Push/Manual OTP Selector** checkbox is selected. This is the "**I want to:**" text to display on the SAML **Login** page.
- **Push OTP Button Text**—(not available with SafeNet Authentication Service – PCE/SPE editions) This field is displayed if the **Enable Push/Manual OTP Selector** checkbox is selected. This is the text displayed for the option to use Push OTP.
- **Manual OTP Button Text**—(not available with SafeNet Authentication Service – PCE/SPE editions) This field is displayed if the **Enable Push/Manual OTP Selector** checkbox is selected. This is the text displayed for the option to use a manual OTP.
- **Push OTP Processing Text**—(not available with SafeNet Authentication Service – PCE/SPE editions) This is the text displayed when a Push OTP request is processing and waiting for a response from the user.
- **Push OTP Cancellation Text**—(not available with SafeNet Authentication Service – PCE/SPE editions) This is the text displayed on the button that is used to cancel a Push OTP request.
- **Push OTP Cancellation Link**—(not available with SafeNet Authentication Service – PCE/SPE editions) This is the text displayed when a Push OTP request is cancelling.
- **Push OTP Authenticating Text**—(not available with SafeNet Authentication Service – PCE/SPE editions) This is the text displayed when a Push OTP request is authenticating.
- **Login Header Text**—This field is no longer used.
- **Login Button Text**—This is the text displayed on the logon button.
- **Login Message**—This field is no longer used.
- **Username Field**—This is the label for the user name field
- **Password Text**—This is the label for the password field.

Step 2: Configure SAML Services

Use this module to manually enable a user to authenticate against one or more configured SAML Service Providers where:

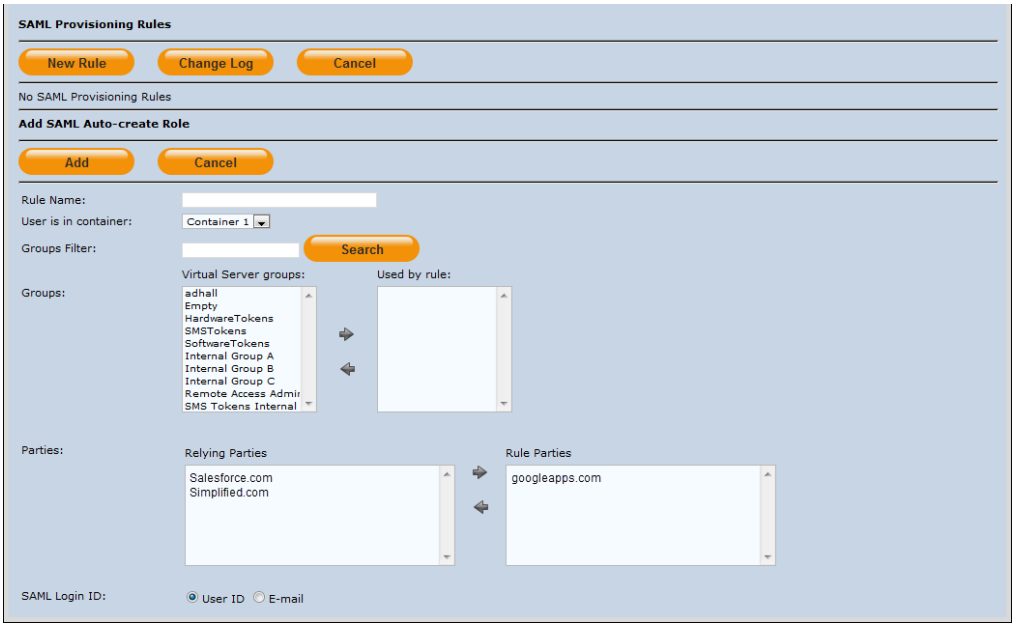


- **Service**—Lists all of the configured SAML Service Providers configured in Step 1.
- **SAML Login ID**—This is the UserID that will be returned to the service provider in the SAML assertion on successful authentication. For example, if your service provider (for example, Salesforce) requires a user ID of name@domain.com, and this is identical to the user's email address, choose the **Email** option. Doing so allows the user to consistently use their user ID to authenticate regardless of the service providers requirements. In most cases, a service provider will require either the user ID or email. For all other cases, choose the **Custom** option and enter the required user ID to be returned.

You can automate the creation/removal of SAML Services for users by creating a SAML provisioning rule. Refer to Step 3: SAML Provisioning Rules.

Step 3: SAML Provisioning Rules

Use this module to automate adding or removing the right for users to authenticate to SAML service providers.



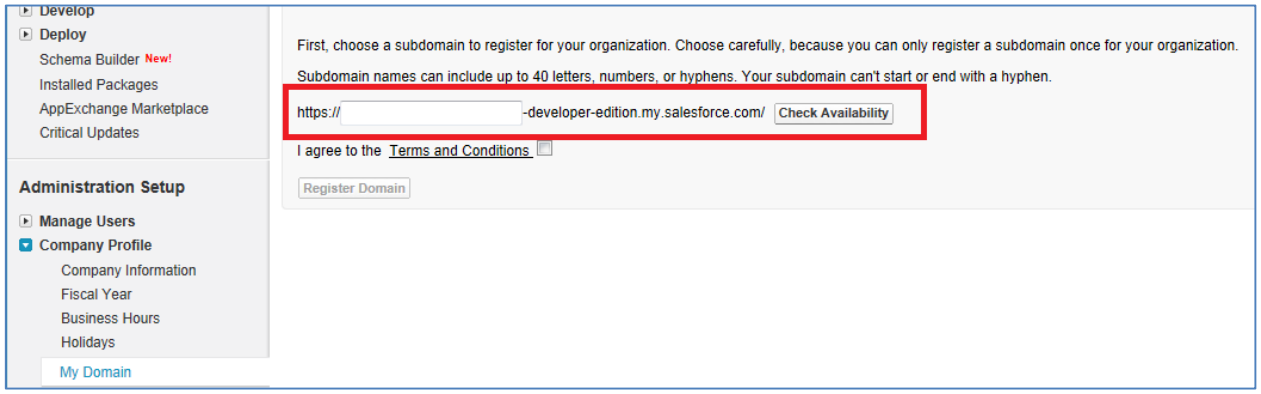
- **Rule Name**—This is a name that describes the rule.
- **User is in container**—Users affected by this rule must be in the selected container.
- **Server Groups**—Users in these groups are not affected by this rule.
- **Rule Groups**—Users must be in one or more of these groups to be affected by this rule.
- **Relying Parties**—Service providers in this section are not affected by this rule.
- **Rule Parties**—Users that belong to one or more of the Rule Groups will be able to authenticate against Service Providers in this section.
- **SAML Login ID**—This is the user ID that will be returned to the service provider in the SAML assertion.

3
Sample SAML Configurations

The following examples illustrate how to configure various SAML service providers to use SafeNet Authentication Service as a SAML IdP. Note that the data used in these examples is for illustration only. Be sure to use data as displayed in your SafeNet Authentication Service and SAML service provider.

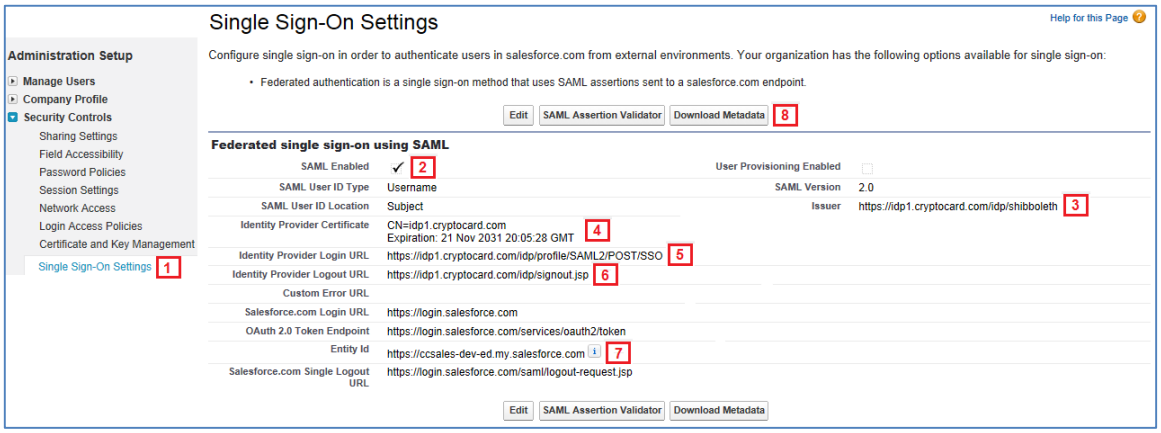
Salesforce

To use SAML with Salesforce, you must configure “My Domain” in Salesforce. Refer to Salesforce **Administration Setup > Company Profile > My Domain**.



Step 1: Configure Single Sign-On

1. Log in to **Salesforce > Administration Setup > Security Controls > Single Sign-On Settings**.
2. Select the option **SAML Enabled**.



3. Upload the SAS Identity Provider Certificate

   Obtain this certificate from the **Download URL for Identity Provider Certificate** link displayed under **COMMS > SAML Service Providers > SAML 2.0 Settings**.

4. Identity Provider Login URL

   Use the value from **Identity Provider AuthenRequest URL** displayed under **COMMS > SAML Service Providers > SAML 2.0 Settings**.

5. Identity Provider Logout URL

   Use the value from **Identity Provider Logout URL** displayed under **COMMS > SAML Service Providers > SAML 2.0 Settings**.

6. Entity ID

   This is a unique ID created by Salesforce for your organization. This information, usually in the form of a URL, must be entered into the **Entity ID** field in SAS.

7. Download Metadata

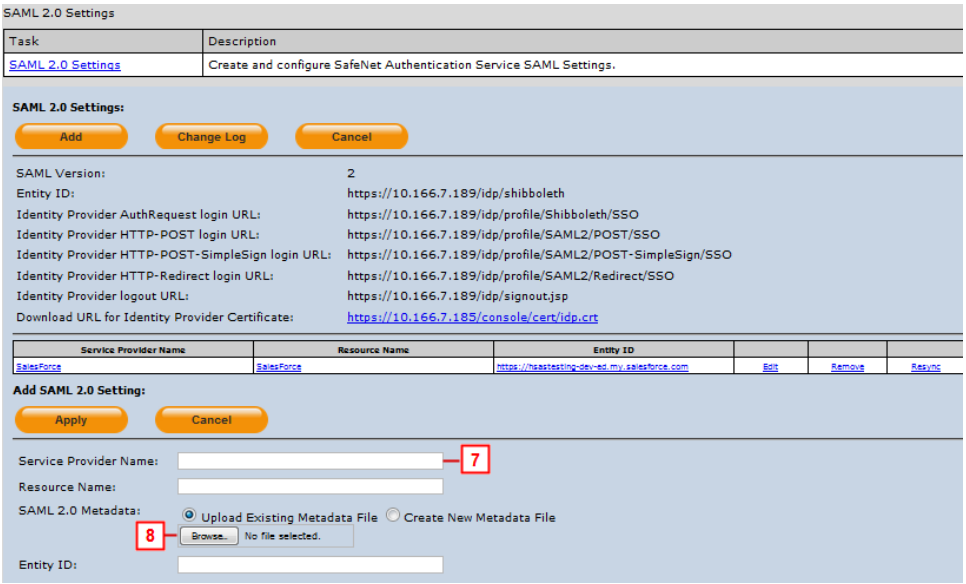
   Download the metadata file from Salesforce and save it to a convenient location. You will upload this file to SAS in step 10.

Step 2: Add Salesforce as a SAML Service Provider

Under **SAML Service Providers | SAML 2.0 Settings**, click **Add** to configure a new SAML service provider.

8. Entity ID

   Copy the **Entity ID** information displayed in Salesforce (step 6 above) into the **Entity ID** field in SAS.



9. Service Provider Name (SAS Cloud)/Friendly Name (SAS PCE)

   This is a name you assign to the Relying Party for easy identification. This name will appear in **SAML Services** lists under **Assignment > SAML Services** and under **Policies > Automation Policies > SAML Provisioning Rules**.

10. SAML 2.0 Metadata

    Upload the Salesforce metadata file from step 7 to SAS.

11. Customize

    Customize the logon page presented to users during logon to Salesforce.

Google Apps

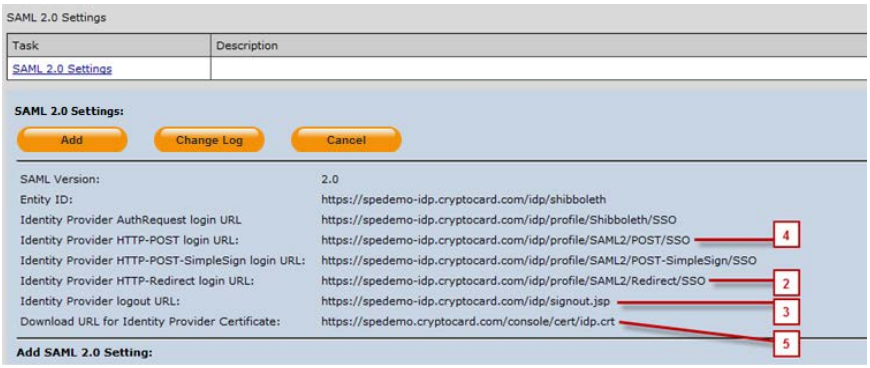
Step 1: Set Up Single-Sign-On

Log in to **Google Apps > Advanced tools > Authentication > Set up Single Sign-on (SSO)**.

1. Select the option **Enable Single Sign-on**.

2. Sign-in page URL

   Use the value from **Identity Provider HTTP-Redirect logon URL** displayed under **COMMS > SAML Service Providers > SAML 2.0 Settings**.



3. Sign-out Page URL

   Use the value from **Identity Provider logout URL** displayed under **COMMS > SAML Service Providers > SAML 2.0 Settings**.

4. Change Password URL

   Use the value from **Identity Provider HTTP=POST logon URL** displayed under **COMMS > SAML Service Providers > SAML 2.0 Settings**.

5. Verification Certificate

   Use the **Download URL for Identity Provider Certificate** link displayed under **COMMS > SAML Service Providers > SAML 2.0 Settings** to obtain the SAS certificate. Upload this certificate to Google Apps.

6. Use a domain-specific issuer

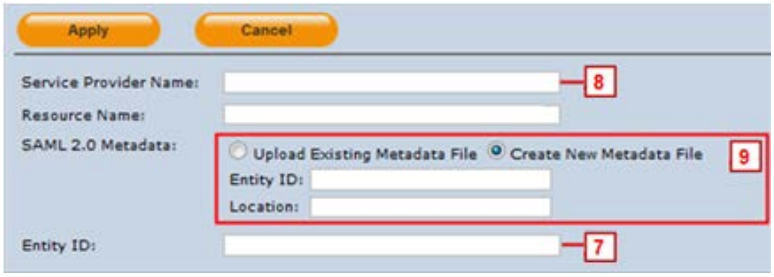
   Ensure that this option is selected. Use the value generated by Google Apps, typically **google.com/a/mycompany**, where **mycompany** is your domain registered in Google Apps. This information will be required in next steps.

Step 2: Add Google Apps as a SAML Service Provider

Under **SAML Service Providers > SAML 2.0 Settings**, click **Add** to configure a new SAML Service Provider.

7. Entity ID

   Copy the domain-specific identifier generated by Google Apps displayed in Salesforce (step 6 above) into the **Entity ID** field in SAS.



8. Service Provider Name (SAS Cloud)/Friendly Name (SAS PCE)

   This is a name you assign to the Relying Party for easy identification. This name will appear in SAML Services lists under **Assignment > SAML Services** and under **Policies > Automation Policies > SAML Provisioning Rules**.

9. SAML 2.0 Metadata

   Google Apps does not generate metadata. To compensate, select the **Create New Metadata File** option, and then enter the following:

   - **Entity ID**—This is the Google Apps Entity ID from step 7 above (for example, google.com/a/mycompany)
   - **Location**—This is the SAML assertion consumer URL, typically the Entity ID preceded by **https://www**. Note: **/acs** must be added at the end (for example, https://www.google.com/a/mycompany/acs).

10. Customize

    Customize the logon page presented to users during logon to Google Apps.