PTT Pro and Profile Manager

Workforce Connect

ZEBRA

SAML Integration Guide

MN-004608-01EN Rev A

2022/10/11

ZEBRA and the stylized Zebra head are trademarks of Zebra Technologies Corporation, registered in many jurisdictions worldwide. All other trademarks are the property of their respective owners. ©2022 Zebra Technologies Corporation and/or its affiliates. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements.

For further information regarding legal and proprietary statements, please go to:

SOFTWARE: zebra.com/linkoslegal.
COPYRIGHTS: zebra.com/copyright.
PATENTS: ip.zebra.com.
WARRANTY: zebra.com/warranty.
END USER LICENSE AGREEMENT: zebra.com/eula.

Terms of Use

Proprietary Statement

This manual contains proprietary information of Zebra Technologies Corporation and its subsidiaries (“Zebra Technologies”). It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the express, written permission of Zebra Technologies.

Product Improvements

Continuous improvement of products is a policy of Zebra Technologies. All specifications and designs are subject to change without notice.

Liability Disclaimer

Zebra Technologies takes steps to ensure that its published Engineering specifications and manuals are correct; however, errors do occur. Zebra Technologies reserves the right to correct any such errors and disclaims liability resulting therefrom.

Limitation of Liability

In no event shall Zebra Technologies or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, consequential damages including loss of business profits, business interruption, or loss of business information) arising out of the use of, the results of use of, or inability to use such product, even if Zebra Technologies has been advised of the possibility of such damages. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Introduction

The support of a shared device model for Workforce Connect (WFC) products focuses on AD/ADFS (Active Directory / Active Directory Federated Services) using the OAuth2 protocol. The widespread adoption of PTT Pro and Profile Manager has created the need to support SAML 2.0 (SAML2) as an authorizing protocol.

Keycloak adds the support of SAML2 without changing the current product support of OAuth2. The SAML2 capability is provided by the WFC Authentication Connection Service (WFC-ACS), which brokers access authorization between the SAML Identity Management infrastructure and the OAuth2 authorization capabilities of Workforce Connect.

This guide describes how the WFC-ACS architecture is positioned in the WFC environment and how to configure the connection services of the PTT Pro and Profile Manager OAuth authorization services into the SAML2 Identity Management (IdP) infrastructure.

Document Layout

This guide includes the following sections.

Solution Components and Architecture

Provides a high-level overview of the components, from the mobile device to the IdP server. This section also includes detailed communication flows identifying each component’s task and the sequence in which these tasks must occur.

Configure WFC-ACS

Describes the ACS configuration process and illustrates the WFC to OAuth configuration as well as the SAML2 to IdP configuration elements.

Configure Workforce Connect

This section describes the system from an operational perspective. Here the configurations of the mobile device, Profile Manager, and PTT Pro server are described. The mobile device configuration should be reviewed and adjusted as necessary to produce a smooth user experience.

Troubleshooting

This section describes issues that can be encountered during the configuration process.

Solution Components and Architecture

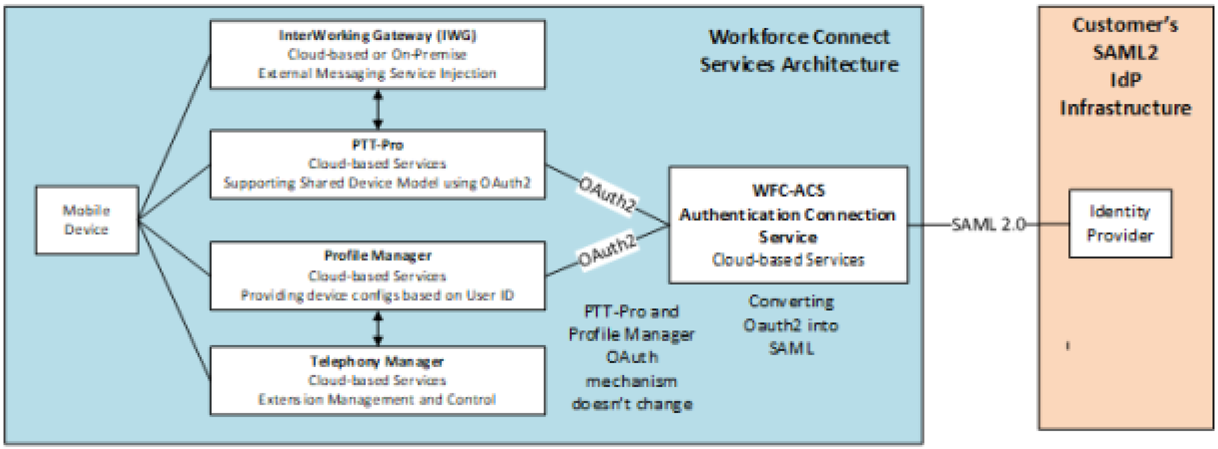
WFC-ACS provides the ability for existing WFC systems to authorize services from a SAML IdP. With the ACS service, no software changes are required to the WFC systemsor to the SAML infrastructure.

NOTE: WFC-ACS is based on a customized open source Keycloak environment. The off-the-shelf Keycloak system does not provide the capabilities described in this document.

WFC-ACS Component Diagram

The figure below includes WFC products, including Telephony Manager, which is not involved with OAuth or SAML. Both the PTT Pro and Profile Manager Server use OAuth for user authorization. The ACS server is the broker between the WFC OAuth services and the customer's SAML IdP infrastructure.

Figure 1 WFC Communication


Workforce Connect Services Architecture
InterWorking Gateway (IWG)
Cloud-based or On-Premise
External Messaging Service Injection
PTT-Pro
Cloud-based Services
Supporting Shared Device Model using OAuth2
Mobile Device
Profile Manager
Cloud-based Services
Providing device configs based on User ID
Telephony Manager
Cloud-based Services
Extension Management and Control
OAuth2
OAuth2
WFC-ACS
Authentication Connection Service
Cloud-based Services
PTT-Pro and Profile Manager OAuth mechanism doesn't change
Converting Oauth2 into SAML
SAML 2.0
Customer's SAML2 IdP Infrastructure
Identity Provider

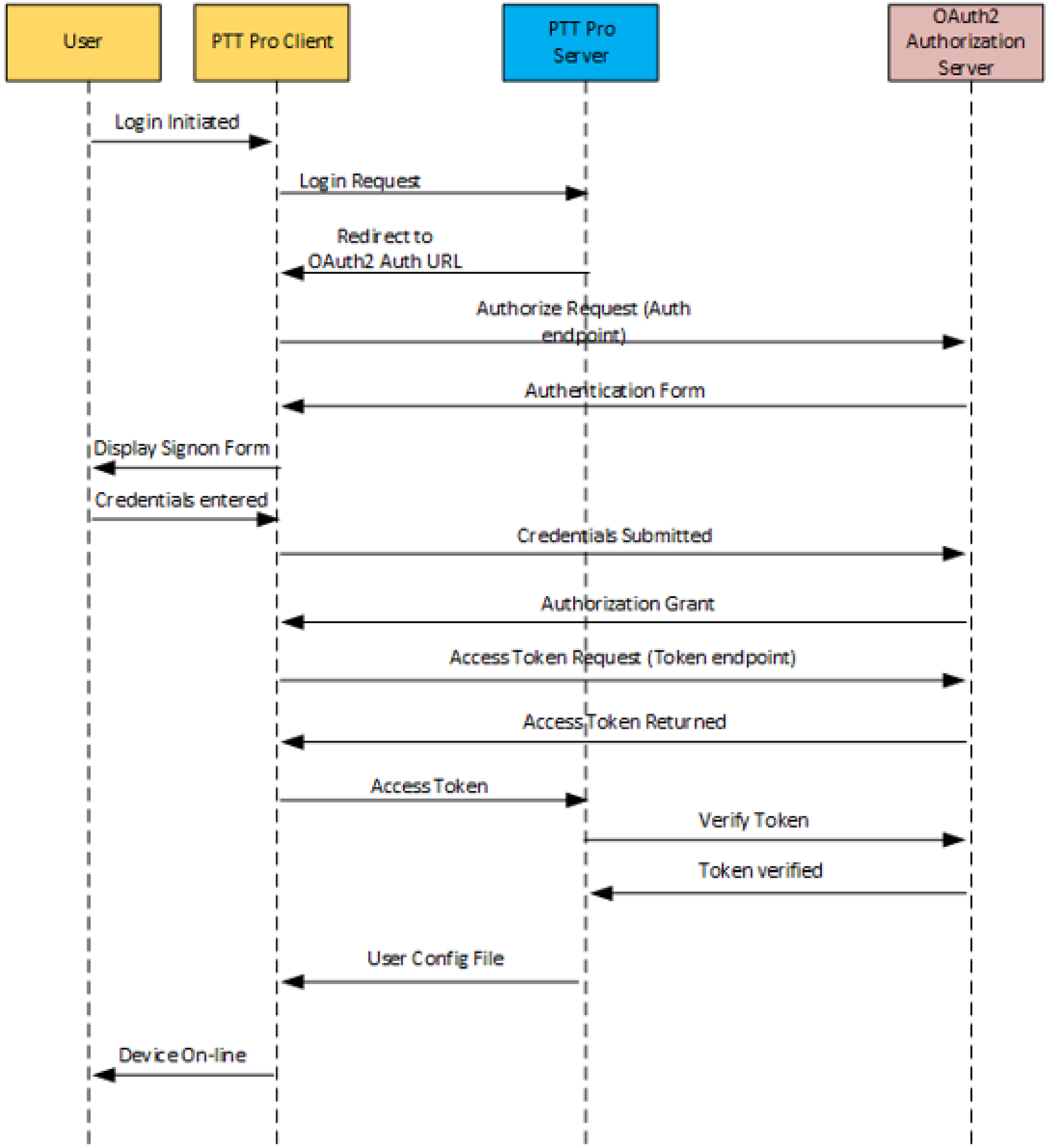
User Authorization Diagrams

The following ladder diagrams illustrate the sequence of authentication events and which component performs which function.

The first illustration shows the existing PTT Pro OAuth sequence to an AD/ADFS infrastructure. This is provided for the administrator to understand operations before introducing WFC-ACS.

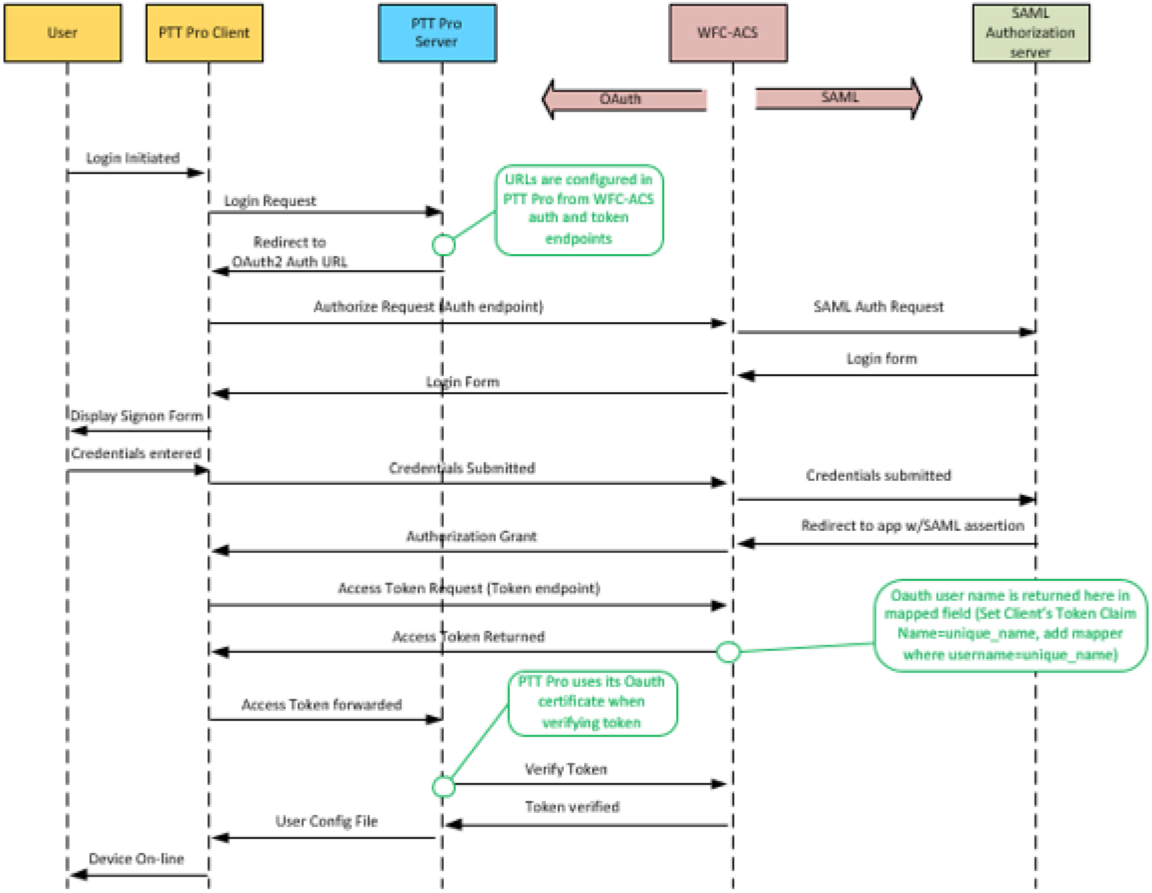
After the administrator understands the AD/ADFS (Active Directory / Active Directory Federated Services) operations, the next diagram introduces WFC-ACS and how the flow of authorization is transferred, or converted from OAuth to SAML. Both PTT Pro and Profile Manager servers are shown.

Figure 2 PTT Pro Shared Device Using OAuth2



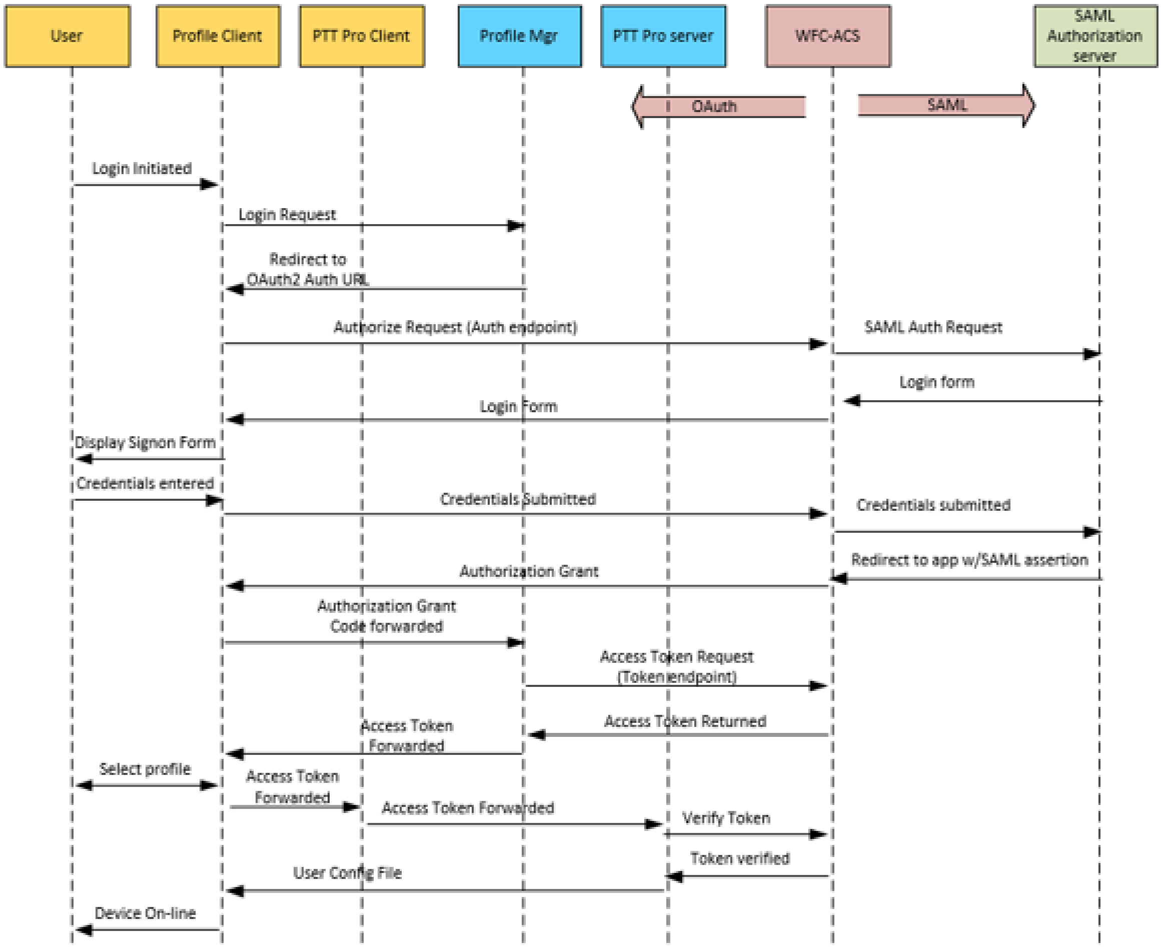
Once an understanding of the AD/ADFS (Active Directory / Active Directory Federated Services) is developed, the following diagram shows the introduction of WFC-ACS and in a standalone PTT Pro configuration.

Figure 3 PTT Pro Shared Device Using WFC-ACS


User
PTT Pro Client
PTT Pro Server
WFC-ACS
SAML Authorization server
OAuth
SAML
Login Initiated
Login Request
URLs are configured in PTT Pro from WFC-ACS auth and token endpoints
Redirect to OAuth2 Auth URL
Authorize Request (Auth endpoint)
SAML Auth Request
Login form
Login Form
Display Signon Form
Credentials entered
Credentials Submitted
Credentials submitted
Redirect to app w/SAML assertion
Authorization Grant
Access Token Request (Token endpoint)
Oauth user name is returned here in mapped field (Set Client's Token Claim Name=unique_name, add mapper where username=unique_name)
Access Token Returned
PTT Pro uses its Oauth certificate when verifying token
Access Token forwarded
Verify Token
Token verified
User Config File
Device On-line

The following ladder diagram describes the flow of authorization in a deployment with Profile Manager servers.

Figure 4 Profile Manager and PTT Pro Shared Device Using WFC-ACS



Prerequisites

The WFC Authentication Connection Service (WFC-ACS) is supported for the software versions listed below. In addition, it requires TCP port 443.

Minimum WFC software versions

- PTT Pro server v4.7.3.1
- PTT Pro client v3.2.10084
- Profile Manager client v 2.0.19406
- Voice Client v9.0.19409 (The Voice client does not require OAuth or SAML for authentication.)
- Profile Manager server v1.14.34

Firewall ports

The WFC-ACS is a cloud service that requires access to the following ports.

- TCP port 443 for the URL to the WFC-ACS service
- Web browser access to the WFC-ACS service

Configure WFC-ACS

You configure the WFC-ACS service by developing a configuration for OAuth communications and a configuration for SAML connectivity.

Create a Realm

The configuration requires creating a realm that contains both the Oauth configuration and the SAML configuration. The Oauth configuration is used by Workforce Connect applications and the SAML configuration is used to connect to the SAML server.

Configure the realm with two endpoints:

- OpenID Endpoint (OAuth): from the ACS service perspective this is called the Client.
- SAML2.0 IdP Endpoint: from the ACS service perspective this is called the Identity Provider.

Create the OAuth Client

The OAuth client communicates with the PTT Pro server and the Profile Manager using the OAuth protocol.

1. Configure the Client ID, Protocol (OAuth), Access Type (Confidential), and Redirect URI.
2. Configure the credentials. Select a client authentication of Client ID and secret (automatically generated), which correspond to the PTT Pro JSON parameters of `oClientId` and `oAuthClientSecret`.
3. Map the `username` parameter to `unique_username`, which is what the WFC system uses.

Create the Identity Provider

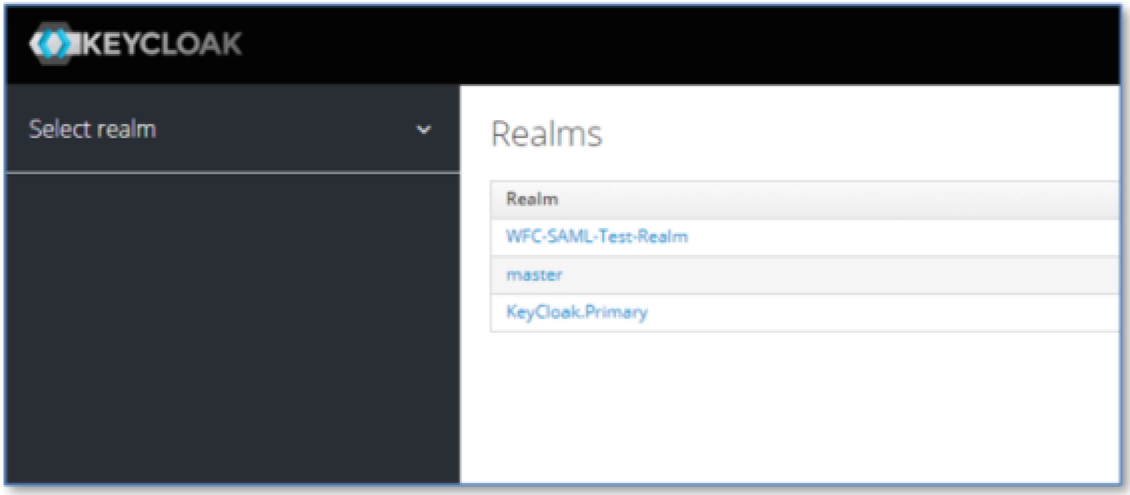
The identity provider communicates with the SAML server using the SAML protocol.

1. - Obtain the SAML descriptor file.
   - Configure the Single Sign-On Service URL.
   - Configure security settings such as Signature Validation (enable), the Signature Algorithm (RSA256), and the Validating x 509 Certificate.
   - Map the User ID entity from the IdP (SAML protocol) to the client (OAuth protocol).
   - Create a default authentication to automatically launch the IdP authentication.
2. Export certificates to the IdP and to the PTT Pro server.
   - Export the ACS SAML certificates to the SAML server.
     - Copy the certificate into a `.pem` file to the SAML server.
     - Import the `.pem` file into SAML server.
   - Export the ACS Realm certificate to the PTT Pro server and copy the certificate into the PTT Pro OAuth configuration.

Creating a Realm

Create a realm to contain the Oauth configuration and the SAML configuration. The Oauth configuration is used by Workforce Connect applications and the SAML configuration is used to connect to the SAML server.

1. Browse to the assigned WFC-ACS service and sign in.



Realm
WFC-SAML-Test-Realm
master
KeyCloak.Primary

Existing realms are shown in the right pane. There will always be at least one Master realm.

2. Create a new realm by clicking **Select Realm** in the left pane and then clicking on **Add Realm**.

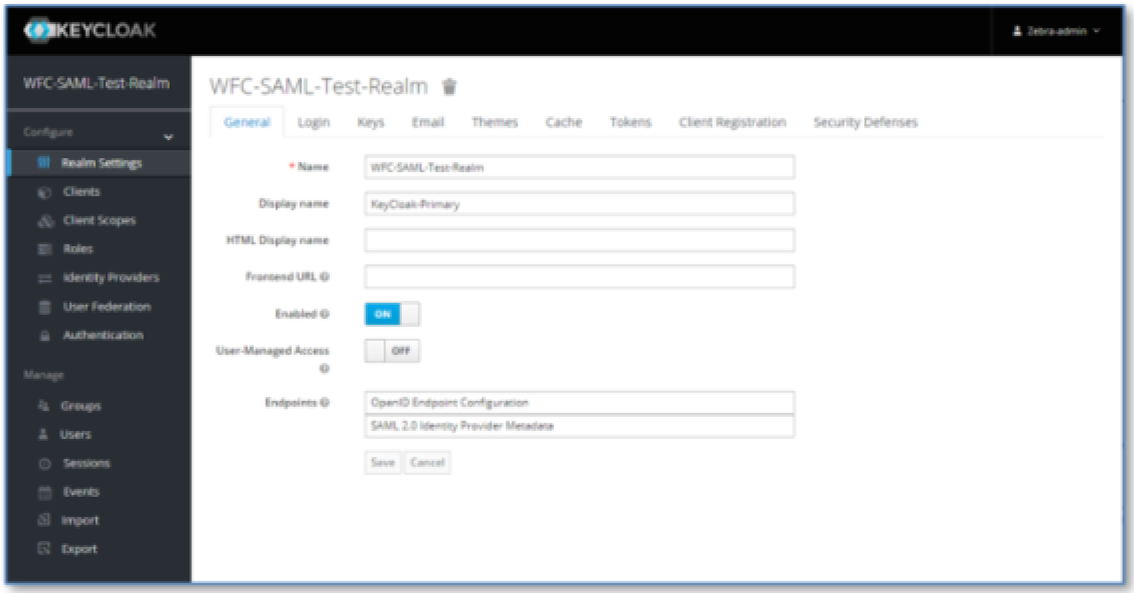
3. Configure the realm.

   a) Enter the Realm name in the **Name** field.

   b) Set **Enabled** to **ON**.

   c) Click **Create**.

   The new realm should automatically be selected in the left pane and additional tabs will appear. It is important to ensure you have switched to the correct realm in the left pane realm drop-down to prevent the configuration of the Master realm.



* Name: WFC-SAML-Test-Realm

Display name: KeyCloak-Primary

HTML Display name:

Frontend URL:

Enabled: ON

User-Managed Access: OFF

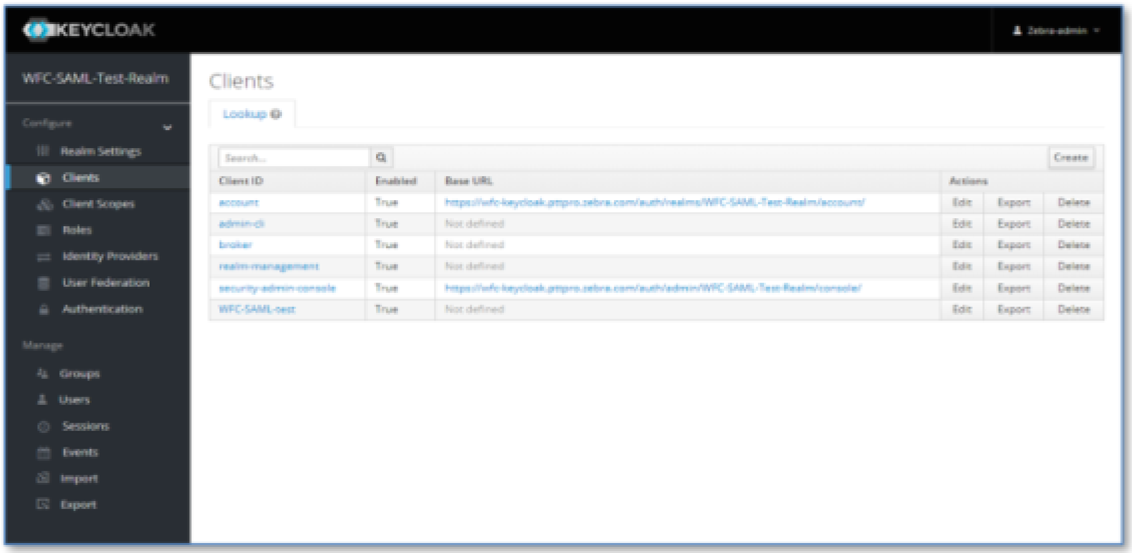
Endpoints: OpenID Endpoint Configuration
SAML 2.0 Identity Provider Metadata

Creating the Clients

Create the OAuth client used by the PTT Pro and Profile Manager servers.

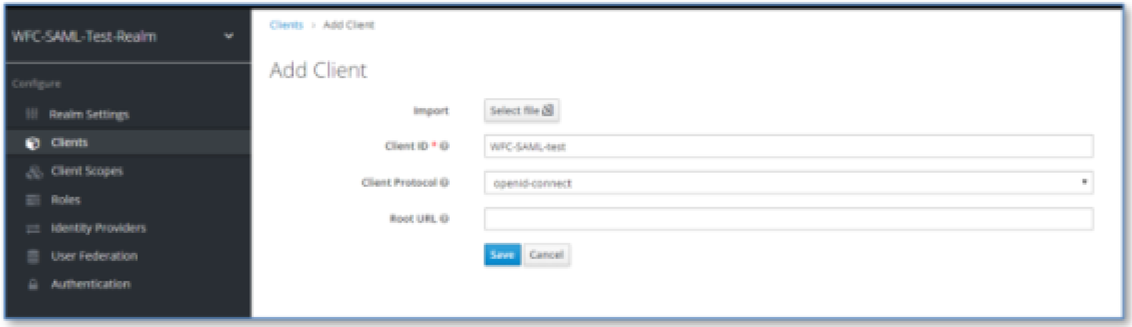
When you select **Clients**, the right pane displays the default client listing for the Realm. The **Client ID** column shows that the **account** and **security-admin-console** fields are automatically populated with the URL for the Realm.

1. Click **Create** to display the **Add Client** dialog.



Client ID	Enabled	Base URL	Actions		
account	True	https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/account/	Edit	Export	Delete
admin-cli	True	Not defined	Edit	Export	Delete
broker	True	Not defined	Edit	Export	Delete
realm-management	True	Not defined	Edit	Export	Delete
security-admin-console	True	https://wfc-keycloak.pttpro.zebra.com/auth/admin/WFC-SAML-Test-Realm/console/	Edit	Export	Delete
WFC-SAML-test	True	Not defined	Edit	Export	Delete

2. Configure the client.
   a) Enter a name in the **Client ID** field. This name is used in the PTT Pro client configuration JSON file as the value for the `oAuthClientID` parameter.
   b) Select `openid-connect` from the **Client Protocol** drop-down menu.
   c) Click **Save**.



Client ID *: WFC-SAML-test

Client Protocol: openid-connect

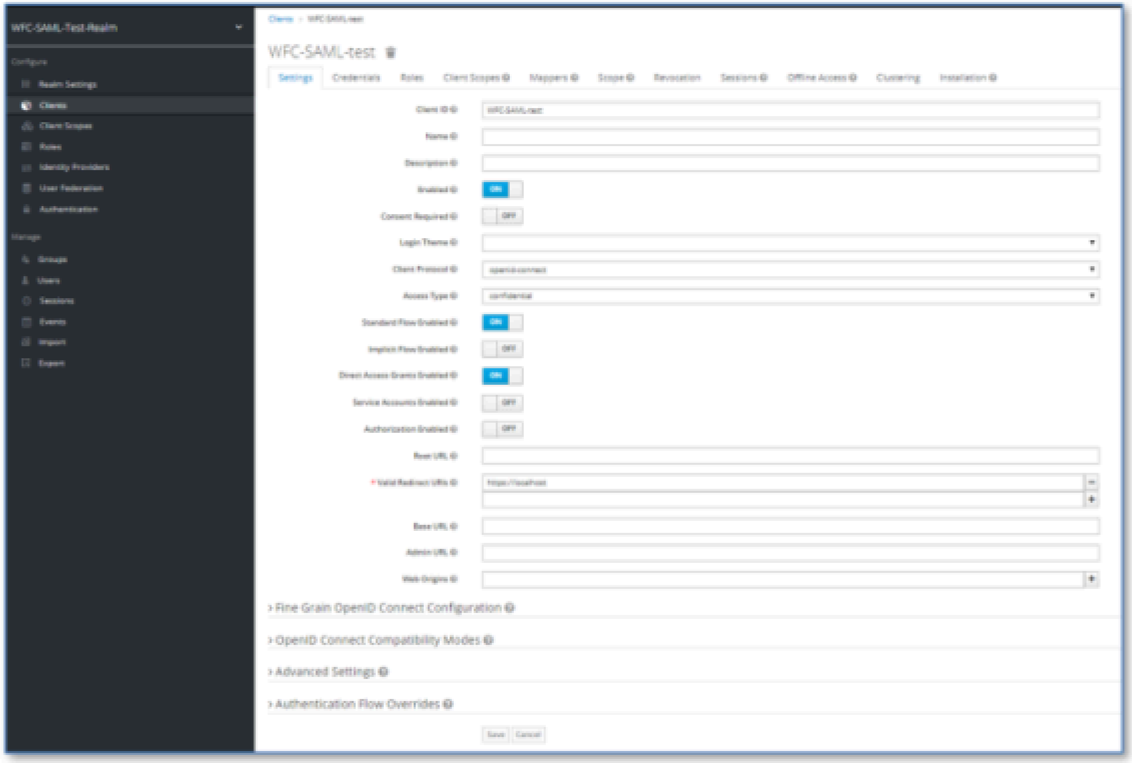
Root URL:

The **Endpoint Settings** tab appears.

Configuring the Endpoint Settings

Configure the settings for the OAuth client endpoint.

1. Set **Enabled** to **ON**.



WFC-SAML-test

Client ID: WFC-SAML-test
Name:
Description:
Enabled: ON
Consent Required: OFF
Login Theme:
Client Protocol: openid-connect
Access Type: confidential
Standard Flow Enabled: ON
Implicit Flow Enabled: OFF
Direct Access Grants Enabled: ON
Service Accounts Enabled: OFF
Authorization Enabled: OFF
Root URL:
* Valid Redirect URIs: https://localhost
Base URL:
Admin URL:
Web Origins:

› Fine Grain OpenID Connect Configuration
› OpenID Connect Compatibility Modes
› Advanced Settings
› Authentication Flow Overrides

2. Select **Confidential** from the **Access Type** drop-down menu.
3. Enter `https://localhost` in the **Valid Redirect URI** field.

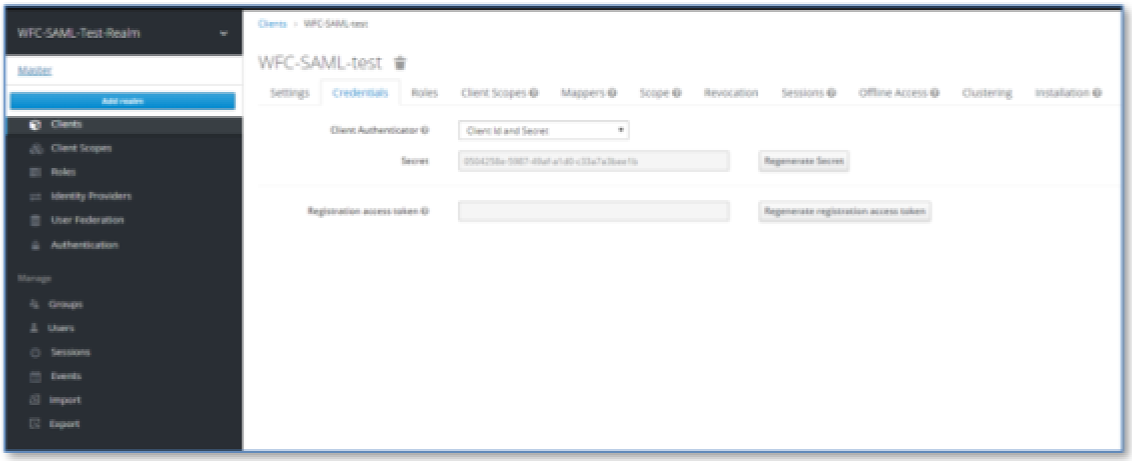
   This field is required but is not used for this implementation. It enables Keycloak to call back to its resource manager.
4. Click **Save** to display the **Credentials** tab.

Configuring the Endpoint Credentials

Configure the endpoint credentials to create the secret key.

1. Select **Client ID and Secret** from the **Client Authenticator** drop-down menu.

   This generates a random secret key.



Client Authenticator: Client Id and Secret

Secret: 0504258e-5987-49af-a1d0-c33a7a3bee1b

Registration access token:

2. Copy the values of the client name and the client secret key to configure the PTT Pro client.

   The secret key, along with the client name must be configured in the JSON configuration file of the PTT Pro client.

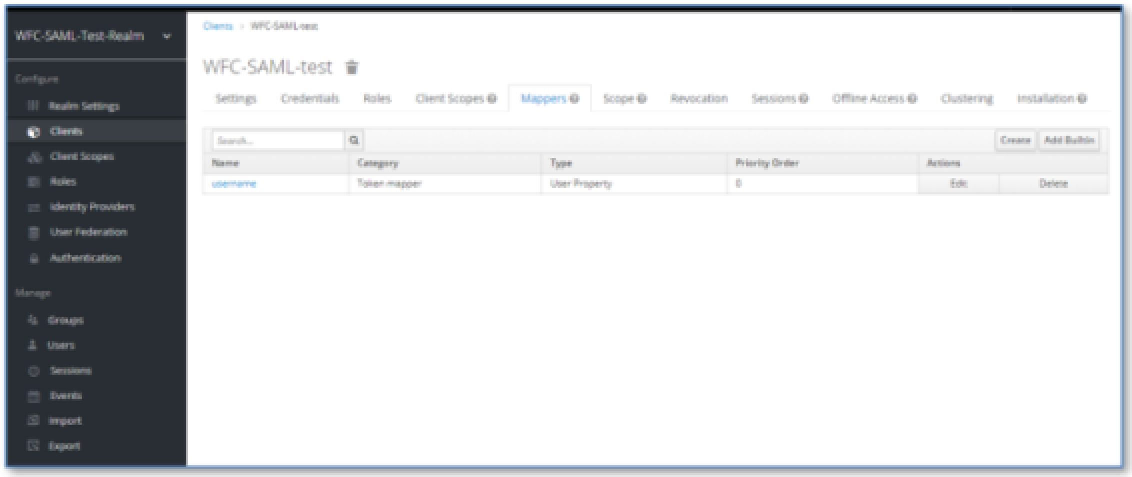
Example

```
"oAuthClientId": "WFC-SAML-test"
"oAuthClientSecret": "0504258e-5987-49af-a1d0-c33a7a3bee1b"
```

Configuring the Endpoint Mappings

Map the Oauth username to the SAML username.

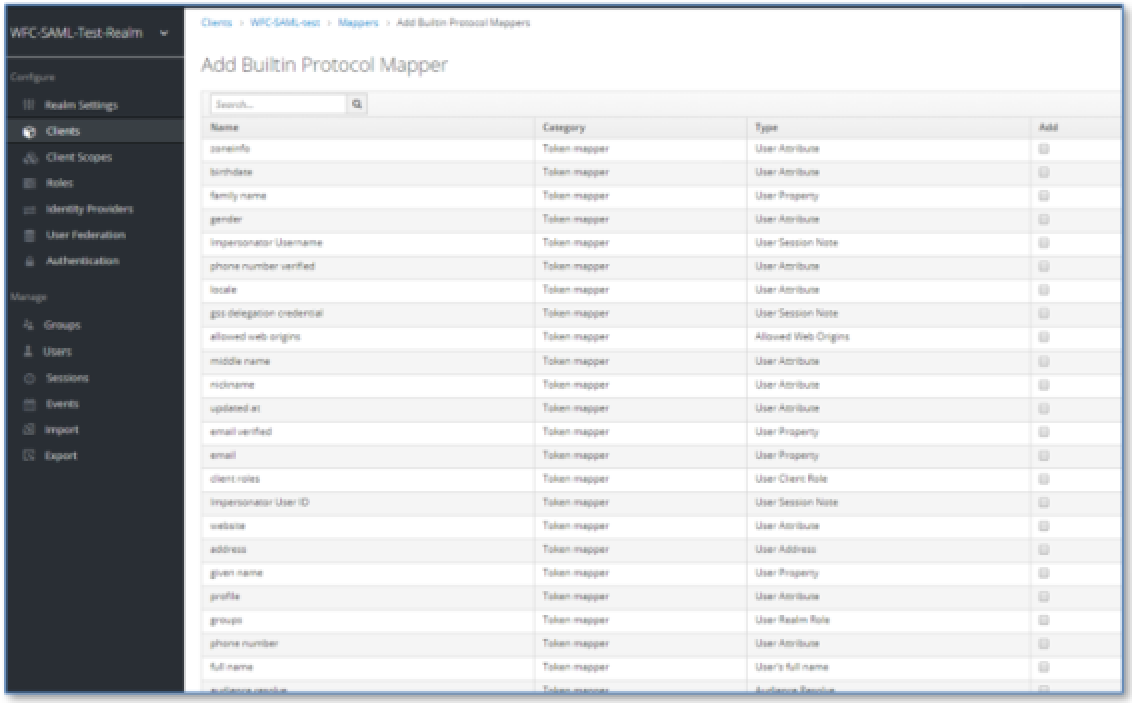
1. Select the **Mappers** tab and click on **Add Builtin**.



Name	Category	Type	Priority Order	Actions	
username	Token mapper	User Property	0	Edit	Delete

2. Scroll to the bottom of the list and click on **username** and select **Add Selected**.

   The username field is added to the **Mappers** tab.



Name	Category	Type	Add
zoneinfo	Token mapper	User Attribute	□
birthdate	Token mapper	User Attribute	□
family name	Token mapper	User Property	□
gender	Token mapper	User Attribute	□
impersonator Username	Token mapper	User Session Note	□
phone number verified	Token mapper	User Attribute	□
locale	Token mapper	User Attribute	□
gss delegation credential	Token mapper	User Session Note	□
allowed web origins	Token mapper	Allowed Web Origins	□
middle name	Token mapper	User Attribute	□
nickname	Token mapper	User Attribute	□
updated at	Token mapper	User Attribute	□
email verified	Token mapper	User Property	□
email	Token mapper	User Property	□
client roles	Token mapper	User Client Role	□
impersonator User ID	Token mapper	User Session Note	□
website	Token mapper	User Attribute	□
address	Token mapper	User Address	□
given name	Token mapper	User Property	□
profile	Token mapper	User Attribute	□
groups	Token mapper	User Realm Role	□
phone number	Token mapper	User Attribute	□
full name	Token mapper	User's full name	□
audience resolve	Token mapper	Audience Resolve	□

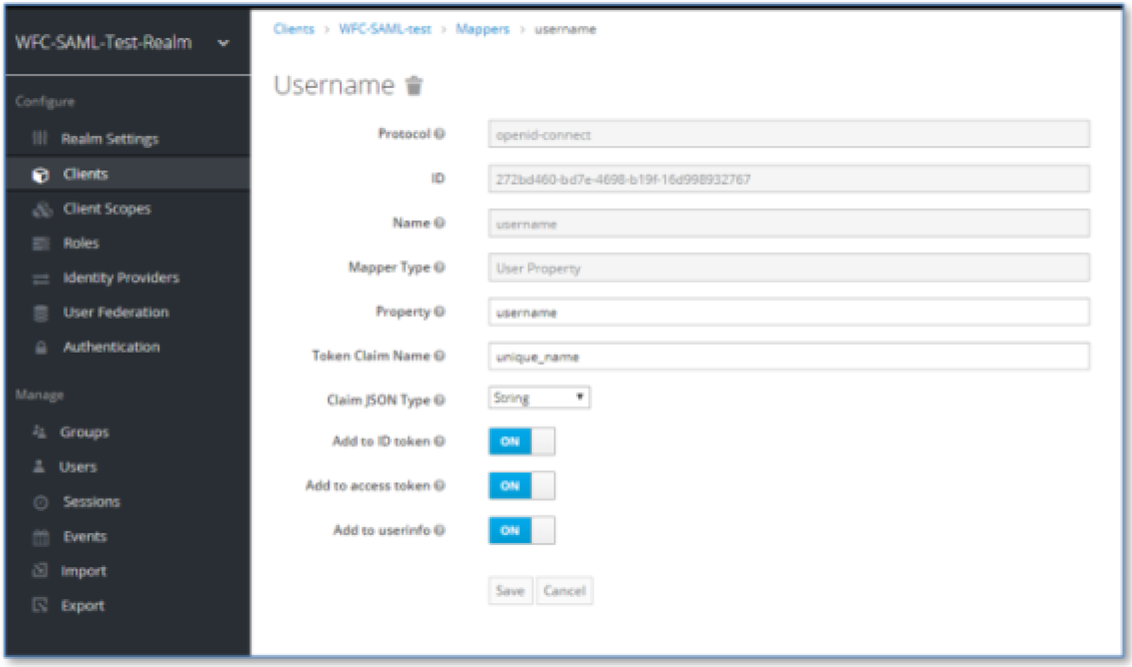
3. Click on **username** to edit and enter the following fields:
   a) Enter `openid-protocol` in the **Protocol** field.
   b) Enter `username` in the **Property** fied.

NOTE: This value might be different based on your implementation.

   c) Change the **Token Claim Name** field to `unique_name`. from to 'Unique_name'

   The default is `preferred_username`.

   d) Enable the **Add to ID token**, **Add to access token**, and **Add to userinfo**.

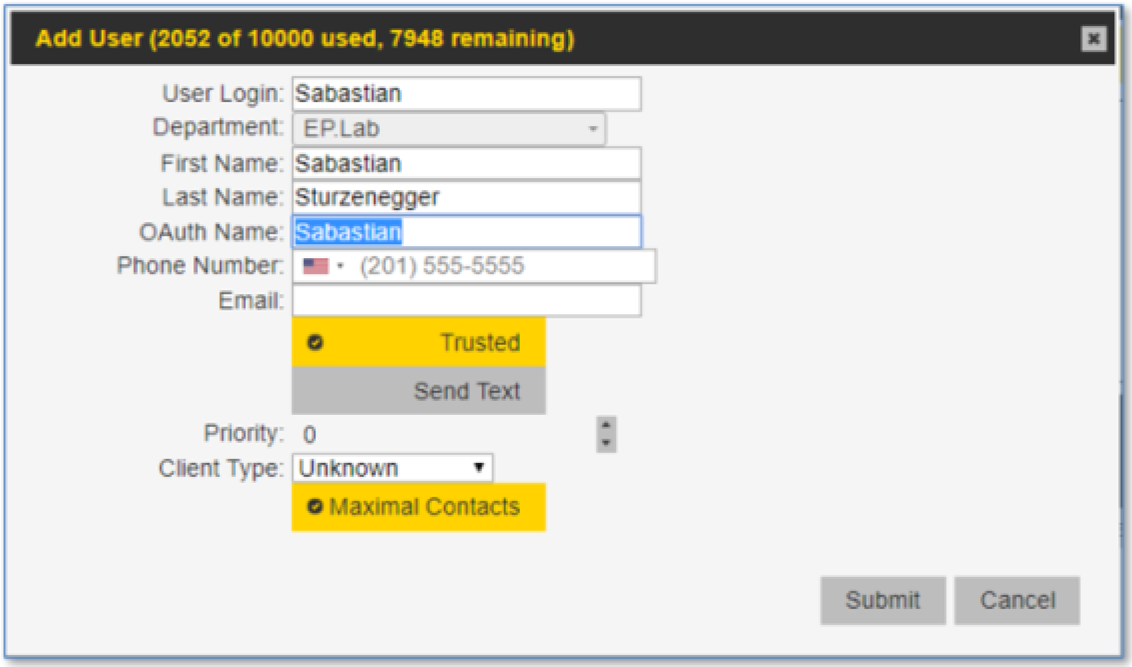


Username

Protocol	openid-connect
ID	272bd460-bd7e-4698-b19f-16d998932767
Name	username
Mapper Type	User Property
Property	username
Token Claim Name	unique_name
Claim JSON Type	String
Add to ID token	ON
Add to access token	ON
Add to userinfo	ON

4. Click **Save**.

The SAML username is returned as the **OAuth Name** in each user definition of the PTT Pro server.



Add User (2052 of 10000 used, 7948 remaining)

User Login: Sabastian
Department: EP.Lab
First Name: Sabastian
Last Name: Sturzenegger
OAuth Name: Sabastian
Phone Number: (201) 555-5555
Email:
Priority: 0
Client Type: Unknown

SAML Descriptor File

The SAML Descriptor file provides information needed to configure the WFC-ACS servic.

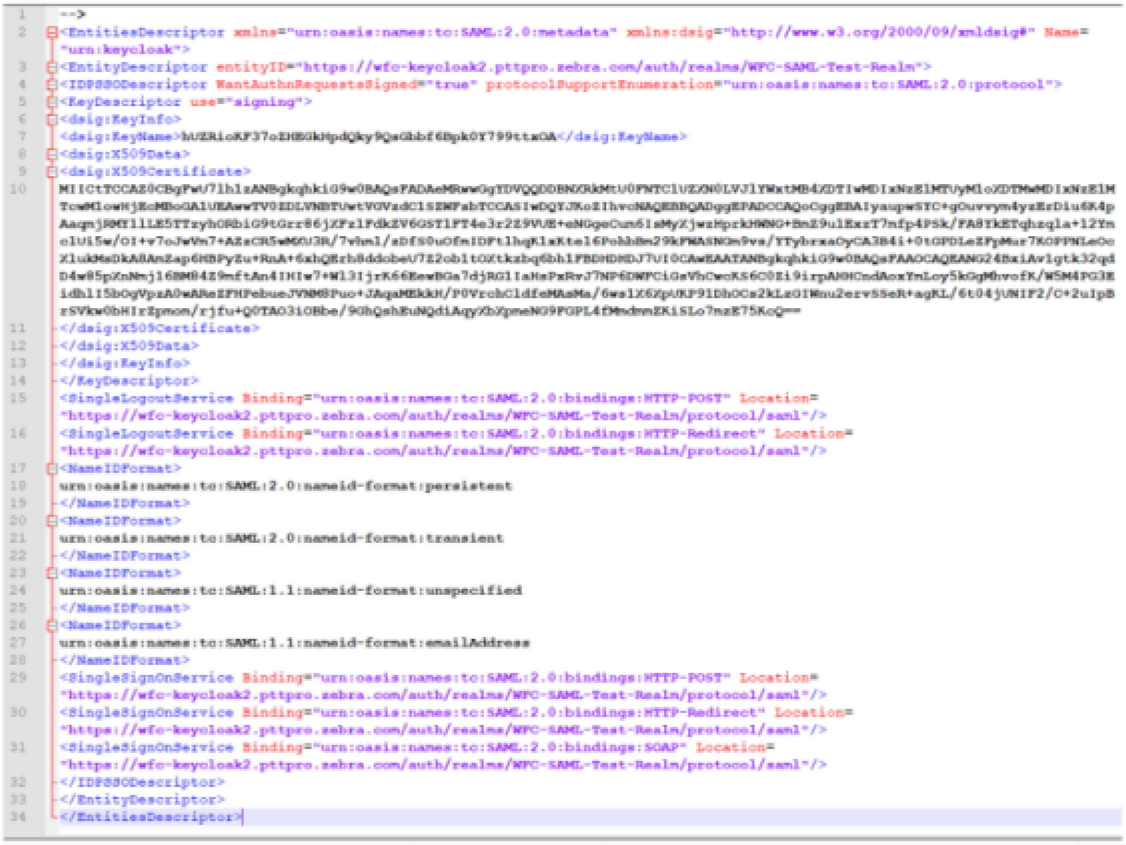
A sample SAML Descriptor file is shown below. You can also view a sample file at the Sample Descriptor File link.

You can access the Descriptor file from the Keycloak user interface. Navigate to the **Realm Settings** > **General** tab. Click on the **SAML 2.0 IdP Metadata** field under **Endpoints** to reveal the descriptor file.)

In the example Descriptor file below, the following lines contain the information needed to configure the WFC-ACS service.

- The tag `<dsig:x509Certificate>` on line 10 contains the x509 certificate to import into the WFC-ACS service to provide access to the IdP.
- The tag `<SingleLogoutService>` on line 15 contains the URL to be copied into the SAML Single Logout Service URL in the WFC-ACS service.
- The tag `<SingleSignOnService>` on line 29 contains the URL to copy into the SAML Single Sign-On Service URL in the WFC-ACS service.

Figure 5 SAML Descriptor File

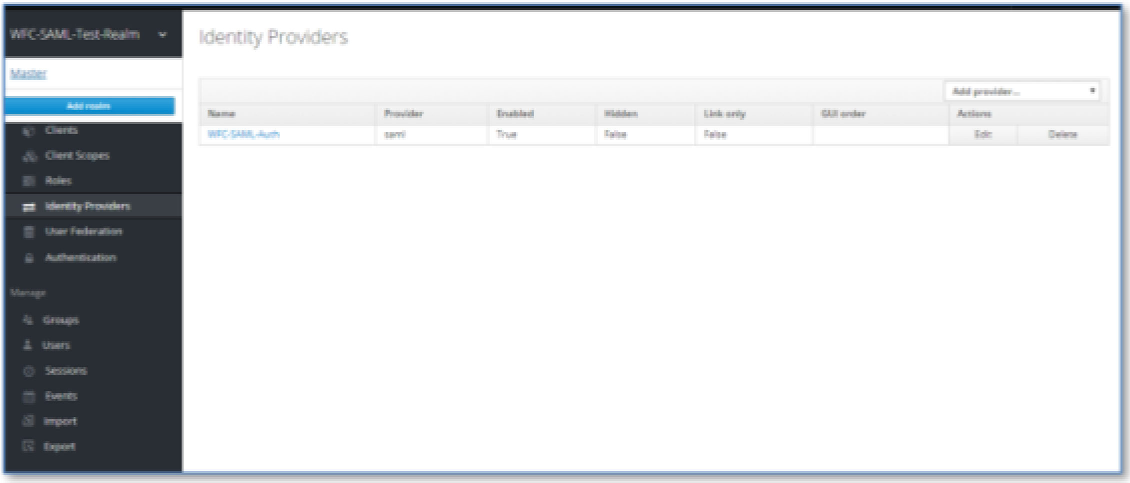


```
-->
<EntitiesDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Name=
"urn:keycloak">
<EntityDescriptor entityID="https://wfc-keycloak2.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm">
<IDPSSODescriptor WantAuthnRequestsSigned="true" protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<KeyDescriptor use="signing">
<dsig:KeyInfo>
<dsig:KeyName>hUZRioKF37oEHEGkHpdQky9QsGbbf6Bpk0Y799ttxOA</dsig:KeyName>
<dsig:X509Data>
<dsig:X509Certificate>
MIICtTCCAZ0CBgFwU7lhlzANBgkqhkiG9w0BAQsFADAeMRwwGgYDVQQDDBNXRkMtU0FNTC1UZXN0LVJlYWxtMB4XDTIwMDIxNzE1MTUyMloXDTMwMDIxNzE1M
TcwMlowHjEcMBoGA1UEAwwTV0ZDLVNBTUwtVGVzdC1SZWFsbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAIyaupwSTC+gOuvvym4yzErDiu6K4p
AaqnjRMYllLE5TTzyhORbiG9tGrr86jXFzlFdkZV6GST1FT4e3r2Z9VUE+eNGgeCun6IsMyXjwzHprkHWNG+Bn29ulExzT7nfp4PSk/FA8TkETqhzqla+12Yn
olUi5w/OI+v7oJwVm7+AZsCR5wMXU3R/7vhml/zDfS0uOfmIDFtlhqKlxKtel6PohbBm29kFWASNGm9vs/YTybrxaOyCA3B4i+0tGPDLeZFpMur7KOFPNLeOo
XlukMsDkA8AmZap6HBPyZu+RnA+6xhQErh8ddobeU7Z2obltOXtkzbq6bh1FBDHDHDJ7UI0CAwEAATANBgkqhkiG9w0BAQsFAAOCAQEANG24BxiAvlgtk32qd
D4w85pXnNmj16BM84Z9mftAn4IHIw7+W131jrK66EewBGa7djRG1IaHsPxRvJ7NP6DWPCiGsVhCwoKS6C0Zi9irpAH0CndAoxYmLoy5kGgMhvofK/W5M4PG3E
idhl15bOgVpzA0wAReZFHPebueJVNM8Puo+JAqaMEkkH/P0VrchCldfeMAsMa/6ws1X6ZpUKF91DhOCs2kLzGIWnu2erv55eR+agRL/6t04jUNIF2/C+2uIpB
rSVkw0bHIrZpnom/rjfu+Q0TAO3iOBbe/9GhQshEuNQdiAqyXbXpmeNG9PGPL4fMndmnZKiSLo7mzE75KoQ==
</dsig:X509Certificate>
</dsig:X509Data>
</dsig:KeyInfo>
</KeyDescriptor>
<SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location=
"https://wfc-keycloak2.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/saml"/>
<SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect" Location=
"https://wfc-keycloak2.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/saml"/>
<NameIDFormat>
urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
</NameIDFormat>
<NameIDFormat>
urn:oasis:names:tc:SAML:2.0:nameid-format:transient
</NameIDFormat>
<NameIDFormat>
urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
</NameIDFormat>
<NameIDFormat>
urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
</NameIDFormat>
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location=
"https://wfc-keycloak2.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/saml"/>
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect" Location=
"https://wfc-keycloak2.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/saml"/>
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP" Location=
"https://wfc-keycloak2.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/saml"/>
</IDPSSODescriptor>
</EntityDescriptor>
</EntitiesDescriptor>
```

Configuring the Identity Provider

Use the information from the SAML Descriptor file to configure the Identity Provider.

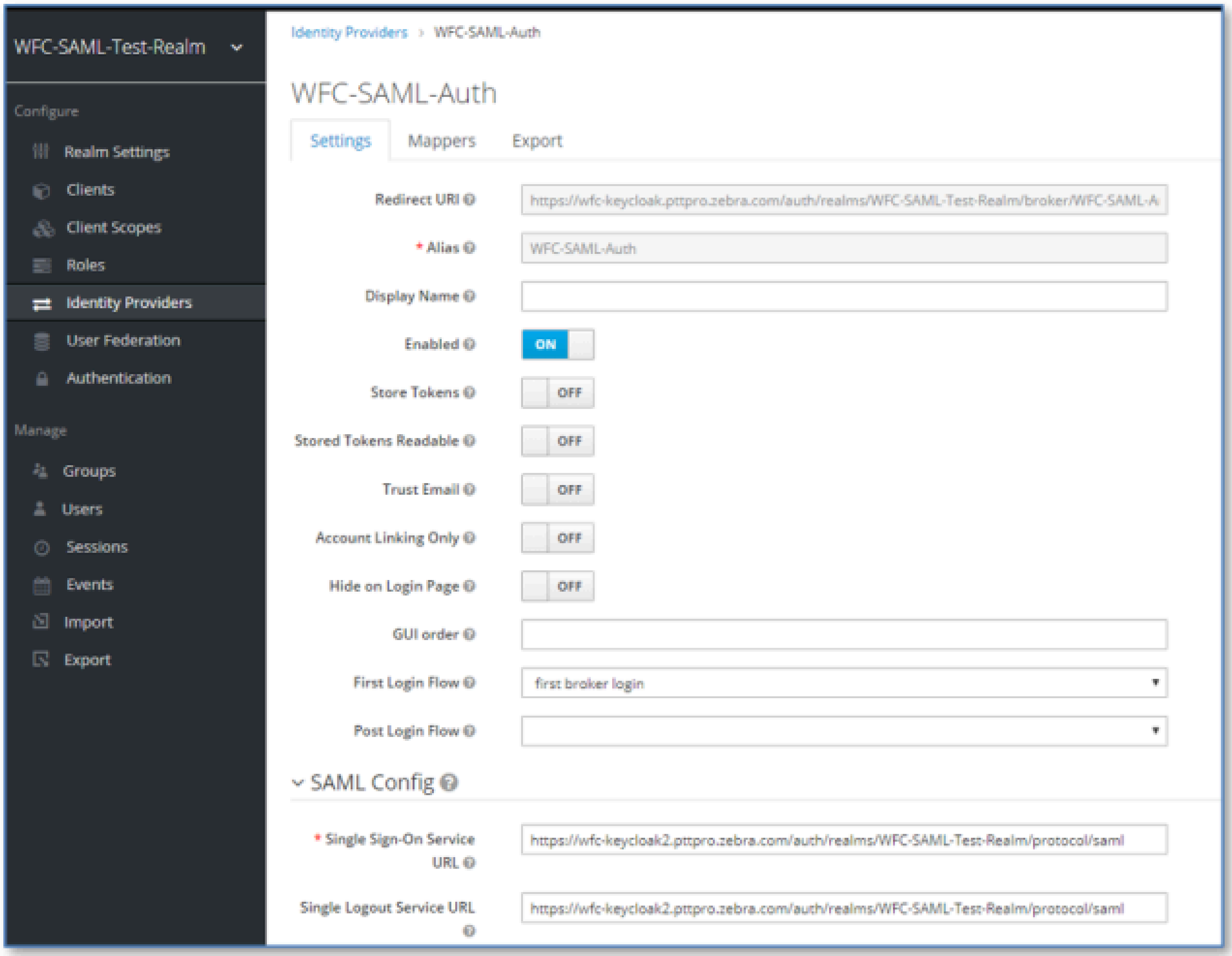
1. Click on **Identity Providers** in the left pane of the Keycloak user interface.



Name	Provider	Enabled	Hidden	Link only	GUI order	Actions	
WFC-SAML-Auth	saml	True	False	False		Edit	Delete

2. Select **SAML v2.0** from the Add Provider drop down menu.

3. Configure Oauth.



WFC-SAML-Auth

Redirect URI: https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/broker/WFC-SAML-A

* Alias: WFC-SAML-Auth

Display Name:

Enabled: ON

Store Tokens: OFF

Stored Tokens Readable: OFF

Trust Email: OFF

Account Linking Only: OFF

Hide on Login Page: OFF

GUI order:

First Login Flow: first broker login

Post Login Flow:

SAML Config

* Single Sign-On Service URL: https://wfc-keycloak2.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/saml

Single Logout Service URL: https://wfc-keycloak2.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/saml

a) Enter the redirect URL in the **Redirect URI** field.

The Redirect URI entry is constructed as follows:

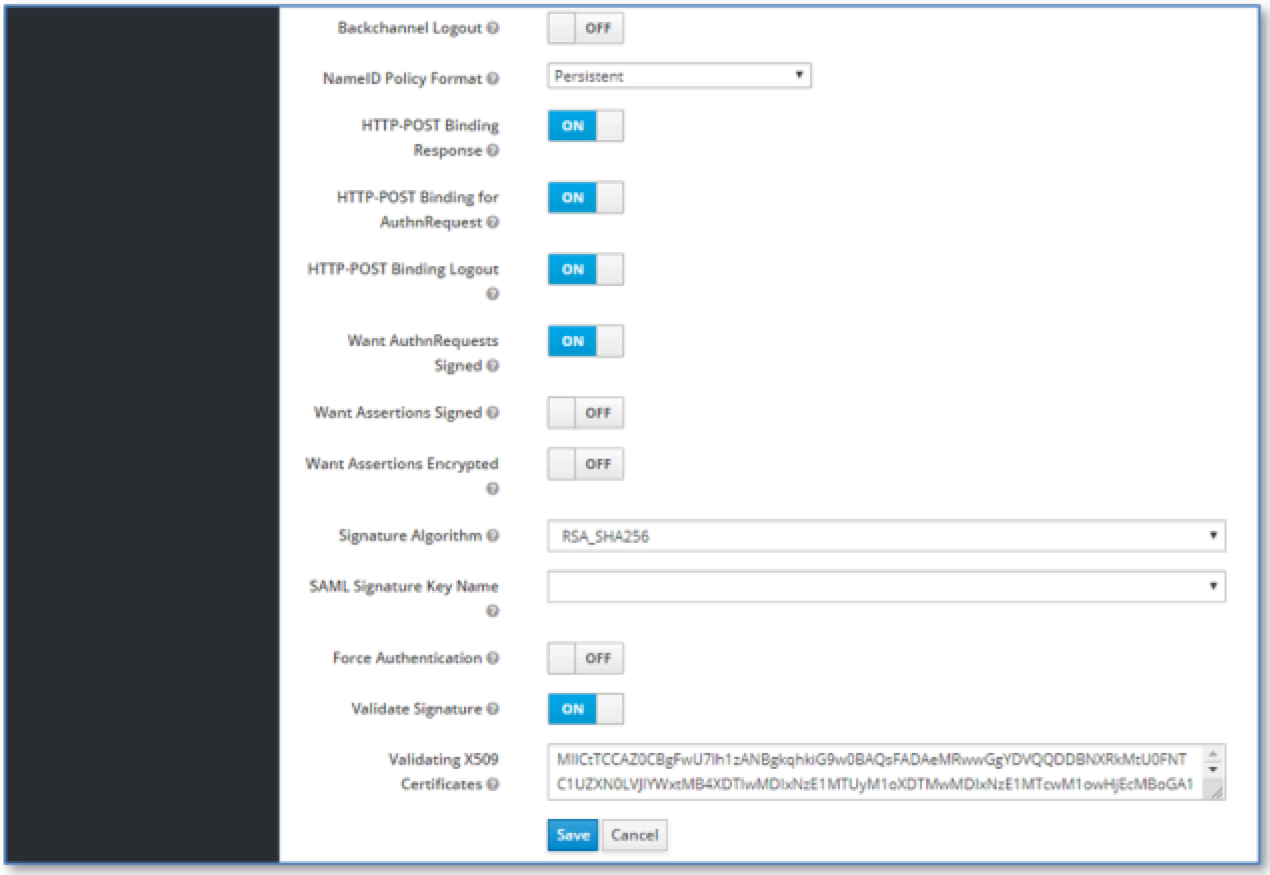
```
https://wfc-keycloak.pttpro.zebra.com/auth/realms/<realm-name>/broker/
<alias>/endpoint
```

The <`Alias`>is a convenient label. Change it to an appropriate value. In this example, the alias is `WFC-SAML-Auth` that results in the following redirect URI:

```
https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/
broker/WFC-SAML-Auth/endpoint
```

b) Set **Enable** to on.

4. Configure SAML.



Backchannel Logout	OFF
NameID Policy Format	Persistent
HTTP-POST Binding Response	ON
HTTP-POST Binding for AuthnRequest	ON
HTTP-POST Binding Logout	ON
Want AuthnRequests Signed	ON
Want Assertions Signed	OFF
Want Assertions Encrypted	OFF
Signature Algorithm	RSA_SHA256
SAML Signature Key Name	
Force Authentication	OFF
Validate Signature	ON
Validating X509 Certificates	MIICzTCCAZ0CBgFwU7Ih1zANBgkqhkiG9w0BAQsFADAeMRwwGgYDVQQDDBNXRkMtU0FNT C1UZXN0LVJlYWxtMB4XDTIwMDIxNzE1MTUyM1oXDTMwMDIxNzE1MTcwM1owHjEcMBoGA1

a) Enter the single sign-on URL in the **Single Sign-On URL Service** field.

The Single Sign-On Service URL can be copied from the `<SingleSignOnService>` paramter in the SAML Descriptor file. In this example it is:

```
https://wfc-keycloak2.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/
protocol/saml
```

b) Enter the single logout service URL in the **Single Logout Service URL** field.

The Single Logout Service URL can be copied from the `<SingleLogoutService>` parameter in the SAML Description file.

c) Enable **Validate Signature**.

d) Enable **HTTP-POST Binding Response**.

e) Enable **HTTP-POST Binding for AuthnRequest**.

f) Enable **HTTP-POST Binding Logout**.

g) Enable **Want AuthnRequests Signed**.

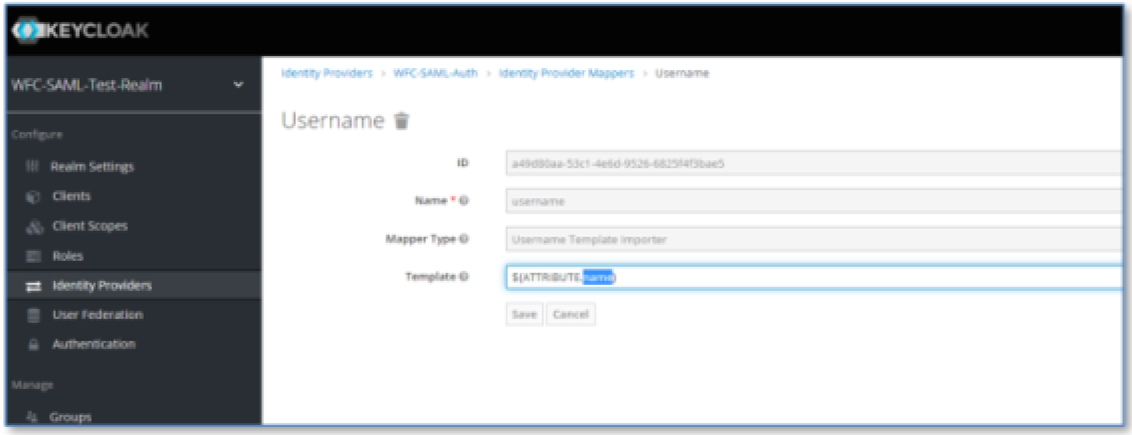
5. Click **Save**.

Entering these URLs causes the WFC-ACS service to read the descriptor file and populate the x509 Certificate and the Signature Algorithm fields.

Mapping the User Name from the IdP

WFC-ACS uses the user name for the roles and profile assignments in the WFC environment. The user name is mapped to the correct profile on the device when the user signs in.

1. Select the **Mappers** tab.



ID: a49d80aa-53c1-4e6d-9526-6825f4f3bae5

Name *: username

Mapper Type: Username Template Importer

Template: ${ATTRIBUTE.name}

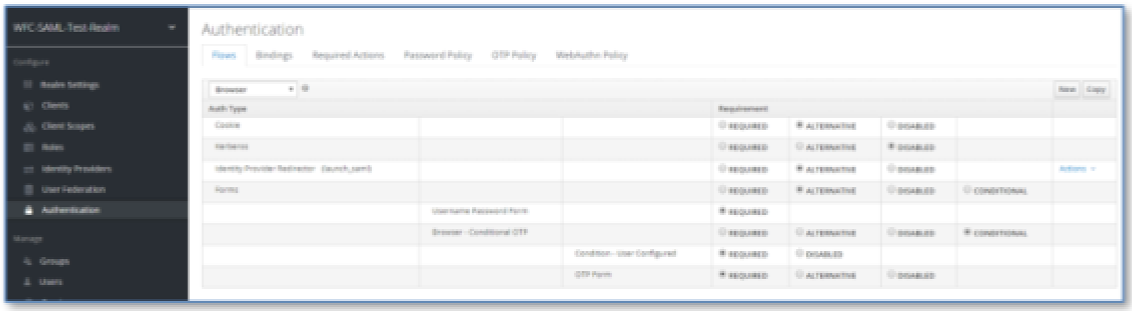
2. Enter `username` in the **Name** field.
3. Enter `${Attribute.name}` in the **Template** field.

   This enables the name field from the IdP to pass to the OAuth services. The attribute name comes from the IdP and may be different in your environment.

Auto Launching the SAML Login

Configure SAML to automatically launch the SAML login page when the mobile device connects to the WFC-ACS.

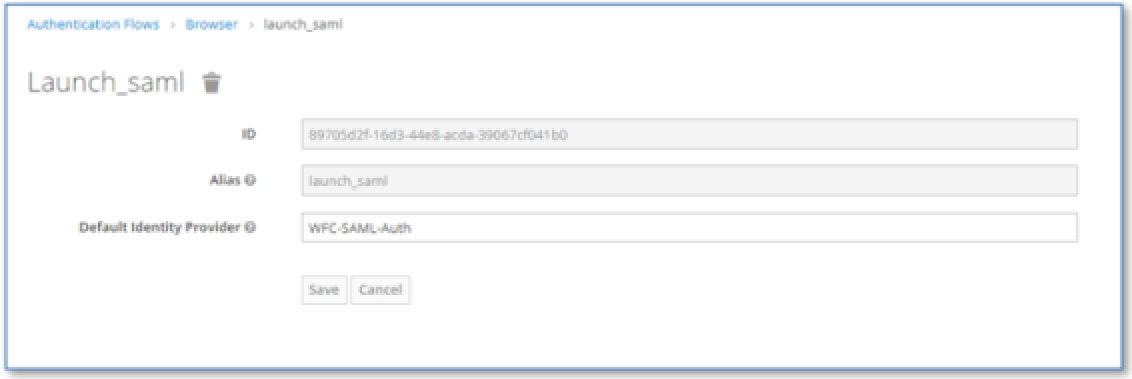
1. Navigate to the **Authentication** view.



Auth Type			Requirement				
Cookie			○ REQUIRED	● ALTERNATIVE	○ DISABLED		
Kerberos			○ REQUIRED	○ ALTERNATIVE	● DISABLED		
Identity Provider Redirector (launch_saml)			○ REQUIRED	● ALTERNATIVE	○ DISABLED		Actions
Forms			○ REQUIRED	● ALTERNATIVE	○ DISABLED	○ CONDITIONAL	
	Username Password Form		● REQUIRED				
	Browser - Conditional OTP		○ REQUIRED	○ ALTERNATIVE	○ DISABLED	● CONDITIONAL	
		Condition - User Configured	● REQUIRED	○ DISABLED			
		OTP Form	● REQUIRED	○ ALTERNATIVE	○ DISABLED		

2. Select the **Flows** tab.
3. Select **Browser** from the drop-down menu.

4. Select **Config** from the **Actions** drop-down menu in the **Identity Provider Redirector** row.



Authentication Flows > Browser > launch_saml

Launch_saml

ID: 89705d2f-16d3-44e8-acda-39067cf041b0

Alias: launch_saml

Default Identity Provider: WFC-SAML-Auth

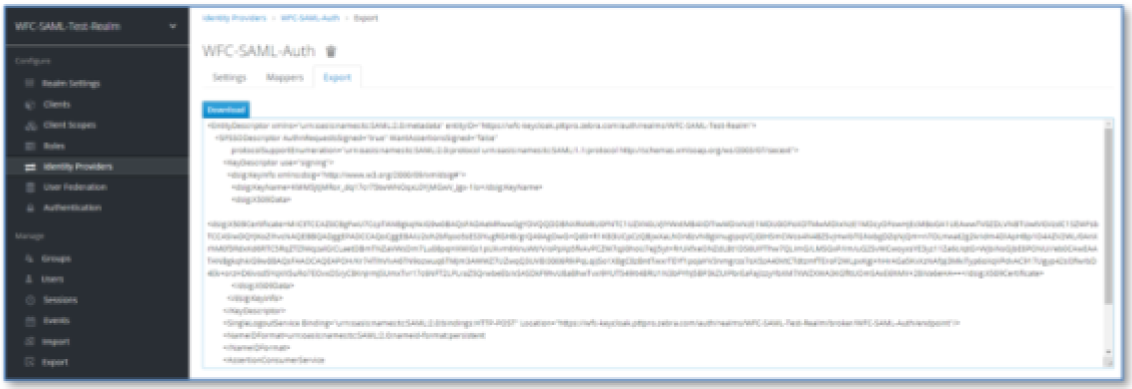
5. Enter a name in the **Alias** field.
6. Enter the client string of the Identity Provider in the **Default Identity Provider** field.

   See Configuring the Identity Provider on page 17 for the client string. In this example, the client string is WFC-SAML-Auth.
7. Click **Save** and then **Close**.

Exporting the WFC-ACS Certificate to SAML

Establish trust from the WFC-ACS to the SAML server.

1. Navigate to the **Identity Providers** view.
2. Click **Export** to display the descriptor file from the ACS instance.



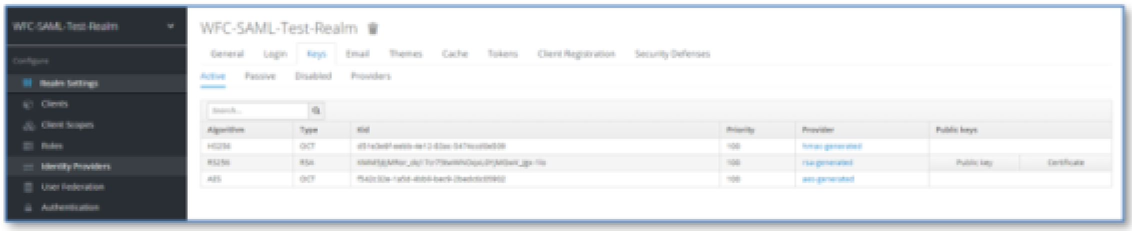
```
<EntityDescriptor [illegible]>
  <SPSSODescriptor [illegible]
      protocolSupportEnumeration="[illegible]">
    <KeyDescriptor use="signing">
      <dsig:KeyInfo [illegible]>
        <dsig:KeyName>[illegible]</dsig:KeyName>
        <dsig:X509Data>

<dsig:X509Certificate>[illegible]</dsig:X509Certificate>
        </dsig:X509Data>
      </dsig:KeyInfo>
    </KeyDescriptor>
    <SingleLogoutService Binding="[illegible]" Location="[illegible]"/>
    <NameIDFormat>[illegible]
    </NameIDFormat>
    <AssertionConsumerService
```

3. Copy the x509 Certificate and paste into a text file with a file extension of `<filename>.pem`.
4. Import `<filename>.pem` into the SAML server.

Exporting the WFC-ACS Certificate to PTT Pro

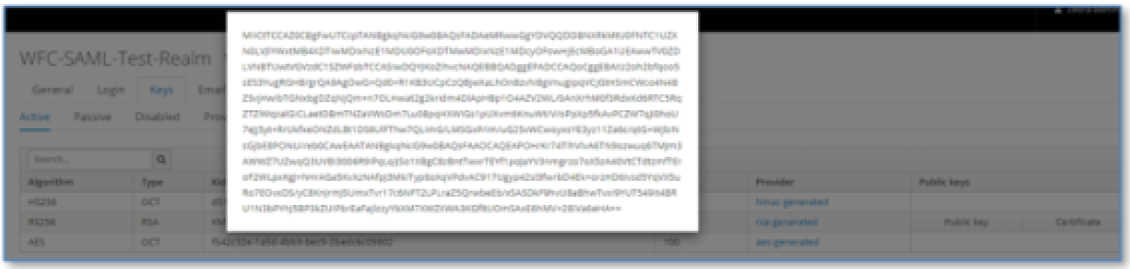
Export the WFC-ACS certificate to the PTT Pro server. The PTT Pro server requires a certificate for the OAuth connection to Keycloak.

1. Navigate to the **Realm Settings** and select the **Keys** tab.



Algorithm	Type	Kid	Priority	Provider	Public keys	
HS256	OCT	[illegible]	100	hmac-generated		
RS256	RSA	[illegible]	100	rsa-generated	Public key	Certificate
AES	OCT	[illegible]	100	aes-generated		

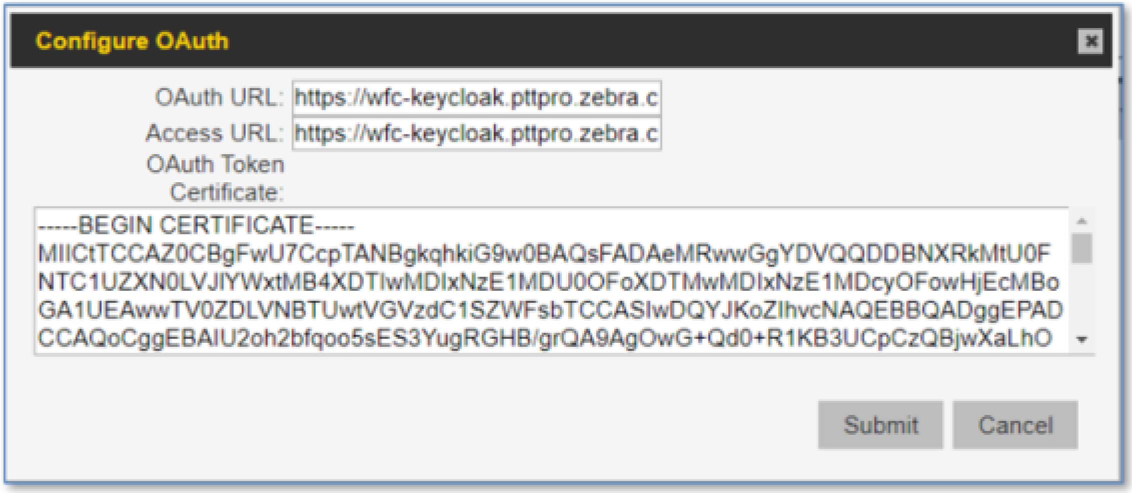
2. Click on **Certificate** in the RS256 row.



3. Copy the certificate to the Oauth definition in the PTT Pro portal as shown below.

NOTE: The CR/LF characters are critical for the server to properly digest the certificate.

```
-----Begin Certificate----- <CR/LF>
<Certificate information pasted here> <CR/LF>
-----End Certificate-----
```



OAuth URL: https://wfc-keycloak.pttpro.zebra.c

Access URL: https://wfc-keycloak.pttpro.zebra.c

OAuth Token Certificate:

```
-----BEGIN CERTIFICATE-----
MIICtTCCAZ0CBgFwU7CcpTANBgkqhkiG9w0BAQsFADAeMRwwGgYDVQQDDBNXRkMtU0F
NTC1UZXN0LVJlYWxtMB4XDTIwMDIxNzE1MDU0OFoXDTMwMDIxNzE1MDcyOFowHjEcMBo
GA1UEAwwTV0ZDLVNBTUwtVGVzdC1SZWFsbTCCASIwDQYJKoZIhvcNAQEBBQADggEPAD
CCAQoCggEBAIU2oh2bfqoo5sES3YugRGHB/grQA9AgOwG+Qd0+R1KB3UCpCzQBjwXaLhO
```

4. Click **Submit**.

Configure Workforce Connect

After you have configure the WFC-ACS service, configure the Workforce Connect servers and clients.

This section describes the configuration of:

- Profile Manager
- PTT Pro Server
- PTT Pro for Android
- Profile Client

Configure Profile Manager

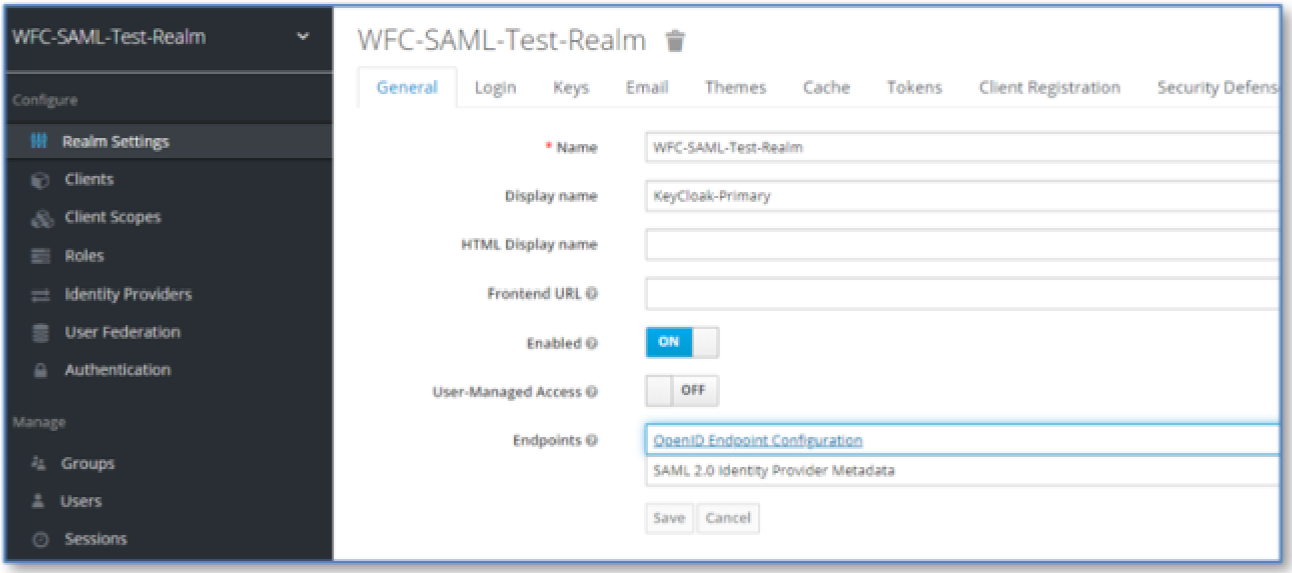
The Profile Manager server requires four configuration elements to connect to an OAuth server because the Profile Manager uses OAuth to authorize users. Using WFC-ACS requires that the Profile Manager use the Keycloak server instead of an ADFS server.

The four configuration elements you need to configure Profile Manager administrator are:

- Authentication URL
- Access Token URL
- Client ID
- Client secret

Authentication URL and Access Token URL

The Authentication URL and the Access Token URL are obtained by clicking on the **OpenID Endpoint Config** under the **General** tab in the **Realm Settings**.



* Name: WFC-SAML-Test-Realm

Display name: KeyCloak-Primary

HTML Display name:

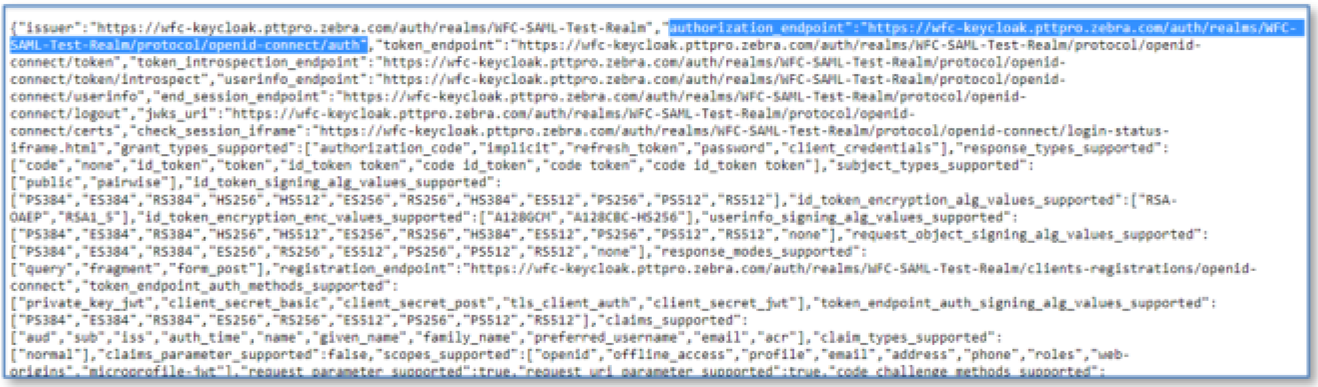
Frontend URL:

Enabled: ON

User-Managed Access: OFF

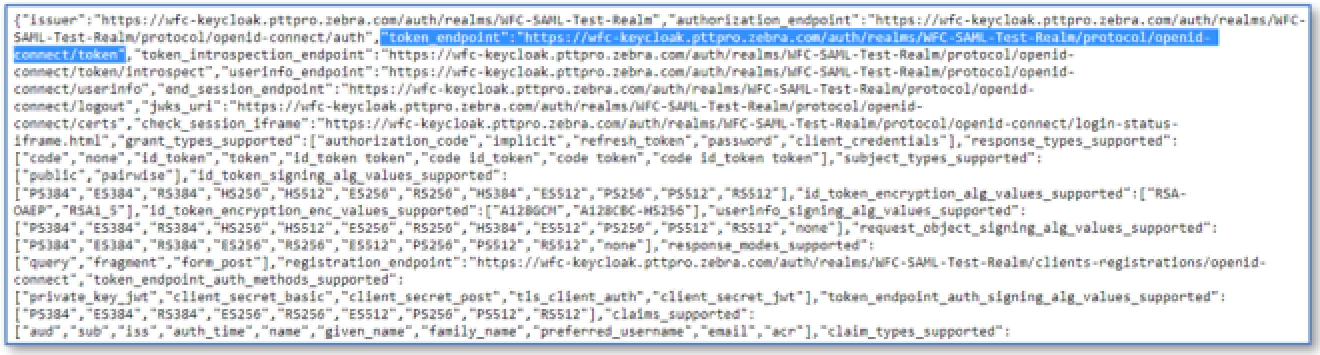
Endpoints: OpenID Endpoint Configuration
SAML 2.0 Identity Provider Metadata

The Authorization Endpoint is copied to the OAuth URL in the PTT Pro server and provided to the Profile Manager administrator.



```
{"issuer":"https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm","authorization_endpoint":"https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-
SAML-Test-Realm/protocol/openid-connect/auth","token_endpoint":"https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/openid-
connect/token","token_introspection_endpoint":"https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/openid-
connect/token/introspect","userinfo_endpoint":"https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/openid-
connect/userinfo","end_session_endpoint":"https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/openid-
connect/logout","jwks_uri":"https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/openid-
connect/certs","check_session_iframe":"https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/openid-connect/login-status-
iframe.html","grant_types_supported":["authorization_code","implicit","refresh_token","password","client_credentials"],"response_types_supported":
["code","none","id_token","token","id_token token","code id_token","code token","code id_token token"],"subject_types_supported":
["public","pairwise"],"id_token_signing_alg_values_supported":
["PS384","ES384","RS384","HS256","HS512","ES256","RS256","HS384","ES512","PS256","PS512","RS512"],"id_token_encryption_alg_values_supported":["RSA-
OAEP","RSA1_5"],"id_token_encryption_enc_values_supported":["A128GCM","A128CBC-HS256"],"userinfo_signing_alg_values_supported":
["PS384","ES384","RS384","HS256","HS512","ES256","RS256","HS384","ES512","PS256","PS512","RS512","none"],"request_object_signing_alg_values_supported":
["PS384","ES384","RS384","ES256","RS256","ES512","PS256","PS512","RS512","none"],"response_modes_supported":
["query","fragment","form_post"],"registration_endpoint":"https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/clients-registrations/openid-
connect","token_endpoint_auth_methods_supported":
["private_key_jwt","client_secret_basic","client_secret_post","tls_client_auth","client_secret_jwt"],"token_endpoint_auth_signing_alg_values_supported":
["PS384","ES384","RS384","ES256","RS256","ES512","PS256","PS512","RS512"],"claims_supported":
["aud","sub","iss","auth_time","name","given_name","family_name","preferred_username","email","acr"],"claim_types_supported":
["normal"],"claims_parameter_supported":false,"scopes_supported":["openid","offline_access","profile","email","address","phone","roles","web-
origins","microprofile-jwt"],"request_parameter_supported":true,"request_uri_parameter_supported":true,"code_challenge_methods_supported":
```

The Token Endpoint URL is copied to the Access URL in the PTT Pro server must be provided to the Profile Manager administrator.

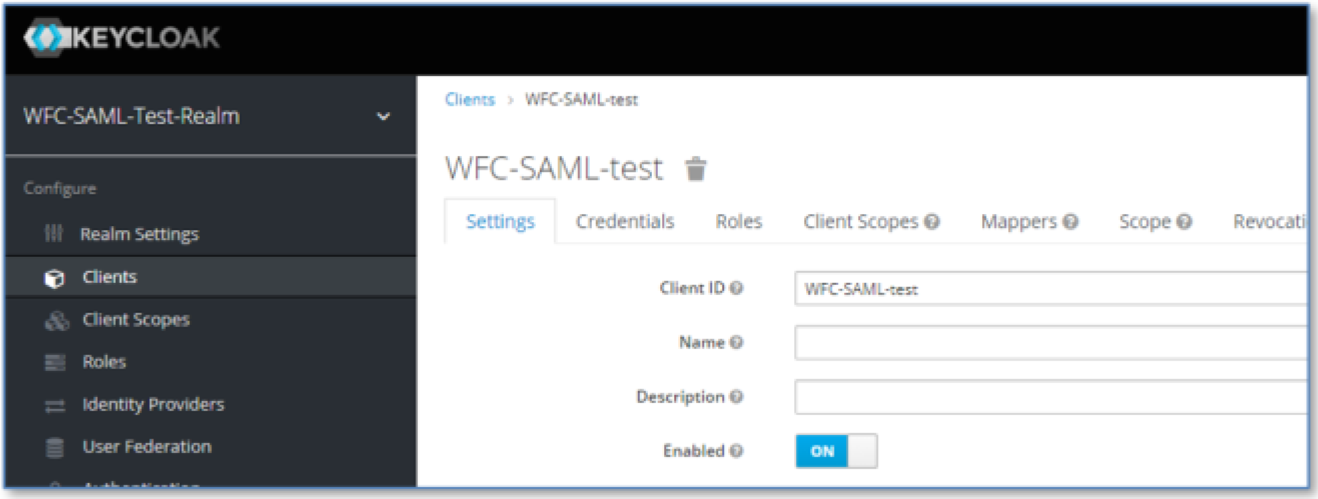


```
{"issuer":"https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm","authorization_endpoint":"https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-
SAML-Test-Realm/protocol/openid-connect/auth","token_endpoint":"https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/openid-
connect/token","token_introspection_endpoint":"https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/openid-
connect/token/introspect","userinfo_endpoint":"https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/openid-
connect/userinfo","end_session_endpoint":"https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/openid-
connect/logout","jwks_uri":"https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/openid-
connect/certs","check_session_iframe":"https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/openid-connect/login-status-
iframe.html","grant_types_supported":["authorization_code","implicit","refresh_token","password","client_credentials"],"response_types_supported":
["code","none","id_token","token","id_token token","code id_token","code token","code id_token token"],"subject_types_supported":
["public","pairwise"],"id_token_signing_alg_values_supported":
["PS384","ES384","RS384","HS256","HS512","ES256","RS256","HS384","ES512","PS256","PS512","RS512"],"id_token_encryption_alg_values_supported":["RSA-
OAEP","RSA1_5"],"id_token_encryption_enc_values_supported":["A128GCM","A128CBC-HS256"],"userinfo_signing_alg_values_supported":
["PS384","ES384","RS384","HS256","HS512","ES256","RS256","HS384","ES512","PS256","PS512","RS512","none"],"request_object_signing_alg_values_supported":
["PS384","ES384","RS384","ES256","RS256","ES512","PS256","PS512","RS512","none"],"response_modes_supported":
["query","fragment","form_post"],"registration_endpoint":"https://wfc-keycloak.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/clients-registrations/openid-
connect","token_endpoint_auth_methods_supported":
["private_key_jwt","client_secret_basic","client_secret_post","tls_client_auth","client_secret_jwt"],"token_endpoint_auth_signing_alg_values_supported":
["PS384","ES384","RS384","ES256","RS256","ES512","PS256","PS512","RS512"],"claims_supported":
["aud","sub","iss","auth_time","name","given_name","family_name","preferred_username","email","acr"],"claim_types_supported":
```

Client ID

The **Client ID** is the name of the clients in the configured realm. In this example the Client ID is `WFC-SAML-test`. The Client ID must be provided to the Profile Manager administrator and included in the PTT Pro JSON configuration file.

Figure 6 Client ID



Client ID: WFC-SAML-test

Name:

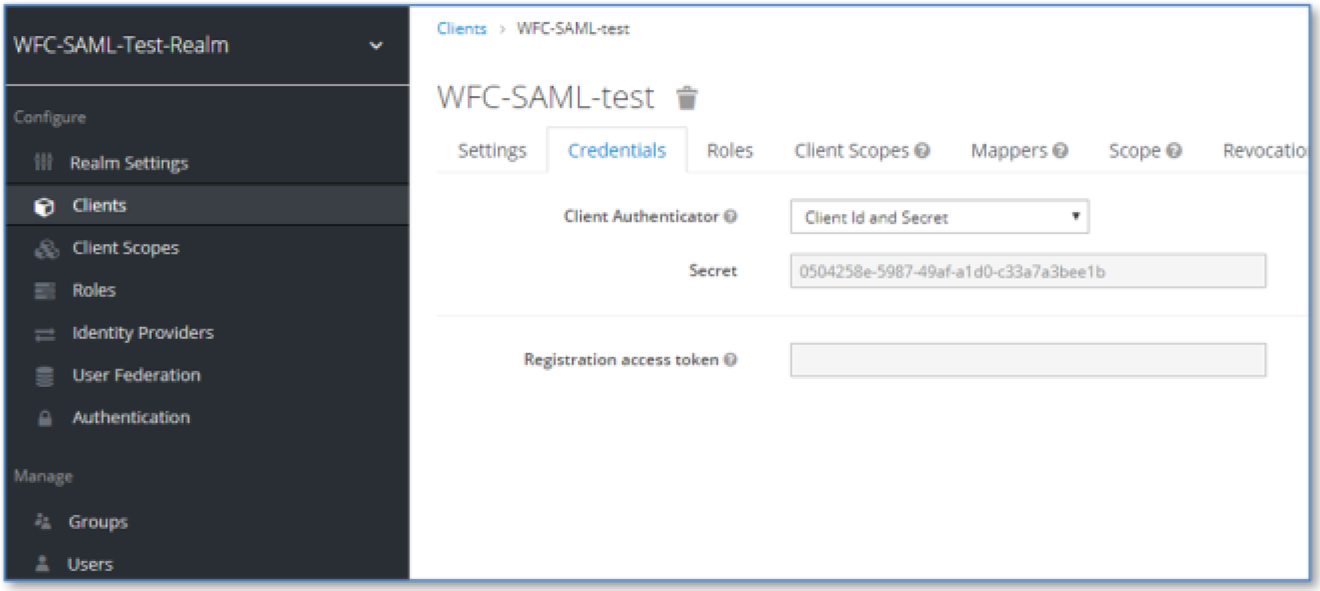
Description:

Enabled: ON

Client Secret

The client secret is also found in the Clients definition. Open the **Clients** definition of the realm and navigate to the **Credentials** tab to reveal the **Secret**. The secret must be provided to the Profile Manager administrator and included in the PTT Pro client JSON configuration file.

Figure 7 Client Secret



Client Authenticator: Client Id and Secret

Secret: 0504258e-5987-49af-a1d0-c33a7a3bee1b

Registration access token:

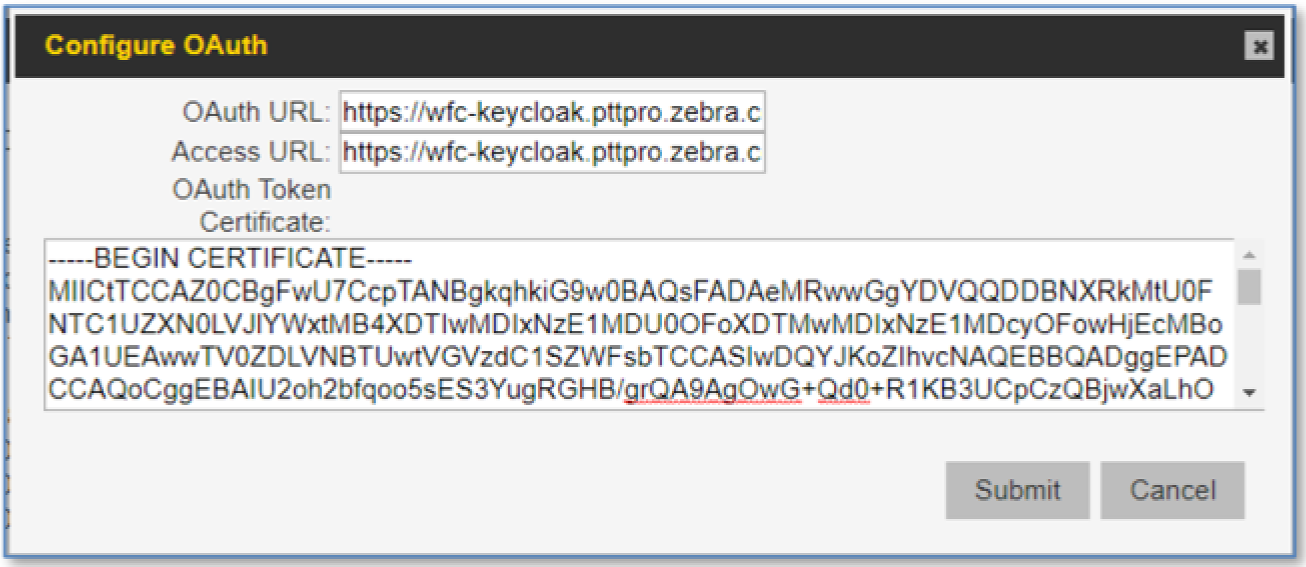
Configure the PTT Pro Server

The customer profile in the PTT Pro server supports an OAuth connection. Modify the configuration to use Keycloak for user authorization.

The configuration of the PTT Pro server requires:

- OAuth URL
- Access URL
- OAuth Certificate

Figure 8 PTT Pro OAuth Configuration



Configure OAuth

OAuth URL: https://wfc-keycloak.pttpro.zebra.c

Access URL: https://wfc-keycloak.pttpro.zebra.c

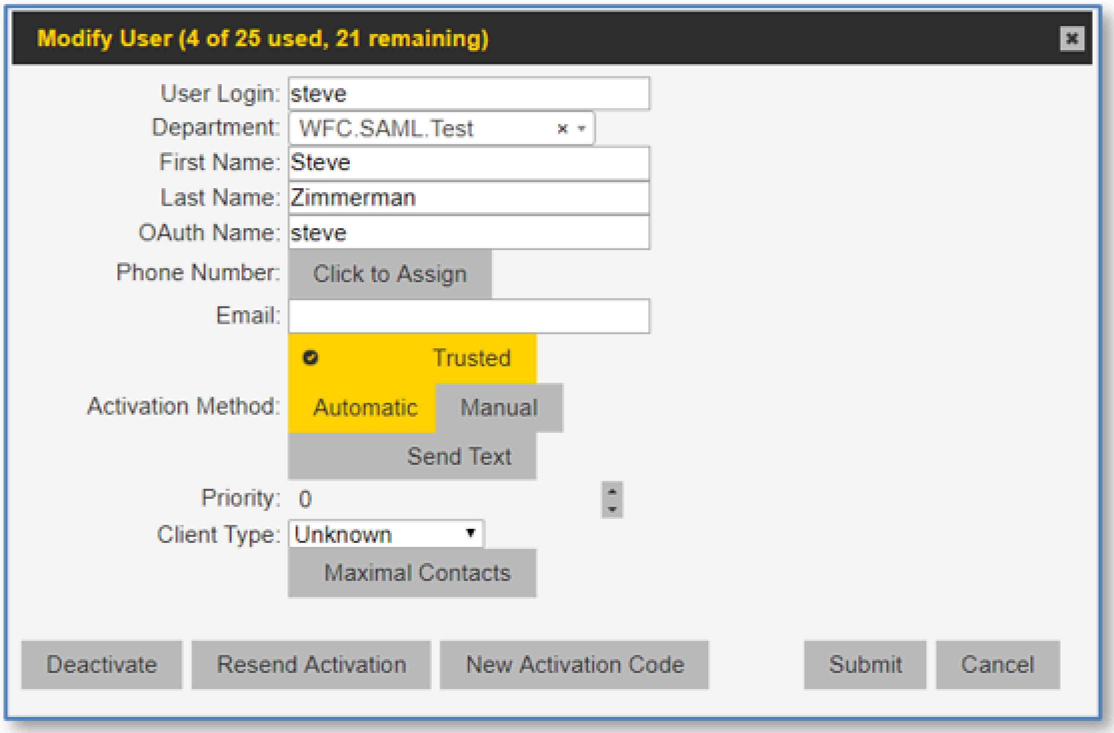
OAuth Token Certificate:

```
-----BEGIN CERTIFICATE-----
MIICtTCCAZ0CBgFwU7CcpTANBgkqhkiG9w0BAQsFADAeMRwwGgYDVQQDDBNXRkMtU0F
NTC1UZXN0LVJIYWxtMB4XDTIwMDIxNzE1MDU0OFoXDTMwMDIxNzE1MDcyOFowHjEcMBo
GA1UEAwwTV0ZDLVNBTUwtVGVzdC1SZWFsbTCCASIwDQYJKoZIhvcNAQEBBQADggEPAD
CCAQoCggEBAIU2oh2bfqoo5sES3YugRGHB/grQA9AgOwG+Qd0+R1KB3UCpCzQBjwXaLhO
```

If you are using a shared device model using OAuth:

- The device serial numbers must be provisioned in the server.
- The User definition requires that the Oauth Name field is populated correctly, as shown below. The Oauth name must match the username in the SAML server.

Figure 9 PTT Pro User with SAML username



Modify User (4 of 25 used, 21 remaining)

User Login: steve

Department: WFC.SAML.Test

First Name: Steve

Last Name: Zimmerman

OAuth Name: steve

Phone Number: Click to Assign

Email:

Activation Method: ● Trusted; Automatic (selected); Manual; Send Text

Priority: 0

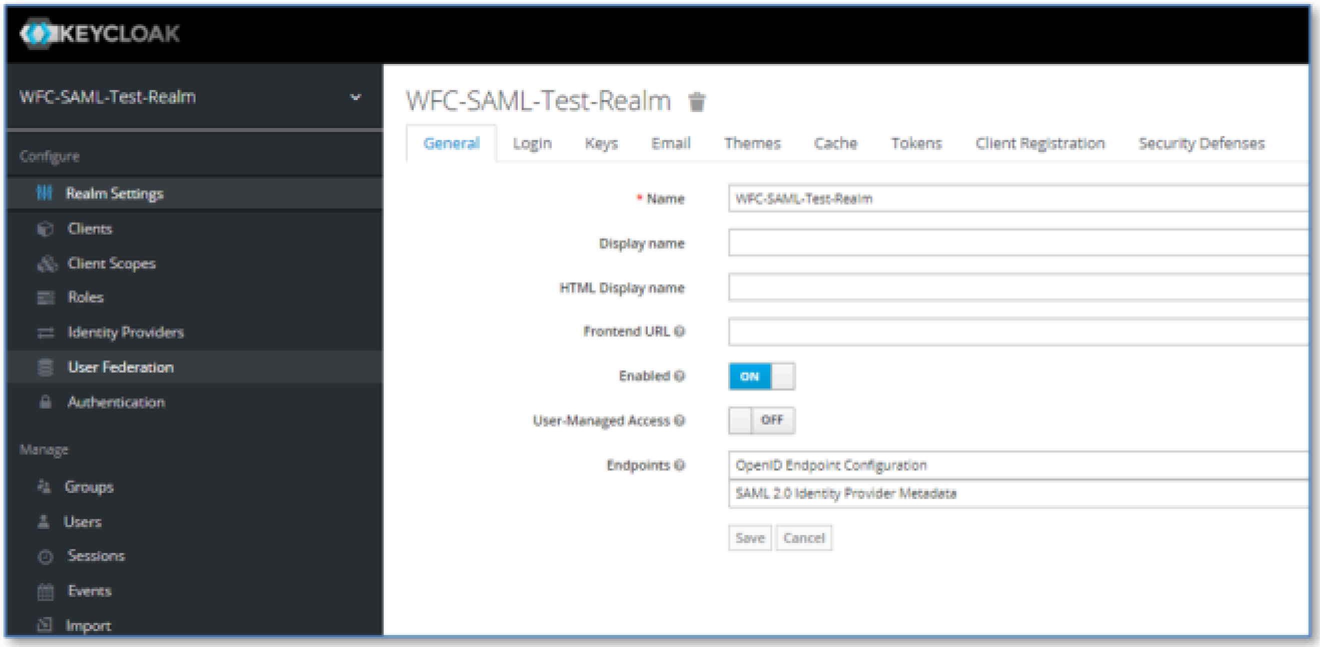
Client Type: Unknown

Obtain the OAuth and Access URLs

You can find the OAuth and Access URLs in the Keyloak server or from a URL.

To find the URLs through the KeyCloak user interface, navigate to **Realm Settings** and click on **OpenID Endpoint Configuration** under **Endpoints**.

Figure 10 OpenID Endpoint Configuration



WFC-SAML-Test-Realm

* Name: WFC-SAML-Test-Realm

Display name:

HTML Display name:

Frontend URL:

Enabled: ON

User-Managed Access: OFF

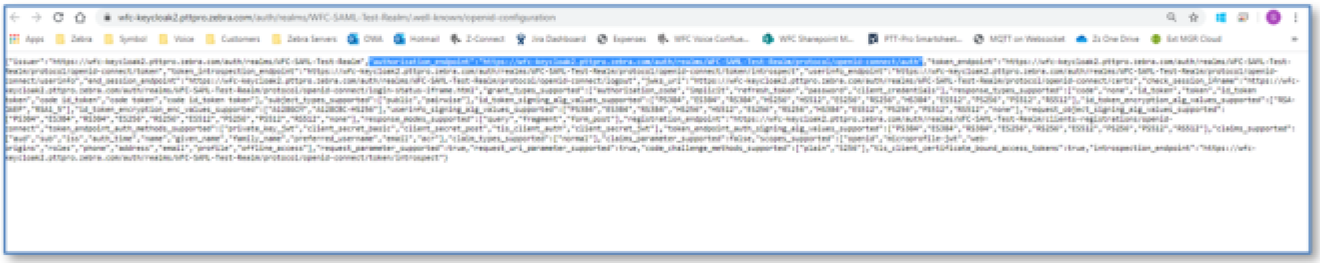
Endpoints: OpenID Endpoint Configuration
SAML 2.0 Identity Provider Metadata

To find the URLs through a web link, substitute `<WFC-SAML-Test-Realm>` with the name of the Realm in the following URL:

`https://wfc-keycloak.pttpro.zebra.com/auth/realms/<WFC-SAML-Test-Realm>/.well-known/openid-configuration`

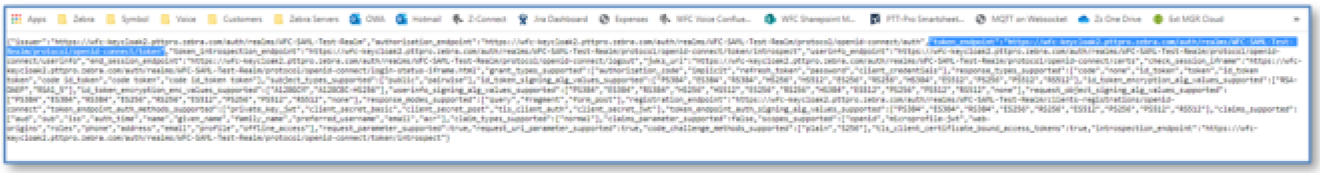
The output from the Keycloak user interface and the URL are shown below.

Figure 11 Authorization URL for PTT Pro OAuth URL Field



{"issuer":"https://wfc-keycloak2.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm","authorization_endpoint":"https://wfc-keycloak2.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/openid-connect/auth","token_endpoint":"https://wfc-keycloak2.pttpro.zebra.com/auth/realms/WFC-SAML-Test-Realm/protocol/openid-connect/token", [illegible]

Figure 12 Access URL for PTT Pro Access URL Field



Obtain the OAuth Certificate

See Exporting the WFC-ACS Certificate to PTT Pro on page 22 for the process of exporting the OAuth certificate to the PTT Pro server.

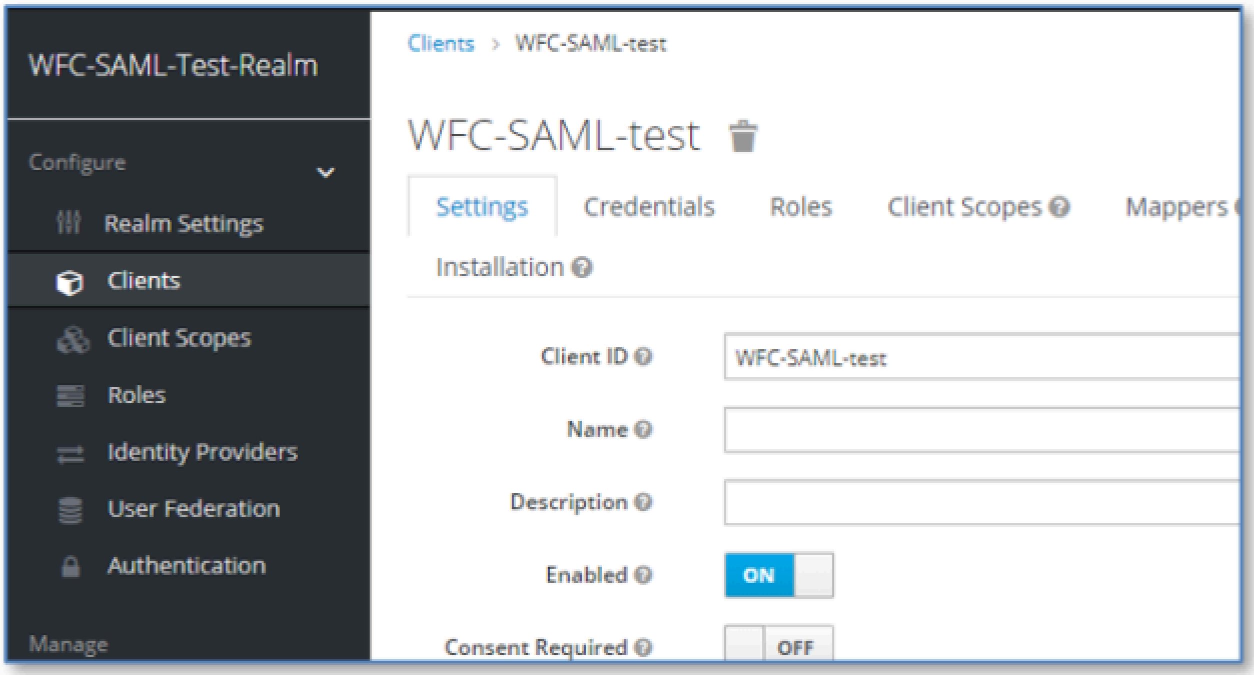
Configure the PTT Pro Client

The PTT Pro client is configured through the `WFCPTTProDefault.json` file. The file contains many elements, but the operation of the OAuth services to support the shared device model requires two parameters.

- oAuthClientID
- oAuthClientSecret

The `oauthClientID` field is obtained from the WFC-ACS service and is the value of the **Client ID** field in the **Settings** tab of the **Clients** view.

Figure 13 oAuthClientID Value



WFC-SAML-test

Client ID: WFC-SAML-test

Name:

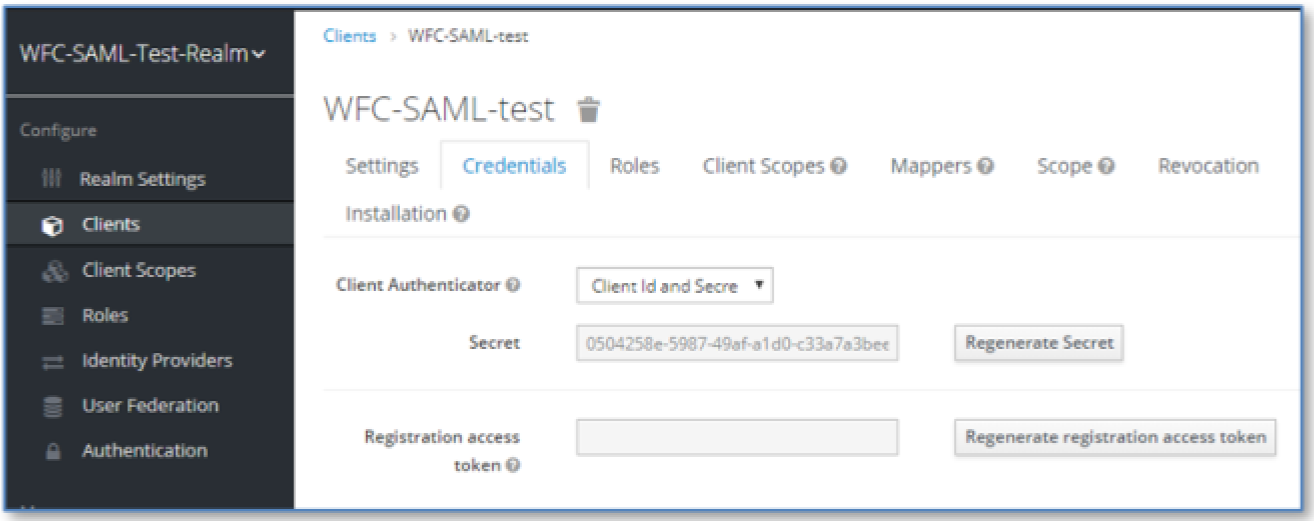
Description:

Enabled: ON

Consent Required: OFF

The oAuthClientSecret is also obtained in the **Clients** configuration. Navigate to the **Credentials** tab to reveal the secret.

Figure 14 oAuthClientSecret Value



Client Authenticator: Client Id and Secre

Secret: 0504258e-5987-49af-a1d0-c33a7a3bee

Registration access token:

Copy and paste this information in to the `WFCPTTProDefault.json` file for the PTT Pro client:

```
{
  "oAuthClientID":"WFC-SAML-test",
  "oAuthClientSecret":"0504258e-5987-49af-a1d0-c33a7a3bee1b"
```

Import the JSON file into the device and consume the configuration with an intent.

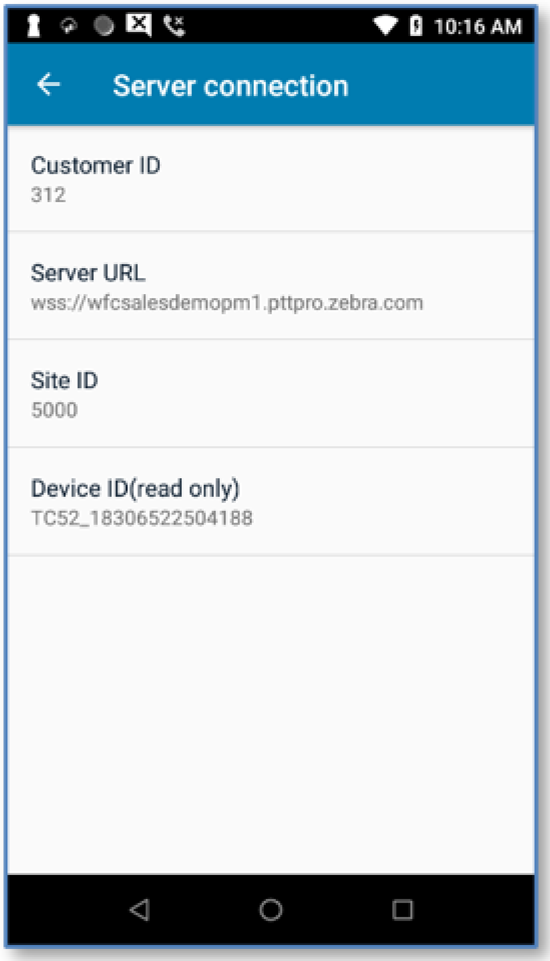
```
adb shell am broadcast -a com.symbol.wfc.pttpro.ACTION_DEFAULT_CONFIG --es
 "configpath" /sdcard/WFCPTTProDefault.json"
```

Configure the Profile Client

The configuration of the Profile Client requires two parameters.

- Customer ID
- Server URL

Figure 15 Profile Client Configuration



Customer ID
312

Server URL
wss://wfcsalesdemopm1.pttpro.zebra.com

Site ID
5000

Device ID(read only)
TC52_18306522504188

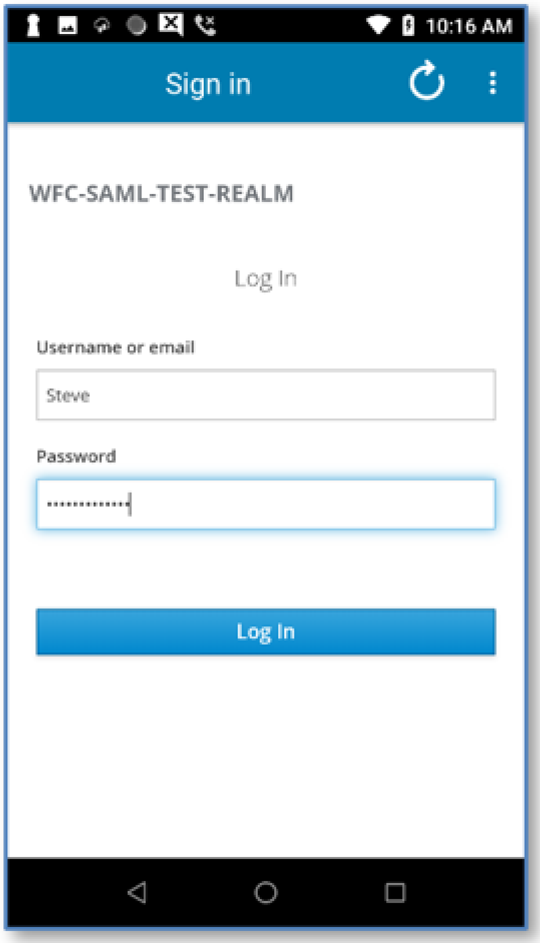
For the Customer ID, use the Tenant ID assigned by the Zebra Administrator in the Profile Manager.

The Server URL is the URL of the Profile Manager server configured to support OAuth.

Device Operation

After you configure the servers and the mobile devices, launch the Profile Client to log in.

Figure 16 Profile Client Log In Screen



10:16 AM

Sign in

WFC-SAML-TEST-REALM

Log In

Username or email

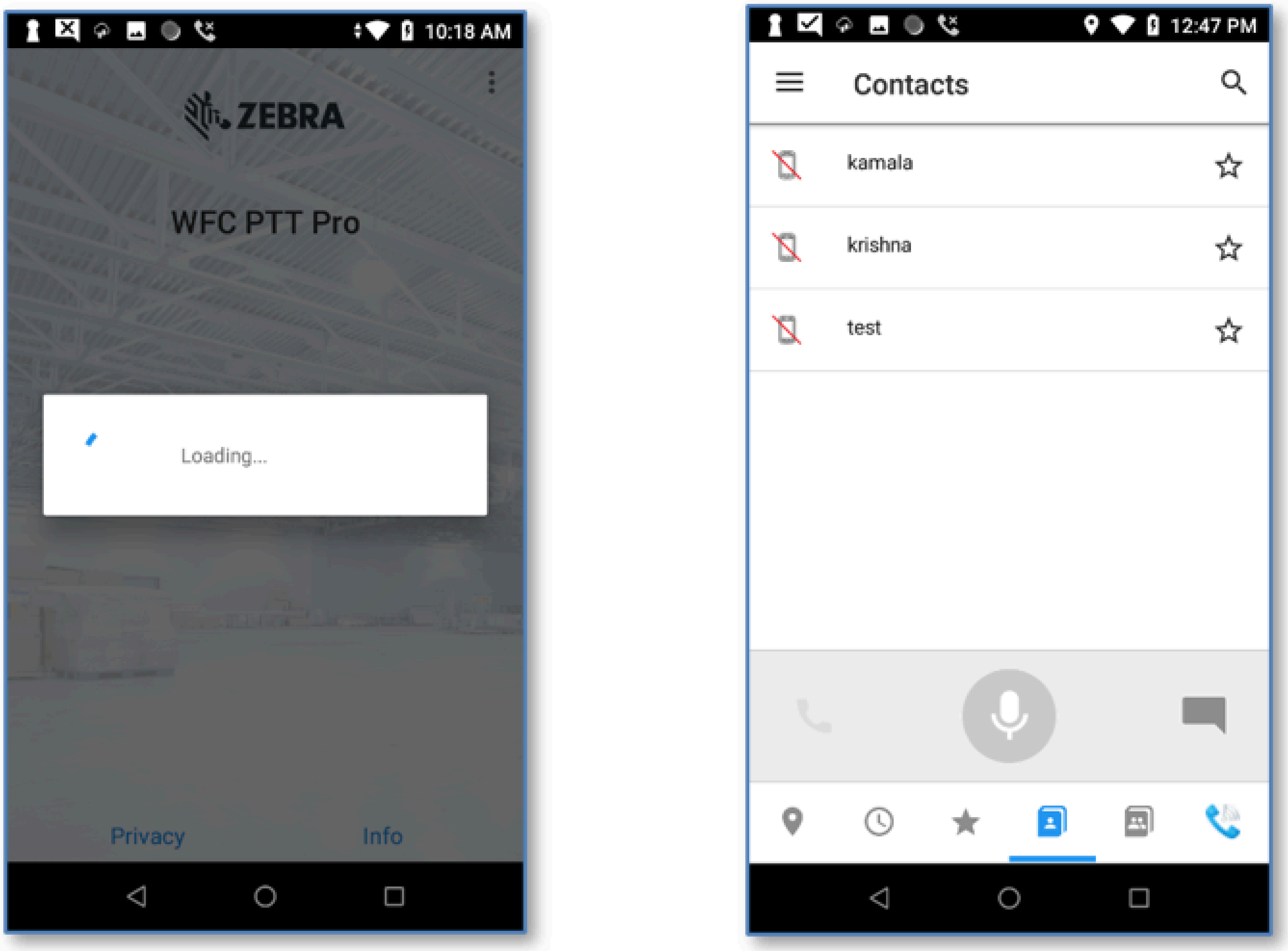
Steve

Password

••••••••••••••

The device connects to Profile Manager which redirects the client to authenticate to the SAML IdP through the WFC-ACS service. If the user is properly authenticated the `username` is returned through the network to the device and provisioned with the correct PTT Pro profile for the user.

Figure 17 WFC Client Provisioning



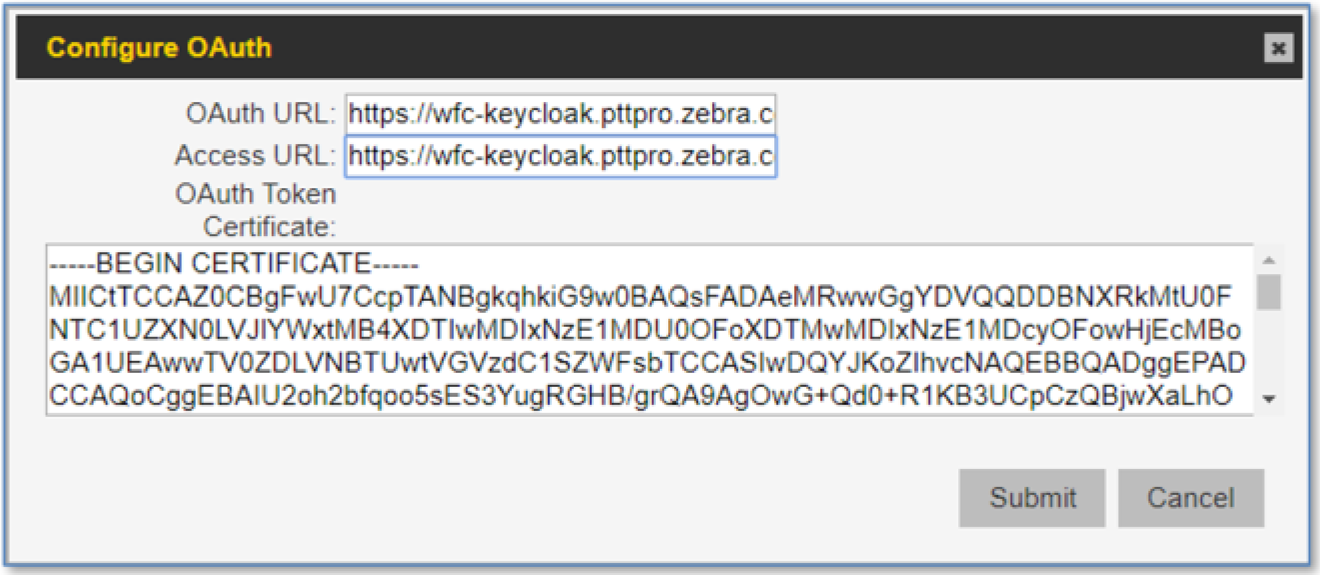
Your configuration is complete.

Troubleshooting the Client Error Message: ADFS Error

After completing the user login sequence, the mobile device stalls trying to connect to the PTT Pro server.

The certificate in the PTT Pro server OAuth configuration is malformed and needs to be correctly loaded.

Check the certificate configuration in the PTT Pro server. The format rendering of the imported certificate looks correct.



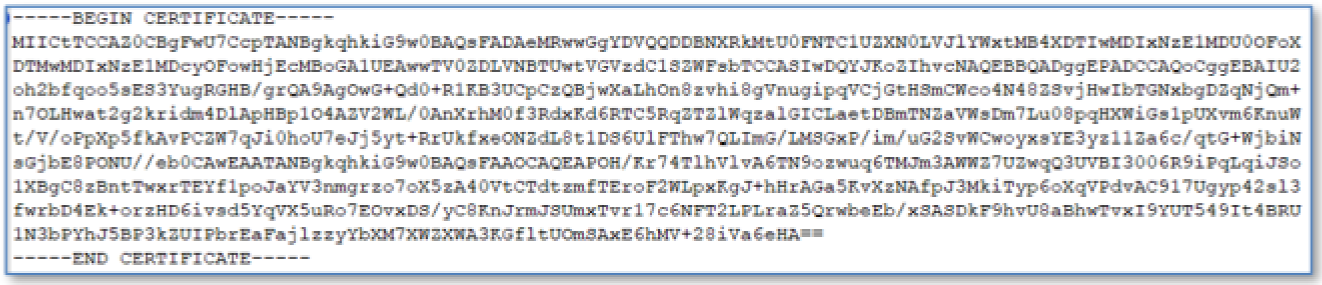
OAuth URL: https://wfc-keycloak.pttpro.zebra.c

Access URL: https://wfc-keycloak.pttpro.zebra.c

OAuth Token Certificate:

```
-----BEGIN CERTIFICATE-----
MIICtTCCAZ0CBgFwU7CcpTANBgkqhkiG9w0BAQsFADAeMRwwGgYDVQQDDBNXRkMtU0F
NTC1UZXN0LVJlYWxtMB4XDTIwMDIxNzE1MDU0OFoXDTMwMDIxNzE1MDcyOFowHjEcMBo
GA1UEAwwTV0ZDLVNBTUwtVGVzdC1SZWFsbTCCASIwDQYJKoZIhvcNAQEBBQADggEPAD
CCAQoCggEBAIU2oh2bfqoo5sES3YugRGHB/grQA9AgOwG+Qd0+R1KB3UCpCzQBjwXaLhO
```

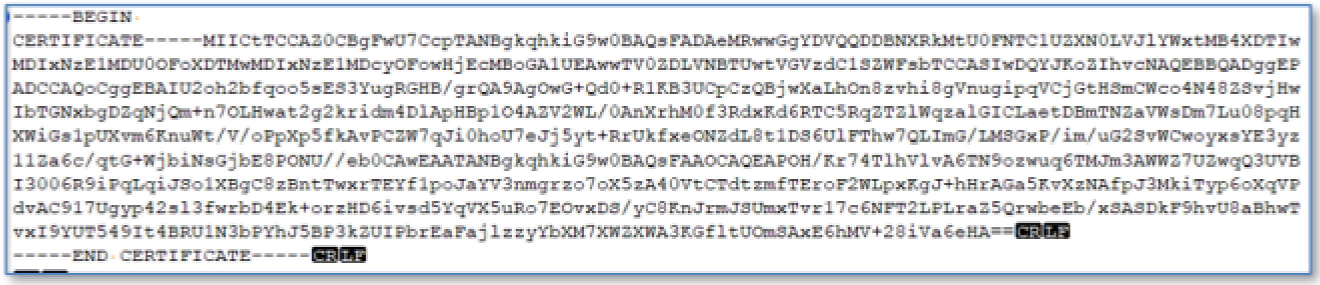
Copy the certificate to the clipboard.



```
-----BEGIN CERTIFICATE-----
MIICtTCCAZ0CBgFwU7CcpTANBgkqhkiG9w0BAQsFADAeMRwwGgYDVQQDDBNXRkMtU0FNTC1UZXN0LVJlYWxtMB4XDTIwMDIxNzE1MDU0OFoX
DTMwMDIxNzE1MDcyOFowHjEcMBoGA1UEAwwTV0ZDLVNBTUwtVGVzdC1SZWFsbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAIU2
oh2bfqoo5sES3YugRGHB/grQA9AgOwG+Qd0+R1KB3UCpCzQBjwXaLhOn8zvhi8gVnugipqVCjGtHSmCWco4N48Z8vjHwIbTGNxbgDZqNjQm+
n7OLHwat2g2kridm4DlApHBp1O4AZV2WL/0AnXrhM0f3RdxKd6RTC5RqZTZlWqzalGICLaetDBmTNZaVWsDm7Lu08pqHXWiGs1pUXvm6KnuW
t/V/oPpXp5fkAvPCZW7qJi0hoU7eJj5yt+RrUkfxeONZdL8t1DS6UlFThw7QLImG/LMSGxP/im/uG28vWCwoyxsYE3yz11Za6c/qtG+WjbiN
sGjbE8PONU//eb0CAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAPOH/Kr74TlhVlvA6TN9ozwuq6TMJm3AWWZ7UZwqQ3UVBI3006R9iPqLqiJSo
1XBgC8zBntTwxrTEYf1poJaYV3nmgrzo7oX5zA40VtCTdtzmfTEroF2WLpxKgJ+hHrAGa5KvXzNAfpJ3MkiTyp6oXqVPdvAC917Ugyp42sl3
fwrbD4Ek+orzHD6ivsd5YqVX5uRo7EOvxDS/yC8KnJrmJSUmxTvr17c6NFT2LPLraZ5QrwbeEb/xSASDkF9hvU8aBhwTvxI9YUT549It4BRU
1N3bPYhJ5BP3kZUIPbrEaFajlzzyYbXM7XWZXWA3KGfltUOmSAxE6hMV+28iVa6eHA==
-----END CERTIFICATE-----
```

The certificate still looks right.

View the certificate with Notepad++ with the view set to show all characters.



```
-----BEGIN·
CERTIFICATE-----MIICtTCCAZ0CBgFwU7CcpTANBgkqhkiG9w0BAQsFADAeMRwwGgYDVQQDDBNXRkMtU0FNTC1UZXN0LVJlYWxtMB4XDTIw
MDIxNzE1MDU0OFoXDTMwMDIxNzE1MDcyOFowHjEcMBoGA1UEAwwTV0ZDLVNBTUwtVGVzdC1SZWFsbTCCASIwDQYJKoZIhvcNAQEBBQADggEP
ADCCAQoCggEBAIU2oh2bfqoo5sES3YugRGHB/grQA9AgOwG+Qd0+R1KB3UCpCzQBjwXaLhOn8zvhi8gVnugipqVCjGtHSmCWco4N48Z8vjHw
IbTGNxbgDZqNjQm+n7OLHwat2g2kridm4DlApHBp1O4AZV2WL/0AnXrhM0f3RdxKd6RTC5RqZTZlWqzalGICLaetDBmTNZaVWsDm7Lu08pqH
XWiGs1pUXvm6KnuWt/V/oPpXp5fkAvPCZW7qJi0hoU7eJj5yt+RrUkfxeONZdL8t1DS6UlFThw7QLImG/LMSGxP/im/uG28vWCwoyxsYE3yz
11Za6c/qtG+WjbiNsGjbE8PONU//eb0CAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAPOH/Kr74TlhVlvA6TN9ozwuq6TMJm3AWWZ7UZwqQ3UVB
I3006R9iPqLqiJSo1XBgC8zBntTwxrTEYf1poJaYV3nmgrzo7oX5zA40VtCTdtzmfTEroF2WLpxKgJ+hHrAGa5KvXzNAfpJ3MkiTyp6oXqVP
dvAC917Ugyp42sl3fwrbD4Ek+orzHD6ivsd5YqVX5uRo7EOvxDS/yC8KnJrmJSUmxTvr17c6NFT2LPLraZ5QrwbeEb/xSASDkF9hvU8aBhwT
vxI9YUT549It4BRU1N3bPYhJ5BP3kZUIPbrEaFajlzzyYbXM7XWZXWA3KGfltUOmSAxE6hMV+28iVa6eHA==CRLF
-----END·CERTIFICATE-----CRLF
```

The PTT Pro server displayed the certificate so that it looked like there is a CR/LF after the `-----Begin Certificate-----` statement. Examining the certificate in Notepad++ revealed that CR/LF was missing.

NOTE: You can verify a certificate at the website https://jwt.io

Revision History

Changes to the guide are listed below:

Change	Date	Description
MN-004608-01 Rev A	09/2022	First version.