



Presto 2.5

Directory Provider Guide

Table of Contents

Overview	3
Introduction	4
Usage	5
Example	7
Groups	13

Overview

The Presto system allows administrators to plug in custom directory service providers to augment built-in Presto support for Microsoft Active Directory and Apple Open Directory.

Supported Platforms:

Windows: Vista 7, 8, 10; Server 2008 R2; Server 2012

Mac: OS X 10.11 +

Linux: Ubuntu 14.04 +

Introduction

Presto Directory Provider plugins run as scripts or programs that output their results to `stdout` as JSON formatted text.

Presto Directory Providers are managed by the Presto Agent component. As such, any customization should be applied to the `agent.conf` file located at:

Windows

```
C:\ProgramData\Collobos\Presto\agent.conf
```

Mac

```
/Library/Application Support/Collobos/Presto/agent.conf
```

This file is formatted as JSON text. It describes additional configuration that the agent will use to manage users and discover services on the network.

This document is about Directory Service Provider plugins, which are responsible for looking up users and groups, and authenticating Presto users.

Usage

To integrate an additional directory provider into the Presto system, follow these steps:

- **Open** the editor of your choice, and **paste** in the following example PowerShell script.

Figure 1: Example PowerShell Script

This script will check if the username is “johndoe” and the password is “x”. If so, it sends JSON text to `stdout` and returns an exit code of 0. Otherwise, nothing is sent to `stdout` and the script returns a non-zero exit code.

- **Save** this document to `C:\ProgramData\presto_directory_provider.ps`. The location of the script is not important, but we will use the location in the following step.
- **Edit** the `agent.conf` file and add a new custom directory provider section.

Figure 2: Example Declaration for Custom Directory Provider

After finishing the edits, the file should look like Figure 2.

- **Save** changes to the `agent.conf` file. Presto Agent automatically re-reads the file and adds the new directory provider to the Presto system.

That's it! When the Presto system tries to authenticate a user, the new directory provider will be invoked with the command line:

```
C:\ProgramData\presto_directory_provider.ps <name of user>
```

The password is read through `stdin`. It is possible for the password to be null.

Expected Output

The output of the authentication command is also JSON formatted text that describes a user to Presto. If the plugin is asked to authenticate the user `jsmith@example.com`, it might respond with the following JSON text if the authentication succeeded:

Figure 3: A Properly Formatted User

The resulting JSON text would contain the provider name, the name of the user and their display name, their GUID, and an array of strings. This array of strings is interpreted as the groups to which that the user belongs. Presto will turn this array into tags that can be used for writing rules.

Note: Presto will insert the string `"group:"` to each of the individual strings in the tags array.

Finally, the script would also return an exit code of 0 which tells Presto that the authentication succeeded.

If the authentication fails, it would print nothing to `stdout` and return a non-zero exit code.

Also note: The plugin can use whatever method it needs to determine if the authentication succeeded. However, it must return the result as JSON formatted text.

PIN Code Authentication

Presto supports PIN Code authentication, which allows users to login with a PIN code in addition to a username and password.

To setup a PIN code directory service provider, use the key `authenticate_user_with_pin` in the `agent.conf` file. The PIN code will be passed to the script via `stdin`, just like the password in `username/password` scripts.

IP Address Authentication

Presto supports IP address authentication. This is ideal for Single-Sign-On support in conjunction with `802.11x` wireless networks. A properly configured Directory Service Provider that implements IP address authentication will allow users to start and use the Presto app without ever having to login.

To setup an IP Address directory service provider, use the key `authenticate_user_with_address` in the `agent.conf` file. The IP Address will be passed to the script via `stdin`, just like the password in `username/password` scripts.

Example Scripts

The following section enumerates scripts written in a variety of scripting languages. Presto's plugin system is completely language agnostic. As long as Presto can invoke the program and the program returns data in the proper JSON format, system administrators are free to use the scripting language with which they feel most comfortable.

Because Presto's plugin system communicates using JSON formatted text, it is our belief that the `node.js` environment is best suited for writing scripts. However, as the following section shows, it is possible to write a script in many languages.

All of the scripts presented below will authenticate the user "johndoe" with the password "x".

bash

agent.conf
script.bash

node.js

agent.conf
script.js

agent.conf
script.bat

DOS

PowerShell

agent.conf
script.ps1