CAS Installer (for 2FA) Value-Added Module (VAM) Deployment Guide
Copyright Information

©2018. SecureAuth® is a registered trademark of SecureAuth Corporation. SecureAuth’s IdP software, appliances, and other products and solutions are copyrighted products of SecureAuth Corporation.

Version 1.0

Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>2018-04-13</td>
<td>Initial draft</td>
</tr>
<tr>
<td>1.0</td>
<td>2018-05-06</td>
<td>First draft completed</td>
</tr>
<tr>
<td>1.1</td>
<td>2018-08-08</td>
<td>Updated version of first draft</td>
</tr>
</tbody>
</table>
For information on support for this module, contact your SecureAuth support or sales representative:

Email: support@secureauth.com
inside-sales@secureauth.com

Phone: +1.949.777.6959 or +1-866- 859-1526

Website: https://www.secureauth.com/support
https://www.secureauth.com/contact
Table of Contents

Introduction ................................................................................................................................. 5
Benefits ........................................................................................................................................ 5
Overview ................................................................................................................................... 5
  CAS Server ............................................................................................................................... 6
  CAS Clients ............................................................................................................................... 6
Installation and Configuration ..................................................................................................... 8
  SecureAuth IdP Configuration ................................................................................................. 8
  IdP-Init Configuration ............................................................................................................. 12
  SP-Init Deployment ................................................................................................................ 14
Use Case .................................................................................................................................... 15
Introduction

SecureAuth Central Authentication Service (CAS) Installer provides robust two-factor authentication (2FA) for CAS clients. After successful primary authentication through the CAS Server, users must successfully complete a secondary authentication through SecureAuth IdP.

Benefits

The benefits of this installer are:

+ Enables seamless integration between CAS authentication and SecureAuth 2FA, greatly enhancing security at the CAS entry point
+ Increases the security of the CAS SSO procedure

Overview

The installer consists of two distinct parts as illustrated in Figure 1.

![Diagram of CAS Installer Architecture](image)

**FIGURE 1.** Overview of CAS Installer Architecture
CAS Server

The CAS server is Java-based application built on the Spring Framework to authenticate users and grant access to CAS-enabled services (CAS clients) by issuing and validating tickets. An SSO session is created when the server issues a ticket-granting ticket (TGT) to the user upon successful login. A service ticket (ST) is issued to a service at the user’s request via browser redirects using the TGT as a token. The ST is subsequently validated at the CAS server via back-channel communication.

CAS Clients

A CAS client is any CAS-enabled application that can communicate with the server via a supported protocol. A CAS client is also a software package that can be integrated with various software platforms and applications in order to communicate with the CAS server via some authentication protocol (such as CAS, SAML, or OAuth). CAS clients support several platforms:

+ Apache httpd Server (mod_auth_cas module)
+ Java (Java CAS Client)
+ .NET (.NET CAS Client)
+ PHP (phpCAS)
+ Perl (PerlCAS)
+ Python (pycas)
+ Ruby (rubycas-client)

The SecureAuth CAS service is actually a CAS overlay project with standard CAS functions and runs as a Windows Service.

To migrate from a preexisting CAS server to a secure CAS service, only one CAS login page (casLoginView.jsp) needs to be updated. Other CAS configurations should remain the same as preexisting CAS.
On the CAS client, only the login/validate URL needs to be switched to SecureAuth CAS Service URL; all other configurations should remain the same.

**FIGURE 2. CAS Client-Side**

The SecureAuth CAS installation package includes SecureAuth CAS Service Installer as shown in Figure 3.

**FIGURE 3. SecureAuth CAS Installer and Service**
Installation and Configuration

To install and configure the CAS installer:

1. Download InstallSaCasService.exe from the SecureAuth fileshare site. If no previous SecureAuthCASService is installed, skip steps 2 & 3.
2. Stop SecureAuthCasService and delete it with the `sc delete SecureAuthCasService` command line.
3. Back up the current version of the SecureAuthCasService folder to the D: drive and remove it.
4. Run InstallSaCasService.exe as an Administrator (see Figure 4) and check in the Windows Services to see if SecureAuthCasService is successfully added and running.

![Services](image)

**FIGURE 4. CAS Service Example**

SecureAuth IdP Configuration

The section details how to configure the SecureAuth portal and its required sub-realms. The following realm and sub-realms are defined in this discussion:

+ Portal Realm 18: Control Workflow SSO + Second Factor
+ Sub-realm 23: Link to Application1 protected by CAS client
+ Sub-realm 24: Link to Application2 protected by CAS client
To deploy this IdP-Init:

1. Create a new Portal Realm and sub-realms which link to each CAS client-protected application.
   An example of this is shown in Figure 5:

   ![Figure 5. CAS Client-Protected Application Example](image)

   **FIGURE 5. CAS Client-Protected Application Example**

   If the Portal realm has not yet been set up, refer to the detailed deployment document in this link:
   https://docs.gosecureauth.com/display/90docs/Secure+Portal+Configuration+Guide

   Make sure that the Data tab configuration is the same across all realms.
2. Link the portal realm to the sub-realms through the Post authentication page as shown in Figure 6.

![Post Authentication Configuration Page Example](image)

**FIGURE 6.** Post Authentication Configuration Page Example

3. Configure the sub-realms in this manner.
a. At the **Workflow** tab, do this:

![Workflow Page Configuration Example](image1)

**FIGURE 7.** Workflow Page Configuration Example

b. At the **Post Authentication** tab, do this:

![Post Authentication Configuration Example](image2)

**FIGURE 8.** Post Authentication Configuration Example

The Authorized/casurl.aspx is only accessible after the deployment of files. The next section discusses this file deployment.

**NOTE:** The Form Authentication and Machine Key values are the same as the Portal Realm (see Figure 9). Copy these values from the Portal Realm to the sub-realm.
**Figure 9.** Form Authentication and Machine Key Examples

**IdP-Init Configuration**

To configure the deployed IdP-Init, perform these steps:

1. Locate the ASP files in `D:\SecureAuthCasService\sa_asp_code\CAS Portal IDP & SP.zip`, add `casurl.aspx`, and update settings in the `web.config` in each sub-realm. For example, `D:/`
For example, SecureAuth/SecureAuth23/Authorized/casurl.aspx.

```xml
<appSettings>
    <add key="casLoginURL" value="http://vm-oc1-cd0504.sacustom.local:8080/cas/login" />
    <add key="applicationURL" value="http://vm-oc1-cd0505.sacustom.local:8080/examples1/" />
</appSettings>
```

2. Add the `saGenericHandler.ashx` web service to the sub-realm root. For example, D:/SecureAuth/SecureAuth23.

3. Add `customHeader` in `web.config` to each sub-realm in this manner:

```xml
<customHeaders>
    <add name="Access-Control-Allow-Origin" value="*" />
</customHeaders>
```

4. Update the required jsp or html file, as required by the CAS version.
   - For older versions of CAS, update `casLoginView.jsp` to call one sub-realm web service and update the portal realm SSO URL.
     For example:
     ```jsp
     D:\SecureAuthCasService\webapps\cas\WEB-INF\view\jsp\default\ui\casLoginView.jsp.
     <script language="javascript">
     function doAutoLogin() {
         ....
         $.ajax{
             //Redirect to Portal realm for SSO
         }
     }
     ``
     - For newer versions of CAS, update `loginform.html` to call one sub-realm web service and update the portal realm SSO URL.
     For example:
     ```jsp
     D:\SecureAuthCasService\webapps\cas\WEB-INF\classes\templates\fragments\loginform.html.
     <script language="javascript"> function doAutoLogin() {
         ....
         $.ajax{
             //Redirect to Portal realm for SSO
             if (get('auto') == 'true') {
                 doAutoLogin();
             } else {
             }
         }
     ```
### SP–Init Deployment

To deploy SP-Init, follow these steps:

1. Replace `secureauth.aspx` and `securePortal.aspx`, then update settings in `web.config` in the Portal realm. For example,
   - D:/SecureAuth/SecureAuth18/SecueAuth.aspx.vb
   - D:/SecureAuth/SecureAuth18/SecurePortal.aspx.vb
   
   The web.config content must include the following code:

   ```xml
   <appSettings>
   <add key="casLoginURL" value="http://vm-oc1-cd0504.sacustom.local:8080/cas/login" />
   ```

2. Update the jsp or html file, depending on your CAS version.
   - For older versions of CAS, update `casLoginView.jsp` with SP-Init added and update the web service call and portal realm SSO URL.
     For example: D:\SecureAuthCasService\webapps\cas\WEB-INF\view\jsp\default\ui\casLoginView.jsp.

     ```javascript
     $.ajax({
     ```

   - For newer versions of CAS, update `loginform.html` to call one sub-realm web service and update the portalrealm SSO URL.
     For example:

     ```javascript
     D:\SecureAuthCasService\webapps\cas\WEB-INF\classes\templates\fragments\loginform.html.
     function doAutoLogin() {
     ....
     $.ajax{
       type: 'POST',
       data: jsonData,
     }```
Use Case

One possible use case is shown in this example where a SecureAuth login begins with a username/login and a second factor:

1. Bring up the CAS server URL. A screen like Figure 10 appears:

   ![CAS Auto Login Example](image)

   **FIGURE 10. CAS Auto Login Example**

2. Enter the specified username and password, then click **Submit** as shown in Figure 11.

   ![CAS Delivery Options Example](image)

   **FIGURE 11. CAS Delivery Options Example**

3. Select the delivery method and click **Submit**.
4. Enter the registration number.
5. Click the required application link as shown in Figure 12.

![Application Link Example](image)

**FIGURE 12.** Application Link Example

Only the IDP-init process lists application links. Click the application link to allow the qualified user to access CAS-protected applications; the SP-init process skips this step and allows the user to access CAS-protected applications directly.