Manual

Servei autenticació SSO-CAS de la UPC

Juny 2019

Índex de contingut

Introducció

L'objectiu d'aquest document és descriure l'autenticació amb el protocol CAS per mitjà del servei SSO de la UPC (SSO-UPC d'ara en endavant en aquest document), utilitzant els dos sistemes disponibles, mitjançant usuari i contrasenya o bé mitjançant el certificat digital.

CAS Single Sign On

1. Informació genèrica

El servei del SSO-CAS permet unificar l'autenticació dels usuaris, de manera que totes les aplicacions que es troben "CASsificades" (utilitzen el protocol CAS per al login), només sol·liciten les credencials un sol cop. A posteriori el SSO-CAS i el navegador de l'usuari (que emmagatzema la cookie del CAS) gestionen la sessió validada entre les aplicacions.

Per tal de CASsificar un aplicació cal implementar la part del client del CAS. Es poden trobar exemples en la documentació oficial:

https://apereo.github.io/cas/4.2.x/integration/CAS-Clients.html

Els paràmetres del servidor CAS de la UPC per a la configuració del client són:

CAS_Hostname: sso.upc.edu
CAS_Port: 443
CAS_URL_Login: https://sso.upc.edu/CAS/login?service=
CAS_URL_Logout: https://sso.upc.edu/CAS/logout?url=

En la url de login del SSO-CAS, en el paràmetre "service" l'aplicació client haurà de proporcionar la seva pròpia url d'accés. De la mateixa manera, en el logout cal proporcionar un link de retorn un cop l'usuari s'hagi desconnectat.

El servidor de SSO-CAS pot proporcionar a les aplicacions el username de l'usuari que ha accedit, així com el mètode que ha usat per accedir (nom d'usuari i contrasenya o bé certificat digital). La majoria d'aplicacions en tenen prou amb saber el username de l'usuari. Tot i així, hi pot haver aplicacions que requereixin l'accés amb certificat de cara a oferir funcionalitats més crítiques o segures, aquestes aplicacions hauran de comprovar que el mètode que l'usuari ha usat per fer login ha estat el certificat digital.

En funció dels paràmetres que la vostra aplicació vulgui recuperar, el client de CAS pot usar un protocol o un altre. A continuació es descriuen aquestes opcions.

2. Protocol bàsic

A continuació es descriu com les aplicacions poden comunicar-se amb el SSO-CAS per obtenir el username de l'usuari que ha fet login, tant si aquest ho ha fet amb usuari i contrasenya com si ho ha fet amb certificat digital.

2.1 Resposta CAS

En aquesta opció d'autenticació CAS, la resposta que retorna segueix l'especificació del protocol del CAS versió 2 i té l'estructura següent:

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
        <cas:authenticationSuccess>
                <cas:user>common_name_usuari</cas:user>
        </cas:authenticationSuccess>
</cas:serviceResponse>
```

A partir d'aquí es pot consultar el username de l'usuari fàcilment. Per fer aquesta consulta, el CAS proporciona varis clients segons el llenguatge utilitzat. A continuació es poden trobar els exemples més comuns.

2.2 Filtre CAS Java

En el CAS de Java és important la versió del client que s'utilitza. Per a l'autenticació bàsica és suficient utilitzar els mòduls cas-client 2.x, en el qual cal incloure, a més dels paràmetres especificats a la secció 1, la validació amb la url:

CAS_URL_Validate: https://sso.upc.edu/CAS/serviceValidate

Després cal configurar un filtre Java en la vostra aplicació que validi contra el SSO-CAS. El client de CAS proporciona les llibreries i funcions necessàries per fer aquesta validació i consultar el nom de l'usuari autenticat. Un exemple:

```
import edu.yale.its.tp.cas.client.ServiceTicketValidator;

ServiceTicketValidator sv = new ServiceTicketValidator();

... inicialitzar variables del sv ...

sv.validate();

sv.getUser();
```

2.3 Filtre CAS php

En la versió php del client CAS, a diferència del de Java, la versió del client inclou el suport per els diferents tipus de validació. Per configurar-ho cal incloure la pàgina CAS.php a la pàgina inicial de l'aplicació. Posteriorment després cal configurar la versió del client utilitzat per a la validació de la següent manera:

```
phpCAS::client(CAS_VERSION_2_0,'sso.upc.edu',443,'/CAS');
```

De la mateixa manera que en Java, la llibreria del client CAS proporciona les funcions necessàries per a consultar l'usuari després de validar contra el CAS.

```
phpCAS::getUser();
```

3. Protocol ampliat

A continuació es descriu com les aplicacions poden comunicar-se amb el CAS per obtenir a més del username de l'usuari que ha fet login, el mètode que aquest ha usat per accedir (nom d'usuari i contrasenya o bé certificat digital). Per fer-ho, cal que usin el protocol SAML1.1.

3.1 Resposta CAS utilitzant SAML1.1

No es propòsit d'aquest document descriure aquest protocol. Es pot trobar la informació sobre la resposta que dóna el cas en aquesta URL:

https://apereo.github.io/cas/4.0.x/protocol/SAML-Protocol.html#saml-11

SAML1.1 encapsula les dades en una resposta xml que conté l'objecte "Principal", que conté l'usuari que s'ha autenticat al CAS. A més proporciona d'altres atributs que s'hagin afegit en el servidor CAS.

La part necessària per diferenciar el mètode d'autenticació el proporciona per defecte el protocol SAML1.1. Es troba en la part de la resposta:

```
<AuthenticationStatement AuthenticationInstant=""
       AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
```

Els dos tipus de mètodes que es poden rebre són:

```
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password"
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:X509-PKI"
```

3.2 Filtre CAS Java

En el CAS de Java, cal actualitzar la versió del client a la CAS 3.0 o posteriors. Aquests han de implementar la validació via SAML1.1.

Es pot trobar la documentació aquí:

https://wiki.jasig.org/display/CASC/CAS+Client+for+Java+3.1

En lloc del filtre antic del CAS, cal configurar en el fitxer web.xml de l'aplicació el filtre:

```
org.jasig.cas.client.validation.Saml11TicketValidationFilter
```

Seguint la documentació de l'enllaç anterior hi ha suficient per a que l'aplicació estigui CASsificada. Si és necessari aconseguir la informació de la resposta SAML, cal definir un Filtre Java per a tota l'aplicació per a parsejar la resposta SAML1.1.

El mètode d'autenticació és un atribut propi de l'Assertion que retorna la resposta SAML1.1. Es pot recuperar a partir de la sessió tal que:

```
Assertion assertion = (Assertion) request.getSession().getAttribute("_const_cas_assertion_");

String userName = assertion.getPrincipal().getName();

Map<String, Object> atributsAssertion = assertion.getAttributes();
```

On atributsAssertion conté el mètode d'autenticació, per exemple:

```
[samlAuthenticationStatement::authMethod=urn:oasis:names:tc:SAML:1.0:am:password]
```

3.3 Fitre CAS php

En php, cal incloure la pàgina CAS.php a la pàgina inicial de l'aplicació. Posteriorment cal configurar la versió del client utilitzat per a la validació de la següent manera:

```
phpCAS::client(SAML_VERSION_1_1,'sso.upc.edu',443,'/CAS');
```

A partir d'aquí, el username s´aconsegueix de manera idèntica a versions anteriors. La llibreria del client CAS proporciona les funcions necessàries per a consultar l'usuari després de validar contra el CAS.

```
phpCAS::getUser();
```

I per obtenir el mètode de validació cal parsejar la resposta SAML1.1.

4. Validació única amb certificat

Si es necessita forçar l'autenticació en l'aplicació mitjantçant certificat digital caldrà demanar a l'ATIC que es configuri al SSO-CAS.

L'antic servei del CAS de la UPC (cas.upc.edu) tenia una URL de login per forçar la validació única i exclusivament mitjançant certificat digital que ja no existeix al CAS-UPC:

CAS_URL_Login: https://cas.upc.edu/login?service

La resta de paràmetres d'accés al CAS es mantenen igual.

Cal que l'aplicació utilitzi el protocol SAML1.1 per obtenir la informació sobre el mètode d'autenticació de l'usuari, tal com s'ha descrit a la secció "Protocol ampliat".

Tot i així, cal contemplar que els següents tres escenaris possibles.

4.1 Usuari autenticat prèviament al CAS amb usuari i contrasenya

Quan l'usuari ha estat autenticat al CAS prèviament i amb usuari i contrasenya, tot i que configureu a la vostra aplicació l'adreça loginCert, el CAS farà sign on i no verificarà el mètode utilitzat per autenticar-se. La vostra aplicació ha de tenir desenvolupat un filtre que ha de consultar el mètode d'autenticació. Si és usuari i contrasenya, és l'aplicació que ha de mostrar una pàgina d'error o informativa. L'única solució per habilitar l'accés és forçar el logout del CAS i redirigir de nou a l'autenticació amb certificat.

4.2 Usuari autenticat prèviament al CAS amb certificat

De la mateixa manera que abans, l'usuari ja està validat, per tant no es mostra el login del CAS, el filtre de l'aplicació ha de consultar el mètode d'autenticació. En aquest cas, ha d'autoritzar l'accés a l'aplicació ja que l'usuari s'ha validat prèviament amb certificat digital.

4.3 Usuari no autenticat al CAS

L'usuari no està autenticat, el SSO-CAS presenta la plana de login donant només l'opció d'usar el certificat digital. Si l'usuari no té el certificat inserit al lector el SSO-CAS mostra la pàgina d'error i dóna l'opció de tornar a autenticar-se. Si el té inserit, el CAS demana automàticament el PIN i redirecciona cap a l'aplicació un cop autenticat amb èxit.