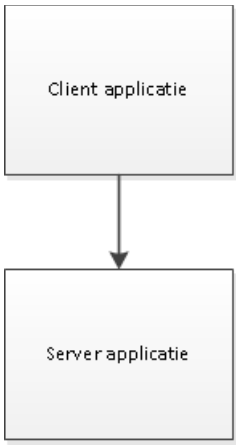
Auteur: André van den Nouweland
Datum: 17 oktober 2017
Betreft: SAML voor authenticatie/autorisatie

Doel:
Dit document laat voorbeelden zien hoe je authenticatie/autorisatie mee kan geven via een SAML assertion (token). Een SAML is te implementeren bij koppelvlakken zoals XML Soap webservices (standaard via Soap Header) of bij Web API services (via HTTP header). De bron SAML kan in beide gelijk zijn. Enige 2 verschillen zijn dat een SAML in de HTTP header gecodeerd moet worden meegegeven (base64 encoding) om ervoor te zorgen dat de SAML niet wijzigt en zo geldig blijft. Tweede verschil is dat een Web server HTTP headers beperkt tot een maximum aantal karakters (8KB). Bij een grotere SAML zou deze opgesplitst moeten worden in meerdere HTTP header attributen.

Scope:
SAML gebruik via authenticatie/autorisatie attributen is in scope. Hoe een SAML is opgebouwd, op welk deel je signing toepast en of je encryptie toepast is buiten scope. Bij gebruik van een IDP en SP laten we de onderlinge SAML uitwisseling achterwege. We tonen de uiteindelijke SAML die aankomt op de SP. We laten de routering verder ook buiten scope. Dit document spitst zich toe op gebruikers en rollen. Gebruik voor overige informatie de beschikbare literatuur.

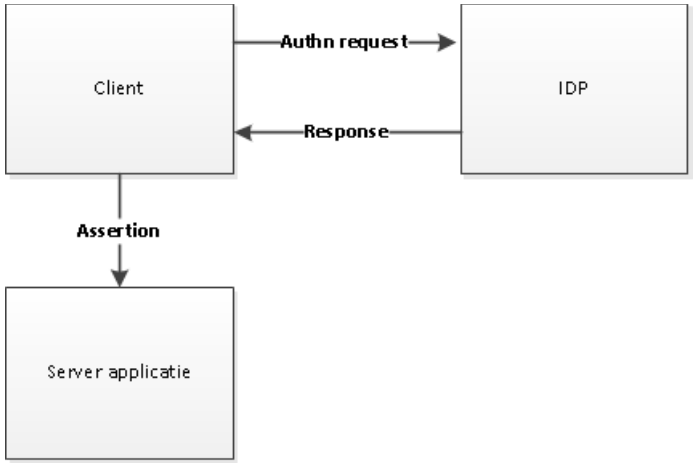
Index

Scenario 1: SAML bij traditionele client/server applicatie



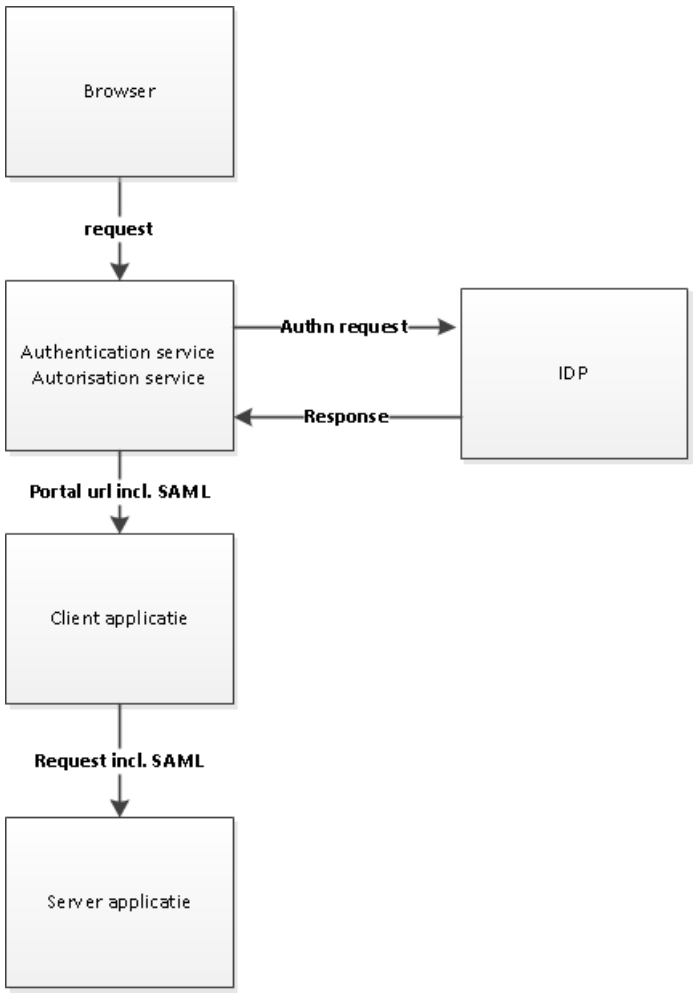
De cliënt applicatie zal hier zelf de SAML assertion moeten opbouwen, de juiste attributen vullen en de SAML doorgeven waarbij de server applicatie de SAML kan valideren alvorens de gegevens te gebruiken in de applicatie.

Scenario 2: SAML bij client/server applicatie met IDP



Hier vraagt de client zelf een SAML response op bij de Identity Provider. De SAML response inclusief authentication kan de client doorzetten naar de server applicatie. Als client kan ook de browser worden ingevuld. Denk aan DigiD. Als de browser de server applicatie wilt bereiken zal er eerst een redirect naar DigiD IDP plaatsvinden om in te loggen. De SAML response wordt weer redirected naar de server applicatie.

Scenario 3: SAML mbv Authentication/Autorisation service



Hier zorgt een Authentication/Autorisation service ervoor dat de client applicatie een SAML krijgt met de benodigde gegevens. Het kan zelf de SAML valideren en de gegevens gebruiken maar de SAML niet manipuleren. De client kan de SAML door passen naar de server applicatie om zo een identity trust in de hele keten te houden. Doorpassen moet ook als het koppelvlak SAML heeft gespecificeerd voor het doorgeven van gebruiker en rollen. De SAML naar de client applicatie hoeft niet gelijk te zijn aan de IDP response. De authentication/autorisation service kan de IDP SAML response niet manipuleren maar kan ervoor kiezen een nieuwe SAML op te bouwen en naast de IDP attributen ook extra authenticatie/autorisatie attributen toe te voegen. Belangrijk hierbij is dat deze service net als de IDP gezien wordt als een trusted component.

Voorbeeld 1 : SAML response

Zoals gezegd is de opbouw van de SAML gelijk. Een basis Response SAML van een IDP ziet er als volgt uit:

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="_8e8dc5f69a98cc4c1ff3427e5ce34606fd672f91e6"
Version="2.0" IssueInstant="2014-07-17T01:01:48Z"
Destination="http://sp.example.com/demo1/index.php?acs"
InResponseTo="ONELOGIN_4fee3b046395c4e751011e97f8900b5273d56685">
        <saml:Issuer>...
        <samlp:Status>...
                <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
        </samlp:Status>

        <saml:Assertion

        - - -

        </saml:Assertion>

</samlp:Response>
```

Het hart is de saml:Assertion. Deze xml tag bevat meestal een ID om aan te geven dat de gehele assertion is gesigned. Deze zal dan ook in zijn geheel en ongewijzigd moeten worden doorgegeven.

Voorbeeld 2 : SAML Assertion

Binnen de Response envelope, Soap header of direct als HTTP header is de SAML Assertion opgenomen:

```
 <saml:Assertion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema" ID="pfx945bbf6a-0e39-fbf7-1d37-cf5d2c261f6e"
Version="2.0" IssueInstant="2014-07-17T01:01:48Z">
 <saml:Issuer>http://idp.example.com/metadata.php</saml:Issuer>
 <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
 <ds:SignedInfo>
 - - -
 </ds:SignedInfo>
 <ds:SignatureValue>cfMu/eoe2Y=</ds:SignatureValue>
 <ds:KeyInfo>
 - - -
 </ds:KeyInfo>
 </ds:Signature>
 <saml:Subject>
 <saml:NameID ...>_ce3 2d7</saml:NameID>
 - - -
 </saml:Subject>
 <saml:Conditions NotBefore="2014-07-17T01:01:18Z" NotOnOrAfter="2024-01-18T06:21:48Z">
 <saml:AudienceRestriction>
 - - -
 </saml:AudienceRestriction>
 </saml:Conditions>
 <saml:AuthnStatement . . . >
 <saml:AuthnContext>
 <saml:AuthnContextClassRef> - - - </saml:AuthnContextClassRef>
 </saml:AuthnContext>
 </saml:AuthnStatement>
 <saml:AttributeStatement>
 <saml:Attribute
 Name="uid" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
 <saml:AttributeValue xsi:type="xs:string">test</saml:AttributeValue>
 </saml:Attribute>
 </saml:AttributeStatement>
 </saml:Assertion>
```

Voorbeeld 3: eHerkenning sleutels als SAML attributen

Voor bijvoorbeeld eHerkenning kunnen de sleutels als attributen binnen de SAML assertion worden meegegeven aan de applicatie:

```
<saml:AttributeStatement xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
        <saml:Attribute
                        Name="urn:nl:eherkenning:1.7:EntityConcernedID:Vestigingsnr"
                        xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
                <saml:AttributeValue>000000012346789890l2</saml:AttributeValue>
        </saml:Attribute>
        <saml:Attribute Name="urn:nl:eherkenning:1.7:EntityConcernedID:KvKnr"
                        xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
                <saml:AttributeValue>00000001234567890000</saml:AttributeValue>
        </saml:Attribute>
</saml:AttributeStatement>
```

De SAML specs bepalen de syntax van de globale attribuut opbouw, niet de overige afspraken zoals attribuutnaam. Bij eHerkenning zijn deze vastgesteld in het Afsprakenstelsel Elektronische Toegangsdiensten

Voorbeeld 4: DigiD

DigiD maakt gebruik van het Subject en Authentication Statement blok binnen de SAML. Vanwege privacy is het subject beveiligd. Via de AuthnStatement kan je achterhalen welk inlog methode is gebruikt door de user. Uiteindelijk krijgt de client alleen de BSN vanuit DigiD.

Voorbeeld 5: Additionele attributen voor autorisatie

Als je autorisatie wilt toepassen en je maakt gebruik van een Authentication/Autorisatie service die zelf de SAML maakt, kan je additionele attributen doorgeven. Een goed idee is om security groepen in bijvoorbeeld een Active Directory te gebruiken voor autorisatie. De Autorisatie service kan dan via LDAP de security groepen ophalen voor betreffende user en deze als attributen meegeven:

```
<saml:Attribute Name="access"
                xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
        <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
                 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:type="xs:string">
                        cn=SECURITY01,ou=groups,dc=medewerker,dc=nl
        </saml:AttributeValue>
</saml:Attribute>
```

Deze autorisatie zou goed samen kunnen gaan met een Identiry en Access Management oplossing. IAM zou bijvoorbeeld de securitygroepen kunnen koppelen aan nieuwe gebruikers die bepaalde rechten krijgen op grond van hun functieprofiel. Ook zou je in IAM met een autorisatie matrix real time een gebruiker toegang tot een applicatie kunnen verlenen of juist rechten intrekken.