2at.

AuthNRequest Scoping onderzoek

SURFnet

Nick Boszhard

Datum: 14-06-2018
Versie: 1.0

Inhoudsopgave

Onderzoeksvraag

Henny Bekker van SURFnet heeft de volgende onderzoeksvraag aan 2at voorgelegd:

“We zouden graag door jullie uitgezocht hebben hoe een AD FS 2016 IdP geconfigureerd moet worden om op basis van het scoping element (de Issuer) in het AuthNRequest toegang tot een bepaalde dienst (of diensten) geautoriseerd kan worden.”

Use case

Bij de onderzoeksvraag heeft Henny de volgende use case als voorbeeld gegeven:

“Een groep gebruikers met een account op een AD FS 2016 IdP mag enkel en alleen bij één bepaalde dienst (zeg "eduroam Visitor Access (eVA - eduGAIN) | SURFnet" met entityID="https://eduroamvisitoraccess.org/simplesaml/module.php/saml/sp/metadata.php/edugain-sp") die is gekoppeld via SURFconext.
De andere gebruikers mogen bij alle diensten die via SURFconext zijn gekoppeld aan de betreffende IdP. De IdP moet aan de hand van het scoping element (de Issuer) in het AuthNRequest de betreffende groep gebruikers met uitzondering van de betreffende dienst de toegang tot alle andere op SURFconext aangesloten diensten ontzeggen.”

Werkzaamheden

Op basis van de onderzoeksvraag en de use case, heeft 2at de volgende werkzaamheden voorgesteld, met de inschatting dat deze 3 dagen zouden kosten:

1. Opstellen van een plan
2. Realiseren van een proefopstelling:
    a. Afstemming m.b.t. welke diensten
    b. Het koppelen van deze diensten
    c. Opzetten/aanpassen Relying Party Trust
3. Testen en debuggen
4. Documenteren

Dit rapport omhelst zowel het plan als de documentatie, alsmede een conclusie/advies.

Onderzoeksaanpak

2at heeft voor deze onderzoeksvraag de volgende aanpak gehanteerd:

1. Een SURFnet proefopstelling inrichten naar voorbeeld van de Harting College AD FS 2016 server
    a. AD FS configureren
    b. SURF Sterke Authenticatie configureren
2. SURFdienst koppelen aan de SURFnet proefopstelling
3. Ophalen meetgegevens en analyse
    a. Fiddler traces
    b. AD FS logfile
4. Desktop research/inventarisatie m.b.t. mogelijkheden
    a. AD FS Access Control Policies
    b. AD FS Application Groups
    c. Uitbreiden van beschikbare claim descriptions
    d. Code uitbreidingen

Bevindingen

2at heeft stap voor stap een aantal werkzaamheden uitgevoerd en daarnaast de informatie geanalyseerd. Hieronder zijn de bevindingen puntsgewijs uitgechreven.

AD FS 2016 configuratie

Voor de AD FS configuratie en het aanmaken van een Relying Party Trust voor SURFconext hebben we de handleiding op de SURFnet wiki gebruikt:

https://wiki.surfnet.nl/display/services/Configure+Office+365+with+AD FS+and+SURFconext+Step-by-Step

Hierbij heeft 2at geen bijzonderheden opgemerkt.

Koppelen SURFdienst

In overleg met SURFnet zijn er twee SURFdiensten gekoppeld:

- Name: SURFdropjes test SP | Braindrops
  entityID: https://beta.surfnet.nl/SURFdropjesSP
- Name: SURFdropjes test SP (Alpha) | Braindrops
  entityID: https://alpha.surfnet.nl/SURFdropjesSP

Traces van het netwerkverkeer

Om vast te stellen of er daadwerkelijk scoping elementen vanuit de gekoppelde diensten werden meegegeven, heeft 2at een aantal netwerk-traces opgenomen. Hieruit bleek dat er inderdaad correct gevormde scoping elementen werden doorgegeven:

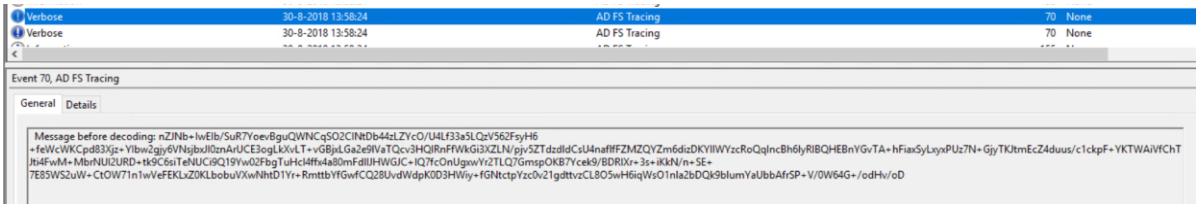
```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
```

```
    ID="CORTO1520a38475325cde7addc3e9e01a1e97fd1b486b" Version="2.0"
    IssueInstant="2018-08-30T07:50:57Z"
    Destination="https://AD FS4-aai.wind.surfnet.nl/AD FS/ls/" ForceAuthn="true"
    AssertionConsumerServiceURL=
        "https://engine.connect.surfconext.nl/authentication/sp/consume-assertion"
    ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST">
  <saml:Issuer>
    https://engine.connect.surfconext.nl/authentication/sp/metadata
  </saml:Issuer>
  <samlp:NameIDPolicy AllowCreate="true"/>
  <samlp:Scoping ProxyCount="10">
    <samlp:RequesterID>
      https://beta.surfnet.nl/simplesaml/module.php/saml/sp/metadata.php/connect
    </samlp:RequesterID>
  </samlp:Scoping>
</samlp:AuthnRequest>
```

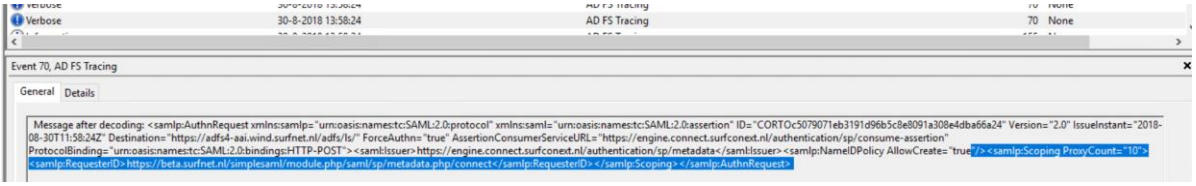
AD FS logging

De volgende stap is geweest om uitgebreide AD FS logging aan te zetten en vervolgens te controleren of het scoping element ook bij AD FS binnen komt en als zodanig wordt herkend.

Hieronder een voorbeeld van de decodering door AD FS:



En daarna:



Het scoping element wordt dus correct doorgegeven en ontvangen door AD FS. De onderzoeksvraag vanaf dit punt is of en - zo ja - hoe dit scoping element gebruikt kan worden om het gewenste usage scenario te realiseren.

Microsoft geeft aan dat er een verbeterde ondersteuning in AD FS 2016 zit voor SAML 2.0[1]. Sectie 3.4.1.2 van de SAML core specification[2] zegt daarbij het volgende m.b.t. het scoping element:

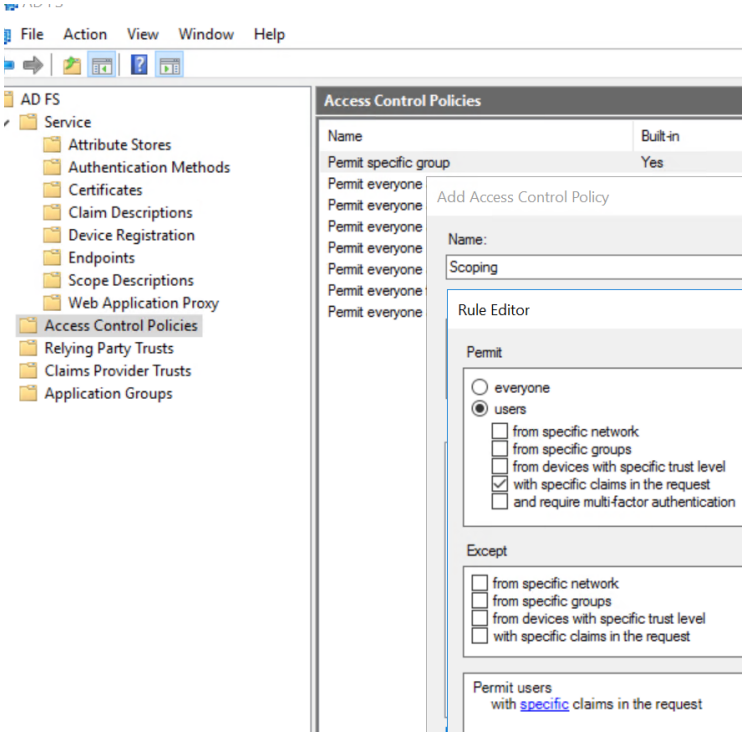
"The element specifies the identity providers trusted by the requester to authenticate the presenter, as well as limitations and context related to proxying of the message to subsequent identity providers by the responder".

Het bovenstaande geeft goede hoop dat het gebruik van het scoping element zoals door SURFnet gevraagd, in AD FS 2016 mogelijk is. Dit wordt versterkt door het feit dat dat AD FS 2016 – in tegenstelling tot eerdere versies van AD FS – geen foutmelding geeft wanneer een relying party een scoping element meegeeft in een request.

Alle verdere (summiere) documentatie die 2at heeft kunnen vinden over het gebruik van scoping in AD FS 2016, betrof echter andere gebruiksscenarios. Op basis van dit alles heeft 2at het onderzoek vervolgd op de wijze zoals hieronder wordt beschreven.

Access Control Policies in AD FS

Binnen AD FS is het mogelijk om Access Control Policies aan te maken. 2at heeft geprobeerd een custom Access Control Policy aan te maken, om op die manier wellicht het scoping element als claim te kunnen gebruiken.



[1] Zie https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/operations/improved-interoperability-with-saml-2.0

[2] Zie https://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

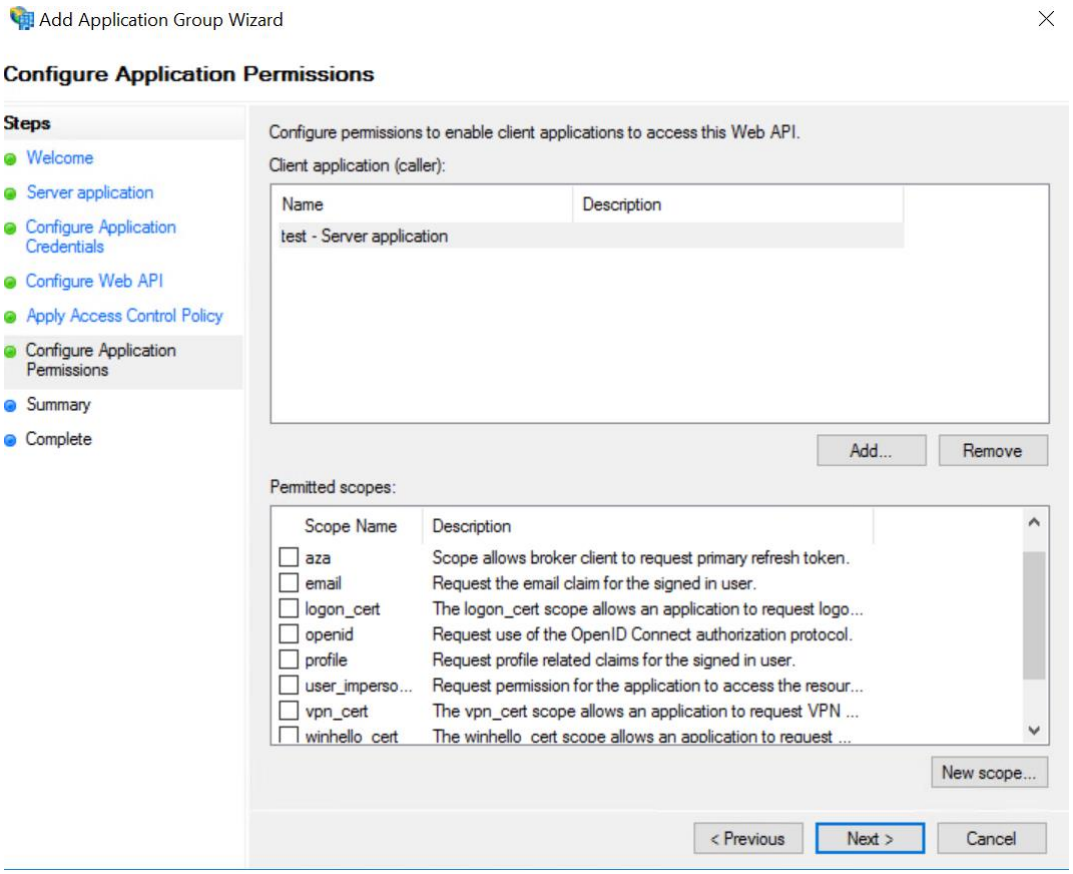
Access Control Policies kunnen standaard gebaseerd worden op een voorgedefinieerde lijst van rule templates. Een voorbeeld hiervan is het filteren van toegang op basis van specifieke claims in de inkomende request. Filteren op basis van het scoping element blijkt hiermee echter helaas niet mogelijk met de standaar beschikbare rule templates.

→ Op deze manier is het daarom niet mogelijk om via Access Control Policies scoping elementen te gebruiken.

AD FS Application Groups

Een volgende mogelijkheid die 2at onderzocht heeft is het gebruik van AD FS Application Groups[3]. Met AD FS Application Groups is het mogelijk om meerdere entiteiten die authenticatie behoeven, samen te voegen in één Application Group. Het is dan niet meer nodig om voor elk van deze applicaties losse Relying Party Trusts aan te maken.

Application groups kennen een vorm van scoping die configureerbaar is, zowel via de AD FS beheerconsole (zie onder) als via Powershell[4]. 2at heeft onderzocht of dit gebruikt kan worden om het gewenste usage scenario te realiseren. Helaas bleek dit niet het geval.

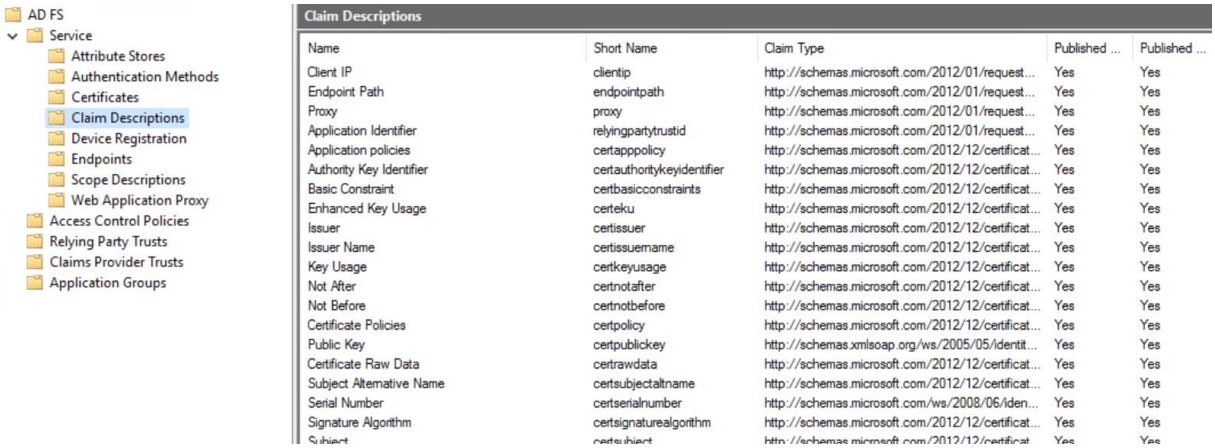


[3] Zie https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/overview/ad-fs-scenarios-for-developers
[4] Zie https://docs.microsoft.com/en-us/powershell/module/adfs/add-adfswebapiapplication?view=win10-ps

Uitbreiden beschikbare Claims Descriptions

Één van de Access Control Policies rule templates die beschikbaar is binnen AD FS, maakt gebruik van claims in de incoming claim set die AD FS opbouwt op basis van de binnenkomende request. In deze rule template wordt gebruik gemaakt van de lijst met claim descriptions om het binnenkomende claimtype te identificeren dat moet worden gevalideerd. Er is standaard geen claim description beschikbaar waarin de waarde van het scoping element of het onderliggende RequesterID als claim beschikbaar wordt gesteld.

Daarom heeft 2at onderzocht of het mogelijk is om een custom claim description aan AD FS toe te voegen om deze vervolgens alsnog in de betreffende rule template te gebruiken. Helaas heeft 2at geen manier gevonden waarop dit via standaard configuratie vanuit de AD FS beheerconsole of via Powershell mogelijk is.



Code-uitbreidingen

Ten slotte heeft 2at onderzocht of het mogelijk is middels een code-extensie in te grijpen in de standaard AD FS request processing engine en bijbehorende pipeline[5] om zo na het valideren van het binnenkomende request een extra claim toe te voegen aan de inkomende claimset met daarin de waarde van het scoping element. In eerdere versies van AD FS was deze request processing pipeline eenvoudig beschikbaar voor uitbreidingen en was het aanpassen hiervan in enige mate door Microsoft gedocumenteerd.

Dit is helaas veranderd met de komst van AD FS 2016. Deze bevat een volledig vernieuwde servicearchitectuur en daarmee ook een nieuwe request processing engine / pipeline. 2at heeft gevalideerd dat de manieren waarop code uitbreidingen in eerdere versies van AD FS mogelijk waren, niet meer beschikbaar zijn en heeft geen (documentatie over) door Microsoft ondersteunde manieren kunnen vinden om de benodigde uitbreidingen door te voeren.

[5] Zie https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/technical-reference/the-role-of-the-claims-pipeline

Conclusie/advies

De conclusie van 2at op dit moment is dat het met out-of-the-box functionaliteit en door Microsoft ondersteunde en gedocumenteerde functionaliteiten niet mogelijk is om gebruik te maken van scoping elementen binnen AD FS 2016 op de wijze die SURFnet wenst.

Een alternatieve oplossing kan zijn om gebruik te maken van uitgebreidere code-aanpassingen aan AD FS. Dit gebeurt dan op basis van een onderzoek middels reverse-engineering van de nieuwe – niet door Microsoft gedocumenteerde – request pipeline en op het hierop inhaken – indien dit mogelijk blijkt – op een wijze die mogelijk niet door Microsoft ondersteund wordt. Het uitwerken van deze mogelijkheid viel buiten de scope van het onderzoek door 2at.