Rollekataloget

Anvendelse af Attribute Store

Version: 1.4.0
Date: 11.03.2023
Author: BSG

Indhold

1 Indledning

Dette dokument er rettet mod teknikere der skal opsætte og konfigurere kommunens AD FS, så det er muligt for kommunens medarbejdere at logge på fagsystemer, hvor medarbejdes roller i dette fagsystem er administreret via Rollekataloget.

Det forudsættes at læseren har kendskab til konfiguration af AD FS.

1.1 Forskellige versioner

Der findes 3 forskellige version af AD FS Attribute Storet. Et til hhv 4.0 og 5.0, dvs til hhv Windows Server 2016 og Windows Server 2019/2022.

Det er vigtigt at den rigtige installationspakke anvendes.

Attribute Storet består af 1 DLL, der skal installeres under c:\Windows\AD FS folderen på AD FS serveren/serverne.

- RoleCatalogueAttributeStore.dll

1.2 Forudsætninger

Hvis man ønsker at attribut storet skal kunne logge til en systemlog, skal systembrugeren havde adgang til at skrive til folderen c:\logs, eller systembrugeren skal kunne oprette en loggruppe i windows event loggen.

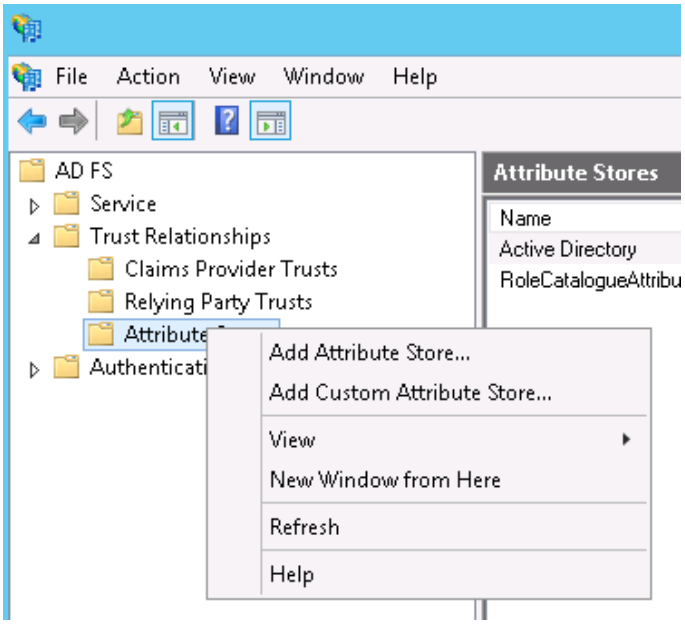
2 Installation af Attribute Store

Den nævnte DLL skal kopieres ind under folderen 'C:\Windows\AD FS' på Windows serveren, og hvis der er tale om en opgradering af attribute storet, skal AD FS servicen midlertidig stoppes, da den låser for overskrivningen af filerne.

Efter installation af filen skal AD FS servicen startes igen hvis den har været stoppet i forbindelse med installationen.

2.1 Opsætning og konfiguration

Inde i AD FS skal man tilføje Rollekatalogets attribute store ved at gå ind under "Trust Relationships" og vælge "Add Custom Attribute Store…"

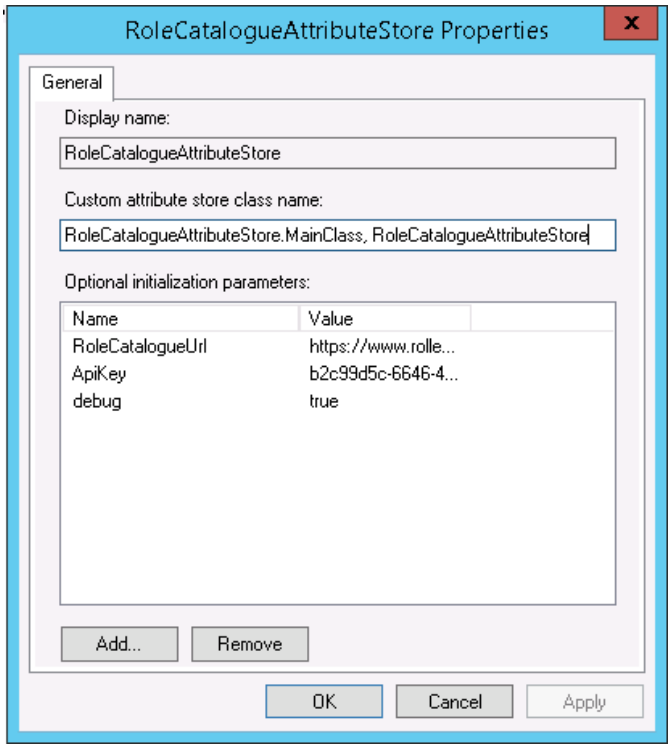


Her udfyldes 'Display Name' med

`RoleCatalogueAttributeStore`

og 'Custom attribute store class name' udfyldes med

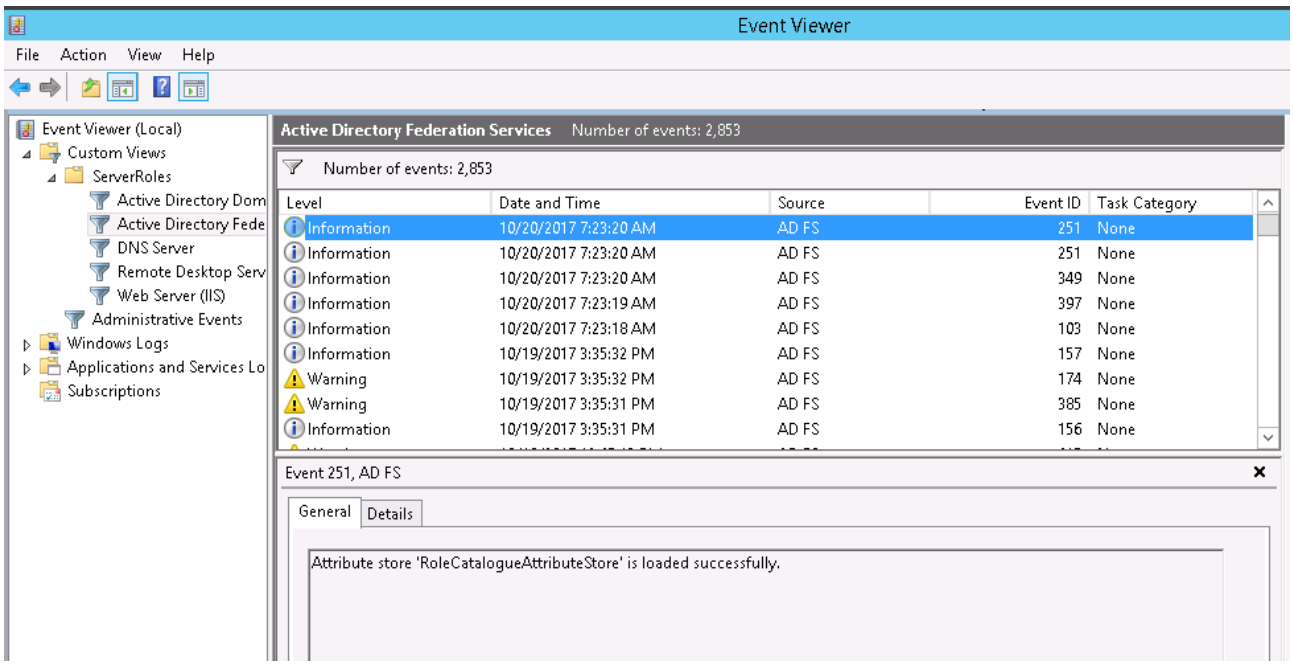
`RoleCatalogueAttributeStore.MainClass, RoleCatalogueAttributeStore`



Endeligt skal der tilføjes 3-4 parametre til opstart, ved navn

- **RoleCatalogueUrl**. URL adressen på Rollekataloget.
- **ApiKey**. Den hemmelige nøgle som bruges til at kommunikere med Rollekataloget. Nøglen kan udleveres af driftsoperatøren.
- **Debug**. Sæt til 'false' som default. Kan sættes til 'true' for at slå detaljeret logning til.
- **LogToEventLog**. Kan udelades, men hvis den er til stede, og værdien er 'true', så logges til Windows Event loggen i stedet for den flade fil.
- **Domain**. Kan udelades, men hvis den er til stede, bruges den til at angive hvilket brugerdomæne i rollekataloget der laves opslag i (anvendes specifikt til brug ved skole AD, hvor værdien sættes til "skole" – eller evt en anden værdi hvis man har flere domæner i spil)

Herefter er Attribute Storet klar til brug, og man kan verificere at den er korrekt konfigureret, ved at åbne Windows Event Loggen, og se at følgende log-entry findes under Active Directory Federation Services



3 Anvendelse af Attribute Store

Attribute Storet anvendes ved at opsætte Claim Rules på ens Relying Parties. Disse Claim Rules henter data fra Rollekataloget, så AD FS kan udstede dem i de tokens der sende til fagsystemet som brugeren forsøger at logge på.

Attribute Storet udstiller 4 opslag som kan foretages

- **getNameID.** Dette opslag henter brugerens identitet, formateret på det X.509 format som KOMBIT kræver. Opslaget er formodentligt kun relevant når en bruger logger på et KOMBIT system, eller et system, der anvender samme NameID konvention som KOMBIT.

- **getBasicPriviligeProfile**. Dette opslag henter brugerens roller (og potentielt dataafgrænsningsværdier) i OIO-BPP format, det er det format som KOMBIT anvender til at udtrykke en brugers rettigheder i et fagsystem.
- **getSystemRoles**. Dette opslag henter brugerens systemroller i det konkrete fagsystem som brugeren forsøger at logge på. Dette er formodentligt det mest almindelige opslag for ikke-KOMBIT fagsystemer.
- **getUserRoles**. Dette opslag henter brugerens jobfunktionsroller, der er knyttet til det konkrete fagsystem som brugeren forsøger at loge på. Dette opslag skal formodentligt ikke anvendes, med mindre der er tale om en KOMBIT-variant, hvor fagsystemet forstår kommunens jobfunktionsroller på en eller anden måde.
- **hasUserRole**. Dette opslag kan bruges til at checke om en bruger er tildelt en bestemt jobfunktionsrolle.
- **hasSystemRole**. Dette opslag kan bruges til at checke om en bruger er tildelt en bestemt systemrolle

Claim Rules afvikles i den rækkefølge de er oprettet, så det er vigtigt at man opsætter dem i den korrekte rækkefølge, hvis man fx skal bruge output fra én Claim Rule som input til en anden Claim Rule.

3.1 Claim Rule 1: Opsæt fagsystem

Den første regel man bør sætte op, er den der definerer navnet på det fagsystem som der logges på. Da Claim Rules opsættes "per Relying Party", kan denne regel fint hardkodes.

Man skal angive det system-id som it-systemet har inde i Rollekataloget når man opsætter fagsystemet med nedenstående Claim Rule.

I nedenstående eksempel er OS2kravmotor valgt som fagsystem.

```
=> add(Type = "http://rollekatalog.dk/itsystem", Value = "OS2kravmotor");
```

3.2 Claim Rule 2+3: Hent brugerens identitet

De fleste fagsystemer forventer at modtage brugerens identitet i det felt i det udgående token der hedder Subject/NameID. Normalt medsender man blot brugerens UPN, SAMAccountName eller lignende identifikation.

I det tilfælde at man skal medsende brugerens identitet på X.509SubjectName format, med det indhold som fx KOMBIT forventer, så kan man bruge Attribute Storet til at hente brugerens identitet i dette format. Her skal opsættes 2 Claim Rules (som kan springes over for ikke-KOMBIT systemer)

Første regel henter data fra Rollekataloget

```
c1:[Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
Issuer == "AD AUTHORITY"] => add(store = "RoleCatalogueAttributeStore", types =
("http://rollekatalog.dk/nameid"), query = "getNameID", param = c1.Value);
```

Anden regel udsteder disse oplysninger til fagsystemet som brugeren forsøger at logge på

```
c:[Type == "http://rollekatalog.dk/nameid"] => issue(Type =
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier", Issuer =
c.Issuer, OriginalIssuer = c.OriginalIssuer, Value = c.Value,
Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/format"
] = "urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName");
```

3.3 Claim Rule 4+5: Hent brugerens rettigheder

For alle fagsystemer hvor roller styres via Rollekataloget (og dermed er relevante for dette Attribute Store), vil der være roller tilknyttet, som skal udstedes til fagsystemet på login tidspunktet. Dette gøres ved følgende 2 claims, hvor indholdet afhænger af typen på det fagsystem man forsøger at logge på

3.3.1 Variant 1 – et KOMBIT system

Første regel, henter brugerens roller, som et OIO-BPP udtræk (query = ”oio-bpp” sikrer at dette sker).

```
c1:[Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
Issuer == "AD AUTHORITY"] && c2:[Type == "http://rollekatalog.dk/itsystem"] =>
add(store = "RoleCatalogueAttributeStore", types = ("http://rollekatalog.dk/oio-
bpp"), query = "getBasicPriviligeProfile", param = c1.Value, param = c2.Value);
```

Anden regel, sender denne OIO-BPP værdi til fagsystemet

```
c:[Type == "http://rollekatalog.dk/oio-bpp"] => issue(Type =
"dk:gov:saml:attribute:Privileges_intermediate", Issuer = c.Issuer, OriginalIssuer
= c.OriginalIssuer, Value = c.Value, Properties["http:/
/schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/attributename"] =
"urn:oasis:names:tc:SAML:2.0:attrname-format:basic", Properties["http://schemas.
xmlsoap.org/ws/2005/05/identity/claimproperties/spnamequalifier"] = "Privileges");
```

3.3.2 Variant 2 – et ikke-KOMBIT system

Hvis fagsystemet ikke understøtter OIO-BPP formatet, men blot ønsker at modtage de roller som brugeren er tildelt, i et bestemt navngivet attribut, så bruger man nedenstående regler.

Første regel, henter brugerens roller, som en liste af roller (query = ”systemroles” sikrer at dette sker).

```
c1:[Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
Issuer == "AD AUTHORITY"] && c2:[Type == "http://rollekatalog.dk/itsystem"] =>
add(store = "RoleCatalogueAttributeStore", types =
("http://rollekatalog.dk/systemroles"), query = "getSystemRoles", param =
c1.Value, param = c2.Value);
```

Anden regel, sender udsteder rollerne til en navngiven attribut, skal se ud som nedenstående. Bemærk at feltet ”Type=xxx” skal tilrettes, så det matcher det som fagsystemet forventer (navnet på den attribut hvor roller skal medsendes i). I eksemplet er angivet ”urn:dk:kravmotoren:roles”, som så skal tilrettes det aktuelle fagsystem.

```
c:[Type == "http://rollekatalog.dk/systemroles"] => issue(Type =
"urn:dk:kravmotoren:roles", Issuer = c.Issuer, OriginalIssuer = c.OriginalIssuer,
```

```
Value = c.Value, Properties["http:/
/schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/attributename"] =
"urn:oasis:names:tc:SAML:2.0:attrname-format:basic",Properties["http://schemas.
xmlsoap.org/ws/2005/05/identity/claimproperties/spnamequalifier"] = "Privileges");
```

4 Specielle KOMBIT Claim Rules

KOMBIT kræver nogle ”konfigurations”-værdier, som skal medsendes alle brugere. Disse værdier kan bare opsætte som hardkodede Claim Rules, der altid sender de samme værdier.

Når man opsætter Claim Rules for KOMBIT skal følgende Claim Rules tilføjes. Bemærk at ingen af disse regler anvender Attribute Storet, men bare er konfigurations claim rules.

4.1 KOMBIT Claim Rule 1: CVR

Kommunens CVR nummer skal udstedes. Tilret værdien 12345678 i nedenstående.

```
=> issue(Type = "dk:gov:saml:attribute:CvrNumberIdentifier", Value = "12345678",
Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/attribu
tename"] = "urn:oasis:names:tc:SAML:2.0:attrname-format:basic");
```

4.2 KOMBIT Claim Rule 2: SpecVer

Nedenstående regel skal bare hardkodet som den er

```
=> issue(Type = "dk:gov:saml:attribute:SpecVer", Value = "DK-SAML-2.0",
Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/attribu
tename"] = "urn:oasis:names:tc:SAML:2.0:attrname-format:basic");
```

4.3 KOMBIT Claim Rule 3: AssuranceLevel

I nedenstående regel skal AssuranceLevel sættes til det sikkerhedsniveau som kan opnås (1-4, 4 er højeste sikkerhedsnivau). Bemærk at KOMBIT kræver niveau 3 for mange af deres fagsystemer.

```
=> issue(Type = "dk:gov:saml:attribute:AssuranceLevel", Value = "3",
Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/attribu
tename"] = "urn:oasis:names:tc:SAML:2.0:attrname-format:basic");
```

4.4 KOMBIT Claim Rule 4: KombitSpecVer

Nedenstående regel skal bare hardkodet som den er

```
=> issue(Type = "dk:gov:saml:attribute:KombitSpecVer", Value = "1.0",
Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/attribu
tename"] = "urn:oasis:names:tc:SAML:2.0:attrname-format:basic");
```