OS2faktor

Udviklingsguide til Connectors

Version: 1.1.0
Date: 02.05.2022
Author: BSG

Indhold

1 Indledning

Dette dokument beskriver hvordan man udvikler en Connector, der gør brug af OS2faktor infrastrukturen.

En Connector er en applikation, der ønsker at bruge OS2faktor som 2. faktor i et login flow. Dette kunne være en SAML Identity Provider, en fagapplikation, eller en infrastrukturkomponent (firewall, vpn concentrator, ...).

Dokumentet er rettet mod arkitekter og udviklere der er ansvarlige for udviklingen af en sådan Connector.

2 Forudsætninger

En Connector kan kun tilgå OS2faktor infrastrukturen, hvis den er registreret som en autoriseret Connector i OS2faktor.

For at blive registreret som en sådan, skal man kontakte OS2 og indgå en tilslutningsaftale til driftsmiljøet på OS2faktor.

Når man er tilsluttet, har man mulighed for at få en API nøgle til de Connectors man ønsker at tilslutte. Der gives en API nøgle per Connector, så man kan styre adgangen per Connector. Der er ingen begrænsninger på hvor mange Connectors (og dermed API nøgler) man kan få registreret, når først man er tilsluttet infrastrukturen.

Under udviklingen af en Connector, kan et udviklingshus få en midlertidig API nøgle der kan anvendes til integrationstest.

3 Anvendelser af API'erne

På OS2faktor infrastrukturen er der udstillet en række API'er til integration. Nogle af disse er rettet mod Connectors, og forudsætter alle (på nær ét endpoint) at man anvender API nøglen i de kald man foretager.

API nøglen skal angives som en HTTP header i det kald der foretages til API'et.

Eksempel

```
GET /api/server/nsis/clients

Host: backend.os2faktor.dk

ApiKey: 00000000-0000-4000-0000-000000000000
```

Hvis man kalder API'et uden en API nøgle, eller med en ukendt/spærret nøgle, så afvises man med en HTTP 401 fejlkode.

4 API'er

Der er 4 API operationer der anvendes af Connectors. Afhængig af den konkrete Connector, er det ikke sikkert at alle API operationer skal anvendes.

4.1 Få liste af OS2faktor klienter for en given bruger

Dette endpoint anvendes af en Connector der har identificeret den bruger der skal udføre 2.faktor i det samlede login flow. Dvs en bruger der har identificeret sig med et bruger-id, og et 1.faktor akkreditiv (fx et kodeord).

Som input skal operationen bruge oplysninger der kan identificere den eller de OS2faktor klienter som er registreret på brugeren. Man kan bruge en eller flere af følgende søge-parametre

- DeviceID på en klient
- Personnummeret på brugeren

Det vil være forskelligt fra Connector til Connector hvilke af disse oplysninger der er tilgængelige, og man kan kombinere flere af disse oplysninger i sin søgning.

Input format for de forskellige oplysninger er

4.1.1 OS2faktor ID'er

Disse angives i klartekst format, dvs som 4 blokke af 3 cifre, adskildt med bindrestreg, eksempel

000-111-222-333

444-555-666-777

Brugere der ikke ønsker at binde deres OS2faktor klient til deres NemID, kan kun fremsøges ud fra kendte OS2faktor ID'er. I disse tilfælde vil kommunen have registreret det eller de OS2faktor ID'er som en bruger anvender et sted i deres infrastruktur (fx i en attribut på deres AD konto).

Den URL parameter der bruges til at angive OS2faktor ID'er hedder "deviceId", fx

```
?deviceId=000-111-222-333
```

4.1.2 Personnummer

De fleste brugere vil have knyttet deres NemID til deres OS2faktor klient(er), hvilket gør det muligt at fremsøge klienterne via opslag på personnummeret.

Personnummeret skal, inden det afgives som en søgepameter i API kaldet, formaterets på følgende måde

1. fjern evt bindestreger og mellemrum, så personnummeret er præcist 10 cifre
2. beregn en SHA256 digest af personnummeret
3. base64 enkod resultatet

Ovenstående vil, på værdien 1111111118 resultere i følgende søgeparameter "K3b9tAV9cSdvl4lwV5v38FGxfZgeIuCaxeTSs1xaa0w="

Den URL parameter der bruges til at angive personnummeret hedder "ssn", fx

```
?ssn=K3b9tAV9cSdvl4lwV5v38FGxfZgeIuCaxeTSs1xaa0w=
```

4.1.3 API endpoint

Det API endpoint der skal kaldes, er lokaliseret her

```
https://backend.os2faktor.dk/api/server/nsis/clients
```

og det kaldes med en HTTP GET operation, med mindst en af ovenstående søgeparameter, fx

```
https://backend.os2faktor.dk/api/server/clients?ssn=K3b9tAV9cSdvl4lwV5v38FGxfZge
IuCaxeTSs1xaa0w=
```

4.1.4 Ouput

Operationen returnerer en HTTP 200 ved succes, og et JSON array som output, der indeholder alle klienter der matcher søgekriterierne. Formatet på JSON arrayet er som følgede

- **deviceId**. Er det unikke ID på klienten, og består af 4 blokke af 3 cifre.
- **type**. Er en tekst-streng, der angiver typen af klient. Følgende værdier anvendes pt: CHROME, ANDROID, EDGE, IOS, WINDOWS, YUBIKEY, TOTP
- **name**. Er en tekst-streng, som brugeren selv har valgt, og som brugeren kan anvende til at identificere den klient som ønskes anvendt.
- **hasPincode**. Angiver om klienten er pinkodebeskyttet. Kan fx bruges til at filtrere alle klienter væk som ikke er pinkodebeskyttet hvis man ønsker dette.
- **nsisLevel**. Angiver det NSIS niveau som klienten er indrulleret under. Kan være ”NONE”, ”LOW”, ”SUBSTANTIAL” og ”HIGH”.
- **prime**. Angiver om brugeren har valgt denne klient som sin primære klient. Fx kan man vælge at forvælge eller fremhæve den primære klient.
- **roaming**. Angiver om klienten roamer (en Chrome feature), og er mest til info formål

```
[
  {
    "deviceId": "000-111-222-333",
    "type": "CHROME",
    "name": "Chromebook A1",
    "hasPincode": true,
    "nsisLevel": "SUBSTANTIAL",
    "prime": true,
    "roaming": false
  },
  {
    "deviceId": "444-555-666-777",
    "type": "ANDROID",
    "name": "Samsung S9",
    "hasPincode": true,
    "nsisLevel": "SUBSTANTIAL",
    "prime": false,
    "roaming": false
  }
]
```

Ovenstående er et eksempel på output fra API operationen. Bemærk at en Connector bør håndtere scenarier hvor der ikke er registreret nogen klienter, hvor der er registreret netop én klient, og hvor der er registreret mere end én klient på en bruger forskelligt.

Hvis der er mere end én klient, og en Connector har mulighed for at lade brugeren vælge, så skal listen af klienter præsenteres for brugeren. Hvis en Connector ikke kan interagere med brugeren, vælges én af klienterne fra listen af Connectoren (vælg evt smartphone klienter først hvis disse er tilgængelige).

4.2 Initiér et 2.faktor autentifikationsflow med en OS2faktor klient

Når en Connector har afgjort hvilken af brugerens OS2faktor klienter der skal bruges til at udføre 2.faktor i autentifikationsflowet, udføres et API kald til OS2faktor infrastrukturen, der initierer 2.faktor flowet.

API operationen tager OS2faktor ID'et som input (deviceId i output fra foregående operation), og returnerer et status-objekt, der indeholder alle relevante oplysninger for at Connetoren kan håndtere det videre flow.

4.2.1 API endpoint

Det API endpoint der skal kaldes, er lokaliseret her

```
https://backend.os2faktor.dk/api/server/client/{deviceId}/authenticate
```

og det kaldes med en HTTP PUT operation, hvor {deviceId} er erstattet med OS2faktor ID'et på den valgte klient, fx

```
https://backend.os2faktor.dk/api/server/client/000-111-222-333/authenticate
```

4.2.2 Ouput

Operationen returnerer en HTTP 200 ved succes, og et JSON array som output, der et status objekt i JSON format. Status objektet har følgende elementer

- **subscriptionKey**. Dette er en hemmelig nøgle, der anvendes i efterfølgende API kald, til at forespørge på status på autentifikationsflowet.
- **pollingKey**. Dette er en offentlig nøgle, der kan anvendes via usikre kanaler (fx i brugerens web-browser) til at forespørge på status på autentifikationsflowet
- **clientNotified**. Dette er en bolsk værdi, der angiver om OS2faktor klienten er blevet notificeret direkte af OS2faktor infrastrukturen (fx via en push notifikation)
- **clientAuthenticated**. Dette er en bolsk værdi, der angiver om OS2faktor klienten har godkendt autentifikationsflowet (værdien er altid false ved dette API kald, da brugeren ikke har kunnet nå at reagere endnu)
- **clientRejected**. Dette er en bolsk værdi, der angiver om OS2faktor klienten har afvist autentifikationsflowet (værdien er altid false ved dette API kald, da brugeren ikke har kunnet nå at reagere endnu)
- **challenge**. For de fleste klienter skal Connectoren vise en kontrolkode for brugeren, som brugeren kan sammenligne med en tilsvarende kontrolkode der vises i brugerens klient. Denne kontrolkode er indeholdt i dette felt. Bemærk at hvis redirectUrl elementet er udfyldt, skal der ikke vises en kontrolkode for brugeren, og så er denne værdi alene til fejlsøgningsformål.
- **redirectUrl**. For de fleste klienter vil denne værdi være tom, men enkelte klienter har brug for at interagere med OS2faktor backenden direkte, via en webbrowser. Hvis

denne værdi er udfyldt, skal Connectoren åbne et browser-vindue på denne URL. Hvis Connectoren selv afvikles i en web-browser, kan man med fordel (forsøge at) åbne browser-vinduet med javascript, da OS2faktor backenden så kan (forsøge at) lukke vinduet igen, så slutbrugeren ikke selv skal gøre dette. Hvorvidt det lykkedes, afhænger af indstillinger i browseren, så under alle omstændigheder bør man tilbyde brugeren aktivt at åbne vinduet via en knap/link.

To eksempler på dette output er vist nedenfor. Det første er for en klient hvor brugerens interaktion med klienten foretages udenfor Connectorens kontrol (dvs redirectUrl er ikke udfyldt, istedet er challenge elementet relevant at vise for slut-brugeren). Det andet er for en klient, hvor Connectoren skal åbne et browservindue for brugeren, hvor klienten skal anvendes (dvs redirectUrl er udfyldt).

Eksempel 1 – uden redirectUrl


```
{
  "subscriptionKey": "00000000-0000-4000-0000-000000000000",
  "pollingKey": "00000000-0000-4000-0000-000000000000",
  "clientNotified": true,
  "clientAuthenticated": false,
  "clientRejected": false,
  "challenge": "ABCD",
  "redirectUrl": null
}
```


Eksempel 2 – med redirectUrl


```
{
  "subscriptionKey": "00000000-0000-4000-0000-000000000000",
  "pollingKey": "00000000-0000-4000-0000-000000000000",
  "clientNotified": false,
  "clientAuthenticated": false,
  "clientRejected": false,
  "challenge": "1vqYB3wiAB8UN21Knxa1FyX1uXc28q0kxduSTElBTWk=",
  "redirectUrl": "https://frontend.os2faktor.dk/ui/yubikey/login/00000000-0000-
4000-0000-000000000000"
}
```


Afhængig af de to scenarier, skal Connectoren enten åbne et nyt browser vindue, der henter indholdet af redirectUrl, eller afvente at brugeren gennemføre autentifikationsflowet udenfor Connectorens kontrol.

De efterfølgende API operationer kan anvendes til at holde øje med status på flowet.

4.3 Forespørg på status på et autentifikationsflow (autoriseret)

En Connector kan foretage kald til OS2faktor backenden for at forespørge på status på et login flow. Afhængig af typen af Connector, kan det ikke altid lade sig gøre. Specielt hvis Connectoren kun aktiveres på forespørgsler fra brugeren (fx hvis Connectoren er en web-applikation).

I disse tilfælde bør man også kigge på den anonyme udgave af dette API håndtag, og så foretage det ude i web-browseren via Javascript. Dette scenarier er beskrevet detaljer under den API operation.

Hvis en Connector har brug for at kende status på et autentifikationsflow, foretages et API kald til denne operation, med subscriptionKey som input. Output fra kaldet er identisk med ovenstående operation, dog vil clientAuthenticated og clientRejected formodentligt have ændret status.

En Connector kan fx kalde dette endpoint med regelmæssige intervaller (fx 1 gang i sekundet) for at forespørge på status.

En Connector bør håndtere det scenarie hvor en bruger aldrig aktiverer sin OS2faktor klient (dvs status på disse felter aldrig ændrer sig), og afbryde autentifikationsflowet efter en passende timeout periode.

4.3.1 API endpoint

Det API endpoint der skal kaldes, er lokaliseret her, og kaldes med en HTTP GET operation

https://backend.os2faktor.dk/api/server/notification/{subscriptionKey}/status

hvor subscriptionKey er udfyldt med den værdi som man fik fra forrige API operation, fx

https://backend.os2faktor.dk/api/server/notification/00000000-0000-4000-0000-000000000000/status

4.3.2 Ouput

Operationen returnerer en HTTP 200 ved succes, og indholdet er identisk med forrige API operation, dog kan status have ændret sig (clientAuthenticated og clientRejected felterne).

4.4 Forespørg på status på et autentifikationsflow (anonymt)

Et almindeligt scenarie for en Connector, specielt en web-baseret Connector, er at selve autentifikationsflowet foregår i en web-browser, og Connectoren har derfor ikke nogen let/praktisk måde at anvende det første API kald, da dette så vil være afkoblet fra brugerens web-session.

En måde at håndtere dette på, er via følgende flow

1. Connectoren gennemfører 1.faktor af autentifikationsflowet (fx brugernavn/kodeord)
2. Connectoren laver et API kald til OS2faktor backenden, for at få listen af OS2faktor klienter for denne bruger
3. Connectoren vælger en af OS2faktor klienterne, og initierer 2.faktor flowet med denne klient, og får et status objekt tilbage, der indeholder både en subscriptionKey og en pollingKey
4. Connectoren præsentere brugeren for en HTML side, hvor kontrolkoden vises (eller hvor der en knap der kan åbne et popup vindue med redirectUrl værdien) – på denne side afvikles en stump javascript i baggrunden, der via pollingKey forespørger på status på autentifikationsflowet (via dette anonyme API endpoint)
5. Når dette javascript opdager at brugeren har afsluttet 2.faktor flowet (fx ved at godkende/afvise via sin OS2faktor klient), sendes denne oplysning til Connectorens web-backend
6. Connectoren foretager nu et kald til status API'et med subscriptionKey for at få resultatet af 2.faktor autentifikationsflowet (godkendt/afvist)

4.4.1 API endpoint

Det API endpoint der skal kaldes, er lokaliseret her, og kaldes med en HTTP GET operation, og modsat de andre API operationer, skal der ikke angives en API nøgle i kaldet

https://backend.os2faktor.dk/api/notification/{pollingKey}/poll

pollingKey udfyldes med den pollingKey som Connectoren fik da den initerede 2.faktor autentifikationsflowet, fx

https://backend.os2faktor.dk/api/notification/00000000-0000-4000-0000-000000000000/poll

4.4.2 Ouput

Operationen returnerer en HTTP 200 ved succes, og indeholder følgende JSON struktur

```
{
  "stateChange": false
}
```

Værdien stateChange er true når brugeren enten har godkendt eller afvist autentifikationsflowet i sin OS2faktor klient.

4.4.3 Eksempel javascript

Den javascript der kan afvikles ude i brugerens browser, kan baserer sig på nedenstående kodestump. Man bør anvende setInterval javascript metoden, så man ikke blokerer GUI'en i browseren, samt sætte et passende interval, fx 1-2 sekunder mellem hvert kald til API'et.

```
// TODO: inject actual pollingKey into javascript
var pollingKey = "00000000-0000-4000-0000-000000000000";

// TODO: abort after 60-120 seconds
setInterval(function() {
  $.ajax({
    url: "https://backend.os2faktor.dk/api/notification/" + pollingKey +
"/poll",
    success: function(data, textStatus, xhr) {
      if (data && data.stateChange == true) {
        // TODO: replace with actual way to inform Connector
        //       web-backend about state-change
        $("#loginForm").submit();
      }
    }
  });
}, 1000);
```

5 Opsummering

Opgaven med at tilsluttet en Connector til OS2faktor infrastrukturen er en forholdsvis overkommelig opgave, da der kun er 3-4 API operationer man skal integrere op mod, og flowet er forholdsvist lineært.

Håndteringen af det at afvente svar fra OS2faktor klienten er det mest komplekse, og man bør sikre korrekt håndtering af de typiske fejlscenarier, herunder

- En bruger afslutter aldrig login-flowet (timeout / max-forsøg i alle status-løkker)
- En bruger har ikke nogen OS2faktor klienter registreret

Og sørg for at der er en passende pause mellem hvert kald til backenden. Den automatiske DOS/DDOS beskyttelse på infrastrukturen vil låse en Connector ude, hvis den laver mange tusinde identiske kald i sekundet.