Gadget van derden integreren met Finse in SSO-modus

Inhoud

Inleiding

Dit document beschrijft wat er nodig is voor de integratie van Gadgets van drie[e] partijen met Finse terwijl het systeem in de Single Sign-On (SSO)-modus staat. Een voorbeeld wordt ook gegeven voor de NON SSO - modus.

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Cisco Finesse
- SSO
- Finse 3de-partijgadgets

Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:

- Cisco Finse versie 11.6
- SSO
- gadget 3
- 3e REST-dienst.

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk levend is, zorg er dan voor dat u de mogelijke impact van om het even welke opdracht begrijpt.

Achtergrondinformatie

Dit zijn de eerste stappen terwijl de agent probeert in te loggen en authenticatie aan te brengen met SSO of NONSSO.

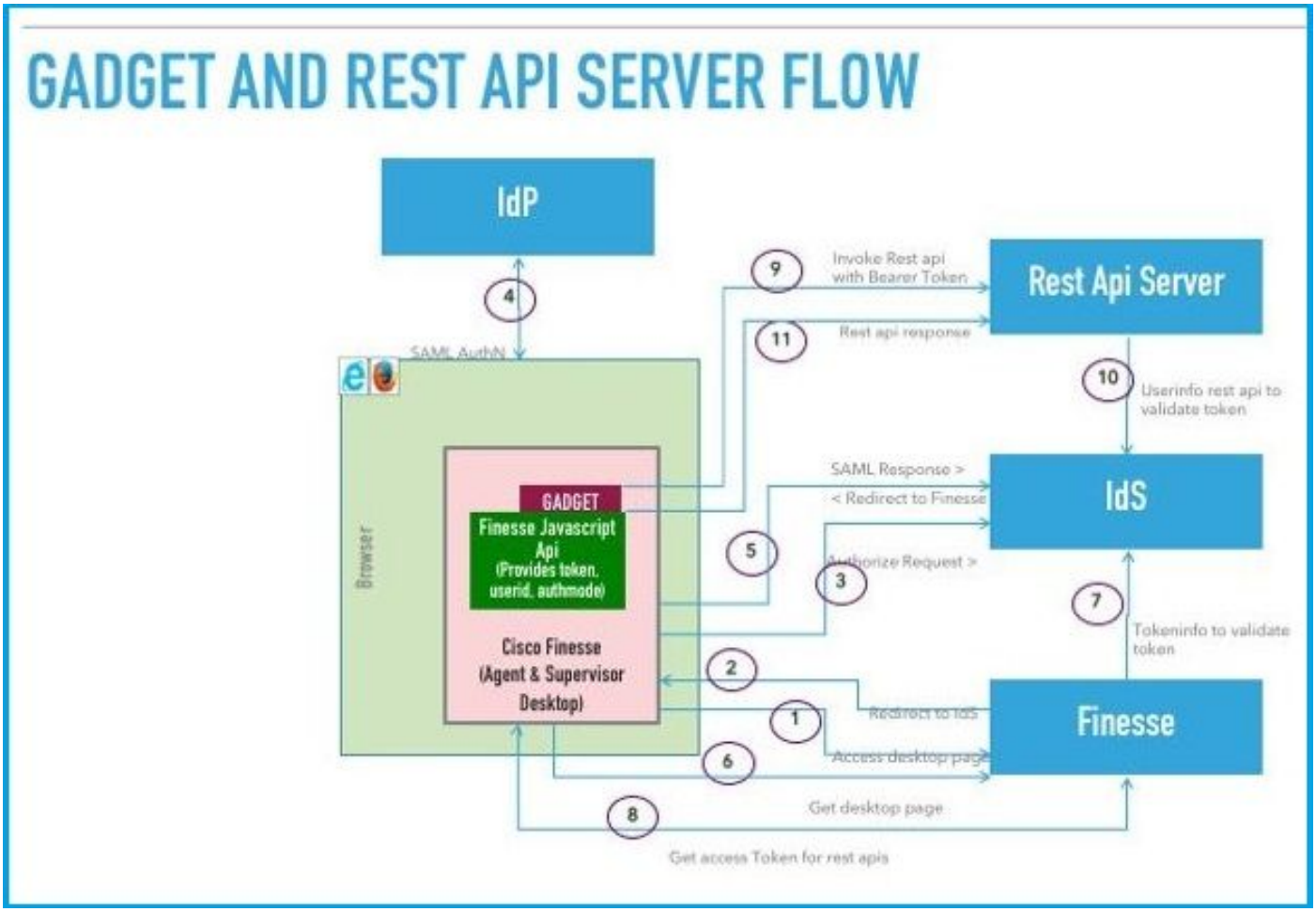
in de tweede stap wordt beschreven wat in overweging moet worden genomen na succesvolle authenticatie in het geval van SSO en NONSSO.

1. Wanneer u een desktoplogbestand hebt ingevoerd, detecteert Finesse de systeemautomatiseringsmodus (SSO/NONSSO) en wordt op basis van de automatische modus Auth een geschikte loginpagina weergegeven. De gebruikers zien de pagina Inloggen IDP in het geval van de SSO-modus en de pagina Inloggen finesse in het geval van de NONSSO-modus.

2. Na succesvolle authenticatie worden alle verzoeken op basis van de automatische autorisatiemodus gewaarmerkt. Voor SSO-implementaties dragen alle verzoeken aan Finesse een toegangstoken als onderdeel van de opvraagpagina. Het token is gevalideerd tegen de IDP-server voor een succesvolle verificatie. Voor verzoeken om webdiensten van derden moet de Auth header echter worden ingesteld op basis van het door de webdienst van derden uitgevoerde verificatieschema. In het geval van NONSSO-plaatsing, dragen alle verzoeken de **Basic** Auth header met basis64 gecodeerde gebruikersnaam en wachtwoord. Alle verzoeken in dit geval worden gevalideerd in het kader van de lokale Finse-gegevensbank.

Uitleg basismodel van de interactie voor de SSO-modus

Deze *afbeelding* toont het basismodel van de interactie tussen een 3de-partijgadget, Finesse, IDS, en een 3de-partijenREST-service, wanneer het systeem in SSO-modus staat.



Afbeelding

Hier is de beschrijving van elke stap die in de afbeelding wordt weergegeven.

1. Agent/Supervisor heeft toegang tot Finse desktop URL. (Voorbeeld: *https://finesse.com:8445/desktop*)
2. Finesse detecteert dat de authenticatiemodus SSO is en stuurt de browser door naar IDS.
3. De browser stuurt een verzoek om doorgifte aan IDS toe. Op dit punt detecteert IDS of *de gebruiker* een geldig toegangstoken heeft. Als *de gebruiker* geen geldig toegangstoken heeft, stuurt IDS-omleidingen naar de Identity Provider (IDP).
4. Als het verzoek opnieuw is gericht aan IDP, geeft IDP de logpagina op voor het authentiseren *van de gebruiker.*
5. De SAML-aanname van IDP wordt naar de IDS gestuurd, die terugkeert naar het Finesse bureaublad.
6. browser krijgt een KRAAN van de Finesse desktop pagina.
7. Finesse krijgt het toegangstoken van IDS met de SAML-autiecode.
8. Desktop krijgt het toegangstoken dat gebruikt wordt om volgende REST API's te authenticeren.
9. Gadget van derden laadt op het bureaublad en maakt gebruik van een REST API van derden met het toegangstoken (toonder) in de auth-header.
10. De REST-service van derden bevestigt het token met IDS.
11. De reactie van de derde partij op REST wordt teruggegeven aan de gadget.

Configuratie van gadgets.io.makeerste vereisten voor SSO en NONSSO-modus

Stap 1. Voor Finse REST API-oproepen die via Shindig zijn gemaakt, moeten gadgets de 'Beonder'-autorisatie toevoegen in gadgets.io.makeApplication-headers.

Stap 2. Gadgets moeten native gadgets.io.makeApplication aanroepen voor alle REST-verzoeken, de autorisatie header moet in de aanvraagparams worden ingesteld.

Voor NON SSO-implementaties is dit de Auth Header.

```
"Basic " + base64.encode(username : password)
```

Voor SSO-implementaties is dit de Auth header.

```
"Bearer "  + access_token
```

Toegangstoken kan worden opgehaald via **finesse.gadget.Config**.

```
 access_token = finesse.gadget.Config.authToken
```

De nieuwe vergunningheader moet aan de aanvraagparams worden toegevoegd.

```
 params[gadgets.io.RequestParameters.HEADERS].Authorization = "Basic " + base64.encode(username
: password);
params[gadgets.io.RequestParameters.HEADERS].Authorization = "Bearer " + access_token;
```

Stap 3. Er is een nutsmethode **GetAuthHeaderString** toegevoegd aan **hulpprogramma's**. Deze nutsmethode neemt het configuratieobject als argument en geeft de autorisatie header string terug. Gadgets kunnen gebruik maken van deze nutsmethode om de autorisatie header in request params in te stellen.

```
params[gadgets.io.RequestParameters.HEADERS].Authorization=
finesse.utilities.Utilities.getAuthHeaderString(finesse.gadget.config);
```

> Opmerking: Voor API-verzoeken om webdiensten van derden moet de autetheader worden ingesteld op basis van het door de webservice van derden uitgevoerde verificatieschema. Gadget developers hebben de vrijheid om gebruik te maken van basisauth- of toonder-toonderechtheidscontroles, of van enig ander authenticatiemechanisme naar keuze.