Intégrer le gadget tiers avec Finesse en mode SSO

Contenu

Introduction

Ce document décrit ce qui est nécessaire pour l'intégration d'un gadget 3 ^{tiers} avec Finesse lorsque le système est en mode SSO (Single Sign-On). Un exemple est également donné pour le mode NON SSO.

Conditions préalables

Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Cisco Finesse
- SSO
- Gadgets tiers Finesse

Components Used

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Cisco Finesse version 11.6
- SSO
- 3gadget tiers
- Service REST tiers.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l’incidence possible des commandes.

Informations générales

Il s'agit des étapes initiales pendant que l'agent tente de se connecter et de s'authentifier avec SSO ou NONSSO.

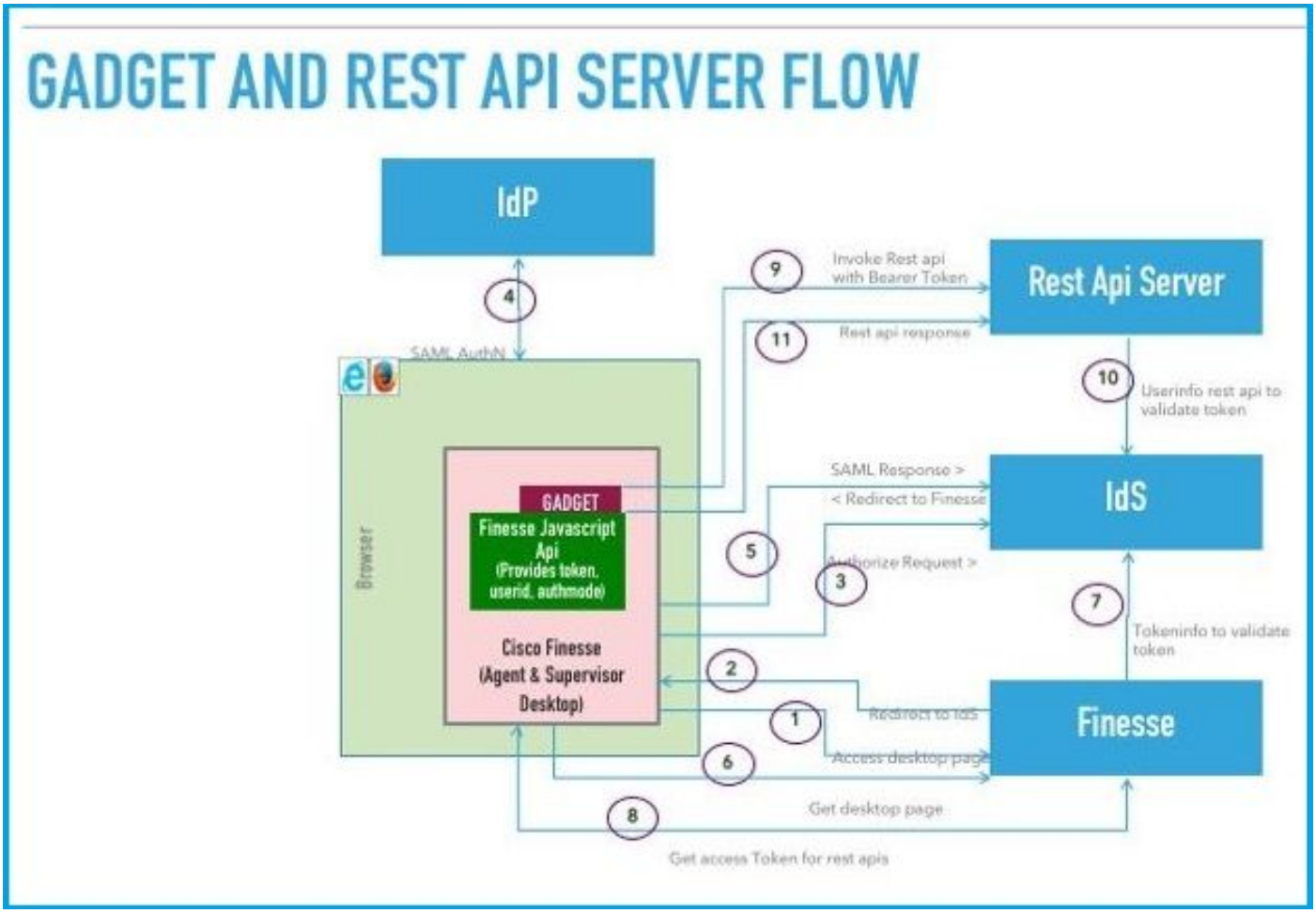
la deuxième étape décrit ce qui doit être pris en compte après une authentification réussie en cas de SSO et de NONSSO.

1. Au moment de la connexion au bureau, Finesse détecte le mode d'authentification du système (SSO/NONSSO) et, en fonction du mode d'authentification, la page de connexion appropriée s'affiche. Les utilisateurs voient la page de connexion IDP en cas de mode SSO et la page de connexion Finesse en cas de mode NONSSO.

2. Après une authentification réussie, toutes les requêtes sont authentifiées en fonction du mode d'authentification système. Pour les déploiements SSO, toutes les demandes à Finesse portent un jeton d'accès dans l'en-tête de demande. Le jeton est validé par rapport au serveur IDP pour une authentification réussie. Toutefois, pour les demandes de services Web tiers, l'en-tête Auth doit être défini en fonction du schéma d'authentification mis en oeuvre par le service Web tiers. Dans le cas d'un déploiement NONSSO, toutes les requêtes portent l'en-tête **Basic** Auth avec un nom d'utilisateur et un mot de passe codés base64. Toutes les demandes dans ce cas sont validées par rapport à la base de données locale Finesse.

Explication du modèle de base d'interaction pour le mode SSO

Cette *image* montre le modèle de base d'interaction entre un gadget tiers, Finesse, IDS et un service REST tiers, lorsque le système est en mode SSO.



Image

Voici la description de chaque étape affichée dans l'image.

1. L'agent/superviseur accède à l'URL du bureau Finesse. (Exemple : *https://finesse.com:8445/desktop*)
2. Finesse détecte que le mode d'authentification est SSO et redirige le navigateur vers IDS.
3. Le navigateur envoie la demande d'autorisation de redirection à IDS. À ce stade, IDS détecte si *l'utilisateur* a ou non un jeton d'accès valide. Si *l'utilisateur* n'a pas de jeton d'accès valide, IDS redirige vers le fournisseur d'identité (IdP).
4. Si la demande est redirigée vers IdP, IdP fournit la page *Login* pour authentifier *l'utilisateur.*
5. L'assertion SAML de IdP est envoyée à IDS, qui redirige vers le bureau Finesse.
6. Le navigateur fait une recherche de la page de bureau Finesse.
7. Finesse obtient le jeton d'accès à partir d'IDS avec le code d'authentification SAML.
8. Desktop obtient le jeton d'accès à utiliser pour authentifier les API REST suivantes.
9. Le gadget tiers se charge sur le bureau et appelle une API REST tierce avec le jeton d'accès (support) dans l'en-tête auth.
10. Le service REST tiers valide le jeton avec IDS.
11. La réponse REST tierce est renvoyée au gadget.

Configuration de gadgets.io.makerequest pour les modes SSO et NONSSO

Étape 1. Pour les appels d'API REST Finesse effectués via Shindig, les gadgets doivent ajouter un en-tête d'autorisation « Bearer » dans les en-têtes gadgets.io.makeRequest.

Étape 2. Les gadgets doivent passer des appels gadgets.io.makeRequest natifs pour toutes les demandes REST, l'en-tête d'autorisation doit être défini à l'intérieur des paramètres de demande.

Pour les déploiements NON SSO, il s'agit de l'en-tête Auth.

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

Pour les déploiements SSO, il s'agit de l'en-tête Auth.

```
"Bearer "  + access_token
```

Le jeton d'accès peut être récupéré à partir de l'objet **finesse.gadget.Config**.

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

Le nouvel en-tête d'autorisation doit être ajouté aux paramètres de demande.

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

Étape 3. Une méthode utilitaire **getAuthHeaderString** a été ajoutée dans **Utilitaires.Utilities**. Cette méthode utilitaire prend l'objet config comme argument et retourne la chaîne d'en-tête

d'autorisation. Les gadgets peuvent utiliser cette méthode d'utilitaire pour définir l'en-tête d'autorisation dans les paramètres de demande.

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

> **Note**: Pour les demandes d'API aux services Web tiers, l'en-tête d'authentification doit être défini en fonction du schéma d'authentification mis en oeuvre par le service Web tiers. Les développeurs de gadget ont la liberté d'utiliser l'authentification de base, l'authentification par jeton de support ou tout autre mécanisme d'authentification de leur choix.