Integration von Drittanbietergeräten in Finesse im SSO-Modus

Inhalt

Einführung

Dieses Dokument beschreibt, was für die Integration von Gadgets von Drittanbietern mit Finesse erforderlich ist, während sich das System im Single Sign-on (SSO)-Modus befindet. Ein Beispiel wird auch für den NON SSO-Modus gegeben.

Voraussetzungen

Anforderungen

Cisco empfiehlt, über Kenntnisse in folgenden Bereichen zu verfügen:

- Cisco Finesse
- SSO
- Finesse Gadgets von Drittanbietern

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf den folgenden Software- und Hardwareversionen:

- Cisco Finesse Version 11.6
- SSO
- Gadget von Drittanbietern
- REST-Service eines Drittanbieters.

Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

Hintergrundinformationen

Dies sind die ersten Schritte, während der Agent versucht, sich bei SSO oder NONSSO anzumelden und sich zu authentifizieren.

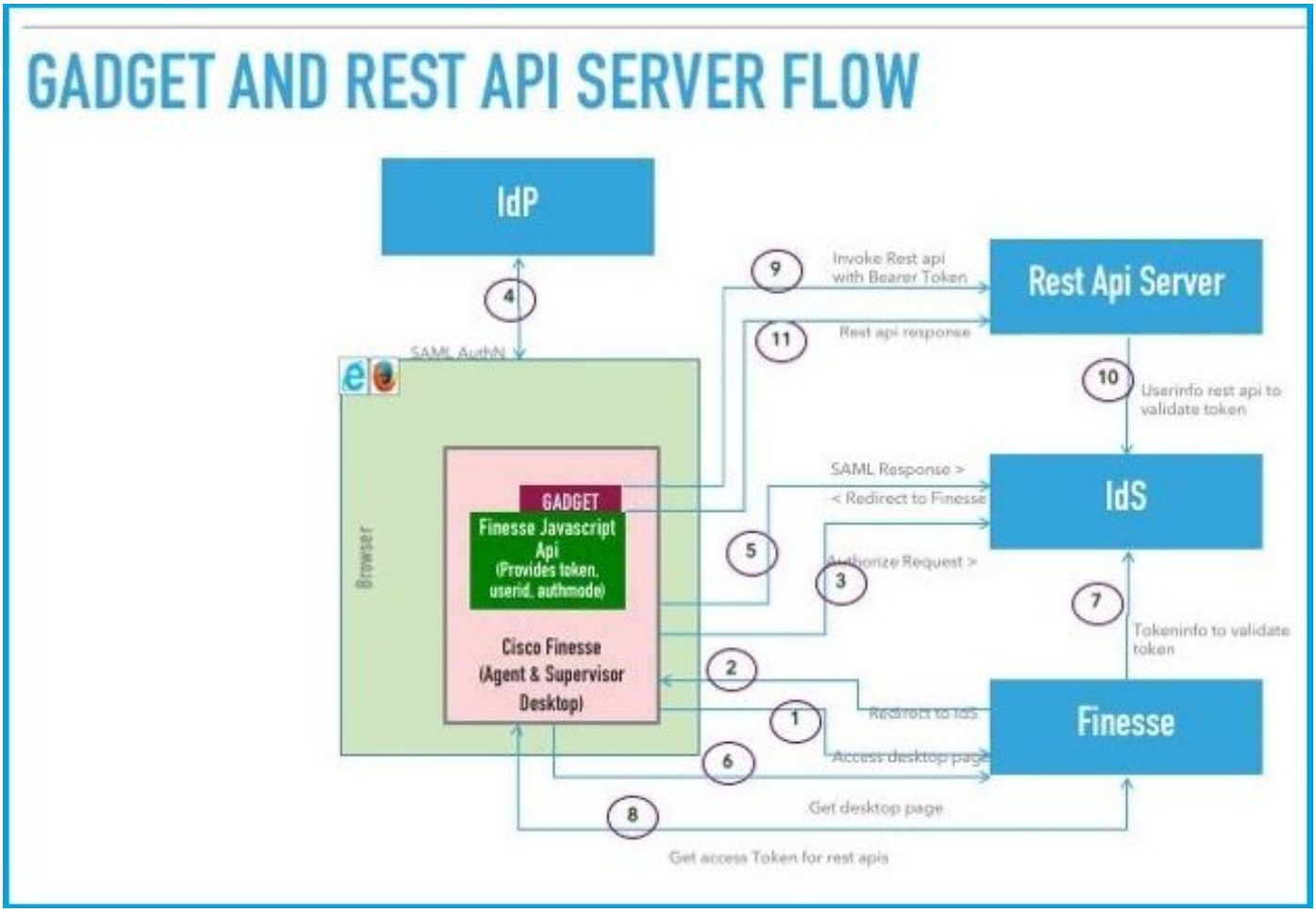
Der zweite Schritt beschreibt, was nach erfolgreicher Authentifizierung bei SSO und NONSSO berücksichtigt werden muss.

1. Bei der Desktop-Anmeldung erkennt Finesse den Systemauthentifizierungsmodus (SSO/NONSSO). Je nach Auth-Modus wird eine entsprechende Anmeldeseite angezeigt. Benutzer sehen die Seite IDP Login (IDP-Anmeldung) im Fall des SSO-Modus und im Fall des NONSSO-Modus die Seite Finesse Login (Finesse-Anmeldung).

2. Nach erfolgreicher Authentifizierung werden alle Anforderungen anhand des System-Auth-Modus authentifiziert. Bei SSO-Bereitstellungen enthalten alle Anforderungen an Finesse Zugriffstoken als Teil des Anforderungsheaders. Das Token wird für eine erfolgreiche Authentifizierung gegen den IDP-Server validiert. Bei Anfragen an Webservices von Drittanbietern muss der Auth-Header jedoch auf Basis des vom Drittanbieter-Webdienst implementierten Authentifizierungsschemas festgelegt werden. Im Falle der NONSSO-Bereitstellung werden alle Anforderungen mit dem **Basic** Auth-Header mit Base64 codierten Benutzernamen und Kennwort übertragen. Alle Anfragen in diesem Fall werden mit der lokalen Finesse-Datenbank validiert.

Erläuterung des grundlegenden Interaktionsmodells für den SSO-Modus

Dieses *Bild* zeigt das grundlegende Modell der Interaktion zwischen einem Drittanbieter-Gadget, Finesse, IDS und einem REST-Service eines Drittanbieters, wenn sich das System im SSO-Modus befindet.



Bild

Hier ist die Beschreibung für jeden Schritt im Bild angezeigt.

1. Agent/Supervisor greift auf die Finesse Desktop-URL zu. (Beispiel: *https://finesse.com:8445/desktop*)
2. Finesse erkennt, dass der Authentifizierungsmodus SSO ist, und leitet den Browser an IDS um.
3. Der Browser sendet die Anfrage zur Weiterleitung an IDS. Zu diesem Zeitpunkt ermittelt IDS, ob *der Benutzer* über ein gültiges Zugriffstoken verfügt oder nicht. Wenn *der Benutzer* über kein gültiges Zugriffstoken verfügt, leitet IDS das Problem an den Identitätsanbieter (IdP) weiter.
4. Wenn die Anforderung an IdP umgeleitet wird, stellt IdP die *Anmeldeseite* zur Authentifizierung *des Benutzers bereit.*
5. Die SAML-Assertion von IdP wird an das IDS gesendet, das zurück zum Finesse-Desktop geleitet wird.
6. Browser erstellt ein GET der Finesse Desktop-Seite.
7. Finesse erhält das Zugriffstoken von IDS mit dem SAML-Authentifizierungscode.
8. Desktop erhält das Zugriffstoken, das zur Authentifizierung nachfolgender REST-APIs verwendet wird.
9. Gadget von Drittanbietern wird in den Desktop geladen, und es wird eine REST-API von Drittanbietern mit dem Zugriffstoken (Träger) im Authentifizierungs-Header aufgerufen.
10. Der REST-Service eines Drittanbieters validiert das Token mit IDS.
11. Die REST-Antwort eines Drittanbieters wird an das Gadget zurückgegeben.

Konfiguration von gadgets.io.makerequest für SSO- und NONSSO-Modus

Schritt 1: Bei Finesse REST API-Aufrufen über Shindig müssen Gadgets den Autorisierungs-Header "Bearer" in den gadgets.io.makeRequest-Headern hinzufügen.

Schritt 2: Gadgets müssen systemeigene gadgets.io.makeRequest-Aufrufe für alle REST-Anfragen durchführen. Der Autorisierungs-Header muss innerhalb der Anforderungsparams festgelegt werden.

Bei NON SSO-Bereitstellungen ist dies der Auth-Header.

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

Bei SSO-Bereitstellungen ist dies der Auth-Header.

```
"Bearer "  + access_token
```

Zugriffstoken können aus dem **finesse.gadget.Config-**Objekt abgerufen werden.

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

Der neue Autorisierungs-Header muss den Anforderungsparams hinzugefügt werden.

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

Schritt 3: Eine Dienstprogrammmethode **getAuthHeaderString** wurde in **Utilities.Utilities** hinzugefügt. Diese Dienstprogrammmethode verwendet das config-Objekt als Argument und gibt die Zeichenfolge des Autorisierungs-Headers zurück. Gadgets können diese Dienstprogrammmethode verwenden, um den Autorisierungs-Header in Anforderungsparams festzulegen.

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

> **Hinweis**: Für API-Anfragen an Webservices von Drittanbietern muss der Authentifizierungsheader auf dem vom Webdienst des Drittanbieters implementierten Authentifizierungsschema basieren. Gadget-Entwickler haben die Freiheit, die auf der Standardauthentifizierung oder der Authentifizierung mit Inhabertoken basierende Authentifizierung oder andere Authentifizierungsmechanismen ihrer Wahl zu verwenden.