SSOモードでのサードパーティガジェットとFinesseの統合

内容

概要

このドキュメントでは、システムがシングルサインオン(SSO)モードの場合に、Finesseと3rd partyガジェットを統合するために必要な内容について説明します。NON SSOモードの例も示します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Cisco Finesse
- SSO
- Finesseサードパーティガジェット

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- Cisco Finesseバージョン11.6
- SSO
- 3サードパーティガジェット
- サードパーティRESTサービス。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

これらは、エージェントがログインを試行し、SSOまたはNONSSOで認証を行う際の最初の手順です。

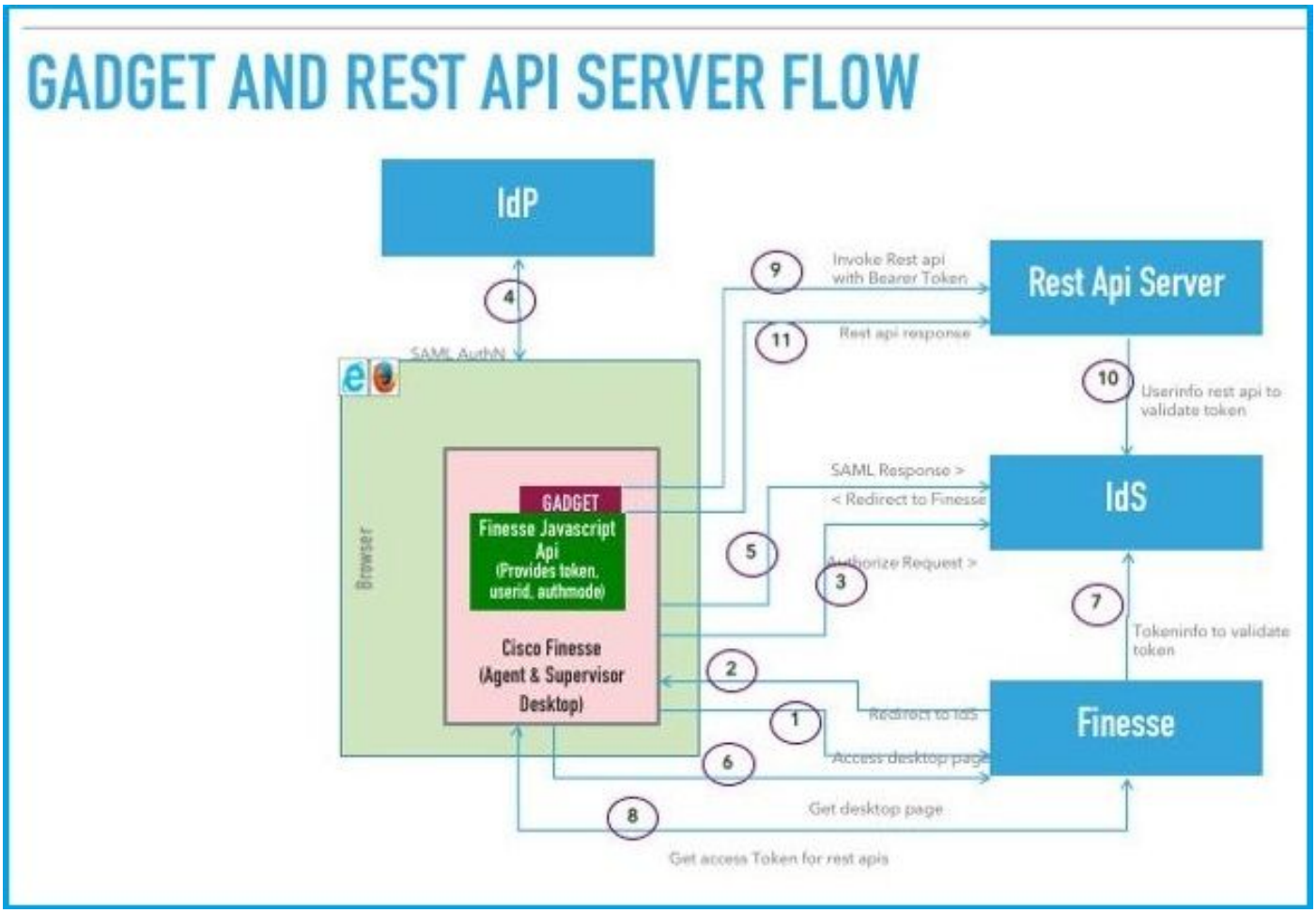
2番目の手順では、SSOおよびNONSSOの場合に認証に成功した後に考慮する必要がある項目について説明します。

1. デスクトップログイン時に、Finesseはシステム認証モード(SSO/NONSSO)を検出し、認証モードに基づいて適切なログインページを表示します。SSOモードの場合はIDPログインページ、NONSSOモードの場合はFinesseログインページが表示されます。

2. 認証に成功すると、すべての要求はシステム認証モードに基づいて認証されます。SSO展開では、Finesseへのすべての要求が要求ヘッダーの一部としてアクセストークンを伝送します。トークンは、認証に成功するためにIDPサーバに対して検証されます。ただし、サードパーティWebサービスへの要求の場合、認証ヘッダーはサードパーティWebサービスによって実装された認証方式に基づいて設定する必要があります。 NONSSO展開の場合、すべての要求はBase64でエンコードされたユーザ名とパスワードを**使用**して、基本認証ヘッダーを伝送します。この場合、すべての要求はFinesseローカルデータベースに対して検証されます。

SSOモードの基本的なインタラクションモデルの説明

この*図*は、システムがSSOモードの場合の、サードパーティガジェット、Finesse、IDS、およびサードパーティのRESTサービス間の基本的なインタラクションモデルを示しています。



画像

図に示す各ステップの説明を次に示します。

1. エージェント/スーパーバイザがFinesseデスクトップURLにアクセスします(例：*https://finesse.com:8445/desktop*)。
2. Finesseは認証モードがSSOであることを検出し、ブラウザをIDSにリダイレクトします。
3. ブラウザがリダイレクト許可要求をIDSに送信します。この時点で、IDSはユーザに有効なアクセストークンがあるかどうかを検出します。有効なアクセストークンがユーザーにない場合、IDSはアイデンティティプロバイダー(IdP)にリダイレクトします。
4. 要求がIdPにリダイレクトされると、IdPはユーザーを認証するための*ログインページを提供します。*
5. IdPからのSAMLアサーションがIDSに送信され、Finesseデスクトップにリダイレクトされます。
6. ブラウザがFinesseデスクトップページのGETを実行します。
7. Finesseは、SAML認証コードを使用してIDSからアクセストークンを取得します。
8. デスクトップは、後続のREST APIの認証に使用されるアクセストークンを取得します。
9. サードパーティのガジェットがデスクトップにロードされ、認証ヘッダーにアクセストークン（ベアラ）を含むサードパーティのREST APIが呼び出されます。
10. サードパーティのRESTサービスがトークンをIDSで検証します。
11. サードパーティのREST応答がガジェットに返されます。

gadgets.io.makerequestのSSOモードおよびNONSSOモードの設定

ステップ1:Finesse REST API呼び出しがShindigを介して行われる場合、ガジェットはgadgets.io.makeRequestヘッダーに「Bearer」認証ヘッダーを追加する必要があります。

ステップ2：ガジェットでは、すべてのREST要求に対してネイティブのgadgets.io.makeRequestコールを行う必要があります。要求パラメータ内に許可ヘッダーを設定する必要があります。

SSO以外の展開では、これは認証ヘッダーです。

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

SSO展開では、これは認証ヘッダーです。

```
"Bearer "  + access_token
```

アクセストークンは、**finesse.gadget.Configオブジェクトから取得**できます。

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

新しい認可ヘッダーを要求パラメータに追加する必要があります。

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

ステップ3：ユーティリティメソッド**getAuthHeaderString**がutilities.Utilities内に追加**されました。**

このユーティリティメソッドは、configオブジェクトを引数として取り、許可ヘッダー文字列を返します。ガジェットでは、このユーティリティメソッドを使用して、要求パラメータに許可ヘッダーを設定できます。

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

> **注**：サードパーティWebサービスへのAPI要求の場合、認証ヘッダーはサードパーティWebサービスによって実装された認証方式に基づいて設定する必要があります。ガジェット開発者は、基本的な認証またはベアラートークンベースの認証、またはその他の任意の認証メカニズムを自由に使用できます。