

在SSO模式下使用Finesse整合第三方小工具

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[背景資訊](#)

[SSO模式互動基本模型的解釋](#)

[為SSO和NONSSO模式配置gadgets.io.makerequest](#)

簡介

本文檔介紹當系統處於單點登入(SSO)模式時，將第三方小工具與Finesse整合所需的內容。並給出了非SSO模式的示例。

必要條件

需求

思科建議您瞭解以下主題：

- Cisco Finesse
- SSO
- Finesse第三方小工具

採用元件

本文中的資訊係根據以下軟體和硬體版本：

- Cisco Finesse版本11.6
- SSO
- 第三方小工具
- 第三方REST服務。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

背景資訊

以下是代理嘗試登入並使用SSO或NONSSO進行身份驗證時的初始步驟。

第二步描述了在SSO和NONSSO的情況下成功身份驗證後需要考慮的事項。

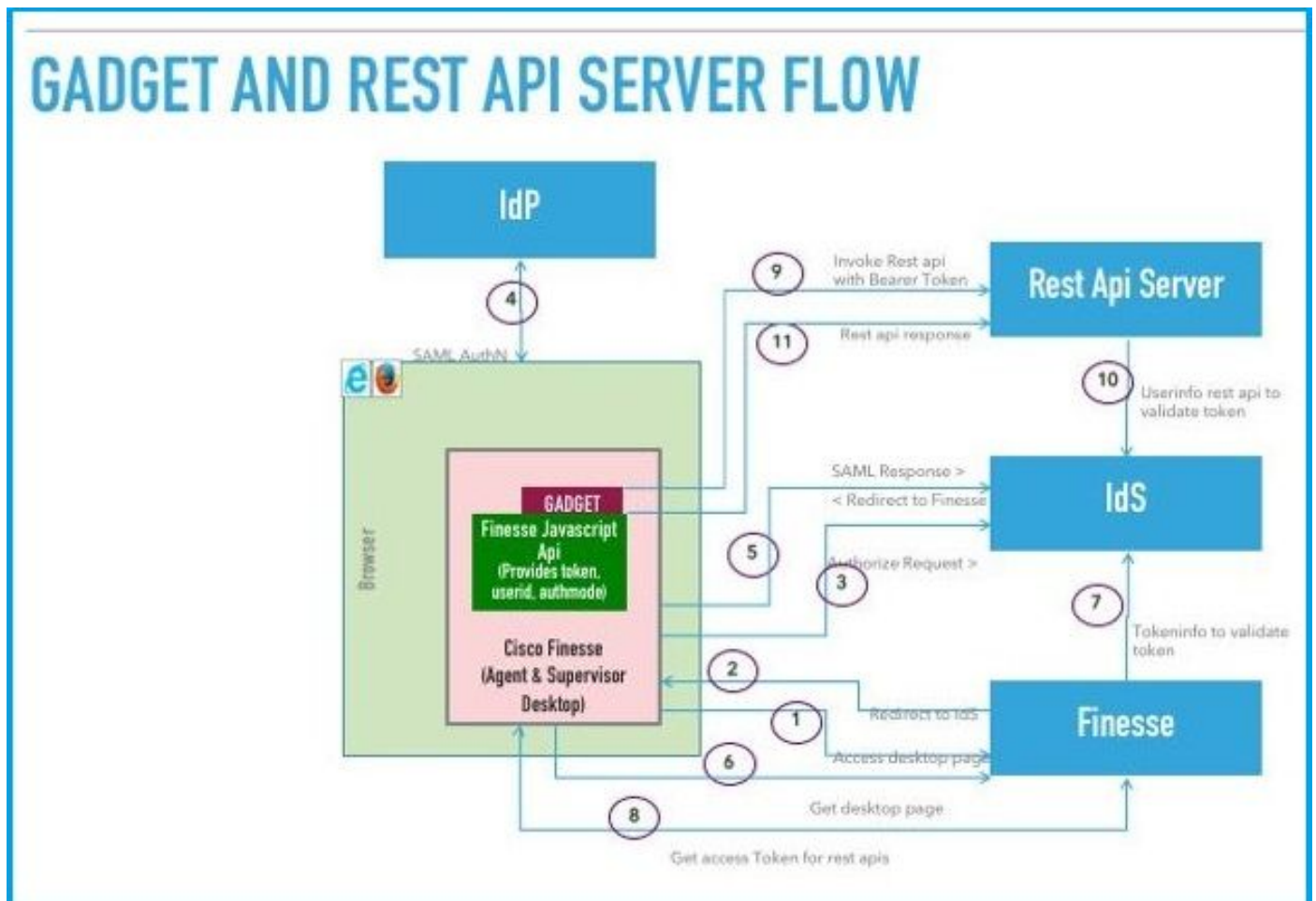
1. 案頭登入時，Finesse檢測系統身份驗證模式(SSO/NONSSO)，並根據身份驗證模式顯示相應

的登入頁面。使用者在SSO模式下會看到IDP登入頁面，在NONSSO模式下會看到Finesse登入頁面。

2. 身份驗證成功後，所有請求都將根據系統身份驗證模式進行身份驗證。對於SSO部署，對Finesse的所有請求都將訪問令牌作為請求標頭的一部分。將針對IDP伺服器驗證令牌以成功進行身份驗證。但是，對於請求第三方Web服務，必須根據第三方Web服務實施的身份驗證方案設定身份驗證標頭。在NONSSO部署中，所有請求都帶有採用base64編碼的使用者名稱和密碼的Basic Auth標頭。此案例中的所有請求都將根據Finesse本地資料庫進行驗證。

SSO模式互動基本模型的解釋

此圖顯示當系統處於SSO模式時，第三方小工具、Finesse、IDS和第三方REST服務之間互動的基本模型。



影象

以下是圖中顯示的每個步驟的說明。

1. 代理/Supervisor訪問Finesse案頭URL。(示例：<https://finesse.com:8445/desktop>)
2. Finesse檢測到身份驗證模式為SSO，並將瀏覽器重定向到IDS。
3. 瀏覽器將重定向授權請求傳送到IDS。此時，IDS會檢測用戶是否具有有效的訪問令牌。如果使用者沒有有效的訪問令牌，則IDS會重定向到身份提供程式(IdP)。
4. 如果請求被重定向到IdP，則IdP會提供登入頁面以對使用者進行身份驗證。
5. 來自IdP的SAML斷言被傳送到IDS，IDS重定向回Finesse案頭。
6. 瀏覽器執行Finesse案頭頁面的GET。

7. Finesse從具有SAML身份驗證代碼的IDS獲取訪問令牌。
8. Desktop獲取用於驗證後續REST API的訪問令牌。
9. 第三方小工具載入到案頭，並呼叫第三方REST API，在auth-header中使用訪問令牌（持有人）。
10. 第三方REST服務使用IDS驗證令牌。
11. 第三方REST響應將返回到小工具。

為SSO和NONSSO模式配置gadgets.io.makerequest

步驟1.對於通過Shindig進行的Finesse REST API呼叫，小工具需要在gadgets.io.makeRequest標頭中新增「持有人」授權標頭。

步驟2.小工具需要為所有REST請求進行本地gadgets.io.makeRequest呼叫，必須在請求引數中設定授權標頭。

對於非SSO部署，這是身份驗證標頭。

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

對於SSO部署，這是身份驗證標頭。

```
"Bearer " + access_token
```

可以從finesse.gadget.Config對象檢索訪問令牌。

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

必須將新的授權標頭新增到請求引數中。

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

```
params[gadgets.io.RequestParameters.HEADERS].Authorization = "Bearer " + access_token;
```

步驟3.實用程式方法getAuthHeaderString已新增到utilities.Utilities中。此實用程式方法將config對象作為引數並返回授權頭字串。小工具可以使用此實用程式方法在請求引數中設定授權標頭。

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

附註：對於對第三方Web服務的API請求，必須根據第三方Web服務實施的身份驗證方案設定身份驗證標頭。小工具開發人員可以自由使用基於基本身份驗證或承載令牌的的身份驗證，或者他們選擇的任何其他身份驗證機制。