SSO 모드에서 Finesse와 타사 가젯 통합

목차

소개

이 문서에서는 시스템이 SSO(Single Sign-on) 모드에 있는 동안 Finesse와 3rd Party 가젯을 통합하는 데 필요한 사항에 대해 설명합니다.NON SSO 모드에 대한 예도 제공됩니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- Cisco Finesse
- SSO
- Finesse 서드파티 가젯

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- Cisco Finesse 버전 11.6
- SSO
- 3rd Party 가젯
- 타사 REST 서비스

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다.이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다.네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 이해해야 합니다.

배경 정보

에이전트가 SSO 또는 NONSSO에 로그인하고 인증하려고 시도하는 초기 단계입니다.

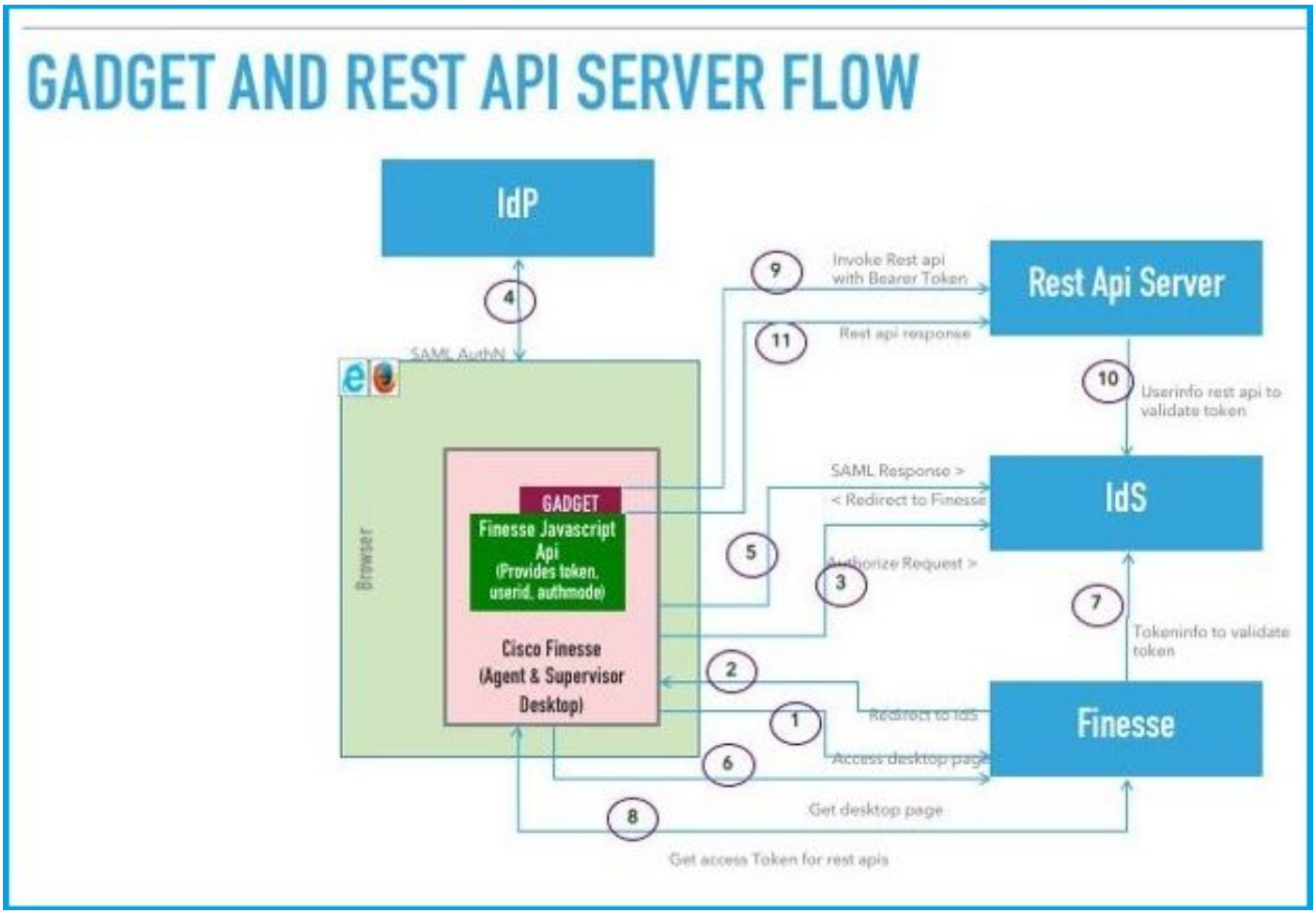
두 번째 단계에서는 SSO 및 NONSSO의 경우 인증에 성공한 후 고려해야 할 사항을 설명합니다.

1. 데스크톱 로그인 시 Finesse는 시스템 인증 모드(SSO/NONSSO)를 탐지하고 인증 모드에 따라 적절한 로그인 페이지가 표시됩니다.사용자는 SSO 모드의 경우 IDP Login(IDP 로그인) 페이지를, NONSSO 모드의 경우 Finesse Login(Finesse 로그인) 페이지를 볼 수 있습니다.

2. 인증에 성공하면 모든 요청이 시스템 인증 모드에 따라 인증됩니다.SSO 구축의 경우 Finesse에 대한 모든 요청은 요청 헤더의 일부로 액세스 토큰을 전달합니다.성공적인 인증을 위해 IDP 서버에 대해 토큰이 검증되었습니다.그러나 서드파티 웹 서비스에 대한 요청의 경우 서드파티 웹 서비스에서 구현한 인증 체계에 따라 인증 헤더를 설정해야 합니다. NONSSO 구축의 경우 모든 요청은 base64 인코딩 사용자 이름 및 비밀번호와 함께 **Basic** Auth 헤더를 전달합니다.이 경우 모든 요청은 Finesse 로컬 데이터베이스에 대해 검증됩니다.

SSO 모드의 상호 작용 기본 모델 설명

이 *이미지*는 시스템이 SSO 모드에 있을 때 서드파티 가젯, Finesse, IDS 및 서드파티 REST 서비스 간의 상호 작용 기본 모델을 보여줍니다.



이미지

다음은 이미지에 표시된 모든 단계에 대한 설명입니다.

1. 에이전트/수퍼바이저가 Finesse 데스크톱 URL에 액세스합니다(예: *https://finesse.com:8445/desktop).*
2. Finesse는 인증 모드가 SSO임을 감지하고 브라우저를 IDS로 리디렉션합니다.
3. 브라우저가 리디렉션 권한 부여 요청을 IDS로 전송합니다.이 시점에서 IDS는 *사용자*에게 유효한 액세스 토큰이 있는지 여부를 탐지합니다.*사용자에게* 유효한 액세스 토큰이 없으면

IDS는 IdP(Identity Provider)로 리디렉션됩니다.
4. 요청이 IdP로 리디렉션되면 IdP는 *사용자*를 인증하기 위한 *로그인* 페이지를 제공합니다.
5. IdP의 SAML assertion은 IDS로 전송되며, 이 ID는 Finesse 데스크톱으로 다시 리디렉션됩니다.
6. 브라우저는 Finesse 데스크톱 페이지의 GET을 수행합니다.
7. Finesse는 SAML 인증 코드를 사용하여 IDS에서 액세스 토큰을 가져옵니다.
8. 데스크톱은 후속 REST API를 인증하는 데 사용할 액세스 토큰을 가져옵니다.
9. 서드파티 가젯이 데스크톱에 로드되고 auth-header에서 액세스 토큰(전달자)을 사용하여 서드파티 REST API를 호출합니다.
10. 타사 REST 서비스는 IDS로 토큰을 검증합니다.
11. 서드파티 REST 응답이 가젯으로 반환됩니다.

SSO 및 NONSSO 모드에 대한 gadgets.io.makerequest 구성

1단계. Shindig를 통해 실행되는 Finesse REST API 호출의 경우 가젯은 gadgets.io.makeRequest 헤더에 "Bearer" 권한 부여 헤더를 추가해야 합니다.

2단계. 가젯은 모든 REST 요청에 대해 네이티브 gadgets.io.makeRequest를 호출해야 합니다. 권한 부여 헤더는 요청 매개변수 내에서 설정해야 합니다.

NON SSO 구축의 경우 인증 헤더입니다.

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

SSO 구축의 경우 인증 헤더입니다.

```
"Bearer "  + access_token
```

액세스 토큰은 finesse.gadget.Config 개체에서 검색할 수 있습니다.

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

새 권한 부여 헤더를 요청 매개 변수에 추가해야 합니다.

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

3단계. 유틸리티 메서드 getAuthHeaderString이 utilities.Utilities 내에 **추가되었습니다**.이 유틸리티 메서드는 config 개체를 인수로 가져와 권한 부여 헤더 문자열을 반환합니다.가젯은 이 유틸리티 메서드를 사용하여 요청 파라미터에서 권한 부여 헤더를 설정할 수 있습니다.

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

> **참고:**서드파티 웹 서비스에 대한 API 요청의 경우 서드파티 웹 서비스에서 구현한 인증 체계에 따라 인증 헤더를 설정해야 합니다.가젯 개발자는 기본 인증 또는 베어러 토큰 기반 인증 또는 선택한 다른 인증 메커니즘을 자유롭게 사용할 수 있습니다.