ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE
Corso di Laurea in Informatica

Progettazione e implementazione di un meccanismo per associare access token OAuth al certificato X.509 di un client

Relatore:
Prof. Ozalp Babaoglu
Correlatore:
Prof. Francesco Giacomini

Candidato:
Ivan De Simone

II Sessione
Anno Accademico 2024/2025

Sommario

Introduzione

Contesto tecnologico

Modifiche al sistema

Test e validazione

Conclusione e sviluppi futuri

Introduzione

Esistono diversi sistemi per garantire l'accesso sicuro a risorse protette all'interno di un'infrastruttura distribuita. Tra questi, uno dei più diffusi è basato sull'utilizzo di access token OAuth, che agiscono come credenziali temporanee.

Un attaccante entrato in possesso di un access token in corso di validità, impersonando un utente legittimo, potrebbe ottenere l'accesso a risorse protette del server.

Questo lavoro si concentra sull'implementazione del meccanismo proposto dal RFC 8705 (*OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens*), dimostrando e valutando la fattibilità tecnica della sua integrazione in un sistema esistente.

Contesto tecnologico - INDIGO IAM

INDIGO IAM è un servizio di Identity and Access Management che implementa il protocollo OAuth 2.0

IAM Login Service:

- Authorization Server → token issuer
- Resource Server → API esposte

Dashboard:

- Client → accesso alle risorse

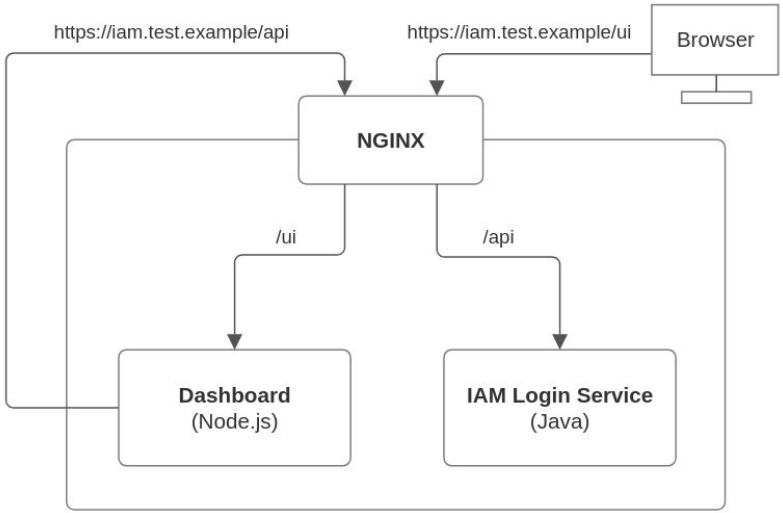


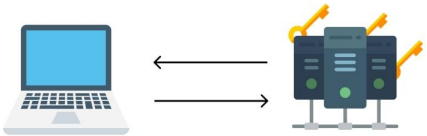
Diagramma dell'architettura attuale

Contesto tecnologico - Tecnologie coinvolte

NGINX

OAuth 2.0

Certificati digitali X.509



Mutual TLS

Modifiche al sistema - Implementazioni effettuate

IAM Login Service:

- inserimento dell'hash del certificato nell'access token
- controllo di corrispondenza

Dashboard:

- flusso OAuth 2.0 manuale
- configurazione Agent

NGINX:

- attivazione client authentication
- inoltro header HTTP

```
{
  "iss": "https://iam.test.example:8443",
  "sub": "73f16d93-2441-4a50-88ff-85360d78c6b5",
  "iat": 1760015939,
  "exp": 1760019539,
  "client_id": "30ce96a7-de5e-4282-bf11-16be13621e45",
  "scope": "openid profile scim:read scim:write",
  "jti": "97de228e-7989-4f24-a2b9-84e448fe0606"
}
```

```
{
  "iss": "https://iam.test.example:8443",
  "sub": "73f16d93-2441-4a50-88ff-85360d78c6b5",
  "iat": 1760015939,
  "exp": 1760019539,
  "client_id": "30ce96a7-de5e-4282-bf11-16be13621e45",
  "scope": "openid profile scim:read scim:write",
  "jti": "97de228e-7989-4f24-a2b9-84e448fe0606",
  "cnf": {
    "x5t#S256": "Qircx-blDMNYi5OhGMe5n23gCiUmy6i-kFJI2iUIe6U"
  }
}
```

Access token con e senza l'hash del certificato X.509 client

Modifiche al sistema - Implementazioni effettuate

IAM Login Service:

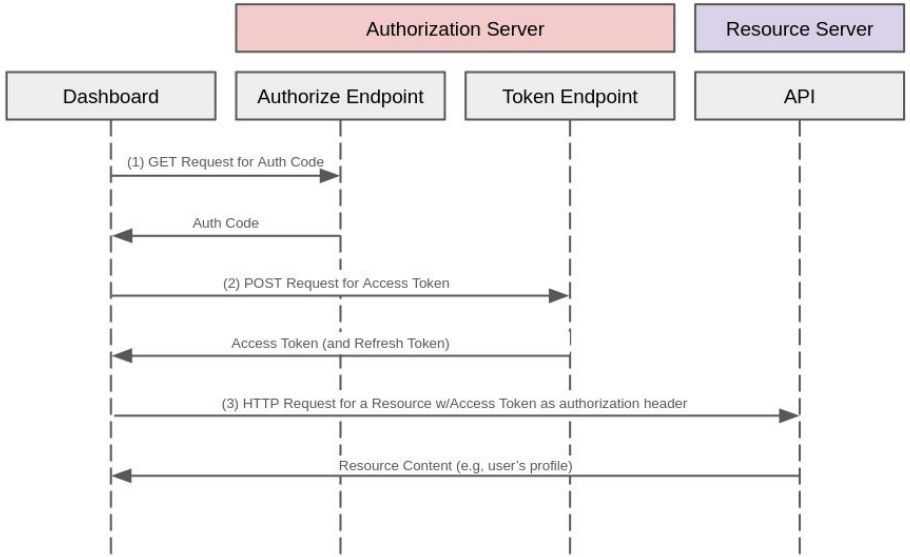
- inserimento dell'hash del certificato nell'access token
- controllo di corrispondenza

Dashboard:

- flusso OAuth 2.0 manuale
- configurazione Agent

NGINX:

- attivazione client authentication
- inoltro header HTTP



Flusso di autenticazione e autorizzazione OAuth 2.0

Modifiche al sistema - Impatto delle modifiche

Interventi circoscritti a componenti specifici, progettati per estendere le funzionalità esistenti, riducendo l'impatto su architettura e codice.

Necessità di lavoro consistente nel caso si utilizzino librerie che non supportano nativamente mutual TLS.

```
$ git diff 30790ac8 bccd985b --stat
 .../security/IamApiSecurityConfig.java       |   2 +
 .../filters/MtlsTokenBindingFilter.java      | 100 ++++++++++++++++++
 .../common/BaseAccessTokenBuilder.java       |  28 +++++
 .../it/infn/mw/iam/util/x509/X509Utils.java  |  17 +++
 4 files changed, 147 insertions(+)
```

Output del comando `git diff --stat` su IAM Login Service

```
$ git diff b8f595b 4a8ab45 --stat
 src/app/api/mtls/callback/route.ts | 16 ++++++
 src/app/mtls/page.tsx              | 36 ++++++++++++
 src/middleware.ts                  |  2 +-
 src/services/oauth-mtls.ts         | 84 ++++++++++++++++++++++++++++++
 src/services/users-mtls.ts         | 28 ++++++++++
 src/utils/fetch-mtls/index.ts      | 67 ++++++++++++++++++++++++
 6 files changed, 232 insertions(+), 1 deletion(-)
```

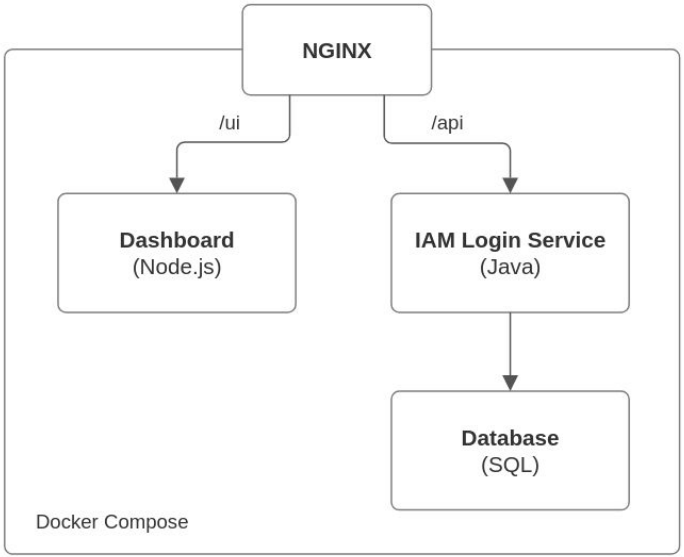
Output del comando `git diff --stat` sulla dashboard

Test e validazione - Ambiente di validazione

Ambiente di esecuzione isolato, basato su Docker Compose

Servizi:

- IAM Login Service
- Dashboard
- NGINX
- Database



Schema dell'ambiente di validazione

Test e validazione - Test effettuati

TEST	OPERAZIONI	RISULTATO
Accesso alle risorse	1. connessione con certificato valido 2. richiesta risorsa con stesso certificato valido	200 - OK
Certificato non corrispondente	1. connessione con certificato valido A 2. richiesta risorsa con certificato valido B	401 - Unauthorized
Certificato scaduto durante l'autenticazione	1. tentativo di connessione con certificato scaduto	495 - SSL Certificate Error
Certificato scaduto durante la richiesta di risorse	1. connessione con certificato valido A 2. richiesta risorsa con certificato scaduto A	495 - SSL Certificate Error
Certificato assente durante l'autenticazione	1. tentativo di connessione senza certificato	200 - OK
Certificato assente durante la richiesta di risorse	1. connessione con certificato valido 2. richiesta risorsa senza certificato	401 - Unauthorized

Conclusione e sviluppi futuri

Questo lavoro ha mostrato che l'implementazione delle specifiche introdotte dal RFC 8705 è possibile con modifiche limitate e non invasive. I test di validazione hanno confermato il corretto funzionamento del meccanismo di associazione tra access token e certificato X.509 client.

Ulteriori sviluppi riguarderanno l'integrazione del protocollo mutual TLS nelle librerie pubbliche comunemente utilizzate, per favorirne l'adozione nei sistemi in produzione. Un'evoluzione naturale del lavoro consiste inoltre nell'estendere l'implementazione ai flussi Token Exchange e Refresh Token, previsti dalle specifiche OAuth 2.0.

Grazie

Crediti

Logo *NGINX* da https://it.wikipedia.org/wiki/Nginx#/media/File:Nginx_logo.svg

Logo *OAuth 2.0* da https://oauth.net/2/

Figura *X.509* da https://www.sectigo.com/enterprise-solutions/certificate-manager/integrations-x509

Figura *mutual TLS* da https://www.wallarm.com/what/mutual-authentication

Figura *flusso OAuth 2.0* da slide INFN