Umbria Digitale

FedERa – Integrazione

Liferay 6.1

Liferay 6.2

Liferay 7

Spring security saml

Shibboleth

Sommario

1. INTRODUZIONE

Questo documento ha lo scopo di illustrare le linee guida per l'integrazione di un'applicazione basata su Spring Framework con l'infrastruttura FedERa, per proteggere, mediante autenticazione, l'accesso ai servizi erogati.

L'integrazione presentata in questo documento si basa sull'utilizzo del framework OpenSAML 2 attraverso Spring Security.
Nei paragrafi seguenti vengono presentate e descritte le corrette modalità per la gestione della fase di auto login e di logout federato.

1.1. CONTENUTI DEL DOCUMENTO

Il capitolo 2 illustra brevemente gli elementi costituitivi dell'architettura FedERa, evidenziando le interfacce tra essi. La descrizione ha lo scopo di fornire una visione d'insieme introduttiva propedeutica alla comprensione del processo di integrazione dei servizi.

Il capitolo 3 illustra le fasi dell'integrazione di un'applicazione basata su Spring Framework tramite l'utilizzo delle librerie Open Saml 2.
In tale capitolo sono riportate le indicazioni per la parametrizzazione dell'integrazione e i dettagli sulla modalità di installazione.

1.2. ACRONIMI

Il seguente elenco riporta gli acronimi utilizzati nel documento:

SIGLA	**DEFINIZIONE**
SP	Service Provider
IDM	Identity Manager, sistema di gestione delle identità
IDP	Identity Provider, sistema di autenticazione
GW	Gateway Multiprotocolo

2. L'ARCHITETTURA DELL'INFRASTRUTTURA FEDERA

La caratteristica principale dell'infrastruttura FedERa consiste nella gestione della fase di autenticazione all'accesso ai servizi erogati da Service Provider integrati con l'infrastruttura complessiva. L'autenticazione degli utenti può avvenire sia presso un Identity Provider registrato presso l'infrastruttura FedERa, allo stato attuale FedUmbria, oppure presso uno qualunque degli Identity Provider SPID.

In generale risulta integrabile qualunque Service Provider rispetti le direttive del protocollo SAML.
FedERa consente la convivenza di IdP e SP funzionanti sia con protocollo SAML 1.1 che 2.0, essendo in grado di adattarsi ai vari protocolli impiegati presso di essi.

2.1. ARCHITETTURA COMPLESSIVA DI ALTO LIVELLO

L'architettura del sistema FedERa è costituita da tre macrocomponenti principali:

- Sistema di autenticazione (Identity Provider): eroga il servizio di autenticazione per gli utenti FedERa verificando le credenziali utente e preparando le asserzioni SAML (1.1 o 2.0) che dovranno essere trasmesse agli erogatori dei servizi finali; allo stato attuale l'IdP che svolge questa funzione è un IdP esterno – FedUmbria – registrato presso l'infrastruttura FedERa e integrato con protocollo SAML 2.0

- Gateway Multiprotocollo di autenticazione: svolge la funzione di gateway di dominio e di mediatore fra i servizi di front-end erogati dagli enti sul territorio che richiedono l'autenticazione e gli Identity Provider della federazione (quindi non solo l'Identity Provider Federa). Il Gateway Multiprotocollo è responsabile di supportare le interazioni con i Service Provider ed Identity Provider della federazione utilizzando entrambi i protocolli SAML 1.1 e SAML 2.0.

- Sistema di gestione di identità digitali (Identity Manager): applicazione per la gestione degli utenti e del relativo ciclo di vita (es. registrazione, attivazione, sospensione, ecc.) con Funzionalità per varie tipologie di soggetti (amministratori, utenti finali, ecc.).

Per poter interagire con FedERa, i Service Provider con essa integrati devono poter comunicare con il componente Gateway Multiprotocollo che svolge il ruolo di mediatore nella fase di accesso ai servizi, ricevendo i messaggi di richiesta di autenticazione, interpellando l'utente su quale Identity Provider deve essere contattato per verificare le credenziali e produrre la relativa attestazione firmata e infine rimandando l'utente al servizio richiesto. In tale fase, il Gateway ha il compito di svolgere tutte le attività di adattamento tra messaggi scambiati dai soggetti interagenti (Service Provider da un lato e Identity Provider dall'altro), in modo trasparente per l'utente e per i Service Provider, sia relativamente alla struttura di tali messaggi, legata al protocollo impiegato, sia relativamente al contenuto.

2.2. ARCHITETTURA DI UN SERVICE PROVIDER

Un Service Provider da integrare con FedERa deve essere dotato di alcuni sotto-componenti aventi lo scopo di instaurare la comunicazione su protocollo SAML 2.0 con il resto dell'infrastruttura, per l'invio di richieste di autenticazione e la ricezione delle relative risposte.

Di seguito una breve descrizione di tali componenti:

- ***Request Handler*** – riceve una richiesta di una risorsa protetta ed avvia (se necessario) il processo di autenticazione e autorizzazione. Una volta terminati tali processi, tale componente ha l'ulteriore compito di invocare il servizio applicativo vero e proprio, che viene erogato all'utente.
- ***AssertionConsumerService*** – riceve i messaggi SAML Response inviati mediante il binding HTTP-POST e autentica l'utente per l'accesso ai servizi. Tipicamente, in risposta ad una richiesta di autenticazione viene prodotto un messaggio SAML Response con uno statement contenente il risultato dell'autenticazione stessa ed eventuali statement di attributo.
- ***Servizio Applicativo*** – è il servizio applicativo vero e proprio che esula dagli scopi di questo documento. E' costituito dalle classi e dalle pagine che costituiscono la logica applicativa del Service Provider.
- ***SP Configuration Loader*** – componente che accede al file di configurazione del SP al fine di reperire informazioni quali i metodi di autenticazione accettati e i vari prefissi degli URL dei servizi esposti.
- ***Metadata Provider*** – componente che gestisce l'accesso ai metadati e le relative richieste provenienti dall'esterno. Ha il compito di inoltrare le richieste dei file dei metadati alle

rispettive entità, e di rispondere inviando i metadati del componente su cui è dispiegato, quando viene contattato.
- ***Signature Manager*** – componente che gestisce le operazioni di firma di oggetti SAML e di verifica della firma.

2.3. SCENARIO DI INTERAZIONE GENERALE

Lo scenario d'interazione complessivo che riguarda i messaggi scambiati dal Service Provider con il resto dell'infrastruttura è di seguito dettagliato:

1. L'utente richiede l'accesso ad un servizio erogato da un sistema informativo regionale
2. Il sistema informativo è protetto mediante il filtro SP Request Handler, perciò la richieste viene intercettata dal tale filtro al quale arriva
3. Il filtro controlla se nella sessione applicativa sono presenti i dati relativi ad una pregressa autenticazione per l'accesso allo stesso servizio e se essi risultano ancora validi. In caso negativo viene attivato il processo di autenticazione. Nel seguito si assume questa seconda condizione. Nel caso invece l'utente risulti già autenticato si procede direttamente allo step 11.
4. Il filtro legge dalla configurazione del Service Provider i dati relativi al servizio richiesto. In particolare viene letto un file di configurazione (cfr. sez. 3.8) in cui sono specificate le varie modalità di autenticazione (es. username e password oppure smart-card) che il SP dovrà richiedere all'infrastruttura FedERa in fase di autenticazione utente.
5. Il filtro Request Handler prepara un messaggio di tipo "SAML AuthnRequest" firmato e conforme alla specifica SAML 2.0 e lo invia al Gateway Multiprotocollo. A seguito di tale messaggio inizia il processo che porta l'utente ad interagire prima con il Gateway e quindi con l'Identity Provider selezionato per effettuare l'autenticazione.
6. Ad autenticazione avvenuta (o fallita) il Gateway Multiprotocollo risponde al messaggio inviato dal Service Provider con un secondo messaggio di tipo "SAML Response" che invia via POST al componente Assertion Consumer Service. Tale POST transita per lo useragent mediante un form web auto-postante.
7. Il componente Assertion Consumer Service legge il messaggio arrivato dal Gateway Multiprotocollo, ne verifica la firma e registra nella sessione applicativa i dati relativi all'autenticazione effettuata, che potranno essere recuperati durante gli accessi successivi al medesimo servizio da parte del filtro Request Handler.
8. Il componente Assertion Conusmer Service invia un messaggio di HTTP-Redirect allo user-agent per inviarlo alla pagina del servizio originariamente richiesto.
9. Come allo step 2, la richiesta di accesso al servizio viene intercettata nuovamente dal filtro Request Handler, il quale è ora in grado di verificare la presenza del necessario contesto di autenticazione nella sessione applicativa.
10. Il filtro consente allo user-agent l'accesso al servizio richiesto

2.4. GESTIONE DELLA LOGOUT FEDERATA

Sebbene l'infrastruttura FedERa, come già detto, implementa per il colloquio con i SP e gli IdP il protocollo SAML, la fase di logout in FedERa non è SAML compliant. La fase di logout, quindi, va gestita in maniera differente, in particolare, sia FedUmbria che FedERa espongono un URL di logout che va invocato con appositi parametri. L'integrazione sviluppata da Umbria Digitale implementa questo meccanismo di logout ed è in grado di eseguire una logout federata indipendentemente dal fatto che l'utente si sia autenticato tramite IdP FedUmbria (nel caso di accesso con user/password) o tramite IdP Federa (autenticazione con CNS).

Va inoltre evidenziato che nell'attuale build di FedERa non è ancora stato implementato un meccanismo di logout per autenticazione utente tramite IdP SPID.

3. INTEGRAZIONE DI UN'APPLICAZIONE MEDIANTE L'UTILIZZO DI SPRING SECURITY SAML

In questo capitolo sono descritte le attività necessarie per integrare un'applicazione web con l'infrastruttura FedERa, facendo uso delle librerie OpenSAML e di Spring Security. Verrà dapprima descritta la soluzione scelta per l'integrazione con l'infrastruttura. Successivamente saranno elencate le attività di integrazione da eseguire ed infine saranno fornite indicazioni utili per la parametrizzazione dell'integrazione.

3.1. LA SOLUZIONE SCELTA PER L'INTEGRAZIONE

L'integrazione, come già detto, è basata sull'utilizzo delle librerie OpenSAML in abbinamento al framework Spring Security.

La configurazione di Spring Security viene fatta in un file xml, denominato security.xml, ed è parametrizzata in un file di properties per una più semplice gestione delle variabili di parametrizzazione.

Inoltre sono state implementate/estese le classi specifiche per la gestione della fase di costruzione delle informazioni di autenticazione necessarie al corretto funzionamento del framework Spring Security a partire dall'asserzione ritornata, a valle del processo di autenticazione, da Federa.

Il pacchetto di integrazione è così composto:

1. Filtro Spring Security configurato nel web.xml per la cattura e gestione di tutte le richieste di accesso alle pagine dell'applicazione.
2. File security.xml che stabilisce la filter-chain per ciascuno degli URL intercettati dal filtro di cui al punto precedente.
3. File authentication.properties, contiene la personalizzazione di alcune proprietà dell'integrazione. In particolare:
    1. authnContexts, indica la modalità di autenticazione che il SP richiede al gateway.
    2. entityBaseURL, indica appunto l'entityBaseURL del file dei metadati generato automaticamente da OpenSAML
    3. federa.gateway.metadata, url dei metadati di Federa
    4. federa.federaIdp, è valorizzato con l'identificativo dell'idp Fed configurato in Federa
    5. federa.idpLogoutURL, è valorizzato con l'url per la logout a FED
    6. "auto.login.hooks" indica la classe di auto login, che a valle dei filtri spring security si occupa di fare l'eventuale provisioning nel DB di liferay per i nuovi utenti e di loggare gli utenti al portale
4. CustomSAMLUserDetailService, è l'implementazione dell'interfaccia SAMLUserDetailsService del framework Spring per la fase di creazione dell'oggetto utente contenente le informazioni relative all'utente loggato.
5. CustomSAMLLogoutHandler.java, per la gestione del logout federato.

3.2. PROCEDURA DI INTEGRAZIONE

L'integrazione di una web application basata su Spring Framework con l'infrastruttura FedERa consiste nell'esecuzione di una serie di passi che portano a proteggere con autenticazione l'accesso ad alcune risorse. In particolare, devono essere svolte le seguenti macro-attività:

- Configurazione della JVM utilizzata per l'avvio dell'application server in modo da utilizzare un truststore nel quale sia posto il certificato X.509 utilizzato dal Gateway Multiprotocollo FedERa per instaurare le connessioni SSL;

- Modifica del file authentication.properties; Modificare questo file in funzione del metodo di autenticazione scelto e dell'ambiente Federa che si vuole integrare (sviluppo, test, produzione). Inoltre in questo file viene definito l'entityBaseURL che verrà poi esposto nel metadata.xml

- Aggiunta delle classi per la gestione delle fasi di autenticazione dell'utente e di gestione della logout. Includere nel classpath dell'applicazione il package it.umbriadigitale presente nel pacchetto di integrazione fornito a corredo del presente documento.

- Aggiunta nel classpath della cartella security e del file samlKeystore.jks presenti nel pacchetto di integrazione fornito col presente documento.

- Creazione/modifica del security.xml. Se necessario può essere utilizzato direttamente il file contenuto nel pacchetto di integrazione fornito col presente documento.

- Configurazione del web.xml; le modifiche da apportare al web.xml sono le seguenti:

    - aggiungere al parametro di contesto contextConfigLocation il riferimento al file security.xml
    - aggiungere il filtro Spring Security e relativo mapping

- Copia dei jar necessari al funzionamento dell'integrazione. Copiare, privilegiando per le librerie dello stesso tipo quelle già presenti nella web application, tutti i jar contenuti nella cartella WEB-INF/lib del pacchetto di integrazione fornito col presente documento.