

fuss-manager - Segnalazione #915

Segnalazione # 789 (Chiuso): Integrazione in octonet

Passare l'autenticazione da octonet a fuss-manager

10/23/2019 11:51 AM - Elena Grandi

Status:	Chiuso	Start date:	10/23/2019
Priority:	Normale	Due date:	
Assignee:	Enrico Zini	% Done:	10%
Category:		Estimated time:	0.00 hour
Target version:	0.7 Autenticazione e autorizzazione		
Resolution:			
Description			

Associated revisions

Revision e4a345c3 - 10/28/2019 12:13 PM - Enrico Zini

Make use of django.core.signing without access to django settings module. refs: #915

Revision 577ae3c0 - 10/28/2019 03:13 PM - Enrico Zini

Implemented auth forward view. refs: #915

Revision 46cf6b3e - 11/15/2019 11:34 AM - Enrico Zini

In mock mode, use a auth_forward_secret. refs: #915

History

#1 - 10/23/2019 12:09 PM - Enrico Zini

Possiamo passare l'autenticazione usando la struttura User di fuss-manager encodata in un HMAC firmato da Django. Entrambi i siti devono condividere una secret key (diversa dalla secret key del sito) apposta per il passaggio dell'autenticazione.

Lato octonet, si può usare django.core.signing.dumps (<https://docs.djangoproject.com/en/2.2/topics/signing/>) con la key custom.

Lato fuss-manager server reimplementare django.core.signing.loads (prendendo i pezzi che servono, se servono, dai sorgenti di django, ma comunque in buona parte è tutto basato sul modulo python hmac <https://docs.python.org/3.5/library/hmac.html>)

Fuss-manager può ricevere questo token in una view di login alternativa, autenticare e decodificare il token, e inserire la struttura User risultante direttamente nella session senza avere bisogno nemmeno di toccare ldap.

Questo sistema di autenticazione in fuss-manager è disabilitato per default, e si abilita se è presente il secret di passaggio autenticazione nella configurazione.

In fase di deploy, sarà fuss-server a impostare questo valore, per non averlo abilitato per default fuori da fuss.

#2 - 10/23/2019 12:46 PM - Enrico Zini

- Assignee set to Enrico Zini

#3 - 10/28/2019 03:28 PM - Enrico Zini

- Status changed from Nuovo to In elaborazione
- Assignee changed from Enrico Zini to Mark Caglienzi
- % Done changed from 0 to 10

Ho implementato una view di forwarding dell'autenticazione.

Per usarlo, si usa `django.core.signing.dumps` per codificare e firmare una struttura che contiene i dati dell'utente e l'url a cui ridirigere dopo l'autenticazione.

Per esempio:

```
from django.core import signing

token = signing.dumps({
    "url": "/test",
    "user": {
        "name": "test",
        "id": 2,
        "display_name": "Test User",
        "groups": [{"name": "test", "id": 1}, {"name": "users", "id": 2}],
    },
}, key=self._app.config.auth_forward_secret, salt="fuss_manager.auth")

# Visit "/auth-forward" + "?t=" + token
```

Ora rimane da provare il lato octonet: creare un iframe, generare il token in base ai dati dell'utente corrente, e aprire l'iframe sulla view di `auth-forward`.

Mark, ti passo la palla, e rimango a disposizione per aiuto se serve

#4 - 10/28/2019 03:29 PM - Enrico Zini

Nota, il secret usato qui è diverso dal secret di django: serve aggiungere un nuovo secret nei setting da condividere con `fuss-manager` per firmare/autenticare i token di forward dell'autenticazione

#5 - 11/15/2019 12:11 PM - Enrico Zini

- Assignee changed from Mark Caglienzi to Elena Grandi

Ho fatto push della parte di embedding preparata al 99% da mark in octonet, e il forward dell'autenticazione sembra funzionare.

Manca un po' di code review, e manca decidere cosa consentire di fust-manager agli utenti anonimi.

Passo a Elena per review e per discutere di questa cosa.

#6 - 11/15/2019 12:34 PM - Enrico Zini

Dopo aver sentito simone, guarda lui ora la situazione di quali utenti possano accedere a cosa e a cosa possono accedere gli anonimi

#7 - 11/21/2019 02:46 PM - Elena Grandi

- *Status changed from In elaborazione to Chiuso*

- *Assignee changed from Elena Grandi to Enrico Zini*

Mergiato in master.

Riguardo agli accessi per i vari utenti c'è [#742](#)