SERVIZI DI POSTA ELETTRONICA UN APPROCCIO INNOVATIVO ALLA LORO INTEGRAZIONE CON TECNOLOGIE DI SINGLE SIGN ON

Università degli Studi di Padova

Centro di Calcolo di Ateneo

Carlo Manfredi, Simone Marzola, Giorgio Paolucci, Stefano Zanmarchi

Conferenza GARR 2010

Torino 26,27,28 ottobre

ABSTRACT

La realizzazione di infrastrutture di Single Sign On (SSO) per ottenere l'autenticazione unica e centralizzata ad applicazioni WEB based ha visto negli ultimi anni una crescente diffusione nel mondo scientifico e accademico.
Questa soluzione ha trovato la sua applicazione nella realizzazione dell'accesso al servizio di posta elettronica istituzionale dell'Università degli Studi di Padova.

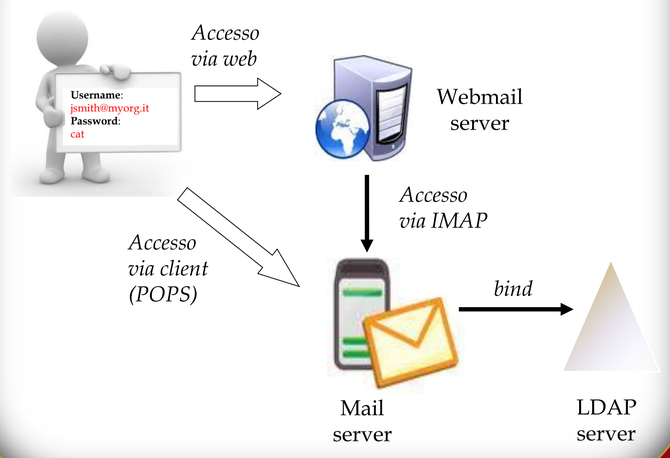
Introduzione

Obiettivo

Far accedere l'utente John Smith con le stesse credenziali a:

- ❑Webmail tramite Single Sign On
- ❑Mail tramite client POPS

La nostra infrastruttura



Problema da risolvere

- ❑ E' relativamente semplice rilasciare il Webmail sotto Shibboleth.
- ❑ Ma è a sua volta un client IMAP del Mail server.
- ❑ Come autenticare l'utente verso il Mail server?

Soluzioni tradizionali

1)Permettere al frontend Webmail l'accesso senza credenziali al Mail server

Facile, ma se un hacker conquista il Webmail può prelevare la mail di tutti!

2)Trasmettere come *shib-attribute* la password dell'utente al frontend Webmail che la utilizza per l'autenticazione verso il Mail server

Tecnicamente possibile, ma sconsigliatissimo: l'hacker accederebbe a ***tutte*** *le applicazioni sotto SSO*

Soluzione

Trasmettere come *shib-attribute* al Webmail l'hash della password contenuto nel server LDAP di autenticazione, usato da IdP e Mail server.

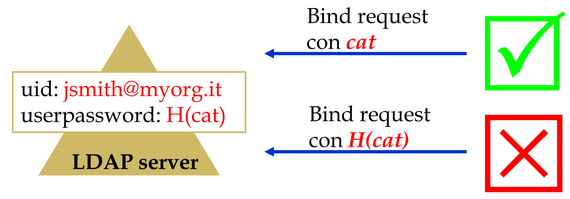
Nel nostro esempio lo *shib-attribute*:

- ❑ non è *cat*
- ❑ bensì H(*cat*) (es: `MD5(cat)`,`SHA(cat)`,ecc.)

Il Webmail server non riceve la password di SSO!!

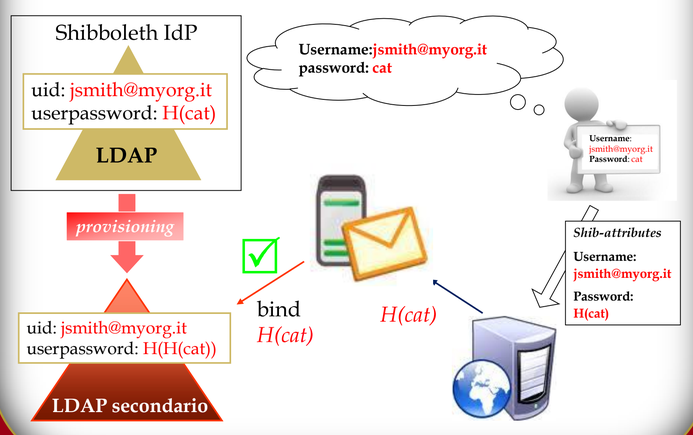
Ostacolo da superare

- ❑ Il server LDAP contiene in *userpassword* il valore H(cat); si può fare il bind solo con *cat*:

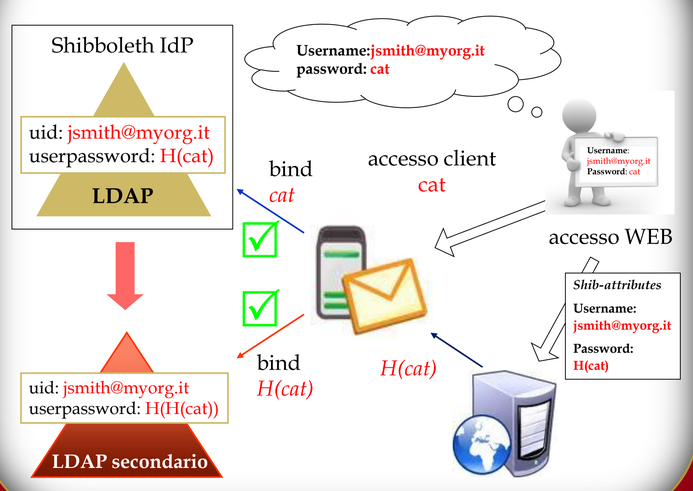


- ❑ Come può allora il Mail server fare il bind all'LDAP di autenticazione solo con H(cat)?

Soluzione un LDAP server secondario



Visione d'insieme



Come realizzare un LDAP secondario?

- ❑ Da evitare l'utilizzo dello stesso LDAP con due campi userpassword *H(cat)* e *cat*: un hacker in possesso di H(cat) potrebbe autenticarsi a qualsiasi servizio sotto SSO.
- ❑ Serve un secondo LDAP (con provisioning veloce per mantenere la sincronia dei cambi password). Possibilità:
  - •un vero server LDAP (problematiche di sincronizzazione);
  - •un overlay software (richiede programmazione);
  - •un Virtual Directory Server (nostra scelta "Penrose").

Virtual Directory

- ❑ Software che risponde a query LDAP prelevando in tempo reale i dati da vari *data source* sottostanti (RDBMS, server LDAP, file di testo).
- ❑Può eseguire elaborazioni complesse sui dati.
- ❑Nel nostro caso il sever LDAP secondario è un LDAP virtuale che preleva i dati dall'LDAP di autenticazione mappando:
  - •uid ➞ uid (nessuna elaborazione)
  - •userpassword ➞ H(userpassword)

Conclusioni

Risultati

- ❑Il Webmail server non riceve più la password di SSO (l'eventuale compromissione del server o dello hash trasmesso, darebbe accesso solo ad una sessione webmail).
- ❑**La soluzione rende disponibile un'architettura per accedere ad applicazioni a 3 livelli che devono autenticarsi su backend.**
- ❑Si può affiancare facilmente ad architetture di autenticazione già presenti senza necessità di rivoluzionarle e senza bisogno di interrompere i servizi di autenticazione già in produzione.

Sviluppi futuri

- ❑ La robustezza della soluzione si basa sulla solidità dell'algoritmo di hashing utilizzato
- ❑ Grazie alla flessibilità del virtual directory è possibile ridurre la validità temporale dello hash trasmesso, aumentando ulteriormente la sicurezza complessiva della piattaforma.