Capitolul Z

MUMSAI- un sistem de autentificare automată

Alboaie Lenuța[1, 2] Alboaie Sînică[2]

[1]Facultatea de Informatică, Universitatea Al.I.Cuza
Str. General Berthelot, nr. 16, Iaşi, România
adria@infoiasi.ro – http://www.infoiasi.ro/~adria

[2] S.C Axiologic Research
Str. C. Negri nr. 39, Iaşi, România
{adria, abss}@axiologic.ro - http://www.axiologic.ro

Rezumat. MUMSAI este un sistem care face posibilă autentificarea automată pe mai multe situri ale unei organizații. Vom prezenta în această lucrare aspectele principale care fac posibile funcționarea unui astfel de sistem.

Cuvinte-cheie: autentificare, SOAP, iframe, PHP

1. INTRODUCERE

Vom începe discuția plecînd de la un simplu scenariu general în care un utilizator doreşte să se logheze pe un anumit sit. La un prim pas se va trimite o cerere către server. Acesta îi va întoarce ca răspuns o pagină şi va seta în header-ul răspunsului HTTP mai multe cook-iuri. Simultan pe partea de server se inițiază o sesiune.

Cook-iurile pot fi vazute ca o pereche (*nume,valoare*) şi sunt folosite pentru identificarea sesiunii.

Într-o sesiune se pot pune diferite informații cum ar fi: utilizator, parolă, adresă de email etc. Aceste informații însă nu pot fi plasate în cookie-uri pentru că nu ar mai exista securitate. Este important să remarcăm că nu se poti citi cookie-urile setate de alt domeniu şi de aici necesitatea apariției softului nostru.

2. ABORDAREA MUMSAI

MUMSAI (*Management of Users on Many Sites and Auto-login all the Internet)* este un sistem de autentificare centralizat compus dintr-un site **server** şi mai multe site-uri **client**.

În discuțiile noastre viitoare vom folosi următoarele notații:

- **S** - serverul principal care autentifică utilizatorii
- **C_1..C_n** - clienți care autentifică utilizatorii cu ajutorul **S**

Serverul principal execută două funcții:

1. Autentificare : se verifica autenticitatea pentru perechea (*user,parola*) trimisă de un client şi se trimite răspunsul corespunzator împreună cu permisiunile utilizatorului în caz afirmativ.
2. Logare automată: logarea pe un sit, implică logarea pe toate siturile organizației

Vom prezenta în această secțiune cele două situații în care se poate afla utilizatorul atunci când doreşte să se autentifice şi vom descrie în paralel mecanismele care stau la baza funcționarii sistemului MUMSAI.

Avem două cazuri:

- cel în care utilizatorul s-a logat direct pe server
- cel în care utilizatorul s-a logat pe unul din siturile client

Caz 1. Utilizatorul este logat pe server

În această situație utilizatorul şi-a introdus id-ul şi parola pe situl server S (suntem la momentul T_0).

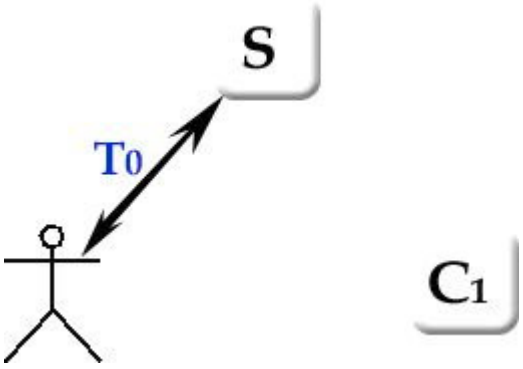

Fig1. Caz1 - Momentul T_0

Întrebarea care se pune este următoarea: care este mecanismul care permite utilizatorului să fie logat automat pe toate celelalte situri ale organizației? Altfel spus dacă utilizatorul intră pe situl client, de exemplu C_1, acesta ar trebui să ştie cine este utilizatorul.

Considerăm că la momentul T_1 utilizatorul face cerere la C_1.

La momentul T_2 au loc:

- C_1 generează un numar random

- C_1 îi va întoarce utilizatorului pagina în care sunt incluse doua iframe-uri.

Aceste iframe-uri au forma:

<iframe src="http://S/getFrame.php?key=nr_random&client=C1 />

<iframe src="http://C1/getFrame.php?key=nr_random />

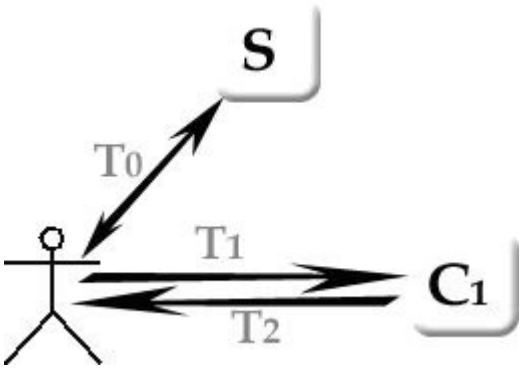

Fig2. Caz1 - Momentul T_1 si T_2

La momentul T_3 , S si C_1 primesc cererile din iframe-uri. S va şti valoarea lui *key* şi în plus S are sesiune cu utilizatorul. Deci în acest moment S ştie că o anumită valoare (a parametrului *key*) este asociată cu un utilizator.

La un moment T3' S va comunica această informație (folosind SOAP) sitului client C_1.

Facem observația că inițierea acestei comunicări de către C1 sau de către S depinde de implementare.

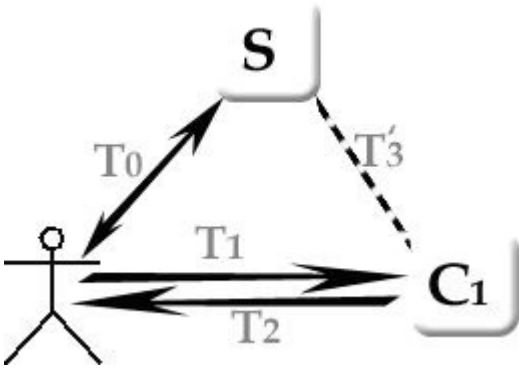

Fig3. Caz1 - Momentul T_3

Iar la momentul T3" C1 comunică utilizatorului faptului că este logat (se face de fapt un refresh al iframe-ului pentru care avem:

src="http://**C1**/getFrame.php?key=nr_random"

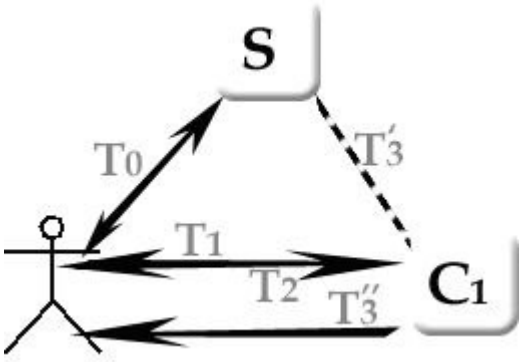

Fig4. Caz1 - Momentul T3″

Utilizatorul va avea senzația ca s-a logat automat.

Caz 2. Utilizatorul este logat pe unul din siturile client C1 .. Cn

În această situație utilizatorul şi-a introdus id-ul şi parola pe un sit client, să zicem C_1 (suntem la momentul T_0). Problema care trebuie rezolvată în acest caz este: cum se face autentificarea pe server?

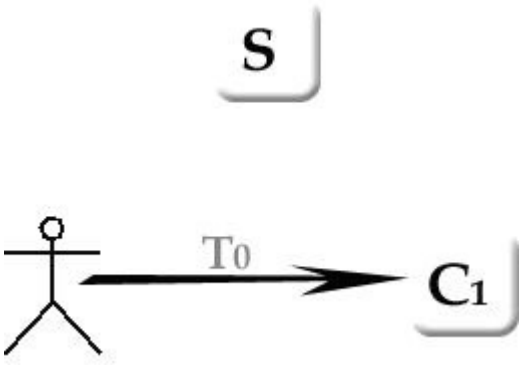

Fig5. Caz2 - Momentul T_0

Facem observația că la un moment T_0', imediat ce C_1 a primit cererea se va încearca logarea utilizatorului. Fiindca nu merge, la momentul T_1 C_1 îi întoarce doua iframe-uri. Unul care conține câmpurile în care utilizatorul trebuie să introducă id-ul şi parola, care face apel la C_1 şi un iframe cu apel la S:

<iframe src=”http:/S/getframe.php?client=C1&key=nr_random />

<iframe src=”http://C1/getFrame.php?key=nr_random />

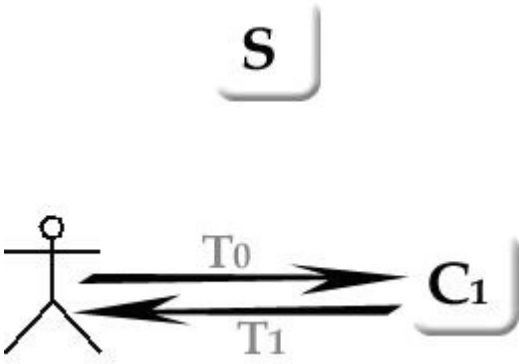

Fig6. Caz2 - Momentul T_1

La momentul T_2 serverul S crează o sesiune, şi întreabă prin SOAP pe C_1 cine este utilizatorul logat căruia îi corespunde acel *Key*. C1 îi răspunde (tot folosind SOAP) user/parola (momentul de timp T_3).

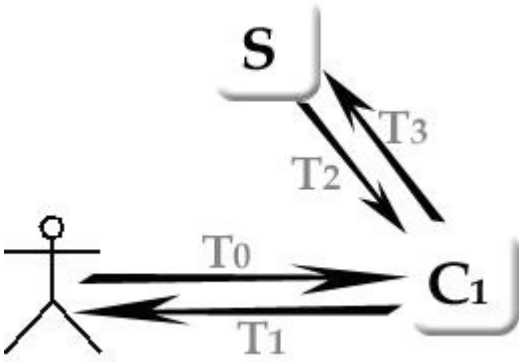

Fig7. Caz2 - Momentul T_2 si T_3

Din acest moment, orice vizită pe siturile client C2…Cn ne duce la cazul 1, discutat anterior.

Mecanismul de autentificare asigura faptul ca o a treia parte care citeşte neautorizat traficul nu poate face o autentificare frauduloasă.

3. TEHNOLOGII FOLOSITE. DETALII DE IMPLEMENTARE.

Pentru implementarea sistemului MUMSAI tehnologiile principale folosite sunt: SOAP-Simple Object Access Protocol, PHP, ifram-uri.

Aşa dupa cum s-a putut vedea în secțiunea anterioară, folosirea iframe-urilor este cheia sistemului MUMSAI. Prin inserarea unui iframe în pagina se poate executa scriptul specificat de atributul *src*. Acesta poate fi şi de pe alt domeniu şi în cazul setării în sesiune a unei variabile aceasta rămâne valabilă pe tot timpul cât browserul este pornit. Dacă se viziteaza acel domeniu, variabila din sesiune poate fi accesată de alt script din acelasi domeniu.

În MUMSAI comunicarea între client şi server se face prin apeluri SOAP.

SOAP este un protocol simplu, utilizat pentru schimbul de informații într-un mediu distribuit, descentralizat.

SOAP permite:

- schimbul de informații structurate într-un mediu distribuit şi descentralizat

- accesarea de servicii, obiecte într-o manieră independentă de platformă

Scopul principal al SOAP este facilitarea interoperabilității între plaforme şi limbaje de programare.

În MUMSAI un client SOAP are următoarea formă:

```
//$address=adresa serverului soap,
$client = new soapclient($address );
//(ex:http://domain/server.php)
//param=vector de parametri ce sunt trimisi
serverului
$param =
array("param1"=>;$param1,"param2"=>;$param2…);
//namespace-ul functiei
$namespace="urn:xmethods-BNNumeFunctie";
//metoda call din clasa soap are ca parametrii
functia din server
// parametrii ei , namespace
$result = $client-
>call('NumeFunctie',$param,$namespace);
if (isset($fault)) {
print "Error: ". $fault;
return false;
}
else
//rezultatul intors de serverul soap(este un
array)
return $result;
```

În serverul SOAP se specifică funcția din client cu acelaşi nume şi parametrii din vectorul *$param* :

```
Function NumeFunctie($param1,$param2…){
//prelucrare param
//….
//serverul poate intoarce catre client un
array
return
array(“param1”=>$param1,”param2”=>$param2);
}
//instantierea unui server soap
$server = new soap_server;
// inregistrare servicii
$server->register('NumeFunctie');
//in caz de eroare se poate apela alt
server,fisier
$fault = $server-
>fault('soap:Server','',$error);
// Trimite rezultatul SOAP
$server->service($HTTP_RAW_POST_DATA);
```

Vom considera mai departe tabelele pentru autentificare folosite în MUMSAI.

Unui utilizator logat îi corespunde o linie in ambele tabele, pe client şi pe server. O înregistrare este ştearsă în momentul acţiunii de *logout*. Daca utilizatorul nu apasa link-ul *logout* şi închide direct browserul, linia rămâne în tabel, dar este suprascrisă la următorul login.

Pentru **Server** avem:

Tabela LOGIN

Camp	Tip	Descriere
User	Varchar	Nickul userului logat
User_key	Varchar	Cheia generata pentru un user
Logintime	datetime	Timpul cand s-a logat

Tabela AutoIncIDs

Camp	Tip	Descriere
IDResources	int	Id-ul site-ului pentru care este valabil un auto id
maxNumber	int	numar incrementat automat la fiecare apel soap
Date	datetime	ziua de valabilitate pentru un maxNumber

Pentru **Client** avem:

Tabela LOGIN

Camp	Tip
User	Varchar
User_key	Varchar
Logintime	datetime

Tabela Autoinc

Camp	Tip	Descriere
id	int	nr incrementat automat la fiecare apel catre server
valid_date	datetime	ziua de valabilitate al id-ului

În secțiunea 2 am discutat despre modul cum se face autentificarea automată a utilizatorilor. Pe lângă această facilitate, MUMSAI asigură şi delogarea automată de pe toate siturile organizației.

Şi în acest caz avem două situații:

1. delogarea de pe situl server

Se parcurg paşii următori:

- ştergerea din tabela proprie a liniei *userlogat*, ştergerea din sesiune
- parcurgerea pe rând a siturilor din tabela *Sites* şi apelarea serverelor SOAP de pe fiecare

```
$sql="select url from ".TABLE_RESOURCES."";
$result=$db->Query($sql);
while(($r=mysql_fetch_array($result))){
$c=new CLOGOUTClient(); $c-
>Connect("http://".$r['url']."/SERVER/
server.php");
$user=$c->SendLogout($usr);
$c->Close();
}
```

- se trimite utilizatorul care trebuie delogat. Un server de pe un C_n şterge din tabela proprie linia corespunzătoare, astfel că la următorul *refresh* , sau apel de pagină, nu va găsi nici o linie în tabelă care să corespundă cu cheia lui, întreabă serverul S şi acesta nu are nimic înregistrat în sesiune, deci nu e logat nici un utilizator pe acea maşină.

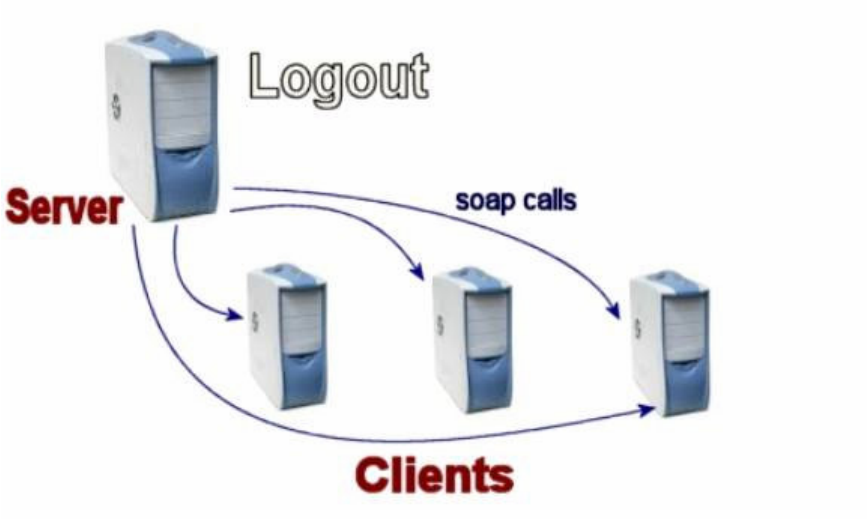

Fig 8. *Logout* şi apeluri SOAP

2. delogarea de pe unul din siturile client

Se parcurg paşii următori:

- ştergere din tabela proprie a liniei (*user,cheie*), ştergerea din sesiune a cheii

- apel SOAP către S :

```
$c=new CLOGOUTClient();
$c->Connect("http://www.your-
domain.com/server.php");
$c->SendLogout($user);
$c->Close();
```

- serverul la rândul lui sterge din tabela proprie LOGIN linia corespunzătoare (*user,cheie*) şi apoi trimite în aceeaşi maniera (fig. 8) un apel SOAP la fiecare sit C_n să şteargă din tabelele lor acel user. La finalul *logout.php* de pe un C_n , se apelează *logoutServer.php* de pe S pentru a şterge şi din sesiunea serverului utilizatorul logat. Apelul se face prin intermediul unui iframe cu parametru cheia curentă de pe un C_n.

4. TESTARE

Pentru a testa modul de funcționare a sistemului MUMSAI vă sugerăm să vă creați un cont pe http://www.axiologic.net/(serverul S din discuțiile noastre). Dupa autentificare veți observa că puteți naviga pe oricare situri care fac parte din sistemul de autentificare comun, fără a mai fi nevoie de nici o operație de login.

O listă a acestor situri este:

http://www.ro.free-test.info/

http://today-news.info/

http://www.intrebare.ro

5. CONCLUZII

MUMSAI este un sistem de autentificare care permite ca un utilizator odată autentificat pe un sit, să poata fi automat autentificat pe alte situri care fac parte din aceeaşi organizație. Mai mult delogarea de pe un sit presupune delogarea automata de pe toate sit-urile organizatiei.

Un alt proiect a carei funcționalitate este similară cu MUMSAI este Passport Network creat de Microsoft şi care permite logarea pe MSN Messenger, MSN Hotmail, MSN Music, precum şi a altor situri şi servicii înrudite.

Însă diferența constă în faptul că sistemul nostru poate fi folosit de către orice organizație care doreşte utilizarea de tehnologii neproprietare.

Referințe

* * *, SOAP: *http://www.w3.org/TR/2003/REC-soap12-part0-20030624/*

* * *, PHP: http://www.php.net/

* * *, IFRAME: *http://www.htmlhelp.com/reference/html40/special/iframe.html*

* * *, MUMSAI: *http://www.axiologic.net/MUMSAI/*