

Tema 2 PoSD

În această temă, dorim să introducem Role Based Access Control pentru server. Fiecare user trebuie asignat unor roluri, și fiecare resursă va avea asociate ACLs pentru fiecare rol. Dacă un rol nu există în ACL, se presupune că acel rol nu are nici un drept.

În continuare se consideră că ownerul resursei are toate drepturile și este singurul care poate crea o resursă. De asemenea, în cazul în care o resursă nu are nici un ACL asociat, le va moșteni pe cele ale părintelui.

Va exista o entitate numită root care poate să creeze roluri, utilizatori și să asocieze utilizatori în roluri. Rolurile și utilizatorii vor fi păstrați într-o bază de date MySQL. Sistemul trebuie să implementeze nivelul RBAC 1 (relația user-role este many-to-many și trebuie să fie ușor să identificăm ce roluri are un anumit user și ce utilizatori sunt într-un anumit rol).

Funcția **changeRights** trebuie modificată astfel încât să asocieze ACL pentru fiecare rol existent (dacă nu se asociază o intrare pentru un anumit rol, acel rol nu are nici un drept).

Va fi implementată o nouă funcție **addRights**, asemănătoare cu **changeRights** din punct de vedere al protocolului, care adaugă un rol cu drepturile respective la ACL-ul asociat resursei. În ambele cazuri, numai ownerul poate schimba drepturile.

Se consideră că un user adoptă drepturile tuturor rolurilor în care este membru.

Punctaj:

Compilare program, rulare: 6p

Rulare TestCases: 4p

Testcases

Aceste comenzi trebuie rulate în această ordine. Presupun un sistem cu doi utilizatori, Alice și Bob, inițial doar cu directoarele implicite: /bob și /alice. Parola pentru Alice o presupun alicia și a lui Bob bob.

1. createRole(bob, bob, role1) **NotAuthorized**
2. createRole(root, root, role1) **ok**
3. changeRights(role1, "r");
4. assignRole(bob, role1)
5. createResursa(alice, alicia, /alice/cursuri.java, 1,"cursuri") **returnează ok**
6. readResursa(bob, bob, /alice/cursuri) **returnează NotAuthorized**
7. addRights(bob,bob,/alice/cursuri,role1) **returnează NotAuthorized**
8. addRights(alice,alice,/alice/cursuri,role1) **returnează OK**
9. readResursa(bob, bob, /alice/cursuri.java) **returnează ok, "cursuri"**

10. writeResource(alice, alice, /alice/cursuri.java, "cursuri2") **returneaza ok**
11. writeResource(bob, bob, /alice/cursuri.java, "cursuri3") **returneaza NotAuthorized**
12. changeRights(role1, "w");
13. writeResource(bob, bob, /alice/cursuri.java, "cursuri3") **returneaza Ok**
14. readResursa(bob, bob, /alice/cursuri.java) **returneaza NotAuthorized**
15. createRole(root, root, role2) **ok**
16. changeRights(role2, "r") **ok**
17. assignRole(bob, role2) **ok**
18. addRights(alice,alice,/alice/cursuri,role2) **returneaza OK**
19. readResursa(bob, bob, /alice/cursuri.java) **returneaza ok, "cursuri3"**

Deadline: 27 Noiembrie 2016.