분산 웹서버 접근제어를 위한 work개념의 RBAC모델

심완보°, 박석
서강대학교 컴퓨터학과
cool96@chch.ac.kr spark@dblab.sogang.ac.kr

The work-concept RBAC Model for the access control of the distributed web servers

Won Bo Shim° , Seog Park
Dept. Computer Science, Sogang University

요 약

오늘날 웹서버를 활용한 업무처리 시스템에서는 웹서버의 기능이 중앙집중화된 정보시스템보다는 웹서버가 처리해야 할 기능별로 별도의 웹서버를 두어 웹서버의 부하를 분산시켜 처리를 하는 시스템 구성이 일반적이다. 이러한 환경에서 자연히 웹서버도 많아지고 사용하려는 사용자도 많아지게 되는데 이때 이러한 사용자가 웹서버에서 제공하는 자원에 대한 접근을 제어할 필요가 있게 된다. 이를 위한 효율적인 방안으로 RBAC(Role Based Access Control)을 사용하는 방법을 생각할 수 있다. 그러나 복수개의 서로 다른 서비스를 담당하는 각각의 서버에는 서로 다른 RBAC 구조가 존재할 수 있게 된다.
이러한 시스템환경에서 일반적으로 한사람의 사용자는 각각의 서버마다 서로 다른 역할을 담당하게 되고 자신의 업무를 처리하는데 있어 각각의 서버별로 별도의 역할을 부여받게 된다. 이에 본 논문에서는 동일 도메인 내에서의 분산 웹서버들이 존재할 때 현재 접근제어의 가장 적합한 개념인 역할기반 접근제어기법을 응용하여 사용자가 복수개의 이들 웹서버를 사용하여 업무를 처리함에 있어 매번 각 서버에서 인증을 받아야하는 불편을 없애 이 문제를 효율적으로 해결해 보고자하며 이를 위해 기존의 RBAC에 Role의 상위 개념인 Work개념을 도입해 사용자가 자신의 업무를 수행시 Role이 아닌 좀더 추상적이고 포괄적 개념인 Work를 선택할 수 있게 함으로 해서 각 서버에서 선택된 Work에 따라 자신에게 부여되는 권한을 이용해 원활하게 업무를 수행할 수 있도록 하는 방법을 제안한다.

1. 서론

Role 개념은 사용자와 자원을 사용할 수 있는 권한 사이에 존재하는 매개적인 개념으로 정보시스템 내에서의 접근제어에 유용하게 사용될 수 있다.[4]

그러나 이 Role개념은 상위 Role이 하위 Role의 권한을 모두 상속받는다는 개념으로 인해 관리에 편리함도 주지만 많은 제약도 주는 것이 사실이다.

실제 정보시스템의 자원에 대한 접근제어를 구현함에 있어 이러한 Role의 수직적 개념도 필요하지만 수평적 개념을 사용하여 문제를 해결할 수 있는 상황도 많이 발생하게 된다.

예를 들면 회사의 구조조정이라는 Task Force팀에 소속된 사용자가 회사의 구조조정이라는 업무를 수행한다고 하자.

이 업무는 회사의 여러 자료들을 사용해야만 수행될 수 있을 것이다.

하지만 회사의 자료들이 업무별로 분리되어 정보시스템상의 별도의 서버들에 의해 각각 분산되어 관리되고 있고 각각의 서버들의 정보자원은 RBAC에 의해 접근제어 되고 있다고 가정해 볼 수 있다.

이러한 상황에서 사용자는 자신의 업무를 수행하기 위해서는 기존의 각 서버가 갖는 수직적 구조에 대한 Integrity를 침해하지 않으면서 각 서버의 여러 복합적인 Role들의 권한을 가질 필요가 있게 된다. 이러한 개념을 본 논문에서는 Work이라 부를 것이다.

이 논문에서는 이 Work개념을 이용해 분산 웹 환경에서 사용자가 여러 서버에 있는 Role의 권한을 수평적으로 가질 수 있게 하여 수직적 Role구조만을 갖는 RBAC운영에 융통성을 갖게 하고 이를 통해 사용자가 각각의 서버 내에 부여된 Role의 권한을 이용해 웹서버들에 대한 접근제어를 받을 수 있도록 하고자 한다. 그렇게 되면 다른 서버에서 업무를 처리하기 위해 또 다시 각각의 서버에서 인증을 받아야 하는 부담 없이 복수개의 웹서버를 사용하는데 있어 사용자는 마치 하나의 웹서버에서 자원을 접근하고 있는 듯한 투명성을 제공할 받을 수 있게 될 것이다.

2. 동일 도메인내의 복수개의 웹서버에서의 역할

2.1. 다른 RBAC구조를 갖는 복수개의 웹서버

위에서 설명한 RBAC개념을 복수개의 웹서버환경 하에서의 상황을 생각해 보면 그림1과 같다.

사이트 A에는 사이트A의 고유의 역할 구조를 갖게 되고 사이트 B는 사이트 B대로 사이트 A와는 다른 역할 구조를 가질 수 있다.

이때 동일한 사용자라도 각각의 사이트 내에서는 서로 다른 역할을 갖게 될 것이다.

예를 들면 그림 1의 사용자가 사이트A에서는 Director의 역할을 갖는 반면 사이트B에서는 Engineer역할을 갖게 된다.

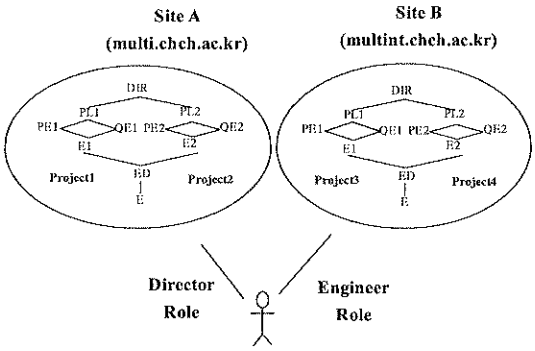

그림 1. 서로 다른 RBAC구조를 갖는 웹서버

이때 사용자는 사이트A에서의 활동영역과 자원접근은 Director에게 부여된 자원접근 권한을 갖게 될 것이고 사이트B에서는 Engineer로서의 접근권한을 갖게 될 것이다.

2.2. 동일 도메인내에서 복수개의 RBAC웹서버로의 접근

예를 들어 자신의 업무를 처리하기 위해서는 여러 개의 웹서버에 분산되어 있는 웹문서를 참조해야만 하는 사용자가 자신의 업무를 처리하기 위해 사이트A에 있는 웹문서를 먼저 접근한다고 할 때 사이트A에서는 사용자가 인증과정을 거쳤는지 체크를 하게 될 것이다.

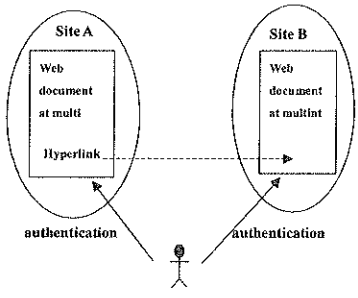

그림 2. RBAC으로 접근제어되는 웹 서버로의 접근

이때 인증이 안된 사용자는 자신의 ID와 패스워드로 인증을 받게 되고 인증에 성공하면 사이트A에서 자신이 처리하고자 했던 업무를 처리할 수 있을 것이다.
사이트A에서 업무를 처리하던 중 자신의 일을 완수하기 위해서는 사이트B에 접근할 필요가 있는 사용자는 사이트B에 있는 문서를 참조하려 할 것이다.
이때 사이트B는 사용자에게 또 다른 인증 과정을 거칠 것을 요구 하게 될 것이며 사용자는 또다시 자신의 ID와 패스워드로 인증을 받아야 만이 사이트B에서도 계속적으로 업무를 해나갈 수 있을 것이다.
현실적으로 웹문서들은 같은 서버내의 문서들 뿐만 아니라 다른 서버내의 문서들간에도 많은 하이퍼링크로 연결 되어 있고 각각의 웹 문서들은 각각의 웹서버의 보안정책에 따라 접근제한을 받게 된다.
이러한 환경에서 사용자는 자신의 업무를 원활히 처리해 나가는데 있어 불필요한 추가적인 인증절차로 많은 어려움을 겪게 될 것이다.
이에 본 논문에서는 Work개념을 도입하여 사용자가 자신의 업무를 처리하기 위해 처음에 접근하는 웹서버에서 인증과정을 거치고 Work을 선택하면 이미 Work에 매핑 되어 있는 각각의 웹서버에서의 역할권한을 가질 수 있게 하여 추가적인 불필요한 인증작업 없이 투명하게 동일 도메인내에 있는 복수개의 웹서버를 접근 할 수 있게 하려한다.

3. Work개념을 도입한 RBAC 모델

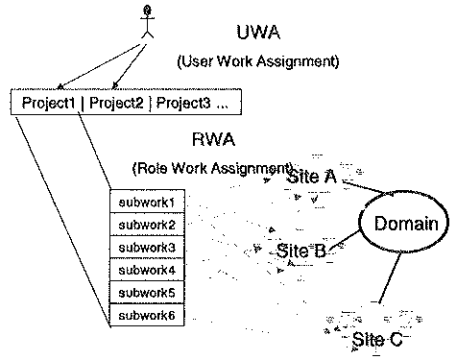

그림 3. Work개념을 도입한 RBAC 모델

그림3은 동일 도메인 내에 복수개의 서로 다른 웹서버가 있을 수 있고 각각의 웹서버에서는 각각의 역할구조를 갖는 역할들이 존재하는 상황에서 사용자가 어떻게 자신의 업무를 처리해 나갈 수 있는지를 보여주는 RBAC 모델이다.

사용자는 Project1과 Project2라는 두 가지 Work에 업무가 할당되어 있고 Project1은 각각 6개의 Subwork으로 나뉘어져 있다.
각각의 Subwork들은 다시 각각 서로 다른 서버에 있는 역할들에 매핑 되어 있게 된다.
이때 사용자가 자신에게 할당된 Work중의 하나인 Project1을 선택한다는 것은 Project1 Work내에 포함되어 있는 Subwork을 통해 SiteA, SiteB, SiteC에 매핑되어 있는 역할들의 권한을 가지고 각각의 웹서버들을 접근할 수 있게 된다는 것이다.
이 모델을 RBAC96 모델에 Work 개념을 포함해 확장해 정의해 보면 다음과 같다.

Work개념을 도입한 RBAC모델의 정의

1) U : Users, W : Works, SW : Subworks,
   ST : Sites, R : Roles, P : Permissions, S : Sessions
2) RH ⊆ R X R : 역할구조
3) W ∈ SW : Work은 Subwork들로 구성된다.
4) SW ⊆ ST X R : Subwork은 사이트들에 있는 역할들로 구성된다.
5) UA ⊆ U X SW : Subwork이 User에 할당된다.
6) PA ⊆ P X R : 각 Role에는 Permission이 할당된다.

7) subworkroles function : SW -> R
subworkroles(sw_i) $\subseteq$ {r | (st_i,r) $\in$ SW} : Site i에서의 Subwork sw_i에 배정된 역할들을 구하는 함수
8) user function : S -> U, session s_i 에 연결된 사용자를 구하는 함수
9) Subworks function : S -> SW, session s_i 에 연결된 Subwork을 구하는 함수,
subworks(s_i) $\subseteq$ {sw|(user(s_i),sw) $\in$ UA}
10) sessionroles function : S -> R,
roles(s_i) $\subseteq$ {r | ($\exists$r' $\geq$ r)[sw' $\in$ subworks(s_i) $\wedge$ (user(s_i),sw') $\in$ UA $\wedge$ r' $\in$ (subworkroles(sw')]}

4. 동일 도메인내의 복수개의 웹서버 접근제어 구현

그림4는 Work개념을 도입한 RBAC모델을 사용하여 동일 도메인 내의 웹서버들에 대한 접근제어 구현을 위한 시스템구성을 보여준다.
중앙에 각 사이트마다의 역할에 대한 정보를 관리하고 제공해 주는 Work 개념의 RBAC Server가 있고 이는 각각의 서버에 분산되어 있어도 별 차이는 없을 것이다.

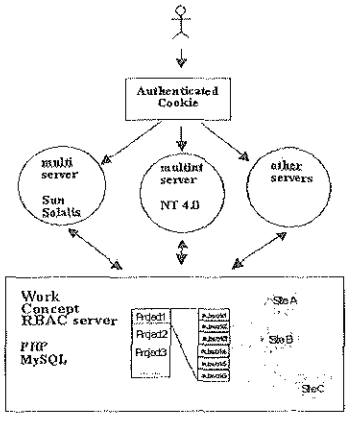

그림 4. 시스템 구성도

사용자는 이중 어느 한 사이트의 문서에 접근을 하게 되고 처음 접근 시에는 해당 웹서버로부터 사용자 인증을 요구받게 될 것이다.

사용자가 해당 웹서버에서의 인증에 성공하면 사용자는 자신에게 부여된 Work들중 하나를 선택할 수 있게 되고 사용자를 인증한 서버는 사용자의 클라이언트로 사용자의 접근권한 정보가 담긴 쿠키를 보내게 된다.

이때 서버로부터 보내진 쿠키는 보안을 위해 메인 메모리에만 생성되고 이후부터는 인증된 사용자는 동일 도메인 내에 있는 다른 모든 웹서버들로부터 또다른 인증 절차 없이 해당서버에 메인 메모리에 있는 인증쿠키를 제시함으로 해서 자신의 업무를 처리하기 위한 웹서버들로의 접근이 가능해지게 된다.

참고로 Role Server에는 사이트에 대한 정보, Work에 대한 정보, Work을 구성하는 Subwork에 대한 정보, 각 사이트마다의 역할에 대한 정보, 사용자에 대한 정보, PRA(Permission Role Assignment)정보, UWA(User Work Assignment)정보, RWA(Role Work Assignment)정보를 가지고 각각의 웹서버에 접근제어정보를 제공해 주게 된다.

5. 결론

분산 웹환경에서는 대규모의 사용자와 웹서버가 존재하게 되고 이때 이들에 대한 효율적인 접근제어가 필요하게 된다.
이에 본 논문에서는 현재 사용자와 네트워크상의 자원에 대한 관리상의 오류를 줄여 주고 관리비용을 감소시켜 줄 수 있도록 할 수 있는 접근제어의 가장 적합한 개념인 역할기반 접근제어기법에 Work개념을 도입한 확장된 RBAC 모델을 제시하고 정의하였으며 이를 통해 동일 도메인 내에서의 분산 웹서버들이 존재할 때 사용자가 복수개의 이들 웹서버를 사용하여 업무를 처리함에 있어 매번 각 서버에서 인증을 받아야하는 불편을 없애 사용자가 불필요한 추가적인 인증 절차 없이 자신의 업무를 수행해 나갈 수 있는 방법을 제시해 보았다.
또한 간단한 구현을 통해 타당성을 확인해 봄으로 해서 우리가 제시한 Work개념을 도입한 RBAC확장 모델이 시스템의 투명성을 제고 시켜 사용자의 효율적인 업무 수행을 가능하도록 할 수 있다는 것을 보였다.

6. 참고문헌

[1]. D. Ferraio, J. Cugini and R. Kuhn, "Role-based Access Control(RBAC): Features and motivations", Proc. of 11th Annual Computer Security Application Conference, 1995.
[2]. Ravi Sandhu and Joon S. Park,"Secure Cookies on the Web", IEEE Internet Computing, July-August, 2000.
[3]. J. F. Barkely, A. V. Cincotta, D. F. Ferraiolo, S. Gavrilla and D. R. Kuhn, "Role Based Access Control For the World Wide Web", 20th NCSC, 1997.
[4]. R. Sandhu, E. Coyne, H. Feinstein, and C. Younman, "Role-Based Access Control Models", IEEE Computer Magazine Vol. 29, 1996.
[5]. Sejong Oh, Seog Park, "Task-Role Based Access Control(T-RBAC):An Improved Access Control Model for Enterprise Environment", *DEXA, 2000*.
[6]. Gail-Joon Ahn, R. Sandhu, Myong Kang and Joon Park, "Injecting RBAC to Secure a Web-based Workflow System", Proc. of the Fifth ACM Wrokshop on Role-Based Access Control, ACM, 2000.
[7]. E. C. Lupu and M.S. Sloman, "Reconciling Role Based management and Role Based Access Control" , Second *ACM Wrokshop on Role-Based Access Control*, 1997.
[8]. Ezedin Barka & Ravi Sandhu, "Framework for Role-Based Delegation Models", ACSAC, 2000.
[9]. Ravi Sandhu, David Ferraiolo and Richard Kuhn, "The NIST Model for Role-Based Access Control: Toward A Unified Standard" , Proc. of the Fifth ACM Wrokshop on Role-Based Access Control, 2000.
[10]. R. Sandhu and V. Bhamidipati, "The URA97 Model for Role-Based User-Role assignment", Proc. of IFIP WG 11.3, 1997.