Rollenbasierte Sicherheitsmodelle

Jon Kowal

Humboldt Universität Berlin, Institut für Informatik

Zusammenfassung

Mit der wachsenden Größe von Unternehmen und IT-Strukturen und der daraus resultierenden globalisierten, vernetzten Verwendung von Software, entsteht das Bedürfnis Zugriffe nicht für nur einzelnen Gruppen und Benutzer zu erlauben/beschränken, sondern vielmehr aufgabenorientiert, das heißt: *rollenbasiert*. Diese Arbeit beschäftigt sich mit dem Grundgedanken Rollenbasierter Sicherheitsmodelle und den Ansätzen von verschiedenen Seiten Standards und Definitionen zu schaffen.

1. Einleitung

Der Ansatz klassischer Sicherheitsmodelle unterliegt der Idee den Zugriff einzelner Subjekte auf einzelne Objekte zu beschränken, um den Subjekten Lesezugriff, Schreibzugriff und ähnliche Rechte über den Objekten zu erteilen. Da dieser Ansatz in großen Systemen schnell zu einem überdimensionalen Verwaltungsaufwand führt, wurden in verbreiteten Implementationen solcher Systeme (z.B. Unix Rechteverwaltung) die Subjekte in Gruppen zusammengefasst, welche dann wiederum Rechte definierte über den Objekten (in Unix Dateien) haben.

Rollenbasierte Sicherheitsmodelle greifen diesen Ansatz auf und ermöglichen es organisatorische Strukturen, Verantwortungen und Befugnisse detaillierter und intuitiver zu Verwalten. Der Gruppengedanke ist genau genommen bereits eine Form Rollen zu definieren, nur wird dieser Gedanke in einem Rollenbasierten Sicherheitsmodell zum Abschluss gebracht und so werden Rollenbeschränkungen und -abhängigkeiten hinzugefügt, Hierarchien und Rollenaktivierung.

Für ein Softwaresystem werden Rollen definiert, welche wiederum bestimmte Zugriffsrechte und -restriktionen besitzen, die zunächst meist unabhängig voneinander definiert werden. Dies soll der natürlichen Struktur innerhalb einer Organisation entsprechen, um eine einfache Verwaltung der Benutzer (Subjekte) zu ermöglichen. So treten Personen in einem Krankenhaus z.B. in den Rollen Krankenschwester oder Arzt, in einer Bank als Fondsmanager, Kundenbetreuer oder Kassenangestellter auf, wobei die Rollen in der Bank beispielsweise auch von einer einzigen Personen erfüllt werden könnten, die im Krankenhaus wiederum einer gewissen Hierarchie unterliegen. Für einen Benutzer wird definiert in welchen Rollen er auftreten kann, das heißt aber nicht, dass er gleichzeitig in allen Rollen aktiv ist bzw. aktiv sein kann.

Die Ansätze zur Definition Rollenbasierter Sicherheitsmodelle variieren vielfach um Details wie „Es kann Rollen geben, die einander ausschließen“, „Ein Subjekt kann gleichzeitig nur in einer Rolle aktiv sein“, etc. und so versucht diese Arbeit verschiedene Ansätze vorzustellen und zwar die Definition von Claudia Eckert [ECKERT], den Standardisierungsversuch des National Institute of Standards and Technology [NIST], sowie das Analytical Role Modelling Framework von Crook, Ince und Nuseibeh [CIN].

2. Definition von Claudia Eckert

Claudia Eckert definiert in [ECKERT] ein RBAC (role based access control) als 7-Tupel:

RBAC = (S, O, RL, P, sr, pr, session)

S ist eine Menge von Subjekten, O eine Menge von zu schützenden Objekten, RL eine Menge von Rollen und P eine Menge von Privilegien (Rechte). *sr* ist eine Zuordnung von Subjekten zu Rollen:

$$sr : S \rightarrow 2^{RL}$$

Die Schreibweise $sr(s) = \{R_1, \ldots, R_n\}$ gibt an, dass ein Subjekt autorisiert ist in den Rollen R_1 bis R_2 aktiv zu sein. *pr* ist eine Abbildung, die jeder Rolle eine Menge an Zugriffsrechten zuordnet:

$$pr : RL \rightarrow 2^{P}$$

session drückt die Rollenmitgliedschaft eines Subjektes aus. Im Unterschied zu *sr* ist hier definiert in welchen Rollen das Subjekt derzeit aktiv ist. Dies ist relativ offen gehalten, um Implementationen Spielraum um die schon oben angedeuteten Variationen Rollenbasierter Sicherheitsmodelle zu erlauben.

$$session \subseteq S \times 2^{RL}$$

Für $s \in S$ gilt also: $(s, rl) \in session$ bedeutet s ist aktiv in den Rollen $R \in rl$.

Zusammenfassend stellt ein von Claudia Eckert definiertes *RBAC* ein vollständiges Rollenbasiertes Sicherheitsmodell dar, da eine Zuordnung der Subjekte zu Rollen und somit zu bestimmten Privilegien ermöglicht wird, wodurch der Zugriff auf die zu schützenden Objekt in einem System für alle Subjekt eindeutig dadurch definiert ist, in welchen Rollen dieses aktiv sind.

3. NIST RBAC model

Das National Institute of Standards and Technology (NIST) schlägt in seinem Standardisierungsversuch zu Rollenbasierten Sicherheitsmodellen [NIST] (engl. RBAC – Role Based Access Control) ein kaskadierendes Modellschema vor. Grund hierfür ist die Tatsache, dass ein System nur so aufwendig sein muss, wie die Ansprüche, denen es genügen soll. Deshalb hat das NIST befunden, dass ein Rollenbasiertes Sicherheitsmodell, welches alle Ansprüche erfüllt für den Einsatz in einfachen Systemen zu unhandlich ist und deshalb in diesen keine Anwendung finden wird. Desweiteren hat das NIST in seinem Modell versucht, die bereits in der Industrie verbreiteten Ansätze aufzugreifen, die es tatsächlich bereits zu einer Akzeptanz gebracht haben, um daraus einen – zu der Zeit gab noch keinen – Standard für Rollenbasierte Sicherheitsmodelle zu formen.

Das Modellschema ist strukturiert in Core RBAC, Hierarchical RBAC, Constrained RBAC:

3.1 Core RBAC

Das Core RBAC umfasst im Grunde alle schon in der Definition von Claudia Eckert beschriebenen Aspekte. An dieser Stelle nochmal kurz genannt sind dies die Zuordnung von Subjekten (users) zu Rollen, welchen wiederum Privilegien (prms = permissions = operations on objects) zugeordnet sind, sowie Sessions, die einem Subjekt die selektive Aktivierung oder Deaktivierung von Rollen erlauben. In diesem Sinne soll das Core RBAC die wesentlichen Grundeigenschaften von Rollenbasierten Sicherheitsmodellen enthalten, ohne auf weitergehende Variationen einzugehen.

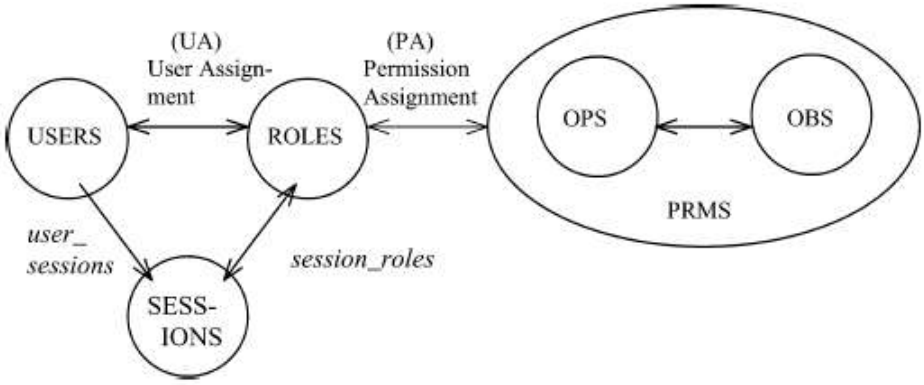

Abb. 1: Core RBAC

3.2 Hierarchical RBAC

Das Hierarchical RBAC fügt dem Core RBAC eine partielle Ordnung über den Rollen hinzu, welche eine wesentliche Vereinfachung von Rollendefinitionen bzgl. häufig zugeordneter Privilegien erlaubt. Es wird unterschieden zwischen dem *General Hierarchical RBAC*, welches eine beliebige partielle Ordnung als Rollenhierarchie zulässt und dem *Limited Hierarchical RBAC*, welches darauf eingeht, dass viele Systeme Restriktionen bzgl. der Ordnung festsetzen, z.B. so dass die Hierarchie auf einfache Strukturen wie Bäume begrenzt ist.

Die Notation von Claudia Eckert erweiternd, ist eine (allgemeine) Rollenhierarchie (*RH*) zu verstehen als:

> $RH \subseteq RL \times RL$, wobei $r_1 \succeq r_2$ nur gilt, wenn alle Privilegien von r_1 auch Previlegien von r_2 und alle Subjekte von r_1 auch Subjekte von r_2 sind.

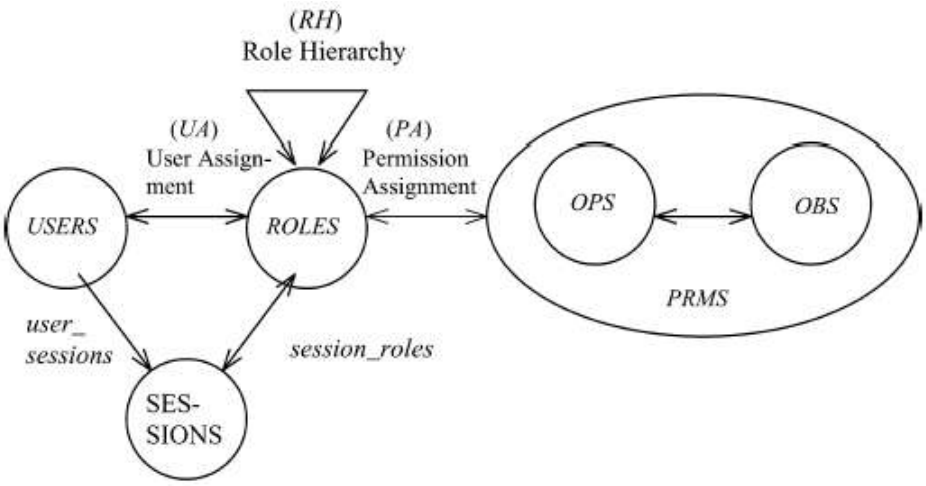

Abb. 2: Hierarchical RBAC

Folgendes Beispiel für ein Hierarchical RBAC ist ebenfalls aus [NIST] entnommen:

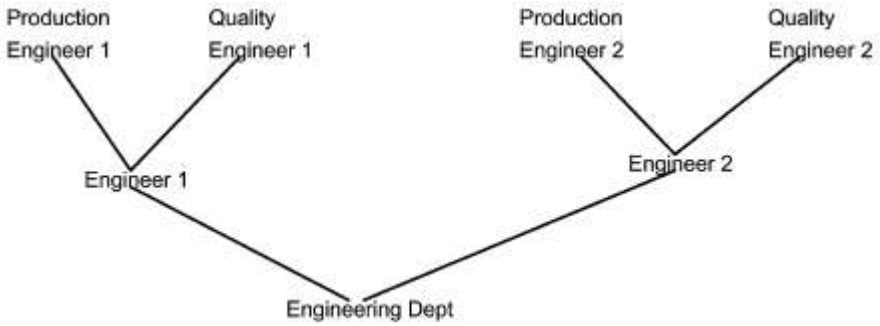

Abb. 3: Beispiel Rollenhierarchie als Baum

3.3 Constrained RBAC

Alle weiteren Spezifikationen, die das NIST einem Rollenbasierten Sicherheitsmodell zugesteht finden im Constrained RBAC Platz. Dazu gehören vor allem die *Static Separation of Duty Relations* (SSD), sowie *Dynamic Separation of Duty Relations* (DSD). Beiden ist gemeinsam, dass einem Subjekt die Möglichkeit bestimmte Beschränkungen bezüglich seiner Rollenmitgliedschaft auferlegt werden, wobei sich Ersteres direkt auf die Rollenmitgliedschaft auswirkt, Letzteres aber nur auf die gleichzeitig aktiven Rollen eines Subjektes. Zum Beispiel besteht die Möglichkeit Rollen zu definieren, die einander ausschließen. SSD verhindert an dieser Stelle, dass ein Subjekt

beide Rollen zugewiesen werden können, DSD verhindert lediglich, dass ein Subjekt gleichzeitig in beiden Rollen aktiv ist. Ziel ist es so einem Subjekt zu einem Zeitpunkt nur die minimal nötigen Privilegien zuzuteilen, die für die Durchführung der momentan aktiven Rolle(n) nötig sind.

Die Eingliederung beider Beschränkungen in das obige Schema sollen die folgende Abbildungen aus [NIST] veranschaulichen.

> $SSD \subseteq (2^{RL} \times \mathbb{N})$ ist eine Sammlung von Paaren (rs, n) mit $n \geq 2$, wobei kein Subjekt n oder mehr Rollen aus der Menge rs für jedes (rs, n) $\in SSD$ zugeordnet ist.

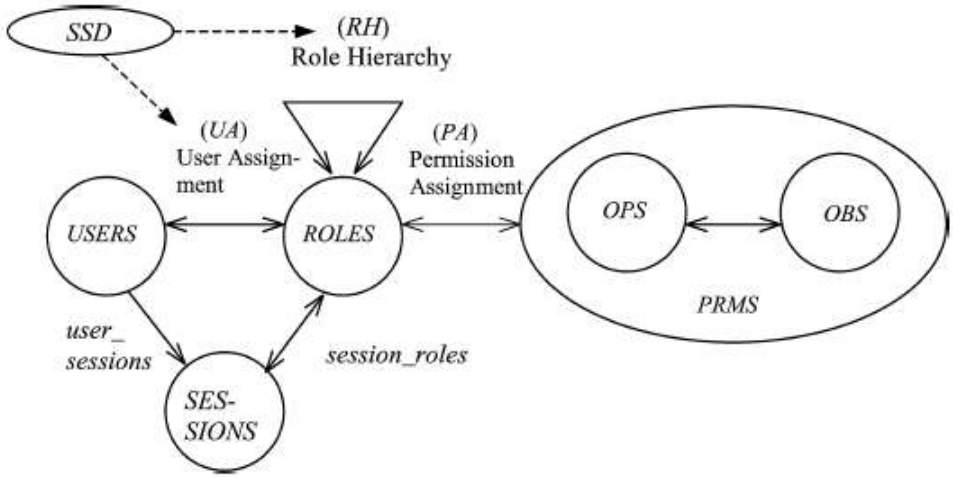

Abb. 4: SSD mit Hierarchical RBAC

> $DSD \subseteq (2^{RL} \times \mathbb{N})$ ist eine Sammlung von Paaren (rs, n) mit $n \geq 2$, wobei kein Subjekt n oder mehr Rollen aus der Menge rs für jedes (rs, n) $\in DSD$ aktivieren darf.

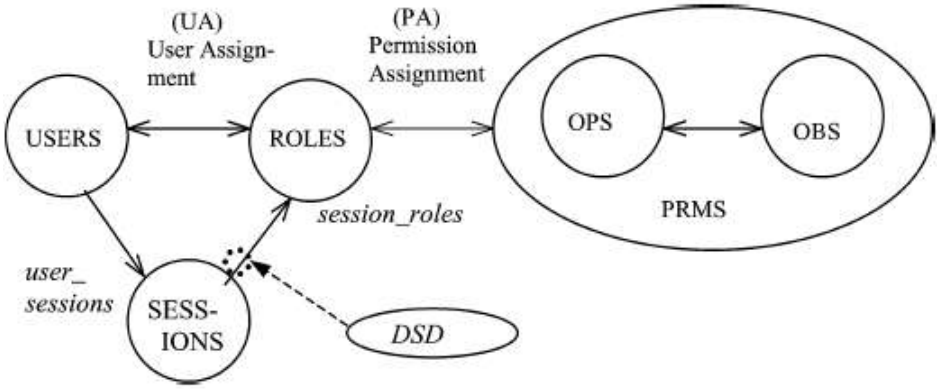

Abb. 5: Dynamic Separation Of Duty relations

3.4 Funktionale Spezifikation

In [NIST] wird eine umfassende funktionale Spezifikation für die Implementation des vorgeschlagenen RBAC-Modells angeboten, auf die an hier nicht weiter eingegangen wird, da Sie das oben genannte lediglich syntaktisch beschreibt, ohne inhaltlich neues hinzuzufügen.

4. Analytical Role Modelling Framework von Crook, Ince und Nuseibeh

Crook, Ince und Nuseibeh greifen das RBAC Verständnis aus [NIST] auf, beschäftigen sich in [CIN] jedoch tiefer gehend mit der Frage was Rollen sind und wie diese strukturiert und hergeleitet werden können. So entstehen drei Kategorien von Rollen, *Roles based on Seniority* und *Roles based*

on Function, *Roles based on Market*. Aus diesen 3 Kategorien werden einem Subjekt (User) die entsprechenden Rollen zugewiesen, um die Stellung, Funktion und Zugriffsrechte (auf Objekte) des Subjekts innerhalb der Organisation so genau wie möglich abzubilden.

4.1 Roles based on Seniority

Seniority (Alter, Vorrang, Stellung) entspricht einer Rolleneinteilung basierend auf organisatorischen Hierarchien. Dies entspricht so ungefähr der bereits in 3.2 beschriebenen Rollenhierarchie.

4.2 Roles based on Function

Da hierarchische Rollen ohne dahinterstehende Funktionen meist nicht sinnvoll sind, werden die *Roles based on Function* hinzugefügt, welche Qualifikation, Funktionen und Arbeitsprozesse einem Subjekt zugeteilt werden. Natürlich besteht zwischen all dem ein unmittelbarer Zusammenhang. So ist eine Arbeitsprozess-Rolle direkt abhängig von der entsprechenden Qualifikation oder Funktion des Subjekts, sowie der zugewiesenen Stellung in der Organisation (Seniority).

4.3 Roles based on Market

Während Rollen, die auf Stellung und Funktion des Subjekts basieren, bestimmen welche Aufgaben das Subjekt erledigen kann, bestimmen die *Roles based on Market*, auf welchen Objekten ein Subjekt diese Aufgaben erledigen darf. Daher werden diese Rollen auch als inhaltsbasierte (contextual) Rollen beschrieben.

4.4 Analytical Framework

Das in [CIN] beschriebene Analytical Framework versucht dem in 4.1 – 4.3 erläuterten Ansatz der Rollenkategorisierung gerecht zu werden und wird formal in der Spezifikationssprache Z [SPIVEY] beschrieben. Es werden folgende Basistypen definiert:

[*ROLE*]	repräsentiert eine funktionale, hierarchische, oder inhaltsbasierte Rolle
[*ROLE_INSTANCE*]	Rolleninstanz ist eine einem Benutzer zugewiesene Rolle
[*CONTEXT_TYPE*]	beschreibt Inhaltstyp, der Zuordnung von inhaltsbasierten Rollen zu Objektinstanzen ermöglicht
[*ASSET_CATEGORY*]	Ein Asset ist eine Kategorie zu schützender Objekte. Alle zu schützenden Objekte müssen einer Kategorie zugeordnet werden
[*ASSET_INSTANCE*]	Instanz eines Asset.
[*USER*]	Subjekte, die Benutzer des Systems darstellen
[*OPERATION*]	Basistyp für Operationen

Aus diesen Basistypen ergibt sich das zusammengesetzte Schema für eine Operationsrichtlinie, welche die oben beschriebene Zugriffsbeschränkung modelliert.

$$OPERATION\_POLICY \triangleq [RoleSets: \mathbb{P}(\mathbb{P} ROLE);\ Operations: \mathbb{P}\ OPERATION;\ AssetCategory: ASSET\_CATEGORY]$$

Das folgend aufgeführte Schema beschreibt nochmals formal den soeben beschriebenen Sachverhalt, für eine detaillierte Beschreibung mit einem ausführlichen Beispiel sei jedoch ausdrücklich auf das Paper [CIN] hingewiesen, da an dieser Stelle nur die Idee der verschiedenen Rollenkategorien beschrieben werden soll. Nur soviel sei zum Beispiel in [CIN] ergänzt Es erläutert die Rollenkategorien und kategorisierten Zugriffe auf Objekte anhand eines Krankenhaus-Szenarios mit Ärzten, Schwestern und verschiedenen Zugriffen auf Patientenakten und -berichten.

Security

Roles: $\mathbb{F}$ ROLE
SeniorityRoles: $\mathbb{F}$ ROLE
FunctionalRoles: $\mathbb{F}$ ROLE
ContextualRoles: $\mathbb{F}$ ROLE
Contexts: $\mathbb{F}$ CONTEXT_TYPE
RoleContext: ROLE $\rightarrow\!\!\!\!\!\!\rightarrow$ CONTEXT_TYPE
RoleInheritance: ROLE $\leftrightarrow$ ROLE
RoleSeniority : ROLE $\leftrightarrow$ ROLE
Policy: $\mathbb{F}$ OPERATION_POLICY
AssetHierarchy: ASSET_CATEGORY $\leftrightarrow$ ASSET_CATEGORY
Assets: $\mathbb{F}$ ASSET_CATEGORY
Operations: $\mathbb{F}$ OPERATION
Users: $\mathbb{P}$ USER
UserRoles: USER $\rightarrow\!\!\!\!\!\!\rightarrow$ $\mathbb{P}$ ROLE_INSTANCE
RoleType: ROLE_INSTANCE $\rightarrow$ ROLE
AssetCategory: ASSET_INSTANCE $\rightarrow$ ASSET_CATEGORY
AssetContext: ASSET_INSTANCE $\rightarrow$ $\mathbb{P}$ CONTEXT_INSTANCE
ContextType: CONTEXT_INSTANCE $\rightarrow$ CONTEXT_TYPE
RoleInstanceContext : ROLE_INSTANCE $\rightarrow\!\!\!\!\!\!\rightarrow$ CONTEXT_INSTANCE

CheckAuthorisation

ΞSecurity
o?: OPERATION
u?: USER
ai?: ASSET_INSTANCE
r?: ROLE_INSTANCE

---

let InheritedAssets == ran ({AssetType ai?} $\lhd$ (AssetHierarchy *));
InheritedRoles == ran ((RoleType (UserRoles u?))
$\lhd$ (RoleHierarchy *))

• $\exists$p: Policy | o? $\in$ p . Operations $\wedge$ p . Asset $\in$ InheritedAssets
• $\exists$rs : p .RoleSets
• $\forall$fr: rs | fr $\in$ FunctionalRoles • fr $\in$ InheritedRoles $\wedge$
$\forall$sr: rs | sr $\in$ SeniorityRoles • sr $\in$ UserRoles $\wedge$
$\forall$cr: rs | cr $\in$ ContextualRoles •
($\exists$uri: UserRoles u? • (RoleType uri = cr $\wedge$ ($\exists$ci: AssetContext ai?
• (RoleInstanceContext uri = ci
$\wedge$ ContextType (RoleInstanceContext uri) = RoleContext cr
$\wedge$ ContextType ci = RoleContext cr))))

CheckAuthorisation dient der Überprüfung, ob ein Benutzer eine Operation auf einer Asset Instanz durchführen darf. (z.B. ein Doktor, der auf die Patientenakte von John Smith zugreifen möchte)

5. Abschluss

Der Gedanke des Rollenbasierten Sicherheitsmodells ist nicht so neu wie man denken mag, ist der Ansatz des Gruppierens von Benutzern (nach Rollen) doch intuitiv auch schon in älteren Softwaresystemen verfolgt worden. Die Formalisierung der Rollenbasierten Sicherheitsmodelle wird von verschiedenen Seiten oft mit ähnlichen, zum Teil aber im Detail doch sehr unterschiedlichen Ansätzen begangen, wie der Vergleich aus den hier erläuterten Arbeiten [NIST] und [CIN] offenbart.

6. Quellen

[ECKERT] Claudia Eckert, „IT-Sicherheit“, Oldenbourg Wissenschaftsverlag GmbH, 2. Auflage 2003

[NIST] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, R. Richard Kuhn, „Proposed NIST Standard for Role-Based Access Control“, 2001 http://csrc.nist.gov/rbac/rbacSTD-ACM.pdf

[CIN] R. Crook, D. Ince, B. Nuseibeh, „Towards an Analytical Role Modelling Framework for Security Requirements“, Proceedings of 8th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ-02), Essen, Germany, 9-10 September 2002

[SPIVEY] J.M. Spivey, „The Z – Notation A Reference Manual“, Second Edition, Prentice Hall, 1992