Maßnahmen zum Schutz der Sicherheitspolitik bei der RBAC-Modellierung insbesondere bei der Verwendung von eXtreme Role-Engineering

Thomas Hildmann
Technische Universität Berlin - tubIT IT-Service-Center
Sekr. EN 50, Einsteinufer 17
10587 Berlin
thomas.hildmann@tu-berlin.de

Zusammenfassung

Das Prinzip der Delegation von Aufgaben wird in großen Organisationen angewandt, um der Tatsache Rechnung zu tragen, dass es nicht möglich ist, alle Aufgaben von einer Person bewältigen oder alle Personen alle Aufgaben in gleichem Maße durchführen zu lassen. Der Einsatz einer verteilten Rechteverwaltung für die IT-Infrastruktur ist hierbei ein logischer Schritt, der eine flexible, schnelle und transparente Anpassung der Prozesse innerhalb einer Organisation von der Basis her ermöglicht. Aus den Erfahrungen heraus, die die TU Berlin mit ihrem rollenbasierten IDM-System mit über 4.000 Mitarbeiterinnen und Mitarbeitern sowie mit ca. 30.000 Studierenden als Nutzer eines webbasierten Portalsystems und einigen weiteren Anwendungen gewinnen konnte, wurde eXtreme Role-Engineering entworfen. Das eXtreme Role-Engineering Verfahren bringt Methoden der agilen Softwareentwicklung mit dem Role-based Access Control (RBAC) zusammen, um so eine schnelle, unkomplizierte Rollendefinition zu ermöglichen. Dabei besteht die Gefahr, dass bei der dezentralen Rollendefinition Fehler unterlaufen. Im RBAC-Umfeld sind zahlreiche Maßnahmen zum Schutz bei der Modellierung bekannt. Diese werden im Folgenden zusammengestellt und insbesondere auf ihre Anwendbarkeit auf das eXtreme Role-Engineering Verfahren betrachtet. Dieser Beitrag kann aber auch als Überblick über Schutzmechanismen in RBAC allgemein gelesen werden.

”Man fällt nicht über seine Fehler. Man fällt immer über seine Feinde, die diese Fehler ausnutzen.” - Kurt Tucholsky, ”Bauern, Bonzen, Bomben”, in ”Die Weltbühne”, 7. März 1931, S. 496

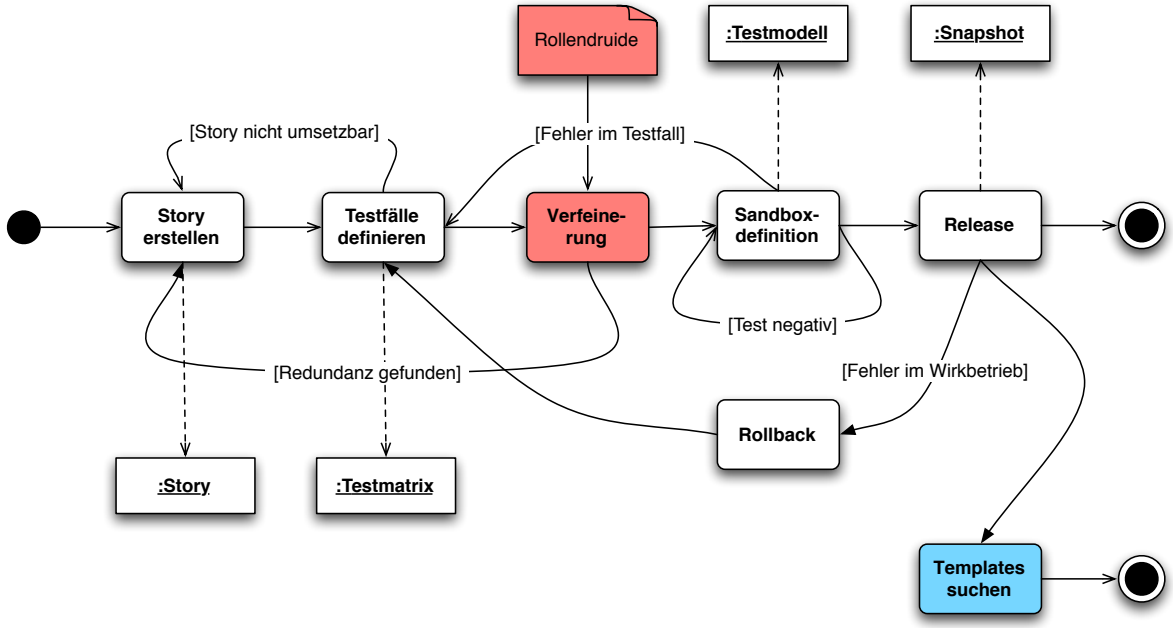

Abbildung 1: Das eXtreme Role-Engineering Vorgehensmodell

1 Einleitung

RBAC wurde aus den Schwächen der weit verbreiteten Zugriffskontrollmodelle MAC (mandatory access control) und DAC (discretionary access control) entwickelt [7]. In der Einleitung zu [8] heißt es sinngemäß: ”Aus einer Unternehmenssicht hat Zugriffskontrolle das Potential, den optimalen Austausch und das Verteilen von Ressourcen zu fördern; es hat jedoch auch das Potential, Benutzer zu frustrieren, große administrative Kosten zu erzeugen und unautorisierte Enthüllungen oder Fälschungen von sensiblen Informationen zu begünstigen.”

DAC-Systeme werden heutzutage ganz selbstverständlich verwendet. Die Zugriffsrechte auf UNIX-Dateisystemen aber z.B. auch auf Netzwerklaufwerken in Windows NT- oder UNIX-artigen Umgebungen wie Linux, MacOS X, Solaris usw. basieren auf der Verwaltung der Rechte durch die Benutzer. Der Eigentümer einer Datei kann anderen Benutzern bestimmte Rechte an seiner Datei anbieten.

In RBAC-Systemen wird der Zugriff über Rollenmitgliedschaften und Zuweisungen von Rechten zu Rollen definiert. Der Wunsch, die Administration dieser Rechte und Mitgliedschaften ebenfalls zu verteilen, ist naheliegend [17] und wurde in unterschiedlichen Projekten verfeinert und umgesetzt [6], [13]. Das eXtreme Role-Engineering Verfahren [11] unterstützt die verteilt arbeitenden Rollenadministratoren bei der Verwaltung ihres Rollenmodellteils, in dem Methoden der agilen Softwareentwicklung [16] mit klassischem Role-Engineering [5] kombiniert werden.

Schon bald nach der Systematisierung des RBAC-Gedankens [7] wurden die ersten Ansätze zum Role-Engineering veröffentlicht [4]. Das Thema ist auch heute noch aktuell, wie jüngere Veröffentlichungen [15] und das Erscheinen eines Buches [5] zum Thema deutlich zeigen.

An der TU Berlin arbeitet man an einer dezentralen Rollenadministration [12]. Für die Benutzer in den Fakultäten, Verwaltungs- und Forschungseinrichtungen müssen jedoch Werkzeuge geschaffen werden, die leichter zu handhaben sind, als das klassische Role-Engineering, jedoch über eine Systematik zu einer konsistenten Modellierung führen. Vor diesem Hintergrund wurde das eXtreme Role-Engineering Verfahren entwickelt, das versucht, Ideen aus der agilen Softwareentwicklung für die Erstellung eines Rollenmodells zu adaptieren. Es verfolgt dabei den Ansatz, die Rollen jeweils für Anwendungsfälle zu definieren und das Modell Schritt für Schritt durch Hinzufügen weniger Rollen aufzubauen.

Es ist davon auszugehen, dass das Verfahren nicht nur für den Hochschulbereich interessant ist. Es ist für alle Bereiche sinnvoll, in denen RBAC eingesetzt wird (oder werden sollte), die eine verteilte, dezentrale Administration benötigen, wobei die Administration des RBAC-Systems jedoch nicht im Mittelpunkt der Administratoren liegt (”Gelegenheitsrollenmodellierer”).

Das extreme Role-Engineering Verfahren soll durch geeignete Softwarewerkzeuge (im Folgenden Rollendruide genannt) unterstützt werden. Abbildung 1 zeigt einen Überblick über das Verfahren:

Erstellung einer Story: Der Verwalter der Organisationseinheit beschreibt anhand von Beispielen, welche Anwendung für welche Personengruppe zugänglich gemacht werden soll.

Definition von Testfällen: Der Verwalter wählt geeignete Testpersonen aus seiner Einheit. Über eine GUI (xreUnit) definiert er in einer Matrix, welche Personen welche Zugriffe auf die Anwendungen bekommen sollen.

Rollenzuweisung in einer Sandbox: In einer Sandbox kann der Verwalter ein “Was wäre wenn”-Szenario aufbauen und prüfen lassen, ob seine Zuweisungen im Organisationsteil der Testmatrix entspricht.

Release: Die in der Sandbox definierten Änderungen werden auf den Produktivserver kopiert. Die jeweiligen Zustände vor und nach der Änderung werden archiviert. Ein Rollback ist jederzeit möglich.

Eine dezentrale Administration des Zugriffsmodells hat viele Vorteile: Das entscheidende Argument für eine Administration in den Untereinheiten der Organisation ist die schnelle und flexible Reaktion auf Veränderungen, die heutzutage immer wichtiger wird. Ebenso wichtig erscheint, dass es den lokalen Rollenadministratoren so eher möglich sein wird, die Prozesse in der Abteilung korrekt im Rollensystem abzubilden. Oft ist das abstrakte Bild der Organisation bezüglich der Personen-Aufgabenzuordnung in den Untereinheiten zu ungenau, um Zugriffsmodelle für effizientes Arbeiten zu ermöglichen.

Auf der anderen Seite besteht die berechtigte Sorge, dass durch die verteilte Administration Sicherheitslücken entstehen könnten. Diese Sorge wird von allen Nutzergruppen geteilt. Die Leitung befürchtet eine Umgehung der allgemeinen Sicherheitspolitik, die Endnutzer befürchten ein Ausspähen sensibler Informationen sowohl intern, als auch durch Dritte. Auch die Rollenadministratoren selbst stehen unter sehr hohem Druck, keine Fehler zu machen.

Tabelle 1: Seperation-of-Duty-Typen (SoD)

SoD-Typ	Wirkzeitpunkt	Wirkungsweise
SSD	Rollenzuweisung	Grundsätzliche Vermeidung von Interessenkonflikten.
DSD	Login / Aktivierung	Transparente Ausübung *eines* Posten.
operational	Aktion	Verantwortungsverteilung: 4-Augen-Prinzip.
historisch	Objektzugriff	Bindung der SoD an Objekte (Tagging).

2 Schutzmechanismen in RBAC-Systemen

Für die Minimierung von Risiken in RBAC-geschützten Umgebungen stehen verschiedene Maßnahmen zur Verfügung, die im Folgenden mit ihrer jeweiligen Wirkungsweise aufgezeigt werden.

Viele dieser Maßnahmen sind für RBAC-Systeme optional. In [18] wird ein Basismodell $RBAC_0$ definiert, in dem es weder Nebenbedingungen (Constraints) noch Hierarchien gibt. Erst das $RBAC_3$ Modell vereinigt alle diese Funktionalitäten. Vor dem Hintergrund der Sicherheit wird hierdurch klar, welchen Stellenwert die Wahl eines geeigneten Zugriffsystems mit geeigneten RBAC-Implementierungen hat.

2.1 Verteilung von Verantwortlichkeiten

Ein mächtiges Konzept zum Schutz der Ressourcen aber auch zur Entlastung der Benutzer ist die Verteilung der Verantwortlichkeiten (SoD, Separation-of-Duty) [7]. Die Konzepte hierzu sind aus der physischen Welt in die IT-Modelle übernommen worden. So sind papierbasierte Arbeitsabläufe hinreichend bekannt, bei denen eine zweite oder dritte Person nach Prüfung des Sachverhalts ebenfalls unterzeichnen muss (Vier-Augen-Prinzip) oder wo sich die simultane Bekleidung von Ämtern ausschließt. Das in der Literatur immer wieder zitierte Beispiel ist hier der Kassenwart, der nicht seine eigene Kasse prüfen darf.

Bedürfen kritische Operationen der Zusammenarbeit mehrerer Benutzer [3], können so Kompromittierungen des Systems durch falsche Rollenzuordnungen in den meisten Fällen verhindert werden.

Die Verteilung der Verantwortlichkeiten kann in RBAC-Systemen auf sehr unterschiedlichen Ebenen umgesetzt werden. Man unterscheidet hier u.a. statisches (SSD) und dynamisches (DSD) SoD.

Das SSD greift bei der Zuordnung von Personen zu Rollen. Ist eine Person Mitglied in einer bestimmten Rolle, die die Mitgliedschaft in einer anderen Rolle ausschließt, dann kann die Person der zweiten Rolle nicht zugewiesen werden.

Im Gegensatz hierzu greift das DSD erst bei der Aktivierung der Rolle, also dann, wenn sich der Benutzer mit der Rolle anmeldet. An dieser Stelle muss sich der Benutzer entscheiden, welche der Rollen er/sie benutzen will oder umgangssprachlich: ”Welchen Hut er bei der Erledigung seiner Aufgaben auf hat.” Es ist jederzeit möglich, sich mit einer der Rollen ab- und mit einer anderen wieder anzumelden.

Der DSD-Mechanismus wird in dem an der TUB eingesetzten System konsequent umgesetzt. Bei Auswahl einer Anwendung aus dem Portalsystem wird der Benutzer jeweils dazu aufgefordert, die Rolle auszuwählen, in der er das System benutzen möchte. Steht ihm nur eine Rolle zur Verfügung, wird diese automatisch aktiviert.

SSD-Mechanismen greifen an der TUB bei der Zuordnung so genannter statischer Rollen oder Standardrollen, die automatisch aus den Tätigkeiten der Personen abgeleitet und aus der Datenbank der Personalverwaltung bezogen werden. Solche statischen Rollen können z.B. ”Professor/in”, ”wissenschaftlicher Mitarbeiter/in” etc. sein. Werden in der Rollenhierarchie nun Rollen an diese statischen Rollen gebunden, kann implizit ausgeschlossen werden, dass eine Person in Besitz von bestimmten Kombinationen von Rollen kommen kann.

Das operational SoD wiederum basiert auf der Tatsache, dass zur Durchführung einer bestimmten kritischen Funktion die Rechte mehrerer Rollen notwendig sind. Wird ferner über die o.g. Mechanismen dafür gesorgt, dass eine Person nicht gleichzeitig im Besitz aller nötigen Rollen ist, kann damit sichergestellt werden, dass die Ausführung dieser kritischen Funktion nur durch mehrere Personen autorisiert werden kann.

Solche Mechanismen sind an der TUB mit Kopplung des Rollensystems mit einer Workflow-Komponente geplant, sind jedoch auch zur Zeit z.B. bei der Beantragung von Subdomains für Webauftritte implementiert. Während die Leiter einer Organisationseinheit der Universität einen Webauftritt beantragen können, müssen Mitarbeiter der Netzwerkabteilung prüfen, ob die Subdomain zur Verfügung gestellt werden kann. Mit Einrichtung des Webauftritts wird die Vergabe der Rechte auf dem neuen Webauftritt allerdings wieder automatisch an die Organisationseinheit gegeben. Hier zeigt sich deutlich die Verteilung der Verantwortlichkeiten, die in diesem Prozess anders abgebildet ist, als in anderen Systemen, wo der Administrator für Einrichtung und Rechtevergabe z.B. universalverantwortlich ist.

In der Literatur ist ferner von ”history and object-based SoD” die Rede. Diese Art der Verantwortlichkeitsverteilung ist jedoch nicht weit verbreitet. Sie basiert darauf, dass eine Person zwar alle Rollen und damit Rechte gleichzeitig besitzen kann, sie jedoch nicht auf ein und das selbe Objekt anwenden darf. So darf ein Kassenwart auch Kassenprüfer sein. Wenn er jedoch eine bestimmt Kasse geführt hat, darf er nicht genau diese Kasse auch prüfen.

Grundsätzlich ist SoD sowohl für die Umsetzung einer Sicherheitspolitik ein mächtiges Werkzeug, als auch für die Benutzer. Verhindert man die ”Allmacht” von Administratoren, so können diese auch nicht in Verdacht geraten, unautorisiert tätig geworden zu sein. Verteilung von Verantwortung geht auch immer mit Verteilung von Aufgaben und damit von Arbeit einher. SoD kann also dazu dienen, die Arbeitsverteilung im Sicherheitsmodell der Organisation abzubilden.

2.2 Administrative Rollen

Die Administration des Rollensystems kann selbst wieder über ”administrative Rollen” gesteuert werden [17]. Diese Rollen besitzen Rechte, die die Veränderung des Rollenmodells zulassen (Abbildung 2). Insbesondere wird die Zuordnung von Benutzern zu Rollen häufig verteilt, weil dies in RBAC-Systemen den größten Aufwand im Betrieb darstellt.

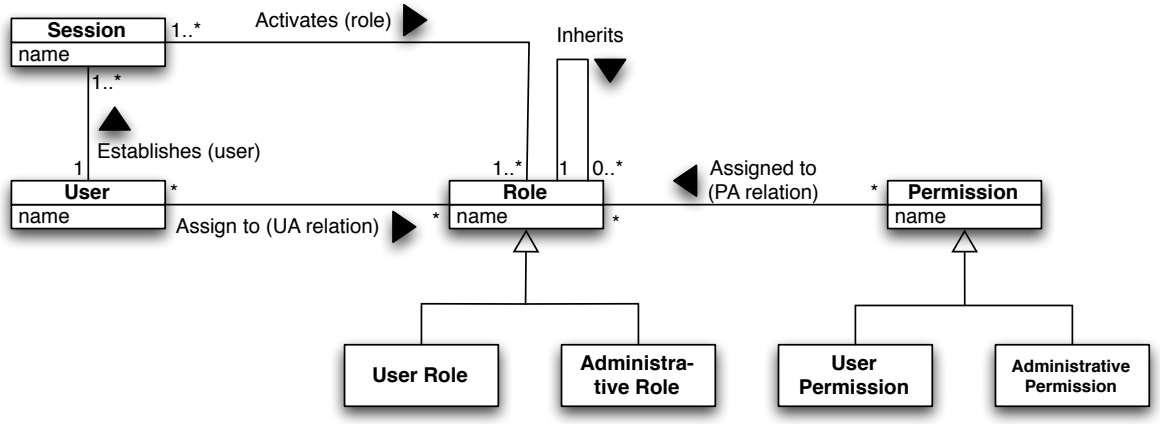

Abbildung 2: Konzeptionelles Klassenmodell für RBAC nach [20]

Die Verteilung dieser Aufgaben hat ferner den Vorteil, dass Blockierungen verhindert werden können, die dazu führen könnten, dass Personen auf Grund fehlender Rechte nicht arbeitsfähig sind.

Es gibt unterschiedliche Modelle, die den Umfang der administrativen Rechte definieren, d.h. die Frage, welche Teile des Modells von welchem Administrator verwaltet werden dürfen.

Die Vergabe von administrativen Rollen hat verschiedene Vorteile: Zum einen kann die Rolle an Nebenbedingungen geknüpft werden. So gibt es z.B. an der Universität die Rolle ”Dekan”. Der Dekan wird von einem Gremium gewählt. Nach Wahl eines neuen Dekans, verliert der amtierende Dekan seine Rolle und der neue Dekan wird Mitglied. Damit gehen dem Ex-Dekan alle administrativen Rollen, die an der ”Dekan”-Rolle hängen automatisch verloren. Die Reduzierung des Verwaltungsaufwands an dieser Stelle erhöht weiterhin die Sicherheit. Vom Zeitpunkt des Entzugs der Rolle steht der ausgeschiedene Dekan ferner nicht mehr in der Verantwortung für seinen Bereich.

2.3 Allgemeine Festlegungen durch Nebenbedingungen

Nebenbedingungen (Constraints) können in RBAC-Modellen dazu verwendet werden, im Modell einige Grundregeln zu verankern. Nebenbedingungen können theoretisch an allen Stellen im Modell definiert werden, sofern das von der Implementierung unterstützt wird:

- Bei der Person-Rollen-Zuordnung,
- der Rollen-Rollen-Zuordnung (siehe Abschnitt 2.1),
- bei der Rollen-Rechte-Zuordnung
- aber auch bei der Aktivierung von Rollen
- oder der Nutzung von Rechten.

Die Nebenbedingungen können dazu genutzt werden, um Sachverhalte auszudrücken, die mit der bloßen Zugehörigkeit einer Rolle nicht definierbar sind. Sie können aber auch dafür verwendet werden, um sicherzustellen, dass die Organisation handlungsfähig bleibt. So können Regeln definiert werden, die z.B. verhindern, dass bestimmte Rollen unbesetzt sind, dass sie nur ein oder x-mal besetzt werden können oder was im Falle einer Nichtbesetzung geschieht.

An der TU Berlin gibt es beispielsweise eine Regelung, die besagt: Ist die Rolle des Rollenverwalters nicht besetzt, so kann der Rollenverwalter der strukturell übergeordneten Einheit die Untereinheit verwalten, bis dort ein Verwalter eingesetzt ist. Hier greift eine Nebenbedingung auf eine Rollen-Rollen-Zuordnung. Der Rolle Rollenverwalter sind wiederum andere Rollen zugeordnet, die das Verwalten ermöglichen. Grundsätzlich besitzt jeder Rollenverwalter auch die Rollen aller Untereinheiten. Diese Zuordnung geht jedoch verloren, sobald auf einer Unterebene ein anderer die Verwalterrolle besitzt. Dem Verwalter der Untereinheit ist es nun freigestellt, die Verwaltungsrolle wieder explizit an den übergeordneten Verwalter zu übertragen. Faktisch ist dies jedoch selten der Fall.

Über eine einfache Nebenbedingung kann so beispielsweise verhindert werden, dass eine Untereinheit handlungsunfähig wird. Auf die gleiche Weise können über Nebenbedingungen auch andere Grundregeln der Sicherheitspolitik verankert werden. Diese können dann auch nicht durch fehlerhafte Zuordnungen im Modell verletzt werden.

Eine Sonderform der Nebenbedingung ist die Verteilung der Verantwortlichkeit (vergl. 2.1).

2.4 Strukturierung des RBAC-Modells

Ein weiteres mächtiges Werkzeug des RBAC ist die Strukturierung des Modells. Dabei gibt es verschiedene Strukturierungsmittel: Die Rollen-Rollen-Zuordnung ist eine Methode. In [14] wird ausdrücklich davor gewarnt, das Organigram einer Firma in eine Rollenhierarchie zu übernehmen. Grund ist die Tatsache, dass die organisatorische Innen- und Außendarstellung sehr unterschiedlich motiviert ist, eine Vielzahl von Zielen gerecht werden muss und in den seltensten Fällen über Arbeitsabläufe und damit verbundene Zugriffsrechte erstellt wurde.

An der TU Berlin werden drei Prinzipien implementiert: Generalisierung, Aggregation und Supervision. Als Basis dient das Kostenstellenverzeichnis (KST-Verzeichnis) der Organisation. Dieses definiert Einheiten und weist Personen eindeutig ein oder mehreren Kostenstellen zu. Es handelte sich hierbei um eine der vollständigsten Abbildungen der Universität, weshalb sie als Grundlage geeignet ist.

Die Rollenzuordnung findet über ein Aggregations-Modell statt. Einer Einheit aus dem KST-Verzeichnis werden eine Menge von Rollen gemäß ihrer Verantwortlichkeit zugewiesen. Die Einheit verteilt dann die Rechte mittels Delegation an die ausführenden Personen. Sobald eine Person Mitglied in einer Rolle wird oder ihm die Mitgliedschaft entzogen wird, wird ihr diese Änderung ihrer Aufgabe (denn mit der Zuweisung der Rechte sind jeweils Verpflichtungen zur Ausführung verbunden) mitgeteilt.

Neben der Delegation gibt es auch die Vertretung [1]. Hierbei handelt es sich um eine temporäre, nicht übertragbare Weitergabe der Rechte z.B. im Urlaubs- oder Krankheitsfall. Inhaltlich kann eine Vertretungsrolle ferner bedeuten, dass mit der Rolle keine unmittelbaren Verpflichtungen verbunden sind, sondern mittelbar die Bereitschaft im Bedarfsfall.

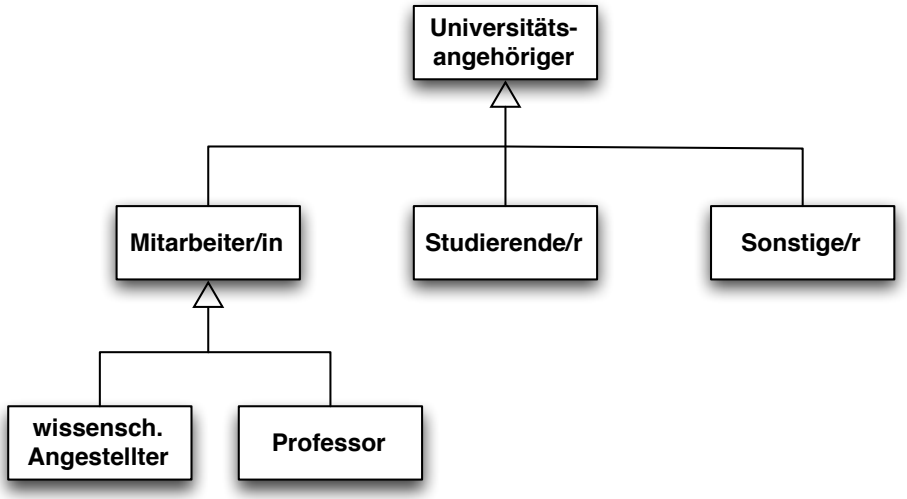

Abbildung 3: Beispiele für Generalisierung

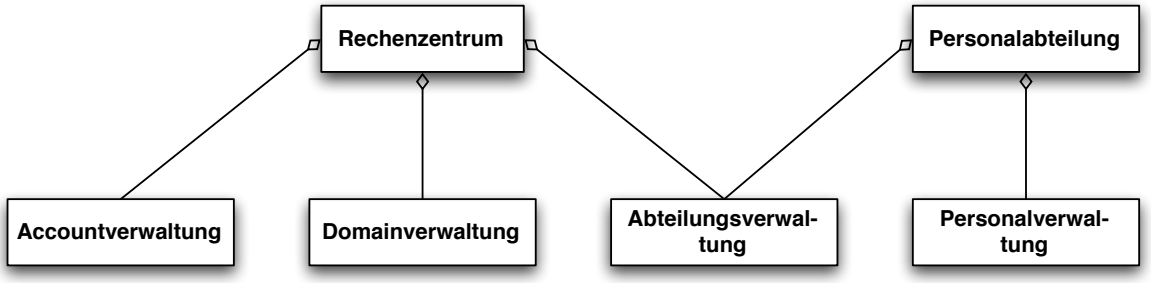

Abbildung 4: Beispiele für Aggregation

Die drei Strukturierungstypen werden von Moffett und Lupu in [14] wie folgt beschrieben:

Generalisierung: Die Generalisierung ist bekannt als eine ”Ist ein”-Hierachie. Sie wird häufig in RBAC-Beispielen verwendet (Ein ”wissenschaflicher Mitarbeiter” ist ein ”Mitarbeiter”, ist ein ”Universitätsmitglied”). Dabei können einige Rollen abstrakt sein. D.h. ”Universitätsmitglied” hat keine direkten Mitglieder, sondern setzt sich aus ”Mitarbeitern”, ”Studierenden” und ”Sonstigen Universitätsangehörigen” zusammen (siehe Abbildung 3).

Aggregation: Die Aggregation ist als eine ”Teil von”-Hierarchie bekannt. Gemäß Moffett und Lupu sollte eine solche Hierarchie über die Relationen ”Verantwortlich für” und ”Führt aus” definiert werden. Die verantwortliche Einheit kann eine Aufgabe entweder selbst ausführen oder an jemand anderen delegieren. In der Realität ist das oft aber nicht zwingend eine Untereinheit. Eine beliebige Einheit ist gemäß diesem Prinzip über die Menge ihrer Verantwortlichkeiten auf der einen Seite und ihren Ausführungen auf der anderen Seite definiert (siehe Abbildung 4).

Supervision: Supervision oder ”Leitung und Überwachung” ist ein typisches Modell, wie es in Organigrammen zu finden ist. Einheiten sind hier über ihre Leitung definiert, die

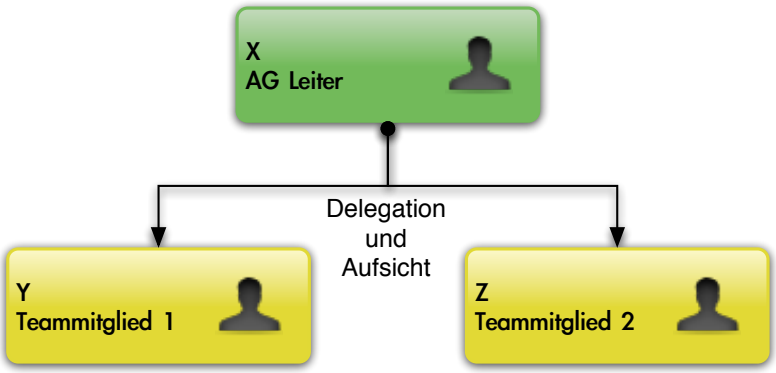

Abbildung 5: Beispiele für Supervision

in der Regel die Aufsicht und Koordination über die untergeordneten Einheiten und Personen besitzt. Wie bereits erwähnt, werden solche Hierarchien selten in Hinblick auf tatsächliche Rechtevergaben entworfen. Zum Teil hängen Zugehörigkeiten hier von der räumlichen Nähe, der Historie oder auch von persönlichen Gegebenheiten ab. Auf der anderen Seite kann es auf Grund der formalen Verantwortlichkeit sinnvoll sein, die ”Vorgesetztenhierarchie” abzubilden (siehe Abbildung 5).

Bezogen auf die Sicherheitsmaßnahmen gegen eine fehlerhafte Rollendefinition bedeuten diese Techniken, dass eine Organisationseinheit grundsätzlich nur die Rollen und Rechte delegieren kann, die aus dem Pool ihrer eigenen Verantwortlichkeit stammen. Das sind naturgemäß Rechte, für die die Einheit nötige Fachkenntnis besitzt. Vertretungen können von den Ausführern selbst für die jeweils geeignete Aufgabe definiert werden. Die de facto Vertretungen aus der physischen Welt können hierbei unkompliziert in das Rollenmodell übernommen werden.

2.5 Unterstützung durch Experten

Im Bereich der Software-Architektur gilt die Anwendung von Mustern als ein Mittel zur Bewältigung der Komplexität und Möglichkeit zur Nutzung von Expertenwissen [9, 2]. Muster können auch auf Rollensysteme angewandt werden [10]. Im Rahmen des eXtreme Role-Engineerings (XRE) ist der Einsatz von Mustern vorgesehen. Hierbei handelt es sich um Zusammenstellungen von Rollen und Rechten, die für typische Aufgaben jeweils auf eine Organisationseinheit angewendet werden. So gibt es sowohl in Fachgebieten, wie auch in Verwaltungseinheiten an der Universität typischerweise eine oder mehrere Personen, die mit dem Erwerb von Soft- und Hardware betraut sind. Eine geeignete Kombination aus Rollen liegt in diesem Fall als Template vor. Wird dieses Template in einer Einheit angewendet, so werden die Rollen bezogen auf diese Einheit vergeben (also z.B. Besteller für ein bestimmtes Fachgebiet und den entsprechenden Konten). Diese Templates werden von Rollenexperten aus Statistiken abgeleitet, die das XRE-Verfahren erzeugt. Werden also von unterschiedlichen Organisationseinheiten immer wieder sehr ähnliche Kombinationen von Rechten zusammengestellt, so prüfen die Experten an Hand dieser Vorgaben, wie ein geeignetes Template hierfür aussehen würde und stellen dies zur Verfügung. Der im XRE eingesetzte Rollenfindungsdruide versucht

an Hand der Vorgaben durch Testfälle, die vom Rollenadministrator der Einheit vorgegeben sind immer auch Templates zu finden, die möglichst nahe an den Vorstellungen liegen. Die spezialisierten Rollen können dann von den Templates abgeleitet und vom Rollenadministrator der Einheit geeignet erweitert werden.

Selbstverständlich würden mindestens in der Statistik auch Fehlkonfigurationen auffallen, wenn sie häufiger vorkommen. Das Template-Review der Rollenexperten verhindert so zumindest häufig gemachte Fehler. Diese können dann in Folge über Nebenbedingungen verhindert werden, wogegen ein sinnvoller Gegenvorschlag über Templates zur Verfügung gestellt werden könnte.

Wie bei praktisch allen sicherheitsrelevanten Methoden ist die Dokumentation, die Schulung und der Support durch die Experten von größter Wichtigkeit. Eine, wenn nicht *die* entscheidende Angriffsfläche eines Sicherheitssystems ist der Mensch und dessen Beeinflussung. Wie jedes Sicherheitssystem, ist auch RBAC gegen Social-Engineering anfällig [19]. Dem kann nur mit einem möglichst hohen Wissensstand (Sensibilisierung) der Mitarbeiterinnen und Mitarbeiter begegnet werden. Dazu gehört selbstverständlich auch eine geeignete Benutzerführung der verwendeten Software, ein umfassendes Lehr-, Auffrischungs- und Weiterbildungsangebot sowie ein kompetenter Support, der wie selbstverständlich im Zweifelsfall Auskunft geben kann.

2.6 Schutz vor zu wenigen Rechten

Die Folgen von Rechteüberschreitungen oder von der fehlerhaften Autorisierung der falschen Person, d.h. der Zuweisung von zu vielen Rechten an diese Person ist intuitiv verständlich und wird gerne und ausgiebig diskutiert. Vergessen wird dabei gerne, dass eine Vergabe unnötiger Rechte zunächst nicht zwangsweise zu einer Ausnutzung dieser Rechte und damit zu einem Schaden führen muss. Hingegen kann die Vergabe zu weniger Rechte unmittelbar dazu führen, dass Prozesse in einer Organisation zum Erliegen kommen. Verantwortungsbewusstes Sicherheitsmanagement muss daher nicht nur die Einschränkung von Rechten bedenken, sondern auch die Sicherstellung von hinreichend vielen Rechten. Leicht lassen sich Fälle konstruieren, in denen Angriffe gegen die IT-Infrastruktur abgewendet werden könnten, wenn geschultes Personal mit hinreichend vielen Rechten ausgestattet ist.

Insbesondere für Ausfälle von Personen z.B. durch Krankheit, Dienstreisen, Urlaub usw. müssen Vertreter mit hinreichend vielen Rechten ausgestattet sein oder im Notfall damit spontan ausgestattet werden können. In [14] wird daher von ”Back-up Roles” gesprochen und in diesem Zusammenhang vom Prinzip ”Passwort im Umschlag” abgeraten, weil hier die Nachvollziehbarkeit (Transparenz) der Aktionen nicht mehr gegeben ist. Später lässt sich nicht mehr nachweisen, wer den Umschlag geöffnet und was mit dem Inhalt gemacht wurde. Stattdessen wird eine Aufwärtsvererbung von Zugriffsrechten auf genau einer Ebene propagiert. Im Bedarfsfall kann der Vorgesetzte die Blockierung auflösen oder durch Delegation dafür sorgen.

Da an der TU Berlin jede Organisationseinheit mit einer geeigneten Menge von Rechten ausgestattet ist, die von der Einheit selbst verwaltet werden kann, muss vor allem sichergestellt werden, dass immer mindestens ein handlungsfähiger Rollenadministrator zur Verfügung steht, der, wenn nötig, Rollen delegieren kann.

3 Zusammenfassung

Die Liste der Maßnahmen scheint lang, erhebt jedoch keinen Anspruch auf Vollständigkeit. Das Grundprinzip von RBAC besteht zwar in der Zuweisung von Rechten zu Rollen und in der Mitgliedschaft von Personen in Rollen, die Durchsetzung der Sicherheitspolitik kann jedoch an sehr unterschiedlichen Positionen im RBAC-Modell verankert und durchgesetzt werden. Das eXtreme Role-Engineering behandelt die Definition von Rollen, greift dabei jedoch bereits auf ein bestehendes RBAC-Modell zurück, in dem eine Struktur, geeignete Rollen-Muster, Anwendungsrollen mit sinnvollen Mengen an Rechten sowie Constraints vorgegeben sind. Diese Vorgaben sowie die Prinzipien der Verteilung von Verantwortlichkeiten (Seperation of Duty), Delegation und Stellvertreterschaften bilden ein Maßnahmenpaket, das Fehler in der Rechtevergabe reduziert und Auswirkungen minimiert. Diese Maßnahmen entlasten den lokalen Rollenadministrator und machen so eine verteilte Administration erst praktikabel.

4 Ausblick

Im Rahmen meiner Dissertation werde ich das eXtreme Role-Engineering Verfahren weiter ausarbeiten und im Detail beschreiben. Ferner wird an der TU Berlin ein Satz an Werkzeugen zur Unterstützung des Verfahrens entwickelt. Dabei hoffen wir einige Teile davon allgemein zur Verfügung stellen zu können. Sobald es möglich ist, einen soliden Erfahrungsbericht zum Einsatz von eXtreme Role-Engineering in der TU Berlin zu verfassen, werden wir diesen veröffentlichen.

Literaturverzeichnis

[1] Ezedin Barka and Ravi Sandhu. Framework for role-based delegation models. Technical report, Laboratory of Information Security Technology, Information and Software Engineering Department George Mason University, Fairfax, VA 22030, USA, 2000.

[2] Frank et.al. Buschmann. Pattern-orientierte software-architektur. Addison-Wesley, Bonn, Reading, Massachusetts, ..., 1998.

[3] D. D. Clark and D. R. Wilson. A comparison of commercial and military computer security policies. In *IEEE Symposium of Security and Privacy*, pages pp. 184–194, 1987.

[4] Edward J. Coyne. Role engineering. In *ACM RBAC Workshop*, MD USA, 1996.

[5] Edward J. Coyne and John M. Davis. *Role Engineering for Enterprise Security Management*. Information Security and Privacy Series. Artech House, 2008.

[6] Fredj Dridi, Björn Muschall, and Günther Pernul. Administration of an rbac system. In *Proc. of the 18th IFIP International Information Security Conference (SEC 2003)*, Athens, Greece, Mai 2003. URL `http://ieeexplore.ieee.org/iel5/8934/28293/01265447.pdf`.

[7] David F. Ferraiolo, Janet A. Cugini, and D. Richard Kuhn. Role-based access control (rbac): Features and motivations. In *Proceedings of 11th Annual Computer Security Application ...*, 1995.

[8] David F. Ferraiolo, D. Richard Kuhn, and Ramaswamy Chandramouli. *Role-Based Access Control*. Artec House, second edition edition, 2007.

[9] Erich Gamma. Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software. Bonn, 1996.

[10] Thomas Gebhardt and Thomas Hildmann. Rollen als Schlüssel für B2B-Anwendungen. *DuD - Datenschutz und Datensicherheit*, 24 (2000) 10, 2000.

[11] Thomas Hildmann, Odej Kao, and Christopher Ritter. eXtreme Role Engineering: Ein neuer Ansatz zur Rechtedefinition und -vergabe. In *Proceedings der GI Tagung Sicherheit 2008*, 2008.

[12] Thomas Hildmann, Odej Kao, and Christopher Ritter. Rollenbasierte Identitäts- und Autorisierungsverwaltung an der TU Berlin. In *Proceedings 1. DFN-Forum Kommunikationstechnologien Verteilte Systeme im Wissenschaftsbereich*, 2008.

[13] Andreas K. Mattas, Ioannis K. Mavridis, and George I. Pangalos. Towards dynamically administered role-based access control. In *Proceedings of the ninth ACM symposium on Access control*, 2004.

[14] Jonathan D. Moffett and Emil C. Lupu. The uses of role hierarchies in access control. In *ACM RBAC Workshop*, Fairfax, VA, USA, October 1999.

[15] Aneta Poniszewska-Maranda. Role engineering of information system using extended rbac model. In *WETICE'05, IEEE*, 2005.

[16] Ralf Reißing. Extremes Programmieren. *Informatik Spektrum*, 23(2):118–121, April 2000.

[17] Ravi Sandhu, Venkata Bhamidipati, Edward Coyne, Srinivas Ganta, and Charles Youman. The arbac97 model for role-based administration of roles. *ACM Transactions on Information and System Security*, 2:105–135, 1999.

[18] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. *Computer*, Volume 29(2):38–47, February 1996.

[19] Bruce Schneier. *Secret & Lies - IT-Sicherheit in einer vernetzten Welt*. dpunkt.verlag, 2001.

[20] Michael E. Shin and Gail-John Ahn. Uml-based representation of role-based access control. In *Proceedings of the 9th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages pp. 195 – 200, 2000.