Firewalls und VPNs: Was kommt danach? Die Stellung des Access-Control-Managements in einem ganzheitlichen IT-Security-Ansatz

Andreas Wolf

NorCom Information Technology AG, München

Zusammenfassung Nach heutiger Kenntnis erfordert ein ganzheitlicher Ansatz zur Absicherung von IT-Systemen eines Unternehmens mehr als Firewalls und virtuelle private Netzwerke (VPNs). Vom *Layered Approach* der Netzwerksicherheit ausgehend, illustriert dieser Artikel die 4A-Dienste Authentifizierung, Autorisierung, Auditing und (Sicherheits-) Administration. Es werden Kriterien vorgeschlagen, mit denen ein geplantes *Access Control Management* (ACM) als Teil eines IT-Security-Ansatzes auf Vollständigkeit für den gewünschten Einsatzzweck überprüft werden kann. Es werden wichtige aktuell eingesetzte Konzepte des ACM gezeigt und als mögliches Anwendungsbeispiel wird wird das Produkt NGS BeanGuard vorgestellt. Der Artikel schließt mit einer Zusammenfassung und mit einem Ausblick auf künftige Entwicklungsrichtungen.

1 Trends der IT-Security

Wird heute über IT-Security gesprochen, so ist eine weitläufig verbreitete Meinung, dass mit dem Vorhandensein und der Pflege von Firewalls und virtuellen privaten Netzwerken (VPN) wenn nicht alles, dann zumindest aber sehr viel getan worden sei. Natürlich sind sowohl Firewalls als auch VPNs sehr wichtige und unverzichtbare Werkzeuge zum Schutz sensibler Unternehmensdaten. Sie schützen Firmennetzwerke zuverlässig vor externen Angreifern und sichern die Kommunikation mit Außendienstmitarbeitern ab.

Seit alters her liegt es in der menschlichen Mentalität, Mauern zu errichten, wenn Feinde abgewehrt werden sollen. Was passiert aber, wenn der „Feind“ bereits in der Burg, also innerhalb der Schutzmauern sitzt? Nach einer Studie [1] lassen sich derzeit weniger als 20% aller Attacken auf IT-Systeme auf unternehmens-externe Verursacher zurückführen. Der durchschnittliche Schaden eines solchen Angriffes von außen beläuft sich nach dieser Studie auf 57.000 US$. Dem gegenüber stehen die 80% der Angriffe, die innerhalb von Unternehmen gestartet wurden und die im Durchschnitt 2.500.000 US$ Schaden nach sich zogen. Diese Angriffe sind meist nicht von Firewalls und ähnlichen Schutzmechanismen abzuwehren. Im Gegenteil, diese können eine Sicherheit vorgaukeln, die so gar nicht vorhanden ist und die Wachsamkeit der Mitarbeiter einschläfern. Notwendig ist es, nicht nur äußerliche Merkmale von Ressourcenzugriffen im Netz, wie etwa die Benutzung von Ports, zu beobachten und zu regeln.

Es ist auch erforderlich, die Inhalte der im Netz zu bewegenden Daten zu inspizieren, was jedoch auf technische und rechtliche Probleme stößt: Was passiert mit verschlüsselten Daten? Woran werden schützenswerte Inhalte erkannt? Big Brother is watching you!

Ein möglicher Ausweg liegt nahe: Es muß die Berechtigung zu einem Zugriff auf eine Ressource bei dieser Ressource selbst geprüft werden. Nur berechtigte Benutzer erhalten die Erlaubnis zum Zugriff. Damit kann bei einem als sicher unterstellten Kommunikationsmedium kein Unbefugter nicht für ihn bestimmte Daten einsehen oder unerlaubte Aktionen auslösen.

Als Konsequenz dieser Erwägungen ist ein Management der Berechtigungen erforderlich, das die einzelnen in einem Intra- oder Extranet agierenden Personen bzw. ihre virtuellen und realen Identitäten verwaltet.

1.1 Allgemeine Anforderungen an eine Lösung zum Access Management

Hat ein Administrator es einmal geschafft, auf Basis der vorhandenen Systeme die Gesamtheit der Zugriffsrechte seines Unternehmens, die Security-Policy, zu konfigurieren, dann werden die ersten Änderungen der beteiligten Personen, Ressourcen und Rechte vermutlich nicht lange auf sich warten lassen. Es sind in diesem Zusammenhang zwei Problemstellungen zu untersuchen, die der *Administrierbarkeit* und die der *Wartbarkeit*.

- *Zur Administrierbarkeit*. Wie kann man sich vergewissern, ob ein Administrator wirklich die vorgegebene Policy realisiert hat und nicht eine andere, von der er nur glaubt, es sei die richtige? Der für die IT-Security verantwortliche Manager muss sich die Frage stellen, ob er bestimmte Sicherheitseigenschaften seiner EDV-Landschaft überprüfen kann, durch Tests oder durch Reviews der Implementierung. Es ergibt sich jedenfalls die Frage, wie die Administration der Security erfolgt. Gibt es nur isolierte Insellösungen? Sichern sie den Zugriffsschutz wirklich überall? Gibt es zentrale Datenhaltungen, beispielsweise für Benutzer? Wenn ja, werden diese auch von allen Applikationen benutzt?
- *Zur Wartbarkeit*. Hat ein Administrator eine Policy implementiert, welche Aufwände sind dann notwendig, wenn sich diese Policy ändert, neue Benutzer oder neue Ressourcen integriert werden müssen? Ist für diese Anpassung Spezialwissen notwendig, vielleicht sogar noch unterschiedliches für jede Applikation? Sind immer Mitarbeiter mit diesem Spezialwissen verfügbar? Ist der Arbeitsaufwand so zu gestalten, dass Änderungen an der Policy immer zeitnah erfolgen?

Wird diese Fragen beispielhaft anhand des weit verbreiteten Apache-Webservers betrachtet, dann läßt sich feststellen, dass hier viele Schutzmechanismen vorgesehen sind, die auch oft sehr heterogen benutzt werden. Man findet Administrationsmöglichkeiten für Rechte und Benutzer in verteilten oder zentralen Files, bis hin zu diversen Datenbank-Adaptern, die an unterschiedlichen Stellen in Zugriffsprüfungen eingreifen. Zur Bedienung des Webservers sind detaillierte Kentnisse der diversen möglichen Schutzmechanismen notwendig, da ohne ihre Benutzung und ohne die Verwendung zusätzlicher Security-Software kaum ein wirksamer Schutz der vom Webserver verwalteten Ressourcen möglich sein wird.

Wenn hier die Frage nach der Beurteilung der Wirksamkeit einer Policy gestellt wird, dann hilft bei gewachsenen Systemen manchmal nur die Neu-Implementierung. In real existierenden IT-Systemen wird bei nicht geplant angelegten Security-Mechanismen eher früher

als später einen Punkt erreicht, an dem weder die Wirksamkeit der Security-Implementierung sinnvoll überprüft, noch irgend etwas an der Policy geändert werden kann, ohne mit hoher Wahrscheinlichkeit Löcher in den Schutzmechanismus zu reißen.

1.2 Überblick

In diesem Artikel wird in Abschnitt 2 untersucht, welche Fragenkomplexe beim Erstellen eines ganzheitlichen IT-Sicherheitsansatzes für einen umfassenden Unternehmensschutz mit speziellem Fokus auf das Access Control (AC) Management zu beachten sind. Dabei wird sichtbar, dass Firewalls und VPNs als Security-Werkzeuge nur einen kleinen Teil der zu lösenden Probleme abdecken und ein AC-Management unverzichtbar ist. Eine Vorstellung der heute gängigen Access-Control-Modelle folgt in Abschnitt 3. In Abschnitt 4 wird mit NGS BeanGuard eine Realisierung eines modifizierten Role-Based Access Control Modelles vorgestellt. Abschließend erfolgt in Abschnitt 5 ein Ausblick, was in kürzerer oder fernerer Zukunft auf diesem Gebiet zu erwarten sein dürfte.

2 Access Control Management und 4A-Systeme

Ein verbreiteter Ansatz für eine Netzwerk-Schutz-Strategie ist der des *Layered Approach* [1]. In diesem Ansatz werden drei Schutzschichten unterschieden:

1. Den *Gateway Layer*. Im Gateway Layer wird die Frage „Darf ich hereinkommen?“ beantwortet. Hier wird festgelegt, *wie* der Zugang zum Netzwerk erfolgen kann. Beispiele für Anwendungen, die in dieser Schicht leben, sind Firewalls, Intrusion-Detection-Systeme, VPNs und Authentifizierungssysteme.
2. Den *Control Layer*. Hier wird die Frage „Wohin kann ich gehen?“ beantwortet. In dieser Schicht sollten Applikationen und Dienste leben, die die Administration von Zugriffserlaubnissen ermöglichen. Diese Administration sollte überall auf die gleiche Weise möglich sein. Sie sollte ein unmittelbares Ein- und Ausschalten von Rechten ermöglichen. Sie sollte so leicht, fehlerfrei und effizient benutzbar sein, dass sie ein „papiernes Antragswesen“ ersetzen kann. Dazu werden in mehr oder weniger großem Umfang Workflow-Management-Systeme als Ergänzung benötigt. Es sollte ein leistungsfähiges Auditing-System anschließbar sein. Eine Erweiterung auf neue Anwendungen sollte ohne größere Probleme ermöglicht werden. Der Control Layer sollte darüber hinaus Mechanismen für ein Single Sign-On bereitstellen.
3. Den *Data Layer*. In diesem Layer wird die Frage „Was kann ich tun?“ beantwortet. Hier werden detaillierte Daten darüber gehalten, welche Benutzer oder Angehörige welcher Rolle oder Gruppe welche Daten lesen, ändern oder löschen können, welche Anwendungen sie ausführen dürfen, auf welche Bibliotheksfunktionen sie Zugriff haben etc. In dieser Schicht sollte sich die Datenhaltung befinden, auf deren Grundlage die Dienste des Control Layers ihre Entscheidungen treffen.

An dieser Stelle konzentrieren wir uns auf eine Auswahl von Aufgaben, die im Control und Data Layer gelöst werden müssen. Andere Aufgaben werden nur insoweit erwähnt, wie sie zur Erfüllung von Aufgaben in diesen Layern erforderliche Daten liefern. Beispielsweise ist die Grundlage für die Autorisierungsentscheidung, ob ein bestimmter Benutzer ein

bestimmtes Recht hat, eine vorausgehende Authentifizierung dieses Benutzers, die aber im Gateway Layer stattfinden sollte.

In dem Marktsegment, das sich mit den Aufgaben des Control Layers und des Data Layers beschäftigt, wird auch von 4A oder AAAA-Systemen gesprochen, die Lösungen von Authentifikations-, Autorisierungs-, Auditing- und (Security-) Administrationsproblemen anbieten. Diese vier Funktionalitätsgruppen sollen in den folgenden Abschnitten 2.1 bis 2.4 diskutiert werden.

2.1 A1: Authentifikation

Authentifikationsdienste dienen dazu, die Identität eines Benutzers zu überprüfen und zu garantieren. Ein Authentifikationssystem sollte zu den folgenden Themen Lösungen zur Verfügung stellen:

- *Feststellung der Authentifikations-Qualität*. Es gibt Authentifikationsverfahren von verschiedener Qualität. Man unterscheidet Verfahren, die Dinge überprüfen, die man *weiss*, wie z.B. Passwörter; Dinge, die man *besitzt*, wie etwa Smart Cards und Eigenschaften, die man selbst *verkörpert*, wie Fingerabdruck etc. Diese drei Qualitätsabstufungen werden auch oft als *was ich weiss, was ich habe, was ich bin* bezeichnet.
  Die einzelnen Verfahren sind unterschiedlich sicher. So ist es leicht möglich, ein Passwort zu kompromittieren. Die Erfahrung zeigt, dass einer der liebsten Aufbewahrungsorte eines Passwortes ein Aufkleber auf der Unterseite der Computertastatur ist. Eine SmartCard muss zum missbräuchlichen Einsatz ihrem Besitzer entwendet werden. Das Nachmachen von Fingerabdrücken erfordert eine noch größere kriminelle Energie. Offenbar ist es also sinnvoll, Zugriffsentscheidungen von der Qualität des verwendeten Authentifikationsschemas abhängig zu machen.
  Außerdem erscheint es sinnvoll, Zugänge aus Intra- und Extranet verschieden zu behandeln, da hierbei in der Regel unterschiedlich sichere Übertragungswege vorausgesetzt werden können. Die Authentifikation liefert hier neben der Feststellung der Identität eines Benutzers auch Daten, die eine Grundlage für nachfolgende Autorisierungen bilden.
- *Sitzungsmanagement*. Ein Authentifikationsdienst sollte die für ein Sitzungsmanagement nötigen Daten liefern. Time-Out-Festlegungen sollten genau so wie Sitzungsverfolgungen für einzelne Sitzungen möglich sein. Es ist sinnvoll, auch „anonyme“ Sitzungen, d.h. Sitzungen von Benutzern, die sich nicht durch Absolvieren eines Authentifikationsschemas angemeldet haben, verfolgen zu können.
  In Web-Shops wollen Kunden wie in einem normalen Geschäft auch zunächst anonym ihren Warenkorb füllen. An der Kasse des Web-Shops wird zumindest in heute existierenden Systemen zur Abwicklung der Bezahlung eine Authentifikation notwendig sein. Nach dieser Authentifikation sollte der Kunde immer noch *seinen* Warenkorb besitzen. Dazu werden Unterscheidungsmöglichkeiten für die einzelnen vorher anonymen Besucher benötigt.
- *Cross-Domain Single Sign-On (SSO)*. Einem Benutzer sollte es ermöglicht werden, mit nur einem Anmeldevorgang auf alle für ihn zugänglichen Ressourcen, selbst auf verschiedenen Systemen und Plattformen, zugreifen zu können. Weitere notwendige

Anmeldevorgänge innerhalb des vom Authentifikationsdienst geschützten Bereiches sollten für den Benutzer nicht sichtbar sein. Dabei ist es unerheblich, ob die benötigten Anmeldeinformationen aus den Zielsystemen in das SSO-System synchronisiert werden oder ob das SSO-Login auch für die Zielsysteme gültig ist.
Die konsequente Verwendung einer SSO-Lösung macht es möglich, dass sich jeder Benutzer nur einmal im System anmelden muss, selbst wenn die dahinter liegenden Anwendungen völlig inkompatible Vorschriften über Struktur und Verwendung der zu ihnen gehörigen Passworte haben. Er muss sich damit auch nur *ein* Password und nur *eine* Password-Policy merken. Neben dem damit verbundenen höheren Komfort bringt das auch eine geringere Wahrscheinlichkeit, dass die Benutzer ihre Passworte vergessen und aus diesem Grund den Help-Desk in Anspruch nehmen müssen. Damit führt ein Mehr an Sicherheit in diesem Punkt auch zu einer direkten Einsparung von Kosten.

– *Integrierbarkeit.* Es ist erforderlich, dass eine neue Lösung in die vorhandene Infrastruktur integriert werden kann. Insellösungen sind zu vermeiden, es sollte immer ein ganzheitlicher Ansatz angestrebt werden. Inkompatible Lösungen für einzelne Problembereiche bergen die Gefahr, dass sie nicht alle Bedrohungen abdecken oder dass sie mehrfach zu pflegende Datenhaltungen benötigen. Vorhandene Nutzerdatenhaltungen sollten direkt als Datenspeicher benutzt werden können, ohne dass sie wesentlich modifiziert werden müssen.
In LDAP-konformen Verzeichnissen sollten vorhandene Schemata unmittelbar verwendet werden können. Sind eigene Datenhaltungen für die Sicherheitslösung notwendig, besteht die Gefahr, dass diese Datenhaltungen nicht aktuell, nicht vollständig oder widersprüchlich sind und damit zu Sicherheitsrisiken führen. Veränderte Rechte, etwa von ausscheidenden Mitarbeitern, müssen sofort wirksam werden können und nicht erst nach einer zukünftigen Synchronisation der Datenhaltungen. Zur Integrationsfähigkeit zählt auch, dass Authentifikationsdienste Programmierschnittstellen nach außen anbieten müssen, z.B. entsprechend dem JAAS-Standard [5], um damit die Einbindung in Applikationen aller Art zu erleichtern.

2.2 A2: Autorisierung

Ein Autorisierungsdienst dient in erster Linie dazu, die Frage zu entscheiden, ob ein identifizierter Benutzer eine bestimmte Aktion auf einer bestimmten Ressource ausführen darf. Unter Ressourcen verstehen wir hier alle Objekte, auf denen ein Benutzer durch bestimmte Anwendungen Aktionen ausführen darf oder wo ihm diese Aktionen verwehrt sind, wie z.B. Files in Filesystemen, URLs, Applikationen, Geräte etc.

– *Deklarativer und programmatischer Zugriffsschutz.* Es kann zwischen deklarativem Zugriffsschutz durch Mechanismen aus der Laufzeitumgebung einer Anwendung und programmatischem Zugriffsschutz, d.h. Aufrufen von Access-Control-Funktionalität aus dem Programmcode heraus, unterschieden werden. Beide Verfahren haben ihre Daseinsberechtigung, allgemein ist aus Gründen der Wartbarkeit eine Trennung von Funktionslogik und Sicherheitslogik empfehlenswert, um bei Releasewechseln die bisher vorhandene Absicherung ohne Wirksamkeitsverlust und ohne administrativen Aufwand weiter verwenden zu können.

– *Lokale und globale Rechtesysteme.* Deklarativer Zugriffsschutz kann z.B. mittels Access-Control-Listen (ACL) realisiert werden. Derartige ACLs werden in manchen Betriebssystemen verwendet. Ihre Auswirkungen sind meist nur lokal und in ihrem Wirksamkeitsbereich meist gut überschaubar. Es ist jedoch sehr schwer, aus ACLs heraus Aussagen über die globalen Eigenschaften des gesamten Schutzmechanismus zu treffen.

  Komplexere Access Control (AC) Strukturen, die auf der Basis von Abhängigkeitsgraphen oder Rollenkonzepten arbeiten, ermöglichen eine Modellierung von umfassenderen Sachverhalten, eine größere Kompaktheit des Modells und eine leichtere Wartbarkeit. Die erstmalige Modellierung komplexerer Strukturen stellt an die vorab erfolgte Beschreibung der Geschäftsvorgänge eines Unternehmens durch Rollen aber hohe Ansprüche. Erfahrungsgemäß ist die Erarbeitung der Rollenmodellierung der Hauptkostenfaktor von Access-Management-Projekten.
– *Bedingte Rechte (Constraints).* Unter Umständen werden Rechte nur bedingt vergeben. So ist etwa denkbar, dass ein Benutzer nur zu üblichen Geschäftszeiten bestimmte Ressourcen benutzen darf. Eine andere Bedingung könnte sein, dass etwa von einem Konto nur abgehoben werden darf, wenn ein Guthaben vorhanden ist.

  Weitere oder andere kundenspezifische Regeln, auch solche, die anwendungsspezifische Daten berücksichtigen müssen, könnten definiert sein (externe Constraints). Derartige erst zur Laufzeit überprüfbare Bedingungen können mit einem Constraint-Handler geprüft werden, dessen Evaluierungsergebnisse in die Zugriffsentscheidung des AC-Systems einfließen.
– *Schutz heterogener Systeme.* Es ist notwendig, auch stark heterogene Landschaften schützen zu können, Webserver verschiedener Hersteller auf verschiedenen Plattformen, verschiedenste Application Server und diverse existierende Applikationen, aber auch Eigenentwicklungen. Für viele dieser Plattformen und Applikationen müssen Mechanismen zur Durchsetzung der Zugriffsentscheidungen beim Anwender selbst entwickelt werden. Dazu werden Programmierschnittstellen (APIs) zum Autorisierungsmechanismus benötigt.
– *Rechteprofile.* Häufig ist die Fragestellung von Interesse, welche Rechte ein konkreter Benutzer hat. Um sie zu beantworten, müssen Rechteprofile ermittelt werden können. Diese Profile kann man zur Untersuchung der Wirksamkeit einer Security-Policy einsetzen. Sie können aber außerdem zur Personalisierung von Anwendungen herangezogen werden.

2.3 A3: Auditing

Auditing spielt eine weitere wichtige Rolle im Kanon der 4A-Dienste.

– *Sitzungsverfolgung.* Auditing ermöglicht die ereignisbezogene oder sitzungsbasierte Verfolgung von Aktivitäten der Benutzer. Es sollte möglich sein, alle Authentifizierungs-, Autorisierungs- und Administrations-Ereignisse, außerdem auch bestimmte technische Ereignisse wie den Start und die Terminierung von Servern, zu protokollieren. Dabei sollten auch verschiedene anonyme Sitzungen voneinander unterschieden werden können. Die Protokollierung ermöglicht es, Handlungen der

Benutzer im System nachzuvollziehen und ebenso Ausfälle von Komponenten festzustellen.

- *Mustererkennung und Profile*. Aus den aufgezeichneten Auditing-Events können Daten für viele Anwendungen gewonnen werden. Neben Intrusion-Detection-Tools, die Angriffsversuche erkennen und darauf reagieren können, sind das auch Analysewerkzeuge für die Gewinnung von Verhaltensprofilen der Benutzer.
- *Reportingwerkzeuge*. Auditing-Dienste müssen mit existierenden Werkzeugen für Auswertung und Reporting verknüpfbar sein. Dazu müssen sie sich an existierende Standards für derartige Tools anpassen können. Weiterhin ist eine API für das Ansprechen von Auditing-Funktionalitäten aus anderen Anwendungen sinnvoll, um auf eine Vereinheitlichung des Auditing-Mechanismus in einem Gesamt-IT-System hinwirken zu können.

2.4 A4: Administration

Das vierte und häufig unterschätzte A-Thema ist die Administration des AC-Modells. Nur ein akkurat gewartetes und aktuelles AC-Management-System kann den wirklichen Schutz der überwachten Ressourcen eines Unternehmens sicherstellen.

- *Administrationswerkzeuge*. Es ist wichtig, dass ein Administrator die Werkzeuge, die er zur Wartung des AC-Systems benötigt, zumindest für alltägliche Aufgaben leicht und vor allem einheitlich bedienen kann. Es wird nicht immer möglich sein, alle Funktionalitäten, die ein AC-System zur Verfügung stellt, beispielsweise in einem einfachen graphischen Benutzerinterface (GUI) anzubieten. Meist wird es zwei oder mehr Werkzeuge für alltägliche und für komplexe Administrationsaufgaben geben müssen. Es ist immer abzuwägen, ob für jede Applikation oder Plattform vollständig passende Werkzeuge verwendet werden sollen, die aber jeweils anders bedient werden müssen, oder ein einheitliches Werkzeug, welches nicht für alle Aufgaben optimal angepaßt ist.
- *Massendatenverarbeitung*. Für die Verarbeitung von Massendaten oder für einen automatisierten Im- und Export von implementierten Policies ist eine Kommandozeilen-Schnittstelle (CLI) erforderlich, die auch eine Steuerung durch Skripte erlaubt. Schnittstellen zum Im- und Export von Policy-Files sind ebenfalls hilfreich.
- *Programmierschnittstellen*. Auch Administrationsfunktionalitäten müssen durch APIs in vorhandene Applikationen einbettbar sein, damit die Administration auch über bereits eingeführte Werkzeuge erfolgen kann oder Funktionalitäten wie etwa die Selbst-Registrierung von neuen Benutzern realisiert werden können. Auf diese Weise können auch benutzerspezifische graphische Oberflächen und Ablaufsteuerungen durch den Anwender realisiert werden.
- *Delegation*. Die Administrationsrechte sollten granular verteilt werden können. Zumindest sind unterschiedliche Rechte zur Administration von Ressourcen, von Benutzern und von Rechten notwendig. Sehr sinnvoll ist die Möglichkeit zur Delegation von Administrationsrechten. So sollte z.B. ein Abteilungsleiter die Mitglieder seiner Abteilung administrieren dürfen und diesen Mitarbeitern Rechte, die er selbst besitzt, weitergeben können. Bindet ein IT-System externe Partner mit ein, so sollten diese Partner ihre jeweiligen Rechte lokal selbst administrieren können.

– *Kosteneinsparung*. Neben dem Ziel der exakten Umsetzung der gegebenen Security-Policy ist es notwendig, die Kosten für die Administration dieser Policy möglichst gering zu halten.

3 AC-Modelle in Theorie und Praxis

Nachdem im vorigen Abschnitt 2 die Anforderungen an AAAA-Dienste diskutiert worden sind, sollen hier mit den Access Control Lists (3.1) und dem Role-Based Access Control (3.2) zwei häufig verwendete Modelle vorgestellt werden. Das Butterfly Modell (3.3), das von der Firma des Autors entwickelt und produktwirksam gemacht wurde, erweitert den rollenbasierten Ansatz.

3.1 Access-Control-Listen

In der Vergangenheit wurde eine Vielzahl von AC-Modellen entwickelt und implementiert. Das einfachste Modell stellen die Access-Control-Listen (ACL) dar. Bei diesen Listen wird davon ausgegangen, dass einzelnen elementaren Ressourcen zu den auf diesen Ressourcen möglichen Aktionen Benutzer oder bestenfalls auch noch Benutzergruppen zugeordnet werden können. Manchmal sind innerhalb der Gruppen Hierarchien erlaubt, meist ist jedoch eine Schachtelung von Gruppen nicht oder nur stark eingeschränkt gestattet. ACLs werden in der Regel in logischer Nähe zu einer Ressource vom System aufbewahrt. Eine Vererbung von ACLs geschieht in der Regel durch Kopieren zu den „erbenden“ Ressourcen. Damit sind ACLs lokal gültig. Diese Lokalität bedingt, dass die Wirksamkeit von ACLs leicht zu beurteilen ist. Sie bedingt aber auch die eingeschränkte Ausdrucksfähigkeit einer einzelnen ACL. Daraus ergibt sich, dass für die Implementierung einer komplexeren Security-Policy sehr viele ACLs benötigt werden. Außerdem müssen für die Beurteilung der Gültigkeit von Eigenschaften für das gesamte AC-Modell alle ACLs einzeln betrachtet werden.

3.2 Role-Based Access Control

Im Bestreben um die Entwicklung von AC-Mechanismen wurde eine Vielzahl von Modellen entwickelt. Zielstellung war dabei unter anderem, komplexere Rechtevergaben zu ermöglichen und außerdem die einfache Abbildung von in der Realität vorhandenen Hierarchien zu erlauben. Eine hervorragende Übersicht dieser Modelle, verbunden mit einer Analyse ihrer Verwendbarkeit für Web-basierte Applikationen, enthält [3].

Als wohl bedeutendstes Modell ist das Role-Based Access Control Model (RBAC) [2] zu nennen. Eine Darstellung der Objektbeziehungen in diesem Modell, das aus diesem Papier stammt, zeigt die Abbildung 1. Im RBAC werden Rollen eingeführt, die ein Benutzer ausfüllen kann. In jeder Sitzung kann ein Benutzer auswählen, ob eine Rolle, zu der er gehört, wirksam sein soll oder nicht. Damit können auch widersprüchliche Rechte eines Benutzers, die sich aus verschiedenen Rollen ableiten, sinnvoll behandelt werden.

Das Rollenkonzept erlaubt die Trennung der Administration von Rechten von der Administration von Benutzern, indem Rechte Rollen und nicht einzelnen Benutzern zugeordnet

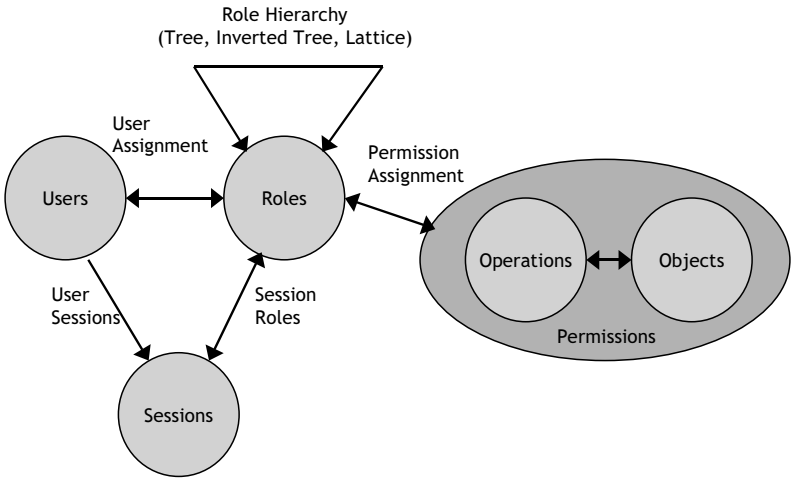

Abbildung 1: Objektbeziehungen im RBAC-Modell

werden. Rollen sind damit Zwischenglieder, die die Beziehungen von Benutzern und Ressourcen vermitteln.

Diese Rollen werden bei einer guten Modellierung sehr viel Ähnlichkeit mit den vorhandenen Funktionsbereichen (Rollen) in den Geschäftsabläufen eines Unternehmens haben. Beim Wechsel des Inhabers einer Rolle ist die Administration dadurch stark vereinfacht. Zwischen Rollen können Abhängigkeiten bestehen, die den Aufbau vieler Varianten von Rollenhierarchien erlauben. Beispiele für solche Hierarchien sind Bäume, invertierte Bäume und Verbände.

Die Modellierung von Rechten mittels RBAC vermindert in der Regel die Komplexität des Modells gegenüber lokal wirksamen Rechtemodellen, wie z.B. ACLs, wie das Beispiel in Abbildung 2 veranschaulicht. Bei geeigneter Modellierung ist in den meisten Anwendungsfällen davon auszugehen, dass sich die Anzahl der zu verwaltenden Zuordnungen von multiplikativer Komplexität (Anzahl der Ressourcen mal Anzahl der Benutzer) auf additive Komplexität (Anzahl der Ressourcen plus Anzahl der Benutzer) verringert.

Die Vorzüge von RBAC gegenüber lokal wirksamen Techniken können wie folgt zusammengefasst werden, einen Vergleich beider Techniken zieht Abbildung 3:

- RBAC ermöglicht die Schaffung von Hierarchien, ACLs bieten nur punktuelle, lokale Wirksamkeit.
- RBAC ermöglicht Vererbung der Wirksamkeit von Rechten, ACLs werden physisch kopiert und müssen später einzeln administriert werden.

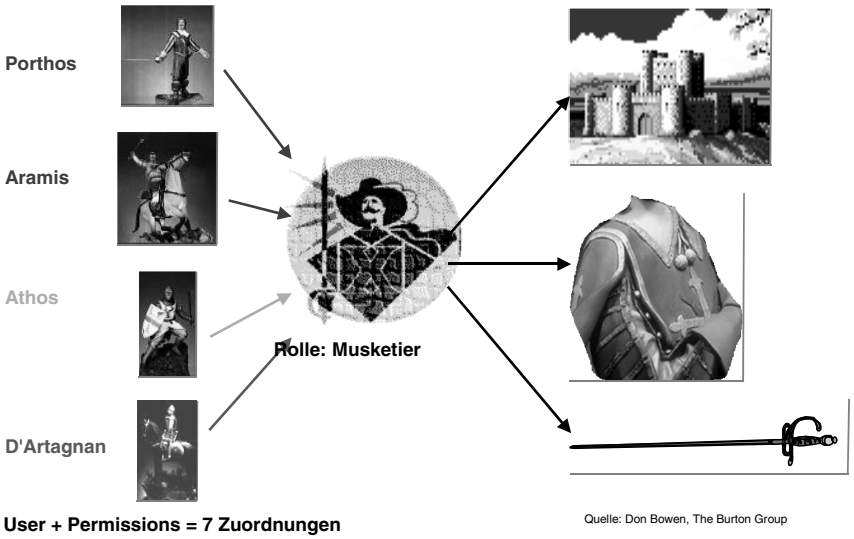

Abbildung 2: Der Vorteil von Rollen in der Rechtemodellierung am Beispiel der vier Musketiere: Jeder Musketier hat Zugriff auf Degen, Uniform und Kaserne. Das sind $4 * 3 = 12$ Zuordnungen. Führt man die Rolle „Musketier“ ein, sind nur noch $4 + 3 = 7$ Zuordnungen zu verwalten.

- RBAC ermöglicht die Antwort auf Fragen, die Kenntnis vom Gesamtzusammenhang erfordern, wie „Auf welche Ressourcen darf ein Benutzer A zugreifen?“. Damit ist im RBAC eine ganzheitliche Modellierung möglich.
- Die Wirksamkeit einer in einem RBAC-Modell implementierten Policy ist leichter zu beurteilen.
- RBAC-Modelle sind leichter und schneller zu verändern.

3.3 Das Butterfly-Modell

Das Butterfly-Modell [6] ist eine Ableitung aus dem RBAC, in der zusätzlich auch Gruppierungen von Ressourcen und Hierarchien von Ressourcengruppen erlaubt sind, wie Abbildung 4 illustriert.

Damit können neben den in der Realität vorhandenen Hierarchien von Benutzer-Rollen auch Hierarchien von Ressourcen im AC-Modell berücksichtigt werden. Derartige Hierarchien kommen in der Realität häufig vor, man denke nur an Filesysteme oder Komponenten von Applikationen. Neben der erleichterten Modellierung der Ressourcen ist eine weitere Aufspaltung administrativer Aufgaben möglich, da die Administration von Benutzern, Ressourcen, Rollen und Rechten der Rollen voneinander getrennt werden kann, wie die Abbildung 5 verdeutlicht.

	Role-Based Access Control	**Access Control Lists**
Objekte	User/Group Role Session Object Operation Permission	User/Group -/- -/- Datei/Verzeichnis Berechtigung -/-
Struktur	hierarchisch graph-basiert	flach punktuell
Ausstrahlung	Vererbung	Duplikation
Sichtbarkeit	ganzer Graph	einzelne Datei einzelnes Verzeichnis

Abbildung 3: Vergleich der wichtigsten Eigenschaften von ACLs und RBAC-Modellen.

Abbildung 6 veranschaulicht die Gemeinsamkeiten des Butterfly-Modells mit dem RBAC-Standard. Die wichtigste Erweiterung stellt die Möglichkeit der Modellierung von Hierarchien auf Benutzer- und Ressourcenseite dar.

4 NGS BeanGuard: Ein Beispiel

Eine Beispiel-Implementierung des Butterfly-Modells als RBAC-Variante ist NGS BeanGuard, ein Produkt der NorCom Information Technology AG (www.norcom.de), München. Dieses Produkt realisiert einen auf dem Butterfly-Modell aufbauenden Autorisierungsdienst. Außerdem werden alle in Abschnitt 2 angesprochenen 4A-Dienste angeboten oder es sind Schnittstellen für die Anbindung dieser Dienste vorhanden. Damit eignet sich BeanGuard als Herzstück eines IT-Security-Systems im Control Layer, speziell im Web-Bereich.

NGS BeanGuard ermöglicht es, ausgehend von einem *Single Point of Security Administration (SPSA)* unternehmensweit die Zugangsinformationen sämtlicher Benutzer und alle Berechtigungen dieser Benutzer mit einer einheitlichen Methode und einem einheitlichen Werkzeug zu verwalten. Diese Verwaltung kann von beliebig verteilten Plätzen und mit beliebig granularen Administrationsrechten erfolgen.

Damit werden Lösungen für die Themenstellungen Single Sign-On, Rechte-Delegation, Rollenhierarchien, standardisierte Benutzer- und Ressourcenverwaltung und standardisierte Rechte-Administration, selbst eingebunden in Workflow-Systeme, möglich.

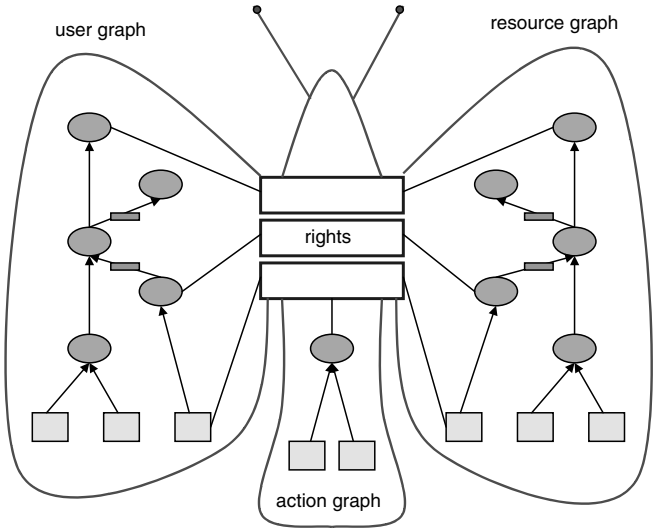

Abbildung 4: Im Butterfly-Modell sind Hierarchien sowohl in den Benutzerstrukturen als auch in den Ressourcenstrukturen erlaubt.

Es werden über geltende Standards hinaus weitere Zugriffskontrollmöglichkeiten bereitgestellt. Ein Beispiel dafür ist der Schutz von Instanzen und Instanzenmethoden von Enterprise JavaBeans. Ein weiteres Beispiel ist der Schutz von parametrisierten URLs und damit von in Servlets realisierten endlichen Automaten auf Zustandsebene.

Durch das Butterfly-Modell wird eine RBAC-Modellierung der Benutzerrechte möglich. Damit ist die Administration von Rechten wesentlich intuitiver, schneller, einfacher und damit auch fehlerfreier durchzuführen. Die für und durch BeanGuard zusammengetragenen Daten bilden eine gute Grundlage für die Gewinnung von Rechteprofilen und für die Analyse des Benutzerverhaltens und unterstützen damit eine weitgehende Personalisierung von Web-Angeboten. NGS BeanGuard liefert Programmierschnittstellen für C++, Java und Perl und ist durch die Java-Implementierung der Kern-Komponenten auf nahezu allen Plattformen lauffähig.

5 Ausblick

Läßt sich mit heute existierenden Access-Control-Lösungen tatsächlich eine Überprüfung der durch sie implementierten Security-Policy durchführen? Bei Access-Control-Listen, die nur lokale Wirksamkeit besitzen, vermutlich nicht. Für graphenbasierte Ansätze, wie sie z.B. diverse RBAC-Varianten darstellen, kann das ein geübter Administrator sicher tun, wenn die Policy nicht allzu komplex ist. Hat der Administrator aber wirklich die rich-

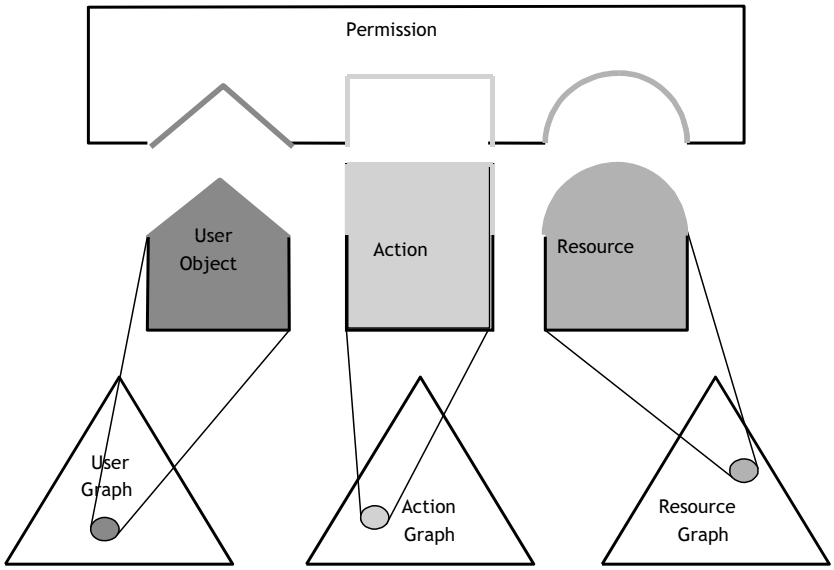

Abbildung 5: Trennung der Administration von Benutzern und Ressourcen im Butterfly-Modell.

tige Policy umgesetzt? Hier wäre es wünschenswert, wenn komplexere Anfragen an ein Access-Control-System gestellt werden könnten, wie beispielsweise „Dürfen alle Marketingverantwortlichen der Fachabteilungen alle Preislisten für die für ihre Abteilungen relevanten Produkte einsehen und die ihnen selbst zugeordneten Preislisten ändern?“.

Derartige Fragen sind mit den Mitteln eines geeigneten Logikkalküls formulierbar. Damit könnten sie mit geeigneten KI-Methoden[1] wie z.B. automatischen Deduktionssystemen [4] untersucht und beantwortet werden. Sogar Beweise dafür, ob ein realisiertes Sicherheitssystem bestimmte Eigenschaften besitzt oder nicht, können auf diese Weise abgeliefert werden. In einer noch weiteren Zukunft reicht es dann bereits aus, die Security-Policy in diesem Kalkül zu formulieren. Die Umsetzung in eine für die Maschine geeignete Darstellung kann dann automatisch erfolgen.

Diese Zukunftsaussichten bezüglich der Einfachheit eines umfassenden Ressourcenschutzes stimmen optimistisch. Aber wie sieht die Gegenwart aus? Sicherheit sieht man nicht. Sie wirft keinen unmittelbaren Gewinn ab. Viele Manager erkennen noch nicht das möglicherweise existenzielle Risiko, das von möglichen Angriffen auf ihr Unternehmen ausgehen kann. Oft findet man noch viele der neun verbreitetsten Irrtümer über Sicherheitsbedrohungen:

1. „Sicherheit ist ein wichtiges Thema, aber nicht für uns, sondern nur für die anderen.“

[1] KI = künstliche Intelligenz

	Role-Based Access Control	**Butterfly Model**
Objekte	User/Role Object Operation Permission	User/User Group/Role Resource/Resource Group Action (Permission)
Zuordnungsobjekte	Role Session	Permission Session
Vergleich	ïHierarchisierung von Rechten ïEinf¸hrung von Sessions ïFlache Ressourcen	ïStrukturierung von Benutzern *und* Ressourcen ïTrennung von Strukturierung und Rechtezuordnung ïButterfly ist RBAC-Variante, die Ressourcenmodellierung erweitert

Abbildung 6: Vergleich der wichtigsten Merkmale von Butterfly und RBAC.

2. „Das Problem wird sich schon lösen, wenn wir es nur fest genug ignorieren."
3. „Wir haben jetzt Technik gekauft. Damit ist das Problem gelöst. Auch wenn wir die Technik noch nicht beherrschen."
4. „Unsere Daten und unser Image sind nichts wert. Darum brauchen wir sie auch nicht zu schützen."
5. „Sicherheit hat mit unserem Geschäft nichts zu tun."
6. „Wir haben doch eine Firewall."
7. „Sicherheitsaspekte können wir immer noch später hinzufügen, jetzt realisieren wir erst einmal die Funktionalität."
8. „Sicherheit ist ein lästiger Kostenfaktor, keine Investition. Wir müssen nur Geld ausgeben, verdienen damit nichts und sparen auch nichts ein."
9. „Wir brauchen keine unternehmensweite Sicherheitsarchitektur."

Sicherheit ist kein Kostenfaktor, Sicherheit ist eine Investition in die Zukunft des Unternehmens. Sicherheit muss ganzheitlich und unternehmensweit organisiert werden. Access-Control-Management gehört untrennbar dazu."

6 NorCom Information Technology AG

Die NorCom Information Technology AG mit Hauptsitz in München und weiteren Standorten in Frankfurt, Dublin, Oslo, Mailand und Silicon Valley, USA, gehört mit rund 280

Mitarbeitern zu den führenden Anbietern von zukunftsweisenden Sicherheitstechnologien. Die Kernkompetenz des 1989 gegründeten Software- und Beratungsunternehmens liegt in Produktlösungen für die Technologie-Bereiche E-Security und E-Business Solutions. Als Full-chain supplier liefert NorCom in beiden Geschäftsfeldern Produkte, Consulting und Projektlösungen für die Absicherung von IT-Infrastrukturen, E-Business-Anwendungen, Portalen und Marktplätzen.

Zu den erfolgreichsten Technologien des Sicherheits-Spezialisten gehört die Produktlinie NGS. Mit NGS Enterprise Framework und NGS BeanGuard bietet NorCom Sicherheit für große heterogene Client-/Server Infrastrukturen, für Application Server Umgebungen und moderne Netcentric Web Architekturen. NorCom ist seit 1. Oktober 1999 am Neuen Markt in Frankfurt notiert. Das dynamische Wachstum und die internationale Expansionsstrategie in europäischen und außereuropäischen Märkten wird gestützt durch strategische Allianzen mit internationalen Blue-chip Partnern wie BEA und Vignette. Über Akquisitionen und Beteiligungen an Unternehmen, wie NSA und V&R, werden Wachstumssynergien frei, die den Eintritt in neue Märkte und Branchen erleichtern und die Marktpenetration beschleunigen. Wichtigste Zielbranchen der NorCom AG sind Banken, Finanzdienstleister, Telekommunikation, Industrie und Behörden. Zu den Kunden des Sicherheitsspezialisten zählen renommierte Unternehmen wie die Dresdner Bank, die ADIG Investment, die Commerzbank, die HypoVereinsbank, die Allianz, die S-Finanzgruppe Bayern oder Ticket-Online.

Literatur

[1] David M. Hager. Building An Information Security Architecture for the Future. In: Best Practices for Preventing Cyber Sabotage, Netegrity Web-Seminar, 2000.

[2] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, Ramaswamy Chandramouli: A Proposed Standard for Role-Based Access Control, 2000.

[3] James B. D. Joshi, Walid G. Aref, Arif Ghafoor, Eugene H. Spafford. Security Models for Web-Based Applications. CACM 44 (2), 2001.

[4] Bernd I. Dahn, Jürgen Gehne, Thomas Honigmann, Andreas Wolf. Integration of Automated and Interactive Theorem Proving in ILF. CADE-14, LNAI1247, Springer, 1997.

[5] Java Authentication and Authorization Service (JAAS) for the Java 2 SDK, Standard Edition, v 1.4. http://java.sun.com/products/jaas/index-14.html

[6] Andreas Wolf, Christian von Hammel, Uwe Würfel. Internet Resource Access Control - The Butterfly Model. SCI2001, IIIS, 2001.