The ASCAA Principles for Next-Generation Role-Based Access Control

Ravi Sandhu
Executive Director and Chaired Professor
Institute for Cyber Security
University of Texas at San Antonio
ravi.sandhu@utsa.edu

Venkata Bhamidipati
Doctoral Candidate
School of IT and Engineering
George Mason University
venkata.bhamidipati@oracle.com

***Abstract*—Role-based access control (RBAC) received serious academic attention in the early 1990's, although traces of the underlying concepts had been in ad hoc commercial practise since the 1970's. Through the 1990's and 2000's RBAC achieved remarkable success, and today is widely practised as the preferred form of access control. Adoption of the 2004 NIST/ANSI Standard RBAC Model [1] marks a maturity of concept and practice. The essential roots of this standard go back to the RBAC96 model [2]. While numerous enhancements and extensions of RBAC96 and related models have been proposed, the core ideas introduced in RBAC96 have proved to be notably stable and robust.**

RBAC96 was based on four principles, viz. abstract privileges, separation of administrative functions, least privilege and separation of duties. Advances in RBAC require reconsideration of its founding principles. In this paper we offer five founding principles for next-generation access control including next-generation RBAC, summarized as ASCAA for Abstraction, Separation, Containment, Automation and Accountability. Abstraction (i.e., abstract privileges) and separation (i.e., separation of administrative functions) are essentially retained from RBAC96. A generalized principle called containment is introduced, to subsume least privilege, separation of duties and other constraints, as well as modern techniques such as usage and rate limits [3], [4]. Next two new principles called automation and accountability are introduced.

Automation covers automated acquisition of privileges as well as automated revocation. Traditional RBAC typically requires that user-role and permission-role assignment and revocation result from explicit actions of appropriately authorized administrators. Some aspects of automated user-role assignment [5], [6], [7] and user-role revocation [8] have been previously proposed. We elevate the notion of automation to a full-blown principle and specifically propose self-assignment of roles as a new element. Automating assignment and revocation enables agile lightweight systems by eliminating repeated human intervention. Crucially, of course, we want to do this without compromising security.

Accountability has recently received considerable attention driven by emerging requirements of secure information sharing and continued recognition of the insider threat. We offer the paradigm of adjustment as a means to achieve accountability. Adjustment acknowledges that not all authorized actions are the same. Sensitive operations require an enhanced level of auditing, notification or authentication. For example, it is common place for websites to require additional authentication and notification for sensitive operations such as change of address.

While we believe these five ASCAA principles (Abstraction, Separation, Containment, Automation and Accountability) are relevant to access control systems in general, the discussion in this paper is limited to their application to RBAC.

I. Introduction

In the past fifteen years or so role-based access control (RBAC) has received strong support from the research and practitioner communities. In this relatively short period it has become the dominant form of access control in commercial products. Nascent ideas, similar in some aspects to modern roles, have been present in the literature since the earliest days of access control (for example, [9], [10]) and have been reiterated over the years (for example, [11], [12]). The traditional dichotomy of discretionary versus mandatory access control (DAC versus MAC) was codified in the highly influential, but ultimately flawed, "Orange Book" [13], [14]. Unfortunately, neither DAC nor MAC was suitable for the needs of most commercial and military applications and there was continued dissatisfaction with this traditional dichotomy (for example, [15], [16]). DAC is too weak for most needs and MAC is simply inappropriate. Alternate approaches which sought to fill this gap, such as Type Enforcement [17], Originator Control [18], and Propagation Models [19], [20], [21], [22], received academic interest but were not successful in influencing real-world commercial practice. It was only with the emergence of RBAC that a practically successful paradigm going beyond MAC and DAC gained real traction.[1]

The modern concept of role-based access control emerged in the early 1990's. The seminal paper of Sandhu et al [2] established the RBAC96 model as the de facto standard for RBAC.[2] Since then RBAC96 has proved to be remarkably robust. A sizable literature on RBAC has developed. A thorough review of notable contributions is beyond the scope and purpose of this paper, so we will only mention a few influential highlights here. Administrative models for RBAC include [26], [27], [28], [29]. Temporal considerations in RBAC were introduced in [30], [31], [32]. Separation of duty constraints in RBAC

[1]RBAC truly goes beyond MAC and DAC because while it can be configured to do either one [23] it really seeks to capture requirements that are not even considered in MAC or DAC. Conversely, MAC is based on the single overriding principle of enforcing one-directional information flow in a lattice of security labels, and DAC on the similarly overriding but different principle of owner-based discretion, neither of which are recognized as foundational principles for RBAC. The fact that RBAC96, based on a completely different set of foundational principles, can support MAC and DAC is serendipitous and not by design.

[2]Other early influential RBAC papers include [24], [25].

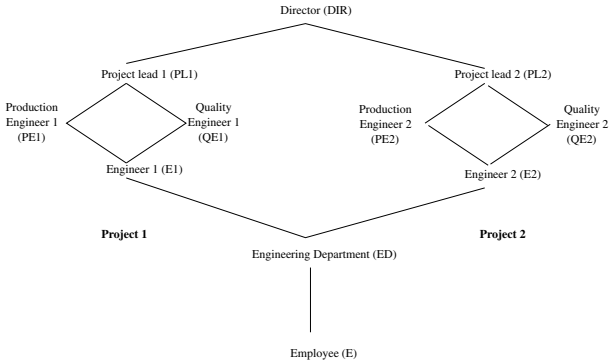

Fig. 1. An Example Role Hierarchy [26]

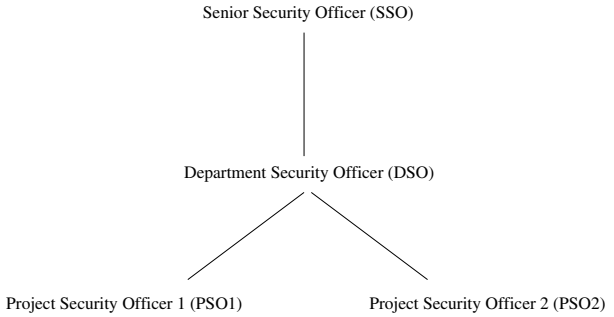

Fig. 2. An Example Administrative Role Hierarchy [26]

were studied in [33], [34], [35] and further explored in [36] along with other constraints. The interaction of RBAC and workflow is discussed in [37], [38], [39]. Delegation models for RBAC were first proposed in [8] and further developed in [40]. The interplay of RBAC and Trust Management has been investigated in [5], [6]. All of these extensions and enhancements have been accomplished without changing the core concepts of RBAC96 which have remained stable. Roles hierarchies and constraints have proven to be two central ideas of RBAC96 that can robustly support many extensions.

The only really significant concept that was not present in the original RBAC96 model is role activation hierarchies [41]. Consider the often used example role hierarchy in figure 1. RBAC96 specifies role hierarchies as inheritance hierarchies. Thus roles higher up in the hierarchy such as DIR inherit permissions from all junior roles. Equivalently junior roles inherit members from senior roles. Inheritance is convenient but can result in aggregating lots of permissions in very senior roles such as DIR. The concept of activation hierarchy authorizes a member of DIR to activate roles junior roles such as PL1 and PL2 instead of simply inheriting their permissions when DIR is activated. This allows the DIR role to exercise oversight without unnecessary aggregation of permissions. The NIST/ANSI RBAC standard [42] allows hierarchies to be inheritance hierarchies or activation hierarchies or some combination, leaving it up to the vendor to specify. This is the sole substantive feature of the RBAC standard missing from the original RBAC96 model.[3]

This brings to the topic of this paper. It is our belief that substantial advances in RBAC will be made only by reconsidering the foundational principles leading to new RBAC models.[4] To this end we first review the foundational principles of RBAC96 and then propose the new set of five ASCAA principles (Abstraction, Separation, Containment, Automation, and Accountability) for next-generation RBAC.

II. RBAC96 Principles

RBAC96 was motivated by four foundational principles, discussed below. RBAC96 does not actually "enforce" these principles, or require "conformance." It is possible to completely ignore them and still technically do RBAC. This is perhaps inevitable in any truly flexible model. It is always possible to do "bad" things if you really want to. RBAC96 does provide the means to make these principles convenient, and even natural, to pursue. In other words it is easier to do "good" things with RBAC96 but it remains possible to do "bad" things.[5]

A. Abstraction

The abstraction principle refers to abstraction of permissions. In general, permissions and operations are system specific. For example operating systems typically support permissions such as read, write and execute. Database management system permissions could be select, delete, update, etc. While RBAC is useful in controlling permissions at these lower levels, its real value lies in managing abstract application-oriented operations such as credit and debit operations on an account. Capturing and managing abstract operations allows for distinction of usage which are not possible if only system level permissions are managed by RBAC. To the contrary MAC takes a reductionist approach of providing protection entirely in term of primitive read and write operations. This is appropriate for MAC due to its single-minded focus on lattice-based information flow which can be reduced to reads and writes. But in this view credit and debit operations both involve read and write of the account balance, and are thereby indistinguishable.

The constructive approach of RBAC accommodates the semantic distinction between credit and debit, familiar to anyone who has balanced a checkbook. The permissions for credit and debit can then be granted to different roles. The fact that the

[3]With regard to constraints the RBAC standard actually degrades the capability of RBAC96 since it only permits static and dynamic separation of duty constraints.

[4]There have been some recent suggestions with respect to improving the NIST/RBAC standard [43] but these do not really offer significant enhancements. Some of these were already considered in RBAC96 and the NIST/RBAC standard and rejected for good reason [44], [45]. Others correct small technical errors and omissions in the standard and are not fundamentally substantive.

[5]This is not a trivial accomplishment because with some models such as MAC we are actually prevented from doing useful things we would like to do because of its sole focus on information flow.

information flow in both operations is identical is irrelevant to RBAC and does not force us to assign both operations to the same role.[6] Because of its fixation on information flow MAC is unable to express authorization policies that distinguish between credit and debit, which is a commonplace requirement in business applications. Abstract permissions raise the level of policy consideration to application-level semantics, and accommodate real-world policies beyond information flow.[7] Operating systems and database management systems have provided mechanisms for implementing abstract permissions since the early 1970's by means of stored procedures or similar constructs, and these mechanisms have been widely used.

The abstraction principle remains unchanged in our proposed set of principles for next-generation RBAC.

B. Separation

Separation refers specifically to separation of administrative functions. This principle is not explicitly articulated in the original RBAC96 paper [2], [44], but it deserves recognition in retrospect because of its utility. One of the attractive aspects of RBAC is the separation of user-role assignment from permission-role assignment. These two tasks require different skills and are operationally distinct. Permission-role assignment requires deep knowledge of application semantics and security needs. This is best done by people who understand the application and the system that supports it. User-role assignment is a human resources and people management task which requires greater understanding of the human side and an appreciation of overall priorities which may need to be balanced with respect to individual decisions. Moreover, the deeper policy issues in managing the overall set of roles, the role hierarchy and constraints can be further separated into a business security organization. This organization can focus on the more important policy issues around RBAC while devolving day-to-day operational details to appropriate business and information technology people.

The separation principle also remains unchanged in our proposed set of principles for next-generation RBAC.

C. Least Privilege

Least privilege is a long-standing tenet of access control. RBAC supports it by having each role assigned with permissions appropriate to the business function of the role. Taken to its extreme least privilege is likely to result in unmanageable proliferation of roles so there is need for judicious balance. By letting the role designer determine the degree of role fragmentation versus least privilege RBAC provides support for achieving this balance.

The least privilege principle is subsumed by the more general containment principle in our proposed set of principles for next-generation RBAC.

[6]In this respect DAC also allows recognition of abstract permissions such as credit, debit and does not force their assignment to be coupled as in MAC.

[7]The aggregation of multiple permissions into a role is itself a significant abstraction benefit of RBAC. The abstraction principle as such is focussed on what are atomic permissions from an application perspective.

D. Separation of Duty

Separation of duty has been a driving principle for RBAC. The fact that roles may be conflicting has been a long standing practice in commerce since ancient times. Modern business practices continue to build on these long-proven insights. A common example is the conflict between purchasing manager and accounts payable manager roles. A single person with both roles would have sufficient power to single-handedly commit fraud, hence the conflict. This requirement is commonly called static separation of duties. A more nuanced conflict exists between a cash register manager and cash register clerk. A single individual could legitimately take on both roles, but not at the same time on the same register. This is called dynamic separation of duties. A common example of separation of duties from the military sector is the use of two-person, or more generally n-person, rules which require two, or more, people to authorize critical actions such as launch of weapons of mass destruction.

The separation of duty principle is also subsumed by the more general containment principle in our proposed set of principles for next-generation RBAC.

III. ASCAA Principles

We propose the following five principles for next-generation access control, and next-generation RBAC in particular. We refer to these as the ASCAA principles for Abstraction, Separation, Containment, Automation and Accountability.

A. Abstraction

The abstraction principle is essentially unchanged from RBAC96 and refers to abstraction of permissions.

B. Separation

The separation principle is also essentially unchanged from RBAC96 and refers to separation of administrative function.

C. Containment

The containment principle unifies the older principles of least privilege and separation of duty, and further incorporates additional constraints and usage control elements. The concept of containment seeks to limit the damage that a user, or a set of users, can perpetrate either by deliberate malice or by victimization from malicious malware. The individual techniques such as least privilege, separation of duty, cardinality constraints [2], [36] and usage limits [3], [4] are means to this end. They should be viewed as applicable mechanisms rather than motivating principles.

Least privilege and separation of duty have been discussed in the previous section and are familiar in the access control literature. There is not much more for us to say about these here. RBAC96 introduced an abstract open-ended notion of constraints to accommodate separation of duties but not be limited only to this specific mechanism. The construction for MAC in RBAC [23] requires separation of duty constraints but in addition requires cardinality and simultaneity constraints on user-role and permission-role assignment. In other words

there are real-world security policies that appear to require constraints beyond separation of duty for their expression in RBAC. Other uses of constraints are discussed in [36]. Containment accommodates these previously published aspects of current-generation RBAC, beyond least privilege and separation of duty.

Looking to next-generation RBAC we believe it is important to incorporate usage and rate limit concepts from recent models for usage control [3]. Usage limits occur in various forms. We can restrict the number of times that a particular permission or role can be exercised. These limits may be absolute so the usage quota runs out at a certain point and requires replenishing by some means for further access. This case is more appropriate for digital rights management where access is purchased by some exchange of value, such as money. Rate limits control the number of times a particular permission or role can be exercised in a specified period of time. For example, many ATM networks limit a user to, say, three withdrawals per day. This limits loss due to misuse of an ATM card. Rate limits are particularly relevant to RBAC. For example, a customer service representative (CSR) can be limited to access a certain number of customer records commensurate with the anticipated workload. This contains the damage from a malicious CSR who is fishing for customer data. Perhaps more importantly the rate limit also contains the damage by malware. By restricting the rate to be within the rate reasonable for a human to consume the information, the human initiated activity is not disrupted but access at machine speeds by malware is cut off. These applications have been motivated in the usage control literature but are also relevant to RBAC, and should be supported in next-generation RBAC.

D. Automation

We believe that automation of access control administration is inevitable in next-generation access control simply to keep pace with scalability requirements of cyberspace. Assignment and revocation have traditionally been viewed as administrative actions requiring intervention by human administrators.[8] Current-generation RBAC offers substantial benefits in this regard by aggregating permissions into roles which can then be assigned in a single step instead of requiring multiple assignments. Moreover, additions and deletions of permissions to and from a role have immediate effect with respect to role members. To this degree RBAC inherently supports automation by aggregation. We propose to extend automation to a much deeper level in next-generation RBAC.

The failure to remove privileges after they are no longer needed continues to be a major source of security problems. Privileges are too often left intact when users leave or are reassigned to different jobs within an enterprise. These unused and unnecessary residual permissions become an attractive pathway for attackers. The role-based delegation models of [8] incorporate a time limit on temporary delegations so that the delegation will expire by the specified time. We believe such limits should be embedded into other permission granting operations of RBAC. We feel this is particularly so for user-role assignment. In general we expect permission-role assignment to be fairly stable and automated revocation is less likely to be useful in this context. User-role assignment on the other hand is likely to change more rapidly over time. We can base automatic revocation in this context on a fixed time period, such as one year, or on time elapsed since last use or similar considerations. Recovery from revocation in this context can be based on explicit human intervention by an appropriate administrator to restore the revoked role. But automation can be used in this context too. We can allow the user to renew role-membership on their own after the proscribed period has expired. Thus users who need the role can keep it alive whereas those who do not can let it expire.[9]

Another aspect of automated revocation is cascading revocation. A single revocation action can result in multiple revocations as conditions for maintaining the permissions change. In the DAC context cascading revocation is tied to the notion that one can only grant a subset of what one has, and one can only revoke what one has given. Cascading revokes then relate to chains of grant operations and their undoing as revocations at the head of the chain automatically propagate to wipe out the remainder of the chain. As argued in [26] these DAC concepts are not appropriate in RBAC. In RBAC we often see situations where an administrator can grant a role the administrator does not possess. Likewise chains of grants do not immediately become illegitimate because one administrator early in the chain leaves the organization. Thus the model of [26] suggests a notion of strong versus weak revocation. Weak revocation does not require cascading revoke whereas strong does. In other words the automation of cascading revokes is relevant to RBAC but it should be interpreted quite differently from its usual interpretation in DAC.

We also propose to apply automation to role assignment. One of the recognized benefits of RBAC is simplicity of administration. However, as roles proliferate the administration burden grows. Several models for decentralized user-role assignment have been proposed [26], [27], [28]. Figures 1 and 2 from [26] respectively show a role-hierarchy and an administrative role hierarchy. Junior administrators such as PSO1 and PSO2 are limited in the roles that they can assign and in the users to whom they can assign these roles. The URA97 model of [26] provides the notion or prerequisite conditions for users to be eligible for assignment, while [27], [28] argue for using organizational structure rather than roles for this purpose. Nonetheless granting and revocation of roles in these models requires explicit administrator action. Imagine these two figures expanded to have hundreds or thousands of projects, with engineers working on dozens of projects at any time with frequent re-assignment to different projects. Administration entirely by human intervention becomes im-

[8]The demand operation for acquiring privileges in the schematic protection model [20], [46] and automated expiry of role delegation in the role-based delegation models of [8] are some exceptions to this tradition.

[9]Anticipating the accountability principle we could require additional authentication to effect the renewal step so it can be attributed to the user rather than malware that simply keeps all the users roles alive.

practical on such a scale. Instead, we could authorize users to take on role membership by self-assignment in a limited number of projects on their own, while restricting them from simultaneous membership in too many projects and too rapid a rate of change of projects. The ability to do this would be predicated on membership in basic roles such as ED or E. This would allow flexibility in which projects engineers can work on and can explore while limiting the damage due to malicious activity of users or malware. We could insist that higher level roles such as PE1 and QE1 require explicit administrator assignment or approval.

Another application of self-assignment can arise in a professional or social community context. Initial membership in the community could be made available on a purely self-assigned basis with minimal, if any, verification of self-asserted user attributes. Self-promotion of membership up to a certain level could be permitted with the understanding that taking on more senior roles implies obligation to participate in community activities, such as reviewing and mentoring in a professional society. The highest levels of membership could require an approval process by appropriate peers. Thus endorsement by a specified number of advanced members would be required for assignment of a user to advanced member status.

We are currently developing formal models for self-assignment. In general we expect that more junior roles would be available for self-assignment with usage and rate limits, and that more senior roles would require human approvals. We also anticipate the need to allow self-assignment of senior roles on a temporary basis in case the additional privileges are required in extreme situations. Thus a user with PE1 privileges may temporarily escalate to PL1 with appropriate accountability provisions, which brings us to our next principle of accountability.

E. Accountability

Accountability has recently received considerable attention driven by emerging requirements of secure information sharing and continued recognition of the insider threat. We offer the paradigm of adjustment as a means to achieve accountability. Adjustment acknowledges that not all authorized actions are identical. Sensitive operations require an enhanced level of auditing, notification or authentication. For example, it is common place for websites to require additional authentication and notification for sensitive operations such as change of address.

The primary goal of accountability is to make a human user take responsibility for actions that the individual performs in a system. This can be achieved in a combination of three basic ways. Sensitive operations can be subjected to a more detailed level of auditing but unless the audit records are brought to some other user's attention the audit trail is useful only as a forensic tool. Detailed audit trails can trigger fraud detection systems to direct their attention to suspicious activity but ultimately some user has to be alerted. Notification is a more direct approach to explicitly require sensitive operations to trigger a message to an appropriate user. Thus temporary self-assignment of the PL1 role by a PE1 user should trigger a message to all PL1 users alerting them to the circumstance. This will inhibit inappropriate use of this escalation privilege. Finally it is important to escalate the authentication required for sensitive operations. Thus a re-authentication may be required when a particularly large transaction is attempted. The re-authentication may fail if a human or malware attacker is attempting the operation and is unable to produce the necessary credentials on demand. Instead of or in addition to a re-authentication we may require an alternate authentication based on credentials other than used for the earlier authentication.

In the context of RBAC we can measure sensitivity of operations with respect to the application semantics such as monetary amount of a transaction. We can also measure sensitivity with respect to roles and various stages of role assignment, activation, usage and release. We are currently developing formal models for this purpose.

IV. Conclusion

In this paper we have proposed a new set of five principles called ASCAA for next-generation RBAC comprising Abstraction, Separation, Containment, Automation and Accountability. We believe these principles are applicable to access control in general, but our immediate focus has been on RBAC. Two of these, abstraction (i.e., abstraction of permissions) and separation (i.e., separation of administrative functions), are essentially unchanged from the RBAC96 principles. The third, containment, generalizes and unifies the two older principles of least privilege and separation of duties to include additional constraints and usage control elements. The final two, automation and accountability, are newly recognized in this paper. Automation applies to revocation and to the granting of roles and permissions. The accountability principle is illustrated in this paper with respect to authentication adjustment, enhanced auditing and targeted notification.

It is our belief that next-generation access control, and next-generation RBAC in particular, should be based on this expanded set of principles so as to address real-world protection needs of next-generation systems. We are currently developing a new RBAC model based on these principles, and their interactions, and will report our results in future papers.

Acknowledgement

Ravi Sandhu would like to thank his students, colleagues and critics for their interest in RBAC and the numerous advances resulting from their efforts, as well as practitioners, many unknown to him, who have translated theoretical and conceptual RBAC ideas into real-world practise.

References

[1] ANSI INCITS 359-2004, *Standard for Role Based Access Control.*

[2] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, February 1996.

[3] J. Park and R. Sandhu, "The UCON$_{ABC}$ usage control model," *ACM Transactions on Information and System Security*, vol. 5, no. 6, 2007.

[4] A. Pretschner, M. Hilty, and D. Basin, "Distributed usage control," *Communications of the ACM*, vol. 49, no. 9, pp. 39–44, 2006.

[5] A. Herzberg, Y. Mass, J. Mihaeli, D. Naor, and Y. Ravid, "Access control meets public key infrastructure, or: Assigning roles to strangers," *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, pp. 2–14, 2000.

[6] N. Li, J. Mitchell, and W. Winsborough, "Design of a role-based trust-management framework," *Proceedings IEEE Symposium on Security and Privacy*, pp. 114–130, 2002.

[7] M. Al-Kahtani and R. Sandhu, "A model for attribute-based user-role assignment," *Proceedings 18th Annual Computer Security Applications Conference*, pp. 353–362, 2002.

[8] E. Barka and R. Sandhu, "Framework for role-based delegation models," *Proceedings of the 16th Annual Computer Security Applications Conference*, 2000.

[9] J. Wimbrow, "A Large Scale Interactive Administrative System," *IBM Systems Journal*, vol. 10, no. 4, pp. 260–282, 1971.

[10] E. Fernández, R. Summers, and C. Coleman, "An authorization model for a shared data base," *Proceedings of the 1975 ACM SIGMOD international conference on Management of data*, pp. 23–31, 1975.

[11] C. Landwehr, C. Heitmeyer, and J. Mclean, "A security model for military message systems," *ACM Transactions on Computer Systems*, vol. 2, no. 3, pp. 198–222, 1984.

[12] R. Sandhu, "Transaction control expressions for separation of duties," *Fourth Aerospace Computer Security Applications Conference*, pp. 282–286, 1988.

[13] National Computer Security Center, *Department of Defense Trusted Computer System Evaluation Criteria (TCSEC)*, 1985.

[14] S. Chokhani, "Trusted products evaluation," *Communications of the ACM*, vol. 35, no. 7, pp. 64–76, 1992.

[15] D. Clark and D. Wilson, "A comparison of commercial and military computer security models," *Proceedings of IEEE Symposium on Security and Privacy*, pp. 184–194, 1987.

[16] D. Brewer and M. Nash, "The Chinese Wall security policy," *Proceedings of the 1989 IEEE Symposium on Security and Privacy*, pp. 206–214, 1989.

[17] W. Boebert and R. Kain, "A Practical Alternative to Hierarchical Integrity Policies," *Proceedings of the 8th National Computer Security Conference*, pp. 18–27, 1985.

[18] R. Graubart, "On the need for a third form of access control," *Proceedings of the 12th National Computer Security Conference*, pp. 296–304, 1989.

[19] M. Harrison, W. Ruzzo, and J. Ullman, "Protection in operating systems," *Communications of the ACM*, vol. 19, no. 8, pp. 461–471, 1976.

[20] R. Sandhu, "The schematic protection model: its definition and analysis for acyclic attenuating schemes," *Journal of the ACM*, vol. 35, no. 2, pp. 404–432, 1988.

[21] M. Bishop, "Hierarchical take-grant protection systems," *Proceedings of the 8th ACM Symposium on Operating Systems Principles*, pp. 109–122, 1981.

[22] R. Sandhu, "The typed access matrix model," *Proceedings of the 1992 IEEE Symposium on Security and Privacy*, p. 122, 1992.

[23] S. OSBORN, R. SANDHU, and Q. MUNAWER, "Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies," *ACM Transactions on Information and System Security*, vol. 3, no. 2, pp. 85–106, 2000.

[24] D. Ferraiolo and R. Kuhn, "Role-based access control," *15th NIST-NCSC National Computer Security Conference, Baltimore, MD, October*, pp. 13–16, 1992.

[25] M. Nyanchama and S. Osborn, "The role graph model," in *Proceedings of the first ACM Workshop on Role-Based Access Control*. ACM, 1996.

[26] R. Sandhu, V. Bhamidipati, and Q. Munawer, "The ARBAC97 model for role-based administration of roles," *ACM Transactions on Information and System Security*, vol. 2, no. 1, pp. 105–135, 1999.

[27] S. Oh and R. Sandhu, "A model for role administration using organization structure," *Proceedings of the seventh ACM symposium on Access control models and technologies*, pp. 155–162, 2002.

[28] S. Oh, R. Sandhu, and X. Zhang, "An effective role administration model using organization structure," *ACM Transactions on Information and System Security (TISSEC)*, vol. 9, no. 2, pp. 113–137, 2006.

[29] J. Crampton and G. Loizou, "Administrative scope: A foundation for role-based administrative models," *ACM Transactions on Information and System Security (TISSEC)*, vol. 6, no. 2, pp. 201–231, 2003.

[30] E. Bertino, P. A. Bonatti, and E. Ferrari, "TRBAC: A temporal role-based access control model," *ACM Transactions on Information and System Security*, vol. 4, no. 3, pp. 191–233, 2001.

[31] J. Joshi, E. Bertino, and A. Ghafoor, "Temporal hierarchies and inheritance semantics for GTRBAC," *Proceedings of the seventh ACM symposium on Access control models and technologies*, pp. 74–83, 2002.

[32] J. Bacon, K. Moody, and W. Yao, "A model of OASIS role-based access control and its support for active security," *ACM Transactions on Information and System Security*, vol. 5, no. 4, pp. 492–540, 2002.

[33] D. Kuhn, "Mutual exclusion of roles as a means of implementing separation of duty in role-based access control systems," *Proceedings of the second ACM workshop on Role-based access control*, pp. 23–30, 1997.

[34] R. Simon and M. Zurko, "Separation of duty in role-based environments," *Proceedings of the 10th Computer Security Foundations Workshop (CSFW'97)*, p. 183, 1997.

[35] V. Gligor, S. Gavrila, and D. Ferraiolo, "On the formal definition of separation-of-duty policies and their composition," *Proceedings IEEE Symposium on Security and Privacy*, pp. 172–183, 1998.

[36] G. Ahn and R. Sandhu, "Role-based authorization constraints specification," *ACM Transactions on Information and System Security (TISSEC)*, vol. 3, no. 4, pp. 207–226, 2000.

[37] E. Bertino, E. Ferrari, and V. Atluri, "The specification and enforcement of authorization constraints in workflow management systems," *ACM Transactions on Information and System Security*, vol. 2, no. 1, pp. 65–104, 1999.

[38] G. Ahn, R. Sandhu, M. Kang, and J. Park, "Injecting RBAC to secure a Web-based workflow system," *Proceedings of the 5th ACM Workshop on Role-Based Access Control*, pp. 1–10, 2000.

[39] S. Kandala and R. Sandhu, "Secure role-based workflow models," *IFIP TC11/WG11. 3 Fifteenth Annual Working Conference on Database and Application Security*, 2001.

[40] L. Zhang, G. Ahn, and B. Chu, "A rule-based framework for role-based delegation and revocation," *ACM Transactions on Information and System Security (TISSEC)*, vol. 6, no. 3, pp. 404–441, 2003.

[41] R. Sandhu, "Role activation hierarchies," *Proceedings of the third ACM workshop on Role-based access control*, pp. 33–40, 1998.

[42] D. Ferraiolo, R. Sandhu, S. Gavrila, D. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," *ACM Transactions on Information and System Security (TISSEC)*, vol. 4, no. 3, pp. 224–274, 2001.

[43] N. Li, J.-W. Byun, and E. Bertino, "A critique of the ANSI standard on role-based access control," *IEEE Security and Privacy*, vol. 5, no. 6, pp. 41–49, Nov./Dec. 2007.

[44] R. Sandhu, "Rationale for the RBAC96 family of access control models," *Proceedings of the first ACM Workshop on Role-based access control*, 1996.

[45] D. Ferraiolo, R. Kuhn, and R. Sandhu, "RBAC standard rationale: Comments on "A critique of the ANSI standard on role-based access control"," *IEEE Security and Privacy*, vol. 5, no. 6, pp. 51–53, Nov./Dec. 2007.

[46] R. Sandhu, "The demand operation in the schematic protection model," *Information Processing Letters*, vol. 32, no. 4, pp. 213–219, 1989.