Role Engineering: Methods and Standards

Edward J. Coyne, Timothy R. Weil, D. Richard Kuhn

1 What is RBAC?

Roles with different privileges and responsibilities are a central feature of most organizations, and some computer applications dating back to at least the 1970s had limited forms of access control based on the user's role in an organization. These early role-based systems were typically *ad hoc* and application-specific, but general-purpose models for role based access control (RBAC) began to emerge in the 1990s. Today, most large firms are using some form of RBAC, and its popularity continues to grow [1].

A key feature of the RBAC model is that all access is through roles. A role is essentially a collection of permissions, and all users receive permissions only through the roles they are assigned, or from roles they inherit through a tree-like role hierarchy. Within an organization, roles tend to be relatively stable, while users and permissions are often numerous and rapidly changing. Controlling all access through roles therefore reduces the administrative burden for security managers and improves auditing of controls. Conceptually, the RBAC model enforces three rules:

1. Role assignment: a user can access a permission only if the user has been assigned a role.
2. Role authorization: a user's active role must be authorized for this user.
3. Permission authorization: a user can access a permission only if the permission is authorized for the user's active role.

Together, these rules ensure that users have access only to permissions authorized for them, but for practical applications, extra controls may be needed. Separation of duty (SoD) is probably the most common additional constraint used in large organizations, and RBAC was designed to support SoD. A typical SoD requirement is that a user may not both initiate and approve a large purchase request, a rule that can be enforced in RBAC by prohibiting any one user from having a collection of roles that give access to both the 'purchase initiation' and 'purchase approval' permissions. Other constraints, such as the number of users allowed in a role, or more complex issues such as access authorization by time of day may be supported by different RBAC systems.

Recognizing the need for some commonality among the various RBAC models then in use, the National Institute of Standards and Technology proposed the NIST Model for RBAC in 2000. Following debate and comment within the RBAC and security communities, NIST made revisions and proposed a U.S. national standard for RBAC through the International Committee for Information Technology Standards (INCITS), approved in 2004 as standard INCITS 359-2004. This standard includes four components: (1) Core RBAC specifies a minimum collection of RBAC elements, element sets, and relations required in any RBAC system, but the other components are independent of each other and may be implemented separately. (2) Hierarchical RBAC adds relations for supporting role hierarchies and introduces the concept of a role's set of authorized users and authorized permissions. (3) Static Separation of Duty makes it possible to prevent authorizing various combinations of roles to users. (4) Dynamic Separation of Duty

Relations allows users to be authorized for some combinations of roles but prevents users from activating these roles simultaneously. Collectively, these standard components can support an extensive variety of complex access control solutions, making RBAC a popular model for organizations that often have to support thousands of users and permission controls.

2 What is role engineering?

While conceptually simple, RBAC can be difficult to implement in large organizations. Different organizational components may not use the same names for similar positions, and in many cases it is not clear exactly what set of permissions users in various positions need to do their jobs. Even after a complete picture of user permissions emerges, structuring these permissions into a practical set of roles may be challenging. The process of developing an RBAC structure for an organization has become known as "role engineering". It is not uncommon for large firms to discover the need for hundreds of roles, which must then be structured into an efficient hierarchy and permissions for the various roles realized fully down to the many IT systems in the organization. While challenging, some efficient strategies – drawing on requirements engineering methodologies – have been devised for the role engineering process. As with requirements, getting the role structure right from the start means significant financial savings and fewer operational problems down the road.

3 How do we do role engineering?

Role engineering follows three fundamental approaches: Top-down, Bottom-up, and Hybrid.

Top-down Approach

The top-down approach may be considered as a “clean slate” method. Here a group of functional stakeholders or subject-matter experts (SMEs), with the help of a facilitator, identifies role names and associates these with permissions. A methodology is employed in which the SMEs identify actors and their operations on objects. The operations on object are, by definition, permissions. The permissions are those needed by an actor when using an IT system. They are high level in the realm of the SMEs and are not the technical permissions that an IT system would work with internally. Once a set of high-level permissions has been established, IT specialists can translate the high-level permissions into low level accesses within one or more IT systems.

One difficulty of the top-down approach is that it can be a challenge to assemble a team of SMEs for the time period required to accomplish the role engineering objectives.

Bottom-up Approach

The bottom-up approach seeks to capitalize on existing access definitions that are available within an IT environment. The concept is that over time the organization has invested time and effort to define a set or sets of access control rules and conventions. Rather than trying to re-invent the wheel, the desire is to channel existing rules for access to objects toward a role-based concept. It is necessary to analyze the IT systems in which it is believed that usable rules and definitions can be harvested. General system management tools can help with gathering relevant data and general analysis tools such databases and spreadsheets can assist with manipulating the data and shaping it to support a role-based concept. Beyond these capabilities specialized tools may be acquired that perform more sophisticated analyses such as clustering of similar items and displaying them for ease in analysis

One difficulty of the bottom-up approach is that since results are collected from individual system, the results are typically not coherent and must be normalized to a common framework.

Hybrid Approach

As its name suggests, the hybrid approach seeks to capitalize on the results of both the top-down and the bottom-up approaches. Data is gathered using bottom-up methods and the results are used by SMEs while conducting top-down activities. This can potentially save time and effort on the part of the SMEs while still producing valid results.

A difficulty of the hybrid approach carries over from both the top-down and bottom-up approaches, namely the problem of assembling SMEs over a sufficient period of time and the fact that access control data from individual systems has to be normalized over the enterprise.

4 What is the need for a role engineering standard?

Federal directives (Critical Infrastructure Executive Order 13010[1], Presidential Directive 63[2]) and post 9/11 mandates (Critical Infrastructure Protection in the Information Age EO13231[3]) highlight the System Security Management (CIP-007) area as mandating nine requirements specifically dealing with end-user account management.. A recent report by a Florida Power and Electric utility (JEA) cited RBAC protection mechanisms as an example of the need for role engineering to meet these CIP needs [2]:

"***Roles that are engineered correctly*** will satisfy several R5 requirements because roles will provide a mechanism that ensures access to information is on a need-to-know basis, verifies access privileges efficiently and minimizes significantly the risk of unauthorized system access".

[1] http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=1996_register&docid=fr17jy96-92.pdf

[2] http://www.fas.org/irp/offdocs/pdd/pdd-63.htm

[3] http://www.fas.org/irp/offdocs/eo/eo-13231.htm

A role engineering standard is needed for its ability to assist in initiating a role engineering effort. This is accomplished by providing a framework for defining one or more role engineering efforts. The framework will guide the project definition and specify the concepts and artifacts involved in role engineering.

Having a role engineering standard will also promote intercommunication among organizations that are conducting role engineering efforts. Comparing methods, results, and identifying lessons learned can all benefit from the use of standard terms and concept. For example, role name catalogs, permission catalogs, and constraint catalogs can be identified for each role engineering effort and with sufficient detail in the standard these items can be inter-compared and possibly exchange content.

5 What kind of standard is being developed?

A role engineering standard must be general enough to encompass a variety of role engineering processes. It should be a framework and not a prescription. It should be able to answer the question of What? Each role engineering effort can answer its own questions of Why?, When?, How?, Who? And Where?.

Thus what is needed is less of a role engineering "cookbook" and more of a reference framework. A cookbook would not practically cover the many role engineering approaches and processes that may be used and if the assumptions behind the cookbook do not match those of the organization, the cookbook approach could lead to waste of resources.

The reference model needed will provide a framework to support role engineering efforts and to organize their results. Concepts and terminology from a reference model can assist in designing and implementing a role engineering effort. Also, a reference model can serve as a tool for assessing the comparability of various role definition policies and practices.

Over the past several years, the ANSI INCITS CS1.1 Task Group[4] has developed three publications to update and enhance the original ANSI 359-2004 RBAC standard. A brief description of this body of work is given here –

To meet the need for consistent sets of RBAC components, the RBAC Implementation and Interoperability Standard (RIIS/ANSI INCITS-459) specifies in greater detail how to design RBAC products to conform to INCITS 359. As published in 2011, this standard can facilitate comparisons among RBAC products, and its interoperability specifications enables translations from one RBAC implementation to another [2].

In response to comments received over the five-year life of the current RBAC standard (ANSI 359-2004), INCITS CS1.1 has developed a policy-enhanced RBAC standard to accommodate

[4] http://standards.incits.org/a/public/group/cs1.1

importation of arbitrary constraints, including attributes of all types. This enhanced model will maintain the advantages of RBAC while providing a mechanism for including attributes in access-control decisions. Publication of this updated standard (RBAC ANSI 359-2011) is expected in 2012 [3].

The CS1.1 working group have recently completed an RBAC Policy-Enhanced standard (RPE) that provides a framework and functional specifications to handle the relationship between roles and dynamic constraints. Some of the administrative and user permission review advantages of RBAC are retained while allowing the access control system to work in a rapidly changing environment. The RPE defines the scope and context for role-role, user-role, and attribute-sensitive dynamic constraints which can be implemented in a run time environment.

To complement this suite of access control (RBAC) standards, the ANSI INCITS CS1.1Task Group on RBAC has a role definition reference model under development. An initial draft is available for further definition and development. Comments are welcome and participation in the CS1.1 Task Group is open to interested parties.

[1] A.C. O'Connor and R.J. Loomis (December 2010). Economic Analysis of Role-based Access Control. Research Triangle Institute. http://csrc.nist.gov/groups/SNS/rbac/documents/20101219_RBAC2_Final_Report.pdf.

[2] K.D. Gordon, J.E. Michelman, B. Waldrup, R.D. Slater, Accounting Data Security at JEA Using Role-Based Access Controls, University of North Florida, pgs. 34, online available http://aaahq.org/AM2011/display.cfm?Filename=SubID%5F2382%2Epdf&MIMEType=application%2Fpdf

[3] Edward J. Coyne, D. Richard Kuhn, Timothy R. Weil (2011) ANSI/INCITS 459-2011 - Information Technology - Requirements for the Implementation and Interoperability of Role Based Access Control pgs, 27. http://www.techstreet.com/cgi-bin/detail?doc_no=incits|459_2011;product_id=1777986

[4] D.R. Kuhn**,** E.J. Coyne, T.R. Weil, "Adding Attributes to Role Based Access Control"**,** IEEE Computer, vol. 43, no. 6 (June, 2010), http://csrc.nist.gov/groups/SNS/rbac/documents/kuhn-coyne-weil-10.pdf.

Disclaimer

Role Engineering: Methods and Standards

Edward J. Coyne, Timothy R. Weil, D. Richard Kuhn

1 What is Role Based Access Control?

Roles with different privileges and responsibilities are a central feature of most organizations, and some computer applications dating back to at least the 1970s had limited forms of access control based on the user's role in an organization. These early role-based systems were typically *ad hoc* and application-specific, but general-purpose models for role based access control (RBAC) began to emerge in the 1990s. Today, most large firms are using some form of RBAC, and its popularity continues to grow [1].

A key feature of the RBAC model is that all access is through roles. A role is essentially a collection of permissions, and all users receive permissions only through the roles they are assigned, or from roles they inherit through a tree-like role hierarchy. Within an organization, roles tend to be relatively stable, while users and permissions are often numerous and rapidly changing. Controlling all access through roles therefore reduces the administrative burden for security managers and improves auditing of controls. Conceptually, the RBAC model enforces three rules:

1. Role assignment: a user can access a permission only if the user has been assigned a role.
2. Role authorization: a user's active role must be authorized for this user.
3. Permission authorization: a user can access a permission only if the permission is authorized for the user's active role.

Together, these rules ensure that users have access only to permissions authorized for them, but for practical applications, extra controls may be needed. Separation of duty (SoD) is probably the most common additional constraint used in large organizations, and RBAC was designed to support SoD. A typical SoD requirement is that a user may not both initiate and approve a large purchase request, a rule that can be enforced in RBAC by prohibiting any one user from having a collection of roles that give access to both the 'purchase initiation' and 'purchase approval' permissions. Other constraints, such as the number of users allowed in a role, or more complex issues such as access authorization by time of day may be supported by different RBAC systems.

Recognizing the need for some commonality among the various RBAC models then in use, the National Institute of Standards and Technology proposed the NIST Model for RBAC in 2000. Following debate and comment within the RBAC and security communities, NIST made revisions and proposed a U.S. national standard for RBAC through the International Committee for Information Technology Standards (INCITS), approved in 2004 as standard INCITS 359-2004. This standard includes four components: (1) Core RBAC specifies a minimum collection of RBAC elements, element sets, and relations required in any RBAC system, but the other components are independent of each other and may be implemented separately. (2) Hierarchical RBAC adds relations for supporting role hierarchies and introduces the concept of a role's set of authorized users and authorized permissions. (3) Static Separation of Duty makes it possible to prevent authorizing various combinations of roles to users. (4) Dynamic Separation of Duty

Relations allows users to be authorized for some combinations of roles but prevents users from activating these roles simultaneously. Collectively, these standard components can support an extensive variety of complex access control solutions, making RBAC a popular model for organizations that often have to support thousands of users and permission controls.

2 What is role engineering?

While conceptually simple, RBAC can be difficult to implement in large organizations. Different organizational components may not use the same names for similar positions, and in many cases it is not clear exactly what set of permissions users in various positions need to do their jobs. Even after a complete picture of user permissions emerges, structuring these permissions into a practical set of roles may be challenging. The process of developing an RBAC structure for an organization has become known as "role engineering". It is not uncommon for large firms to discover the need for hundreds of roles, which must then be structured into an efficient hierarchy and permissions for the various roles realized fully down to the many IT systems in the organization. While challenging, some efficient strategies – drawing on requirements engineering methodologies – have been devised for the role engineering process. As with requirements, getting the role structure right from the start means significant financial savings and fewer operational problems down the road.

3 How do we do role engineering?

Role engineering follows three fundamental approaches: Top-down, Bottom-up, and Hybrid.

Top-down Approach

The top-down approach may be considered as a “clean slate” method. Here a group of functional stakeholders or subject-matter experts (SMEs), with the help of a facilitator, identifies role names and associates these with permissions. A methodology is employed in which the SMEs identify actors and their operations on objects. The operations on object are, by definition, permissions. The permissions are those needed by an actor when using an IT system. They are high level in the realm of the SMEs and are not the technical permissions that an IT system would work with internally. Once a set of high-level permissions has been established, IT specialists can translate the high-level permissions into low level accesses within one or more IT systems. One difficulty of the top-down approach is that it can be a challenge to assemble a team of SMEs for the time period required to accomplish the role engineering objectives.

Bottom-up Approach

The bottom-up approach seeks to capitalize on existing access definitions that are available within an IT environment. The concept is that over time the organization has invested time and effort to define a set or sets of access control rules and conventions. Rather than trying to re-

invent the wheel, the desire is to channel existing rules for access to objects toward a role-based concept. It is necessary to analyze the IT systems in which it is believed that usable rules and definitions can be harvested. General system management tools can help with gathering relevant data and general analysis tools such databases and spreadsheets can assist with manipulating the data and shaping it to support a role-based concept. Beyond these capabilities specialized tools may be acquired that perform more sophisticated analyses such as clustering of similar items and displaying them for ease in analysis

One difficulty of the bottom-up approach is that since results are collected from individual system, the results are typically not coherent and must be normalized to a common framework.

Hybrid Approach

As its name suggests, the hybrid approach seeks to capitalize on the results of both the top-down and the bottom-up approaches. Data is gathered using bottom-up methods and the results are used by SMEs while conducting top-down activities. This can potentially save time and effort on the part of the SMEs while still producing valid results.

A difficulty of the hybrid approach carries over from both the top-down and bottom-up approaches, namely the problem of assembling SMEs over a sufficient period of time and the fact that access control data from individual systems has to be normalized over the enterprise.

4 What is the need for a role engineering standard?

Federal directives (Critical Infrastructure Executive Order 13010[1], Presidential Directive 63[2]) and post 9/11 mandates (Critical Infrastructure Protection in the Information Age EO13231[3]) highlight the System Security Management (CIP-007) area as mandating nine requirements specifically dealing with end-user account management.. A recent report by a Florida Power and Electric utility (JEA) cited RBAC protection mechanisms as an example of the need for role engineering to meet these CIP needs [2]:

"***Roles that are engineered correctly*** will satisfy several R5 requirements because roles will provide a mechanism that ensures access to information is on a need-to-know basis, verifies access privileges efficiently and minimizes significantly the risk of unauthorized system access".

A role engineering standard is needed for its ability to assist in initiating a role engineering effort. This is accomplished by providing a framework for defining one or more role engineering

[1] http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=1996_register&docid=fr17jy96-92.pdf

[2] http://www.fas.org/irp/offdocs/pdd/pdd-63.htm

[3] http://www.fas.org/irp/offdocs/eo/eo-13231.htm

efforts. The framework will guide the project definition and specify the concepts and artifacts involved in role engineering.

Having a role engineering standard will also promote intercommunication among organizations that are conducting role engineering efforts. Comparing methods, results, and identifying lessons learned can all benefit from the use of standard terms and concept. For example, role name catalogs, permission catalogs, and constraint catalogs can be identified for each role engineering effort and with sufficient detail in the standard these items can be inter-compared and possibly exchange content.

5 What kind of standard is being developed?

A role engineering standard must be general enough to encompass a variety of role engineering processes. It should be a framework and not a prescription. It should be able to answer the question of What? Each role engineering effort can answer its own questions of Why?, When?, How?, Who? And Where?.

Thus what is needed is less of a role engineering "cookbook" and more of a reference framework. A cookbook would not practically cover the many role engineering approaches and processes that may be used and if the assumptions behind the cookbook do not match those of the organization, the cookbook approach could lead to waste of resources.

The reference model needed will provide a framework to support role engineering efforts and to organize their results. Concepts and terminology from a reference model can assist in designing and implementing a role engineering effort. Also, a reference model can serve as a tool for assessing the comparability of various role definition policies and practices.

Over the past several years, the ANSI INCITS CS1.1 Task Group[4] has developed three publications to update and enhance the original ANSI 359-2004 RBAC standard. A brief description of this body of work is given here –

To meet the need for consistent sets of RBAC components, the RBAC Implementation and Interoperability Standard (RIIS/ANSI INCITS-459) specifies in greater detail how to design RBAC products to conform to INCITS 359. As published in 2011, this standard can facilitate comparisons among RBAC products, and its interoperability specifications enables translations from one RBAC implementation to another [2].

In response to comments received over the five-year life of the current RBAC standard (ANSI 359-2004), INCITS CS1.1 has developed a policy-enhanced RBAC standard to accommodate importation of arbitrary constraints, including attributes of all types. This enhanced model will maintain the advantages of RBAC while providing a mechanism for including attributes in

[4] http://standards.incits.org/a/public/group/cs1.1

access-control decisions. Publication of this updated standard (RBAC ANSI 359-2011) is expected in 2012 [3].

The CS1.1 working group have recently completed an RBAC Policy-Enhanced standard (RPE) that provides a framework and functional specifications to handle the relationship between roles and dynamic constraints. Some of the administrative and user permission review advantages of RBAC are retained while allowing the access control system to work in a rapidly changing environment. The RPE defines the scope and context for role-role, user-role, and attribute-sensitive dynamic constraints which can be implemented in a run time environment.

To complement this suite of access control (RBAC) standards, the ANSI INCITS CS1.1Task Group on RBAC has a role definition reference model under development. An initial draft is available for further definition and development. Comments are welcome and participation in the CS1.1 Task Group is open to interested parties.

[1] A.C. O'Connor and R.J. Loomis (December 2010). Economic Analysis of Role-based Access Control. Research Triangle Institute.
http://csrc.nist.gov/groups/SNS/rbac/documents/20101219_RBAC2_Final_Report.pdf.

[2] K.D. Gordon, J.E. Michelman, B. Waldrup, R.D. Slater, Accounting Data Security at JEA Using Role-Based Access Controls, University of North Florida, pgs. 34, online available http://aaahq.org/AM2011/display.cfm?Filename=SubID%5F2382%2Epdf&MIMEType=application%2Fpdf

[3] Edward J. Coyne, D. Richard Kuhn, Timothy R. Weil (2011) ANSI/INCITS 459-2011 - Information Technology - Requirements for the Implementation and Interoperability of Role Based Access Control pgs, 27.
http://www.techstreet.com/cgi-bin/detail?doc_no=incits|459_2011;product_id=1777986

[4] D.R. Kuhn**,** E.J. Coyne, T.R. Weil, "Adding Attributes to Role Based Access Control"**,** IEEE Computer, vol. 43, no. 6 (June, 2010), http://csrc.nist.gov/groups/SNS/rbac/documents/kuhn-coyne-weil-10.pdf.

Disclaimer