NIST National Institute of Standards and Technology

Small Business Innovation Research (SBIR)

A Marketing Survey of Civil Federal Government Organizations to Determine the Need for a Role-Based Access Control (RBAC) Security Product

SETA

Small Business Innovation Research (SBIR)

A Marketing Survey of Civil Federal Government Organizations to Determine the Need for a Role-Based Access Control (RBAC) Security Product

Phase 2

Charles L. Smith, Sr.
Edward J. Coyne
Charles E. Youman
Srinivas Ganta

July 1996

This material is based upon work supported by the Department of Commerce under contract number 50-DKNB-5-00188. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the Department of Commerce.

SETA CORPORATION
6862 Elm Street
McLean, VA 22101-3833

TABLE OF CONTENTS

P
age #

TABLE OF CONTENTS (Concluded)

P
age #

LIST OF FIGURES

P

age #

LIST OF TABLES

P

age #

EXECUTIVE SUMMARY

There is a recognized need for a more robust method of performing security controls in computer network systems [Ferraiolo *et al.* 1992]. One promising method is called role-based access control (RBAC). Although the basic ideas for RBAC have existed for over 20 years, there has been a recent resurgence of interest in RBAC, largely because of the disenchantment with traditional mandatory and discretionary access controls by many users. The essence of RBAC is that rights and permissions are assigned to roles rather than to individual users. Users acquire these rights and permissions by virtue of being assigned membership in appropriate roles. This method makes the administration of security access much simpler than with current approaches.

Although RBAC is receiving much attention among potential users and vendors, it is not known what the consumer demand will be for RBAC products. Consequently, this marketing survey was conducted. This study is essentially a marketing survey to identify customer requirements regarding their security needs for information processing systems and to determine whether an RBAC product can meet these requirements.

Information system requirements must originate from the system users, that is, from the organizational stakeholders who are concerned about the performance of their system and whose jobs are affected by the system's capabilities. Regarding security aspects of a system, these stakeholders are generally called security managers, security officers, security administrators, or some similar name.

There are existing packages that sometimes purport to be role-based security implementations, but these packages are greatly limited in their capabilities to emulate the robustness of an RBAC product as manifested in the reference material.

It should be understood that RBAC is not a replacement for the existing mandatory access control (MAC) and discretionary access control (DAC) products, it is an adjunct to them. Moreover, RBAC adds security capabilities that are not resident in the current security products.

Some stakeholders understand that security needs represent a set of complex issues, yet their purchased security packages are often a response to "we need a security product" without understanding what the actual security issues are nor having an appreciation for the need of a capable security system. The complexity of a software issue should be clearly understood prior to the creation or purchase of a software solution [Draier 1994]. Thus, before a vendor (who may create a commercial solution) or a customer (who may purchase or create a solution) invents a more robust security package, it is mandatory that the targeted security requirements be well defined. This is the principal purpose of this study.

One approach for identifying information system security requirements is to use a questionnaire.

The questionnaire should have at least the following characteristics [GAO 1991]:

.Contain relevant questions

.Be clear and understandable to the stakeholders

.Be comprehensive

The questionnaire used in this marketing survey is for a *descriptive evaluation* of specific conditions and needs of relevant customers. The survey itself involves identifying the following six elements:

.The appropriate information required (we used pertinent reference material to identify the information needed for identifying customer security needs),

.The relevant population (we selected federal civil government organizations with networked computer systems),

.The approach for sampling an appropriate number of stakeholders from the population to assure accurate results (we used random sampling of nearly 50% of the identified population, where at least 33% is recommended),

.The method for eliciting the relevant knowledge from the stakeholders (we used a structured questionnaire for recording stakeholder responses using face-to-face interviews by SETA personnel),

.The proper timing and frequency of information collection (we used a process of calling and arranging interviews with stakeholders who were randomly drawn from a population list offered by the National Institute of Standards and Technology), and

.An analysis plan (we created an analysis plan based on pertinent references prior to forming the questionnaire).

Much of the data analysis process consisted of simply presenting customer security requirements information in a manner that is clear and readily understandable. This is partly because the information is both qualitative (i.e., descriptive) and quantitative (i.e., numerical). No sophisticated statistical information is offered since this type of data might tend to cloud the issue rather than clarify it for many of the readers. The fundamental results being sought here are "What did the customers say their security needs were?" and "Might these security needs be met by using a complementary RBAC product?"

The sample size from the relevant population is not mathematically rationalized but does represent nearly one-half of the relevant population, which was considered large enough to support statements that the summarized questionnaire results accurately represent federal civil government security requirements. Through inference, because of similar situations, it is believed that these security requirements also represent the security needs for state government and commercial users.

The study questionnaire used here was developed by a group through an iterative consensual evaluation of an original strawman set of questions. Using a face-to-face interview technique, although the questionnaire was never modified, the manner in which some of the questions were read to the interviewee changed to a more appropriate form for ease in understanding.

Also, many of the questions required that the interviewer use an example to explain the question, and sometimes, more than one example was required. This made the stakeholder response process simpler and easier. On occasion, the interviewee did this to the benefit of both the interviewee and the interviewers.

The questions generally involved yes/no answers or multiple choice answers to minimize the effort required of the interviewee. However, even questions with yes/no answers can be answered as "yes," "no," "sometimes yes and sometimes no," or "I don't know," and even "I do not wish to answer that question" so that there is not necessarily a binary option. The questions were asked in the exact order in which they appeared in the questionnaire. The interviewee's anonymity was guaranteed and no results were attributed to any interviewee or to their specific organization.

With each interview, a rapport was developed between the interviewers and the interviewee, or interviewees, by ensuring anonymity of the interviewees and that the interviewers' opinions about the subject matter (i.e., information system security requirements) were unbiased.

Unfortunately, the security requirements (the what) for information systems are dynamic and represent a moving target. Thus any recommended solution to meeting security requirements (the how) should include considerations of extensibility for modifying, adding, or deleting different capabilities at a later time for meeting the dynamic changes of security requirements in response to new demands on the system. Therefore, whatever statements are made here about the security needs of organizational information systems, are at best a temporal approximation of the security needs (current and anticipated) as they are perceived by the stakeholders we interviewed.

1.0 INTRODUCTION

SETA has a Small Business Innovation Research (SBIR) Phase 2 contract with the National Institute of Standards and Technology (NIST) to identify the potential market for RBAC products that will improve the effectiveness and administration of access control for information systems.

The Phase 1 effort [Feinstein *et al*. 1995] explored the need for RBAC, developed a theoretical model and high-level specification of RBAC, demonstrated the concepts of the theoretical model with a working demonstration, explored approaches to RBAC implementation, and identified tools for RBAC administration. The Phase 1 study found that RBAC is a useful concept for controlling an organization's information assets and in administering that control.

After the Phase 1 effort, SETA felt that the application of RBAC research could be implemented as both a service and a product. Services such as role engineering could be provided to users who need such assistance in implementing an RBAC product. An RBAC product could consist of add-ons to existing security products. However, before exploring these alternatives, there was a need to identify the potential security needs among computer system users.

1.1 Background

The following definitions are useful in understanding the results of this study:

- **Role** - is a job function within an organization that briefly describes the activities or responsibilities of a system user who is assigned to the role.

- **RBAC** - is the process of assigning rights and permissions to roles rather than to individual users, with users then assigned to roles as appropriate. An RBAC product is a complementary capability to existing mandatory access control (MAC) and discretionary access control (DAC) security capabilities.

- **Role Administration** - is the capability to define the roles of system users.

More definitions are listed in Appendix G, *Some Relevant Definitions*.

The concept of a role in access control is currently being used in at least two different ways. Some use 'role' to mean a named collection of permissions.

Others use 'role' to mean a named collection of permissions and a named collection of users. There has been no adjudication as to which of these is the more acceptable and both are considered, at present, as reasonable and correct. For a more detailed discussion of RBAC, see Appendix F, *A Brief Overview of an RBAC Product*.

The purpose of computer security is to protect an organization's valuable resources, such as information, software (namely applications and databases), and hardware.

Based on the following eight elements [Guttman and Roback 1995], computer security should:

- Support the organization's mission,
- Be an integral part of sound management,
- Be cost-effective,
- Have responsibilities and accountability that are made explicit,
- Require a comprehensive and integrated approach,
- Be periodically assessed,
- Be constrained by societal factors, and
- Be the responsibility of system owners who are also responsible for system-related events outside the organization.

However it is achieved, computer security for federal organizations ought to improve the services provided to the citizens who are served by those organizations.

The implementation of a security product to protect an organization's information or applications requires a trade-off between the depth of protection and the ease of access by users. The dilemma is generally that the more security one has, the more of a barrier it becomes. For example, if you were to place five deadbolt locks on your door, you would feel more secure but you would have a more difficult time greeting your friends [Scott 1996]. If at all possible, a desirable benefit of a secure system would be the capability to provide the client with the safety of five deadbolt locks but with the ease of permitted entry equivalent to turning a single doorknob. That is, the security product, however complex, should be essentially transparent to the user, because it depends on recognizing the user in some essential way, e.g., the user's role.

An RBAC product is generally complementary to the existing security measures and can be an adjunct, not a replacement, although that is an option.

Security is defined as protection against unwanted disclosure, modification, or destruction of data (and applications). Security not only means protecting classified information but also means protecting unclassified-but-sensitive data, or private information.

A person who is in charge of the security aspects of a computer system (or a part of it) is called the system administrator or security administrator or the database administrator.

The *security administrator* has essentially three responsibilities [Russell and Gangemi 1991]:

)*Overall security planning and administration*, which includes working with management to set a security policy for the organization, publicizing it, and gaining management support for it; performing risk analyses and disaster planning; monitoring employees; training users; answering their questions; and performing many other related activities.

)*Day-to-day security administration*, which includes creating accounts and assigning security profiles for users such as their passwords, password controls, login controls, and role-based activities, if they apply. The security administrator should make sure that there are no security holes in the system.

)*Day-to-day system administration*, which includes keeping the system running, doing daily backups, creating user accounts, and performing many other related activities.

A concept related to security is *trustworthiness*, which is the level of confidence that a system will do what it is expected to do with respect to security, safety, availability, and reliability. *Safety* means freedom from danger, risk, or injury; *availability* means the capability of a system to perform properly when it is needed; and *reliability* means the capability of a system to do what it is expected to do for some defined period. The opposite of availability is often referred to as *denial of service*.

Examples of trust issues include:

- Maintaining the privacy of employees or customer records,
- Ensuring the correctness and accuracy of private records,
- Responding to users' requests for service at the time they desire the service, and
- Protecting national or organizational sensitive information from unauthorized disclosure or modification.

Security requirements include protecting the integrity, secrecy, and confidentiality of key software systems, databases, and data networks. *Integrity* (or accuracy) means that the system must not corrupt the information or allow any unauthorized malicious or accidental changes to it. A related variation of integrity is *authenticity*, which is a way to verify the origin of data by determining who entered or sent it and by recording when it was sent and received. The objectives of *data integrity* typically include:

- Ensuring the consistency of data values within a computer system,
- Recovering to a known consistent state in the event of a system failure,
- Ensuring that data are modified only in authorized ways, whether by users or by the system, and
- Maintaining consistency between information internal to the computer system and the realities of the outside world.

Secrecy ensures that users access only information that they are allowed to see. *Confidentiality* ensures the protection of private information, such as payroll data, as well as sensitive corporate data, such as internal memos or competitive strategy documents [Russell and Gangemi 1991].

Security requirements are usually described in terms of the following control objectives: user accountability and access control. *Accountability* means that the system knows who you are and what you are doing. Accountability usually involves user identification and authentication (who are you?), trusted path (what occurred?), and audit (what did you do?). Security products that attempt to control access and data integrity also perform accountability. There are three basic types of products for *security access control*: 1) discretionary access control, 2) mandatory access control, and 3) role-based access control. These are described in Appendix A, *Access Control*.

Organizations rely heavily on information processing systems to meet the individual access requirements for their computer system network environment security. Access issues usually involve the following: who can access, what data can be accessed, how the data can be accessed, and when the data can be accessed. If an organization's users are limited in their capability to utilize the information processing system due to security requirements, then the organization may become less efficient and hence less competitive. In the case of Federal organizations, the organization may become less able to effectively communicate with the public and relevant others.

A great deal of importance is placed on being in-step with other organizations of like size and function in terms of a security program. Also, with national and international competition among most commercial and Federal organizations, the privacy and value of data are becoming more highly guarded against intruders, internal and external. Such considerations, along with more traditional risk analyses, are used as the basis for determining security requirements and implementing safeguards.

There is always a trade-off between the cost associated with buying, installing, and operating a better security system versus the potential losses associated with owning a poorer security system. The security concerns for both civil government and commercial organizations usually involve budgetary, regulatory, and reputation considerations. The difference is that government organizations are usually more concerned about maintaining public confidence.

A set of general security requirements, based on interviews by NIST, has been developed by NIST. NIST interviewed many Government and commercial organizations to assess these organizations' current and future general security needs in all three major areas: commercial, civil government (Federal or state [one case]), and military. The principal security requirements are listed in Appendix D, *User Security General Requirements as Determined by NIST*, but include that these organizations:

- Have unique security needs,
- Have organizational security requirements that change over time and cannot be totally specified at the time of security product acquisition, and
- Feel that security standards have not emerged that will allow integrating security across a multivendor environment (i.e., a heterogeneous open system).

For organizations that process unclassified-but-sensitive information, the availability of a greater variety of trusted security products should:

- Go beyond command and control in terms of functionality and flexibility;
- Address data integrity in a more direct and user-friendly manner; and
- Ensure that vendors consider new products that directly address discretionary and non-discretionary controls, such as 1) RBAC, 2) separation of duties, 3) separation of transactions, and 4) user-oriented least privilege.

While the Trusted Computer System Evaluation Criteria (TCSEC, sometimes called the "Orange Book" [U. S. DoD 1985]) C2-level provides many baseline security requirements, it falls short in providing security features (such as resource access control) and assurances that many users need. An overview of C2 is provided in Appendix I, *C2 Security Control*. The TCSEC label-based mandatory access control policy is commonly believed by commercial and civil government users to be inappropriate for their security needs. Users generally believe that another method for providing security is required. The RBAC approach is being investigated as a possible method that will satisfy most computer network users.

An RBAC product offers at least three advantages over other models of access-control security systems:

)it simplifies existing access-control management functions so that security control can be performed quickly and at lower cost,

)it provides access control functions not readily available in the other models, and

)it provides a model for a security system that can be easily extended to provide additional security capabilities to meet dynamic and unique user security requirements.

By providing a lower-cost robust extensible security-control product, an RBAC product may prove to be the security system that meets the requirements expressed by many commercial, government, and military users for the 1990s and into the 21st century.

1.2 Purpose

The purpose of the SETA security requirements study is to determine the specific system-oriented security requirements as expressed by a selected set of users.

Once these security requirements have been determined, the requirements will be used to develop an RBAC concept design or high-level architecture. Thus a statement of the issue to be resolved here is defined as follows:

> **Identify the computer system security requirements of civil government organizations to determine if an RBAC product can be formulated that will economically and effectively meet user needs.**

1.3 Approach

A process was used that included the following nine steps:

)Formulate the study objectives

)Develop a questionnaire

)Develop an analysis plan

)Identify potential interviewees

)Set up structured interviews

)Gather the information

)Perform an evaluation of the data

)Tabulate and formally display the collected data

)Interpret the data in the form of client security requirements

1.4 Summary

The results of the study are shown in Table 1-1, *Summary of Federal Organization Security Environment versus RBAC Capabilities*, to reveal the Federal organization security requirements and potential RBAC solutions. An assessment of the correlation between interviewee security requirements with RBAC capabilities showed a strong relationship between an RBAC product and the security needs of the organizations we interviewed.

Table 1-1. Summary of Federal Organization Security Environment versus RBAC Capabilities

Client Security Environment	Benefits of an RBAC Product
Many personnel changes	Provides for easily handling personnel changes
Many job assignment and definition changes are needed in many organizations	Provides a simple product for handling changes to job assignments and definitions
All security principles are desired	All security principles are easily provided
Systems have many users and many organizations have a high job turnover rate	Provides for easily handling many users and high user flux
Many security administrators are involved	Provides for easily handling administrative duties
Security management needs to be unified in many organizations	Provides for a simplified management of user access
Roles and role constraints are desired by many organizations	Provides access for user-defined roles and role constraints
An analysis of user roles is needed by many organizations	Role engineering can be provided to assure accurate role definitions for any organization
Many administrators are needed to manage security	Provides for a greater capability for each security administrator
All types of accountability are needed, especially user accountability	Provides for a simplified auditing of user accountability

2.0 ANALYSIS AND RESULTS

This section of the report describes the analysis performed (the process and questionnaire generated) and the results of this analysis in terms of the findings that were identified.

2.1 Analysis Process

A marketing survey was conducted as a first step in understanding the potential demand for a RBAC product. Marketing surveys are a commonly-used approach in the commercial world to establish if there is a consumer need (or demand) for a product (or service) and quantify this need, if possible, to determine its potential market value. A marketing survey can be considered to be an initial phase of a complete life cycle process for products which are eventually considered to be marketable. The life cycle process will extend from a successful market survey through concept, development, product prototype, product manufacture, commercialization, advertising, distribution, improvement, and to its eventual discontinuation. A marketing survey process is illustrated in Figure 2-1, *An RBAC Product Marketing Survey Process*, for the RBAC product. The survey process assumes that one has already developed the needed questionnaire.

Figure 2-1. An RBAC Product Marketing Survey Process

When a product that is in great demand is the first product of its kind to enter the market, and it truly satisfies consumer demand, then the product can be quite profitable, usually for many years. Furthermore, if the product is designed for easy extensibility (e.g., as the personal computer was), then it can be periodically upgraded to keep pace with consumer demand (and perhaps even drive consumer demand) and stay atop its product list of competing products for many years.

Potential customers are people who might consider purchasing an appropriately-modified system environment (e.g., Windows NT, ORACLE 7, or NetWare 4) for meeting their security needs. A principal objective of this marketing survey is to ascertain the security requirements of potential users and what they think about RBAC as a product for potentially meeting their security needs. In other words, SETA needs to characterize the user's application environment relative to RBAC potential capabilities. If there is a good match, then the user is a good candidate for adding RBAC to their system, if not, then they are not a good candidate. However, since an RBAC product has not really been rigorously defined as yet, then the identified client security requirements can actually be used to define the system requirements for an RBAC product. Once these requirements have been defined, the remainder of an RBAC product life cycle process can be properly activated.

When creating a concept design or architecture and implementing an RBAC product there will be a trade-off consideration to make. The trade-off is RBAC capability (or complexity) versus simplicity. The trade-off affects the designers or architects, implementers, administrators, and users. The simpler an RBAC product, the easier it will be to develop, maintain, administer, and use. The more capable the product, the more options administrators and users will have and the more complex the product will be. The fewer capabilities, the less options will be available and the simpler the product will be.

By making the RBAC transactions transparent to the users, users will find that they cannot do things that they might try to do. They also may not really understand the reason. A well-designed RBAC product should optionally list the privileges that go with a particular role to which the user is assigned. This will enable users to be knowledgeable of what they are allowed (or not allowed) to do.

The analysis of the information provided in our filled-in questionnaire should reveal at least the following items: 1) information about the interviewee's system environment, and 2) information regarding the interviewee's needs or desires for RBAC security capabilities.

An analysis can have up to three levels of information as discussed below. Analyses may be either qualitative or quantitative. A *qualitative analysis* ascertains the nature of the attributes, behavior, or opinions of the entity being measured. In describing a person, a qualitative analysis might conclude that the person is tall, thin, and middle-aged. A *quantitative analysis* ascertains the magnitude, amount, or size of the attributes, behavior, or opinions of the entity being measured. In describing a person, a quantitative analysis might conclude that the person is 5 feet 10 inches tall, weighs 185 pounds, and is 37 years old.

The three levels of analysis are:

)**First-Level Analysis** - Describe the data using frequency tabulation or frequency tables, e.g., count the number of "yes" answers from respondents.

)**Second-Level Analysis** - Analyze the data one question at a time. Certain statistics such as the mean or median can be obtained with the description of the data for questions where such statistics would be useful or appropriate.

)**Third-Level Analysis** - These analyses often address differences between subgroups. Two of the many analytic tools available to investigate more analytic questions are multiple regression analysis and discriminant function analysis.

The analysis presented in the report includes only the first two levels described above for both qualitative and quantitative data.

The various methods for access control are described in Appendix A. The initial phase of the marketing process requires the development of a questionnaire or structured interview, which is described in Appendix B. An approach to designing RBAC modifications to current legacy computer environments is presented in Appendix C. The user security general requirements, as determined by NIST, are presented in Appendix D. Some information about databases is presented in Appendix E. A brief overview of an RBAC product is presented in Appendix F, and definitions of some relevant terms are given in Appendix G. Relevant acronyms are given in Appendix H, and the C2 security level is described in Appendix I. The questionnaire used in this study is presented in Appendix J.

2.2 Questionnaire Process

A principal objective of the questionnaire development process was to formulate a set of questions that allowed us to determine the security requirements of the organizational representatives that we interviewed. Care was taken in constructing the questionnaire so that questions that might infer a flaw in the security of the organization's computer system were avoided.

2.2.1 Questionnaire Construction

The questionnaire for this survey contained questions relevant to the information needed to identify the user's needs for an RBAC system as discussed in Appendix B, *Structured Interviews,* on structured interviews. Questionnaires can be developed from two perspectives: 1) the objectives of the study (i.e., what is the study attempting to find out?) and 2) the analytical process to be used for evaluating the filled-in questionnaires (i.e., how will the data be analyzed?).

This double-perspective process is shown in Figure 2-2, *An Approach to Creating a Questionnaire*. By considering both perspectives, one can develop a questionnaire that not only appropriately determines the desired information but also is easy to analyze and evaluate. Because the process is iterative, the arrows are double-headed.

Figure 2-2. An Approach to Creating a Questionnaire

A process for developing and conducting a questionnaire is presented below and illustrated in Figure 2-3, *An Illustration of the RBAC Phase 2 (Task 2) Study Process*:

1. Identify the information (i.e., study objectives) that is required concerning both the interviewee and the relevant subject matter.

2. Identify the question which when answered will give data that will infer the desired information (i.e., achieve the stated objectives of the study).

3. Simplify the question so that it is easy to understand and easy to answer.

4. Repeat steps 1 through 3 until all the required information and concomitant questions have been identified.

5. Formulate the complete questionnaire.

6. Ensure that the questions are ordered in an appropriate manner and prepare the interview protocol (i.e., the questionnaire).

7. Formulate and prepare a tutorial that will assist in ensuring that the interviewee clearly understands the relevant environment and what is required.

8. Conduct pretests, which consist of the questionnaire, by giving them to a selected small set of sample interviewees in order to determine if the questionnaire is okay or needs to be modified. After the pretests, any required modifications to the questionnaire should be made and the pretests repeated unless the modifications are minor.

Figure 2-3. An Illustration of the RBAC Phase 2 (Task 2) Study Process

9. Prepare and implement a process for conducting the interview that assures that all the questions are asked in the proper order, that the responses are properly noted, and that the interviewee is not unduly inconvenienced.

10. Formulate and implement an evaluation methodology for analyzing and evaluating the filled-in questionnaire.

11. Formulate and implement a process for identifying the relevant interviewees, contacting the interviewees, conducting the interview protocol, and conducting the analysis and evaluation.

12. Using the study results or findings, make the appropriate conclusions, and determine the appropriate recommendations for further activities.

13. Using the study results, write a final report describing the findings and conclusions of the marketing survey.

This is the process we followed for performing this study. The study results, in the form of findings and conclusions, are included in the latter part of this section.

2.2.2 A Structured Interview Process

The process for collecting data relevant to the security needs of the targeted users was to use face-to-face interviews. The process used to collect this data is called a structured protocol or structured interview. A questionnaire is required to perform a structured interview and this process is described in Appendix B. The questionnaire or data collection instrument is described in Appendix J. Once an acceptable questionnaire has been developed, the interviewees need to be identified.

2.2.2.1 **Selection of Interviewees**. There are just too many commercial clients to interview even a small fraction. Also, the commercial users are distributed all over the country. The same is generally true for military users. That leaves civil government, which is the category we chose. We limited that group to federal civil government organizations in the Washington, D. C., area. Thus, our approach was as follows:

- An original list of interviewees was created based on contracts that existed with these interviewees and SETA (this led to about six interviews).

- Only interviewees in the Washington, D. C., area were selected.

- The additional interviewees (24 of them) were contacted using the Federal Computer Security Program Managers' Forum Membership list supplied to us by NIST.

The quality and knowledge of the security managers which we interviewed were quite high. This greatly facilitated the process of filling in the questionnaires. The method for selecting interviewees was based on "judgment sampling" [GAO Oct. 1993]. Judgment sampling is based on the judgment of the evaluator, namely SETA. In our judgment, there is no bias in the particular selection of client agencies that SETA happens to have or the representatives from organizations in the NIST list.

In the initial stages, when two people from the same organization were willing to be interviewed, we interviewed them separately, which proved to be highly redundant. Later when more than one person from a single organization wanted to interview, we interviewed them at the same time.

2.2.2.2 **Validation and Verification.** *Validation* is an effort to ensure that the questionnaire is actually measuring the variables it was designed to measure. A consensual approach to creating the questionnaire was taken.

The review group consisted of the following SETA employees: Ravi Sandhu, Fred Holland, Ed Coyne, Chuck Youman, Srinivas Ganta, and Charlie Smith. The questionnaire was reviewed and comments were received from each member of the group and then later implemented or rejected after carefully considering the credibility and applicability of the comment.

Verification is a way of checking or testing questionnaire answers to reduce the risk of using data that are inaccurate. Verification was achieved using two approaches:

- We asked some of the questions more than once with different wording for the redundant questions.

- We conducted interviews with two test subjects who provided us with early feedback on the accuracy of the questionnaire.

Verification was also performed whenever information was available that had been corroborated elsewhere.

2.2.2.3 Interview Procedure. A proper procedure for implementing a questionnaire as a face-to-face protocol process is to:

- Ask the questions exactly as they are worded in the questionnaire
- Ask the questions in the order in which they appear
- Ask every question in the questionnaire
- Read each question slowly
- Repeat questions that are misunderstood
- Accurately record the interviewee's responses
- Do not let the respondent stray from the questions in the interview
- Keep nonverbal cues as neutral as possible

In addition, we allowed each interviewee to make comments that were relevant to the security requirements. These comments were recorded in the margins of the form that we used to record the interviewee's questionnaire responses.

2.2.3 Analysis Plan

The objective of the questionnaire is to gather information that can be used to characterize both the various organizations that we interviewed and their security requirements. The characterizations are presented in tabular form. The analysis process becomes one of selecting the information off the filled-in questionnaires and filling in the analysis tables. The analysis tables are shown in Section 2.3, *Study Findings on User Security Requirements*.

It is also of interest how the requirements statements from different interviewee-organizations vary. That is, do the statements from each interviewee reveal a similar set of requirements or do they significantly disagree? This is important because if client requirements are roughly similar, then it will be easier to create a single RBAC product to satisfy their needs than if they are not similar.

The roles and role constraints data were quantified using two methods. Each role or role constraint was given a "yes" or "no" by the interviewee as well as a weight. Based on the Yes/No answers, the roles or constraints were evaluated using the following equation:

$$\text{value}_1 \text{ (role k)} = (1/N) \Sigma j \text{ V-role}_j \text{ (k) so that } 0 \leq \text{value}_1 \text{ (role k)} \leq 1.0,$$

where j = 1, 2, ..., number of interviewees = N; k = 1, 2, ..., number of roles in the table; and where V-role$_j$ (k) is the value (yes = 1 and no = 0) for the kth role assigned by the jth interviewee. The numerical value, value$_1$ (role k), is the proportion of users who stated "yes" to the need for the kth role.

The method for rank ordering the roles and constraints is to use the weights that each interviewee gave to the roles that they gave a "Yes, I need it" answer. The mathematics for this rank ordering is as follows:

$$\text{value}_2 \text{ (role k)} = (1/N) \Sigma j \text{ weightj (k) V-role}_j \text{ (k) so that } 0 \leq \text{value}_2 \text{ (role k)} \leq 5.0,$$

where j = 1, 2, ..., number of interviewees = N; k = 1, 2, ..., number of roles in the table; V-role$_j$ (k) is the value (Yes = 1 and No = 0) for the kth role assigned by the jth interviewee; and weightj (k) is the *strength of preference* for the kth role assigned by the jth interviewee (a number such that 1 ≤ weight ≤ 5). The rank ordering for the roles in this case is based on the numerical values of value$_2$ (role k) which reflect the strength of each interviewee's feeling towards their need for the kth role.

In addition to the tabulated data, there were many statements made by the interviewees. We recorded many of these and carefully reviewed them before deciding on which ones to include in this report. We excluded any statements that were clearly of a personal nature.

2.3 Study Findings on User Security Requirements

The data gathered from the filled-in questionnaires are presented in the various tables in the following sections. From the tabulated data, some findings have been selected. These findings are included in the sections with the tabulated data.

2.3.1 Federal Organizations Interviewed

Representatives from 27 Federal organizations were interviewed. Two organizations received more than one interview (viz., the Agency for International Development had two and the Department of Justice/Information Management Security System had three). The Federal organizations interviewed are listed in Table 2-1, *Federal Organizations Interviewed.*

Table 2-1. Federal Organizations Interviewed

1. Internal Revenue Service 15. U. S. House of Representatives
2. Walter Reed Army Institute of Research 16. Securities and Exchange Commission
3. Department of Justice/Consolidated Assets Tracking System 17. Department of Housing and Urban Development Agency
4. Department of Agriculture/Food and Consumer Service 18. Department of Treasury

5. Defense Information Systems Agency 19. Bureau of Alcohol, Tobacco and Firearms
6. U. S. Agency for International Development (2) 20. Federal Emergency Management Agency
7. Department of Justice/Immigration and Naturalization Service 21. Department of Energy
8. Department of Transportation/U. S. Coast Guard 22. Department of Transportation/
Federal Highway Administration
9. Library of Congress 23. Department of the Interior
10. National Science Foundation 24. Social Security Administration
11. Department of Justice/Information Management Security System (3) 25. General Services Administration
12. U. S. Senate 26. Veterans Health Administration
13. Department of Justice/Federal Bureau of Investigation 27. Department of Education
14. National Institutes of Health/National Cancer Institute

2.3.2 Statistics on Interview Duration

We interviewed 30 security managers representing 27 federal government organizations which was roughly 50% of the 57 relevant organizations based on the Federal Government phone book listings of Federal organizations with large computer systems. For the conducted interviews, the average duration of an interview was 47 minutes with a maximum interview time of 105 minutes and a minimum interview time of 25 minutes.

2.3.3 Application Environment

The application environment is the domain of the computer system. For these inputs, there were 27 since the inputs for two or more users from a single organization were clumped together. The gathered data are shown in the following tables for application area and include types of system and data used (Table 2-2, *Types of System and Data Used*); security principles for current and desired (Table 2-3, *Security Principles (Current and Desired)*); types of applications (Tables 2-4, *Desired Security Principles*, and 2-5, *Types of Applications*); and number of applications generated by the organization (Table 2-6, *Number of Applications Generated*).

Table 2-2. Types of System and Data Used

Types of System Used

Mainframe/Mini/LAN	Mini/LAN	Mainframe/LAN
20 (74%)	3 (11%)	4 (15%)

Types of Data Used

All	UBS + Unclassified	UBS	Unclassified	UBS + Classified
9 (33%)	9 (33%)	4 (15%)	1 (4%)	4 (15%)

UBS = unclassified but sensitive, All = unclassified + UBS + classified

The findings from this table include the following items:

- Most Federal organizations interviewed have computer systems that are composed of mainframes, minicomputers, and local area networks (74%).

- All Federal organizations except one that we interviewed have data that include unclassified-but-sensitive information (96%).

We offer a word of caution on reading from these tables. For example, from Table 2-2, one cannot infer that 15% of all data is unclassified-but-sensitive and classified, only that 15% of the users (federal government organizations) use only unclassified-but-sensitive and classified data.

For Table 2-3, the findings include the following items:

- Over two-thirds (70%) of Federal organizations interviewed plan to keep their current security principles, 27% want a change, and 4% have no plans for future security principles.

- Most Federal organizations interviewed (93%) plan to include the least privilege security principle among its security principles for their future computer system.

The last row in Table 2-3 was placed in a separate table to facilitate reading the results and is shown in Table 2-4.

Table 2-3. Security Principles (Current and Desired)

Desire No Change 19 70%

All - All
LP - LP LP/SD - LP/SD
SD/AP - SD/AP
11 (41%) 3 (11%) 4 (15%) 1 (4%)

Desire Change or Have No Plans 8 30%

LP/SD - All
All - LP/SD All - No Plans
SD - LP/AP
SD - LP
4 (15%) 1 (4%) 1 (4%) 1 (4%) 1 (4%)

Desired Security Principles

All	LP/SD	LP	SD/AP or LP/AP	No Plans
15 (56%)	5 (19%)	4 (15%)	2 (8%)	1 (4%)

LP = least privilege; SD = separation of duties; AP = abstract permissions; All = all principles

Table 2-4. Desired Security Principles

Principle(s)	**Number of Organizations**	**Percentage**
Least Privilege-Separation of Duties-Abstract Privileges	15	56%
Least Privilege Only	4	15%
Least Privilege-Separation of Duties	5	19%
Least Privilege-Abstract Privileges	1	4%
Sep. Duties-Abstract Privileges	1	4%
No Plans	1	4%

In some cases, the percentages are rounded and may not sum to exactly 100%.

The finding for types of applications shown in Table 2-5 is listed as follows:

- Nearly all Federal organizations interviewed (96%) have unique applications and all (100%) have common and commercial off-the-shelf (COTS) applications.

The findings for number of applications generated shown in Table 2-6 are listed below:

- About one-fourth of Federal organizations interviewed (26%) have over 1,000 applications generated for their computer system.

- Nearly one-half of Federal organizations interviewed (45%) have over 100 applications generated for their computer system.

Table 2-5. Types of Applications

Type	**Yes**	**No**
Unique Applications	26 (96%)	1 (4%)
Common Applications	27	0
Commercial off-the-shelf	27	0

Table 2-6. Number of Applications Generated

Number Applications	**Number Organizations**
1s	1 (4%)

10s	11 (41%)
100s	5 (19%)
>1000	7 (26%)
Don't Know	2 (7%)
Uses contractors	1 (4%)

2.3.4 User Characteristics

The gathered data for user characteristics are shown in Table 2-7, *User Characteristics*. The findings for user characteristics include the following items:

- Nearly all Federal organizations interviewed (92%) have over 1,000 users.

- About one-half of all Federal organizations interviewed (48%) change job assignments frequently.

- About one-fourth of all Federal organizations interviewed (26%) change job definitions frequently.

- Over one-third of all Federal organizations interviewed (37%) have a high turnover rate (more than 10% per year).

Table 2-7. User Characteristics

Number of Users	**>1,000**	**100-1,000**	**50-100**
Users in each category	25 (92%)	1 (4%)	1 (4%)

Organizational Environment	**Number**	**Percentage**
Changes job assignments frequently	13	48%
Changes job definitions frequently	7	26%
Has high turnover rate (>10% per year)	10	37%

2.3.5 Data and Application Characteristics

The gathered data for database information are shown in Table 2-8, *Database Information*. A database entity is any organizational parameter for which there exists a database. The findings from this table include the following items:

- Over one-half of all Federal organizations interviewed (55%) have at least thousands of different databases in their computer system.

- Over four-fifths of all Federal organizations interviewed (85%) have at least hundreds of different databases in their computer system.

- Over four-fifths of all Federal organizations interviewed (81%) have databases which are subdivided for user access for security reasons.

Table 2-8. Database Information

No. DB Entities	10,000s	1000s	100s	Don't Know
No. Organizations	9 (33%)	6 (22%)	8 (30%)	4 (15%)

Are any databases subdivided for user access? 22 (81%)

Data regarding security administrators are shown in Table 2-9, *Number of Security Administrators*, and the findings from this table include:

- Just under one-half of all Federal organizations interviewed (44%) have hundreds of different security administrators for their computer system.

- Nearly three-fourths of all Federal organizations interviewed (74%) have at least 31 security administrators.

Table 2-9. Number of Security Administrators

Number of Organizations		**Number of Administrators** **Percentage of Organizations**
1 - 9	3	11%
10 - 19	3	11%
20 - 30	1	4%
31 - 100	7	26%
100s	12	44%
1000s	1	4%

Full time equivalents run about 25% on average with a minimum value of 10% and a maximum value of 100%.

Security and system descriptions are shown in Table 2-10, *Security and System Description.* The findings from this table include the following three items:

- Over two-thirds of all Federal organizations interviewed (70%) perform their security at both the applications and database levels.

- Nearly all Federal organizations interviewed (96%) have distributed systems.

- Over four-fifths of all Federal organizations interviewed (81%) require separate logins.

Table 2-10. Security and System Description

	Where Security is Performed	
System Description	**Separate Logins Required?**	
	Applications and Data	
	19 (70%)	Distributed
	25 (92%)	Yes
22 (81%)		

Applications Only
8 (30%) Centralized
1 (4%) No
3 (11%)
Data Only
0 (0%) Distributed/Centralized
1 (4%) Yes and No or Don't Know
2 (7%)

2.3.6 Organizational Characteristics

The data for the organizational characteristics are shown in Table 2-11, *Organizational Characteristics.* The findings from this table include the following three items:

- Nearly two-thirds of all Federal organizations interviewed (63%) have organizational structures that are similar to their computer system access structure.

- Two-thirds of all Federal organizations interviewed (67%) have a stable organizational structure.

- Three-fourths of all Federal organizations interviewed (74%) do not have a unified security process for their computer systems.

A system that is *not unified* means that a user who needs access to two or more databases, each of which is controlled by a different security administrator, must get the permission from each database administrator for access to each database under that administrator.

Table 2-11. Organizational Characteristics

Issue	**Yes**	**No**	**Yes/No**	**Don't Know**
Organizational Structure = Computer Access Structure?	17 (63%)	7 (26%)	2 (11%)	1 (4%)
Organizational Structure Stable?	18 (67%)	8 (30%)	1 (04%)	
Organization Owns Data/Apps?	11 (41%)	13 (48%)	3 (11%)	
Organization Controls Data/Apps?	14 (52%)	11 (41%)	2 (07%)	
Is Security Unified?	7 (26%)	17 (63%)	3 (11%)	

The data for user accountability are shown in Table 2-12, *User Accountability (Now or Desired) and Periodic Assessment.* The accountability data are for both current systems and for a desired capability in the future. The findings for this table include the following items:

- Nearly all Federal organizations interviewed (93%) have security audits that are presently, or will eventually be, capable of tracing system activities back to the user.

- Over one-half of all Federal organizations interviewed (59%) have security audits that include identifying the security administrator.

- Over two-thirds of all Federal organizations interviewed (70%) have periodic security access assessments.

Table 2-12. User Accountability (Now or Desired) and Periodic Assessment

			All	
			User/Group	
			User/Admin	
User-Only	**Admin-Only**			
7 (26%)	3 (11%)	7 (26%)	8 (30%)	2 (7%)
Periodic assessment of security access?			Yes - 19 (70%)	No - 8 (30%)

The data for either current or desired RBAC or RBAC-like capabilities for performing security control are shown in Table 2-13, *Current or Planned RBAC (or RBAC-Like) Capability*. The findings from this table include the following two items:

- Less than one-half of all Federal organizations interviewed (44%) currently have an RBAC or RBAC-like product as a complementary capability to their security system.

- Two-thirds of all Federal organizations interviewed (66%) have or plan to have an RBAC or RBAC-like product complement to their security system.

Table 2-13. Current or Planned RBAC (or RBAC-Like) Capability

Currently or planning role-based (or RBAC-Like) access capability

	Currently Using
12 (44%)	Plan to Use
6 (22%)	No Plans
8 (30%)	Don't Know
1 (4%)	

Even though many of the interviewees stated that they had no plans for implementing RBAC in the future, that does not mean that they understand what RBAC is and have decided not to use it. In many cases, it meant that they did not really understand what RBAC is but if they knew more about it, then they might wish to implement it. These data are also shown in the pie chart in Figure 2-4, *RBAC Capabilities or Desires*.

Figure 2-4. RBAC Capabilities or Desires

2.3.7 Desired RBAC Security Product

For this part of the questionnaire, the interviewees were asked what they desired to have if they could define an RBAC product of their choice. The potential role and constraint selections and definitions are shown in Table 2-14, *Descriptions of Roles and Role Constraints*. The interviewee responses are shown in Table 2-15, *Desired RBAC Security Product*. For these inputs, all 30 interviews were used since each interviewee had an opinion of what they would like to have in an RBAC product. For interviews that included more than one interviewee, the selections made by the group were used.

Table 2-14. Descriptions of Roles and Role Constraints

Role, Hierarchy, or Role Constraint

Description

Application- Independent Role   Allows RBAC controls to be inserted into a legacy system. This is for users who wish to retain much of their current system by implementing an application-level RBAC capability.

Basic Role Allows role designers, whomever they are, to define roles. Usually the most knowledgeable people about roles, that is, job functions, are managers.

Duty Role Allows a user to perform a specific duty. For example, the duty might be certain element of a larger task. When the duty is complete, the role goes away.

Null Role Allows a user to be logged on to the system but not have any roles that are active. The user will have access to the system for some off-the-shelf application but cannot perform any role-based activities.

Delegate-Role Single Level Allows a first-user to authorize a second-user to enter a transaction the first-user is authorized to use.

Delegate-Role Multiple Levels Same as Delegate-Role Single Level, but a second-level user can repeat the process.

Single-Role Hierarchy Allows a role designer to define a related set of roles within an application. A role may inherit the capability to enter a transaction from a previous role in the hierarchy.

Multiple-Role Hierarchies Allows a role designer to define more than one role hierarchy, as described above, within an application.

Collaborative Roles
(Constraint) Allows role designers to establish a role where multiple users must be active in the role (or a related role) to enter a transaction. That is, multiple users must collaborate to perform a transaction.

Concurrent Roles
(Constraint) Allows a user to have more than one role active at the same time. This usually would not be true for all roles, just for some roles.

Prerequisite Roles
(Constraint) Allows a role designer to specify a role that must be active before the user can assume a second role. This means that for some roles, a user can only enter into it from an acceptable lower level role, thus forming a hierarchy of role access.

Exclusive
[one user] (Constraint) Allows a role designer to specify that no other user can have this active role while a particular user assumes this role. For example, during database updating an administrator would have this role so that no user can access the database during this time.

Exclusive
[one role] (Constraint) Allows a role designer to specify that a specific user can have no other active role while this role is active. This constraint can limit the possibilities for users doing damage to the system.

Mutually Exclusive (Constraint) Allows a role designer to use exclusive roles that are mutually exclusive. This gives the designer the capability to implement "separation of duties."

Mutually Exclusive Transaction Basis

(Constraint) Allows a role designer to use mutually exclusive roles that are applied on a per transaction basis. This gives the designer the capability to implement "separation of duties" on a transaction basis.

Although Table 2-15, *Desired RBAC Security Product*, does reveal the roles and role constraints as desired by the interviewees, to make the data easier to interpret, the data were divided into two tables, desired roles (Table 2-16, *Desired RBAC Roles (Rank Ordered)*) and desired role constraints (Table 2-17, *Desired RBAC Role Constraints (Rank Ordered)*). The data were also rank ordered so that the reader can easily see which roles and role constraints are preferred by the interviewees.

The data were rank ordered on the basis of *strength of preference*. Strength of preference is based on a numerical value of 1 to 5 (where 1 means very weak and 5 means very strong). Whenever the interviewee stated "yes" to selecting the role or role constraint, a strength of preference was requested. If the interviewee said "no," then a value of 0 was given for that item and this value is also factored in when the strength of preference is computed.

Table 2-15. Desired RBAC Security Product

Role	**Yes**	**No**	**% Yes**	**Strength of Preference**
Application-Independent Role	22	8	73%	3.07
Basic Role	30	0	100%	4.53
Duty Role	27	3	90%	3.78
Null Role	16	14	53%	2.20
Delegate-Role, Single Level	19	11	63%	2.20
Delegate-Role, Multiple Levels	8	22	27%	0.87
Single-Role Hierarchy	27	3	90%	3.47
Multiple-Role Hierarchies	25	5	83%	3.23
Collaborative Roles (Constraint)	19	11	63%	2.47
Concurrent Roles (Constraint)	30	0	100%	3.92
Prerequisite Roles (Constraint)	21	9	70%	2.53
Exclusive [one user] (Constraint)	26	4	87%	3.68
Exclusive [one role] (Constraint)	27	3	90%	3.60
Mutually Exclusive (Constraint)	27	3	90%	4.07
Mutually Exclusive, Transaction Basis (Constraint)	24	6	80%	3.30

From Tables 2-16 and 2-17, the following findings are true when all the interviewees are considered:

- For the RBAC roles, only three have a strength of preference less than 2.50, namely, the Null Role and the Delegate-Role for both the Single and Multiple Levels.

- For the RBAC role constraints, all of the constraints have reasonably high values (2.47 or greater) for strength of preference.

- For the RBAC role constraints, all but two of the constraints have very high values (3.30 or greater) for strength of preference, the exceptions being Prerequisite and Collaborative Roles.

2.3.7.1 Desired RBAC Roles (Rank Ordered). The desired RBAC roles are shown in Table 2-16 where they are rank ordered according to the *strength of preference*.

Table 2-16. Desired RBAC Roles (Rank Ordered)

Role	**Yes**	**No**	**% Yes**	**Strength of Preference**
Basic Role	30	0	100%	4.53
Duty Role	27	3	90%	3.78
Single-Role Hierarchy	27	3	90%	3.47
Multiple-Role Hierarchies	25	5	83%	3.23
Application-Independent Role	22	8	73%	3.07
Delegate-Role, Single Level	19	11	63%	2.20
Null Role	16	14	53%	2.20
Delegate-Role, Multiple Levels	8	22	27%	0.87

2.3.7.2 Desired RBAC Role Constraints (Rank Ordered). The desired RBAC role constraints are shown in Table 2-17 where they are rank ordered according to the *strength of preference*.

Table 2-17. Desired RBAC Role Constraints (Rank Ordered)

Role Constraint	**Yes**	**No**	**% Yes**	**Strength of Preference**
Mutually Exclusive	27	3	90%	4.07
Concurrent Roles	30	0	100%	3.92
Exclusive [one user]	26	4	87%	3.68
Exclusive [one role]	27	3	90%	3.60
Mutually Exclusive, Transaction Basis	24	6	80%	3.30
Prerequisite Roles	21	9	70%	2.53

Collaborative Roles	19	11	63%	2.47

2.3.7.3 Statistics for Clients Using Some Form of RBAC at Present. There were 12 clients (44%) who were already using some form of the RBAC product. The findings for these users are presented below.

- Three of the desired roles had a strength of preference that was less than 3.00, namely, the Null Role and the Delegate-Role for Single and Multiple Levels.

- For the RBAC role constraints, all of the constraints have reasonably high values (2.25 or greater) for strength of preference.

- For the RBAC role constraints, all but two of the constraints have very high values (3.50 or greater) for strength of preference, the exceptions being Prerequisite and Collaborative Roles.

All of these findings for the clients currently using RBAC or some form of it, are the same as for the case of all clients interviewed. The desired RBAC roles for clients now using some form of RBAC are shown in Table 2-18, *Desired Roles for Clients Now Using RBAC*, where they are rank ordered according to the *strength of preference*. The desired RBAC role constraints for clients now using some form of RBAC are shown in Table 2-19, *Desired Role Constraints for Clients Now Using RBAC*, where they are rank ordered according to the *strength of preference*.

Table 2-18. Desired Roles for Clients Now Using RBAC

Role	**Yes**	**No**	**% Yes**	**Strength of Preference**
Basic Role	12	0	100	4.33
Duty Role	12	0	100	4.21
Application-Independent Role	10	2	83	3.75
Single-Role Hierarchy	11	1	92	3.58
Multiple-Role Hierarchies	10	2	83	3.33
Null Role	6	6	50	2.42
Delegate-Role, Single Level	7	5	58	2.00
Delegate-Role, Multiple Levels	3	9	25	0.75

Table 2-19. Desired Role Constraints for Clients Now Using RBAC

Role Constraint	**Yes**	**No**	**% Yes**	**Strength of Preference**
Mutually Exclusive	11	1	92	4.08
Exclusive [one user]	10	2	83	3.83
Concurrent Roles	12	0	100	3.79
Mutually Exclusive, Trans. Basis	10	2	83	3.58
Exclusive [one role]	11	1	92	3.50
Collaborative Roles	8	4	67	2.92
Prerequisite Roles	8	4	67	2.25

2.4 Additional Statements from Interviewees

In addition to eliciting the answers to the questions stated in the questionnaire, many of the interviewees made relevant statements which we recorded and are presented in Table 2-20, *Interviewee Comments on Organizational Requirements*. Whatever statements we have recorded are merely repeated here as interviewee comments. In no case is any statement attributed to a specific person or organization. Our objective is to list the recorded comments that are relevant to the determination of the client needs so as to accurately define an appropriate RBAC security product, particularly for federal civil-government organizations.

The statements in Table 2-20 are randomly ordered and some of the interviewees made no comments at all outside of their responses to the questions in the questionnaire. A summary of the more important statements from Table 2-20 is presented in Table 2-21, *Summary of Interviewee Comments*.

Table 2-20. Interviewee Comments on Organizational Requirements

Comments
Role engineering will be very important in the future. Goal is to migrate to a unified access capability (one request or one role). Believes interoperability is important and desires it among different platforms (3-tier architecture). Wants to have an expert system to assist in automating definitions of roles (automated role-engineering and security administration).
ORACLE doesn't satisfy all of the organization's needs. Improving the profile process, at present, is limited to processing of customer inputs but should include security administration. Believes that RBAC could alleviate unified logon by enabling the pooling of resources to establish and administer the roles.
Would like for SETA to do a check on the organization's RBAC process. Much of legacy software will be retained and the organization has defined all the roles. Wants unified logon. Don't make users ask for many roles to gain access to desired databases. What does it take to identify a fraudulent act? An organizational group is doing role engineering.
Permissions may change with every logon. Wants to have permissions adjusted on the fly. May wish to have read-only permissions. Needs to have someone authenticate any changes in roles or permissions. The organization doesn't like users inputting viruses. (This must be happening or it would not have been stated.) Many legacy systems are emulators of emulators of .... etc. and cannot have a security system added on top of them. Believes that some legacy systems cannot be retrofitted for security.
Administation should be performed using a "domain" scheme where roles are controlled for multiple applications. Needs centralized databases administered by distributed access.

This organization is role oriented. Access is based on the job function and is computer automated. Has identified 6400 different access positions.

Table 2-20. Interviewee Comments on Organizational Requirements (Continued)

Comments

ORACLE doesn't satisfy the organization's needs, is not user-friendly, and the organization is designing its own RBAC security system. Each application has its own set of role definitions. Looking for dynamic role capability based on user functions. ORACLE is pushing their RBAC definition for SQL3. Sybase is implementing a more robust RBAC capability. A major effort to reinvent the agency through business process engineering is ongoing.

The LANs are decentralized and may be organized differently from one another. Role definitions are application dependent. Wants a different operating system (OS) that can talk to all OSs on network (this is a big-time problem). Would like a central point of control regarding access privileges. The organization has had many problems attempting to implement RBAC concepts.

Currently using a crude version of RBAC. Wants a security system that is a better way to implement MAC and DAC. Can RBAC be used to detect intrusion and notify a security officer that something weird is going on? Wants this at the C2 level. This capability is not in current security software system with B3 or A1 level security. Actions or capabilities required are:

1) Needs decisions on roles as the organization changes so RBAC must be dynamically reworked.
2) Needs auditing of roles and utilization of roles (if not used within some period then retract role).
3) Don't select roles off the screen, the role must be assigned by an administrator.

Has large flux every two years of users.

Has custom applications for control of access. Good audit trail is needed. User role versus application role should be different with different rules for each. Developer roles should be constrained but user roles should not be constrained. Look at the function rather than the individual. If a role is critical then auditing should be performed in more detail. Make role definitions more specific to the user. The organization has different organizational groups with different rules for each.

Hasn't found a satisfactory means for reacting to insider threat. RBAC is a good approach, likes it. Effort to define roles is nontrivial. Even a MAC environment is nontrivial (for assigning required clearance levels to data). Trying to introduce RBAC concept and using business process engineering approach to identifying user roles. Some applications have user groups. Security is tough to enforce except in a dictatorial way and has been quite unsuccessful. Good security should include explaining to the users why security is needed.

Has lots of missions. Missions are independent.

If there were RBAC products, then it is important to have an educational program to administer RBAC wisely. The more complex this becomes, the more likely a mistake can be made. RBAC architectures could be useful for users. How are roles contained in certificates (e.g., Kerberos) (i.e., personality or capability certificates)? Planners or architects should be informed for viewing RBAC in their planning context as a useful new security product. Be objective. Can RBAC offer a single sign-on capability?

Table 2-20. Interviewee Comments on Organizational Requirements (Concluded)

Comments
Needs a template for roles that can be used to create a role on the fly. Implementation issues cause the most problems. Assigning rights to roles rather than to persons is better.
Desires a mainframe platform for security. Is RBAC workable for minicomputers and LANs for PCs?
Every organizational member has a security administrator. There are constant changes to meet certain demands, e.g., new implementation of E-mail is causing problems. In paperless transactions, parts of a form are filled in, then later another part is filled in by someone else. An RBAC product should keep a good audit of the users activities.
System provides total legislation support, constituent requirements, issues and questions, inventory control, and personal needs. Likes idea of RBAC, it is appealing. Thinks that it is workable at this organization. Wants an RBAC product that is flexible and dynamic, provides quick response, and has a short-term focus.
RBAC should be centrally controlled and auditable. Single staff writes new user profile from a central location. Access should be tied to the job function. Currently being done now by this organization.
Security should be implemented at the operating system level. Security policies are expected to be software independent. ACF2 was much simpler than RACF. Security can become an overwhelming burden for those who must manage it. Wants a security system that allows for a quick setup of organizational systems to monitor a specific event. Can roles be defined for basic event or occurrence?
Wishes to learn more about RBAC. RBAC could be an asset to this organization. Process for administrators should be very simplified (i.e., automated). Thinks that information in transit is most vulnerable. Are roles more tied to information than the processes that developed the role?
The organization is very structured so roles are easily defined. Issues of accountability and user actions are associated with auditing capabilities. Current security system is not good enough. Wants a better auditing capability.
Access decisions are made by program people. Underestimated the complexity of implementing a new security system, (estimated 18 months but took 30 months).
Principle issue with security is the management of security, not security per se. Users need to understand the need for security. Security products are available and not so complicated. Security is not a big issue. Security may be complex but not too complex to acquire. Technology people should not be making decisions regarding identifying roles. Must get management to understand the importance of protecting the data.
Thinks that SETA covered RBAC well in the questionnaire. Where this person worked before, thinks that one of biggest problems there was that roles were defined but users did not stick to the roles. This caused confusion with respect to roles. Process falls apart when resources get scarce and adds to the confusion.

Table 2-20. Interviewee Comments on Organizational Requirements (Concluded)

Comments
Desires a template for roles that can be used to create a role on the fly. Implementation issues cause the most problems. Assigning rights to roles rather than to persons is better.

Table 2-21. Summary of Interviewee Comments

Simple, accurate, and complete auditing is very desirable.
Current systems with RBAC-like capabilities are not satisfactory.
Would like to retain legacy systems, in many cases.
Interoperability of security systems is important.
Selecting or changing roles on-the-fly is desirable, as long as an administrator is involved. Perhaps role templates can assist in this.
Role engineering for accurate identification of roles is very important. Ensure that user roles are adhered to.
Would like for centralized databases to be accessible from distributed security administrator decisions.
Some systems have a large flux of users, in and out.
Wants an RBAC capability at the server or client position in a LAN.
Wants a security product that can combat the insider threat as well as the outsider threat.
Wants an RBAC educational system to ensure that managers and users are knowledgeable about security.
Variable auditing for critical roles is desired.
Because of constant changes to a dynamic computer system, the RBAC security product must be extensible.
The security system should be capable of being set up in a short period to respond to specific events.

2.5 Summary of Findings

Some of the findings are summarized in the list that follows:

- There are many personnel changes (i.e., turnover rate is high).

- There are many job definition or job assignment changes.

- Security management needs encompass all security principles (least privilege, separation of duties, and abstract permissions).

- There are many users on the systems.

- Many applications need to be controlled.

- Many security administrators are involved in managing the computer systems.

Some additional findings include the following items:

- Security needs to manage access to both applications and databases.

- Security management needs to be unified.

- Many roles and role constraints are required.

- Many administrators are required to manage the security aspects of large distributed computer systems.

2.6 Findings versus RBAC Capabilities

The findings are compared with the potential RBAC capabilities of a new RBAC product and are shown in Table 2-22, *Study Findings Versus Potential RBAC Product Capabilities*. The basic question here is "Is there a match between the stated needs by security management representatives from Federal organizations interviewed and the potential capabilities of a RBAC security product?" We believe that there is an emphatic "yes!" answer to this important question.

Table 2-22. Study Findings Versus Potential RBAC Product Capabilities

Comments on Findings and RBAC Capabilities
When there are lots of personnel changes (turnover rate is high), an RBAC product provides a simpler and easier method for handling administrative changes, that is, it provides a simple and straightforward method for handling large changes in users.
When there are many job definition or job assignment changes, an RBAC product provides a simple method for handling these changes.
When there are needs for all security principles (least privilege, separation of duties, and abstract permissions), then an RBAC product can easily provide these.
When there are many users on a system, an RBAC product can easily handle these.
When there are many applications that need to be controlled, an RBAC product provides a simple means for handling access to these applications
When there are many security administrators involved, RBAC tools can diminish the time required for managing the security aspects of a computer system.
When security is needed to manage access to both applications and databases, an RBAC product can provide for many different accesses, even partial database access.
When security management needs to be unified, an RBAC product can provide for unification of database or applications access through a linking of administrators.
When there are many roles and role constraints required, an RBAC product can be designed to easily provide these.
Administrators can delegate the security administrative aspects to other administrators as a way of distributing the workload among administrators.

3.0 CONCLUSIONS AND RECOMMENDATIONS

The conclusions and recommendations for this study are presented in this section.

3.1 Study Conclusions

Based on the summarized findings listed in Sections 2.5 and 2.6, we conclude that there is a need in the federal government community for an RBAC product.

3.2 Recommendations for Marketing RBAC

Some alternative proposals for marketing an RBAC product are the following:

- Embed an RBAC product in the existing client software as a unique capability,

- Embed the RBAC product in a large existing popular software package such as Windows NT 3.5, ORACLE 7.x, or NetWare 4.x, or

- Embed the RBAC product in an applications-level software vehicle such as VEIL [TECSEC 1996] or a firewall system such as Sidewinder [Secure Computing Corp. 1996].

3.2.1 Unique RBAC Software

The first alternative means that a specific package must be developed to meet the particular needs of each user. This is not a very economical alternative. However, some Federal organizations are already taking this approach on their own.

The latter two alternatives offer an RBAC product as part of an existing COTS package. Since all the federal organizations we interviewed have COTS packages, this should present no problem. These two alternatives are much more promising. However, we are not sure which is the better. At the present time SETA has not decided what the proper path is for marketing the RBAC product.

3.2.2 Three Popular COTS Packages

There were three COTS packages examined for this study, Windows NT 3.5, ORACLE 7, and NetWare 4.1.

3.2.2.1 Windows NT 3.5. Windows NT 3.5 does support a kind of RBAC-like capability. Windows NT doesn't directly address the concept of roles but it has a concept of groups, user profiles, and domains which directly or indirectly relate to the notion of roles, namely [Microsoft 1994]:

- A *group* is a collection of resource permissions and rights which can be assigned to multiple users.

- A *user profile* is a file that contains information about a user's desktop operating environment.

- A *domain* is the basic unit of security and centralized administration.

Groups provide a way to manage access to resources for users who use Windows NT to perform similar tasks. Memberships in the group can be determined by job assignment, specific access requirements, or any other criteria. The notion of a group may be mapped to a role, since a group is a set of permissions (and rights) assigned to multiple users [Cowart 1995]. However, the notion of a role is much more general than the notion of a group.

The Windows NT Server provides for built-in groups designed to accomplish administration responsibilities. Additional groups can be created when needed. Windows NT supports three types of accounts [Minasi *et al*. 1995]:

- **User Accounts**. Each user of the system has a user account. User accounts may be defined at a local machine level or on a domain. Accounts defined on a local machine may only be utilized on that machine and accounts defined on a domain level may be used on any machine in that domain. An account can be a member of more than one group.

- **Local groups**. Local groups are defined on each machine. Local groups may have both user accounts and global groups as members. Windows NT supplies a number of built-in local group accounts. Some of them are Administrators, Power Users, Users, Guests, Everyone, Backup Operators, Server Operators, and Print Operators groups. Local groups other than built-in groups can be created by the administrator.

- **Global Groups**. Global groups are defined at the domain level. Users in a global group are assigned rights or permissions by including the global group in a local group that has the desired permissions. Windows NT supplies two built-in global groups: Domain Admins and Domain Users.

User profiles can be created to set up secure environments for a variety of jobs. The administrator's use of profiles can lead to simplified administration and enhanced security as a single user profile can be set for users who perform similar tasks. For example, if Pat is the administrator of a system in the Engineering department, then she has created a user profile for the engineers in her department, stored in a shared directory, and assigned it to the engineer's user accounts. Pat receives a new forecasting application for the engineers. Instead of creating icons for each engineer for the new application, she will create the icon in the shared profile. Since the profile is shared, all of the engineer user accounts will be able to see the icon when they log on to the system. Administrators can assign users, local groups, and global groups to profiles.

A domain is a collection of computers that are grouped for common viewing purposes and that share a common user account for access to databases and for security policy. In a domain each user needs only one account.

Users that are not members of a specific domain are by definition excluded from all resources in that domain. In a multiple domain configuration, local groups are useful for assembling global groups into one manageable unit. Instead of assigning permissions to each global group separately, the administrator simply assigns permissions to a local group and then assigns global groups from other trusted domains to the local group. Through membership in a local group, users from other domains are allowed access to resources in the current domain.

A *trust* between two domains is established to honor the users of one domain in another. There are two components of a trust relationship: the *trusting domain* and the *trusted domain*. The trusting domain recognizes all users and global groups accounts from the trusted domain. The trusted domain therefore is a domain upon which an administrator or the system relies. These accounts can be placed in local groups of the trusting domain and given permissions and rights in the trusting domain. The trust between two domains allows a user to have an account in one domain and still be able to access resources of the entire network.

Since user profiles are files which are defined for a set of users who perform similar tasks, their functionality is similar to the role concept. The concepts of a group and a domain somewhat match to the concept of a role.

There are two types of permissions, rights and permissions. A *right* is an authorization for a user to perform certain actions on the system. Rights apply to the system as a whole. *Permissions* are rules associated with a particular object (for example, a directory, file, or printer). Objects are some of the system resources which include the system itself, files and directories, registry, printers, memory, ports, disks, applications, screens, sound cards, and processes.

Rights can override permissions on an object. For example, if user Chris is assigned the right to *Restore Files* and *Directories*, then she needs to be able to read all files on the system, including files for which the owners have set permissions that deny access to all users. The *Restore Files* and *Directories* right overrides permissions set by users and Chris will be able to perform her task.

Permissions are cumulative. However, *No Access* overrides all permissions. For example, the Engineering Group has *Full Control* and the Data Entry Group has *Print Access*. User Dean is a member of both groups and he will therefore have *Full Control*. However, if the Data Entry Group had *No Access*, Dean would have *No Access*. Some of the permissions for files and directories are *No Access, List, Read, Add, Add* & *Read, Change*, *Full Control, Execute, Delete, Change Permissions*, and *Take Ownership*. Permissions for printers include *No Access, Print, Manage Documents,* and *Full Control*.

Rights and permissions in Windows NT are allocated only to groups and not directly to individual users. Users who need any of these rights can be placed in the groups which have been assigned the rights. Windows NT allows many-to-many relationships between groups and users. It also allows for many-to-many relationships between groups and permissions.

The concept of a session equates to the traditional notion of a subject in the access control literature. In Windows NT, a token is created for each user and it represents all the information concerning them. A copy of the token is associated with each process that the user runs. This process/token combination is called a subject. This leads to the fact that every session (subject) corresponding to a user possesses the same set of permissions.

Role hierarchies cannot be directly supported in a Windows NT environment. If the environment is a single domain or a master domain one, local groups include global groups but not other local groups. Hence only two levels of nesting are possible. Also Windows NT does not directly support role constraints.

Windows NT provides some built-in groups to help in administration. Some of these groups are Administrators Group, Server Operators, Account Operators, Print Operators, etc. Also, additional groups can be created to help administer groups.

Other Issues

Access Review: Subjects operate on Windows NT objects by calling system services. Access validation routines determine whether a subject can access an object. The Security Reference Monitor (SRM) is the Windows NT Server component responsible for enforcing the access validation. It protects resources or objects from unauthorized access or modification. The SRM provides services for validating access to objects, and testing subjects for privileges.

The SRM contains the only copy of the access validation code in the system. This ensures that object protection is provided uniformly throughout Windows NT, regardless of the type of object accessed.

For example, when a user opens a file to edit, Windows NT first compares the security descriptor for the file with the security information that is stored in a user's token and a decision is made whether or not to allow the user to edit the file. The security descriptor for the file includes all of the access control entries (ACEs) that make up the file's access control list (ACL). A file without an ACL indicates that any user can access the file for any type of access. A file with an ACL indicates that the SRM must check each ACE in the ACL and determine if the user can access the file for the particular type of access.

Even though Windows NT is designed to be modular, SETA is not sure how extensible it is. Windows NT is discretionary access control oriented. This is due to the fact that any user creating an object has complete discretion on that object.

3.2.2.2 <u>ORACLE 7</u>. ORACLE 7 does support a kind of RBAC-like capability. In ORACLE 7, a *role* is a collection of related privileges that an administrator can grant collectively to database users. A user is not made a member of a role, but roles are granted to users. Some individuals might be maintained as multiple users in the system [Bobrowski 1995].

Permissions fall into two categories: system privileges and object privileges. A *system privilege* authorizes a user to perform a specific operation at the system level. An example of a system privilege is the *create user* privilege, which allows a user to create a database user-name. Another example is *insert table*, which allows a user to insert a table in the database. ORACLE 7 provides over 80 different system privileges. ORACLE 7 allows many-to-many relationships between groups and privileges.

Objects are primarily database objects like a database itself, tables, sequences, procedures, functions, packages, indexes, triggers, views, etc. An *object privilege* authorizes a user to perform a specific operation on a specific object. For example, a user can grant another user the ability to update a table by granting them the *update* privilege on that table. With this privilege, the user can update only that table, but cannot update any other table in the database. ORACLE 7 provides a varying number of object privileges per object type, such as: *insert, select, update, delete, alter, execute, index, references,* etc.

The use of roles in an ORACLE 7 client/server database system provides another important benefit, dynamic privilege management for application users. As a user moves from one application to another, each application's role ensures that the user has only privileges necessary to run the current application and does not carry over privileges from another application. ORACLE 7 allows role hierarchies to manage application privileges. Triggers can be used to provide for role constraints which can be enforced by stored procedures.

ORACLE 7 provides roles to help administer roles. Some examples of built-in roles are *administrator role* and a *resource role.* A resource role is intended for application developers.

<u>Other Issues</u>

Access Review: ORACLE 7 does have good access review. For any given role, it can determine what permissions are associated with that role and similarly for any given user, it can determine what permissions the user has. But we are not sure whether it can determine which roles the user belongs to (it can answer for sure the explicit roles assigned to a user) nor who are the users who are associated with a particular role.

Extensibility: ORACLE 7 does provide *stored procedures.* A stored procedure is a procedure written in the ORACLE 7 procedural language, PL/SQL. A stored procedure is a compiled collection of SQL statements, flow-of-control statements, variable declarations, assignment operators, and so on, that a developer creates and stores in a database. An example of a stored procedure is a *trigger*. A trigger is a stored procedure that ORACLE 7 automatically fires under the appropriate conditions.

DAC/non-DAC: ORACLE 7 is DAC-oriented. This is due to the fact that a typical user is not given *create* privileges, but if given such they have the complete discretion on the created object.

3.2.2.3 Novell NetWare 4.1. NetWare 4.1 does have an RBAC-like capability. Roles can be implemented using any NetWare Directory Services (NDS) object. There is a built-in class role, but it has no special significance as such (other than its name). NetWare has two types of objects: *NDS* and *file system objects.* NetWare directory objects represent abstractions such as users, roles, groups, and computers. File system objects provide a traditional hierarchical file system [Epstein and Sandhu 1996].

Permissions are the NetWare file rights. Examples of permissions are *supervisor, create, delete, rename, modify, file scan, add, erase, read, write, and access control.*

In NetWare, users can be mapped to an arbitrary number of objects (*i.e*., roles) and objects can be mapped to any number of users. Permissions can be assigned to any number of roles and, similarly, roles can be assigned to any number of permissions.

NetWare has no concept of sessions operating in different roles. Users obtain all rights that are assigned to them. Thus, there is no capability for dynamic activation and deactivation of roles during a session; a user must log out from one role and log onto a different NetWare account to change their role.

NetWare supports only tree hierarchies, but does not support general partial orders. Hence NetWare does not meet the proposed requirement of role hierarchies being partial orders.

It appears that NetWare cannot be used to implement role constraints directly or indirectly, but could be implemented administratively. NetWare's file access control policy can be used to provide roles with access to files and directories and, similarly, NDS access control policy can be used to provide roles with access to NDS objects.

Other Issues

Access Review: NetWare 4.1 does have a good access review. For any given object (i.e., role), it can answer what permissions are associated with that object and similarly for any given user, it can answer what permissions the user has. But we are not sure whether it can determine all the roles the user belongs to nor who are the users associated with a particular role.

Extensibility: NetWare provides NetWare Loadable Modules (NLMs), which extends the server operating system.

DAC/non-DAC: NetWare is fundamentally non-DAC-oriented as it has the ability to assign permissions independently, i.e., a user who is an owner of an object does not have any discretion on granting permissions on that object.

3.2.2.4 <u>Some Conclusions</u>. Some conclusions about the three COTS packages described above are summarized below.

- **Windows NT 3.5.** The advantages, disadvantages, and opportunities for Windows NT 3.5 are:

  Advantages: The biggest advantage with Windows NT 3.5 is that it supports decentralized administration. In a network consisting of multiple domains, the administrator can use trust relationships to centralize administration of user accounts and user-role relationships into one domain instead of administering these in each domain separately. This way each user only needs one account to access resources across multiple domains. Trusts divide user's accounts and resources into two separate areas of administration. Windows NT thus allows for separation of account management, i.e., user-role management versus role-permission management.

  Disadvantages: Windows NT does not directly support:

  - role-hierarchies
  - users having different sets of permissions for different roles
  - role constraints

  Opportunities: We are not sure if Windows NT 3.5 is extensible enough to support role-hierarchies, role constraints, sessions involving users with different set of permissions, and to make it non-DAC-oriented.

- **ORACLE 7**. The advantages, disadvantages, and opportunities for Oracle 7 are:

  Advantages: The main advantages with ORACLE 7 are that it allows role hierarchies and it allows users to have different sets of permissions for different roles.

  Disadvantages: It is DAC-oriented and requires stored procedures to enforce constraints.

Opportunities: We are not sure if the stored procedures can be used to make ORACLE 7 non-DAC-oriented nor whether the procedures can be used to enforce role constraints.

- **NetWare 4.1.** The advantages, disadvantages, and opportunities for NetWare 4.1 are:

  Advantages: The main advantage is that NetWare 4.1 is a non-DAC-oriented system.

  Disadvantages: NetWare does not directly support:

  - role hierarchies in the form of general partial orders
  - users having different sets of permissions for different roles
  - role constraints

  Opportunities: ORACLE 7 may be extensible enough to support role hierarchies, sessions, and constraints.

All the results are summarized in Table 3-1, *Summary of RBAC Capabilities of the Three COTS Packages*. From the summary, it appears that the ORACLE 7 database management system is the best of the three examined COTS options.

Table 3-1. Summary of RBAC Capabilities of the Three COTS Packages

Areas	**Windows NT 3.5**	**ORACLE 7**	**NetWare 4.1**
Sessions	No direct support (opportunity for improvement)	Direct support	No direct support (opportunity for improvement)
Role Hierarchies	No direct support (opportunity for improvement)	Direct support	No direct support (opportunity for improvement)
Role Administration	Direct support	Direct support	Direct support
Constraints	Does not support directly (opportunity for improvement)	Does not support directly (opportunity for improvement)	Does not support directly (opportunity for improvement)
Extensibility	Modular	Stored procedures	NetWare Loadable Modules
DAC/ non-DAC Options	DAC oriented (opportunity for improvement)	DAC oriented (opportunity for improvement)	Non-DAC oriented
Access Review	Good support	Good support	Good support

3.2.3 VEIL and Sidewinder

An existing security package, *VEIL* (Variable Encryption, Intelligent Labeling) from TECSEC, Inc., of Vienna, Virginia, is an object-oriented security package that resides at the application level (Layer 7 of the Open Systems Interconnection protocol) and is transparent to protocols and communications standards [TECSEC 1996]. VEIL determines who can unlock files according to the authority granted to the user by a security manager. The label reflects the readership groups that the user deals with on a day-to-day basis. A VEIL client can embed new encryption algorithms if they so choose including RBAC options provided the RBAC options have been added to the list of VEIL options. We believe that this approach could be used to add the RBAC capabilities to the VEIL software package.

VEIL provides confidentiality and data integrity. It is transparent to any network topology, protocol, or architecture. It provides audit trails and operates on the basis of user-selected algorithms. VEIL operates as a Windows stand-alone application with Word, WordPerfect, and Microsoft Mail.

In the near future, VEIL will operate with the Macintosh OS, UNIX, Windows NT, OS/2, and MS-DOS. VEIL is a very reasonably priced package at $129 for a single user with significant discounts for networked users.

Another existing security system, *Sidewinder* from Secure Computing Corporation of Roseville, Minnesota, is a firewall software system that uses a patented security technology to shield internal networks from outside intruders [Secure Computing Corp. 1996]. Sidewinder protection is based on *type enforcement* (a mandatory access control security product) which provides a domain structure that isolates applications, that is, it controls which files an application can access, and restricts which applications users can access. The types used by Sidewinder are the World Wide Web (WWW), File Transport Protocol, users, network, News, and telnet. A method for defining domain definitions that could be used for type enforcement is shown in Table 3-2, *Type Enforcement Domain Definitions*.

Table 3-2. Type Enforcement Domain Definitions

	File Types			
Process Domains	**FTP Files**	**Mailbox File**	**Mail spool**	**WWW Files**
FTP	read/write	no access	no access	no access
Mail System	no access	read/write	read/write	no access
WWW	no access	read	no access	read/write

The key secure principles provided by Sidewinder are:

- A simple, unifying security and integrity control product
- A modular security architecture

- Mandatory access control
- Least privilege design
- Audit of security critical events
- A well-defined security policy

Sidewinder software costs $19,995 for a single firewall system. The type enforcement capabilities offered by Sidewinder are a far cry from the potential capabilities possible with an RBAC product. We are not sure if Sidewinder will provide a reasonable vehicle for an RBAC product.

REFERENCES

Bobrowski, Steven, *Mastering Oracle 7 & Client/Server Computing,* SYBEX Inc, 1995.

Computer Systems Laboratory (CSL) Bulletin, "An Introduction to Role-Based Access Control," *NIST*, December 1995.

Cowart, Robert, *Windows NT Unleashed,* Second Edition, SAMS Publishing, 1995.

Draier, Enrique, "Complex Challenges for a Complex World," *IEEE Computer*, February, 1994, p. 10.

Epstein, Jeremy and Ravi Sandhu, "NetWare 4 as an Example of Role Based Access Control," *Proceedings of the First ACM Workshop on Role-Based Access Control, 30 November - 1 December 1995*, National Institute of Standards and Technology, Gaithersburg, MD, 1996.

Feinstein, H., R. Sandhu, C. Youman, and E. Coyne, *Role-Based Access Control, Phase 1*, National Institute of Standards and Technology, Gaithersburg, MD, May 1995.

Ferraiolo, D., R. Gilbert, and N. Lynch, "Assessing Federal and Commercial Information Security Needs," *NIST*, November 1992.

Government Accounting Office (GAO), *Using Structured Interviewing Techniques*, Program Evaluation and Methodology Division, GAO/PEMD-10.1.5, July 1991.

General Accounting Office (GAO), *Developing and Using Questionnaires*, Program Evaluation and Methodology Division, GAO/PEMD-10.1.7, October 1993.

Gligor, V., "Characteristics of Role-Based Access Control," *Proceedings of the First ACM Workshop of Role-Based Access Control, (30 Nov - 1 Dec 1995),* 1996.

Guttman, Barbara and Edward Roback, "An Introduction to Computer Security: The NIST Handbook," NIST Special Publication 800-12, *NIST*, October 1995.

Jolitz, William F. and Lynne Greer Jolitz, "Role-based network security: network security at the operating-system level," *Dr. Dobb's Journal*, May 1995, pp. 80-83.

Microsoft, *Windows NT 3.5, Guidelines for Security, Audit, and Control*, Microsoft Corporation, Seattle, WA, 1994.

Minasi, Mark; Christa Anderson and Elizabeth Creegan, *Mastering Windows NT Server 3.5*, SYBEX Inc., 1995.

Munro, Neil, "Foiling the Wily Hacker," *Washington Technology*, June 13, 1996, pp. 14-24.

Russell, D. and G. Gangemi Sr., *Computer Security Basics*, O'Reilly & Associates, Inc., 1991.

Sage, Andrew and James Palmer, *Software Systems Engineering*, John Wiley & Sons, New York, 1990.

Sandhu, R., E. Coyne, H. Feinstein, and C. Youman, "Role-Based Access Control Models,*" IEEE Computer*, February 1996, pp. 38-47.

Sandhu, Ravi, "Role Hierarchies and Constraints for Lattice-Based Access Controls," *European Symposium on Research in Computer Security (ESORICS-96)*, Rome, Italy, September 1996.

Scott, Raynovich, "The Proxy's Price: Managers Seek Firewalls that Match Network Applications," *LAN Times*, 19 February 1996.

Secure Computing Corp., *Technical Summary for Sidewinder release 2.2*, Secure Computing Corp., Roseville, MN, 1996.

TECSEC, *What is Veil?*, TECSEC, Inc., Vienna, VA, 1996.

U. S. Department of Defense, *Department of Defense Trusted Computer System Evaluation Criteria*, DoD 5200.28-STD, Ft. Meade, MD: National Computer Security Center, December 1985.

APPENDICES

APPENDIX A

ACCESS CONTROL

1.0 INTRODUCTION

This appendix contains a description of the three principal access control methodologies. The third methodology, role-based access control (RBAC), is discussed in some detail. There are three basic types of security access control: 1) discretionary access control, 2) mandatory access control, and 3) role-based access control [Ferraiolo *et al.* 1992, Russell and Gangemi 1991].

2.0 DISCRETIONARY ACCESS CONTROL (DAC)

DAC is an access policy that restricts access to files (and other system objects such as directories and devices) based on the identity of the users and/or the groups to which they belong. DAC requirements have been perceived as being technically correct for commercial and civil security needs, as well as for single-level military systems. Access is based on the discretion (i.e., the choice or selection) of another person who has been granted that privilege, but often not a security administrator. Usually access is based on an access control list (ACL). Unfortunately, DAC requirements do not meet the general needs of the commercial and civil government sectors [Russell and Gangemi 1991].

3.0 MANDATORY ACCESS CONTROL (MAC)

MAC is used for multilevel secure military systems, but its use in other applications is rare. Access is granted specifically to an individual for access to specific types of data. MAC assigns sensitivity labels to *all subjects* (e.g., users and programs) and *all objects* (e.g., files, directories, devices, windows, and sockets) in a computer network system. This means that there should be a mapping connection between any subject in the system and any object in the system. MACs use sensitivity labels to determine which subjects can access what objects in the network. A specific *subject's sensitivity label* specifies the level of trust or clearance associated with that user. An *object's sensitivity label* specifies the level of trust that a user or program must have to access that object. A MAC and its concomitant labeling implement a multilevel security policy for handling multiple information classifications at a number of different security levels (i.e., clearances) within a network [Russell and Gangemi 1991].

4.0 ROLE-BASED ACCESS CONTROL

An RBAC product can be based on a set of mathematically rigorous theories, definitions, and concepts which have been investigated to assure that the product can perform as promised. In this sense, RBAC is similar to the relational database management systems first identified and analyzed by E. F. Codd of IBM in 1970.

Many analysts have investigated RBAC models and have supplied a rigorous basis for the various alternative capabilities attributed to RBAC. Although an RBAC product has not been given a consistent definition in the general literature, when it is used in analytical studies it is usually well-defined for the specific study.

RBAC is policy neutral but it can be easily configured to specify a variety of policies [Sandhu 1996]. It is a means for articulating policy rather than embodying a particular security policy (such as one-directional information flow in a lattice for assuring confidentiality and/or integrity policies, or for aggregation policies). The policy enforced in a particular system is the net result of the precise configuration and interactions of various RBAC components as directed by the client [Sandhu 1996]. This access policy can evolve incrementally over the life cycle of the system since an RBAC product is extensible. The capability to modify policy to meet the changing needs of a dynamic organization is an important benefit of RBAC.

RBAC can be used by many organizations that prefer a centrally administered, non-discretionary set of controls to meet their security policies and objectives. Some organizations have policies and objectives that include maintaining and enforcing the rules and ethics associated with legal considerations such as the laws and respect for privacy in a medical environment when diagnosing ailments, treating a disease, and administering medicine. To support such policies, a capability to centrally control and maintain access rights is needed [CSL 1995].

The security administrator is responsible for enforcing policy and represents the organization as the "owner" of the system objects. An owner of an object is usually the person who created the object or the person or people who have all rights to the file, such as reading and writing to the file. Access control decisions were found to be based on the roles individual users take on as part of an organization. This includes the specification of duties, responsibilities, obligations, and qualifications. For example, the roles may include doctor, nurse, clinician, and pharmacist associated with a hospital or teller and loan officer associated with a banking system. The doctor's role includes privileges to perform formal diagnoses, prescribe medicine, or add an entry to a record of treatments performed on a patient. Nurses are not allowed to do the first two of these. The privileges defined for the role of pharmacist include those to dispense, but not prescribe, prescription drugs.

Within an RBAC product, permissions are associated with roles and users are made members of appropriate roles thereby acquiring the permissions granted by the concomitant roles. This arrangement greatly simplifies management of permissions. Roles are created for the various job functions in an organization and users are assigned roles based on their responsibilities (or needs) and qualifications. Roles can be granted new permissions as new applications and systems are incorporated, and permissions can be revoked from roles as appropriate, or as the objectives and requirements of a dynamic organization occur.

Role engineering is the process of identifying the roles performed by members of an organization and must include, as an input, the knowledge of relevant managers in the organization. Role engineering can be performed by companies that are knowledgeable about security and business process engineering methodologies.

The determination of membership and the allocation of privileges to a role are not so much in accordance with discretionary decisions on the part of a system administrator, but rather in compliance with organization-specific protection guidelines. These guidelines derive from existing laws, ethics, regulations, or generally accepted practices. The guidelines are non-discretionary in the sense that they are unavoidably imposed on, or not imposed on, particular users based on how specific job functions are performed. For example, a doctor has the privilege to prescribe medication but cannot pass that privilege on to a nurse.

Once roles are established with the system, the privileges associated with these roles remain relatively constant or change slowly over time. The administrative task is then to grant and revoke user membership to the set of specified roles within the system. The capability of an administrator to simply grant and delete membership to existing roles has been described as desirable. When a user's function changes within the organization, the user's memberships to all previous roles can be easily deleted and new ones granted. Finally, when a person leaves the organization, all of that person's memberships to all roles are easily deleted. When a new person is hired, that person can easily be added to the membership lists of pertinent roles.

For an organization that experiences a large turnover of personnel, a role-based implementation security policy is a logical choice. There are three different security management activities involved in role management:

- Defining roles
- Allocating roles to users
- Allocating access rights to resources based on roles

A National Institute of Standards and Technology (NIST) study team has interviewed several organizations that felt role-based access or access based on function was a control more suited to their needs than DAC or MAC. While add-on packages will give an organization access based on function for their systems, RBAC should be generally promoted just as DAC and MAC are.

Reports such as this one will provide useful inputs to the NIST process for setting standards for RBAC products.

5.0 RBAC FOR NETWORK SECURITY

The number of Internet hosts is expected to increase from 3.8 million in 1995 to 100 million by 1999, which means network security will be an increasingly important issue in the future [Jolitz and Jolitz 1995]. Role-based security is complementary to standard authentication, encryption, and threat-detection products. RBAC can be implemented as a policy that limits access with a product that requires little knowledge or maintenance from users. The concept of roles is used to simplify the description of allowable access characteristics of a host's user, while the concept of access path can be combined with roles to provide a level of geographic classification to determine a specific role based on the location of the user [Jolitz and Jolitz 1995]. The inclusion of access path should serve to diminish the possibility of an intruder gaining access to the network.

The role-based model requires a high degree of transparency. The manner in which this affects areas of role-based security design is presented below. A comparison of the RBAC product with the MAC and DAC security systems is shown in Table A-1, *Comparison of RBAC with MAC and DAC.* Some of the positive attributes of RBAC are:

- It is simple
- It is easily extensible to a more elaborate arrangement
- It is easy to understand
- It is easy to administrate

It is often the case that the security product resides in the kernel of the operating system. The *kernel* of an operating system is the portion of the operating system (OS) that interfaces with the system hardware and hence is specific to the particular system characteristics. That is, the kernel of any OS consists of the basic instructions required for interfacing with the unique system hardware of a given system.

5.1 Approaches to Security

The objective of network security is to deny access to the system capabilities to those who have no authority. Routes to access are complicated by locks requiring a combination (password authentication) and file-access-mode discrimination access control. Often, passwords provide a reasonable degree of security when machines are relatively isolated.

Intruders can pick the combination of an account or intercept the use of the password as it transits the network. Computers can be compromised even if the intruder has no idea where it is.

Table A-1. Comparison of RBAC with MAC and DAC

Area	**MAC or DAC**	**RBAC**
Administrative Overhead	Overhead increases with complexity of the user-object population	Low overhead that does not increase appreciably with additional users or objects
Operating Mode for Making User Assignments	Can only operate in manual mode	Can operate in either manual or automatic modes
Transparency	Requires changing the operating or the host system, security system is highly embedded in the overall system	High degree of transparency is a requirement, security may be located in the kernel program
Upgrade Complexity	Upgrades may require changing the operating system and some applications programs	Upgrades usually require only that the kernel be changed
User Access Determination Complexity	Fine-grained access is difficult to manage and requires a complex list-of-privileges	Privileges are identified in the program kernel on a case-by-case basis
Network-Level Security	Often firewalls are required to yield sufficient security	RBAC security is simple to employ but hard to bypass
Access Path	Overhead increases with degree of security offered	RBAC security utilizes both role-based products and access path considerations
Geographic Access	Users can access files no matter where the user is relative to the file location	Files can be marked to disallow access outside a geographic area

Many existing facilities and packages have been enhanced to cope with this increased threat. A basic package for dealing with the increased threat is Kerberos. It can plug holes in the network by beefing up existing authentication products. It can also discover vulnerabilities before an intruder does. Kerberos guards transmitted data using cryptographic keys known as *tickets* to protect the security of the messages that are sent to the system. Kerberos never transmits passwords over the network, even in encrypted form. Passwords reside only in a highly secure machine called a *key server*.

Security systems such as Kerberos require additional administrative overhead and this overhead increases with the sophistication of the product and the model of security it provides. When the client does not attend to the details of managing the security system by keeping the system current, the purpose of the security system may be defeated.

5.2 Security Elements at the Network Level

Unlike a gateway firewall, role-based security does not require administration and monitoring to review new access requirements and intrusion attempts. The role-based security model may consist of roles and privileges, access path, and transparency. Each of these elements can be crucial to creating a properly designed product for access control.

5.2.1 Roles and Privileges or Permissions

In some security systems, privileges can be recorded as bits set in a list of privileges. Most privileges are granted because of system-management functions such as manipulating or adding new devices; reformatting disks; changing the access privilege, protection, or ownership of a file; and so forth. As a system grows, more privileges are added. Privileges may be created because of the existence of other privileges as a way of enabling groups of them or offering special treatment. The management or control of these privileges can become a difficult job.

The RBAC model is essential to diminishing the difficulties associated with providing network security. Users have access through their authorized roles, and generally roles do not change. This means that although security needs in the operating system become more elaborate, the model stays simple from the user's perspective. Roles can be set once by the access path and never allowed to change. Since they are independent of the user/group identification concepts, roles can be used by lower levels of the OS kernel to bound access [Jolitz and Jolitz 1995].

5.2.2 Access Path for Information or Services

Users can access information or services from the computer based on their *access path*. The way in which the information is accessible may be different, depending on the information's destination. Since the mechanisms that determine path are extremely low level, an intruder must find ways to imitate access, by either gaining or simulating physical access. With a role-based model based on access path, the scope of access is limited without increasing system-management overhead, that is, no user-account profile needs to be maintained [Jolitz and Jolitz 1995].

5.2.3 Security Transparency

Attempts at improving system security affect both the operating system and the host on which it is installed. Because many security products are deeply embedded in the operating system and the host on which they are installed, these security products become less desirable. With RBAC, a high degree of transparency is a requirement; otherwise it would be too troublesome for a user to consider using it [Jolitz and Jolitz 1995]. The demand for transparency affects all areas of the role-based security design. Some comments on RBAC transparency are shown in Table A-2, *Advantages of RBAC Transparency*.

Table A-2. Advantages of RBAC Transparency

Comments on RBAC Transparency
RBAC may be located in the kernel program entirely (except for a single utility program), requiring no changes to utility or application programs.
There are no external interfaces for programmers to subvert or oversee.
There are no conflicts with existing industry standards, either *de facto* or *de jure*.
RBAC is entirely independent of other security facilities for encryption, authentication, and intrusion detection.
RBAC requires minimal knowledge by the security administrator to manage.
RBAC may require no changes to system operation, network management, or other procedures.

5.3 Some Comments on the RBAC Model

The primary advantage of the RBAC model is its simplicity. It should fit easily into any existing modern operating system or at an application level. The model is easy to understand and administrate, yet may be difficult to subvert because it remains so fundamental. Simplicity helps avoid the possibility that holes might develop as security complexity increases.

It is possible to incorporate security at installation time without requiring additional administrative obstacles to gain access to simple, straightforward procedures. Easing the decision-making burden for the user during installation in a safe way is critical. The security software system can automatically take care of sensitive, system-related files [Jolitz and Jolitz 1995].

The inflexible binding of privilege and access rights with the path of access is a strength as well as a weakness. If the RBAC product is so low level that it is nearly impossible to bypass, it can be very difficult when you may want to read a privileged file using the network. Also, remote system management may be hampered. Since users and administrators may do their own system administration on-site, security administration should not be a problem.

5.4 Sensitive File Restrictions

A utility is needed to allow a user with the appropriate role to remove file restrictions. The use of this utility requires authorization directly from the console through a known secure path. If the authorization is not supplied, the restriction is not removed.

For example, suppose an intruder somehow got a user running with a privileged role to execute something useful for the intruder. In this case, the console would unexpectedly request authorization. This attempt at subversion can be tracked down and the intruder revealed without loss of integrity [Jolitz and Jolitz 1995].

5.5 Subverting Role-Based Security

Like other security products, RBAC is not foolproof. However, the identified roles provide the scope of access to privileges and may be governed by access path. The easiest way to compromise the system is to gain access to a trusted path. Thus, RBAC security is not intended to deal with insider related threats. The implicit assumption is that physical access to the machine itself is trusted, and access to the immediate LAN is trusted within limits [Jolitz and Jolitz 1995].

RBAC security is not intended as a complete answer for all security needs, but it should make the job of subverting the system much more difficult. Moreover, the RBAC model does offer certain security capabilities that can meet the needs of many commercial, civil government, and perhaps military organizations interested in protecting information privacy.

APPENDIX B

STRUCTURED INTERVIEWS

A structured interview is a situation where evaluators ask the same questions of numerous individuals or individuals representing numerous organizations in a precise manner, offering each interviewee the same set of questions and possible responses.   To conduct a structured interview, a data-collection instrument (DCI) may be used. A DCI is a questionnaire containing questions presented in a systematic, highly precise fashion. Its purpose is to enable the evaluator to obtain uniform data that can be compared, aggregated, and subjected to additional statistical analysis if the data are quantitative. Some attributes of a good questionnaire are listed and described in Table B-1, *Some Attributes of a Well-Designed Marketing Questionnaire*. The attributes are not listed in any priority order.

Table B-1. Some Attributes of a Well-Designed Marketing Questionnaire

Attribute	**Description**
Objectives	Interviews are meant to obtain data that may otherwise not be documented or, if documented, may need some interpretation.

Relevance Questions should be relevant to the study being conducted and should yield answers that may be used to determine further actions relative to a service or product.
Attendance Questionnaires should be presented to a single interviewee by an interviewer who asks questions and a scribe who records responses.
Interviewee Rapport Initial questions should be especially easy and should set the interviewee at ease.
User-Friendliness Questions should be worded so that they are easy to answer for knowledgeable interviewees. Formulate the questions so that the answers are preferably either yes/no or multiple choice.
Ease of Analysis Questions should be worded so that data contained in the filled-in questionnaire are directly applicable to a straightforward planned analysis and evaluation process.
Logical Order Questions should be created and placed in a logical order so that they make sense to the interviewee.
Respondents Selection Give preliminary consideration to identifying interviewees who are capable of answering the questions in a knowledgeable manner.
Advanced Information Mail the following information to prospective interviewees: 1) letter from mandating authority, 2) copy of the questionnaire, and 3) tutorial information.
Tutorial In some cases, present the interviewee with a brief tutorial so that they may better understand the relevant context.

APPENDIX C

AN APPROACH TO DESIGNING ROLE-BASED ACCESS CONTROL IMPLEMENTATIONS

1.0 INTRODUCTION

The approach to designing the modifications for implementing the role-based access control (RBAC) model capability in a major computer system environment, such as Windows NT, ORACLE 7, or NetWare 4, is critical. This appendix contains a description and rationale of some basic rules for creating an architecture that should minimize the effort to create an RBAC product. The thrust of this architecture is to provide a modification to an existing commercial off-the-shelf (COTS) package. However, there are other ways to implement an RBAC product.

The modification to the system environment to accommodate the users' needs for an RBAC security product should be based on the requirements elicited from the system users. Even if the user requirements are defined very accurately, requirements are dynamic and will change over time. Therefore, it is important that an RBAC modification package be designed and implemented so that it is easily extensible to include different capabilities (more, less, or modified) as user needs or desires change.

2.0 APPROACH

When creating an RBAC modification to be embedded in an existing computer system environment, we will call the modification the "RBAC adjunct." For a modification to any existing environment, the RBAC adjunct should be designed to deliver the capabilities that satisfy the requirements expressed by most of the system users interviewed in the marketing survey. The user requirements can be expressed in the form of: 1) *absolute requirements* that include all access control security needs independent of their current system environment, or 2) *relative requirements* that include all access control security requirements not included in their current computer system environment.

The first form can be thought of as a *generic set* of requirements that satisfies the majority of interviewed user-clients, independent of their particular system environment. The second form is dependent on the particular legacy system currently installed by the user-client and is called the *legacy set* of requirements. In the generic set case, the requirements from all the interviewees will be collated into a single framework. In the legacy set case the security requirements will be defined for the majority of a particular segment of user-clients, for example, for Windows NT, ORACLE 7, or NetWare 4 users. Interviewed users were first identified as Windows NT, ORACLE 7, or NetWare 4 users and then their requirements were defined relative to their particular environment.

The RBAC adjunct should be designed with two primary objectives:

- It meets the needs of a majority of clients for each particular legacy environment (or it meets the needs of the generic requirements set from all the interviewees).

- It uses an architecture that easily allows for extending the RBAC adjunct to include the additional security capabilities desired by specific users.

Extensibility can be achieved through modularity of the RBAC adjunct design or by using object-oriented analysis and design (OOAD) so that the adjunct is composed of objects which can be easily modified. OOAD is probably the best method but this approach depends on the computer system environment into which the RBAC adjunct will be embedded. A potential architecture for the RBAC implementation is shown in Figure C-1, *Illustration of an RBAC Software Configuration*.

Figure C-1. Illustration of an RBAC Software Configuration

This RBAC software approach will allow each user to have a security system that meets their specific needs at reasonable acquisition and operational costs and without having to implement additional security systems to give them their needed security capabilities, as is often the case at present. The architecture should be designed so that additional roles, additional permissions, additional constraints, etc., can easily be added to the basic structure. The client should have the capability to easily form the RBAC adjunct system to meet their immediate needs and to later alter the system to meet their emerging needs as they arise.

This kind of approach to the RBAC software could be performed by using computer screens which are designed with the alternative capabilities listed for user/administrator selection of the capabilities that they desire to meet their initial system requirements. Also include the options to later extend the adjunct so that modifications to the security product can be easily provided.

By creating a software package that has the capabilities needed by a large set of users and that also can be extended to include additional capabilities that these users may someday want in the future, the security package can be created at a reasonable cost by a commercial vendor. Then the user who purchases the package can adjust it to fit their immediate needs and gradually extend it to include additional capabilities as the needs arise. By attempting to configure security packages to specific users based on a too-limited basic package, the costs for each user may go up too steeply to cover costs for their required specific programming. In addition, these specifically added program instructions will be very difficult to debug and may, in some cases, create much havoc for the client.

With respect to the proper location of security software, Brian O'Higgins of Northern Telecom states that "over time, [computer security] will get sucked into the operating system. It is the natural evolution [Munro 1996]." If this is true, then it would behoove the vendor to thoroughly investigate the need to implement the RBAC security product at the operating system level.

APPENDIX D

USER SECURITY GENERAL REQUIREMENTS AS DETERMINED BY THE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST)

In 1992, NIST interviewed many Government and commercial organizations to assess these organizations' current and future security needs in all three major areas: commercial, civil government (Federal or state [one case]), and military [Ferraiolo *et al.* 1992]. NIST discovered some high-level, or general security requirements, from these interviews, namely that these organizations:

- Had unique security needs;

- Had organizational security requirements that changed over time and could not be totally specified at the time of security product acquisition;

- Felt that security standards had not emerged that will allow integrating security across a multivendor environment (i.e., a heterogeneous open system);

- Felt that security standards should include a wide range of assurances (e.g., a generally accepted commercial practice level that minimizes cost of developing new systems or retrofitting new security functionality in an existing system);

- Felt that administering computer security, particularly access control, was burdensome in a heterogeneous-distributed environment because this function took more time and effort than they felt was appropriate;

- Felt that computer security-related products should easily reflect organization security policy while managing security functions and providing improved security administrator interfaces to balance the increasing need for protection and limitations on the staff resources devoted to it;

- Wanted vendor security products to be flexible enough to serve a broad spectrum of security needs at the: operating system, application,
- organizational, and site levels;

- Felt that vendor security systems should provide a single-user view of security services across a wide range of operating systems.

Some additional NIST findings included that these organizations:

- Felt that a third party vendor should provide a "stamp of approval" with regard to the trustworthiness of the security systems they were buying, because the current evaluation and certification process with respect to the Trusted Computer System Evaluation Criteria (TCSEC) security standard (which is oriented toward Department of Defense security needs only) was not perceived as meeting user needs for a variety of reasons;

- Felt that security features should interoperate with other security services on both local and remote machines without the need to train users in new security products (i.e., be amenable to an open system environment with a user-friendly security system);

- Felt that security technology should support users in working effectively together by allowing the sharing of information, resources, and network applications from whatever desktop device users have chosen, while providing a common set of security services.

APPENDIX E

DATABASE INFORMATION

A *data model* defines the types of data objects that may be manipulated or referenced with a database management system (DBMS). A DBMS is a management system that has the following capabilities:

- It manages a large quantity of data in physical storage.

- It provides logical data structures with which humans can interact so that they are independent of the framework used for physical data storage.

- It reduces data redundancy and maintenance needs while increasing flexibility of use of the data by providing independence between the data and the applications programs that use the data.

- It provides effective access to the data by users who may not be expert programmers.

Databases may consist of one or more of the following data models: record-based models, structural models, and expert database models. There are three levels of data model [Sage and Palmer 1990]:

- An *external model* which represents a data model at the level of the user's application,
- A *conceptual model* which is an aggregation model that includes several external models, and
- An *internal model* which is a technical-level model that describes how the conceptual model is actually represented in computer storage.

A *mapping* is the relation between the various levels of data models and architectures in a generic diagram of a database system. Mappings specify and describe the transformations that are needed to obtain one model from another. The user specifies a *data description language* (DDL) which provides the source and target data structures that describe the mapping that is desired between source and target data. Schemas represent knowledge about concepts and are structurally organized about some theme. *Schemas* are data structures for representing generic concepts. The user of a database must interpret the real world that is outside of the database in terms of real-world objects (where objects are entities-with-relationships) and actual activities that involve these objects.

The *data manipulation language* (DML) provides the basis for the operations submitted to the DBMS as a sequence of queries or programs for storing, accessing, modifying, or creating data. The development of schemas or logical data results in a *DBMS architecture* or *DBMS framework*. The *metadata* or data about the data are stored in a *data dictionary* or *data directory*. A data dictionary tells about the contents of the records in the database as well as the relationships among the data records.

Interface languages allow for interactions with the databases. There are three types of interface languages:

- *DDLs* provide the basis for definition of schemas and subschemas (for simplifying access to databases),
- *DMLs* are used to develop database applications, and
- Data query languages or *structured query languages* (SQLs) are used to write queries and reports.

A single database language can be composed of all three of the interface languages and the user of these languages is called a database administrator (DBA). Data models represent a paradigm for representing, storing, organizing, and managing data in a database using the following three components:

- A *set of data structures* which defines the fields and records that are allowed in the database, e.g., lists, tables, hierarchies, and networks;

- A *set of operations* which defines the admissible manipulations that are applied to the fields and records that comprise the data structures, e.g., retrieval, combine, subtract, add, and update; and

- A *set of integrity rules* which defines or constrains allowable or legal states or changes of state for the data structures that must be protected by the operations, e.g., only dates between 05/12/69 and 12/25/75 are acceptable.

A *relational database* consists of data records that are the rows of a physical database. A *record* consists of a set of *fields* which reveal additional information about the record. An *individual record database* is a collection of records. A *relational database* is a modification of the individual record model that provides a mathematical basis for operations on records. Data structures in a relational database consist of relations or field sets that are related. Every relation may be considered to be a *table*. Each *row* in the table is a record or *tuple*. Every *column* in each table or row is a field or *attribute*. Each field has a domain that defines the admissible values for that field. The *operations* in a relational database form a *relational algebra* and are defined mathematically. The operations in a relational model must operate on entire relations or tuples, rather than on individual records. A relational database can be described as a database:

- That has data which are presented in tabular form without the need for navigation links or pointer structures between various tables,

- That has a relational algebra that can be used to automatically prepare joins and unions of logical record files, and

- Where new fields can be added to the database without the necessity of rewriting any programs that used previous versions of the database.

Other models for databases include the structural models, namely hierarchical, network, and entity-relationship (ER). The *hierarchical model* represents data with a hierarchical structure where nodes are connected by directed links from a "child" to a "parent." A *network model* is more general than the hierarchy model so that a child can have more than one parent. The *ER model* is a generalization of both the hierarchy and network models. The ER model, which is based on well-established graph-theoretic concepts, generally has the following representations: 1) *rectangles* represent entities, 2) *diamonds* represent relationships, 3) *circles* represent attributes of entities, and 4) *relational tables* represent a collection of entities. See Figure E-1, *Illustration of an Entity-Relationship Model*.

Figure E-1. Illustration of an Entity-Relationship Model

APPENDIX F

A BRIEF OVERVIEW OF A ROLE-BASED ACCESS CONTROL PRODUCT

1.0 INTRODUCTION

A *role-based access control* (RBAC) system is a security control product that grants access privileges to users based on a user's role and users are made members of appropriate roles. Roles can be created for various job functions in an organization. Users can be made members of roles as determined by their responsibilities and qualifications and can be easily reassigned from one role to another. Roles can be granted new permissions as new applications and systems are incorporated, and permissions can be revoked from roles as needed. The RBAC model is based on the notion of users, roles, permissions, and sessions [Sandhu *et al*. 1996].

A *user* in this model is a human being. A *role* is a job function or job title within the organization with some associated semantics regarding the authority and responsibility conferred on a member of the role. A *permission* is an approval of a particular mode of access to one or more objects in the system. Permissions can range from very coarse grain (*e.g.*, access to an entire network) to very fine grain (*e.g.*, access to a specific record in a particular database). Roles are assigned during sessions. Each *session* is a mapping of one user to possibly many roles. The notion is that a user establishes a session during which the user activates some subset of roles to which they are a member. Multiple roles can be simultaneously activated during a session [Sandhu *et* *al.* 1996]. Sessions are under the control of individual users. The roles activated in a session can be changed at the user's discretion. In the basic model, permissions are assigned to roles and users are assigned to roles and these assignments are called administrative permissions.

One driving motivation behind RBAC is to facilitate security administration and review. An appealing attribute of RBAC is that it can be used to manage itself. Managing the roles and their interrelationships in large systems can be a formidable task. An objective of RBAC is to provide a flexible framework that is easy to use, understand, manage, and customize for articulating and enforcing access control policy. RBAC is policy-neutral and hence does not represent any means for implementing or articulating any particular policy. The ability to modify policy to meet the changing needs of an organization is an important benefit of RBAC [Sandhu *et al.* 1996].

RBAC supports three well-known security principles [Sandhu *et al.* 1996]:

- *Principle of least privilege* (only those permissions precisely required for the tasks performed by the user in the role are assigned to the role),

- *Separation of duties* (ensure that mutually exclusive roles must be invoked in order to complete a sensitive task), and

- *Abstract permissions* (allows permissions for roles such as credit and debit for an account object).

Role hierarchies can be identified for structuring an organization's lines of authority and responsibility. Higher level roles inherit permissions from lower level roles. The more powerful roles are at the top. *Limited inheritance* provides the capability to limit the inheritance associated with a hierarchy. *Private hierarchies* provide private permissions that can be shared without inheritance for roles defined at a specific level or levels.

The RBAC model also allows for constraints. *Constraints* place limitations on a user, role, or session and they are a powerful product for laying out the higher-level policy requirements of an organization. Constraints include *mutually exclusive roles* where the same user can be assigned to at most one role in a mutually exclusive set. This supports the separation of duties. Other constraints are shown in Table F-1, *Potential Role Constraints*.

Table F-1. Potential Role Constraints

Descriptions of Some Role Constraints
Cardinality limits the number of users assigned to a particular role.
Prerequisite roles are based on competency and appropriateness where a user can be assigned to Role A only if they are already assigned to Role B.
A user can belong to two roles but cannot be active in but one of them at the same time.

The number of sessions a user can be active in can be limited.
The number of sessions to which a permission is assigned can be limited.
Constraints on role hierarchies can be established.
Private roles may be established to assure mutual exclusivity without conflict.

RBAC cannot enforce the way these principles are applied. The system administrator can configure the RBAC system to violate these principles if they so desire. RBAC is not a panacea for all access control issues. *User-assignment constraints* are effective only if suitable external discipline is maintained in assigning user identifiers to people (i.e., users). Similarly, for *permission constraints,* if the same operation is sanctioned by two different permissions, an RBAC product cannot effectively enforce cardinality and separation constraints.

During sessions, a user may be a member of more than one role, but cannot be active in more than one role at the same time. There are still some issues within the RBAC model concerning hierarchies and constraints which have yet to be resolved so that a consistent RBAC model can be developed.

The major components of an RBAC architecture include the following:

- Administrative tools
- Run-time system
- Repository
- Audit system
- Role engineering tools

An RBAC administrative system needs the following tools to operate:

- Constructor syntax
- Relational database management system (e.g., ORACLE 7)
- SQL or similar language (e.g., ORACLE SQL*Net)
- Preprocessor or code generator for SQL statements
- Expert system

Role engineering is the process of defining the roles to be established to reflect an organization's operating environment [Feinstein *et al*. 1995]. The RBAC system itself makes no assumptions regarding what roles will be defined or whether there are constraints or hierarchies among roles. Thus role engineering is external to an RBAC system. A similar situation exists with expert systems. Like RBAC, an expert system shell provides a structure for implementing a particular expert system, but the knowledge engineering process (that is, identifying relevant experts, creating a structured questionnaire, eliciting decision data and rules from experts, analyzing and synthesizing data, and creating a process for constructing the rule-base and database systems) is done external to the expert system shell.

When role engineering is done independently from the RBAC environment, it may be supported by guidance and tools. Guidance can be provided to supply: the criteria for the definition and naming of roles, the granularity of roles, definition of permissions, and the assignment of permissions to roles.

Tools can be used to keep track of definitions and relationships and to enforce consistency and completeness constraints.

Once a role-based access control policy has been completely defined for some client's system, an artificial intelligence method can be implemented in the RBAC software package that will allow a user to obtain an explanation of the facts regarding an individual user, a particular role, and/or a specific session.

2.0 RBAC DISCUSSION

The use of RBAC is motivated by three fundamental goals: 1) it simplifies existing access-control management functions so that security control can be performed quickly and at lower cost, 2) it provides access control functions not readily available in the other models, and 3) it provides a model for a security system that can be easily extended to provide additional security capabilities to meet dynamic and unique user security-requirements. This means that RBAC security systems offer advantages for both classified and unclassified government clients as well as some commercial clients. RBAC is suited to client environments with the following three types of characteristics [Gligor 1996]:

- **User Characteristics**
    - Large number of users (e.g., more than 100)
    - Few security administrators (e.g., less than 20)
    - Frequent change of job responsibility per user or high user turnover-rate
- **Data and Application Characteristics**
    - Very large number of data objects (e.g., greater than 10,000)
    - Stable set of applications (e.g., little or no application development)
    - Data and application sharing-patterns among users depends on the job functions and do not change frequently
    - Applications access to data objects may be user-independent
- **Organizational (= Enterprise) Characteristics**
    - Stable organization structure (e.g., job definitions change infrequently)

- Ownership of data and application is restricted to the organization

- Control data and application access are retained to the organization

- Individual user, group, and administrator accountability is required by the organization (e.g., both before-the-fact and after-the-fact audit are necessary)

- Periodic assessment of the access-control policy enforcement across the organization is required

An RBAC product can be managed in at least three different modes:

- A single security administrator
- Multiple administrators
- Self-administered by the RBAC product itself

A client may wish for a single RBAC product that possesses all three management modes. For example, the client may wish for the system to operate on the basis of a single administrator but should the system become more complex and require hiring additional administrators, then the system should operate properly when more than one administrator is making assignments and granting permissions. In the case of multiple security administrators or officers, one of them is usually designated as the chief security officer. In addition, the client may wish for the system to take over in a self-regulating mode to relieve the work load on the administrators. An RBAC product must be designed so that no matter how an RBAC product is managed there will be no conflicts among the assignments, no matter how they are determined.

APPENDIX G

SOME RELEVANT DEFINITIONS

Access - is a specific type of interaction between a subject and an object that results in the flow of information from one to the other.

Access Control - is the process of limiting access to the resources of a system only to authorized persons, programs, processes, or other systems (in a network).

Access Review - checks who has access to what.

Administrative Role - includes permission to modify the set of users, roles, or permissions, or to modify the user assignment or permission assignment relations.

Constraint - is a relationship between or among roles for limiting the higher-level policy requirements of an organization.

Constraints Maintenance - is a check to see if the security product is properly maintaining the consistency of constraints.

Discretionary access control (DAC) - is an access policy that restricts access to files (and other system objects such as directories and devices) based on the identity of the users and/or the groups to which they belong.

Extensibility - is the capability of a product (such as ORACLE 7 or NetWare 4) to be extended to include an RBAC product for performing security.

Group - is a set of users.

Mandatory access control (MAC) - is a security system which grants access specifically to an individual for access to specific types of data.

Object - is a passive entity that contains or receives information.

Permission - is a description of the type of authorized interaction a subject can have with an object.

Permission Assignment - in the RBAC model is the relationship between permissions and roles and is a many-to-many relation.

Proxy - is a go-between that can communicate with the Internet and with an internal network.

Resource - is anything used or consumed while performing a function. The categories of resources are time, information, objects, or processors.

Role - is a job function within an organization that briefly describes the activities or responsibilities of a system user who is assigned to the role.

Role-based access control (RBAC) - is the process of assigning rights and permissions to roles rather than to individual users, with users then assigned to roles as appropriate. An RBAC product is a complementary capability to existing mandatory access control (MAC) and discretionary access control (DAC) security capabilities.

Role Administration - is the capability to define the roles of system users.

Role Engineering - is the methodology for establishing a valid set of roles with assigned permissions. The definition of roles is a requirements engineering process and the goal is to define a set of roles that is complete, correct, and efficient. Role engineering must capture the organization's business rules, as these relate to access control, and reflect these rules in defining, naming, structuring, and constraining a valid set of roles.

Role Hierarchy - is a partial order relationship established among roles and addresses the need for structuring roles to reflect an organization's lines of authority and responsibility.

Sensitive Information - is any information, the loss, misuse, or unauthorized access to, or modification of, which could adversely affect the national interest or the conduct of federal programs, or the privacy to which individuals are entitled under section 552a of title 5, United States Code (the Privacy Act), but which has not been specifically authorized under criteria established by an Executive Order or an Act of Congress to be kept secret in the interest of national defense or foreign policy [NIST 1995].

Session - is a mapping between a user and an active subset of roles to which a user is assigned.

Subject - is an active entity, generally in the form of a person, process, or device, that either causes information to flow among objects or changes the system state.

System Administrator - is the individual who establishes the system security policies, performs the administrative roles, and reviews the system audit trail.

User - is any person who interacts directly with a computer system.

User Assignment - in the RBAC model is the relationship between users and roles and is a many-to-many relation.

APPENDIX H

SOME RELEVANT ACRONYMS

ACL	Access Control List
BSD/OS	Berkeley Software Design/Operating System
CICS	Customer Information Control System
COTS	Commercial off-the-shelf
DAC	Discretionary Access Control

DBMS	Database Management System
DBA	Database Administrator
DCI	Data-Collection Instrument
DDL	Data Description Language
DES	Data Encryption Standard
DML	Data Manipulation Language
ER	Entity-Relationship
FTP	File Transfer Protocol
GOTS	Government off-the-shelf
HTTP	HyperText Transfer Protocol
HW	Hardware
LAN	Local Area Network
LOCK	Logical Coprocessor Kernel
MAC	Mandatory Access Control
MISSI	Multilevel Information Security Systems Initiative
MVS	Multiple Virtual Storage
NC	Network Computer
NCSC	National Computer Security Center
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
OOAD	Object-Oriented Analysis and Design
OS	Operating System

PC	Personal Computer
PIN	Personal Identification Number
RBAC	Role-Based Access Control
RDBMS	Relational Database Management System
SNS	Secure Network Server
SQL	Structured Query Language
SSL	Secure Socket Layer
SW	Software
TCP/IP	Transmission Control Protocol/Interface Protocol (Internet Protocols)
TCSEC	Trusted Computer System Evaluation Criteria (i.e., the Orange Book)
TSDM	Trusted Software Development Methodology
UDP	User Datagram Protocol
UNIX	Uniplexed Information and Computing System
URL	Uniform Resource Locator
WWW	World Wide Web

APPENDIX I

C2 SECURITY CONTROL

The C2 environment includes the capabilities of C1 which is an environment that includes passwords (or some similar product) for identifying and authenticating users before letting them use the system, and a discretionary access control protection of files and other objects. The C1 system is designed to keep users from making honest mistakes that could damage the system or could interfere with other users' work. C1 systems must also provide a system architecture that is capable of protecting system code from tampering by any application. The system must be tested to ensure that it works properly and that the security features cannot be bypassed in any obvious way. There are also specific documentation requirements. C2 security systems provide the following additional security protections [Russell and Gangemi 1991]:

- Accountability of individual users through individual password controls and auditing to keep track of who is doing what in the system;

- More detailed discretionary controls to include granularity down to a single user; and

- Object reuse to ensure that any data left in memory, on disk, or anywhere else in the system does not accidentally become accessible to another user or intruder.

The C2 system architecture must allow system resources to be protected via access control features and to provide more rigorous testing and documentation.

APPENDIX J

THE QUESTIONNAIRE
SETA SBIR Research on Role-Based Access Control
Interview Protocol (Questionnaire)
March 1996

Interviewer: ______________________________________ Time at Start: ____________

Organization: __________________ Date: ____________ Time at End: ____________

OPENING - INTERVIEWER SELF-INTRODUCTION

SETA has been selected by the National Institute of Standards and Technology (NIST) to conduct research in the area of Role-Based Access Control (RBAC), a security product for information systems. The purpose of this interview is to gather information about present and future security requirements and their priorities from a cross-section of organizations similar to yours. You were selected based on your knowledge of Federal government sector systems security. Please be assured that this is not an audit or an evaluation. This interview is strictly to gather information for our task and your privacy will be protected.

With your permission, we would like to record this interview.

PERSON BEING INTERVIEWED

Name: ______________________________ Phone No.: _____________

Organization: ____

APPLICATION ENVIRONMENT

We want to understand what types of applications are being used by your organization and how you classify them.

1. What types of system do your applications run on? (Check one or more.)
( ) Mainframe ( ) Minicomputer ( ) LAN

2. What kind of operating environment (i.e., OS/2 or DBMS) are you using (e.g., MVS, ORACLE, NetWare, Windows NT, or transaction management system such as CICS)?
List here: ____

3. Is there a new commercial system that you desire or plan to purchase?
( ) Yes ( ) No
If you do, what system is it? ____________________________________

4. What classifications of data do your organization's users deal with? (Check one or more.)
( ) Unclassified ( ) Unclassified but sensitive ( ) Classified

5. What principle(s) **do you actually use** for security? (Check one or more.)
( ) Principle of least privilege [only those permissions precisely required for the tasks performed by the user in the role are assigned to the role]
( ) Separation of duties [ensures that mutually exclusive roles must be invoked in order to complete a sensitive task]
( ) Abstract permissions [allows permissions for roles such as credit and debit for an account object]

6. What principle or principles **do you desire to use** for security? (Check one or more.)
( ) Principle of least privilege ( ) Separation of duties ( ) Abstract permissions

7. Does your agency use software (services or applications) that is unique to your organization? ( ) Yes ( ) No

8. Please define the categories of unique service/applications software that you use.
List here: ____

9. Does your agency use any software (services or applications) that is common to other organizations (i.e., COTS or GOTS software)? ( ) Yes ( ) No

10. Please define the categories of common service/applications software that you use. List here: ____

11. Do you use COTS applications software? ( ) Yes ( ) No
List here: ____

12. How many of the applications for your system are generated by your programmers?
( ) None ( ) 10s ( ) 100s ( ) 1,000 or more

13. What is(are) the size(s) of these applications? (Check one or more.)
( ) 1s ( ) 10s ( ) 100s ( ) > 1,000 large applications
( ) 1s ( ) 10s ( ) 100s ( ) > 1,000 medium applications
( ) 1s ( ) 10s ( ) 100s ( ) > 1,000 small applications

14. Does the concept of user-roles appear in your applications software?
( ) Yes ( ) No

15. Please list the security product(s) you currently use.
( ) ACF2, RACF, or TOPSECRET ( ) DBMS Security
( ) Other, please specify:____

IV. USER CHARACTERISTICS

We want to gather information about the users on your system and how jobs are defined.

1. How many users does your system have ( ) or anticipate having ( )?
( ) Less than 50 ( ) 50 to 100 ( ) 100 to 1,000 ( ) More than 1,000

2. Do the users on your system change job assignments frequently?
( ) Yes ( ) No

3. Do the definitions of the job assignments change frequently?
( ) Yes ( ) No

4. How did you interpret the word "frequently," e.g., 1/week? ____________

5. Is there a high-turnover rate for users on your system?
( ) Yes ( ) No

V. DATA AND APPLICATION CHARACTERISTICS

We want to gather information about the databases and programs that you use.

1. How many database entities or record-types does your system have ( ) or anticipate having ( )? ( ) 100s ( ) 1,000s ( ) 10,000s

2. Are your databases subdivided according to user-access for security reasons, e.g., personnel database may allow anyone to access "employee number" but only allow some managers access to "salary"? ( ) Yes ( ) No

3. Is the need for applications access to database-stored items dependent on the particular users whom the system services? ( ) Yes ( ) No

4. Are any of your operational applications controlled by a security administrator?
( ) Yes ( ) No

5. How many security administrators do you have ( ) or anticipate having ( )?
( ) 1 - 9 ( ) 10 - 19 ( ) 20 - 30 ( ) 31 - 100 ( ) 100s

6. How many full time equivalent administrators are required? ___________

7. At what level is your security performed?
( ) Applications ( ) Data Object ( ) Both

8. Do your users access from distributed platforms? ( ) Yes ( ) No

9. Are separate log-ins required for each distributed platform? ( ) Yes ( ) No

VI. ORGANIZATION (= ENTERPRISE) CHARACTERISTICS

We want to gather information your organization.

1. Does your organizational structure reflect the enterprise's job definitions?
( ) Yes ( ) No

2. Is your organizational structure stable?
( ) Yes ( ) No

3. Is ownership of data and applications restricted to your organization?
( ) Yes ( ) No

4. Is control of access to data and applications restricted to your organization?
( ) Yes ( ) No

5. What accountability is required by your organization? (Check one or more.)
( ) User ( ) Group ( ) Administrator ( ) None

6. Is periodic assessment of access-control policy enforcement across your organization required? ( ) Yes ( ) No

VII. APPLICABILITY OF ROLE-BASED ACCESS CONTROL

We want to gather information your use or desires for an RBAC capability.

1. Have you tried to implement RBAC? Did you encounter any problems?
( ) Yes ( ) No If yes, what were they?____________________________
____

2. Are you using an RBAC capability at present? ( ) Yes ( ) No

2.1 If yes, how is RBAC managed? ( ) Using roles ( ) Not using roles
Please explain here, e.g., how are roles assigned? ________________________
2.2 If no, do you have plans to use RBAC in the future? ( ) Yes ( ) No

3. Is your system managed such that a user access that pertains to multiple databases must be handled by several different database administrators in order to get the appropriate permissions, instead of being unified? ( ) Yes ( ) No

VIII. IDENTIFICATION OF YOUR ORGANIZATION'S USER-ROLES

Please fill in the appropriate answers for the roles listed and defined in the following table. Answer "Yes" or "No" according to your need or desire for the listed role. Assess the role's importance factor (= weight) where the weight is a number from 1 to 5 which is the importance of the specific role to your system, where the scale is:
1 = very unimportant, 2 = unimportant, 3 = so-so, 4 = important, 5 = very important.

Table - Desirability of Role-Based Access Control for Your Organization

Role	**Explanation**	**Yes/No**	**Weight**
Application- Independent Role	Allows role-based access controls to be inserted into a legacy system.		
Basic Role	Allows role-designers to define roles.		
Duty Role	Allows a user to perform a specific duty.		
Null Role	Allows a user to be logged onto the system but not have any roles that are active.		
Unknown Role	Allows a user to have a role active that is unknown to the RBAC application.		
Delegate-Role Single Level	Allows a first-user to authorize a second-user to enter a transaction the first-user is authorized to use.		
Delegate-Role Multiple Levels	Same as Single Level, but the second-user can repeat the process.		
Single-Role Hierarchy	Allows a role-designer to define a related set of roles within an application.		
Multiple-Role Hierarchies	Allows a role-designer to define more than one role hierarchy within an application.		

Collaborative Roles
(Constraint) Allows role-designers to establish a role where multiple users must be active in the role (or related role) to enter a transaction.

Concurrent Roles
(Constraint) Allows a user to have more than one role active at the same time.

Prerequisite Roles
(Constraint) Allows a role-designer to specify a role that must be active before the user can assume a second role.

Exclusive
[one user] (Constraint) Allows a role-designer to specify that no other user can have this active role while a particular user assumes this role.

Exclusive
[one role] (Constraint) Allows a role-designer to specify that a specific user can have no other active role while this role is active.

Mutually Exclusive (Constraint) Allows a role-designer to use exclusive roles that are mutually exclusive.

Mutually Exclusive Transaction Basis
(Constraint) Allows a role-designer to use mutually exclusive roles that are applied on a per transaction basis.

IX. OTHER SUGGESTIONS

Do you have any specific suggestions about role-based security controls?

____

____

____

We have been talking about a wide range of topics concerning application security controls. Is there anything else about role-based security controls you would like to add?

____

____

X. CONCLUSION OF INTERVIEW

Thank you very much for your cooperation. Your inputs are important to our study. If we have any further questions or need any clarification, may we call you? If you need to talk to us, call Charlie Smith at SETA (703-821-8178). Again, thank you very much for your time and the information that you have provided us.

Could you give us the name or names of other computer system security users who are knowledgeable about security and who you think we should contact for our study?

Name Organization Phone Number

Time at End of Questionnaire Process: _______________