A Persona Framework for Attribution, Delegation and Least Privilege[1]

By

Coimbatore S. Chandersekaran
William R. Simpson
Institute for Defense Analyses, 4850 Mark Center Dr.
Alexandria, Virginia 22311

[1] The publication of this paper does not indicate endorsement by the Department of Defense or IDA, nor should the contents be construed as reflecting the official position of these organizations.

ABSTRACT

There are many business needs for implementing delegation in IT systems. However, existing approaches to delegation in IT systems are limited in their usability, flexibility, and capability to implement least privilege. The result is that delegation is either not implemented or is implemented informally (e.g. by sharing credentials between users), resulting in serious security concerns and a lack of accountability and auditability. This paper describes a proposed framework for delegation based on the *persona* concept. A persona is a special category of user that embodies only delegated privileges, and which is explicitly assumed only after the "real" human user taking on that persona explicitly chooses it. This paper describes the persona delegation framework in the context of a large enclave-based architecture currently being implemented by the US Air Force. Benefits of the framework include increased flexibility to handle a number of different delegation business scenarios, decreased complexity of the solution, and greater accountability with only a modest amount of additional infrastructure required.

Keywords: Delegation, enterprise, information security, least privilege, attribution, information sharing.

THE NEED FOR DELEGATION

Delegation is the handing of a task over to another person, usually a subordinate. It is the assignment of authority and responsibility to another person to carry out specific activities. It allows a subordinate to make decisions, i.e., it is a shift of decision-making authority from one organizational level to a lower one. Delegation, if properly done, is not abdication. The opposite of effective delegation is micromanagement, where a manager provides too much input, direction, and review of 'delegated' work[2]. The need for delegation in IT systems often arises out of the need to manage time and prioritize an activity, establish a posture of least privilege, and/or provide for transitioning between assignments.

- Time management issues happen when a user has a tasking that requires careful consideration of time and activity investment. In an IT system it may take the form of an administrative assistant reading and screening e-mail, or a task group leader seeking information and options to be placed in the reading files of a decision maker.
- Least privilege issues occur when an individual is assigned two or more roles within the organization, with differing privilege sets. Ideally, we wish the user to only have access to the minimum set of privileges associated with the role they are currently acting as in the system.
- Transitioning issues occur when an overlap exists between new and old assignments that have different access and privilege, but both must be maintained for an overlap period.
- All aspects of a delegation cannot be foreseen, but current practice of giving away login details or letting someone else use an access card (e.g., in a US DoD context, a Common Access Card or CAC), or even generating multiple logins, are unacceptable from an attribution standpoint. Delegation must be formalized so that appropriate audit and forensics can be done when system anomalies occur, or compliance measurements concerning security policy is required.

Delegation in a Large Military Organization

In the context of a large military organization (such as the US Air Force), there are also additional complexities associated with delegation. For example, individuals can only be authorized to view documents and data no higher than the security clearance level they have been granted (e.g., Secret, Top Secret). These restrictions have to be enforced in addition to any restrictions associated with any other delegated privileges In addition, consider the case of military units that must rapidly deploy to a theater of engagement to replace another unit. Many delegation activities must take place during the transition period when both units overlap in the field.

PROPOSED ARCHITECTURE

In this paper we propose a solution that uses a created persona for the delegate that is activated through a delegation service. A persona is a special category of user that embodies only delegated privileges, and which is explicitly assumed only after the "real" human user taking on that persona explicitly chooses it. The existence of a persona delegation is flagged in the user file, and the logon script will include a call to the delegation service for revised identification of the user. The system opens a session with delegation credentials that are inherited for the individual providing the delegation. The delegation must be recorded and registered in advance through a delegation registration service, and the delegation must be approved by written policy. The delegate persona is the responsible for actions and attribution. Actions taken by the delegate persona are recorded by audit records that have the session number assigned and the delegate persona identity (id). The delegate persona is persistent, although it should have an expiration date at the end of which it is renewed or expires ("persona non grata"). The delegate persona can be retrieved as a delegate by query to the delegation data base. When a related persona is created, the attributes under the user are modified. The last entry is provided with "Delegate," as an indication for delegation services. This field may have a default of "Normal," and a created Persona may have a value "Persona."

[2] Definition adapted from Wikipedia.

ARCHITECTURAL DETAILS

Registration Service for Principal-Agent Delegation

Principal-Agent policies are promulgated by the appropriate authority. Such policies may apply to a large class of individuals (as in the pre-screening of e-mails by administrative assistants) or to a specific instance (as in the task group lead). The principal-agent delegation registration creates a user persona that links two individuals and the delegated authority. This process involves three branches of the Directory Information Tree (DIT). Figure 1 shows the delegation registration process. The delegation registration service is invoked and current policy is checked to see if User 2 can actually delegate. If User 2 can delegate by policy, then he is asked for the identification of the agent. If User 2 by policy can accept delegation then the registration authority creates the persona (user n), together with names and PKI and other credentials. In order for this service to work, the semantics of policy must be worked out by the Community of Interest (COI)[3]. It is expected that the policy elements will change from time to time, and the registration service should be able to read these from an input file.

At this point, the principal is offered groups that are allowed delegation. The latter is important because a number of rules will be invoked. In the absence of offered groups, the individual specified groups must be heavily screened for overall and specific policies (e.g., a principal cannot delegate privileges associated with his security clearances). Finally, the delegate persona (user n) is populated with access groups from the delegation and the agent's attributes. The delegate persona is persistent and appears in the DIT as any other user. User credentials associated with user n are the credentials associated with a new identity created by the registration service.

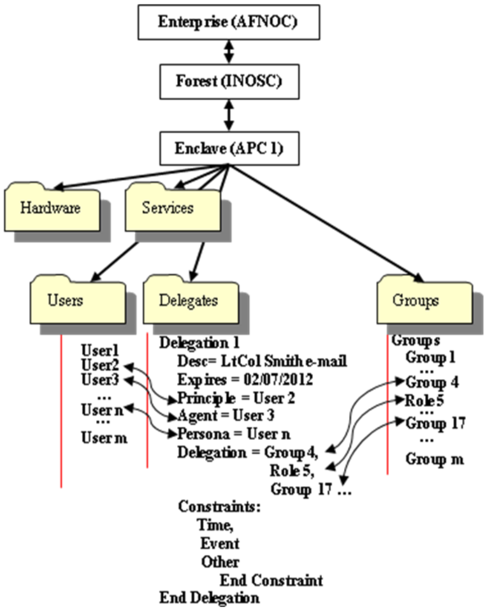

Figure 1 Principal-Agent Delegation Architecture

Least Privilege as a Principal-Principal Delegation

User Based Least Privilege[4]

In computer science and other fields, the principle of minimal privilege, also known as the principle of least privilege or just least privilege, requires that in a particular abstraction layer of a computing environment every module (such as a process, a user or a program on the basis of the layer we are considering) must be able to access only such information and resources that are necessary to its legitimate purpose. The principle of least privilege is widely recognized as an important design consideration in enhancing the protection of data and functionality from faults and malicious behavior.

In operating systems like Windows, there is no security enforcement for code running in kernel mode and therefore such code always runs with maximum privileges. The principle of least privilege therefore demands the use of user mode solutions when given the choice between a kernel mode and user mode solution if the two solutions provide the same results.

Registration Service for Principal-Principal Delegation

Principal-Principal policies are promulgated by the appropriate authority. Such policies may apply to a large class of individuals (as in the assignment of multiple roles) or to a specific instance (as in the task breakdown for the individual). The principal-principal delegation registration creates a user persona that links two instances of an individual and the delegated authorities (or roles in some instances). This process involves three branches of the (DIT). In Figure 2 we show the delegation registration process. The delegation registration service is invoked by either user 6 or the enclave[5] administrator on behalf of user 6 and current policy is checked to see if User 6 needs least-privilege delegation. If User 6 can delegate by policy, then he is asked for the identification of the roles or other descriptors for each self delegation including privileges associated with each. User 6 has three roles designated. The first is overall enclave administrator, the second is the COI data base manager, and the third is as a normal enclave user. Disjointness in roles will help insure that users carefully chose the role for each session. If roles are proper subsets of one another, then the maximum privilege is usually taken. This is an important principle for administration (make roles disjoint to the extent possible).

The registration authority creates the personae (user p, and q), together with names and PKI and other credentials. In order for this service to work, the semantics of self delegation must be worked out by the COI (this may be as simple as roles initially). The COI may wish to work out super groups, where a super group is a group of groups that can be used to represent a role, task, or other unique combination of authorities. It is expected that the policy elements will change from time to time, and the registration service should be able to read these from an input file. At this point, the principal or administrator is offered groups (or super groups) that are allowed in the defining of roles. The latter is important because a number

[3] COI are formal entities in the Air Force architecture.

[4] Definition adapted from Wikipedia.

[5] An enclave is defined as a set of capabilities realized by hardware, software, networks, devices, and people.

of rules will be invoked. In the absence of offered (super)groups, the individual specified groups must be heavily screened for overall and specific policy. Finally, the delegate personae (users p, and q) are populated with access groups from the delegation and the agent's attributes. The self-delegate persona is persistent and appears in the DIT as any other user. User credentials associated with user p and q are the credentials associated with the original identity in self-designation (user 6).

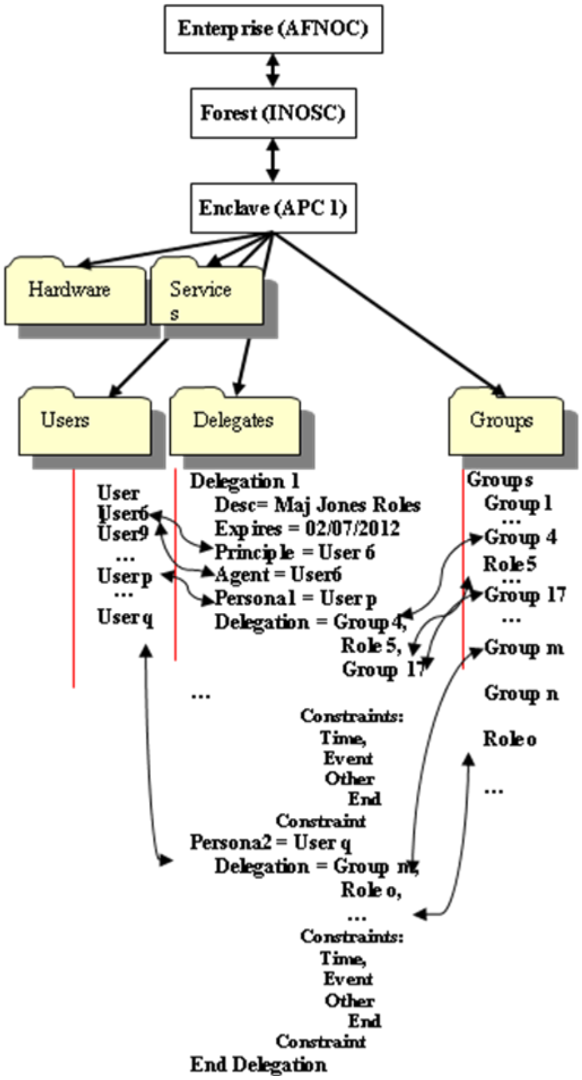

Figure 2 Pricipal-Principal Delegation Architecture

Registration Service for Admin-Principal Delegation

Admin-Principal policies are promulgated by the appropriate authority. Such policies may apply to a large class of individuals (as in the movement of a group of individuals between assignments) or to a specific instance (as in the movement of an individual between assignments). The admin-principal delegation registration creates a user persona for the old assignment with an appropriately short expiration and a second persona that is the new assignment of a longer expiration. This process involves three branches of the Directory Information Tree (DIT). Figure 3 shows the delegation registration process.

The delegation registration service is invoked and current policy is checked to see if User 8 can be provided two identities. The registration authority creates the persona (user z), together with names and PKI and other credentials associated with the old assignment. In order for this service to work, the semantics of policy must be worked out by the COI. It is expected that the policy elements will change from time to time, and the registration service should be able to read these from an input file. At this point, the administrator is offered groups that are allowed for the new assignment. The latter is important because a number of rules will be invoked. In the absence of offered groups, the individually specified groups must be heavily screened for overall and specific policy such as no delegation of clearances. Finally, the original user designation (user 8) is populated with access groups from the new assignment and the user's attributes. The new persona is temporary and appears in the DIT as any other user. User credentials associated with user z are the credentials associated with an old assignment and identity.

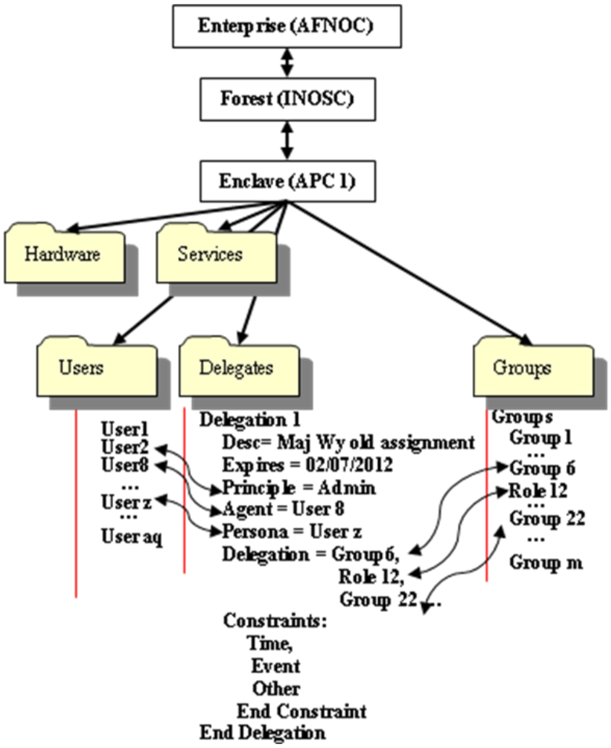

Figure 3 Admin-Principal Delegation Architecture

NAMING FOR PERSONA

Delegate personae will be named using naming criteria for users. The user will also be given an alias that appears early in the list of identity attributes. For Principal-Agent delegation this alias will be created as "OnBehalfof" added to the EDIPI of the principal. The first name under attributes will be given the "OnBehalfof" label and the last name will be the name of the principal. For other delegations the alias for persona will be the alias of the user using the persona.

Naming for Delegation Groups

It is recommended that delegation groups simply be named sequentially as shown in Figures 1-3. This will provide information hiding. Release of a delegation does not have to renumber the delegation groups.

DELEGATION INVOCATION SERVICE

As described above, no user has the authority to log in as the persona. In order for persona to be invoked, a user delegation service must be called. It is recommended that every user that has a delegation also have a flag in his/her file and the initial logon script calls the delegation service on his behalf. When a related persona is created, the attributes under the user are modified. The last entry is provided with "Delegate", as an indication for delegation services. This field may have a default of "Normal", and a created Persona may have a value "Persona". The user delegation service will examine the DIT delegation structure for the user and offer him/her the agencies recorded in the DIT. For example, User 3 may be an agent for User 2 with persona n and an agent for User 7 with persona m. Only one delegation may be made at a time. The delegation service will then change the user identity for the session to the appropriate persona for the balance of the session. Personas will not be authorized to invoke the delegation service so that no chaining of delegations is possible. Figure 4 shows the delegation invoking process. Once the delegation is invoked, the old user is replaced by the persona (or not, if no delegation is chosen) and all access to delegation mechanisms and the old user are broken. Each action is audited as discussed in the next section.

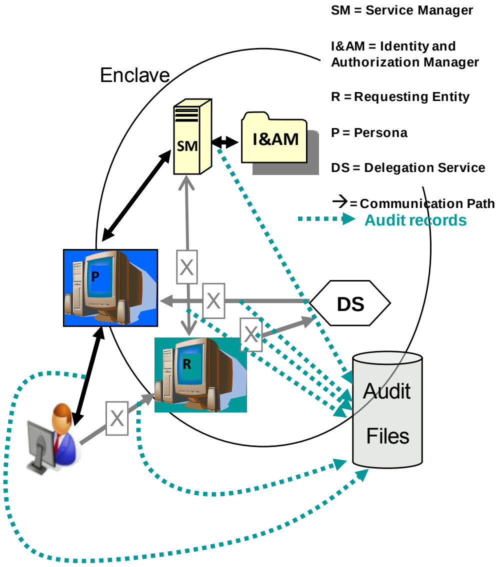

Figure 4 Delegation Invocation Process

THE IMPORTANCE OF AUDIT IN DELEGATION

There are many delegations that happen throughout a session. Most are done by impersonation (appearing to be another entity). Lower level (level 1-4) service-to-service delegations may be done by impersonation; however in every instance the session id is preserved. Tight logging must include session id so that an intrusion detection program, security analysis program, or an individual can obtain a trace of activity by session id. The session id is the tie to the invocation of delegation, which provides attribution. Audit files may reside within the enclave or elsewhere

DELEGATE PERSONA VULNERABILITIES

As with any vulnerability, the final implementation, including the code developed for services will determine vulnerabilities to the system. However, several vulnerability areas come to mind.

Spoofing

No user can login as a delegate. In order to spoof the delegate persona, the spoofer would have to be an insider, or have breached the system. Since delegation is registered, the spoofer would have to create his own persona by having access to the DIT. Activating the delegate persona is logged and attribution is assigned to the user who activated the delegation.

Elevation of Rights

Recursive calls to the delegation service are prohibited. Elevation of rights during creation of the delegate persona is prohibited. The intruder (insider or external) would first have to edit the persona which would require access to the DIT and knowledge of the delegate, or creation of a new delegate.

DELEGATION USE CASES AND SERVICES

Tables 1 and 2 list the key use cases that must be implemented to provide delegation registration and delegation invocation services. These capabilities may form one basis for developing new standards for delegation (e.g., a new WS-* standard). Table 3 identifies key services that must be built to support these use cases.

Table 1 Delegation Registration Use Cases

Function	User Role	Interface Notes
Invoke Registration authority	Invoke Service	User Identity Details and authorities
Identify Delegation Agent Principal-Agent Delegation	Any Potential Authorized User	Must be able to read delegation policy, and access DIT. Must screen delegation pair and limit choices.
Identify Delegation Agent Principal-Principle Delegation	Administrator	Must be able to read delegation policy, and access DIT. Must screen delegation pair and limit choices.
Identify Delegation Agent Admin-Agent Delegation	Administrator	Must be able to read delegation policy, and access DIT. Must screen delegation pair and limit choices.
Identify delegation attributes	Any Potential Authorized User	Probably a choice of attributes are presented that meet policy. Otherwise choices must be screened.

Function	User Role	Interface Notes
Release of Delegation	User identified as principal in one or more delegations	Presentation of choices for delegate deletion. Persona is removed from registry. Expiration is also a release of delegation.

Table 2 Delegation Invocation Use Cases

Function	User Role	Interface Notes
Invoke Delegation	Login script invokes Service	User Identity Details and authorities. Present delegations for the user that have been registered
Chose delegation for session	Any Potential Authorized User	Must be able to read delegation policy, and access DIT. Must redirect user to persona and break all links with prior user.
End Delegation	Any Persona	Terminate session only.

Table 3 Delegation Invocation Services Needed

Service	Level for Service	Other Services Needed
Set up Delegation Service	Admin	Provide rules and linkages to delegation services, update rules as policy changes.
Create Delegation	Any Potential Authorized User	User Identity Details and authorities. Present delegations for the user that have been registered
Delete Delegation	Any Principal for Principal-Agent delegations, others require admin authority	Must be able to read delegation policy, and access DIT. Must be able to eliminate persona.
Invoke Delegation	Any Potential user flagged in login script	Must be able to read delegation policy, and access DIT. Must redirect user to persona and break all links with prior user.

NOTES AND ASSUMPTIONS

The following assumptions about delegation are made

- The delegate persona is persistent, but with expiration dates so that it must be renewed. This reduces instances of unintended access to the system by unauthorized users.
- Only one delegation is allowed per session
- The only way to end delegation is to terminate the session. This simplifies the user experience and the implementation of delegation.
- Audit logging is verbose (every transaction of relevance is recorded) during delegation process.
- Session ID is a key element of every audit record. This enables the audit process to determine accountability, since session ID is tied to the persona.

CONCLUSION

We have presented a framework for improving delegation involving personas. This framework provides greater flexibility, usability, and accountability for the delegation process, with a minimum of additional infrastructure and services required. We are currently vetting this solution with the larger Air Force community, and believe that it has great promise for improving the practice of delegation and accountability throughout the enterprise.

REFERENCES

[1]. Liu, Ranganathan, Riabov, "Specifying and Enforcing High-Level Semantic Obligation Policies," *policy*, pp. 119-128, Eighth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'07), 2007

[2]. Jacques Wainer , Akhil Kumar , Paulo Barthelmess, DW-RBAC: A formal security model of delegation

[3]. and revocation in workflow systems, Information Systems, v.32 n.3, p.365-384, May, 2007

[4]. He Wang , Sylvia L. Osborn, Delegation in the role graph model, Proceedings of the eleventh ACM symposium on Access control models and technologies, June 07-09, 2006, Lake Tahoe, California, USA

[5]. James B. D. Joshi , Elisa Bertino, Fine-grained role-based delegation in presence of the hybrid role hierarchy, Proceedings of the eleventh ACM symposium on Access control models and technologies, June 07-09, 2006, Lake Tahoe, California, USA

[6]. Jacques Wainer , Akhil Kumar, A fine-grained, controllable, user-to-user delegation method in RBAC, Proceedings of the tenth ACM symposium on Access control models and technologies, June 01-03, 2005, Stockholm, Sweden

[7]. Roberto Tamassia , Danfeng Yao , William H. Winsborough, Role-based cascaded delegation, Proceedings of the ninth ACM symposium on Access control models and technologies, June 02-04, 2004, Yorktown Heights, New York, USA

[8]. Jacques Wainer, Paulo Barthelmess, and Akhil Kumar. WRBAC - a workflow security model incorporating controlled overriding of constraints. International Journal of Cooperative Information Systems, 12(4):455--486, 2003.

[9]. Walt Yao. Fidelis: A policy-driven trust management framework. In Trust Management, First International Conference, iTrust, volume 2692 of Lecture Notes in Computer Science, pages 301--317. Springer, 2003.

[10]. Longhua Zhang , Gail-Joon Ahn , Bei-Tseng Chu, A rule-based framework for role-based delegation and revocation, ACM Transactions on Information and System Security (TISSEC), v.6 n.3, p.404-441, August 2003

[11]. Xinwen Zhang , Sejong Oh , Ravi Sandhu, PBDM: a flexible delegation model in RBAC, Proceedings of the eighth ACM symposium on Access control models and technologies, June 02-03, 2003, Como, Italy

[12]. JongSoon Park, YoungLok Lee, HyungHyo Lee, and BongNam Noh. A role-based delegation model using role hierarchy supporting restricted permission inheritance. In Proceedings of the International Conference on Security and Management, SAM '03, pages 294--302. CSREA Press, 2003.

[13]. Chun Ruan , Vijay Varadharajan, Resolving Conflicts in Authorization Delegations, Proceedings of the 7th Australian Conference on Information Security and Privacy, p.271-285, July 03-05, 2002

[14]. Jean Bacon , Ken Moody , Walt Yao, A model of OASIS role-based access control and its support for active security, ACM Transactions on Information and System Security (TISSEC), v.5 n.4, p.492-540, November 2002

[15]. Vijayalakshmi Atluri , Avigdor Gal, An authorization model for temporal and derived data: securing information portals, ACM Transactions on Information and System Security (TISSEC), v.5 n.1, p.62-94, February 2002

[16]. Åsa Hagström , Sushil Jajodia , Francesco Parisi-Presicce, Duminda Wijesekera, Revocations-A Classification, Proceedings of the 14th IEEE Workshop on Computer Security Foundations, p.44, June 11-13, 2001

[17]. Evgeny Dantsin , Thomas Eiter , Georg Gottlob , Andrei Voronkov, Complexity and expressive power of logic programming, ACM Computing Surveys (CSUR), v.33 n.3, p.374-425, September 2001

[18]. E. Barka , R. Sandhu, Framework for role-based delegation models, Proceedings of the 16th Annual Ezedin S. Barka and Ravi Sandhu. A role-based delegation model and some extensions. In 23rd National Information Systems Security Conference, October 2000. http://csrc.nist.gov/nissc/2000/proceedings/papers/021.pdf.

[19]. Computer Security Applications Conference, p.168, December 11-15, 2000

[20]. Ravi Sandhu , Qamar Munawer, The ARBAC99 Model for Administration of Roles, Proceedings of the 15th Annual Computer Security Applications Conference, p.229, December 06-10, 1999

[21]. Cheh Goh , Adrian Baldwin, Towards a more complete model of role, Proceedings of the third ACM workshop on Role-based access control, p.55-62, October 22-23, 1998, Fairfax, Virginia, United States

[22]. Ravi S. Sandhu , Edward J. Coyne , Hal L. Feinstein , Charles E. Youman, Role-Based Access Control Models, Computer, v.29 n.2, p.38-47, February 1996

[23]. Ronald Fagin, On an authorization mechanism, ACM Transactions on Database Systems (TODS), v.3 n.3, p.310-319, Sept. 1978

[24]. Patricia P. Griffiths , Bradford W. Wade, An authorization mechanism for a relational database system, ACM Transactions on Database Systems (TODS), v.1 n.3, p.242-255, Sept. 1976

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number**. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE (DD-MM-YY) January 2010	2. REPORT TYPE Study	3. DATES COVERED (From – To)

4. TITLE AND SUBTITLE A Persona Framework for Attribution, Delegation and Least Privilege	5a. CONTRACT NUMBER DASW01-04-C-0003
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBERS
6. AUTHOR(S) Coimbatore S. Chandersekaran William R. Simpson	5d. PROJECT NUMBER
	5e. TASK NUMBER C5127
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESSES Institute for Defense Analyses 4850 Mark Center Drive Alexandria, VA 22311-1882	8. PERFORMING ORGANIZATION REPORT NUMBER IDA Nonstandard Document NS D-4171 Log no. 10-001007
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR'S / MONITOR'S ACRONYM IDA
	11. SPONSOR'S / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited: 16 February 2010.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

There are many business needs for implementing delegation in IT systems. However, existing approaches to delegation in IT systems are limited in their usability, flexibility, and capability to implement least privilege. The result is that delegation is either not implemented or is implemented informally (e.g. by sharing credentials between users), resulting in serious security concerns and a lack of accountability and auditability. This paper describes a proposed framework for delegation based on the *persona* concept. A persona is a special category of user that embodies only delegated privileges, and which is explicitly assumed only after the "real" human user taking on that persona explicitly chooses it. This paper describes the persona delegation framework in the context of a large enclave-based architecture currently being implemented by the US Air Force. Benefits of the framework include increased flexibility to handle a number of different delegation business scenarios, decreased complexity of the solution, and greater accountability with only a modest amount of additional infrastructure required.

15. SUBJECT TERMS

Delegation, Enterprise, Information Security, Least Privilege, Attribution, Information Sharing

16. SECURITY CLASSIFICATION OF: Unclassified			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Dr. William R. Simpson
a. REPORT Unclass	b. ABSTRACT Unclass	c. THIS PAGE Unclass	Unlimited	6	19b. TELEPHONE NUMBER (Include Area Code) (703) 845-6637