Computer Technology and Application 2 (2011) 413-423

Personal Delegation by Persona Creation

Coimbatore S. Chandersekaran and William R. Simpson

Institute for Defense Analyses, 4850 Mark Center Dr., Virginia 22311, USA

Received: April 28, 2011 / Accepted: May 25, 2011 / Published: June 25, 2011.

Abstract: There are many business needs for implementing delegation in IT (Information Technology) 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 [passwords or hardware tokens] between users), resulting in serious security concerns and a lack of accountability. This paper describes a methodology 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 a major enterprise. The creation of a persona solves a lot of downstream problems by allowing the persona to be treated like any other entity in the system. That is, identity, authentication, authorization, and other security processes already know how to handle an entity of this type. 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.

Key words: Delegation, enterprise, information security, least privilege, attribution, information sharing.

1. Introduction

Delegation has been treated as a policy problem [1-3], an aspect of access control [3-4], a role definition issue [5-14], a workflow issue [15-16], an authorization issue [17-18], and some hybrid approaches that combine these processes [19-22]. We have been unable to find an approach that treats the problem as an identity issue. In this paper, we treat the problem of delegation as an authorized identity issue. The delegated individual is assigned an identity for each assigned delegation. The paper is divided into ten major sections starting with this introduction.

Section 2 describes the need for delegation, including some specific cases for delegation activity.

Section 3 covers a proposed architecture based on the authorized identity process, including three types of identity based applications and the attendant data structures.

Section 4 indicates some additional uses of the methodology although it is difficult to foresee all of the applications.

Section 5 covers naming considerations. Naming is important because the process increases the number of identities in the enterprise.

Section 6 covers the necessary delegation invocation service. This service creates the identities and their attributes.

Section 7 covers the importance of auditing the delegation process and the log records required. Attribution is now multi-valued, including both the delegator and delegate.

Section 8 covers the vulnerabilities associated with the identity transfer process and possible mitigations for these vulnerabilities. Again, with an increase in the number of identities in the enterprise, it is important to revisit these.

Section 9 describes the use cases and required services to implement the delegation process.

Section 10 provides conclusions, including some advantages and disadvantages to the method.

Corresponding author: William R. Simpson, Ph.D., research field: internet size enterprise systems. E-mail: rsimpson@ida.org.

2. The Need for Delegation

Delegation is the handing of a task over to another person, usually a subordinate [23]. Delegation should not be confused with the authority and responsibility that goes with a title and position and are already baked into the process. For example, a payroll clerk is already expected to have the credentials to access and modify payroll information. 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 also allows for temporary assignment of duties during absences or incapacitation. 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.

The need for delegation in IT systems often arises out of the need to manage time and prioritize an activity, allow for an alternate way of doing business in unforeseen circumstances, establish a posture of least privilege, and/or provide for transitioning between assignments.

Ÿ 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.

Ÿ Alternate ways of doing business may include temporary assignments to cover personnel outages or experimentation with methods and processes.

Ÿ 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.

Ÿ 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.

Ÿ All aspects of a delegation cannot be foreseen, but current practice of giving away login details or letting someone else use an access card, 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 Organization:

In the context of a large organization (such as a large corporate enterprise), there are also additional complexities associated with delegation. For example, individuals can only be authorized to view documents and data no higher than the access level they have been granted (e.g., Company Confidential, Trade Secret). These restrictions have to be enforced in addition to any restrictions associated with any other delegated privileges In addition, consider the case of field operations of the Red Cross or FEMA that must rapidly deploy to a locality to stand up a field presence or to replace another unit. Many delegation activities must take place during the transition period when both units overlap in the field.

3. 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. A single real user may have many personae. 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 individual 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 an authorized 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”.

3.1 Architectural Details

The next few sections will cover three instances in which persona delegation can be used to accomplish the IT objectives of attribution and/or least privilege. The instances are

Ÿ Direct delegation of authority;

Ÿ Maintaining least privilege when multiple roles exit;

Ÿ Maintaining two or more separate roles during job function transitions.

3.2 Direct Delegation of Authority

Henry Smith (User 2) chooses to delegate the keeping of his time sheet to Henrietta Jones (User 3). In this case Henrietta is Henry’s administrative assistant and keeps his calendar and appointments, so it seems to make sense to Henry to delegate the time sheet business to Henrietta.

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 persona that links two individuals and the delegated authority. This process involves three branches of the Directory Information Tree (DIT). Fig. 1 shows the delegation registration process. The delegation registration service is invoked and current policy is checked to see if User 2—Henry can actually delegate (delegation of timekeeping may be prohibited by company policy, or government regulation, etc.). If User 2 can delegate by policy, then he is asked for the identification of the agent. In this case, the delegator (User2 or Henry) chooses Henrietta (User 3) as the delegate or agent. If User 3—Henrietta by policy can accept delegation (again policy may prohibit anyone working for the company less than two years from assuming timekeeping of another person) 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 administrators of the enterprise. Note that persona n is tagged as Henrietta Jones OnBehalfOf Henry Smith. 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 and interpret them in the context of delegation.

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 corporate 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, many of which are also Henrietta’s credentials. An exception would be created in the SAML [24-33] credentials which would have the persona ID and groups and roles in the attribute assertion.

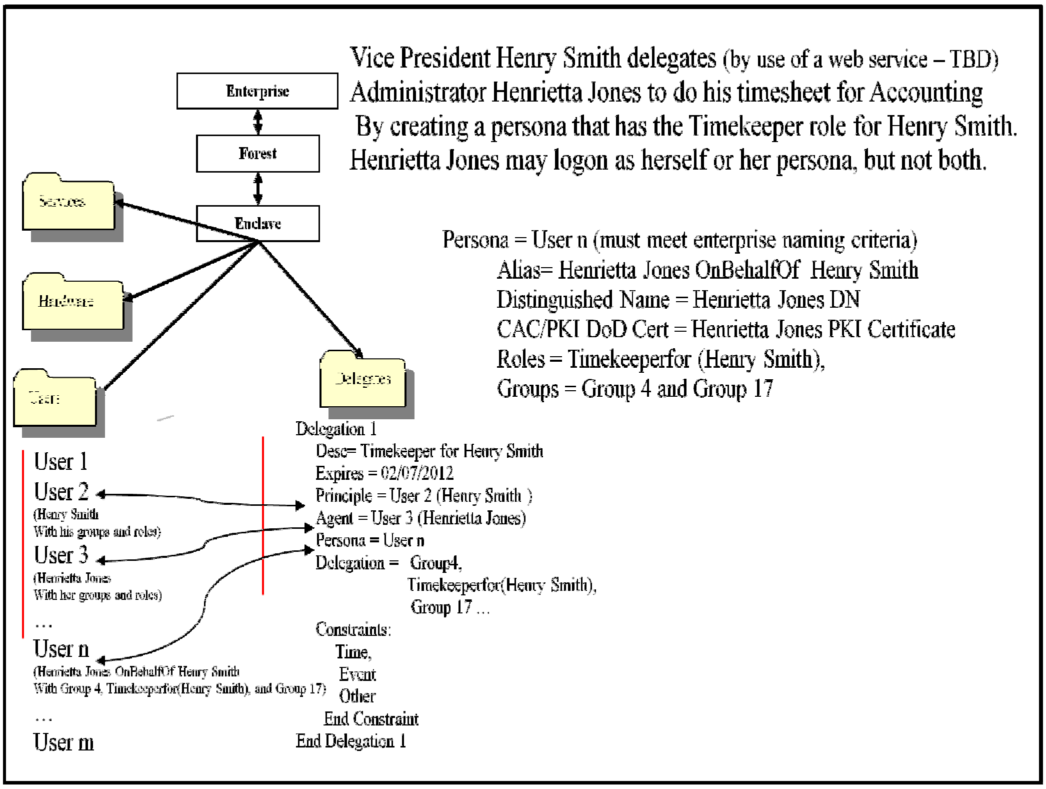

Vice President Henry Smith delegates (by use of a web service – TBD)
Administrator Henrietta Jones to do his timesheet for Accounting
By creating a persona that has the Timekeeper role for Henry Smith.
Henrietta Jones may logon as herself or her persona, but not both.
Enterprise
Forest
Enclave
Services
Hardware
Users
Delegates
Persona = User n (must meet enterprise naming criteria)
Alias= Henrietta Jones OnBehalfOf Henry Smith
Distinguished Name = Henrietta Jones DN
CAC/PKI DoD Cert = Henrietta Jones PKI Certificate
Roles = Timekeeperfor (Henry Smith),
Groups = Group 4 and Group 17
User 1
User 2
(Henry Smith
With his groups and roles)
User 3
(Henrietta Jones
With her groups and roles)
…
User n
(Henrietta Jones OnBehalfOf Henry Smith
With Group 4, Timekeeperfor(Henry Smith), and Group 17)
…
User m
Delegation 1
Desc= Timekeeper for Henry Smith
Expires = 02/07/2012
Principle = User 2 (Henry Smith)
Agent = User 3 (Henrietta Jones)
Persona = User n
Delegation = Group4,
Timekeeperfor(Henry Smith),
Group 17 …
Constraints:
Time,
Event
Other
End Constraint
End Delegation 1

Fig. 1 Principal-agent delegation.

3.3 Least Privilege as a Principal-Principal Delegation

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.

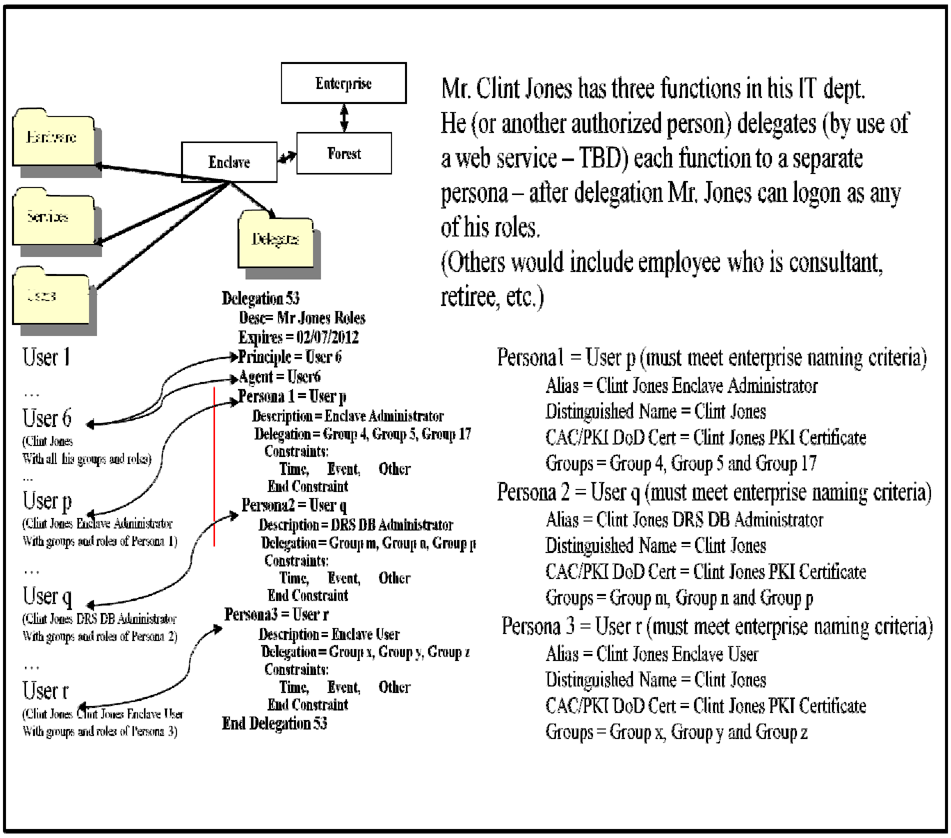
In operating systems like Windows, there is no security enforcement for code running in kernel mode and therefore such code always runs with maximum privileges. The principle of least privilege therefore demands the use of user mode solutions when given the choice between a kernel mode and user mode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

3.4 User Based Least Privilege [34]

Clint Jones has several assignments within the company, including enclave administrator, data base administrator and he is also just a user. He, and the company, would like to limit his authority, so that while checking hi e-mail, he does not accidently format a hard drive (which an enclave administrator is allowed to do).

Principal-Principal Delegation:

Principal-Principal policies are determined by the appropriate authority within the enterprise. 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 (groups and/or roles in some instances). This process involves three branches of the (DIT). In Fig. 2 we show the delegation registration


Enterprise
Forest
Enclave
Hardware
Services
Users
Delegates
User 1
...
User 6
(Clint Jones
With all his groups and roles)
...
User p
(Clint Jones Enclave Administrator
With groups and roles of Persona 1)
...
User q
(Clint Jones DRS DB Administrator
With groups and roles of Persona 2)
...
User r
(Clint Jones Clint Jones Enclave User
With groups and roles of Persona 3)

Mr. Clint Jones has three functions in his IT dept.
He (or another authorized person) delegates (by use of a web service – TBD) each function to a separate persona – after delegation Mr. Jones can logon as any of his roles.
(Others would include employee who is consultant, retiree, etc.)

```
Delegation 53
  Desc= Mr Jones Roles
  Expires = 02/07/2012
  Principle = User 6
  Agent = User6
  Persona 1 = User p
     Description = Enclave Administrator
     Delegation = Group 4, Group 5, Group 17
       Constraints:
         Time,    Event,    Other
       End Constraint
   Persona2 = User q
     Description = DRS DB Administrator
     Delegation = Group m, Group n, Group p
       Constraints:
         Time,    Event,    Other
       End Constraint
 Persona3 = User r
     Description = Enclave User
     Delegation = Group x, Group y, Group z
       Constraints:
         Time,    Event,    Other
       End Constraint
End Delegation 53
```

Persona1 = User p (must meet enterprise naming criteria)
Alias = Clint Jones Enclave Administrator
Distinguished Name = Clint Jones
CAC/PKI DoD Cert = Clint Jones PKI Certificate
Groups = Group 4, Group 5 and Group 17

Persona 2 = User q (must meet enterprise naming criteria)
Alias = Clint Jones DRS DB Administrator
Distinguished Name = Clint Jones
CAC/PKI DoD Cert = Clint Jones PKI Certificate
Groups = Group m, Group n and Group p

Persona 3 = User r (must meet enterprise naming criteria)
Alias = Clint Jones Enclave User
Distinguished Name = Clint Jones
CAC/PKI DoD Cert = Clint Jones PKI Certificate
Groups = Group x, Group y and Group z

Fig. 2 Principal-principal delegation.

process. The delegation registration service is invoked by either user 6—Clint Smith or the enclave [1] administrator on behalf of user 6—also Clint Smith in this example and current policy is checked to see if User 6 needs least-privilege delegation. Actually policy might prevent an enclave administrator from being a data base administrator. 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—Clint has three roles designated. The first is overall enclave administrator, the second is the 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, q and r), 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 enterprise administrators (this may be as simple as roles initially). The administrators 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 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 (user’s p, q and r) are populated with access groups

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

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, q and r are the credentials associated with the original identity in self-designation (user 6—Clint).

In our implementation, improved circular correlation algorithm output of a visible satellite is shown in Fig. 4.

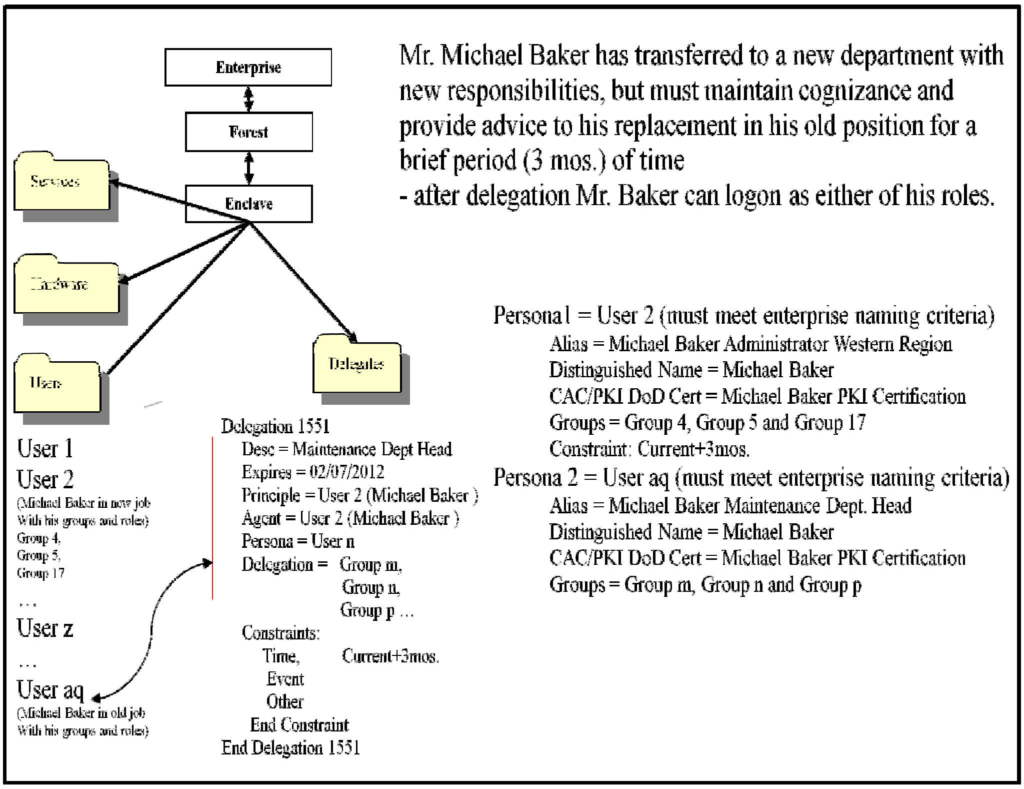
And the execution time of the method of acquiring one satellite is 0.25 s. The improved algorithm acquiring 32 satellites needs 7 seconds. In comparison with the traditional algorithm, the time of acquisition decreases from 0.47 s to 0.25 s.

3.5 Overlapping Assignments

Mr. Michael Baker is moving up in the organization. He has transferred to a new department with new responsibilities, but must maintain cognizance and provide advice to his replacement in his old position for a brief period (3 mos.) of time. He would like to maintain his old IT privileges for a time (3 mos.), but needs to get on with the IT privileges associated with his new job.

Admin-Principal Delegation:

Admin-Principal policies are determined by the appropriate authority within the enterprise. 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 and stored in the usual identity of the individual. This process involves three branches of the Directory Information Tree (DIT). Fig. 3 shows the delegation registration process. The delegation registration service is invoked and current policy is checked to see if User 2—Michael Baker can be provided these two identities. There may actually be federal regulations or company policies that prohibit the mixing of these two jobs. If User 2 can be provided these two identities, the registration authority creates



Mr. Michael Baker has transferred to a new department with new responsibilities, but must maintain cognizance and provide advice to his replacement in his old position for a brief period (3 mos.) of time
- after delegation Mr. Baker can logon as either of his roles.

Delegation 1551
Desc = Maintenance Dept Head
Expires = 02/07/2012
Principle = User 2 (Michael Baker)
Agent = User 2 (Michael Baker)
Persona = User n
Delegation = Group m,
Group n,
Group p …
Constraints:
Time, Current+3mos.
Event
Other
End Constraint
End Delegation 1551

Persona1 = User 2 (must meet enterprise naming criteria)
Alias = Michael Baker Administrator Western Region
Distinguished Name = Michael Baker
CAC/PKI DoD Cert = Michael Baker PKI Certification
Groups = Group 4, Group 5 and Group 17
Constraint: Current+3mos.

Persona 2 = User aq (must meet enterprise naming criteria)
Alias = Michael Baker Maintenance Dept. Head
Distinguished Name = Michael Baker
CAC/PKI DoD Cert = Michael Baker PKI Certification
Groups = Group m, Group n and Group p

Fig. 3 Admin-principal delegation.

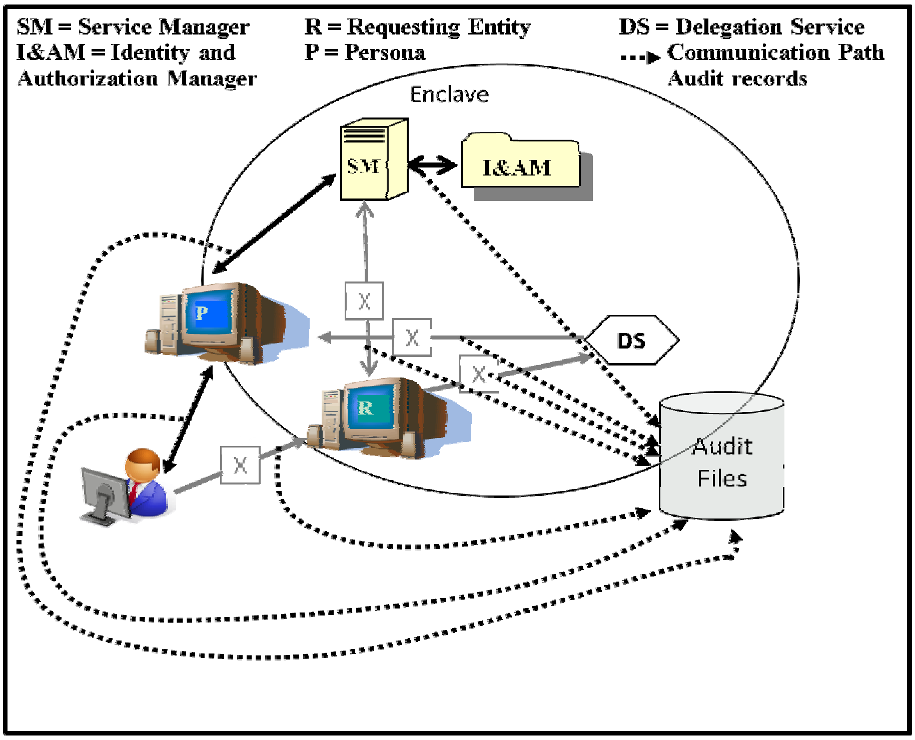


Fig. 4 Delegation invocation process.

the persona (user aq), 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 administrators of the enterprise. 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 corporate clearances. Finally, the original user designation (User 2—Michael Baker) is populated with access groups from the new assignment and the user’s attributes. The new persona is permanent and appears in the DIT as any other user. User credentials associated with user aq are the credentials associated with an old assignment and the identity of Michael Baker.

4. Additional Uses of Personae

It is impossible to foresee all uses at this time, but one that is being actively explored is the use of persona for attribution of virtual machines. Virtualization offers some particular challenges to the high assurance paradigm by implementing multiple entities under one identity, redirection of communication and insertion of surrogates in the process, all of which cause a loss of attribution. The persona mechanism would need to be created at the time of virtual machine implementation and implemented each time a virtual instance of a machine or service is created.

5. Naming for a 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 Common or Distinguished name 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 Figs. 1-3. This will provide information hiding.

6. 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. Fig. 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.

7. 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 (levels 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.

8. 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.

8.1 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. This degree of access by malicious entities has far greater implications than spoofed persona. Activating the delegate persona is logged and attribution is assigned to the user who activated the delegation.

8.2 Elevation of Rights

This is a common step in the attack progression. The key is to not let the delegation process be the key to rights elevation. 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. This degree of access by malicious entities has far greater implications than rights elevation of persona.

9. Delegation Use Cases and Services

Tables 1-2 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.

Notes and Assumptions:

The following assumptions about delegation are made:

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	Read delegation policy, and access DIT Screen delegation pair and limit choices
Identify delegation agent principal-principal delegation	Administrator	Read delegation policy, and access DIT Screen delegation pair and limit choices
Identify delegation agent Admin-agent delegation	Administrator	Read delegation policy, and access DIT Screen delegation pair and limit choices
Identify delegation attributes	Any potential authorized user	Probably choices of attributes are presented that meet policy. Otherwise choices must be screened.
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 registration authority	Invoke service	User identity details and authorities
Identify delegation agent principal-agent delegation	Any potential authorized user	Read delegation policy, and access DIT. Screen delegation pair and limit choices.
Identify delegation agent principal-principal delegation	Administrator	Read delegation policy, and access DIT. Screen delegation pair and limit choices.
Identify delegation agent admin-agent delegation	Administrator	Read delegation policy, and access DIT. Screen delegation pair and limit choices.

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	Read delegation policy, and access DIT; Eliminate persona
Invoke delegation	Any potential user flagged in login script	Read delegation policy, and access DIT; Redirect user to persona and break all links with prior user

Ÿ 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;

Ÿ Only one persona (of the personas attached to the identification of an entity) is allowed per session;

Ÿ The only way to end delegation is to terminate the session. This simplifies the user experience and the implementation of delegation;

Ÿ Audit logging is verbose during delegation process;

Ÿ 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.

10. Conclusions

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. The Persona approach offer both advantages and disadvantages.

(1) Advantages

Ÿ All infrastructure software (authN, authZ, SAML, logging, etc. will work with persona as with any active entity);

Ÿ Roles and Other Access Control data (groups) do not need to be formally defined to the lowest level;

a) They can be defined to an intermediate level and delegated from there;

b) Makes provisioning simpler with a delegation process controlling downstream access control;

Ÿ Flexibility and Usability of the delegation process is demonstrated with additional uses being explored;

Ÿ Tracking and accountability are improved dramatically over current delegation processes.

(2) Disadvantages

Ÿ Need a delegation service and a delegation data structure;

Ÿ There is a minimal latency at logon, but this should not persist through the session;

Ÿ Modest additional infrastructure needed.

References

[1] Z. Liu, A. Ranganathan, A. Riabov, Specifying and enforcing high-level semantic obligation policies, in: 8th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'07), 2007, pp. 119-128.

[2] W. Yao, Fidelis: a policy-driven trust management framework, Lecture Notes in Computer Science 2692 (2003) 301-317.

[3] H. Wang, S.L. Osborn, Delegation in the role graph model, in: Proceedings of the 11th ACM Symposium on Access Control Models and Technologies, Lake Tahoe, California, USA, 2006.

[4] L.H. Zhang, G.J. Ahn, B.T. Chu, A rule-based framework for role-based delegation and revocation, ACM Transactions on Information and System Security (TISSEC) 6 (2003) 404-441.

[5] J.B.D Joshi, E. Bertino, Fine-grained role-based delegation in presence of the hybrid role hierarchy, in: Proceedings of the Eleventh ACM Symposium on Access Control Models and Technologies, Lake Tahoe, California, USA, 2006.

[6] J. Wainer, A. Kumar, A fine-grained, controllable, user-to-user delegation method in RBAC, in: Proceedings of the 10th ACM Symposium on Access Control Models and Technologies, Stockholm, Sweden, 2005.

[7] R. Tamassia, D. Yao, W.H. Winsborough, Role-based cascaded delegation, in: Proceedings of the 9th ACM Symposium on Access Control Models and Technologies, Yorktown Heights, New York, USA, 2004.

[8] X. Zhang, S. Oh, R. Sandhu, PBDM: a flexible delegation model in RBAC, in: Proceedings of the 8th ACM Symposium on Access Control Models and Technologies, Como, Italy, 2003.

[9] J.S. Park, Y.L. Lee, H.H. Lee, B.N. 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, 2003, pp. 294-302.

[10] J. Bacon , K. Moody , W. Yao, A model of OASIS role-based access control and its support for active security, ACM Transactions on Information and System Security (TISSEC) 5 (2002) 492-540.

[11] E. Barka, R. Sandhu, A role-based delegation model and some extensions, in: 23rd National Information Systems Security Conference, October, 2000.

[12] R. Sandhu, Q. Munawer, The ARBAC99 model for administration of roles, in: Proceedings of the 15th Annual Computer Security Applications Conference, December 06-10, 1999, p. 229.

[13] C. Goh, A. Baldwin, Towards a more complete model of role, in: Proceedings of the 3rd ACM Workshop on Role-Based Access Control, Fairfax, Virginia, United States, 1998, pp.55-62.

[14] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, C.E. Youman, Role-based access control models, Computer 29 (1996) 38-47.

[15] J. Wainer, P. Barthelmess, A. Kumar, WRBAC: a workflow security model incorporating controlled overriding of constraints, International Journal of Cooperative Information Systems 12 (2003) 455-486.

[16] J. Wainer, A. Kumar, P. Barthelmess, DW-RBAC: a formal security model of delegation and revocation in workflow systems, Information Systems 32 (2007) 365-384.

[17] C. Ruan, V. Varadharajan, Resolving conflicts in authorization delegations, in: Proceedings of the 7th Australian Conference on Information Security and Privacy, 2002, pp. 271-285.

[18] V. Atluri, A. Gal, An authorization model for temporal and derived data: securing information portals, ACM Transactions on Information and System Security (TISSEC) 5 (2002) 62-94.

[19] Å. Hagström, S. Jajodia, P. Francesco, D. Wijesekera, Revocations—a classification, in: Proceedings of the 14th IEEE Workshop on Computer Security Foundations, 2001, p. 44.

[20] E. Dantsin, T. Eiter, G. Gottlob, A. Voronkov, Complexity and expressive power of logic programming, ACM Computing Surveys (CSUR) 33 (2001) 374-425.

[21] R. Fagin, On an authorization mechanism, ACM Transactions on Database Systems (TODS) 3 (1978) 310-319.

[22] P.P. Griffiths , B.W. Wade, An authorization mechanism for a relational database system, ACM Transactions on Database Systems (TODS) 1 (1976) 242-255.

[23] Multiple contributors, Delegation, from Wikipedia, the Free Encyclopedia, available online at: http://en.wikipedia.org/wiki/Delegation, 2011.

[24] N. Ragouzis et al., Security Assertion Markup Language (SAML) V2.0 Technical Overview, OASIS Committee Draft, available online at: http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf, 2008.

[25] P. Madsen et al., SAML V2.0 Executive Overview, OASIS Committee Draft, available online at: http://www.oasis-open.org/committees/download.php/13525/sstc-saml-exec-overview-2.0-cd-01-2col.pdf, 2005.

[26] P. Mishra et al., Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, available online at: http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf, 2005.

[27] S. Cantor et al., Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, available online at: http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf, 2005.

[28] S. Cantor et al., Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, available online at: http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf, 2005.

[29] S. Cantor, et al., Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, available online at: http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf, 2005.

[30] S. Cantor, et al., Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, available online at: http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf, 2005.

[31] J. Kemp, et al., Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, available online at: http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf, 2005.

[32] F. Hirsch, et al., Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, available online at: http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf, 2005.

[33] J. Hodges, et al., Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, available online at: http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf, 2005.

[34] Principle of_Least_Privilege, from Wikipedia, the Free Encyclopedia, available online at: http://en.wikipedia.org/wiki/Principleof_least_privilege, 2011.