Autonomic Privilege Management - Extending PERMIS to Contribute to the TrustCoM Objectives

David CHADWICK
University of Salford, The Crescent, Salford, M5 4WT, UK
Tel: +44 7796 44 7184, Fax: + 44 161 295 5351, Email: d.w.chadwick@salford.ac.uk

Abstract: This paper briefly describes the existing PERMIS privilege management infrastructure (PMI), the new TrustCoM European Integrated Project, and autonomic security. It then provides the business case for an autonomic PMI, and looks at the issues that will need to be resolved in order to make PERMIS more autonomic. In addition, it addresses the issues that TrustCoM will need to solve in order to maximise its use of an autonomic PERMIS.

1. Introduction

1.1 PERMIS

PERMIS [1] is a standard's based Privilege Management Infrastructure (PMI), built according to the ISO 10181-3 Access Control Framework [2] and X.509 [3] standards. It utilises X.509 Attribute Certificates (ACs) [3] to store user privileges (roles), and XML data structures to hold the authorisation policy. PERMIS makes its decisions by being passed the name of the subject, the name of the target, the action the subject wants to perform, and optional environmental parameters such as time of day. PERMIS then fetches the roles of the subject, which are typically stored as X.509 ACs in LDAP directories, to use in its decision making process. PERMIS implements the NIST role based access control (RBAC) model [4], and its XML policy states which roles subjects must have in order to perform which actions (e.g. project managers can update project files and team members can read project files). Because PERMIS supports hierarchical RBAC, then superior roles inherit the privileges of subordinate roles, for example, project managers could inherit the ability to read project files from team members. Optionally, additional conditions can be placed on access permissions. These conditions must be fulfilled in order for the role to be granted access, for example, grant access if the time of day is between 9am and 5pm.

A recently completed development project, in collaboration with Argonne National Laboratory, USA, has been to integrate PERMIS with Globus Toolkit v3 [5], using the GGF specified SAML interface [6]. This will allow Grid applications, often used by virtual organisations (VOs), to seamlessly use the PERMIS RBAC infrastructure to control which users, from which organisation, are authorised to do what.

1.2. TrustCoM

The TrustCoM Integrated Project [7] aims to develop a framework for trust, security and contract management in dynamically evolving VOs. This will enable secure collaborative business processing within on-demand created and self-managed, dynamic collaborative networks of organisations built on top of the emerging convergence of Web Services, agent

and Grid technologies. This will allow the VOs to tackle collaborative projects that their participants could not undertake individually, and to collectively offer enhanced services to customers that could not be provided by any of the individual enterprises.

Innovations expected from TrustCoM include: mechanisms for establishing of trust, autonomic security, electronic contract management and business process enactment. PERMIS is one of the many middleware components and services that will be needed by TrustCoM in order to realise its vision. However, PERMIS will need to become more autonomic, so that the overhead of managing this security component of the trust, security and contract management framework will be lessened.

1.3. Autonomic Security

Autonomic computing has been defined by IBM as an approach to self-managed computing systems with a minimum of human interference. The term derives from the body's autonomic nervous system, which controls key functions without conscious awareness or involvement [8]. According to IBM, there are eight crucial elements in an autonomic computing system [9]: it must maintain comprehensive and specific knowledge about all its components; it must have the ability to self-configure to suit varying and possibly unpredictable conditions; it must constantly monitor itself for optimal functioning; it must be self-healing and able to find alternate ways to function when it encounters problems; it must be able to detect threats and protect itself from them; it must be able to adapt to environmental conditions; it must be based on open standards rather than proprietary technologies; and it must anticipate demand while remaining transparent to the user.

An autonomic security system will therefore have knowledge about all of the components that make up its trusted computing base. It will know which parts of the overall system can be trusted and which parts cannot. It will be able to monitor its environment and reconfigure itself with different policies and components according to the current environment and existing threats. It will be constructed from standard’s conformant components so that reconfiguring is possible.

PERMIS is a standard’s conformant PMI that can already make access control decisions based on the environmental conditions, so it forms a solid basis from which to build an autonomic authorisation system. What additional features will a fully autonomic self-managing, self-healing, self-configuring, self-monitoring, threat-detecting, adaptable PERMIS contain? And how will these relate to the trust and contract management infrastructure that is to be provided by TrustCoM? These are the issues that are to be (at least partially) addressed in this paper.

The rest of this paper is organised as follows. Section 2 provides the business case for an autonomic PMI such as PERMIS, its industrial significance and potential benefits. Section 3 describes the existing autonomic features of PERMIS that already contribute to the overall TrustCoM vision. Section 4 describes additional features that are required to make PERMIS more autonomic, and that we plan to provide during the TrustCoM project. Section 5 describes how it is proposed to integrate PERMIS into the TrustCoM trust management infrastructure. Finally section 6 provides a conclusion and summarises where extensions to PERMIS are needed in order to fulfil the TrustCoM objectives.

2. The Business Case for an Autonomic PMI

Attribute certificates and Privilege Management Infrastructures have been researched for more than a decade. The SESAME project [12], co-funded by the EC RACE programme in the early 1990s, was a notable milestone, and first defined privilege attribute certificates. PMIs have a significant number of advantages over traditional access control lists. These include: reduced management costs, delegation of authority, scalability, and homogeneous

access control policy enforcement over a whole domain of resources. The latter ensures that "holes" do not exist in the access controls, which can often be the case with complex access control lists. PMIs allow managers who control resources to set the access control policy for the resource, while other managers are responsible for allocating user privileges. This significantly reduces the administrative workload of managers, and allows those who are know the users, to set their privileges in signed attribute certificates. Delegation of authority allows these managers to delegate their rights to subordinates, further simplifying the allocation of user privileges. Such a system is infinitely more scalable than requiring an administrator to set the access control list for all users of a resource, who must be known in advance. Policy based controls do not require this. Any user who has the necessary privilege attribute certificate will be granted access by the policy. Now that Internet use has become widespread, supporting millions of users, policy based controls become essential. Further, where short term business relationship (VOs) are continually being formed and re-formed, we need an agile system that can instantly be reconfigured to cater for the new environment. Re-setting the access control policy of a PMI is one way of achieving this.

Despite their advantages, PMIs have not yet made their way into mainstream commercial security infrastructures. Why is this? Part of the reason is that companies in the past did not have the need for such a sophisticated solution, since access controls list were quite adequate for their requirements. The cost of migration to a PMI could not be justified. But now, with the creation of virtual organisations, where resource owners wont even know all of the VO members, setting access control lists is not viable. Even if it were, they would be too time consuming and expensive to maintain. Consequently, it makes much more sense to migrate to a PMI, and to delegate the allocation of privileges to the managers in the collaborating organisations. An autonomic PMI further increases the business benefits, since it will be able to react to its environment and reconfigure itself, whilst simultaneously reducing the overall management costs.

3. Existing Autonomic Features of PERMIS

The existing PERMIS access control decision function is policy controlled, and thus PERMIS is already capable of making decision according to the prevailing environmental conditions. The PERMIS policy can contain rules, which state that access may only be granted if certain environmental parameters fall within preset values. Time of day is the most common example of this, but environmental parameters can also refer to such things as: a user's creditworthiness, a user's bank balance, a user's calling IP address etc. The implementer will need to define these environmental parameters, e.g. say what data type they are (integer, string etc) and configure the interface to pass the actual values across to PERMIS at decision time. PERMIS will then compare the actual values against those in the policy to determine if access should be granted.

PERMIS comes pre-configured with a set of operators (e.g. EQ, GT, GE, LE, LT etc.) with which to make the comparisons. It also has a plug-in capability that allows the implementer to add new operators (e.g. logically equivalent to $\Leftrightarrow$). The Java object performing this new operation is called by the PERMIS decision engine at the appropriate time. This allows PERMIS to be adapted to changing environments and applications.

PERMIS also has the ability to continue to make access control decisions when things start to fail. For example, if a denial of service attack is under way, which prevents PERMIS from accessing a user's X.509 ACs, then PERMIS will revert to granting default access rights to the user, rather than denying the user access altogether. At the moment this functionality is rather basic, and requires the user to actually ask for services that are protected by default rights whilst (s)he is denied access to more protected services, but a future version might be expected to return the default services (as hints) if a more protected service is asked for and denied.

PERMIS can already be configured with knowledge about all its components. These comprise: the targets that it is protecting, the subjects who may be granted access, its roots of trust, and the locations of the various LDAP repositories where user attribute certificates (ACs) and attribute certificate revocation lists (ACRLs) may be obtained. It has a well defined trust model that allows it to compute its trusted computing base.

Subjects and targets are specified as domains, so that if new ones spontaneously arise within a known domain, they will already be catered for and access control decisions can be made for them. If new subjects arise from unknown domains, then they will be denied access (actually an exception is thrown). If known subjects attempt to access targets in unknown domains, they are also denied access (again, an exception is thrown). We intend in a future version to categorise different exception classes so that the processing of different exceptions classes can be handled automatically by the calling application.

Different types of targets within the same naming domain can also be selected e.g. all printers at the University of Salford, or all cluster servers at Rutherford Laboratory. This allows sites to reconfigure, add to, and subtract from their resources without having to change the access control policy for the domain.

PERMIS recognises two different types of root of trust, namely the authorisation root of trust and the PKI root of trust. The PKI root of trust is the root CA or CAs trusted to validate all the X.509 ACs that PERMIS encounters. The actual specification of the PKI root or roots of trust is left up to the configured PKI to determine, and as such is viewed as out of scope of the PERMIS policy. In other words, the configured PKI is trusted to manage itself. The authorisation root of trust configured into PERMIS at initialisation time, is the name of the manager who sets the PERMIS authorisation policy, and it is (s)he who is ultimately responsible for defining who is allowed access to which resources controlled by the policy. PERMIS allows the manager to delegate trust down to authorised subordinates, who then become part of the trusted computing base. Delegation is however tightly controlled, in that the policy states what privileges the authorised subordinates are allowed to assign to which subjects. This delegation of authority is currently static, in that the policy must be changed if new subordinates are to be authorised. A more autonomic PERMIS will allow dynamic delegation of authority to take place, and this is needed to support dynamically evolving VOs. Consequently dynamic delegation of authority is one essential feature that needs to be added during the lifetime of the TrustCoM project.

LDAP repositories are modelled as a Java subclass of a more generic repository object class, so that in future different types of repositories can be added e.g. web pages (similar to the VeriSign certificate store accessible via the HTTP protocol), file stores etc. In addition, new LDAP repositories can be dynamically made known to PERMIS at decision time, which allows the system to respond to a changing environment without reconfiguration.

4. Future Autonomic Features of PERMIS

However, PERMIS's current knowledge about and use of repositories is rather basic. They either exist or they do not, and if they exist they will be contacted. PERMIS has no ability to be configured with alternate types of repositories, backup repositories, or the detailed characteristics of its repositories, such as their performance, throughput, hours of operation etc. It has no ability to choose between different repositories according to the prevailing conditions. These are future enhancements that will need to be added in order to make PERMIS more autonomic.

Other areas in which PERMIS needs enhancements, in order to lessen the management load, are in the areas of self-healing and threat detection and protection.

PERMIS currently has the ability to be configured with different policies for different occasions e.g. a red policy for times of high alert and a green policy for low risk periods. But human intervention is required to tell PERMIS when to switch between policies. A

more autonomic PERMIS will be able to automatically switch between policies when it detects threats/incidents to itself or to the environment. For example, if PERMIS detects that ACRL repositories that should be available are not available, it might decide to switch to a higher security policy, or if unusually many access requests are being denied, this could indicate an attack. If PERMIS detects that an attacker is trying to gain unauthorised access to resources, it could reduce the attacker's existing access rights, or notify a manager, or both. When PERMIS is linked to the trust infrastructure (see later), if it detects a user has become less trustworthy it could reduce their existing access rights, or conversely, if a user becomes more trustworthy they could gain access rights.

More sophisticated decision-making is required in order to enforce the mutual exclusivity of roles. Mutual exclusivity mandates that a user be not allowed to hold conflicting roles, for example, buyer and seller, or examiner and pupil. In the context of TrustCoM, mutual exclusivity could refer to membership of competing VOs that access common resources. If a user does posses conflicting roles, then they should be denied access to all actions that require any of the conflicting roles. Implementation of this feature will require an upgrade to the PERMIS policy and decision engine, but also may require knowledge of role memberships to be remembered by the decision engine.

Several of the more advanced autonomic features require PERMIS to keep an historical record of past access attempts and role memberships etc. in order to inform future access decisions. This feature is referred to as Retained ADI in [2]. In order to implement Retained ADI, PERMIS will need to have access to a secure transaction database, in which all past access decisions are logged. Fast searching of this database will be essential if the authorisation decision-making is not to be unduly slowed down as a result. We intend to implement this feature during the lifetime of the TrustCoM project.

5. Linking PERMIS to the Trust Infrastructure

The TrustCoM framework will establish a trust infrastructure, in which entities reputations are computed and shared between all the entities in VOs. As entities participate reliably in contracts, one might expect their reputations to increase, and conversely, if they default on contracts, one might expect their reputations to diminish. By adding a new condition to the PERMIS policy, namely the trustworthiness (or reputation) of a subject, the policy will be able to state that subjects are only allowed access to perform certain actions on certain targets if their trustworthiness is greater or equal to a particular value, and if not, that they are only to be granted lower access rights. (We note that a subject's inherent trustworthiness is manifest through interactions with the subject, but that it is the subject's reputation that is transferred between third parties and is thus measured. We use the terms interchangeably in the discussion here.) Below a certain trustworthiness, a subject could be denied access completely. There are however particular difficulties with this model, which have to be overcome, before it can be fully implemented in PERMIS.

Firstly, trustworthiness or reputation is linked to entities (individual, organisations, trusted third parties etc.) but the PERMIS infrastructure currently makes access control decisions based on the roles of the entity. Just because one role holder acts in an untrustworthy manner does not mean that all role holders should be trusted less. Therefore PERMIS will need to be modified to include the identity of the subject in its decision making process (which means that the decision making process will no longer be pure RBAC). In the TrustCoM project we will determine the best way of achieving this. One possible solution is that the conditional IF statement supported by PERMIS could be enhanced to include the identity of the subject, for example, IF trustworthiness of <subject> GE <some value> THEN grant access.

Secondly there is the issue of apportioning responsibility in order to determine who is to be held accountable when a contract is breached in some respect, and as a consequence,

whose reputation should be decremented. We need to clearly distinguish between the various actors such as employees, role issuing authorities and organisations. When an employee, assigned roles by an Attribute Authority (AA), participates in a contract on behalf of their organisation, and fails to fulfil the obligations of the contract, is it the organisation that is less trustworthy as a result, or the employee acting on behalf of the organisation, or the AA that issued privileges/roles to the individual? At what point should the organisation audit the actions of its employees and make recompense? When should an AA revoke the privileges/roles of the holders who abuse them? When should all three entities be held accountable in some way and all three have their reputations decreased? These are some of the issues that the TrustCom consortium must resolve since the outcomes will affect the trust information feeding into the decision making process of PERMIS.

Thirdly, the TrustCoM framework assumes that trustworthiness or reputation can be measured and given a value to be used in comparison operations. The metrics of trustworthiness are something that will need to be developed within the TrustCoM project.

Fourthly we need to consider the trustworthiness of the reputation broker that issued the reputation statement about the subject. What if the reputation broker itself is not that trustworthy? We thus have a recursive problem, in that in order to determine the trustworthiness of a subject we need to first determine the trustworthiness of the reputation broker that issues reputation statements about subjects. This recursive process can only stop with the trustors themselves, who ultimately will have to decide how much they trust certain reputation brokers or trusted third parties that act as trust anchors or roots of trust in the trust computing chain. This is sometimes termed plausibility considerations in the literature [10]. An interesting alternative scenario has recently been proposed by Philipp Obreiter [10] who suggests that rather than having plausibility considerations about reputation brokers, instead the transacting parties themselves could provide digitally signed evidences that show they have participated faithfully in transactions. We thus no longer need to trust the reputation brokers themselves. We only need to trust the roots of trust in the PKI (i.e. the Certification Authorities) that provided identity certificates to the transacting parties (to ensure that masquerade is not taking place) and the parties providing the evidences to ensure that collusion is not taking place (i.e. making up false claims between themselves).

Previous research by the author [11] addressed the former problem by computing trust quotients for Certification Authorities, trust quotients taking values being between 0 and 1 (where 0 represents completely untrustworthy and 1 represents completely trustworthy). The system also had a TrustCheck server that polled the network retrieving information upon which the trust quotient was based, so that dynamic trust quotient calculations could be made. Extending this work could provide some of the autonomic features of the trust infrastructure that we are looking to provide in TrustCoM. However, the knowledge involved in deciding whose trust-quotient to alter, and by how much, and under which conditions, and indeed what elements of trustworthiness are involved, is complex and highly ill-structured. Furthermore, it will be important to explore the juridical ramifications of such trust-altering decisions, and to build these into the knowledge base. To acquire such knowledge to a degree that is commensurate with normal juridical practice will require considerable knowledge acquisition expertise. Such a knowledge base could potentially be generated during the TrustCoM project.

6. Conclusions

We have shown that the PERMIS authorisation infrastructure already contains a number of features that constitute autonomic computing. However, in order to make the PERMIS authorisation infrastructure fully autonomic, and to better integrate it into the TrustCoM framework, addition enhancements are needed such as: the ability to grant default rights

automatically if a user's requested rights are denied; dynamic delegation of authority which will allow new managers to be automatically included in a VO; the addition of new types of repository such as web pages and file stores; the ability to select between different repositories according to the prevailing access conditions; automatic switching between access control policies according to the changing security environment; implementing mutual exclusion of roles to prevent users abusing their privileges; adding a secure historical database/audit function so as to base future decisions on past ones; and making decisions based on the trustworthiness or reputation of the accessing subject. The latter PERMIS enhancement places a number of requirements on the TrustCoM infrastructure, namely: how to measure the trustworthiness/reputation of subjects, how to determine the trustworthiness of reputation brokers, how to apportion responsibility between the various actors when a contract fails to be fulfilled, and how to collect and use evidences on which trustworthiness/reputation calculations are based. Whilst it may not be possible to solve all of these challenging problems during the lifetime of the TrustCoM project, we certainly intend to make significant progress towards doing so.

References

[1] D.W.Chadwick, A. Otenko, E.Ball. "Implementing Role Based Access Controls Using X.509 Attribute Certificates", IEEE Internet Computing, March-April 2003, pp. 62-69.

[2] ITU-T Rec X.812 (1995) | ISO/IEC 10181-3:1996 "Security Frameworks for open systems: Access control framework"

[3] ISO 9594-8/ITU-T Rec. X.509 (2001) The Directory: Public-key and attribute certificate frameworks

[4] The NIST RBAC Standard can be downloaded from http://csrc.nist.gov/rbac/

[5] A.Otenko, D.W.Chadwick, Von Welch "Integrating PERMIS with GT3" – in preparation

[6] Von Welch, Frank Siebenlist, David Chadwick, Sam Meder, Laura Pearlman. "Use of SAML for OGSA Authorization", Jan 2004, Available from https://forge.gridforum.org/projects/ogsa-authz

[7] The TrustCoM website is at http://www.eu-TrustCoM.com/

[8] http://www.research.ibm.com/autonomic/glossary.html#ac

[9] http://www.research.ibm.com/autonomic/overview/elements.html

[10] Philipp Obreiter "A Case for Evidence-Aware Distributed Reputation Systems" in Proceedings of the 2nd International Conference on Trust Management, 2004

[11] E.Ball, D.W Chadwick, A. Basden. "The Implementation of a System for Evaluating Trust in a PKI Environment" in *Trust in the Network Economy*, Evolaris vol 2. Eds Otto Petrovic, Michael Ksela, Markus Fallenbock, Christian Kitti. Pp 263-279, SpringerWein, 2003. ISBN 3-211-06853-8. (also available from http://sec.isi.salford.ac.uk/Papers.htm)

[12] SESAME project. See https://www.cosic.esat.kuleuven.ac.be/sesame/