

The dynamic adaptation of security policies in pervasive environments, with contextual information as the catalyst

Keara Barrett

Telecommunications Software & Systems Group,

Waterford Institute of Technology,

Cork Rd, Waterford, Ireland

+353 51 302916

kbarrett@tssg.org

Abstract

The concept of exploiting context information to produce dynamically adapting, accurate and timely security policies for pervasive environments and the entities of these environments are discussed briefly in this paper. The challenges relating to the effective establishment and enforcement of security policies in unpredictable pervasive environments are outlined, along with issues faced in the automated refinement of the security policies.

Keywords

Pervasive Environment, Context Information, Security Policies, Dynamic Adaptation, Automated Policy Refinement

Introduction

The ubiquity of computational presence (an inherent feature of pervasive environments) will augment the knowledge and decision making ability of people, in ways far beyond the ability and potential of today's computers, by means of constant access to information and computational capabilities. Yet, the advantages of constant accessibility to relevant information, services and computational power will arguably be rendered futile without the assurance of security and privacy [1].

The security breaches experienced with today's computer infrastructure are a mere sample of

what is expected with the proliferation of embedded computer power and the increased reliance on such power from individuals, households and businesses alike. Traditional security mechanisms will fail to meet the demands of pervasive environments, as features of these environments, including volatility, invisibility, heterogeneity and accessibility, will produce new opportunities for accidental and deliberate security violations. This suggests that the development of new security solutions, which consider the properties of pervasive environments, is essential [2].

Challenges Introduced

Policies allow the functionality of a system to be adapted without altering the implementation of the entities involved and are therefore suitable to volatile pervasive environments. Security requirements (rules and regulations) that ensure the appropriate use of facilities are usually specified through security policies [3]. As pervasive environments are highly adaptive and unpredictable it would be impossible for a user to identify all the potential security requirements for a given situation without assistance. The enormous amounts of computer infrastructure and the concealment of the underlying computing activities, needed for the realisation of Mark Weiser's¹ vision, exacerbates this challenge.

¹ Mark Weiser is the pioneer of pervasive computing (also known as ubiquitous computing or ubicomp).

Consequently, manually adjusting low-level operational security policies to generate the most optimal security in a particular situation and to accommodate the mobility and changing demands of users would be impossible. Even if manual management were feasible it would be unadvisable to adopt this approach, as it would negate the invisibility requirements crucial to the vision of pervasive computing. Furthermore, many researchers believe that the management of security would benefit if concealed from users because of the difficulty and distraction involved [4].

For these reasons, it is proposed to use situational information, otherwise known as context information, to dynamically trigger the removal and renewal of existing security policies and the creation of new security policies. Using context information² to generate security policies should result in the most applicable security for a particular situation without the addition of much complexity. Complexity will be minimised as context information is central to pervasive environments and must be considered irrespective of the technique employed to generate and manage security policies.

Context Information

Schilit and Theimer, the pioneers of context-aware computing, regard context to be location, identities of nearby people and object, and changes to those objects. They consider where you are, whom you are with, and what resources are nearby to be the important aspects of context. Abowd et. al.'s more recent classification defines context as: [5]

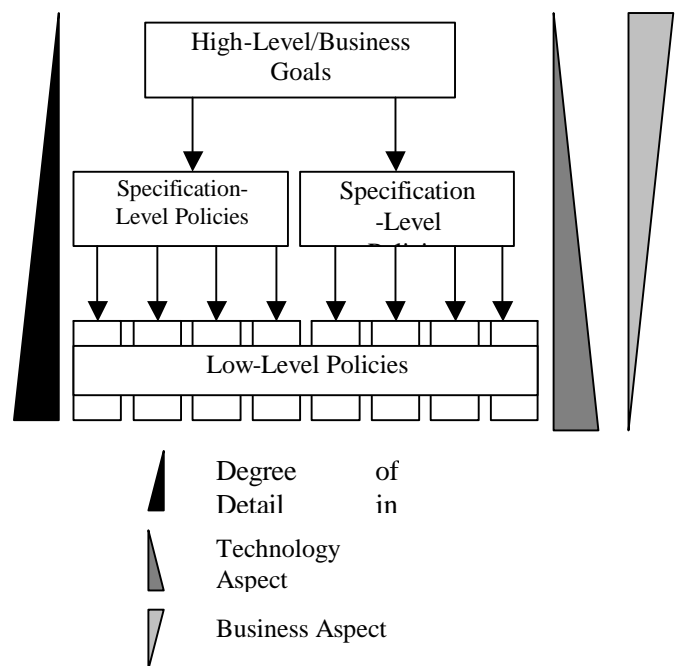
“...Any information that can be used to characterise the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an

application, including the user and application themselves.”

Any information that depicts the situation of an entity can be entitled context; therefore a change in the context information of a pervasive environment implies that a change in the state of the environment (or an entity of the environment) is pending, underway or concluded. When a state change occurs it is imperative that the high-level abstract security goals, which define the security requirements of the environment, are adhered to. To ensure that the high-level security requirements are met, the low-level operational security policies, which enforce the requirements of the high-level abstract policies, may need to be revised. In highly adaptive pervasive environment the rate at which revision is required will be overwhelming if managed manually. Hence, context information will be used to trigger and enhance an automated refinement process.

Policy Refinement

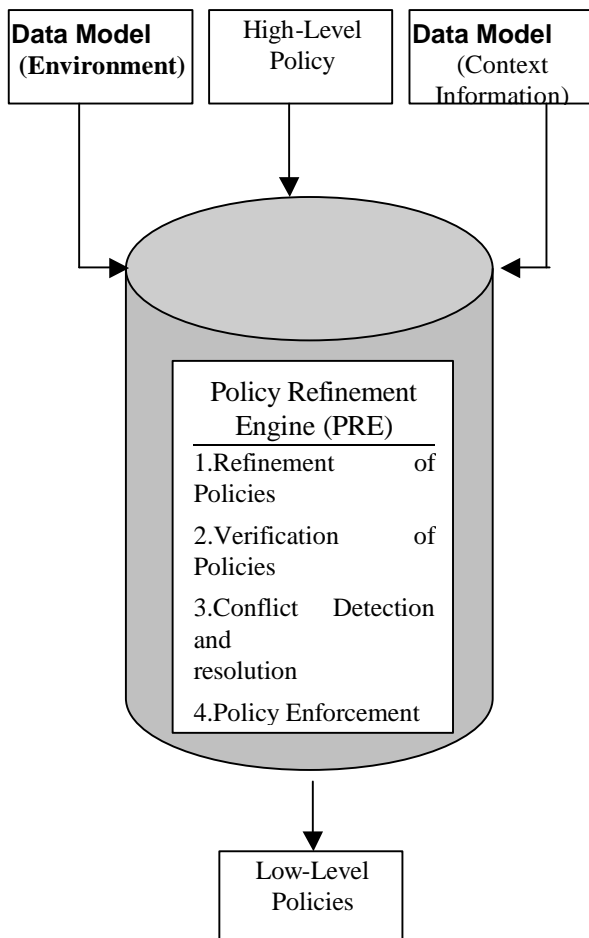
Policy refinement initiated by Moffet and Sloman [6] entails the process of deriving low-level concrete policies from high-level abstract policy specifications. This process operates around a policy hierarchy, with each layer of the hierarchy defining a specified level of abstraction.



² Any information that depicts the situation of an entity can be entitled context, the temperature, the presence of another person, nearby devices, the devices a user has at hand and the orientation of the user are examples.

Automated policy refinement, the transformation of a policy from one level of abstraction to another, will eliminate the need for a human operator who possesses a detailed knowledge of both business level and operational policies. In addition to the translated from one level of abstraction to another, the policy refinement process will investigate the accuracy and consistency of the low-level policies to verify that they meet the requirements outlined by the high-level policies and to identify conflicts.

The Policy Refinement Engine (PRE) will utilise information supplied from the data model of the pervasive environment, the data model of context information and the defined high-level policies to generate and verify low-level policies. The PRE will also identify and resolve conflicting policies. Once the low-level policies has been generated and verified for both conflicts and conformity with the high-level policy they will be enforced onto the environment or the intended entity of the environment.



References

1. Xiaodong Jiang and James A. Landay, University of California, Berkeley, “Modelling Privacy Control in Context-Aware Systems IEEE Pervasive Computing”, Vol.1, No. 3 July - September 2002. Available at <http://csdl.computer.org/dl/mags/pc/2002/03/b3059.pdf>
2. Lalana Kagal, Tim Finin, and Anupam Joshi, University of Maryland, Baltimore County “Trust based security for pervasive computing environments”, IEEE Communications, December 2001 Available: at <http://www.cs.umbc.edu/~finin/papers/vigil/vigil.pdf>
3. The SANS Security Policy Project <http://www.sans.org/resources/policies/>
4. Narendar Shankar, Dirk Bilfan , “Enabling Secure Ad-hoc Communication using Context-Aware Security Services”, Workshop on Security in Ubiquitous Computing, UBICOMP September 2002, Sweden Available at <http://www.teco.edu/~philip/ubicomp2002ws/organize/palo.pdf>
5. Gregory D. Abowd, Anind K. Dey, Robert Orr, and Jason A. Brotherton (1997). Context awareness in wearable and ubiquitous computing. In *ISWC*, pages 179–180.
6. Jonathan D. Moffett and Morris S. Sloman IEEE Journal on Selected Areas in Communications, 1993. vol 11(9): pp 1404-1414 “Policy Hierarchies for Distributed Systems Management”