

# Security in UbiComp: Protection through Commandments

Günter Müller   Rafael Accorsi   Martin Kähler  
Institute of Computer Science and Social Studies, Dept. of Telematics  
University of Freiburg, Germany  
{mueller,accorsi,kaehmer}@iig.uni-freiburg.de

## ABSTRACT

The increasing potential to combine devices with different capabilities and purposes leads to new security issues. Traditional security prescribes what has to be prohibited and is thus too inflexible to cope with a constantly changing context. In this paper, we propose an alternative approach to formulating security policies in UbiComp based on the notion of *commandments*. In essence, commandments formulate situations that should occur within an indefinite time span, thus being conceptually similar to, albeit qualitative distinct from, the notion of obligation. Here, we focus on the characteristics of commandments and not on how they are realised. To this end, we demonstrate the shortcomings of the prohibition-based approach to security and argue in favour of commandments as an instrument to provide long-lasting security guarantees in UbiComp environments. We conclude with a description of the upcoming research issues involved in applying commandments.

## 1. A NEW SECURITY PERSPECTIVE

Ubiquitous computing arises from advances made in distinct aspects of computing, namely reachability [5], pervasiveness [11], and autonomy [1]. These features lay the technical foundation for dynamic environments and, thus, for a plethora of novel services. While security is an essential requisite in every system, the unprecedented combination of these characteristics raises a number of challenges for traditional security mechanisms. Indeed, security is no longer only the protection against attacks, but, due to e.g. autonomy, also uses policies to decide whether parameters and configuration settings for self-management could threaten the system. Similarly, pervasiveness and reachability pave the way for heterogeneous contexts regarding devices' cryptographic capabilities. To address this security perspective, we propose a policy language based on the notion of commandments. Its goal is to provide long-lasting security guarantees in UbiComp environments.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*UbiComp '05* Tokyo, Japan.

Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

## 2. TOWARDS COMMANDMENTS

Traditionally, known security threats are remedied by devising security policies, which in essence describe threatening situations that should not occur. Security mechanisms then enforce these policies by prohibiting undesirable situations from happening. For example, AAA policies are prohibition-based [7]: Authentication protocols prevent peers from committing transactions with wrong partners (in particular, with an adversary); authorisation techniques preclude illicit subjects from accessing objects to which they do not have the right; and accountability mechanisms avoid repudiation of the actions of a peer.

However, to anticipate and describe each and every unwanted situation is an infeasible task in UbiComp, where emergent, at design-time unforeseen behaviour plays an essential role [4]. In this setting, one simply does not know what to forbid! We therefore argue that, although security policies based on prohibitions should not be disregarded, they are insufficient to actually provide a thorough account for security in highly dynamic systems such as UbiComp.

A promising approach to address this problem is based on the idea of *commandments* [2]. Instead of describing only threatening situations, security policies expressed by means of commandments also characterise situations that should eventually take place. Commandments make explicit statements about *what* is to be achieved, thereby contrasting to prohibitions, which state *how* a situation is to be achieved. In the context of information security, these specifications are called *extensional* and *intensional*, respectively [10]. Conceptually, the commandment-based approach to specifying security properties allows for more flexible specifications, while allowing for long-lasting security policies.

## 3. SECURITY BY COMMANDMENTS

Current security policies refer to specific security mechanisms. As they work at a very low level of implementation, these policies need to be continuously adapted for a changing environment. For example, an ACL that tells a reference monitor who is allowed to access resources and who is not, needs to be changed if a component is newly introduced in the system. Similar for IPSec, the policy that configures the security parameters of the protocol for the setup of new communication connections needs to be adjusted if the group of communication partners changes.

The primary goal of commandments is to endow a safer behaviour for highly dynamic systems based on UbiComp technologies. As an effect thereof, the overall availability (and thereby dependability) of these systems improves con-

siderably, as the occurrence of security incidents decreases. Observe, however, that this happens despite a continuously changing combination of components, services, and relationships between them.

Commandments work at application level and improve its security, and as a result, also that of the layers underneath. Take “use an up-to-date virus scanner” as a possible commandment. It neither gives instructions which scanner to use, nor which parameters need to be set: the use of an up-to-date scanner definitely protects the system as a whole. Which applications commandments should be referred to and which security benefits may result from them is part of ongoing research.

To realise this new security approach, we need a new form of policies that can continuously guide the system and hold at all times. Commandments describe situations that should occur within an indefinite time span or system properties that need to be preserved forever.

Even though the adherence of commandments is desired, it is not obligatory. Considering the virus scanner example, a monitor that observes the progress of the fulfillment would never be able to determine violations and trigger its enforcement. A component can always update its scanner in the next minute and, thus, fulfill the commandment. However, to urge the component to adhere, incentive systems need to be developed, to reward it for the desired behaviour.

In our setting, if a component does not adhere to a commandment – because it does not want to yet or it simply is not able to – it is not automatically excluded from accessing a specific service. Hence, commandments weaken the binary “yes or no” decisions of access control mechanisms and realise a first step towards a flexible and durable approach to security. UbiComp environments, in particular, take advantage of this property, as commandments impose security rules every device tries to satisfy according to its capabilities, without being too restrictive with devices incapable of adhering to them.

## 4. RELATED WORK

Currently, security policies are mostly enforced by prohibitive access control mechanisms. A common practice for more sophisticated policies is to enrich them with constraints [8]. These constraints can be classified into provisions and obligations and are similar to pre-conditions and post-conditions. While provisions relate to actions that need to be performed before an access decision, obligations impose conditions on the future behaviour of components, e.g. “release resource five minutes after access” [6].

Although the description of future actions makes commandments conceptually similar to the notion of obligations, they are qualitative distinct therefrom. Obligations are tightly related to traditional access control. If a component gains the requested access, it commits itself to corresponding obligations and satisfies them during a fixed time span. Due to this time constraint, the access control mechanism is able to enforce obligations, as it can decide violations at runtime and trigger appropriate countermeasures. Several enforcing mechanisms for obligations have been developed and use techniques such as aging of the access rules or lining up the access and its obligatory post-actions to an atomic sequence, which happens either completely or – with the use of undo-logs for resetting the system or the definition of compensational actions – not at all [3, 9].

## 5. OUTLOOK AND FURTHER WORK

In this paper, we put forward the commandment-based approach to security for UbiComp systems. In particular, we argue that: (a) prohibition-based security is an insufficient basis for security in UbiComp; (b) commandments provide a more flexible approach, thereby complementing prohibitions; and (c) since commandments cannot be enforced, incentive mechanisms have to be devised and tailored for this setting.

This is just the first step in this direction and there is considerable work to be done. First, the class of security properties that should be circumscribed with commandments is to be investigated. Given these properties, we will investigate current policy languages in order to enhance one with commandments, so that they can be described in a uniform fashion, being thus adequate for UbiComp systems.

Finally, appropriate incentive mechanisms have to be developed. Our initial investigation focused on applying economical concepts to tackle this problem and its adequacy is being investigated.

## 6. REFERENCES

- [1] Autonomic computing initiative. <http://www.research.ibm.com/autonomic/>, 2005.
- [2] R. Accorsi, S. Höhn, M. Kähler, G. Müller, and M. Strasser. Security by Commandments for Ubiquitous Computing Environments? <http://www.informatik.uni-freiburg.de/~accorsi/papers/AHKM05.pdf>, 2005.
- [3] C. Bettini, S. Jajodia, X. S. Wang, and D. Wijesekera. Provisions and obligations in policy management and security applications. *Network and System Management*, 11(3):351–372, 2003.
- [4] D. M. Chess, C. C. Palmer, and S. R. White. Security in an autonomic computing environment. *IBM Systems Journal*, 40(1):107–118, 2003.
- [5] G. H. Forman and J. Zahorjan. The challenges of mobile computing. *IEEE Computer*, 27(4):38–47, 1994.
- [6] M. Hilty, D. Basin, and A. Pretschner. On obligations. In *10th European Symposium on Research in Computer Security*, LNCS. Springer-Verlag, 2005.
- [7] International Organization for Standardization (ISO). Information processing systems – open systems interconnection – basic reference model part 2: Security architecture, 2000.
- [8] D. Marriott and M. Sloman. Implementation of a management agent for interpreting obligation policy. In *IFIP/IEEE 7th Int. Workshop on Distributed Systems Operations and Management*, 1996.
- [9] C. Ribeiro, A. Zúquete, and P. Ferreira. Enforcing obligation with security monitors. In *3rd International Conference on Information and Communications Security*, volume 2229 of LNCS, pages 172–176. Springer-Verlag, 2001.
- [10] B. Roscoe. Intensional specifications of security protocols. In *9th IEEE Computer Security Foundations Workshop*, pages 28–38. IEEE Computer Society Press, 1996.
- [11] M. Satyanarayanan. Pervasive computing: Vision and challenges. *IEEE Personal Communications*, 8(4):10–17, 2001.