Secure Smart Environments: Security Requirements, Challenges and Experiences in Pervasive Computing

Jun Wang†, Yaling Yang‡ and William Yurcik†

† NCSA, University of Illinois at Urbana-Champaign, {wangj, byurcik}@ncsa.uiuc.edu
‡ Computer Science Department, University of Illinois at Urbana-Champaign, yyang8@uiuc.edu

I. Introduction

In an ideal pervasive computing environment, a large number of connected smart devices are deployed to collaboratively provision seamless services to users. Pervasive computing is enabled by various advanced technologies, particularly wireless technologies and the Internet. It has become a trend for our future lives. A pervasive computing environment can be extremely heterogeneous. We can imagine how many different devices are involved in a smart home: TVs, phones, cameras, coffee makers, or even books and bookshelves. Since these devices are smart and communicate with each other mainly via wireless links, security must be ensured. Otherwise, the smart devices deployed around us would come back to hunt us and the result would be catastrophic. In this abstract we briefly discuss some important security issues in pervasive computing.

II. Security Challenges and Requirements in Pervasive Computing: The Big Picture

Due to the very high heterogeneity and complexity of pervasive computing, there are unique challenges and requirements for security insurance in a pervasive computing environment. Figure 1 shows a brief connection of these challenges and requirements.

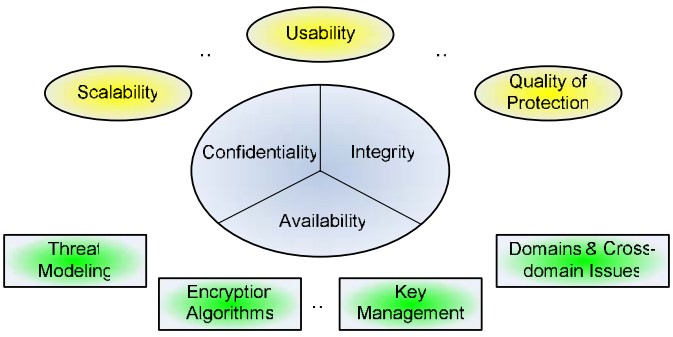

Fig. 1. Security issues in pervasive computing

Firstly, the scalability is a big challenge. Since a pervasive computing environment, such as a smart home or a smart office, may consist of enormous tiny smart devices that communicate with each other via wireless or wired links, to provide information security protection (e.g., authentication, data encryption, access control, etc.) is not a trivial task, especially when the resources of such tiny devices are limited. Moreover, for larger environments that cross multiple security/trust domains [8], scalability is even more challenging. For example, let us imagine a person who is equipped with wearable computing and monitoring devices – he himself is a separate security domain – enters his smart office and wants to remotely control some devices in his smart home. In this case, at least three security domains are involved: home, office and himself. The security solution must be scalable not only within one domain, but also across multiple domains. This scalability issue might be even more complicated than the PKI (Public Key Infrastructure) scalability issue, because in some pervasive computing environments, such as sensor networks, no centralized authority is available.

Secondly, given such a heterogeneous environment of pervasive computing, to clearly understand and define the usability of security measures and their Quality of Protection (QoP) is difficult. Some QoP-related issues in pervasive computing have been studied by Dr. Nahrstedt et al in MONET group [1], [2]. However, there has been a well-known paradox between usability and security QoP [3]: the more sophisticated the security technology, the more likely users will thwart the security, because users become more reluctant to follow the security measures due to low usability. Hence, how to solve this paradox in a pervasive computing system is a challenge. An effective solution may require a deeper understanding and a more precise measurement of usability for different security technologies, which is still wide open.

Thirdly, to build a security model for a pervasive computing system is difficult. Building a security model includes the following six steps.

- Threat model: threat modeling is an important step to understand the factors that pose the main threats to confidentiality, integrity, and availability in a pervasive computing system. Due to the high complexity and heterogeneity, building a formal threat model for a pervasive computing system is a difficult task.
- Security domain identification and interface definition between domains: a pervasive computing system can be divided into multiple security domains following physical boundaries or trust-based boundaries [6]. For example, a person's home and office can be considered as two separate security domains. Even the person himself can be regarded as another separate domain. However, sometimes identifying such domains can be subtle. Moreover, to clearly define the interfaces between these domains is

difficult.

- Layered structure definition within a security domain: for any large domain, it is necessary to divide it into multiple subsystems according to different security requirements/levels of devices. In particular, how to define such subsystems within a domain in a pervasive computing environment is still open. One possible way is to use layered structures (e.g., an onion structure). Additionally, the interface between these layers should be defined carefully.
- Quantitative security measurements and more systematic optimizations: Quantitative study of security has been a longstanding research problem. Stochastic methods may be used. But it is still wide open in terms of stochastic security modeling and systematic optimization.
- Security verification of a pervasive computing system: due to its high complexity and heterogeneity, an automatic security verification method would be very helpful. We will discuss this issue later in Section III-B.
- Balance between security and QoS: providing security requires extra resources. For example, it has been shown by Klepzig [7] that supporting PKI in a networked system significantly increases network bandwidth consumption and response delays. Since the tiny devices deployed in a pervasive system may have limited resources, there could be a tension between security and QoS. Furthermore, additional requests of authorization, authentication, identification, and privacy protection [4], may take even more resources in a pervasive computing system due to mobility. Therefore, how to balance security and QoS is not trivial. It may require a precise threat model for the entire system, which is also difficult.

III. A Lesson Learned: Smart Security in Pervasive Computing?

A. An Incident – How Our Devices Were Compromised

Several months ago, multiple devices in our laboratory were compromised by hackers. After carefully examining the forensics in the logs, we found that the devices were hacked through a hidden security hole that crossed multiple security domains and was not able to be prevented by normal security measures such as firewalls. This incident inspires us that we may need an automatic tool to manage security issues crossing different domains, finding any possible security holes in design, just as we need a tool to automatically manage QoS in the system. Ideally, this tool should be easy to use and self-adaptive.

B. The Solution: Smart Tools for Security Management?

Since there are multiple security domains in a pervasive computing environment, each of which may be administrated by different operators, a "smart" tool that can automatically operate across these multiple domains may be helpful to prevent devices from being compromised. This smart tool can either be used during the design process to find and clear out all possible security holes, or it can operate in parallel with the entire pervasive computing system, actively or passively monitoring the system and setting off an alarm if any security problem is found. In order to do so, the security domains in the system and the interfaces between them must be carefully defined and formulated by using, for example, graph theory. A protocol must be designed across the multiple domains in the system. If the tool is used during the design process, it is responsible for collecting configuration information from each domain and looking for any possible security holes in the system by using, for example, formal methods or searching methods based on graph theory. If the tool is used in parallel with the system to manage security issues, smart agents [5] may be deployed in each domain to collect information and fulfill the monitoring task. In this case, the tool can be considered as a peer-to-peer subsystem and a protocol or agreement must be implemented among those agents for information sharing and decision making. It is still unclear if such a tool is feasible or how to implement it, mainly due to the crossing domain issues.

This automatic security management tool motivates at least the following three research efforts. Firstly, a formal model of the entire system needs to be built. At the same time, a threat model that covers different security aspects should be built both for individual elements in the system and for the entire system itself. Secondly, based on the system model and the threat model, a formal method should be used to verify the security for the system and eliminate any possible security holes. Finally, an automatic mechanism should be designed to monitor the security status of the system. In addition, it should be able to detect incidents and contain local damages if the system or a part of the system is compromised.

IV. Conclusion and Future Work

In this extended abstract, we have discussed the requirements and major challenges for ensuring security in pervasive computing environments. We also briefly discussed lessons that we have learned from an intrusion incident. From this incident, we found that an automatic security management tool is needed. In order to do so, we need a further research on threat modeling, formal security verification, and automatic incidence detection and containment in a pervasive computing system.

References

[1] http://cairo.cs.uiuc.edu/security/QoPS.htm.

[2] http://cairo.cs.uiuc.edu/security/smv.htm.

[3] http://deyalexander.com/resources/security.html.

[4] Roy Campbell, Jalal Al-Muhtadi, Prasad Naldurg, Geetanjali Sampemane, and M. Dennis Mickunas. Towards Security and Privacy for Pervasive Computing. In *Theories and Systems, Mext-NSF-JSPS International Sympsoium, ISSS 2002*, Tokyo, Japan, 2002.

[5] Howard Chivers, John Clark, and Susan Stepney. Smart Devices and Software Agents: the Basics of Good Behaviour. In *First International Conference on Security in Pervasive Computing*, 2003.

[6] Lalana Kagal, Tim Finin, and Anupam Joshi. Trust-based security in pervasive computing environments. *IEEE Computer*, 34(12):154–157, December 2001.

[7] Kelley R. Klepzig. Modeling and Simulation of Public Key Infrastructure Applications. In *SANS White Paper: http://www.giac.org/certified_professionals/practicals/gslc/0016.php*.

[8] Philip Robinson and Michael Beigl. Trust Context Spaces: An Infrastructure for Pervasive Security. In *First International Conference on Security in Pervasive Computing*, 2003.