Towards Semantic Resolution of Security in Ambient Environments

Mario Hoffmann[1], Atta Badii[2], Stephan Engberg[3], Renjith Nair[2], Daniel Thiemert[2], Manuel Matthess[1], and Julian Schütte[1]

[1] Fraunhofer SIT, GER
[2] IMSS, University of Reading, UK
[3] Priway, DK

Abstract. Driven by new network and middleware technologies such as mobile broadband, near-field communication, and context awareness the so-called ambient lifestyle will foster innovative use cases in different domains. In the EU project Hydra high-level security, trust and privacy concerns such as loss of control, profiling and surveillance are considered at the outset. At the end of this project the Hydra middleware development platform will have been designed so as to enable developers to realise secure ambient scenarios. This paper gives a short introduction to the Hydra project and its approach to ensure security by design. Based on the results of a focus group analysis of the user domain "building automation" typical threats are evaluated and their risks are assessed. Then, specific security requirements with respect to security, privacy, and trust are derived in order to incorporate them into the Hydra Security Meta-Model. How concepts such as context, semantic resolution of security, and virtualisation support the overall Hydra approach will be introduced and illustrated on the basis of a technical building automation scenario.

1 Introduction

A digital revolution is changing our life and work styles powered by an embedded ICT-empowered environment. From washing machines used in our homes over logistics tracking to mobile phones and PDAs on which we depend to communicate and work, they all deploy embedded systems. World Semiconductor Trade statistics show that 98 percent of the programmable digital devices are embedded devices [1]. Whilst the plethora of embedded programmable devices is re-assuring of a competitive, diverse and hopefully enduring creative base of Research and Development in such critical components, it also makes for a heterogeneous array of devices distributed in the ambient environment which cannot communicate with each other due to lack of a common protocol to provide for the much needed seamless integration. Imagine any of your devices being able to interact with any other device so that even a customised PDA with an interface that is familiar to the user can manage devices such as TVs and door locks in a hotel room, a short-distance device can use long-distance capabilities of other devices and users can manage devices in other domains remotely. Further, every application or service could use all devices in place so that e.g. an application can utilise all available sensors or support devices deployed independently of the application. This depends of course on permissions of the developer and requirements of the user – e.g. users could choose services that respect security and privacy according to a certain policy. Security challenges are hard in homogeneous solutions, but escalate when moving to enable inclusive interoperability. Here we need to depart from traditional thinking based on device identification with significant use of implicit knowledge and manual administration to a model-driven and semantically open security model based on explicit assertions and shared ontologies. For developers to open the digital access to devices and applications, they require a flexible and much more nuanced security model; for users to be able to trust communication between devices, they need new models for user controls and security fault tolerance. Imagine what happens if biometric sensors in people's homes suddenly turn up to be accessible and controllable by neighbours and criminals, acting as commercial spyware or even political control. The fear of such scenarios significantly reduces the value potential of this embedded networked revolution. The EC co-funded FP6 IST project Hydra (Networked Embedded System Middleware for Heterogeneous Physical Devices in a Distributed Architecture) to support some of the leading companies and research institutes in Europe in attempting to fulfil the vision of such seamless integration in the ambient environment of heterogeneous devices. Hydra aims to develop a middleware layer for building secure, fault-tolerant networked embedded systems where diverse heterogeneous devices co-operate [2]. The emergent world of ambient intelligence and pervasive computing would be closer to realising its full potential if the embedded devices deployed, for example in a home, are able to communicate semantically interoperable with each other and cooperate to fulfil tasks. The Hydra mission is to provide this capability by designing the required middleware facilitating semantic interoperable security.

2 Hydra Challenge

When speaking of interoperability the challenges we are facing are manifold. Starting with the simple issue of having two devices, one being able to use Wifi, the other being able only to use Bluetooth we are confronted with different types of protocols, not only in terms of communication but also in terms of security. In most projects, industrial or research, security is often a neglected area as developers tend to ignore its importance. It is mostly thought to be an add-on which can be implemented later, if at all. This holds several threats as most security leaks can only be closed afterwards with an immense effort. Considering these security leaks from the very beginning is the aim of the Hydra project. Such Security by Design with the main focus on interoperability of security helps to build a powerful tool to enable manufacturers and developers to develop secure applications and devices in an ambient environment. To demonstrate the middleware in various areas the project is primarily focusing on 3 domains: *Home*

Automation, *Healthcare* and *Agriculture*. The Security by Design approach itself is focusing on enabling secure interoperability. This means that a developer of embedded applications for ambient environments should not need to take care that the devices his applications uses or communicates with have the same specifications, e.g. same communication protocol or security protocol. If one device interacting with the application uses protocol A, and another device interacting with the application uses protocol B, then the developer of the application should be able to handle this using the Hydra middleware. This will be achieved by semantic resolution of security, i.e. turning physical capabilities and functionalities into semantically understandable descriptions, making the interaction independent from the specifications of network, devices and applications/services. In the next sections we will present a bird's eye view of our research within the Hydra project to derive the requirements and the approaches which we will use in order to fulfil these requirements. In this way we intend to provide some answers to our common concerns to achieve not just secure interoperability but potentially also cooperativity amongst heterogeneous embedded systems serving us in the emergent ambient environment.

3 Security Requirements Engineering

In the Hydra project the following security requirements specification process (cf. 1) is performed in order to ensure security by design: First, we derive a technical scenario from the building automation user domain scenario. Then, we conduct discussion rounds with focus groups of expert developers who are potential future Hydra middleware users. In the focus group analysis, actors, assets, and roles are identified. Based on the analysis of multilateral communication schemes between those roles we identify high-level threats to Hydra. Following the concept of "security by design" we derive the overall protection goals that have to be taken into account for the design of the Hydra middleware platform. The results of the focus group analysis in combination with the state-of-the-art are the basis for the risk analysis. Here, the identified threats and potential (threat) actors are analysed and described. Probability, impact and effects of successfully performed attacks are assessed and used as input to calculate the risk of a threat. From that point it is then possible to estimate how serious actors should take a threat. Finally, the process results in derived and prioritised security and trust requirements based on the results of the risk analysis.

A. Technical Scenario The technical scenario used in our approach is built on the vision scenario for the user domain "Building Automation". Since the vision scenario is not very detailed in terms of technical aspects the technical scenario adds this information. The aim of the technical scenario is to give the members of the focus group a better and more detailed starting point for their technical interpretations to elicit requirements for the security and trust within the Hydra project.

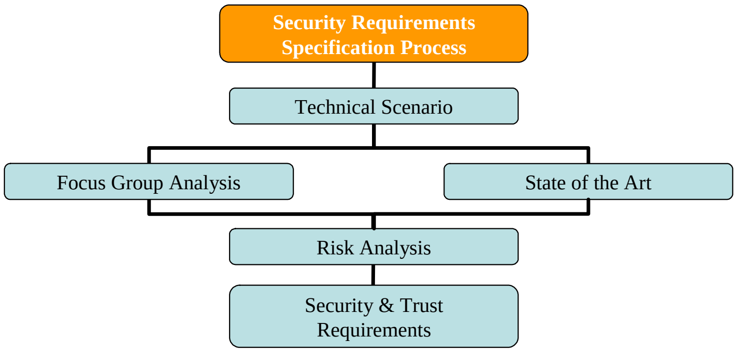
Security Requirements Specification Process
Technical Scenario
Focus Group Analysis
State of the Art
Risk Analysis
Security & Trust Requirements

Fig. 1. Security Requirements Specification Process

B. Results of Focus Group Analysis The technical scenario is the starting point of focus group analysis. Here, an initial threat analysis of the technical implications identifies assets to protect such as billing information, user preferences and profiles, as well as communication data, actors such as building operators, service agents and occupants, and roles such as network operators, content providers and end-user. The analysis of multilateral communication schemes between such roles derives the main protection goals that the developers would expect to be met taking advantage of the future Hydra middleware platform. These comprise: (1) Confidentiality, (2) Integrity, (3) Authenticity, (4) Authorisation, (5) Availability, (6) Non-repudiation, and (7) Privacy.

C. Results of Risk Analysis On the basis of these protection goals the risk analysis defines eight steps as part of a Hydra specific user-centric framework for risks analyses and evaluation. This comprises a pattern-based description of assets, (threat) actors, and threats as well as the assessment of attacks, their impact, their probability, and security implications. The highest risks in our analysis according to the usage scenario are expected to affect user data and identity, where identity comprises both user identities as well as device identities.

D. Security and Trust Requirements The derivation of the security and trust requirements based on the previous results conclude the security analysis. The requirements are prioritised according to their classification into the categories mandatory, desirable and optional requirements. With respect to the risk analysis the most important requirements concern securing confidential information, e.g. private data during transactions, and empowering the user to control both his individual context and the disclosure of personal information to the immediate vicinity as well as to authorised (virtual) parties.

E. Hydra Synthesis The more personalised information has to be collected, linked and analysed by ambient systems in order to serve users according to their individual context, the more the specific protection goals have to be balanced between actors in those scenarios. More than 80% of the security and trust requirements have been classified "mandatory" to be fulfilled by the Hydra security model. Most important requirements aim at (1) securing confidential information, (2) authentication mechanisms, (3) context-aware access control, (4) context and semantic reasoning, (5) interoperability of (security) communication protocols, and (6) distributed trust models. In order to fulfil these requirements we propose a security meta-model with the following key characteristics: "be interoperable with existing security models", "be extendable", "allow developers to semantically define security requirements", "allow developers to virtualise end-users, services, and devices", and "simplify implementation". The concepts needed to realise the Hydra Security Meta-Model, i.e., (1) context security, (2) semantic security resolution, and (3) virtualisation, will be introduced in detail in the next section.

4 Hydra Security Approach

In this section, the main concepts of Hydra's security capabilities are presented and the approach to the Hydra Security Meta-Model is outlined.

4.1 Context

One of the main concepts of Hydra is the notion of context. By context, we understand any information that can be used to describe the situation of an entity, whereas the information is observer-specific, i.e. there is no global context [3]. The processing of context is structured in four layers: *Context Sensing*, *Context Awareness*, *Situational Awareness* and *Reasoning*. At the first two layers, raw data, e.g. from sensor nodes is collected and processed in a way that allows defining a structured representation of context. At the next layer, the awareness of the situation the entity currently behaves in is created by linking the contexts of other entities nearby. Reasoning finally is the process of deducing possible consequences of the current situation.

Although higher layers of context processing are application-specific and cannot be part a middleware, the idea of context will play a major role in Hydra. On the one hand, context data will be a part of the integrated security model, e.g. by supporting Attribute-Based Access Control (ABAC) mechanisms [4,5]. On the other hand, Hydra aims to enable the development of ambient environments by providing context data along with processing operations and tools for context-aware applications while reducing the security problems which may be introduced by the concept of context:

Context contains a lot of sensitive data (e.g. location or interests of a user) and thereby raises the risk of privacy violations. Especially due to uncontrolled linkage of different contexts, it would become impossible for an user to keep his personal data under control. Thus, it is critical to provide only as much context information as needed to an application or a service. Vice versa, every entity may be a data source for other context sensing entities and thereby could unintentionally reveal information about itself. Hydra will address these problems by providing concepts which help limiting the amount of information that is gathered and exposed across different contexts while still allowing to link contexts in order to generate situational awareness. One of these concepts is virtualisation which will be described in the next section.

4.2 Virtualisation

As interconnection increases and users and devices behave in different contexts, perimeter security tends to fail. Moreover, a number of security problems and functional requirements arises and needs to be addressed by appropriate mechanisms:

At first, it must be possible to avoid the tracking of users and devices across different contexts. Hence, information leakage from one context to another must be prevented. Further, an entity might need different context-specific representations. An example would be a home automation system which provides a different interface and different functionalities, depending on whether it is used by a technician or by a normal user, whether it is in maintenance mode or in normal operation mode. It may be required as well to apply mechanisms to legacy devices which do not have the capabilities to provide these mechanisms by themselves.

Hydra uses the concept of virtualisation to address these issues. By virtualisation, we understand the process of creating a logical representation of an entity. As the logical representation is an entity in itself, it is feasible to nest and combine virtualisations and by that way e.g. create a single logical representation of multiple different entities. As virtualisation refers to generic entities, not only hardware devices can be virtualised but also applications, users and their identities.

Thus, Hydra proposes to apply virtualisation mechanisms to different entities: *Virtual devices* or *proxies* act as logical representations of devices. By defining a proxy for a physical device, it is thus possible to integrate non-Hydra-enabled devices into a Hydra-enabled network and to enable further high-level concepts like semantic description of device capabilities or resolution of security. In addition, physical devices can be combined to virtual devices which are tailored to the application – e.g. it is possible to define a “virtual” global light switch that controls all lights within a building. Virtual devices will also allow representing a device with a reduced set of functionality – either to reduce complexity for the user or in terms of access control[4].

Virtual identities are an important aspect, as well. They allow a user to define different identities for different contexts. Through virtual identities it is possible to recognise a user within a certain context while not being able to identify

[4] This will of course only prevent accesses to the device going through the proxy. Controlling direct physical access to a device is out of scope of a middleware such as Hydra.

the same user in a different context. Thus, virtual identities help preserving the user's privacy by avoiding the linkage of identities across context boundaries which would otherwise lead to accumulated private information about a user. Another advantage is that virtual identities help a user not to disclose more information about himself than required. For example, if a service agent enters a house in order to carry out a maintenance task, he can identify himself as a delegate of the service company instead of providing personal information about himself. It is also conceivable to extend the concept of virtual identities to *virtual users* in form of personal agents, performing tasks (semi-) autonomously on behalf of the actual user.

Further virtualisation techniques are possible; however, the above described mechanisms will make up the main part of Hydra's virtualisation design.

4.3 Semantic Resolution of Security

Interoperability of heterogeneous devices and applications also requires security to be resolved at a semantic level. This is to ensure translation between heterogeneous devices, to delegate security decisions from applications to the middleware layer and to ensure adaptability according to the specific context. While the Hydra middleware will not enforce a specific security model on devices or applications, it is nonetheless responsible for ensuring interoperability in even sensitive applications. The goal is the middleware to be an abstraction layer between the security models and protocols supported by devices and applications and the specification of security requirements made by the developer. Thus, a model-driven approach is needed which allows the representation of security requirements, policies and capabilities at a semantic level and translates these specifications to a concrete environment. One approach would therefore be the usage of ontologies for the semantic representation of protection goals, access rules, security capabilities as proposed in [6] and [7]. However, Hydra itself will not provide ontologies, but rather define the requirements and interfaces to integrate such.

4.4 Security Boundaries

The interface between Hydra and non-Hydra defines the security boundary. The security parameters of all entities within the security boundary can be represented in a semantic way and thus be controlled by Hydra (but don't have to). Entities that are outside the security boundary cannot be controlled by the middleware and thus their security parameters are not subject to the rules specified within Hydra. The security boundary is flexible and depends primarily on developer and user choices about the extend to which devices and applications will be part of the Hydra environment. In this way, as a facilitator rather than a guarantor of security, Hydra provides for security-aware design and development by enabling developers of embedded systems and to include security and privacy aspects in their applications.

4.5 Towards a Security Meta-Model

In heterogeneous environments, one impediment to interoperability are the differences between security protocols, identity schemes, authentication mechanisms, etc. In order to overcome this drawback, Hydra will make use of a Security Meta-Model which will mainly comprise of the above described concepts context, virtualisation, flexible security boundaries and semantic resolution of security. This model will be a meta-model, i.e. it will be a “model of models”, abstracting from concrete security mechanisms to semantic descriptions. Developers will have the opportunity to define security requirements at a semantic level and leave the mapping from semantic specification to concrete security mechanisms to the middleware. So, although Hydra is a middleware and thus can neither make context-based decisions by itself, nor enforce security, it will provide developers with concepts which allow creating context-aware, yet secure applications in heterogeneous environments.

5 A Usage Scenario

In order to illustrate the necessity and benefits of the Hydra Security Meta-Model, we implement a demonstrator scenario (cf. Fig. 2). The demonstrator is based on the technical building automation scenario used as the starting point of the security analysis in section 3. In this scenario, a service agent sent by a service provider needs physical access to a faulty heating system of a resident who is currently not at home. The steps 1 to 4 in Fig. 2 focus on the security challenges and how these will be resolved through the realisation of specific parts of the Hydra Security Meta-Model: The scenario starts with a critical malfunction in the heating system that has been detected by a device specific Hydra proxy in step 1. In current home and office automation systems Hydra proxies serve as virtual representations of legacy devices in the Hydra network as defined in our virtualisation concept in section 4.2. On the one hand they take into account device specifics by semantic description of device capabilities while on the other hand they take advantage of the Hydra security mechanisms by semantic resolution of security for example. Future devices are envisioned to be Hydra-enabled so that they can run Hydra middleware by themselves. Once the heating system's Hydra proxy has sensed the malfunction and changed its status the Hydra based building automation system (HBAS) is aware due to the fact that the contexts have been linked a priori. Therefore, changes to the context of the Hydra proxy are known to the HBAS. The HBAS then reasons taking into account further information like season or temperature to determine the criticality of the error and sends an error message to the resident. The HBAS request includes the error protocol and recommends calling a service provider to fix the problem.

In step 2 the resident receives the authentic request from his HBAS and decides to follow the recommendation. He digitally signs the error protocol and sends it – including a context restricted authorisation token – to a service provider of his choice. The authorisation token will be used in step 3 which

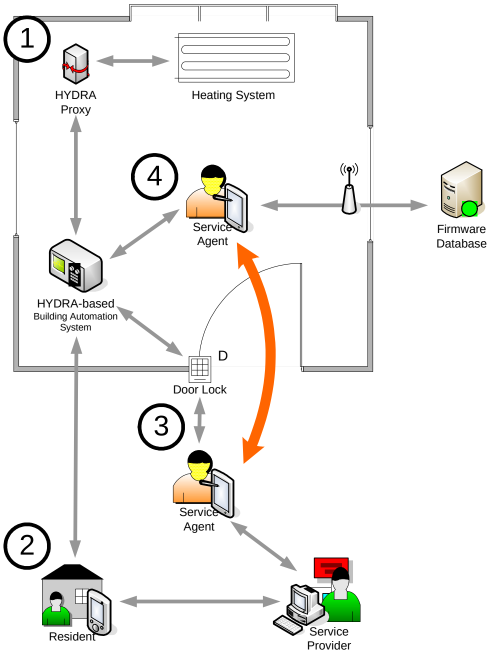

Fig. 2. Demonstrator Scenario

describes the situation in front of the resident's house. In this step, three different objects perform context sensing: the mobile device of a service agent, the door lock and the HBAS. The service agent presents the authorisation token stored on a Hydra-enabled PDA to the door. The door forwards the token which has been additionally signed by the service provider to the HBAS that proves it valid and trusted. Thus, the service agent is now allowed to enter the house and gets access to the HBAS in the final step. Note that the HBAS does not ask for the service agent's identity – the double-signed authorisation token (by the resident and the service provider) is sufficient to guarantee liability. In the final step – step 4 – the service agent gets context restricted access to the Internet in order to download the latest version of the heating system's firmware update. After fixing some configuration settings and installing the update of the firmware the heating system works smoothly inside of its specification again. In addition to the authorisation process based on trusted credentials and virtualisation, the demonstrator will be improved by two steps: Firstly, semantic resolution of security will add trusted authentication in the Hydra network (even to non Hydra devices by using Hydra proxies as mentioned above). Secondly, the rather simple Role-Based Access Control (RBAC) above will be enhanced to Attribute-Based Access Control (ABAC) mechanisms (e.g., XACML [8]) to support more dynamic and unforeseen scenarios. The demonstrator will be shown at CeBit fair 2008.

6 Summary and Outlook

In this paper we have presented the approach to security, privacy and trust in ambient environments supported by a context-aware middleware. We have presented our process of gathering the requirements for a middleware for heterogeneous networked embedded systems in the Hydra project. Furthermore, we have introduced our approach to meet those requirements which is based on semantic resolution of security, virtualisation, and context, forming a security meta-model. Further research in the project will be focused on applying different technologies of virtualisation on different types of entities, e.g. users, devices or applications. Further, we plan to investigate how different security models can be represented semantically based on ontologies in order to realise interoperability. Such ontologies will also be used to realise semantic models of devices and applications to enable semantic interoperability. The concept of context to support security will be detailed in terms of representation of context information. The final outcome will then be the security meta-model, in addition to a software development kit and an integrated development environment, which will enable developers to involve security aspects from the initial stages of embedded application development.

References

1. FAST GmbH for the European Commission: Study of worldwide trends and R&D programs in embedded systems in view of maximizing the impact of a technology platform in the area (2005)
2. HYDRA: Networked embedded system middleware for heterogeneous physical devices in a distributed architecture. http://www.hydra.eu.com (2007) contract number: IST-2005-034891, duration: 07/2006-06/2010.
3. Schilit, B., Adams, N., Want, R.: Context-aware computing applications. In: IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, US (1994)
4. Damiani, E., di Vimercati, S.D.C., Samarati, P.: New paradigms for access control in open environments. In: Proc. of the 5th IEEE International Symposium on Signal Processing and Information. (2005)
5. di Vimercati, S., Samarati, P., Jajodia, S.: Policies, models, and languages for access control. In: Proc. of the Workshop on Databases in Networked Information Systems. (2005)
6. Naval Research Lab: NRL Security Ontology (2007) http://chacs.nrl.navy.mil/projects/4SEA/ontology.html.
7. Dritsas, S., Gymnopoulos, L., Karyda, M., Balopoulos, T., Kokolakis, S., Lambrinoudakis, C., Katsikas, S.: A knowledge-based approach to security requirements for e-health applications. Electronic Journal for E-Commerce Tools and Applications (2006)
8. OASIS: eXtensible Access Control Markup Language (XACML) Version 2.0. http://www.oasis-open.org/committees/xacml (2004)