Enabling Secure Ubiquitous Interactions

Kevin Eustice, Shane Markstrum, Venkatraman Ramakrishna,
Peter Reiher, Leonard Kleinrock, and Gerald Popek

Laboratory for Advanced Systems Research,
Department of Computer Science,
University of California, Los Angeles, CA 90095
{kfe,smarkstr,vrama,reiher,lk,popek}@cs.ucla.edu

Abstract. Future ubiquitous computing environments will require devices to be automatically and safely configured together to perform important tasks for the users they support. Security concerns based on known vulnerabilities of the Internet make it clear that any widely deployed new computing infrastructure must be designed with substantially more security. The highly dynamic relationship between applications, devices, and environments defies existing security models, and requires new techniques to deal with its unique demands. We propose a new paradigm for creating and maintaining safe, ubiquitous computing environments, based around the novel idea of organizing related devices into spheres of influence, a concept used to capture both geographical and semantic groupings. Spheres are used to encapsulate policy and provide well-defined boundaries for interactions. Intra-sphere interaction requires policy-based negotiation between principals.

1 Introduction

We stand at a technological watershed; gazing ahead, we see a world populated with numerous intelligent devices that offer an immense amount of computational power and a rich communications infrastructure [Weiser1991]. A new paradigm of smart appliances, intelligent homes and offices is nearly upon us. However, for these exciting technologies to emerge from research labs and be deployed ubiquitously, a solid foundation of security and safety must first be in place. Existing research has focused its energy principally on developing interesting applications and novel infrastructures to manage mobile users and devices, leaving open the question of how to deal with system security, manage complex, domain-specific policy, and handle complex, access-control issues in an environment composed of a heterogeneous mix of devices, infrastructures, individuals and applications.

We have passed the point where it was sufficient to merely enable communications between entities. It is essential to establish this framework for extensible secure ubiquitous computing at an early stage. This is the only way to provide some assurance that the kinds of difficulties currently being encountered at considerable cost in today's Internet do not also plague a much larger setting.

We require new models for representing the complex dynamics of ubiquitous interactions, as well techniques to analyze and manage the flow of information and control in such environments. We require techniques to assess the appropriate level of required access for an entity within a ubiquitous environment. Additionally, *least privilege* must be maintained throughout interactions, requiring that entities be granted the minimum set of privileges necessary to accomplish a task, and that interactions be actively managed by policy-aware system components.

2 Challenges

Current commercial, government and academic research projects are working toward exciting goals of virtually omnipresent network access and device services. However, in this new environment we require security, safety and policy components that mediate and manage resources and devices. There are multiple challenges that we must address, including problems of integrity, policy, and privilege management.

2.1 Integrity

As more homes and public areas offer interactive services to mobile clients, they will also provide new vectors for attacks on critical infrastructure. These attacks will not necessarily come in the form of strangers outside our homes attacking our household WiFi network – they will also come in such forms as electronic hitchhikers who latch onto PDAs and electronic jewelry inside shopping malls, Trojan horses resident in the unbranded digital video recorder just added to our home infrastructure, or a museum visitor who bypasses a museum's wireless tour and accesses payroll information.

These observations have sobering implications for ubiquitous computing. A ubiquitous computing framework must support integrity analysis and assessment of devices and applications operating in the environment. Without such a component, even well-known and trusted devices may return home carrying unwanted, possibly malicious intruders. Additionally, we would additionally desire mechanisms to update or repair vulnerable or exploited devices. For example, this would allow devices to be cleaned of viruses or Trojan horses before entering the environment, in the manner of a virus repair tool; as well as dynamic updating of vulnerable system packages, in the manner of an operating system auto-update. It is ultimately up to the ubiquitous environment to decide what integrity requirements it places on entities that wish to receive service; however, these mechanisms allow the environment to make that choice.

Unfortunately, these concerns do not simply end when the device enters an environment; instead, they are an ongoing concern. New vulnerabilities are being discovered constantly, and firmware and software updates are unavoidable. This implies that ongoing maintenance is necessary to keep local entities up-to-date.

2.2 Policy for Ubiquitous Computing Environments

A second concern is the policy management in ubiquitous environments. Policy specifies environmental and service-specific behaviors and constraints on entity interactions. Examples of policy include location or temporal constraints on access to services, integrity requirements, and restrictions on accessed content. In addition to restricting interactions, policy can also enable; it may specify desired behaviors and responses within the environment. Currently, devices are configured individually, often per-user. The home PC, the television, and the game console all may possess similar types of configurable policies, yet each has to be configured in isolation. This is ultimately unworkable.

Devices in the ubiquitous environments need to be able to share local policy; it should be sufficient to set a content restriction for an environment, and have that policy apply to all appropriate interactions. However, it is not sufficient to provide policy information to devices; policy must be enforced. A framework must provide mechanisms for ensuring that entities adhere to local policy.

Additionally, there is need for further development of policy languages that are appropriate for ubiquitous computing environments. Typical policy languages have focused on a specific domain application. It would be desirable to have a policy language that can describe security constraints, as well as enabling other types of desirable interactions. There has been some progress in this area [Kagal2002a], but more work is necessary to understand the policy requirements of ubiquitous computing environments.

2.3 Privilege Management

Privilege management is difficult, especially within extremely dynamic systems. However, it must be addressed within the ubiquitous computing context. Typical systems grant users and devices a broad set of privileges for any given session, with little or no attention paid to the actual stated intent of the given task. This is undesirable in any system as no system is without vulnerabilities. If the set of privileges granted to a device exceeds the minimum and necessary set of privileges needed to accomplish a task, it is much more likely that the device would be capable of exploiting a yet-undiscovered vulnerability.

Privilege management is a security parachute we employ to protect ourselves from the intruders we cannot detect and the vulnerabilities we cannot find. By assigning and enforcing *least privilege* semantics to ubiquitous interactions, we greatly reduce the chance that an undiscovered malicious user or device will be able to exploit an environmental vulnerability. To enforce least privilege in ubiquitous interactions, we would like to assign and enforce the appropriate degree of access based on the type of interaction a device initiates.

3 Our Approach

Our own analysis of these problems has led us to a new abstraction for modeling ubiquitous interactions based on the concept of a *sphere of influence*. Politically, a sphere of influence is the geographical region within which a nation is influential.

Socially, we each have our own spheres – the locales we frequent, the organizations we associate with, and our set of friends, family, and acquaintances. The many relationships we participate in affect the others, often in subtle and unseen ways. Abstracting this notion to ubiquitous computing, a ubiquitous sphere of influence is the set of entities over which a given context can influence interactions. A given context can be geographical, such as a room in a building, or it can be based on some other metric, such as membership in a group, or an inherent property of an entity. A given entity may participate in many such spheres, and spheres may be involved with relationships with other spheres. An example would be the hierarchical structure of a building, where the sphere of the building would include the sub-spheres of rooms.

This abstraction provides a clean demarcation of contexts; additionally, the spheres serve as containers for policy. Entrance into a sphere, whether a social group or a physical location, implies accepting applied constraints and granted privileges extended by the sphere's policy. These constraints and privileges may be based on policy local to the immediate context, or alternately inherited from a sphere higher in a hierarchy.

We believe this model captures the complex, dynamic relationships present in ubiquitous environments. Relationships between entities are represented through linkages between spheres. These can include parent-child or peer relationships, and may represent constraints, extended privileges, or semantic linkages that serve to provide a form of electronic annotation. As devices and agents move, regroup, and change properties, the associated spheres must change accordingly—merging, splitting, or coalescing. This structure thus provides a natural abstraction for managing information and control flow in mobile and highly dynamic ubiquitous environments.

The sphere serves to organize policy and privilege within a scoped domain. However, we need to address integrity, access control, and privilege management concerns within the sphere and among interacting spheres. In the physical world, when people organize into political and social units, organization occurs in stages. The first stage is one of examination. When an individual is introduced to a group, a decision for admission is made based on information gleaned about this person's background. Such data-gathering may occur in the form of a simple introduction and a handshake, a background check, or a pass through an airport's metal detector. Upon acceptance, negotiation of the terms of membership must begin. These terms are a contract that specifies what is expected from the new member and what is to be provided them. After negotiating, an identity card is produced—a credential that identifies the new member. The group then takes on a management role in helping members use members-only services.

4 Design of a Framework for Secure Ubiquitous Interactions

The concept of *spheres of influence* is the unifying abstraction behind our approach towards designing a secure ubiquitous computing infrastructure. Each sphere is a cluster of entities, such as devices, environments or other spheres, which has a set of policies and services associated with it. Within the sphere's context, the entities are governed by the local policy.

Using this paradigm, we will extend our investigations into three areas: decontamination, negotiation, and policy-guided connection management.

4.1 Decontamination

In a ubiquitous environment, it is essential that devices operating within the sphere meet high integrity standards. Our proposed solution provides a framework for monitoring of device behavior and examination of device state.

To perform integrity checks on a device, or to track its behavior, the infrastructure needs to obtain full knowledge of its OS/BIOS, applications, I/O, data, transmission characteristics, resource use, and a specification of required and exported services.

To ensure that devices are safe to operate, they must go through a decontamination phase in which a *security manager* (SM) in the local sphere runs various tests. A simple check for most devices would be a system scan for viruses, worms or other suspect code or vulnerabilities. In the event that malicious code or a vulnerability was found, the device could be quarantined until it rectifies the problem. The environment could possibly aid the device, by providing signed software updates or repair software.

To track system state updates effectively, and for decontamination ease, checkpointing and logging must be performed. Logs can be used to restore a device to a safe previous state, in case as infection is detected. Checkpointing and monitoring are essential for the SM to know at all times which devices and services are being used, how long they have been in use and what they are currently doing, without the device perceiving any noticeable change in the quality of service.

4.2 Policy Negotiation

After an entity is decontaminated, it must negotiate policy with the policy manager (PM) of the sphere it enters. The sphere has a set of policy rules that govern device interactions. The entity has a set of requirements which represent the resources or services necessary for normal operation. The entity also offers services that are available to others within the sphere. Policy negotiation results in the entity being granted permission to access resources within the sphere, in the form of capabilities or insertion into access control lists (ACLs).

Policy rules are constraints imposed by the sphere on member entities and privileges that it grants to them; they can be of temporal, locality, communication, content type or cryptographic nature. These rules, and device requirements and services can be expressed using a formal notation or algebra that can be expressed within the framework of first-order logic.

Policy specification must take into account the changed context in case of sphere interaction. For instance, if the sphere arrangement is hierarchical, each sphere could inherit the policies of their parents. As a general rule, policy conflict resolution should obey the principle of *most-restrictive policy*.

4.3 Policy-Guided Connection Management

After policy negotiation is performed, the sphere's connection manager (CM), which acts both as a service discovery service and a session mediator, builds a *plan* to enable devices to interact with each other. This plan specifies a set of connections between devices, as in a producer-consumer relationship; it must take sphere policy and system context into account. Plan-building, or connection management, is usually activated in an on-demand fashion, whenever a device issues a service request.

After determining a connection plan, the CM also needs to validate the low level credentials that each device needs to initialize a connection with; as mentioned in the above section, these credentials could be in the form of ACLs or capabilities.

There are multiple techniques that can be used to attack the actual planning problem. *Template planning* statically determines a plan based on a template that incorporates all the policy and security constraints of the environment; the result may be far from optimal. *Brute-force search* considers all resource allocation possibilities, and chooses the best one from the entire search space. For large-scale environments, *heuristic-based planning* could strike the best balance.

5 Existing Approaches

Many projects have investigated infrastructure for ubiquitous computing [Brumitt2000, Brooks1997, Kindberg2002, Román2002]. These projects have contributed to the development of UPnP [UPnP] and other commercial ubiquitous computing projects. Traditional system security relies upon user-level authentication and access control to restrict access to individual services or machines. The highly dynamic and unpredictable ubiquitous computing environment requires a more flexible, distributed solution, capable of dealing with dynamic relationships and policies, than traditional computer system security.

Support for dynamic, extensible control must be substantially automated, and there are no reasonable solutions available today. Additionally, since this infrastructure is intended to be easily deployed in common environments, necessary administration must be minimal, and the human-device interface, when necessary, must be intuitive and easy to use. No existing system attempts to address all of these concerns as we do. However, there are several interesting and related systems that address isolated portions.

CoolTown [Kindberg2002] is a Hewlett Packard Research project that is exploring ways of enabling smart spaces. CoolTown builds on the Taligent [Postel1995] paradigm of "People, Places, and Things" by extending web services into the physical environment and enhancing physical objects with web content. It provides infrastructure support to enable encoding of location or context information in URLs for mobile devices. Their security mechanisms are interesting, but fairly specific to their model of providing web-enabled spaces.

Universal Plug and Play [UPnP] assists in automated infrastructure-based device interaction but its relevance is limited to home networks. UPnP is essentially a client-server system consisting of devices and control points (CP). The CP accesses the devices by remote procedure calls (RPCs), and keeps an Access-Control List for maintenance of security.

UPnP uses a security console (SC) to centrally handle security-related operations for devices. The SC may pass any information that it has to any other SC or CP as it sees fit. This approach to security has some scaling problems and, in the face of high mobility, may not provide sufficient security.

UPnP does not perform automated policy management, assuming instead that human interaction with the SC will determine what device interaction can occur. This approach will have difficulties with high scale and complex interactions that are not foreseen by the human controller.

Role-based Access Control for Ubiquitous Computing (RBAC) is used by MIT for their Intelligent Room project [Tuchinda2002]. In RBAC, users are assigned one or more roles which specify their permission set. Roles are hierarchical, and specialized roles can be created by subclassing a high-level role. RBAC is flexible enough to allow exceptional needs for permissions outside a user's current role. We believe our approach will allow greater security through closer adherence to the principle of least privilege, with the added benefit of increased flexibility.

Centaurus [Kagal2001] provides an infrastructure and communication protocol for interoperation of heterogeneous mobile devices and typical smart spaces consisting of communication managers, service managers, clients and services. The basic Centaurus infrastructure provides security by combining the ticket access control approach of Kerberos and distributed trust to determine access policies. Vigil [Kagal2002b], an extension of Centaurus, is similar to our model as far as local environment management is concerned. Certificate controllers generate and assign digital certificates to entities that request them while a security agent maintains trust information for validation and revocation purposes.

Vigil differs from our model in various aspects. It does not suggest integrity methods similar to our device analysis and decontamination model. Interaction between smart spaces is not described, other than the fact that service managers are arranged hierarchically. It associates a static set of rights with a role a device can assume, which does not allow devices to dynamically negotiate for privileges.

Several ongoing research projects involving trust models for ubiquitous computing seem extremely promising. The SECURE project [English2002] has developed a formal trust model with a fine granularity of trust levels; these values change based on perceived success or failure of interactions. Shankar and Arbaugh [Shankar2002] use a continuum of trust and define a unified trust model that combines identity-based and context-based models. This research is largely orthogonal to ours, and complementary to our integrity analysis phase.

6 Conclusion

Ubiquitous computing environments present difficult security challenges to systems designers. The complex, dynamic relationships in ubiquitous environments exacerbate traditional security problems, and require new solutions and techniques. This paper has sought to outline some of the difficult challenges in securing ubiquitous computing. Specifically, we have examined problems of integrity, policy, and session management. Additionally, we have proposed a rich model, based on the notion of a *sphere of influence,* to represent relationships between entities. This model is core to an integrated approach to securely manage these complex interactions, focusing on integrity, policy management and enforcement, as well as session mediation. We believe these techniques are widely applicable to problems that will arise in ubiquitous computing environments.

References

[Brooks1997] Brooks, R. "The Intelligent Room Project." *Proceedings of the 2nd Intl. Cognitive Technology Conference,* 1997, Aizu, Japan.

[Brumitt2000] Brumitt, B., Meyers, B., Krumm, J., Kern, A. And S. Shafer. "EasyLiving: Technologies for Intelligent Environments." *Proceedings of the Intl. Conf on Handheld and Ubiquitous Computing 2000*. pg. 12-27.

[English2002] English C., Nixon P. "Dynamic Trust Models for Ubiquitous Computing." Workshop on Security in Ubiquitous Computing, UBICOMP 2002, Göteborg Sweden.

[Kagal2001] Kagal, L., Korolev, V., Chen, H., Joshi, A., and Finin, T. "Centaurus: A Framework for Intelligent Services in a Mobile Environment." 21st International Conference on Distributed Computing Systems Workshops (ICDCSW '01), April 16 - 19, 2001, Mesa, Arizona.

[Kagal2002a] Kagal, L. "Rei: A Policy Language for the Me-Centric Project." Hewlett Packard Tech Report HPL-2002-270, November, 2002.

[Kagal2002b] Kagal, L., Undercoffer, J., Perich, F., Joshi, A., and Finin, T."A Security Architecture Based on Trust Management for Pervasive Computing Systems." In Proceedings of Grace Hopper Celebration of Women in Computing 2002.

[Kindberg2002] Kindbert, T. et al. "People, Places, Things: web presence for the real world." In *Mobile Networks and Applications*. Vol 7, issue 5. October 2002.

[Postel1995] Postel, M. and Cotter, S. *Inside Taligent Technology.* Addison-Wesley, 1995.

[Román2002] Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R. and Nahrstedt, K. "Gaia: A Middleware Infrastructure to Enable Active Spaces." *IEEE Pervasive Computing*, pp. 74-83, Oct/Dec 2002.

[Shankar2002] Shankar N., Arbaugh W. "On Trust for Ubiquitous Computing." Workshop on Security in Ubiquitous Computing, UBICOMP 2002, Göteborg Sweden.

[Tuchinda2002] Tuchinda, R. "Access Control Mechanism for Intelligent Environments." *Bitstream, the MIT Journal of EECS Student Research*. Spring 2002.

[UPnP] http://www.upnp.org.

[Weiser1991] Weiser, M."The Computer for the 21st Century." *Scientific American* 265(30), pg. 94-104, 1991.