Enabling Secure Ubiquitous Interactions

Kevin Eustice, Leonard Kleinrock, Shane Markstrum,
Gerald Popek, Venkatraman Ramakrishna, Peter Reiher
Laboratory for Advanced Systems Research
Department of Computer Science
University of California, Los Angeles, CA 90095
{kfe,lk,smarkstr,popek,vrama,reiher}@cs.ucla.edu

Abstract. Future environments will require devices to be automatically and safely configured to perform important tasks. Security concerns based on known vulnerabilities within the Internet make it clear that any widely deployed computing infrastructure must be designed from the start with substantially more security. However, existing security models cannot handle the highly dynamic relationships between applications, devices, and environments fundamental to ubiquitous computing. We need new techniques to address these unique demands. We propose a new paradigm for creating and maintaining safe, ubiquitous computing environments, based on the novel idea of organizing related devices into *spheres of influence*. This concept captures both geographic and semantic groupings. Spheres are used to encapsulate policy and provide well-defined boundaries for interactions. Intra-sphere interaction requires policy-based negotiation among principals.

1 Introduction

We stand at a technological watershed; gazing ahead, we see a world populated with intelligent devices that offer an immense amount of computational power and a rich communications infrastructure [Weiser1991]. Within the near future, our intelligent homes and offices will be filled with smart appliances and ubiquitous computing support. We will be able to seamlessly change contexts, and the environment will automatically adapt to our presence [Kleinrock2001]. These technologies are extremely exciting—however, this future may not be so idyllic. For these new technologies to emerge from research labs and be successfully deployed, a solid foundation of security and safety must first be in place. Existing research has focused principally on developing interesting applications and novel infrastructures to manage mobile users and devices, leaving open the question of how to achieve system security, manage domain-specific policy, and handle complex access-control issues in an environment composed of a heterogeneous mix of devices, infrastructures, individuals and applications.

Merely enabling communications between entities is insufficient. To avoid the kinds of difficulties being experienced by today's Internet, we must establish a framework for extensible secure ubiquitous computing at an early stage.

This paper proposes a model based on *spheres of influence* for representing the complex dynamics of ubiquitous interactions, as well as techniques to analyze and manage the flow of information and control in such environments. It also proposes techniques to assess the appropriate level of required access for an entity within a ubiquitous environment. Additionally, *least privilege* must be maintained for all interactions. This requires that entities be granted the minimum set of privileges necessary and sufficient 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. 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 homes attacking household wireless networks—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 a 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 and possibly malicious intruders. Additionally, we need mechanisms to update or repair vulnerable and exploited devices. Devices may need to be screened for viruses or Trojan horses before entering an environment, or dynamically updated with software patches. It is ultimately up to the ubiquitous environment to decide what integrity requirements to place on entities that wish to receive service. These mechanisms enable the environment to make that choice.

Device and network integrity in a ubiquitous environment is a continuing concern. New vulnerabilities are discovered daily, and necessary firmware and software updates are frequently released. This implies that ongoing maintenance is necessary to keep local entities up to date, and that this maintenance is part of the functionality of the ubiquitous infrastructure.

2.2 Policy for Ubiquitous Computing Environments

A second concern is policy management in ubiquitous environments. Policy specifies environmental and service-specific behaviors and constraints on entity interactions. Examples of policy include spatial or temporal constraints on access to services, integrity requirements, and content restrictions. 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 must be configured in isolation. As the number of devices requiring configuration grows, the situation becomes an administrative nightmare. This is ultimately unworkable.

Devices in the ubiquitous environments must 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 just provide policy information to devices; a framework must provide mechanisms for enforcing policy.

There is a need for further development of policy languages that are appropriate for ubiquitous computing environments. Typically, policy languages have been designed for domain-specific applications. It is desirable to have a more general language that can describe security constraints, as well as describe other types of desirable interactions, such as environmental responses. 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 heterogeneous distributed systems. 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, as no system is without vulnerabilities. If the set of privileges granted to a device exceeds the minimum set necessary to accomplish a task, the device is much more capable of exploiting a yet-undiscovered vulnerability.

Privilege management is used as a last line of defense to defend against the intruders we cannot detect and the vulnerabilities we cannot find. By assigning and enforcing *least privilege* semantics to ubiquitous interactions, it is possible to 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, it is necessary to assign and enforce the appropriate degree of access based on the type of device-initiated interaction.

3 Our Approach

Our own analysis of these problems 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 geographic region within which a nation is influential. Socially, we each have our own spheres—the locales we frequent, the organizations with which we associate, and our set of friends, family, and acquaintances. Many of the relationships in which we participate affect our other relationships, 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 geographic, 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 in relationships with other spheres. An example would be the hierarchical structure of a building, where the sphere of the building would include the subspheres of rooms.

This abstraction provides a clear 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 the constraints and privileges offered 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 sphere of influence 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 begins. 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 toward 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 extend our investigations into three areas: decontamination and quarantine, negotiation, and policy-guided connection management.

4.1 Decontamination and Quarantine

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 device behavior and examining device state.

In general, it may not be feasible to thoroughly examine devices as they enter and leave ubiquitous environments; however, in practice there is a family of practical solutions we can apply to increase the security of a sphere. There are numerous tradeoffs that can be made here between assurance and flexibility. It is important to note that privacy and integrity are not mutually exclusive. There are a number of techniques that can be used to perform examinations without disclosing confidential information.

To ensure that devices are safe to operate, they must pass through a decontamination phase in which the local sphere's *security manager* 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 is found, the device could be quarantined until the problem is rectified. 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 an infection is detected. Checkpointing and monitoring are essential for the security manager to determine which devices and services are being used, how long they have been in use, and what they are currently doing.

4.2 Policy Negotiation

After an entity has gone through this integrity examination, it must negotiate policy with the *policy manager* of the sphere it enters. The sphere has a set of policy rules that govern device interactions. The entity has a set of requirements that 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 receiving 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, spatial, communication, content, cryptographic nature. These rules, device requirements, and services can be expressed using a formal algebra based on first-order logic.

Policy specification must consider changed contexts when spheres interact. For instance, if the sphere arrangement is hierarchical, each sphere could inherit the policies of its parents. In the absence of more sophisticated conflict resolution techniques, entities should obey the principle of *most-restrictive policy*.

4.3 Policy-Guided Connection Management

After policy negotiation is performed, the sphere's *connection manager*, which acts both as a service discovery mechanism and a session mediator, must build a *plan* to enable devices to interact with each other. This plan specifies a set of connections between devices, based on sphere policy and system context. Plan-building, or connection management, is usually initiated in an on-demand fashion whenever a device issues a service request.

After determining a connection plan, the connection manager also needs to validate the low-level credentials with which each device needs to initialize a connection; 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 [Reiher2000]. *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* will likely strike the best balance [Rudenko2000].

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 on user-level authentication and access control to restrict access to individual services or machines. The dynamic and unpredictable ubiquitous computing environment requires a more flexible, distributed solution that can deal with changing relationships and policies.

Dynamic, extensible control must be substantially automated, and there are no reasonable solutions available today. Additionally, since this infrastructure is intended for easy deployment in common environments, the administrative burden 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.

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. The control point accesses the devices by remote procedure calls (RPCs) and keeps an Access-Control List for maintenance of security. UPnP uses a security console to centrally handle security-related operations for devices. This approach to security has some scaling and mobility problems. UPnP does not perform automated policy management, assuming instead that human interaction with the security console 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 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 the 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 complementary to ours and would strengthen our integrity analysis mechanism.

6 Conclusion

The technological marvels of tomorrow are the research challenges of today. Ubiquitous computing environments present many difficult security challenges to systems designers. By not addressing these problems, we are offering our homes and offices up freely to potential attackers. This paper has outlined some of the difficult challenges in securing ubiquitous computing. Specifically, we have examined problems of integrity, policy, and privileged 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 secure management of these complex interactions, focusing on integrity, policy management and enforcement, as well as session mediation. We believe these techniques are widely applicable to the types of problems that will arise in the ubiquitous computing environments of the future, and are critical to ensure their safety.

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 Shafer, S. “EasyLiving: Technologies for Intelligent Environments.” *Proceedings of the Intl. Conf on Handheld and Ubiquitous Computing 2000*. pp. 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.

[Kleinrock2001] Kleinrock, L. "Breaking Loose." *Communications of the ACM*, Vol 44, No. 9, pp 41-45, September 2001.

[Reiher2000] Reiher, P., Guy, R., Yarvis, M., and Rudenko, A. "Automated Planning for Open Architectures." *Proceedings of OPENARCH 2000*, Tel-Aviv, Israel, March 2000.

[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.

[Rudenko2000] Rudenko, A. and Reiher, P. "Experience with Automated Planning for Panda." UCLA Tech Report CSD-TR-010041, November 2000.

[Shankar2002] Shankar N. and 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), pp. 94-104, 1991.