Security Issues with Pervasive Computing Frameworks

Michael Collins[1], Simon Dobson[2], Paddy Nixon[2]

Systems Research Group, School of Computer Science and Informatics,
UCD Dublin, Ireland
[1]michael.collins@comp.dit.ie
[2]{simon.dobson, paddy.nixon}@ucd.ie
[1]http://www.comp.dit.ie/mcollins
[2]http://www.csi.ucd.ie/Staff/AcademicStaff/{sdobson, pnixon}

Abstract. The deployment of pervasive computing systems inevitably raises security concerns. The ability of such systems to synthesise partial information and react to situations without explicit user interaction, confirmation or consent means that issues of trust, safety and privacy become exceptionally important. We examine the general issues inherent in any pervasive computing system, focusing on the hazards that emerge from the interactions between different forms of context. We apply this analysis to one particular context-aware infrastructure, and use this to derive some important research topics for future investigation.

1 Introduction

Many people using Internet services have done so from the comfort of their home, office or a place they often frequent. Reducing potential security vulnerabilities is one way that service providers try to satisfy the concerns of their customers. Recently, many users avail of services that allow mobility such as sending and receiving email on their mobile phone or accessing the Internet on their Personal Digital Assistant (PDA). As computing becomes more pervasive, people expect to access services and information at anytime and anywhere. This is leading to the development of pervasive computing.

In addition to the security problems applicable to the Internet paradigm e.g. eavesdropping and phishing [1], there are particular problems in providing adequate security in pervasive environments. Many of these issues arise because such systems are often both distributed and ad-hoc in nature [2]. Adding security to such open models presents challenges at many levels. For example, how does one decide whether a person who does not work in an office but has access to it as a consultant can use certain services? [3]

What can be done to minimize the security risks associated with pervasive computing systems? One way is to identify risks in the framework itself. Using this knowledge, pervasive computing frameworks can be redesigned to eliminate or reduce such security vulnerabilities and thereby diminish the potential security threats that accompany services they support. This paper discusses possible security vulnerabilities in a

pervasive computing framework and suggests ways in which these issues can be addressed.

Section 2 discusses the general security issues with a pervasive computing framework. Section 3 introduces an example of a pervasive computing framework called ConStruct [4] and discusses known security vulnerabilities with it making reference to an application that it supports. Section 4 discusses research topics for future investigation. Section 5 concludes the paper.

2 Security issues

Many attempts have been made to apply traditional security concepts and solutions to pervasive platforms. However, in most cases, a lot of modifications are needed in order for the security infrastructure to fit within the pervasive framework leading to a high level of risk of introducing new breaches. A generic security framework is needed [5]. One way to minimise the security issues with pervasive computing frameworks is to identify them in the early stages of their development. These issues include:

2.1 Reliability

Pervasive systems expose a larger attack surface with many points of failure in comparison to traditional computing environments [2]. If people depend upon pervasive systems to mediate day-to-day activities, such systems will quickly become mission-critical. They need to be robust, dependable, and always available. This is a very difficult problem for a pervasive computing framework to address. Due to the wide variety of computing technologies that entities will use, the risk of introducing security issues that may threaten the reliability of services will increase.

2.2 Trust

English et al. state that the infrastructure that supports a pervasive computing system introduces new security challenges not addressed in existing security models, including in the domain of trust management [6, 7]. The issue of trust will arise when an entity, such as a PDA, mobile phone or laptop computer that is unknown to other entities offers services. Entities offering services may have an established connection history and are trusted. An entity connecting to a pervasive environment for the first time will have no historical records on which to base a measure of trust. Such an entity may act maliciously and try to disrupt services being offered or it may be genuinely trying to avail of a specific service. The question of trust arises and whether existing members of the environment trust the intentions of this new entity. For example, two new entities (A and B) that are connected to the pervasive environment may wish exchange data. Do they trust each other?

2.3 Data access

Entities that are supported on a pervasive computing framework may provide services that collect data. This data may be private and confidential to a user. Entities that collect data may be able to exchange it freely without any authorisation or authentication [8]. The user needs to have control over who has access rights to this data. There is also the problem of data exchange between entities without encryption. This makes it possible for sensitive data such as medical records and credit-card numbers to be intercepted and accessed by an unauthorised entity using eavesdropping, snooping, etc.

2.4 Malicious attacks

A common form of malicious attack is a Denial of Service (DoS) attack. A DoS is an attack with the purpose of preventing legitimate users from using a specified network resource such as a website, web service, or computer system [9]. A Distributed Denial of Service (DDoS) attack is a coordinated attack on the availability of services of a given target system or network that is launched indirectly through many compromised computing systems [10]. The threat of a deliberate attack by an entity connected to a pervasive computing framework will exist. Any entity can act maliciously and may deliberately try to access a service provided by another entity that it is not permitted. A DoS attack can not only cause the innocent entity to malfunction but may also introduce other problems such as severe network latency, scalability problems when other entities try to connect to the framework and service unavailability.

2.5 Information propagation

When data is transmitted between entities, it is possible that it may be altered or corrupted accidentally or maliciously. It is important that the integrity of the data in a pervasive environment is maintained. If the data is altered, the entities may not have any mechanism to detect this.

2.6 Recourse

Ranganathan states that because pervasive systems will mediate everyday physical activity, technical mechanisms to facilitate recourse must be built-in from the start [2]. When one entity provides a service to another, the problem exists when one of the entities reneges before a successful termination. A pervasive computing framework should have mechanisms to deal with this.

In the next section we introduce an example of a pervasive computing framework called ConStruct. We use this to motivate discussion of how security issues impact services that are supported in a pervasive environment.

3 ConStruct - A pervasive computing framework

ConStruct is a component-based infrastructure for the collection, processing, and distribution of context information [4]. The problem space that ConStruct tries to address is the difficulty in the collection, reasoning about and distribution of context information in an ad hoc environment.

There are a number of security vulnerabilities inherent with ConStruct. Applications that are supported by a pervasive computing framework are impacted by these vulnerabilities.

The user activity monitor is an application that monitors a user's keystrokes and provides this data to ConStruct. When the activity monitor senses activity on the keyboard, mouse, etc., no feedback is given to the user indicating this detection. For example, if a user interacts with an entity that is supported by ConStruct, all keyboard strokes may be monitored, recorded and exchanged with one or more entities without express permission. All entities can therefore access this data and do not require authorisation or authentication. Data is freely accessible to all nodes and ConStruct does not provide any mechanism to restrict private and confidential information.

Presently, all entities running the activity monitor on ConStruct have no means of denying the infrastructure of certain data, for example, certain input actions taking place. If an entity desired to restrict the data it provides to ConStruct such as keyboard activity and not mouse activity, there is no mechanism to enable this. This data can be transmitted between entities on ConStruct with no means of checking the integrity of the data at the destination. It is possible that data may have been altered before reaching the destination entity and ConStruct makes no attempt to check this.

Like most pervasive computing frameworks, ConStruct allows entities to connect and disconnect freely and does not require authentication. ConStruct does not assume that an entity may act maliciously and therefore does not monitor for any deliberate attack on an innocent entity. An entity acting maliciously may initiate a DoS attack and it will be the responsibility of the innocent party to detect the attack and act accordingly. ConStruct makes no attempt to counteract the attacking entity and take appropriate action such as disconnect the node.

In the next section we discuss possible research areas to address the security vulnerabilities in ConStruct and other similar pervasive computing frameworks.

4 Research areas

ConStruct is one example of a pervasive computing framework. In order to address the security issues inherent to similar frameworks, there are a number of initial security requirements to integrate into a framework's architecture. Applications that are supported by these should inherit these security features. These may include but are not limited to the following:

4.1 Group mechanisms

All entities availing of services supported by a pervasive computing framework are assigned to a specified group. Each group is assigned permissions with specified rights associated with a service. These rights can be full, partial or none. We will investigate the process of deciding group members and what data are they permitted to access.

4.2 Trust levels

Each entity availing of services supported by a pervasive computing framework will be assigned with an initial trust level. This trust level is dynamic and can be changed at any time. Before entities exchange data, their trust level is checked for their 'trustworthiness'. This may take the form of some arithmetic value, rank, or some other mechanism. Data exchange will occur depending on an entities level of trust. As entities become 'trustworthy', their trust level can be increased thereby allowing the use of extra services.

Another mechanism might involve a trusted third party entity or Trust Broker. For example, entities A and B wish to communicate but do not trust each other. However, if one of these entities trust a third entity (C) who itself trusts both entities, entity C may recommend that both A and B are trusted entities and that they can communicate directly (and may specify certain constraints in their communication).

The use of roles is similar to the concept of Trust. A pervasive computing framework can assign roles to entities where each role is assigned certain privileges. For example, entities assigned the role of 'Master' have full rights to access all data exchanged on the framework. Entities assigned the role of 'Child' have restrictions in accessing certain data and services.

4.3 Service publishing

When an entity connects to a pervasive computing framework, it may wish to offer one or more services to other entities. Publishing this information may do this and is a mechanism of informing entities connected to the framework what is available for use. However, there may be instances where entities do not wish to provide their service to certain other entities. To address this issue, the framework should have a mechanism of restricting the services available to all entities and only specified entities should be told about services being offered. If there are no restrictions imposed, the framework should be free to inform all entities of certain services available for use.

4.4 Encryption

Data may be exchanged between entities running on a pervasive computing framework unencrypted. This makes it vulnerable to many security problems. To overcome

this, encryption should be applied to all data exchanged between entities. A pervasive computing framework should apply the use of certain encryption algorithms to encrypt and decrypt data as it passes between entities. However, the interface between the framework and the entity where data is transferred unencrypted needs to be addressed. This is a security vulnerability. One solution to this problem may be the encryption and decryption of data at the entity itself.

4.5 Data monitoring

A pervasive computing framework needs to be able to monitor communication between entities. One method to do this is to generate statistical data of communication passing between entities. If an entity sends data exceeding the normal statistical rate for the time of day, type of data, etc., the framework could block this entity. The entity could then be questioned about its intentions and if verified, allowed to continue.

4.6 Information propagation

One way the framework can maintain the integrity of data as it propagates between entities is by calculating the bit-sum of each packet before being delivered. This can be compared to the bit-sum calculated by the original entity that sent the data. If there are any discrepancies, the framework can discard the data and possibly request the data to be sent again.

5 Conclusion

It is not possible to account for every possible security and privacy risk in pervasive computing systems. However, such systems can be designed to develop better security models and interaction techniques to prevent and minimize foreseeable threats [11].

The goal of a pervasive computing framework is to support services offered by entities in a heterogeneous environment. It is highly probable that most entities have little or no knowledge of other entities and the services they offer. This introduces important security issues especially when a new entity makes available a new service for the first time.

As a result of this 'unknown' factor introduced in this environment, there is a strong expectation for a pervasive computing framework to provide adequate security to all entities. The framework is transparent to these entities and as such they appear to communicate peer-to-peer. Unless entities are aware of appropriate security mechanisms, there may be reluctance for these to connect to the framework and avail of services on offer.

The importance of security should be a predominant factor in designing pervasive computing frameworks. By performing an analysis of the security issues early in the

design stage of a framework, security risks can be minimised. This may involve applying encryption to data as it is exchanged between entities. Another factor would involve a framework's ability to support the use of roles. We will investigate these and other security topics as part of our research into secure and predictable pervasive computing.

6 Acknowledgements

The work described in this paper was supported in part by a grant "Secure and Predictable Pervasive Computing" from Science Foundation Ireland.

References

1. Warwick Ford, Michael S. Baum: “Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption” 2nd Edition, 2001, Prentice Hall
2. Kumar Ranganathan "Trustworthy Pervasive Computing: The Hard Security Problems", Proceedings of the Second IEEE Annual Conference on Pervasive Computing, March 14-17, 2004.
3. Kagal, L.; Finin, T.; Joshi, A. "Trust-based security in pervasive computing environments", Computer Volume 34, Issue 12, Dec. 2001 Page(s):154 - 157
4. Graeme Stevenson, Lorcan Coyle, Steve Neely, Simon Dobson and Paddy Nixon "ConStruct - a decentralised context infrastructure for computing Environments", IT&T Annual Conference. 2005.
5. Ghita Kouadri Mostéfaoui "Security in Pervasive Environments, What's Next?", Security and Management Journal, 2003
6. Colin English, Paddy Nixon, Sotirios Terzis, Andrew McGettrick and Helen Lowe "Security Models for Trusting Networked Appliances", 5th IEEE International Workshop on Networked Appliances, 2002
7. Colin English, Paddy Nixon, Sotirios Terzis, Andrew McGettrick and Helen Lowe "Dynamic Trust Models for Ubiquitous Computing Environments", First Workshop on Security in Ubiquitous Computing at the Fourth Annual Conference on Ubiquitous Computing (Ubicomp2002), October 2002
8. P. A. Nixon, W. Wagealla, C. English, S. Terzis "Security, Privacy and Trust Issues in Smart Environments", John Wiley & Sons, 2004, Chapter 11.
9. David Karig and Ruby Lee, "Remote Denial of Service Attacks and Countermeasures," Princeton University Department of Electrical Engineering Technical Report CEL2001-002, Oct 2001
10. Stephen M. Sprecht, Ruby B. Lee "Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures", Proceedings of the International Workshop on Security in Parallel and Distributed Systems, 2004
11. Lalana Kagal, Jeffrey L Undercoffer, Filip Perich, Anupam Joshi, Tim Finin "A Security Architecture Based on Trust Management for Pervasive Computing Systems". Proceedings of the Grace Hopper Celebration of Women in Computing, 2002