ABSTRACT

Title of dissertation: Shrink-Wrapped Security:
Tightly Coupling Situation and Security

Gleneesha Myra Johnson Williams
Doctor of Philosophy, 2014

Dissertation directed by: Professor Ashok Agrawala
Department of Computer Science

The mobile workforce, which consists of employees that do not have one fixed place of work and are linked to a corporate base using a mobile computing device, is expected to grow to 75% of the total United States workforce, or approximately 212.1 million people, by 2015 [28]. Advances in technology, such as the increasing abundance of portable computing devices and the prevalence of wireless broadband, combined with the fact that more companies are allowing employees to use their own devices to access the enterprise, create an environment in which these workers can access corporate resources anytime, anywhere, with a myriad of devices having varying configurations. Having ubiquitous access to resources has its benefits, like increased productivity, but also creates unique challenges to ensuring appropriate security. Traditional approaches to security are not suitable for this emerging computing environment, because they are based on assumptions that no longer hold, such as well-defined situations, consistent configurations, and static contexts. For this reason, these approaches typically base security decisions on statically assigned

attributes like identity or role. In the highly dynamic computing environment of mobile workers, context-aware security, in which context is utilized to allow security to adapt to the current situation, is essential. This dissertation presents our efforts to address the mismatch between traditional, context-insensitive security and this emerging dynamic computing environment with a novel security paradigm, shrink-wrapped security, in which a tight coupling is provided between a user's current situation and security. It features the following:

- A novel security paradigm, shrink-wrapped security, which involves utilizing context to tightly fuse a user's situation and security.

- A usable definition of security-relevant context, along with goal-oriented guidelines and a corresponding taxonomy to facilitate the systematic identification of contextual attributes that are most pertinent to a security service.

- A context acquisition and management framework to facilitate the development and use of shrink-wrapped security services for the mobile workforce. The layered architecture of this framework supports secure context acquisition, utilization, and monitoring by security services and was designed with the resource constraints of mobile devices in mind.

- An approach based on logic programming to practically incorporate the use of security-relevant context into the security policies that govern security services. This technique is aligned with the shrink-wrapped security concept of utilizing a comprehensive set of relevant context, while remaining practical and manageable by abstracting relevant contextual attributes to a security level associated with the objectives of a security service.

- The implementation and evaluation of shrink-wrapped access control, which serves as a practical demonstration of the feasibility of shrink-wrapped security.

Shrink-Wrapped Security: Tightly Coupling Situation and Security

by

Gleneesha Myra Johnson Williams

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy
2014

Advisory Committee:
Professor Ashok Agrawala, Chair/Advisor
Professor Carol Espy-Wilson, Dean's Representative
Professor Atif Memom
Professor Adam Porter
Professor Don Perlis

Dedication

To God; my husband and best friend, Alvester Williams III; my parents, Margie and Ronald Robertson, and Glenn Johnson; and all of my family and friends. Words cannot express my gratitude for your love, support, encouragement.

Acknowledgements

First, and foremost, I would like to thank God for His many blessings during my academic journey.

I am incredibly thankful for my husband and my parents. Their faith in me provided the encouragement to persevere though the challenging process of obtaining a Ph.D. I would also like to thank my in-laws, Carol Williams and Alvester Williams II for your reassurance and prayers. This journey would not have been possible without the support of my family and friends, and I am appreciative for all of them.

I would like to thank my advisor, Dr. Ashok Agrawala, for encouraging and believing in me and my research. I would like to thank my committee members, Dr. Atif Memon, Dr. Adam Porter, Dr. Don Perlis, and Dr. Carol Espy-Wilson for their insightful comments and feedback.

I am grateful to my PROMISE family for being a wonderful support system, including, but not limited to: Dr. Johnetta Davis, Dr. Renetta Tull, Dr. Carol Parham, Dr. Wendy Carter, Gloria Anglon, Dr. Asha-Lateef Williams, Angel Miles, Dr. Charles Glover, Dr. Patrice Gregory, Dr. Sophoria Westmoreland, Dr. Sean Barnes, Erika Thompson, Wynsome Bryan, Dr. Geriel Ettienne-Modeste , Dr. Heather Holden, and Dean Christopher Jones.

I appreciate the Faculty, Staff, fellow graduate students, and friends that I worked with, in one way or another, while attending the University of Maryland, including, but not limited to: Dr. Christian Almazan, Dr. Neha Gupta, Dr. Matthew Mah, Preeti Bhargava, Morgan Kleene, Praveen Vaddadi, James Lampton, Tomas

Lampo, Dr. David Thaw, Dr. Shivsubramani Krishnamoorthy, Dr. Yi-Jung Lo, Dr. Paulo Shakarian, Dr. Alexandros Tzannes, Dr. Ben Shneiderman, Dr. Sandro Fouche, Leslie Milton, Adelaide Findlay, Fatima Bangura, Jennifer Story, Brenda Chick, Gwen Kaye, Dr. Eric Ditrich, Emely Martinez, Dr. Mugizi Rwebangira, Dr. Joel Branch, and Dr. Elodie Billionniere.

I am thankful for the guidance provided by my wonderful mentors, including, but not limited to: Dr. Yul Williams, Dr. Dwayne Rosenburgh, Dr. Brian Cohen, Dr. Cyntrica Eaton, Dr. Shaun Gittens, Dr. Juan Gilbert, Julina Edwards, and Dr. Legand Burge.

Finally, to all whose names I did not mention, charge it to my head and not my heart. I thank everyone who played a part in the successful completion of my doctoral studies.

Table of Contents

List of Figures

Chapter 1

Introduction

The mobile workforce, which consists of employees that do not have one fixed place of work, and are linked to a corporate base using a mobile computing device, is rapidly increasing. According to the International Data Corporation (IDC), the mobile workforce is expected to grow to 75% of the total United States (US) workforce, or approximately 212.1 million people, by 2015 [28]. The increasing abundance and convenience of powerful and portable computing devices, combined with the prevalence of wireless broadband availability, creates an environment in which these workers can access an array of corporate data and resources anytime and anywhere. The fact that more information technology (IT) departments are exploring allowing employees to use their own devices to access the enterprise (bring your own device or BYOD)means that they can also access these resources with a myriad of devices with varying configurations. Although advances in technology have enhanced the practicality of ubiquitous access to resources, which has its benefits like increased productivity, they also create unique challenges to ensuring appropriate security.

Traditional approaches to security are not appropriate for the emerging, dynamic computing environment of mobile users because they are based on assumptions that no longer hold. Traditional computer security mechanisms were developed when computing was typically in a static, stationary environment and are based on assumptions of relatively stable and well-defined situations, consistent configurations, and static contexts [27]. Accordingly, these mechanisms typically base security decisions on statically assigned attributes like identity or role. In the dynamic computing environment of mobile workers, users may: 1) use a variety of mobile computing devices with varying configurations; 2) connect over various networks; and 3) be in varying physical settings when requesting access to resources. In this computing paradigm, security should be based not only on the identity of the user, but also the user's context(e.g., co-location, network characteristics, and device characteristics), which can change frequently and rapidly.

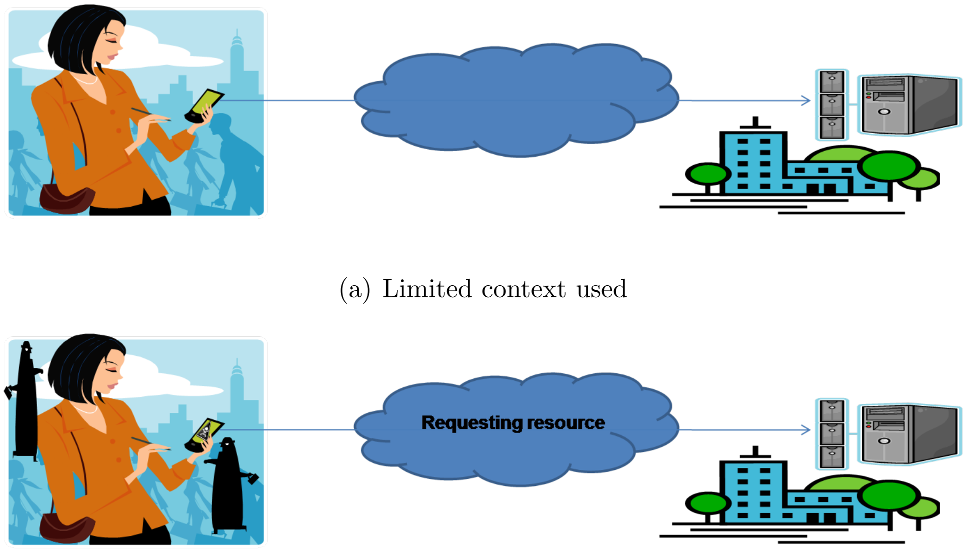
In order to address the security challenges of such mobile access, context-aware security, in which context is utilized to allow security to adapt to the current situation, is essential. Context is commonly defined as "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves" [22]. Consider the following scenario: Alice, a mobile worker who has a Secret security clearance, should be allowed to access documents classified at Secret or below via her mobile device in any room of one of her employer's buildings, as long as there are no individuals in the room without the appropriate clearance. Such situation-specific restrictions are not supported in traditional access control, so a security administrator would likely statically grant Alice access to these documents to avoid hindering her from performing her duties. It would be left up to Alice to ensure that she does not access classified documents outside of her employer's building or in the company of unauthorized individuals.

Allowing a user to have more privileges than necessary or appropriate for the current situation creates the potential for accidental and intentional abuse. For example, Alice may not know that her coworker, Bob, had his clearance revoked and that she should close classified documents when he enters the room. Likewise, Alice may be a disgruntled employee who intentionally lets unauthorized individuals view Secret information on her mobile device. By using context-aware security, the security administrator would be able to express the appropriate restriction in the security policy. In addition, a context-aware security mechanism would have the ability to enforce the policy by detecting when an unauthorized party has come into close proximity to Alice and temporarily revoke her access to Secret documents until the threat of unauthorized access is removed.

Several researchers have realized the importance of utilizing context in security, but there are still issues that need to be investigated. Although existing approaches consider context, we contend that they are still relatively static, based on factors such as limited context use, which has negative security implications, and infrequent context consideration, which limits the dynamicity of an approach [16, 11]. Many systems are vague about what context they consider (e.g., [61, 37, 59, 38]), but systems that are explicit about what context they utilize, primarily consider the traditional aspects, such as user location (e.g., [10, 27]). While this common form of context is certainly important to characterize the situation of a mobile user, it is often inadequate. The context utilized by a system, forms that system's model of a situation. Therefore, merely considering the most commonly used elements of context abstracts away important aspects of a situation that could be useful in decision-making, leading to a deficient view of a situation. Basing security-related decisions on such a view could be just as damaging as having no context at all, because it can lead to poor decisions based on an incomplete picture.

Consider the situation illustrated in Figure 1.1(a). Alice is at one of her company's satellite locations and requests access to a Secret corporate file. Her employer has started using context-aware access control, but currently only utilizes limited context, the location of the user. Since Alice is at an acceptable location, one of her employer's satellite locations, access to the resource is granted. If a more comprehensive set of context is used, then a more informed and appropriate decision could be made. For example, utilizing context of the user's computing device, access mechanism and surrounding environment can reveal that Alice has a virus on her device; that she is connecting over an insecure network; and that she is co-located with unknown people (see figure 1.1(b)). With all of this additional information, a more informed and appropriate decision of denying access while the user is in this insecure state can be made. Given the dynamic nature of mobile users' situations, even if Alice had been in an approved state at the time access was requested, and was accordingly granted access, it would have been critical to monitor her state and maintain control of the resource in case a change in her context warranted a revocation of access.

In order to meet the needs of the dynamic computing environment of the mobile workforce, our perspectives on security have to change, requiring the development of new approaches. We propose the notion of *shrink-wrapped security*, a security paradigm in which a tight coupling is provided between a user's current



(a) Limited context used

(b) Additional context used

Figure 1.1: Mobile worker requesting access to remote resource

situation and security. This is not possible when only limited context is utilized. In order to support shrink-wrapped security, a more comprehensive notion of context than what is currently used by context-aware security systems is necessary. In addition, it requires a highly dynamic approach. As such, with shrink-wrapped security, as the situation changes, the security changes also, and is therefore highly dynamic and constantly reflects the needs dictated by the situation. We use the term "shrink-wrapped" to convey the tight fit of security to what is appropriate, based on the current situation.

Various security services (e.g., cryptography, authentication, and access control) can be shrink-wrapped, but this research focused on shrink-wrapped access control. With shrink-wrapped access control, context is utilized to to dynamically adjust a user's permissions so that at any given time she only has permissions that are appropriate based on her current situation.

The purpose of this research was to explore and gain a greater understanding of our proposed paradigm, shrink-wrapped security, and to assess its feasibility through implementation. In order to study shrink-wrapped security, we first had to develop a solution that made it possible. This required addressing the need for a more comprehensive notion of security-relevant context, which entailed first defining exactly what constitutes such context. We used our definition to develop a taxonomy of security-relevant context and corresponding guidelines to facilitate the systematic identification of relevant context. We developed a framework to facilitate the secure acquisition of such context, and a technique to practically incorporate the use of such context in a security service. Finally, we used these tools and techniques to develop a prototype shrink-wrapped access control service. This prototype allowed us to more concretely explore our ideas.

Contributions of this dissertation include the following:

- A novel security paradigm, shrink-wrapped security, which involves utilizing context to tightly fuse a user's situation and security, to address the security challenges associated with the highly dynamic computing environment of the emergent mobile workforce.

- A usable definition of security-relevant context, along with goal oriented guidelines and a corresponding taxonomy to facilitate the systematic identification of contextual attributes that are most pertinent to a security service. These contributions deal with a key challenge of context-aware system development - identifying relevant context.

- A context acquisition and management framework to facilitate the development and use of shrink-wrapped security services for the mobile workforce. The layered architecture of this framework supports secure context acquisition, utilization, and monitoring by security services and was designed with the resource constraints of mobile devices in mind.

- An approach based on logic programming to practically incorporate the use of security-relevant context into the security policies that govern security services. This technique is aligned with the shrink-wrapped security concept of utilizing a comprehensive set of relevant context, while remaining practical and manageable by abstracting relevant contextual attributes to a security level associated with the objectives of a security service.

- The implementation and evaluation of shrink-wrapped access control, which serves as a practical demonstration of the feasibility of shrink-wrapped security.

The remainder of this dissertation is organized as follows. In chapter 2 we present an overview of the concepts that are fundamental to our work, including: traditional access control, context, general context-aware systems, and context-aware security. Chapter 3 highlights the various components necessary to realize shrink-wrapped security. We introduce our definition and corresponding guidelines to facilitate the systematic identification of security-relevant context. We also present an approach to practically and easily incorporate the use of such context into security services. Finally, we present a framework to facilitate the secure acquisition, monitoring and use of relevant context. Details about the implementation of our logical framework, and shrink-wrapped access control are contained in chapter 4, along with usage cases that describe real-world examples of how users interact with the system. In chapter 5 we present an evaluation of shrink-wrapped access control, and chapter 6 reviews related work. We conclude with chapter 7, which contains an overview of our research and future research directions.

Chapter 2

Background

In this chapter we present an overview of the concepts that are fundamental to our work, including: traditional access control, context, general context-aware systems, and context-aware security.

2.1 Traditional Access Control

Access control is an essential aspect of security as it aims to prevent unauthorized access to protected resources. It involves controlling access to resources *after* a user has been authenticated and granted access to a system. Access to various resources, such as files, sensitive business data, and business applications, can be managed. This is accomplished by controlling what subjects (such as users and processes) can access what resources, and what operations they can perform on these resources (e.g. read, write, execute).

There are various types of access control, including discretionary, mandatory and role-based. With discretionary access control, the owner of a resource determines what subjects should be able to access the resource and the range of operations they should be allowed to perform on the resource. With mandatory access control, access decisions are based on classifications that are beyond the control of the owner of a resource. Users are assigned a classification, which typically denotes their verified trustworthiness, and resources are assigned a classification, which typically denotes the level of damage that could result from unauthorized access. With role-based access control, users are assigned to roles based on various properties, typically functional responsibility within an enterprise, and permissions are associated with roles instead of individual users.

Figure 2.1 shows a basic access control model. A subject requests access to a resource. This request will identify the particular resource and an access operation appropriate for that type of resource. All access requests are mediated by an access control mechanism, which decides if the subject should be granted access or not based on an access control policy. The session with the access control mechanism ends after the access decision has been made.

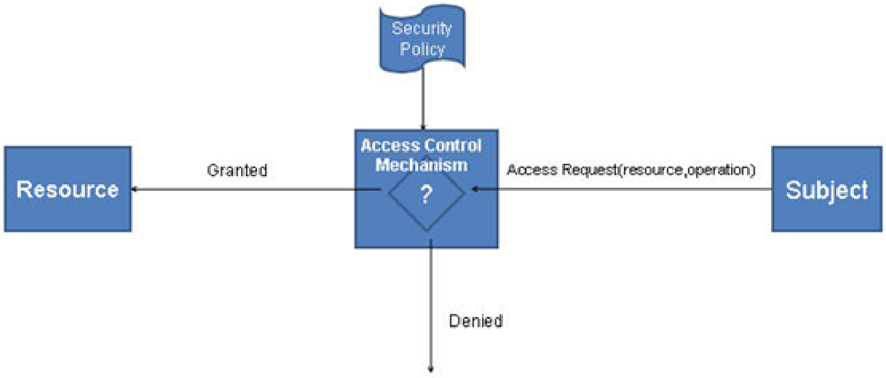


Figure 2.1: Basic access control model

An access control policy contains a set of rules which specify when access should be allowed or denied. Resource access has traditionally been based on statically assigned attributes such as identity, role, or classification. For example, a traditional rule for controlling access to a particular resource might be "If subject role = Administrator, then grant access". In this case, if the subject requesting access has an active role of Administrator, she will be granted access to the resource. Otherwise, she will not. When computing was confined to a static environment, this was acceptable, because there were not many other factors to take into consideration. However, much has changed, and users are increasingly computing in a dynamic environment with changing context. This context needs to be considered when making security-related decisions.

2.2 Context

Context has been defined in many ways by different researchers. We have adopted the common and well accepted definition from Dey et al. [22].

Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves.

A contextual attribute is a measurable context primitive (e.g., a user's location), and context is the full set of contextual attributes that comprise the situation of an entity [18].

Two important words from Dey's definition of context are "characterize" and "relevant." Characterization of a situation requires an examination of everything related to the situation, not just common elements such as user location and time. A very large amount of information may have to be used to characterize the situation of an entity, but only certain characteristics are relevant for a particular application.

2.3 Context-aware Systems

Context-aware systems can discover and utilize context to adapt their behavior based on the current situation [48]. Context may be supplied from a variety of sources, including: sensors embedded in a computing device, external sensors in the environment, a context providing service, and system state.

Mark Weiser's seminal paper, "The Computer for the 21st Century", has greatly influenced the field of context-aware computing. This paper explains his vision of Ubiquitous Computing, in which many computers are seamlessly integrated into the physical environment [57]. The objective was to support and enhance a user's experience by making life and tasks easier. A significant body of research explores ways in which that goal can be accomplished. For example, there is research on virtual tour guide applications that use location, one of the most commonly used forms of context, to display information about places or objects in the user's vicinity [17]. There is research on applications geared towards an office environment that use location to route a user's incoming calls to the phone closest to the user, or to present information on the closest display [55]. Less research effort has focused on how context-awareness can be used to enhance security.

2.4 Context-aware Security

Context-aware security involves security mechanisms dynamically adapting to the user's situation based on context. Context has been incorporated into different security services in various ways and at different levels. For example it has been used to supplement user attributes that are traditionally used in authentication and access control, such as username and password [20]. It has also been used in a more primary way and replaced common user attributes, resulting in security decisions based solely on context [27]. This is useful when the identities or roles of users are not known in advance. A majority of the research on context-aware security focuses on access control. Integrating context into access control allows permission assignments to be based on more than just the identity or role of the user, but also on the current situation of the user.

2.5 Summary

In this chapter, we presented an overview of the concepts that are fundamental to our research. We described different types of access control, access control polices, and presented a basic access control model. We presented the definition of context we utilize, described general context-aware systems, and context-aware systems that focus on security. In the next chapter we present our notions of context-aware security, which we call Shrink-Wrapped security.

Chapter 3

Shrink-Wrapped Security

In this chapter we present the various components necessary to realize shrink-wrapped security. Recall that shrink-wrapped security is a security paradigm that involves providing a tight coupling between a user's current situation and security, and this requires a more comprehensive notion of context than what is currently used by context-aware security systems. Accordingly, we present a definition and corresponding guidelines to facilitate the systematic identification of relevant context. We also present an approach to practically and easily incorporate the use of such context into security services. Shrink-wrapped security is highly dynamic and context must be monitored to assure that as a mobile user's context changes, the security changes accordingly. Thus we also present a framework to facilitate the secure acquisition, monitoring and use of relevant context.

3.1 Defining Security-Relevant Context

Before any efforts to provide a comprehensive notion of security-relevant context can be successful, the term needs a specific definition. Providing a specific definition is essential because it will prevent ambiguity and confusion about what actually constitutes such context.

"Context" and "security" are terms that are already overloaded with many definitions. We aim to provide a definition of security-relevant context that can be agreed upon within the community. Therefore, instead of adding yet another definition to each term, we derived our definition of security-relevant context from a widely accepted and agreed upon definition of context [22], and an authoritative definition of information security [5]. The following is our definition of security-relevant context:

Security-relevant context consists of the set of contextual attributes that can be used to characterize the situation of an entity, whose value affects the choice of the most appropriate controls or the configuration of those controls to protect information and information systems from unauthorized access, use, disclosure, disruption, modification or destruction in order to provide confidentiality, integrity and availability.

Confidentiality refers to preserving authorized restrictions on access to, and disclosure of, information. Integrity refers to guarding against improper information modification or destruction. Availability refers to guarding against the unauthorized disruption of access to information, ensuring the timely and reliable access to and use of information [5].

The key insight we provide to the definition of security-relevant context lies in the phrase "whose value affects the choice of the most appropriate controls or the configuration of those controls." The values of security-relevant contextual attributes affect the choice of the most appropriate controls, because they impact the likelihood of certain threats to confidentiality, integrity, and availability being realized. Therefore, based on their values, the most appropriate controls to mitigate those threats can be employed. For example, the value of the contextual attribute, *connection security*, impacts what controls should be used to ensure confidentiality. The value of that attribute can determine the most appropriate encryption algorithm (control) to use for application level encryption (for example, DES or AES) and the configuration of the selected control (for example, key length).

3.1.1 Definition Discussion

There may be context that will not be considered security-relevant according to our definition that designers of context-aware security systems deem useful. Consider the example of a context-aware access control system. Temporal context, such as the time of day or day of the week, does not directly or inherently affect a system's attempt to protect resources from unauthorized access. However, temporal context is frequently used in context-aware access control systems. We believe that such context is usually utilized to specify and enforce corporate policy compliance, not because of its innate effect on security. We do not deny the utility of such context, but it is our objective to facilitate the identification of context that is most pertinent to a security system, i.e., context that impacts a system's attempt to achieve its security objectives.

3.2 Identifying Relevant Context

Identifying relevant context is one of the key challenges of context-aware system development [49, 39]. This difficulty has been attributed to the fact that there are no elicitation methods for context [50]. Without an elicitation method, developers must use ad-hoc approaches to identify context. We have developed a taxonomy to be used in conjunction with guidelines, which will enable designers of context-aware security services to use a systematic approach to identify relevant context [31]. These tools can be utilized in the design of context-aware security systems to help prevent forms of security-relevant context from being overlooked. In addition they can be used in the evaluation of existing systems to determine if additional context needs to be considered, or if existing context may not be pertinent for that system.

3.2.1 Taxonomy of Security-Relevant Context

There are various features of security-relevant context that can be used as a basis for classification, including: context origin, the potentially affected security objective, and the entity the context is inherently related to. Determining which features are most appropriate to use depends on the purpose of the taxonomy. The purpose of our taxonomy, which is specific to our focus domain of access control for the mobile workforce, is to facilitate a comprehensive notion of security-relevant context. Accordingly, our taxonomy classifies context along the two dimensions [34] we feel best accomplish that goal: the *relevant entity* and the *affected security objective.*

3.2.1.1 Relevant Entity Dimension

The relevant entity dimension stems from Dey's definition of context. The categories of this dimension consist of the entities that are relevant to the interaction between a mobile user and a context-aware security service, including: the user, the user's computing device, the communication mechanism between the user and the security service, and the surrounding environment. Context should be classified by the entity to which it is inherently related and which it characterizes. This is intuitive for the *user, user's computing device*, and *communication mechanism.* Context that characterizes the situation surrounding the user, but not the user herself, including the physical environment, falls into the *surrounding environment* category.

Table 3.1 shows relevant entities and examples of corresponding context.

3.2.1.2 Affected Security Objective Dimension

The purpose of our taxonomy is not to facilitate a comprehensive notion of context in general, but of *security-relevant* context. Accordingly, classifying context by an additional dimension, the *affected security objective*, will ensure that context is indeed relevant. The categories for this dimension consist of the fundamental security objectives: confidentiality, integrity and availability. Context should be classified by the security objective(s) it affects.

To prevent categorization of security-relevant context at too high of a level, we identified sub-categories. This was done by considering domain-specific threats

Table 3.1: Relevant entities and example context

Entity	**Example Context**
The user	User location, authentication mechanism used, etc.
The user's computing device	Orientation of device, antivirus software status, etc.
The communication mechanism	Connection security, networks traversed, network topology, etc.
The surrounding environment	Co-location of people, co-location of devices, lighting, noise level, temperature, etc.

to the security objectives.

Threat Model

Resources may be accessed by an unauthorized entity, resulting in a breach of confidentiality. The following lists some of the ways in which this may happen:

- An unauthorized entity may impersonate an authorized entity and obtain access to a resource
- An unauthorized entity may obtain physical or remote access to an authorized user's computing device

- An unauthorized entity may "sniff" a resource while in transit to an authorized user's computing device
- An unauthorized entity may gain indirect access to a resource through proximity

Resources may be modified by an unauthorized entity, resulting in a breach of integrity. The following lists some of the ways in which this may happen:

- An unauthorized entity may impersonate an authorized entity and make unauthorized alterations
- An unauthorized entity may alter a resource while in transit to an authorized entity
- An unauthorized entity may alter a resource once it arrives at an authorized user's computing device

An unauthorized entity may intentionally deny resource access to authorized entities, resulting in a breach of availability [1]. The following lists some of the ways in which this may happen:

- An unauthorized entity may forge or alter the context of an authorized entity to deny them access to a resource

[1] We do not attempt to address denial-of-service attacks that result from a malicious entity jamming the communication mechanism so that resources can't be delivered, or context can't be sent to the shrink-wrapped security service.

3.2.1.3 Some Guidelines for Classifying Context by the Affected Security Objective

Confidentiality Category

Contextual attributes whose value affects the likelihood of threats of unauthorized access to a resource being realized should be classified in this category. This category is divided into four subcategories:

- *access by entity impersonating an authorized entity*
- *access via unauthorized access to an authorized users computing device*
- *unauthorized access via resource sniffing while in transit*
- *indirect, unauthorized access through proximity to an authorized user*

An example of context related to the user's surrounding environment that should be classified in the subcategory *indirect access through proximity* is the co-location of people. If the user is co-located with people lacking the proper authorization to access a resource, then it is possible for those people to obtain unauthorized access. For example a co-located person can look at the screen of the requesting user's computing device. This would result in a breach of confidentiality.

Integrity Category

Contextual attributes whose value affects the likelihood of threats of unauthorized modification to a resource being realized should be classified in this category. This category is divided into three subcategories:

- *modification by an unauthorized entity impersonating an authorized entity*

- *unauthorized resource modification while in transit*

- *modification via unauthorized access to an authorized users computing device*

An example of context related to the communication mechanism that could be classified in the subcategory *resource altered while in transit* is the security of the connection. If the connection is not secured, then it is easier for a malicious party to alter a resource on the path to the user, resulting in a breach of integrity.

Availability Category

Contextual attributes whose value affects the likelihood of threats of intentional denial of access to a resource being realized should be classified in this category. This category has one subcategory:

- An unauthorized entity may forge or alter the context of an authorized entity to deny them access to a resource

An example of context related to the user's computing device that could be classified in this category is the status of the antivirus software. If the antivirus software is missing or severely outdated, then it is probable that the device is infected with some type of malicious software. This malicious software could cause a number of problems, including forging and altering context to prevent resource access.

3.2.1.4 Taxonomy Discussion

Taxonomy Completeness

The categories that were identified for the relevant entity and affected security objective dimensions do not cover all possibilities. Our goal here is not to deliniate an exhaustive list of possibilities, but to create an extensible framework. Our taxonomy is extensible, and additional categories can be added to both dimensions, if necessary. For example, although confidentiality, integrity, and availability are the fundamental goals of security, an additional objective such as non-repudiation may be necessary, depending on the particular context-aware system.

Affected Security Objective Dimension

Some security-relevant contextual attributes may be classified in multiple categories in the *Affected Security Objective* dimension. This is acceptable because the primary dimension, *Relevant Entity*, is mutually exclusive, and the main objective of this dimension is to ensure relevance.

3.2.2 Some Guidelines to Facilitate the Identification of Relevant Context

The following guidelines have been prepared to be used in conjunction with our taxonomy to facilitate the systematic identification of security-relevant context. The application of the guidelines is illustrated by using them to identify context for the development of a shrink-wrapped access control service. Figure 3.1 illustrates our guidelines, which use a goal-oriented approach to identify relevant context. Such an approach is similar to goal-oriented requirements engineering in which goals are used to elicit system requirements [54]. A goal-oriented approach helps avoid irrelevant context as goals provide a precise criterion for context pertinence [60].

- **Step 1. Identify the security objective(s) of the target security ser-**

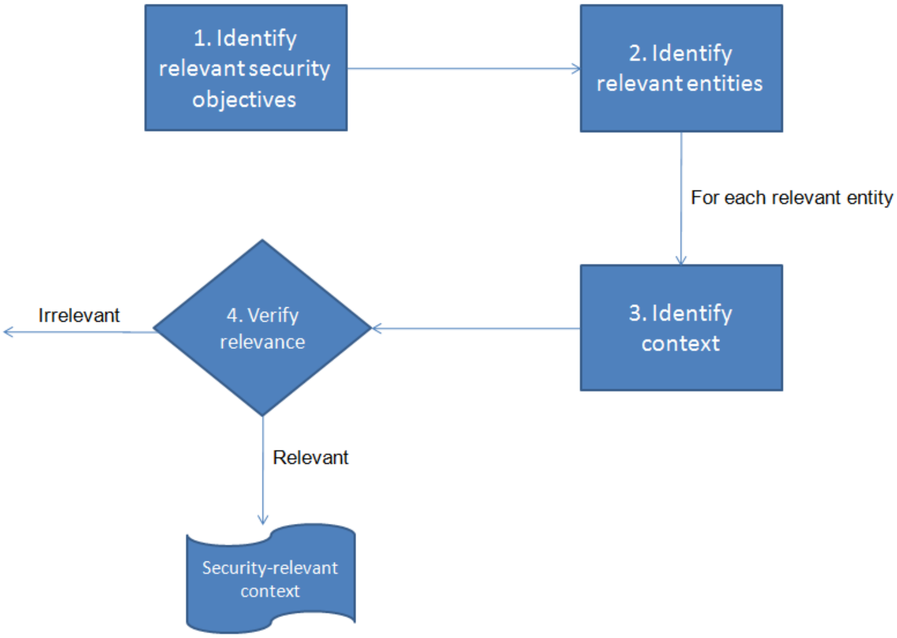

1. Identify relevant security objectives
2. Identify relevant entities
For each relevant entity
3. Identify context
4. Verify relevance
Irrelevant
Relevant
Security-relevant context

Figure 3.1: Flow chart for identifying relevant context

vice.

Identify the security objectives of the target security service. Recall that a security service is a capability that supports one, or more, security requirements, such as confidentiality, integrity, and availability. Examples of security services are encryption, access control, and authentication. The security objective of a security service can be identified by considering the high-level goals of the security service and deciding which security objectives from the *affected security objective* dimension align with those goals. For example, the goal of access control is to ensure that subjects are only allowed to access the resources for which they have authorization and access those resources in authorized ways. Therefore, the security objectives with which access control aligns are confidentiality and integrity.

- **Step 2. Identify the relevant entities for the target security service.**

  An entity is a person, place, or object. An entity is relevant if its state affects the achievement of the security objectives identified in step 1 of the guidelines. The entities that are directly a part of the interaction between a user and a security service are a good place to start identifying relevant entities. However, entities that are not directly part of the interaction may also be relevant. For example, the entity, *surrounding environment*, is not directly part of the interaction between a user and an access control service, but as illustrated in section 3.2.1.3, its state does impact one of the identified security objectives (confidentiality). The entities relevant to the interaction between a user and an access control service include all of the categories of the relevant entity dimension: the user, the user's computing device, the communication mechanism, and the surrounding environment. However, this may not be the case for every security service.

- **Step 3. Identify context that may impact security, related to each entity identified in step 2.**

  This is a brainstorming step. For each relevant entity identified in the previous step, use brainstorming techniques to identify contextual attributes that may be security-relevant. Consider what aspects of each entity's situation impact the likelihood of threats to the security objectives identified in step 1 being realized. The threats to the security objectives correspond to the subcategories of the *affected security objective* dimension. For example, consider what aspects of the situation of a user can impact the likelihood of someone impersonating them, or how the state of a computing device can impact the likelihood of an unauthorized entity gaining access to it. The objective is to identify as many attributes as possible. Do not evaluate during this step. Simply write down all attributes that come to your mind.

This step facilitates a holistic view of a situation by illustrating the importance of considering context related to *every* entity that is relevant to the interaction between a user and a security service. It stresses the fact that characterization of a situation requires an examination of everything related to the situation, not only common factors. Refer to Table 3.1 as an abridged illustration of this step.

- **Step 4. Verify that the identified context is relevant by attempting to classify it by the affected security objective(s) identified in Step 1.**

  This step is important because although we advocate using a comprehensive set of context, we only recommend utilizing context that is pertinent to a particular security service. If context can not be classified by the security objective(s) identified in step 1 then it is not relevant for that security service. To illustrate this step, we attempted to classify the context related to the user's computing device as found in Table 3.1.

Antivirus Status The status of a computing device's antivirus software affects the likelihood of the device being infected with malicious software. For example, if no antivirus software is being utilized or it has not been updated in a long time, then the computing device has an increased likelihood of being infected with some type of malicious software (malware). The antivirus status does not, however, indicate what type of malicious software the device may contain. For example, the device could contain malware that either allows remote connection by the malware author, steals, alters, or destroys information. Therefore, this context could fall into the following categories: confidentiality - unauthorized access to an authorized user's computing device; integrity-altering resource once at user's computing device and availability. Accordingly, the antivirus software status is clearly relevant.

Orientation of device The orientation of a user's computing device has been utilized by context-aware systems to adjust the display of information on a user's screen. However, this information does not impact a systems ability to ensure the confidentiality or integrity of resources, and is thus not security-relevant.

3.2.2.1 Using the Guidelines to Evaluate Current Systems

In addition to being utilized during the design phase, by slightly altering the guidelines in section 3.2.2, our taxonomy can be used to evaluate existing security systems. Identifying the relevant entities and security objectives for the system to be evaluated will allow the context currently being used by that system to be assessed. By classifying the currently used context by our taxonomy, it can be determined if additional context needs to be considered, and if existing context may not be pertinent for that system. The following guidelines have been prepared to be used in conjunction with the taxonomy to assist in the evaluation of existing systems.

1. **Refer to step 1 from the guidelines in section 3.2.2.**

2. **Refer to step 2 from the guidelines in section 3.2.2.**

3. **Classify the currently used context by the relevant entities identified in step 1.** At a minimum, context related to every relevant entity should be represented. If this is not the case, then additional context needs to be considered. See step 3 from the guidelines in section 3.2.2.

4. **Refer to step 4 from the guidelines in section 3.2.2.**

3.2.3 Repository of Security-Relevant Context

We have created a repository of security-relevant context classified by our taxonomy (see Appendix A). With the participation of System Administrators and an Information Assurance Engineer, we have identified a set of discrete values for each identified contextual attribute. For example, for the contextual attribute, *Antivirus Status*, we identified the following values: *Not present, Present but not up-to-date, Present and up-to-date, Present and up-to-date with on-access scanning enabled.* By definition, for something to be relevant it has to have a demonstrable bearing on the matter at hand [56]. Accordingly, we also include an explanation for each relevant attribute's inclusion in the list. We envision this information serving as a resource that will allow designers that have identified the relevant entities and security objective(s) for their system to quickly and easily identify relevant context.

3.3 Incorporating Context into a Security Service

In the previous section we presented a technique to identify security-relevant context. In this section we present an approach aligned with the shrink-wrapped security paradigm, to practically incorporate the use of such context into the security policies that govern a security service, tailored to our focus domain. In line with the concept of shrink-wrapped security, we advocate the incorporation of as much relevant context as possible into security policies, to assist more informed decision making. To facilitate this, we present an approach to abstract relevant contextual attributes to a security level associated with the security objectives of a system. Such an approach is useful because it removes the need to explicitly incorporate each individual attribute into security policies while still obtaining the value of each attribute, which eases policy development and management.

Before presenting the details of our approach, we present additional details of the motivating scenario presented in chapter 1, which will help illustrate our approach. Alice's company has a set of resources that have different protection requirements based on their importance. Let us assume that each resource belongs to one of the following three classes: *unclassified, secret* and *top secret*,which are ordered by "<". Different types of resources have different access operations, which define allowable ways that they can be accessed. For example, the access operations for a text file may be *read only, append*, or *read and write.* In order for a user to perform a certain access operation on a resource belonging to a certain resource class, they must have the required permission. A permission is a tuple <*access operation*, *resource class*> that specifies an authorized interaction.

3.3.0.1 Likelihood

By our definition, the values of security-relevant contextual attributes affect the likelihood of threats to a system's security objectives being realized. As such, our approach involves associating a number in the interval $[0, 1]$ with each value of an attribute, representing that likelihood. For example, the values in table 3.2 represent that if the attribute, *currency of device patches*, currently has a value of patches not up-to-date, then the likelihood of a threat to the security system being realized is .9. However, if the current value of the attribute is patches up-to-date, then the likelihood of a threat to the security system being realized is only .1. The likelihood values can be determined in various ways.

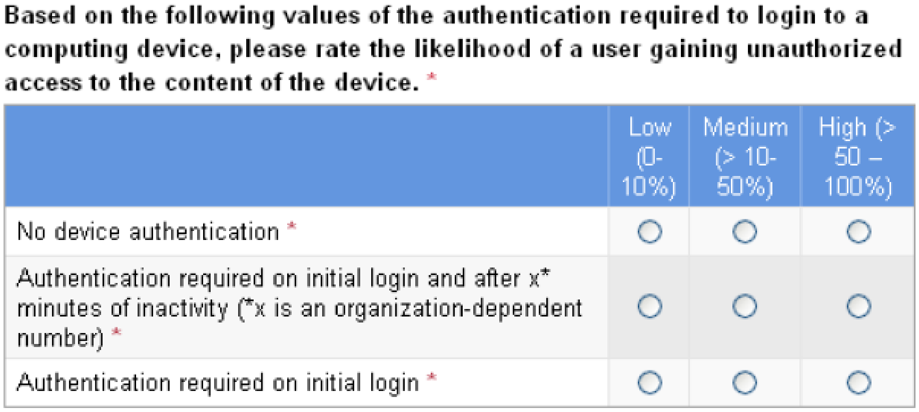
There are several approaches to computing the likelihood that an event will occur, including: classical, frequency, and subjective. Classical probability is based on having precise models. For example, to calculate the probability that a certain side of a six-sided die results from tossing the die, a model of the die in which each side is equally sized and weighted is considered. Frequency probability involves em-

Contextual Attribute: *currency of device patches*	
Value	**Likelihood**
patches not up-to-date	.9
patches up-to-date	.1

Table 3.2: Example of likelihood values associated with each value of an attribute

pirically determining the probability. For example, a die would be tossed several times by several people and ideally their aggregate distribution will approach the correct distribution. Subjective probability involves asking experts for their opinion on the likelihood of an event. The selection of an approach should be based on the situation and available information. In security it is often not possible to directly evaluate an event's likelihood using classical techniques [42]. Frequency probability depends on a system already being built and being in use for some time. It also assumes environmental stability and replication. We were not focused on any specific system, but systems in general, so this technique was not appropriate either. Accordingly, we decided to use a subjective probability approach.

We communicated with a group of eleven security experts to determine how the values of contextual attributes affect the likelihood of various security threats being realized. The group consisted of individuals with at least three years of personal experience, and 105 years of cumulative experience doing security-related work from various sectors, including industry, government, academia, and military. We contend these individuals were able to make informed estimates based on their experience. Figure 3.2 illustrates an example question that was posed to the group. The information obtained from this exchange allowed us to associate a threat likelihood with each value of an attribute. We used the value selected by a majority of participants. We also collected information on the perceived level of relevance of each attribute, as some attributes may be more pertinent to decision making than others. The relevance values are also numbers in the interval $[0, 1]$.



Based on the following values of the authentication required to login to a computing device, please rate the likelihood of a user gaining unauthorized access to the content of the device. *

	Low (0-10%)	Medium (> 10-50%)	High (> 50 – 100%)
No device authentication *	○	○	○
Authentication required on initial login and after x* minutes of inactivity (*x is an organization-dependent number) *	○	○	○
Authentication required on initial login *	○	○	○

Figure 3.2: Example likelihood question

The exchange taught us that the values provided by the security professionals were based on implicit assumptions that are specific to their particular organizations. As such, it may be most useful if organizations derive the likelihood values associated with each value of an attribute internally using the approach, or combination of approaches, most appropriate to the available information. Nevertheless, we contend that this information can serve as a starting point, and we utilized the values in the development of our prototype system. In the next subsection we present our approach, based on logic programming, to practically incorporate such information into the security policies that govern access to resources [30].

3.3.1 Using Generalized Annotated Programs for Shrink-Wrapped Security

We wanted to develop a principled approach to abstract a user's context to a security level, so we decided on a logic-based framework. Specifically, we selected generalized annotated programs (GAPs), which are logic programs that allow real-numbered values called *annotations* to be associated with atomic propositions. Below, we describe GAPs [32] and provide details of how we have tailored this technique to our focus domain.

Definition 1 (annotated atom/GAP-rule/GAP) *GAPs are defined as follows.*

- *Given* $A \in \mathsf{Prop}$ *and annotation* x *(which is a number in* $[0, 1]$*, a variable symbol, or a function over* $[0, 1]$*),* $A : x$ *is an* ***annotated atom****.*
- *Given annotated atoms* $A_0 : x_0, A_1 : x_1, \ldots, A_n : x_n$*, the following:* $A_0 : x_0 \leftarrow A_1 : x_1 \wedge \ldots \wedge\ A_n : x_n$

  is a ***GAP rule****. The annotated atom* $A_0 : x_0$ *is the* ***head*** *and the conjunction* $A_1 : x_1 \wedge \ldots \wedge\ A_n : x_n$ *is the* ***body****.*
- *A generalized annotated program* ***(GAP)***Π *is a finite set of GAP rules.*

We represent every contextual attribute, relevant entity, and security objective as an atomic proposition in the set Prop. We also have additional atomic propositions that represent other aggregate security levels.We divide Prop into four subsets - $\mathcal{A}, \mathcal{E}, \mathcal{O}, \mathcal{G}$ - containing atomic propositions associated with contextual attributes, relevant entities, security objectives, and aggregates respectively. Specific to our application, we introduce two functions. The first is the function $\textit{threat} : \mathcal{A} \rightarrow [0, 1]$, which represents the threat likelihood (e.g., .1 for low threat, 0.5 for medium, and 1 for high) associated with the current value of an attribute. The second function, $\textit{relev} : \mathcal{A} \rightarrow [0, 1]$, represents the relevancy of each attribute (e.g., .1 for low relevancy, 0.5 for medium relevancy, and 1 for high-relevancy). These values are entered a-priori.

Specific to our application, we have some restrictions on the composition of GAP rules. For rules with an atom from set $\mathcal{A}$ in the head, the annotation must be a function of the associated *threat* and *relev* functions for that atom. Further, for such rules, the body must be a tautology (see Rules 3.1-3.6 of Figure 3.3). Intuitively, we want the threat level assigned to a contextual attribute to be a function of the threat likelihood and relevancy. If the head of a rule contains a relevant entity atom, then the body can only have contextual attribute atoms associated with that entity. Likewise, rules with security objective atoms in the head can only have contextual attribute atoms associated with that objective in the body [2] . Following, is an example of how GAPs can be used by an administrator to abstract a user's situation to security levels.

[2] Recall that contextual attributes are classified with our taxonomy by the *relevant entity*, and *security objective*

Suppose Alice's company has context providers that can determine the strength of the user's password pwd, the authentication technique used by the user auth_tech, who the user is co-located with co-location, the antivirus status of the user's computing device antivirus, the currency of the user's system patches patches, and the connection encryption con-encrypt. Figure 3.3, illustrates a portion of a GAP, Π, for this scenario.

$$\mathsf{pwd} : \mathit{relev}(\mathsf{pwd}) \cdot \mathit{threat}(\mathsf{pwd}) \leftarrow \tag{3.1}$$

$$\mathsf{auth\_tech} : \mathit{relev}(\mathsf{auth\_tech}) \cdot \mathit{threat}(\mathsf{auth\_tech}) \leftarrow \tag{3.2}$$

$$\mathsf{antivirus} : \mathit{relev}(\mathsf{antivirus}) \cdot \mathit{threat}(\mathsf{antivirus}) \leftarrow \tag{3.3}$$

$$\mathsf{co\text{-}location} : \mathit{relev}(\mathsf{co\text{-}location}) \cdot \mathit{threat}(\mathsf{co\text{-}location}) \leftarrow \tag{3.4}$$

$$\mathsf{con\text{-}encrypt} : \mathit{relev}(\mathsf{con\text{-}encrypt}) \cdot \mathit{threat}(\mathsf{con\text{-}encrypt}) \leftarrow \tag{3.5}$$

$$\mathsf{patches} : \mathit{relev}(\mathsf{patches}) \cdot \mathit{threat}(\mathsf{patches}) \leftarrow \tag{3.6}$$

$$\mathsf{user} : \sqrt{x_1 \cdot x_2} \leftarrow \mathsf{pwd} : x_1 \wedge \mathsf{auth\_tech} : x_2 \tag{3.7}$$

$$\mathsf{computing\ device} : x \leftarrow \mathsf{antivirus} : x \tag{3.8}$$

$$\mathsf{confidentiality} : \operatorname{avg}(x_1, x_2, x_3, x_4) \leftarrow \mathsf{pwd} : x_1 \wedge \mathsf{antivirus} : x_2 \wedge \mathsf{auth\_tech} : x_3 \wedge \mathsf{patches} : x_4 \tag{3.9}$$

$$\mathsf{integrity} : \min_i x_i \leftarrow \mathsf{pwd} : x_1 \wedge \mathsf{antivirus} : x_2 \wedge \mathsf{auth\_tech} : x_3 \wedge \mathsf{patches} : x_4 \tag{3.10}$$

$$\mathsf{availability} : x \leftarrow \mathsf{antivirus} : x \tag{3.11}$$

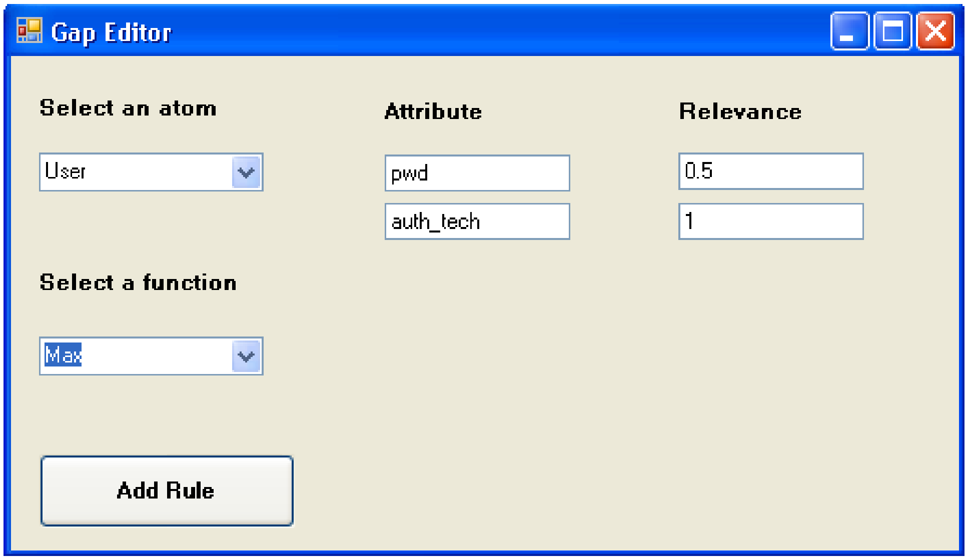
$$\begin{aligned}\mathsf{overall} : \max(x_1, x_2, x_3, x_4) \leftarrow\ & \mathsf{user} : x_1 \wedge \mathsf{computing\ device} : x_2 \wedge \\ & \mathsf{communication\ mechanism} : x_3 \wedge \mathsf{surrounding\ environment} : x_4\end{aligned} \tag{3.12}$$

Figure 3.3: Example GAP (Π)

Rules 3.1-3.6 are standard. Each contextual attribute is assigned an initial annotation based on the relevancy of the attribute and the threat likelihood. The value returned by the *threat* function is simply weighted by the value returned by the *relev* function.

Note, that while the administrator adheres to our previously mentioned restrictions, she has much flexibility in creating the remaining rules. For example, in rule 3.12 she aggregates the annotations of the relevant entities by selecting the entity that poses the greatest threat. However, in 3.9, she aggregates the annotations of the security objective confidentiality in a different manner - using the average.

We have implemented a graphical user interface (GUI) for security administrators to allow them to easily define a GAP (see figure 3.4). An administrator selects an atom, and the corresponding attributes are automatically displayed, along with their default relevancy values (which can be edited). The administrator is then able to define a function to aggregate the attributes, and add the current rule to the GAP.



Select an atom: User

Select a function: Max

Attribute	Relevance
pwd	0.5
auth_tech	1

Figure 3.4: GUI for our GAP editor

Definition 2 ($\mathbf{T}_\Pi$ and Multiple Applications of $\mathbf{T}_\Pi$) *Given GAP* Π *and Interpretation* I*:*

- *An interpretation I is simply an assignment of values to all atomic propositions - formally a mapping from the set* Prop *to* $[0, 1]$.

- *The operator* $\boldsymbol{T}_{\Pi}(I)$ *produces an interpretation that assigns atom A an annotation that is the supremum of all annotations assigned to A in the head of a rule in Π which satisfies I. Formally:* $\boldsymbol{T}_{\Pi}(I)(A_0) = \mathbf{sup}\{x_0 \mid A_0 : x_0 \leftarrow A_1 : x_1 \wedge \ldots \wedge A_n : x_n$ *is a rule in* Π *and for all* $1 \leq i \leq n$, $I \models A_i : x_i\}$.

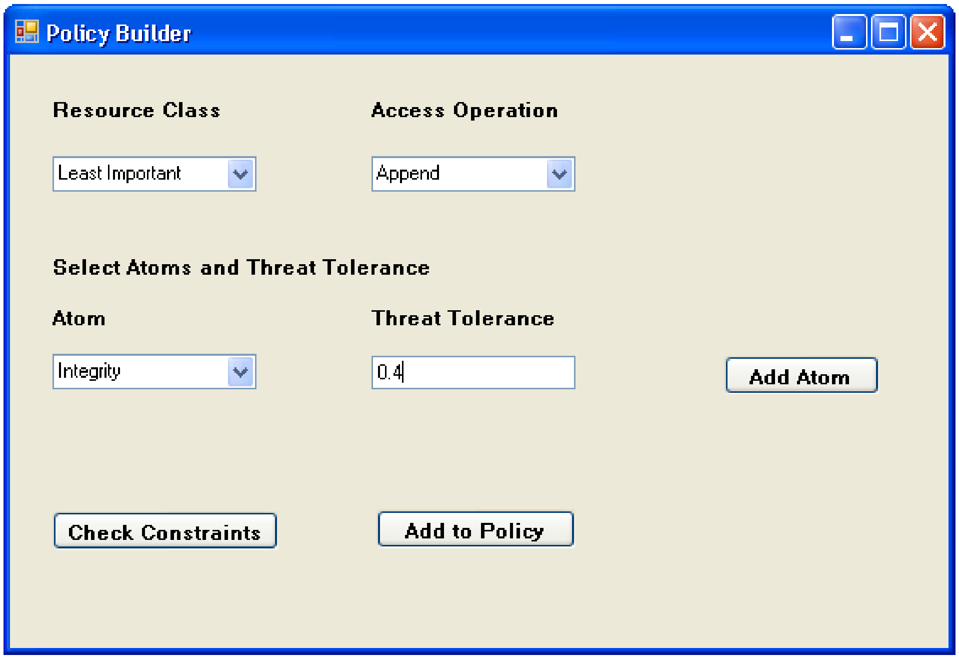
- *For application i of* $\boldsymbol{T}_{\Pi}$*, we write* $\boldsymbol{T}_{\Pi}^{(i)}$ *and define it as follows: for* $i = 0$, $\boldsymbol{T}_{\Pi}^{(i)} = \boldsymbol{T}_{\Pi}(I_0)$ *(where I_0 assigns zero to all atoms), otherwise,* $\boldsymbol{T}_{\Pi}^{(i)} = \boldsymbol{T}_{\Pi}(\boldsymbol{T}_{\Pi}^{(i-1)})$.

Let $\mathcal{I}$ be the set of all possible interpretations. The operator $\mathbf{T}_{\Pi} : \mathcal{I} \rightarrow \mathcal{I}$, when given an interpretation, produces a new interpretation based on the program Π. This operator can be applied multiple times until the interpretation converges. In[32] it is shown that $\mathbf{T}_{\Pi}$ is monotonic and that upon convergence, it has a least fixed point,*lfp*$(\mathbf{T}_{\Pi})$, that captures the maximum annotations of all the atomic propositions entailed by Π. Hence, the annotation assigned by *lfp*$(\mathbf{T}_{\Pi})$ corresponds to the highest threat level associated with each atom. This information can be incorporated into security policy to make decisions about access control.

Recall that shrink-wrapped access control aims to dynamically adjust a user's permissions so that at any given time she only has permissions that are appropriate based on her current situation. In order to explain what we mean by "appropriate", let us consider the fact that security in any system should be commensurate with its risks. Risk is a function of the likelihood of threats to the system being realized and the resulting impact [42]. We presented an approach to estimate the likelihood of threats to the system being realized. The class of a resource will determine the impact of such an exploit (which is organization dependent). Based on these two factors, the most appropriate (commensurate) security measure(s) can be employed.

For example, consider the GAP Π presented in figure 3.3. Suppose that $relev(\mathsf{patches}) = 1$ and for any atom $A \neq \mathsf{patches}$, $relev(A) = 0.5$. Additionally, based on the user's current situation, suppose the *threat* function returns the following values: $threat(\mathsf{pwd}) = 0.5$, $threat(\mathsf{auth\_tech}) = 0.1$, $threat(\mathsf{antivirus}) = 0.5$, $threat(\mathsf{patches}) = 1$, $threat(\mathsf{co\text{-}location}) = 1$, and $threat(\mathsf{con\text{-}encrypt}) = 0.1$. Presume after computing the least fixed point ,$lfp(\mathbf{T}_\Pi)$, we obtain an annotation of 0.5 for confidentiality, and an annotation of 0.27 for integrity. These annotations indicate the maximum threat likelihood associated with each atom, based on the combination of the user's context and the rules generated by the security administrator. Policy construction is as simple as setting a maximum tolerable threat level for various entities, security objectives, or other aggregates of interest, for each <*access operation*, *resource category*> tuple (permission). For example, if a user wants read access to a resource in the *unclassified* class, say a company newsletter, the company may be willing to accept a high likelihood that the resource will be disclosed because the information isn't sensitive or private. In other words, they are willing to accept a high threat likelihood for the security objective atom, *confidentiality*, and add the following rule to their policy "<read, unclassified>, confidentiality <= .9." Based on the annotation of .5 for confidentiality, this user would be granted access because it is less than the tolerance level of .9.

Figure 3.5 illustrates our interface to facilitate policy construction. A security administrator selects a resource class and access operation, then sets threat tolerance levels for various atoms. The ”Check Constraints” button allows an administrator to ensure the rule she is about to add is consistent with the existing policy rules. For example, a policy should not contain a rule that allows a user to perform an access operation on a resource of high importance if they are not allowed to perform that same operation on a resource of low importance.



Resource Class

Least Important

Access Operation

Append

Select Atoms and Threat Tolerance

Atom

Integrity

Threat Tolerance

0.4

Figure 3.5: Policy builder GUI

3.3.1.1 GAPs Discussion

There are several advantages to using logic programming, specifically GAPs, to incorporate security-relevant context into shrink-wrapped security policies. First, logic programming in general provides a principled approach. An annotated logic is well aligned with our approach, as each contextual attribute is associated with a threat likelihood value. GAPs extend annotated logic programming by allowing variables and evaluable functions to be used as annotations. This provides an administrator with more flexibility in deciding how various contextual attributes should be combined.

Another benefit of using GAPs is that we can give a user feedback about why a certain access was denied. This is due to the constructive nature of the least fixed point operator, $\mathit{lfp}(\mathbf{T}_{\Pi})$. In our current implementation we can simply do a backwards trace to identify each rule that causes the annotation of the atom in the rule head to exceed the tolerable threat value specified by an administrator. The union of security attributes in the bodies of all such rules is then the set of atoms that led to the security decision in question. Therefore, a user can be informed of the aspect(s) of her context that caused the denial.

Finally, it is important to note that the least fixed point computation takes polynomial time and obtains the same solution regardless of the arrangement of the rules in the GAP specified by the administrator. This allows us to quickly make consistent decisions about access in a principled manner that is a direct, logical consequence of an administrator's policies and user's context.

3.4 A Framework for Context Acquisition and Use

We developed a framework to facilitate the development and use of shrink-wrapped security services for the mobile workforce [29]. Our framework was designed with the following requirements in mind.

1. The framework should allow developers to focus on the purpose of a shrink-wrapped security service without having to deal with the intricacies of sensing and processing low-level context data.

2. The resource constraints of mobile devices, such as limited memory and computing power, must be addressed.

3. As shrink-wrapped security involves the transmission, processing, and possibly storage of users' contextual information, it is critical to ensure the privacy of this data and make sure that it isn't disclosed to any unauthorized party.

4. To prevent attempts to circumvent security by forging context, the forging of context must be prevented.

Figure 3.6 illustrates our layered architecture, which we designed to meet the previously stated requirements.

Context Providers

It is unlikely that a mobile computing device will have the capability to sense all of the security-relevant context of interest. Therefore, our framework allows context to be provided from a variety of distributed sources, including: sensors embedded in a

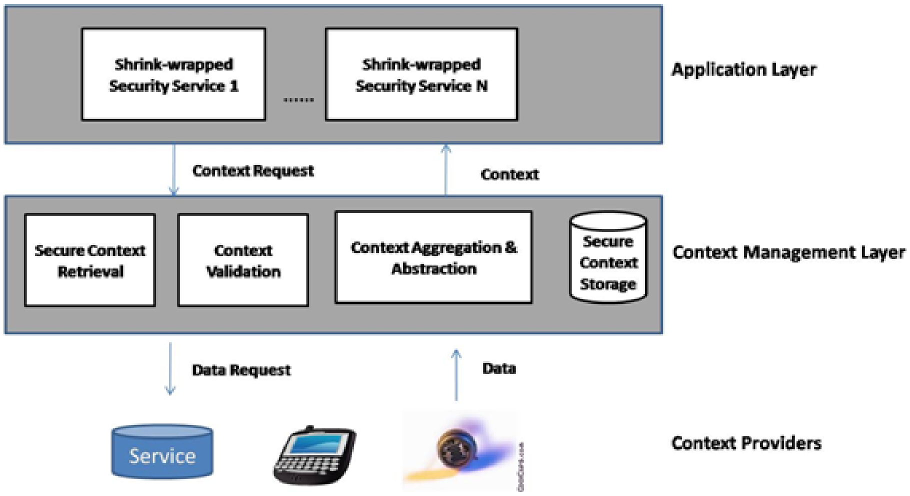


Figure 3.6: Shrink-wrapped Security Framework

computing device, external sensors in the environment, context providing services, and system state [3].

Context Management Layer

The Context Management layer handles retrieving raw data from context providers, processing and storing it, and providing context to the Application Layer. This layer contains a component that securely retrieves data from the various context providers. Validation that the context has not been altered and originated from a valid source also occurs at this layer. After the data has been validated, it can undergo additional processing, so that it can be delivered in the most useful format to shrink-wrapped security services. Various types of processing, such as abstraction/aggregation, are done at this layer. Due to the processor and memory constraints of mobile devices,

[3] We assume that the selected context providers are trusted to provide accurate context. There is an abundance of research on establishing the accuracy of context, but that is outside of the scope of this research.

this layer is hosted on a centralized server. Context is securely stored at the server in either a raw or processed form, depending on the intended use of the information.

An easy interface is provided for shrink-wrapped security services to retrieve relevant context.

Application Layer

The shrink-wrapped security layer is where security services, such as shrink-wrapped access control are implemented.

3.4.1 Framework Discussion

Our framework meets the requirements we identified. The layered architecture, in which context sensing and processing is separated from its use, addresses requirement 1. In addition, the separation is beneficial, because it increases extensibility and reusability [9]. The fact that the processing and storage of context data that occurs in the Context Management layer is done on a server as opposed to the mobile device, and that context can be provided from numerous sources, addresses requirement 2. The secure context retrieval and storage done by the Context Management layer addresses requirement 3. The context validation component addresses requirement 4.

3.5 Summary

In this chapter we presented the components necessary to realize shrink-wrapped security. By defining what constitutes relevant context for security services, and by developing a taxonomy and corresponding guidelines to facilitate its identification, we have addressed what has been identified as a key challenge in context-aware system development. We presented a technique to practically incorporate the use of context into security policies, allowing administrators to obtain the value of each attribute without explicitly including each attribute in a policy. Finally, we presented the design for a framework to facilitate the secure acquisition and use of such context. In the next chapter we present details about the implementation of our framework.

Chapter 4

The Implementation and Use of Shrink-Wrapped Access Control

In this chapter we present details about the implementation of the framework presented in section 3.4, and shrink-wrapped access control. We first present the shrink-wrapped access control model. We then introduce the baseline system that we extended, Rover, and present the updates to the system architecture that were necessary to implement our framework and shrink-wrapped access control. Finally, we discuss details of the the actual implementation, and present two usage cases that describe real-world examples of how users interact with the system.

4.1 Shrink-Wrapped Access Control

Recall that shrink-wrapped access control involves utilizing context to dynamically adjust a user's permissions so that at any given time she only has the permissions that are appropriate, based on her current situation. Figure 4.1 illustrates the basic model for shrink-wrapped access control. A subject requests access to a resource. This request will identify the particular resource and an access operation appropriate for that type of resource. All requests for access are mediated by an access control mechanism. The mechanism consults the current security policy and requests the values of the contextual attributes that are included in the constraints for access to the requested resource. The values of the contextual attributes are returned. Based on the subject's situation, determined by the values of the contextual attributes, access is granted or denied. Because the situation of a subject is not likely to stay constant, the contextual attributes of interest are monitored to make sure that access is still appropriate according to policy. If the constraints are no longer fulfilled, then access is revoked. The session with the access control mechanism ends once access has either been denied or revoked. In this way, permissions are a function of the subject's context and are dynamically adjusted based on context.

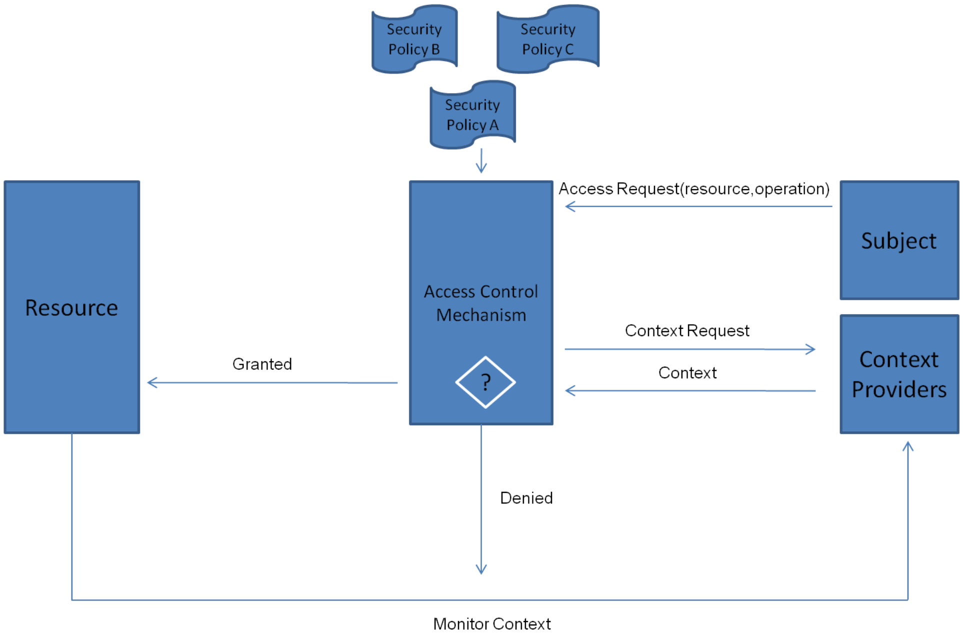


Figure 4.1: Shrink-wrapped Access Control Model

With shrink-wrapped access control, permissions are not the only thing that can change dynamically. In certain domains, it may even be appropriate for the actual security policy, or rules that determine access, to change dynamically. The policy would not necessarily change based on the situation of a mobile user, but based on external circumstances. For example, in response to an increase to the overall threat level on the Internet, a policy with more stringent constraints might be appropriate.

4.2 Baseline System

We extended an existing context-aware framework, Rover, to to implement the shrink-wrapped security framework and shrink-wrapped access control. Rover is an integration framework developed at the Maryland Information and Network Dynamics Laboratory (MIND Lab). It was designed to ease the movement of information between divergent entities. It provides the ability to handle context, manage entities and resources, and to integrate heterogeneous data sources. This framework can be molded to be used in a variety of domains, ranging from university and business campuses to public safety. For a detailed description of Rover see [8]. Figure 4.2 shows the basic components that comprise a Rover system. A Rover system represents a single domain of administrative control, managed and moderated by a Rover Server.

Rover Server - A Rover server has the responsibility for mitigating the flow of information between all entities in a Rover system

- Logging component – Logs all messages that pass through the Rover server in a database
- Resource Interfaces - Interfaces to local and remote resources
- Context Manager - Manages context for all Rover entities. The context man-

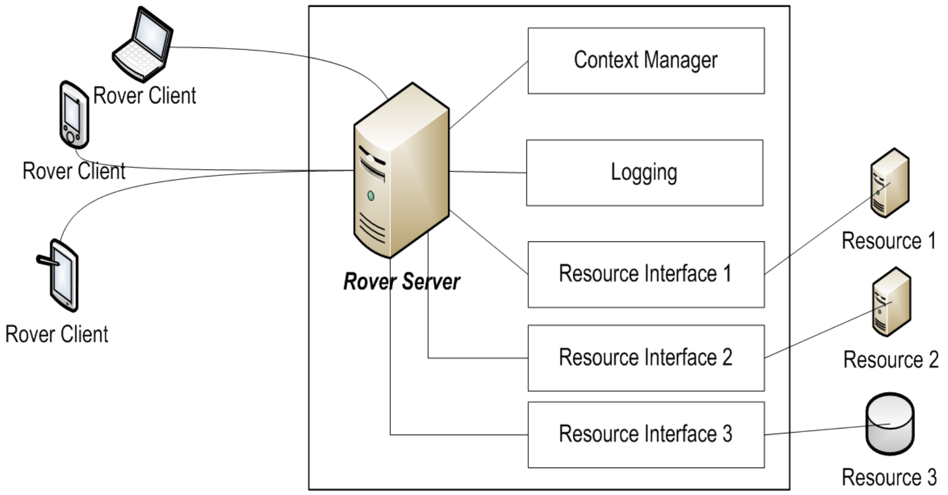


Figure 4.2: Baseline Rover System

ager contains context enhancers, which can abstract or refine contextual attributes. For example, if a subject's location is provided as a building number, a context expander can provide the corresponding latitude and longitude and vice-versa.

Rover Clients - Rover clients include people interacting in a Rover system through a desktop, laptop, or handheld device.

Resources - Resources, including various services and data sources, can be either local or remote to the Rover server

4.3 Updated System

We extended Rover to implement the shrink-wrapped security framework presented in section 3.4 and shrink-wrapped access control (see figure 4.3). Recall that our framework has the following layers: Application, Context Management, and

Context Provider. These layers are hosted on Rover Clients, the Rover Server, or third party context providers. The application layer is where shrink-wrapped security services are implemented, such as shrink-wrapped access control, and is hosted on Rover Client devices. The components of the Context Management layer are hosted on the Rover Server. Components of the Context Provider layer can be hosted on various components of a Rover system, including the server, clients and third party context providers.

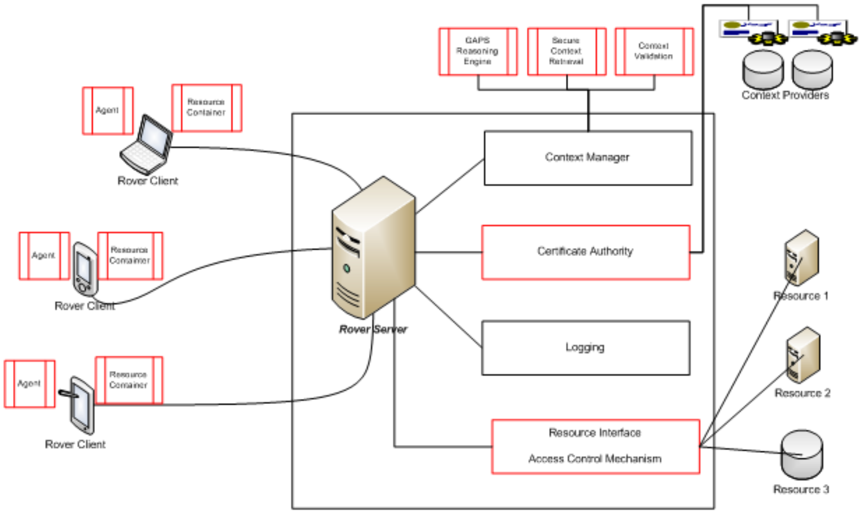


Figure 4.3: Updated Rover System

Some of the components of our framework were already, at least partially, implemented. For example, the context enhancers of the baseline system align with our Context Aggregation and Abstraction component. Additionally, the existing logging component of the baseline system was used to implement our Context Storage component.

The following include some of the ways we changed and extended the baseline system to implement the shrink-wrapped security framework and shrink-wrapped access control:

- Created a single, uniform interface for resource access
  - In the baseline system, each resource had its own separate interface. We created a single, uniform interface for all resource access, facilitating implementation of an access control mechanism that mediates all resource requests from Rover clients.
- Secure Context Retrieval
  - In the baseline system, communication between Rover entities occurs via unsecured TCP or HTTP connections, leaving both context and resources transferred between them vulnerable to sniffing and modification. Maintaining the integrity and confidentiality of context are fundamental design requirements of the shrink-wrapped security framework. As such, we secure the transfer of context between Rover entities.

    One of the assumptions of our framework is that there is a pre-existing trust relationship between the Rover Server and the selected context providers. This trust is affirmed and verified via digital certificates. The Rover server acts as a Certificate Authority and creates and signs the certificates of trusted context providers. We use TLS/SSL sockets to allow the Rover Server to verify that it is communicating with an authorized context provider, and secure context transmission from unauthorized tampering or disclosure.

- Automatic Context Acquisition
  - In the baseline system, Rover client users can provide and update their own context to the system. To prevent users from attempting to circumvent security by forging context, we added an agent to the Rover client that automatically retrieves context.
- GAPS Reasoning Engine
  - In addition to the existing context enhancers, we added a GAPs reasoning engine to the Context Management Layer hosted on the Rover Server. Such an addition allows the abstraction of a user's context to a security level and facilitates the incorporation of context into security policies.
- Resource Container
  - Achieving shrink-wrapped access control requires a controlled environment for resource access on a client device. Control of the resource must be maintained to allow the timely adjustment of access as determined by policy and a user's dynamic context. As such, we introduce the concept of a resource container, which allows us to control the functionality of default resource viewers and ensure that resources are not stored in cleartext on non-volatile storage of a client device. We have embedded

the Rover Client with a resource container that controls the usage of resources granted by the Rover Server.

Figure 4.4 illustrates the sequence of events that occur when a Rover client user requests access to a resource.

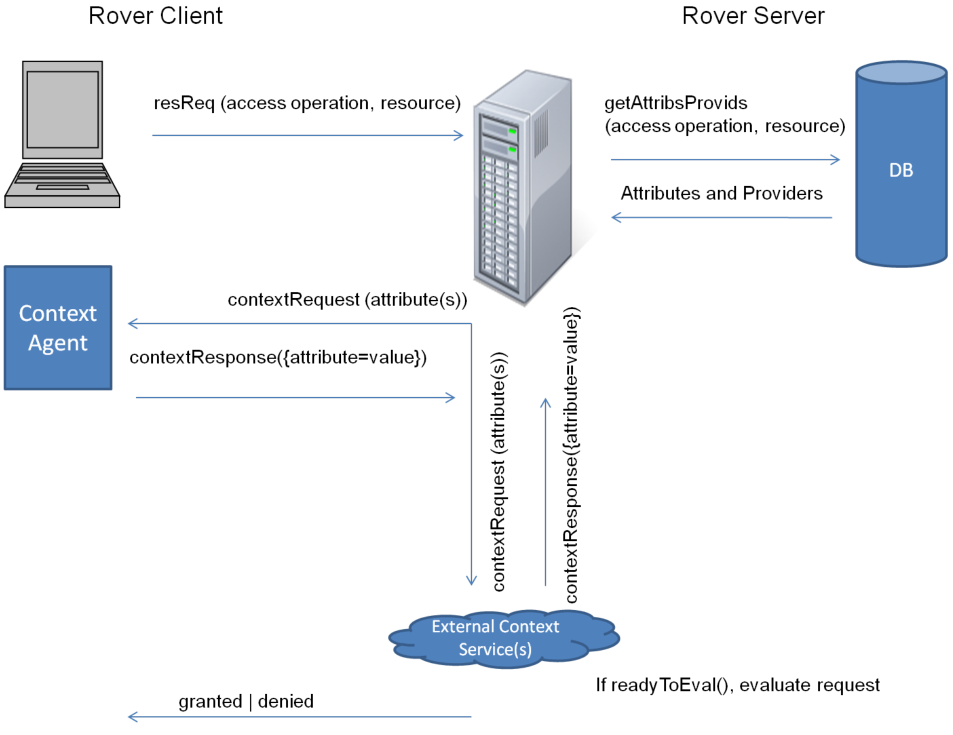

Rover Client
Rover Server
resReq (access operation, resource)
getAttribsProvids
(access operation, resource)
DB
Attributes and Providers
Context Agent
contextRequest (attribute(s))
contextResponse({attribute=value})
contextRequest (attribute(s))
contextResponse({attribute=value})
External Context Service(s)
If readyToEval(), evaluate request
granted | denied

Figure 4.4: Rover client user resource request

1. The resource access request goes to the Rover sever, where it is handled by the access control mechanism contained on the server

2. The access control mechanism obtains a list of contextual attributes used in the rule that corresponds to the particular access request, and the providers of those attributes. For example,{Client Agent:*Antivirus Status*, '128.8.126.42:28007':*Co-location of people*} indicates that the attribute *Antivirus Status* is available

from the client agent, and *Co-location of people* is available from an external context providing service at IP address 128.8.126.42, port 28007

3. The access control mechanism requests the values of the contextual attributes from the appropriate providers
4. When requested context is received, the access control mechanism determines if all requested values have been returned and the rule is ready to evaluate. If all requested values have been received, the rule is evaluated to determine if access shall be granted or denied
5. If access is granted, contextual attributes are monitored for changes to make sure constraints remain satisfied.
6. If access is denied or revoked, the access control session is terminated

4.4 Implementation Details

4.4.1 Client

Our prototype Rover client was implemented with the C# programming language. This allows the client to be run on any machine with the .NET Framework installed. The Rover client agent was implemented using Windows Management Instrumentation (WMI). WMI is the Microsoft implementation of Web-based Enterprise Management (WBEM), an industry initiative to develop a standard technology for accessing management information in an enterprise environment [1]. We used WMI to query and monitor components of the security-relevant context of a

Rover client device. Our resource container was implemented using the Edraw Office Viewer Component (EOVC), an ActiveX document container for hosting Office documents and PDFs [2]. This allowed us to tightly control resource usage by removing functionality from the default resource viewers. For example, we removed the ability copy, save as, and print, to ensure the Rover client maintains control of a resource. In addition, this component allows the display of password protected resources without the user having to know the password, or revealing the password to the user. This allowed us to ensure that resources are not stored in cleartext on non-volatile storage of a Rover client, which is a key factor of maintaining control of a resource. Finally, this component facilitated the immediate revocation of a resource, if contextual constraints were no longer met.

4.4.2 Server

We extended the codebase of the baseline Rover server. The server was implemented using Python, a cross-platform scripting language. This allows the server to run on a variety of platforms. We added a function for requesting access to resources to the server application programming interface (API). This function serves as the single gateway to all resources and is where access control is implemented.

OpenSSL, an open source toolkit that implements the secure sockets layer (SSL) and transport layer security (TLS) protocols, as well as a full-strength general purpose cryptography library [3], was used to implement the certificate authority on the Rover Server. This was used to generate and sign certificates for trusted context providers.

The baseline system used SQLite, a self-contained, serverless, zero-configuration, transactional SQL database engine [4], for authentication, context storage and logging. In addition, we added tables to store access control rules, information about contextual attributes and providers, and the likelihood and relevancy values associated with contextual attributes used by our GAPS reasoning engine.

4.5 Usage Cases

The following usage cases further illustrate the implementation of our system and the interaction between a Rover Server, Rover Client devices, and context providers.

4.5.1 Bring Your Own Device

Alice works for an Intelligence Agency. Her company recently decided to let employees bring their own mobile devices to work for personal and work related use, because it has been proven to increase productivity. Her company is aware of the security risks introduced by this and decided to utilize the Rover Framework to implement shrink-wrapped access control for a file server that employees use to access corporate resources.

Alice's Agency operates under two conditions 'normal', when there is no apparent hostile activity against the Agency or computer networks in general, and 'high alert', when there is an increased risk of attack against the Agency or computer networks in general. Admission to the Agency campus is controlled, and everyone must wear a badge with embedded sensors that allow identifying and tracking the location of that individual. There are employees and visitors on the campus. Employees can have different clearance levels and some employees don't have a clearance.

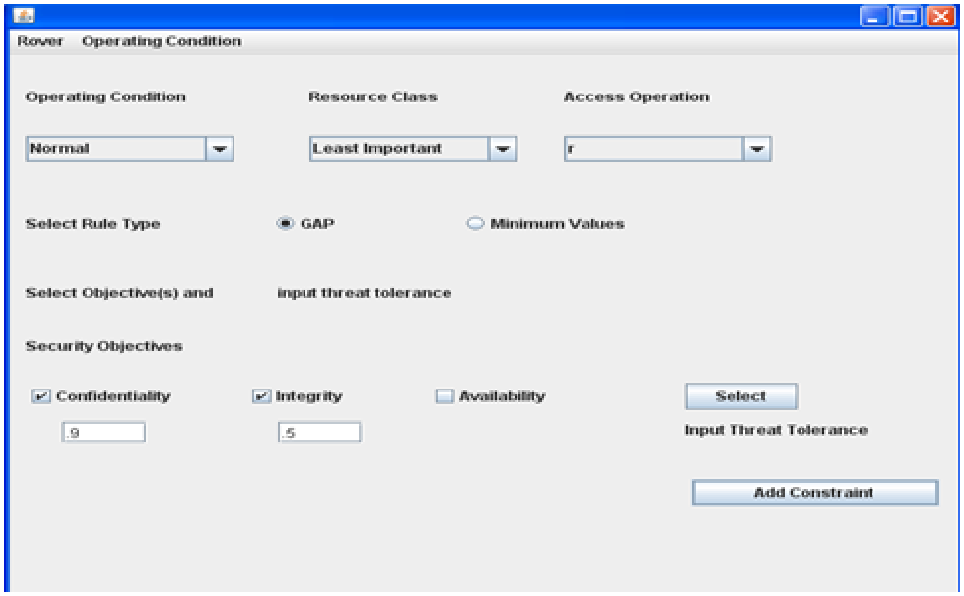
Alice's Agency currently has context providers that can determine the authentication technique used by the user, who the user is co-located with, the antivirus status of the user's computing device, the status of the user's firewall, and the connection encryption status. The security administrators performed the following prerequisite tasks necessary to implement shrink-wrapped access control:

- They associated a likelihood with each discrete value of every contextual attribute they utilize, and a relevancy value for each attribute. They decided to use the default attribute values, likelihoods, and relevancy values from our survey of security professionals.

- They used the administrator's interface to define a GAP that specifies how attributes will be aggregated by various entities, security objectives, and the overall situation.

- They used the administrator's interface to define the access control rules. For every operating condition, resource class, and access operation combination, they set a tolerable threat level for either security objectives, entities, or the overall situation (see figure 4.5).

- They used the certificate authority on the Rover Server to issue digital certificates to each context provider for authentication and secure communication.



Operating Condition: Normal

Resource Class: Least Important

Access Operation: r

Select Rule Type ● GAP ○ Minimum Values

Select Objective(s) and input threat tolerance

Security Objectives

☑ Confidentiality .9

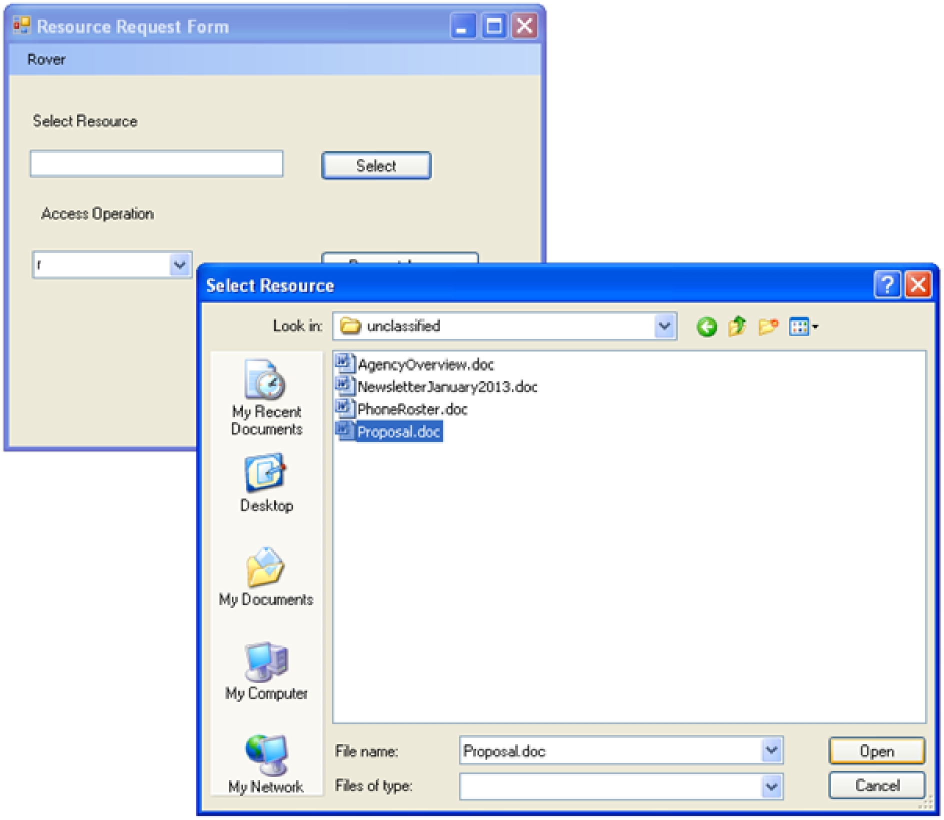
☑ Integrity .5

□ Availability

Figure 4.5: Administrator's Interface - defining access control rules

Alice has a Secret clearance. She is about to go to lunch and is waiting for a friend in a company lounge area. While waiting, she decides to use her mobile tablet to read an Agency proposal document, which is an unclassified, but proprietary document. She requests access to the file via the Rover Client application on her tablet (see figure 4.6).

Alice's request is sent to the Rover Server where the access control mechanism handles the request. Based on the access operation (read), the resource class (unclassified), and the current operating condition (normal), the server queries the database for the corresponding access control rule, the contextual attributes necessary to evaluate the rule, and the providers of those attributes. The rule associated with this particular request is "integrity <= .4, confidentiality <=.5." That means that the likelihood of a threat to integrity has to be less than or equal to .4 and



Look in: unclassified

AgencyOverview.doc
NewsletterJanuary2013.doc
PhoneRoster.doc
Proposal.doc

File name: Proposal.doc

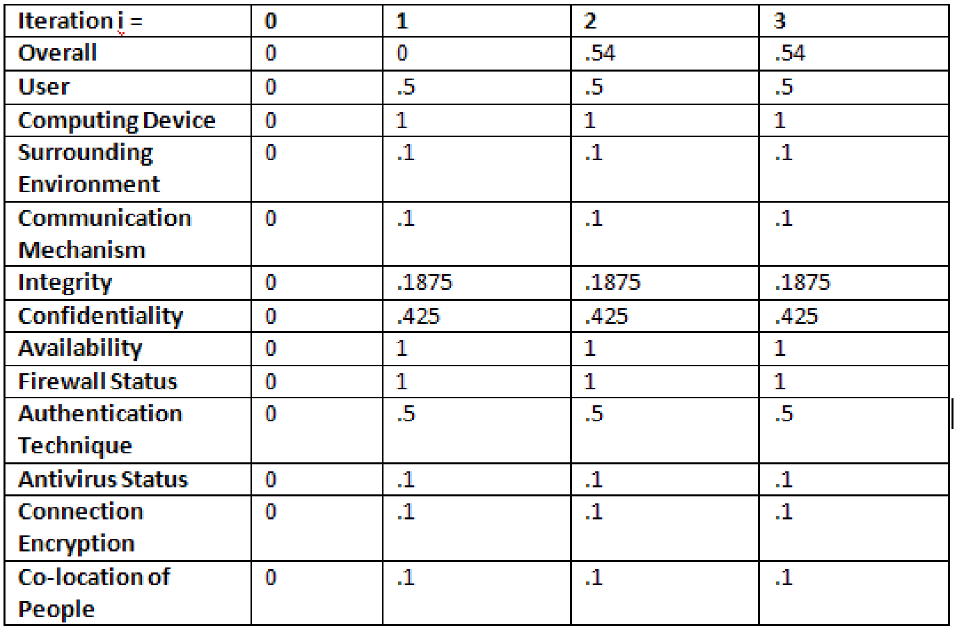
Access Operation: r

Figure 4.6: Requesting access to a resource

the likelihood of a threat to confidentiality has to be less than or equal to .5 for access to the resource to be granted. The contextual attributes necessary to estimate the threat to integrity and confidentiality based on the user's current situation are the authentication technique used by the user, the antivirus status of the user's computing device, the status of the user's firewall, the connection encryption status, and the co-location of people. The providers of the attributes are {Client Agent:Antivirus Status, firewall status, 128.8.126.42:65001:Co-location of people, RoverServer:Authentication Technique, Connection Encryption Status}.

The Rover Server uses multiple threads to securely connect to the context providers via TLS/SSL sockets and requests the needed values. The TLS/SSL protocol verifies that the context providers' posses a Rover Server signed certificate, and that the identity presented in the certificate matches the provider. If this is not the case, the connection fails. The protocol then cryptographically secures the connection to ensure the confidentiality and integrity of the transmitted context. As the providers return the values of the requested contextual attributes, the Rover Server checks to see if there is any outstanding context left to evaluate the rule. If the rule is ready to evaluate, the server retrieves the likelihood values that correspond with the current value of the contextual attributes from the database and that information is provided to the GAPS reasoning engine. In Alice's case, the agent included on her Rover client returned the following - 'Antivirus Status: Present and up-to-date with on-access scanning enabled, Firewall Status: Not present'. The Rover Server itself maintains some context on all connected clients and returned the following values - 'Authentication Technique:Password-based, Connection Encryption:Encrypted'. The external location server provided the following value 'Co-location of people: Not co-located with unauthorized users'. The corresponding likelihoods associated with these values are as follows: Antivirus Status: Present and up-to-date with on-access scanning enabled = .1, Firewall Status: Not present = 1, Authentication Technique:Password-based = .5, Connection Encryption:Encrypted = .1. The relevancy value associated with each of the mentioned attributes is 1.

The following table illustrates the annotation assigned to all atoms based on the GAP illustrated in figure 3.3. Each column represents iteration i of the fixpoint operator, and is computed based on values from the previous column, i-1. Note that as $\mathbf{T}_{\Pi}^{(2)} = \mathbf{T}_{\Pi}^{(3)}$, the operator has reached a fixed point, and the final annotations correspond to the maximum threat level for each atom.

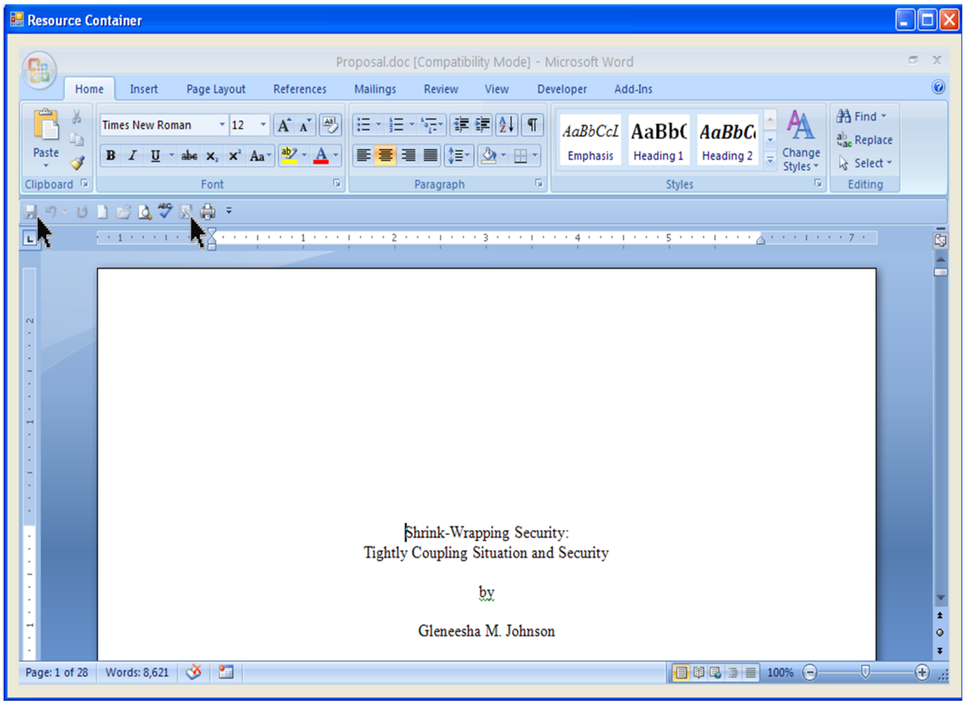


Iteration i =	0	1	2	3
Overall	0	0	.54	.54
User	0	.5	.5	.5
Computing Device	0	1	1	1
Surrounding Environment	0	.1	.1	.1
Communication Mechanism	0	.1	.1	.1
Integrity	0	.1875	.1875	.1875
Confidentiality	0	.425	.425	.425
Availability	0	1	1	1
Firewall Status	0	1	1	1
Authentication Technique	0	.5	.5	.5
Antivirus Status	0	.1	.1	.1
Connection Encryption	0	.1	.1	.1
Co-location of People	0	.1	.1	.1

Figure 4.7: Interpretations produced by multiple applications of $\mathbf{T}_{\Pi}$

The final value of integrity is .1875 and the final value of confidentiality is .425. The rule to access the resource, which is "integrity <= .4, confidentiality <=.5." is satisfied, and Alice's request is granted.

The resource is transmitted from the Rover Server to the Rover Client on Alices device where it is displayed by the resource container. As illustrated in Figure 4.8, the resource container has disabled functionality of the default DOC file viewer to maintain control of the resource and grant the appropriate access. For example, since Alice was only granted read access to the document, the ability to save any changes to the document has been disabled. Also to maintain control of the document, the ability to save the document under another name or location has been disabled.



Shrink-Wrapping Security:
Tightly Coupling Situation and Security

by

Gleneesha M. Johnson

Page: 1 of 28
Words: 8,621

Figure 4.8: Resource displayed in the Resource Container

While perusing the proposal document, a notification pops up on Alice's device that she has received an email. The email claims to contain a link to the latest version of the Angry Birds application. Alice clicks on the link to download the application and her on-access scanner blocks the download and presents a warning message indicating that the file is malicious. Alice assumes that the on-access scanning functionality of her device is being overly aggressive and decides to disable it so that she can try the latest Angry Birds application (see Figure 4.9). The Context Agent on Alice's device detects that a change to a security-relevant contextual attribute has occurred and sends the updated value, 'Antivirus Status: Present and up-to-date', to the Rover Server where the current access rights are re-evaluated.

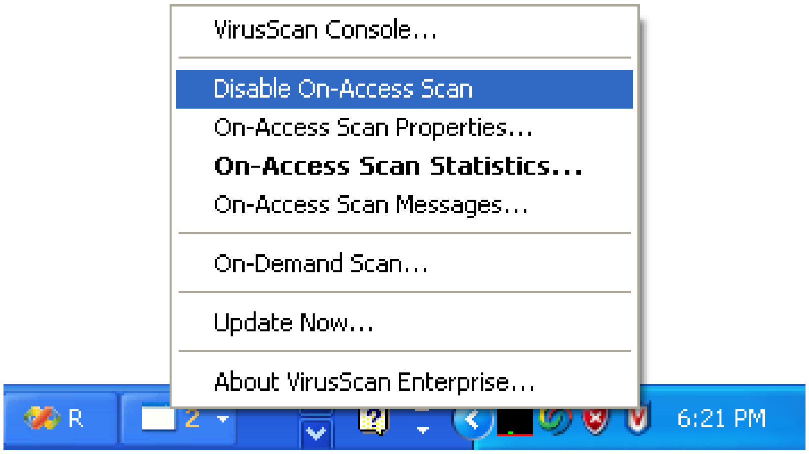
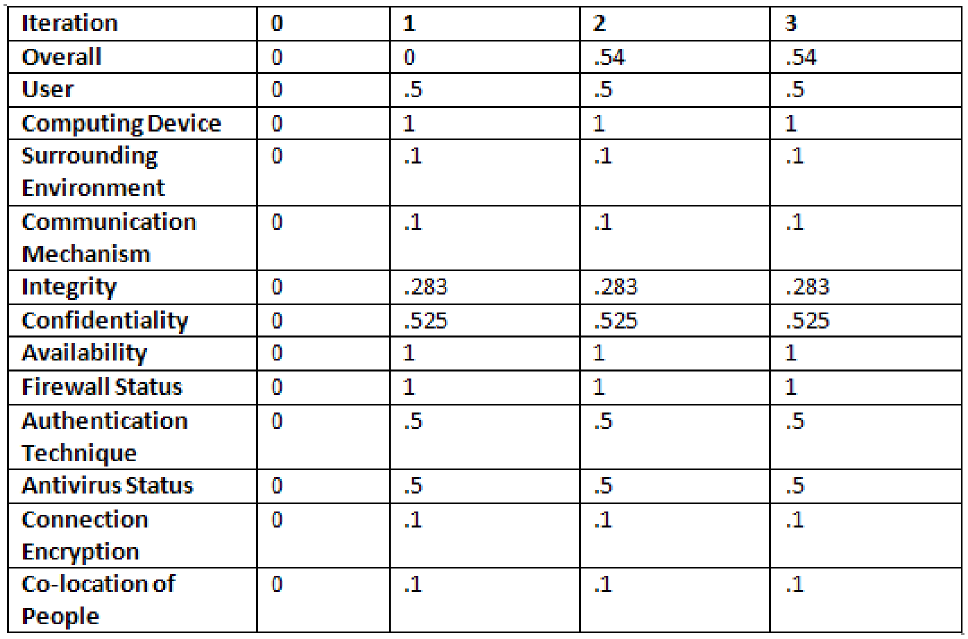

VirusScan Console...
Disable On-Access Scan
On-Access Scan Properties...
On-Access Scan Statistics...
On-Access Scan Messages...
On-Demand Scan...
Update Now...
About VirusScan Enterprise...
R
2
6:21 PM

Figure 4.9: Disabling on-access scanning capability

The following table illustrates the annotation assigned to all atoms based on the updated context and the GAP in figure 3.3 .

The updated, final value of integrity is .283 and the updated, final value of confidentiality is .525. The rule to access the resource, which is "integrity <= .4, confidentiality <=.5." is no longer satisfied and therefore the Rover Server sends a message to the Rover Client on Alice's device to revoke her current access. The resource container on Alice's device informs her via popup message that her access has been revoked and closes the resource, thus ending the current session (see Figure 4.11).



Iteration	0	1	2	3
Overall	0	0	.54	.54
User	0	.5	.5	.5
Computing Device	0	1	1	1
Surrounding Environment	0	.1	.1	.1
Communication Mechanism	0	.1	.1	.1
Integrity	0	.283	.283	.283
Confidentiality	0	.525	.525	.525
Availability	0	1	1	1
Firewall Status	0	1	1	1
Authentication Technique	0	.5	.5	.5
Antivirus Status	0	.5	.5	.5
Connection Encryption	0	.1	.1	.1
Co-location of People	0	.1	.1	.1

Figure 4.10: Interpretations produced by multiple applications of $\mathbf{T}_{\Pi}$

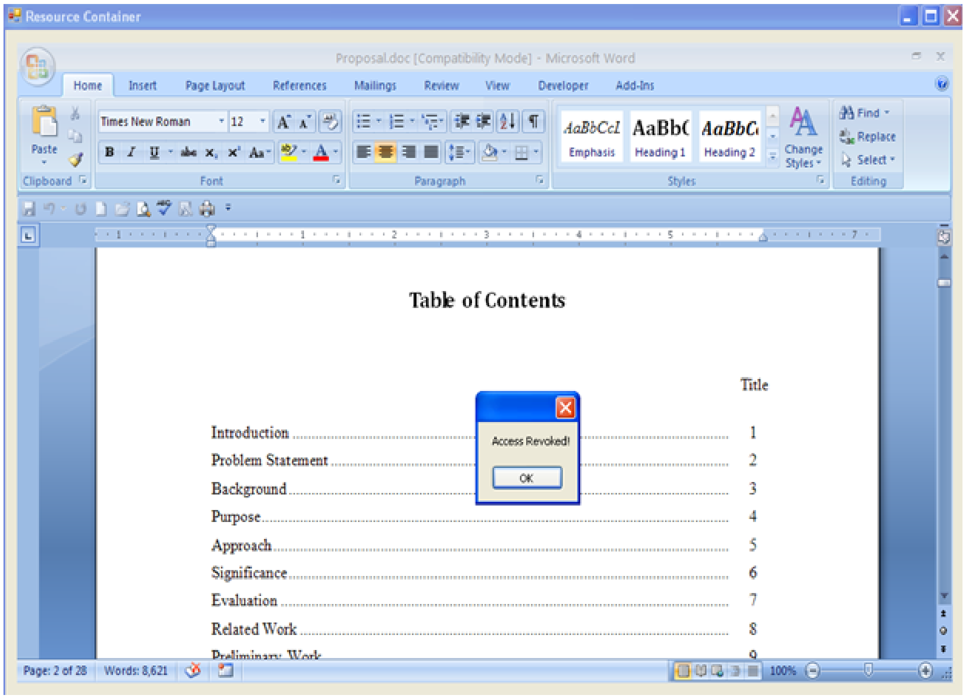


Table of Contents

Title

Access Revoked!

OK

Figure 4.11: Access Revoked after rule no longer satisfied

4.5.2 Targeted Attacks

Bob works for the same company as Alice, and like her, meets colleagues in the company lounge. While waiting in the lounge, Bob decides to read the same Agency proposal document presented in Scenario 1 on his mobile, biometric-enabled, phone. Recall that the rule associated with this particular request is ”integrity <= .4, confidentiality <=.5.”. The context providers return the following values for Bob, ’Antivirus Status: Present and up-to-date with on-access scanning enabled, Firewall Status: Present and up-to-date, Authentication Technique: Biometric-based, Connection Encryption:Encrypted, Co-location of people: Not co-located with unauthorized users’. The values assigned to integrity and confidentiality based on Bob’s context and the GAP defined in figure 3.3 are integrity = .136 and confidentiality = .200. These values satisfy the rule and access to the resource is granted. A short time later, the Agency security administrator received notification that the Agency was being targeted for cyber attacks. Accordingly, she logs into the Rover Administrator’s Interface and updates the operating condition from ’Normal’ to ’High Alert’ (see figure 4.12. This update causes the Rover Server to re-evaluate access for all Rover Clients that are currently accessing resources. This involves fetching the new access rule based on the updated operating condition and seeing if the new restriction is met. The new rule based on the access operation (read), the resource class (unclassified), and the current operating condition (high alert) is ”integrity <= .1, confidentiality <=.1. Bob no longer meets the requirements and a message is sent to the Rover Client on Bob’s device to revoke his current access. The resource controller on Bob's device informs him via popup message that his access has been revoked and closes the resource, thus ending the current session.

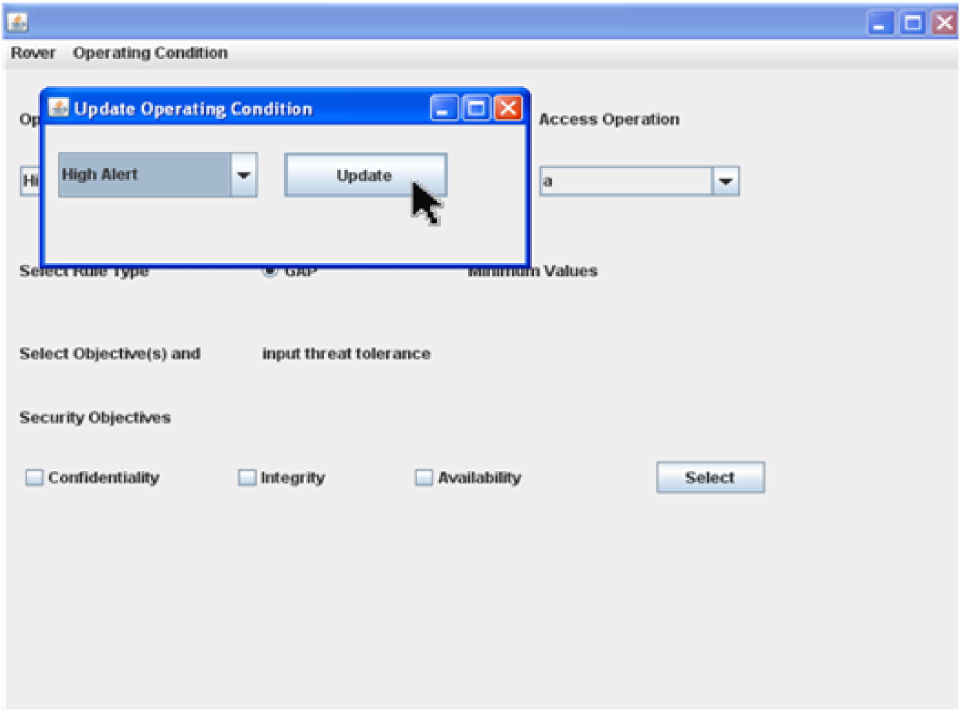


Figure 4.12: Updating Operating Condition to 'High Alert'

4.6 Summary

In this chapter we presented our implementation of the logical shrink-wrapped security framework presented in section 3.4, and shrink-wrapped access control. We described the extensions and modifications we made to our baseline system, Rover, to complete the implementation. We also detailed real-world usage cases that highlight the interaction between a user and the system, and demonstrate the feasibility of shrink-wrapped security. In the next chapter we present our evaluation of shrink-wrapped access control.

Chapter 5

The Evaluation of Shrink-Wrapped Access Control

In this chapter we present an evaluation of our implementation of shrink-wrapped access control. We first present our performance evaluation. We describe our performance objective, and experiment setup. Initial experiment results and an analysis of those results are then presented. We discuss the evaluation of our revised implementation, which resulted from analyzing the results of our initial experiment. Finally, we present a discussion of the security enhancements provided by shrink-wrapped access control.

5.1 Performance Assessment

We conducted a performance assessment to evaluate the performance of shrink-wrapped security. The specific aspect of performance we evaluated was the response time, which is the amount of (wall) time between when a resource request is made and the response is received. This is critical because although we want to prevent inappropriate access to resources, we also want to ensure timely access to resources when appropriate. The objective was to ensure the response time was acceptable. Table 5.1, below, shows the maximum acceptable response times for various categories of user interface (UI) events [15, 12, 36, 40].

Table 5.1: Acceptable response times

User Interface Event Category	**Maximum Acceptable Response Time**
Instantaneous response - Events for which users should feel that they are directly and instantaneously manipulating objects in the UI. For example, the time from when a user selects a column in a table until that column is highlighted, or otherwise provides feedback that it is selected.	.1 second
Immediate response - Events that users typically perceive as easily performed and thus would expect an immediate system response. For example, the time to resize a selected column in a table by double clicking it.	1 second
Unit task response - Events that users typically expect to take time. For example displaying a graph of the data in a table, or processing all user input to any task.	10 seconds

5.1.1 Experiment Setup

There are three main categories of techniques used to conduct performance assessments: simple timers, profilers, and instrumentation methods [58]. We chose manual instrumentation because it provided the most control and granularity of what to measure. We added code to the Rover Client and Rover Server to capture the time to complete the following tasks involved in the execution of shrink-wrapped security (see figure 4.4):

- Securely retrieve context from the Rover Client Agent
- Securely retrieve context from an external Context Providing Service
- Securely retrieve local context from the Rover Server
- Run the GAPS Reasoning Engine
- The overall response time

We used three devices for the experiment, one to host the Rover Client, one to host the Rover Server, and one to host a Context Providing Service. Table 5.2 describes these devices. The Rover Client was hosted on a device running the Windows operating system, with 1 GB RAM, a 3.20 GHz processor, and a 233 GB hard drive. The Rover Server was hosted on a device running the openSuse operating system, with 3.2 GB RAM, a 3.00 GHz dual-core processor, and a 208 GB hard drive. The Context Providing Service was hosted on a device running the openSuse operating system, with 1 GB RAM, a 3.2 GHz processor, and an 89 GB hard drive. All three devices were connected via Ethernet.

For each of the following configurations, we sent 100 resource requests en block and measured the amount of time taken to complete the previously listed tasks:

- Shrink-Wrapped Rover System utilizing 5 contextual attributes
- Shrink-Wrapped Rover System utilizing 10 contextual attributes

Table 5.2: Summary of Devices

	Rover Client	**Rover Server**	**Context Providing Server**
Operating System	Windows	openSUSE	openSUSE
Processor	3.2 GHz	3.00 GHz (dual-core)	3.2 GHz
RAM	1 GB	3.2 GB	1 GB
Hard Drive	233 GB	208 GB	89 GB

- Shrink-Wrapped Rover System utilizing 15 contextual attributes

The Rover Client, Rover Server and Context Providing Service were restarted after each 100 requests to prevent previous runs from influencing subsequent runs. For each of the configurations, the same access control rule and GAP was used, and context from the Rover Server, Rover Client, and Context providing service was utilized.

5.1.2 Experiment Results

Figure 5.1 illustrates the results of our assessment. The overlapping column chart shows the average time in milliseconds (ms) to complete the previously mentioned tasks for each configuration. As we measured the wall time, which includes all system overhead, such as system calls and context switching, we also measured the system load on the devices hosting the Rover Client, Rover Server, and Context Providing Service during each experiment (see Figure 5.2). We used the processor queue length, the number of threads in the processor queue, to approximate the system load.

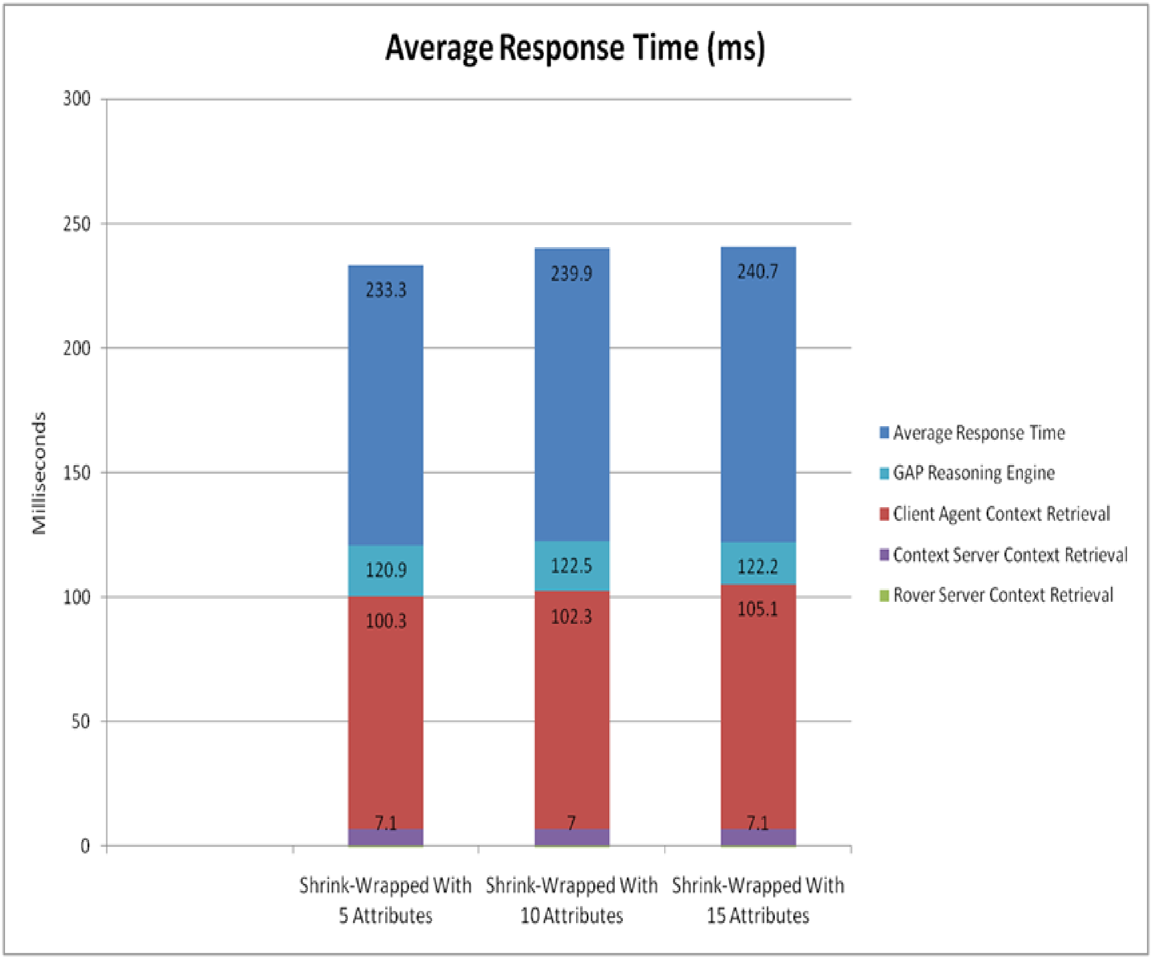


Figure 5.1: Average response time in milliseconds

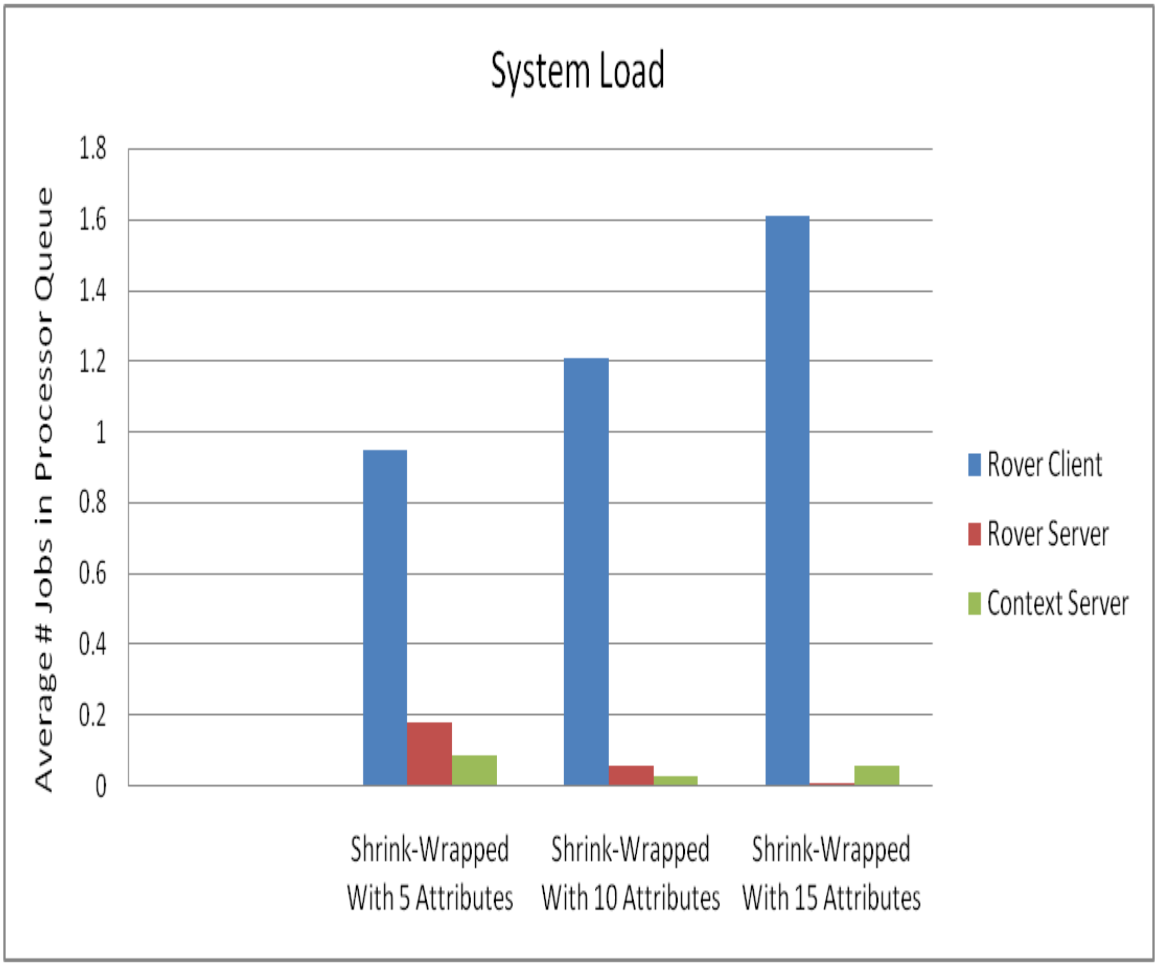


Figure 5.2: System load of devices during experiment

5.1.3 Analysis of Results

The average total response times of the shrink-wrapped system utilizing 5, 10 or 15 contextual attributes were comparable, ranging from 233.3 ms to 240.7 ms. The proximity of response times for the different shrink-wrapped system configurations illustrates that adding additional contextual attributes to the system has minimal effect on the response time. As illustrated in Figure 5.2 the load on the devices was comparable for each configuration, so we do not believe varying loads significantly affected the experiments. The load on the machine hosting the Rover Client slightly increases as the number of attributes increase, likely because of the context agent hosted on the client performing additional WMI queries. The response times of the shrink-wrapped security configurations are well under the acceptable response times of 1 second for easily completed tasks, and 10 seconds for completing tasks a typical user would expect to take time.

Although the response times observed in our experiment were acceptable, they could be improved. The amount of time to securely retrieve context from the Rover Client was significantly longer than the time to retrieve local context from the Rover Server, and context from the Context Server. This was due to the Rover Client Agent performing WMI queries each time a context request was received. We implemented caching of security-relevant context on the Rover Client Agent to improve response time. We updated The Rover Client Agent to query WMI for the state of relevant contextual attributes when the Rover Client initially starts and cache those values. The Rover Client Agent also subscribes to notifications of changes to those attributes, so the cache is automatically updated when a change in relevant context occurs, maintaining the freshness of the values. Figure 5.3, below, shows the results of our experiment with caching on the Rover Client Agent. As illustrated, caching reduces the time to retrieve context from the Rover Client Agent by almost 100 milliseconds, reducing the total response time by approximately the same amount. It also results in a lighter load on the Rover Client device (see figure 5.4).

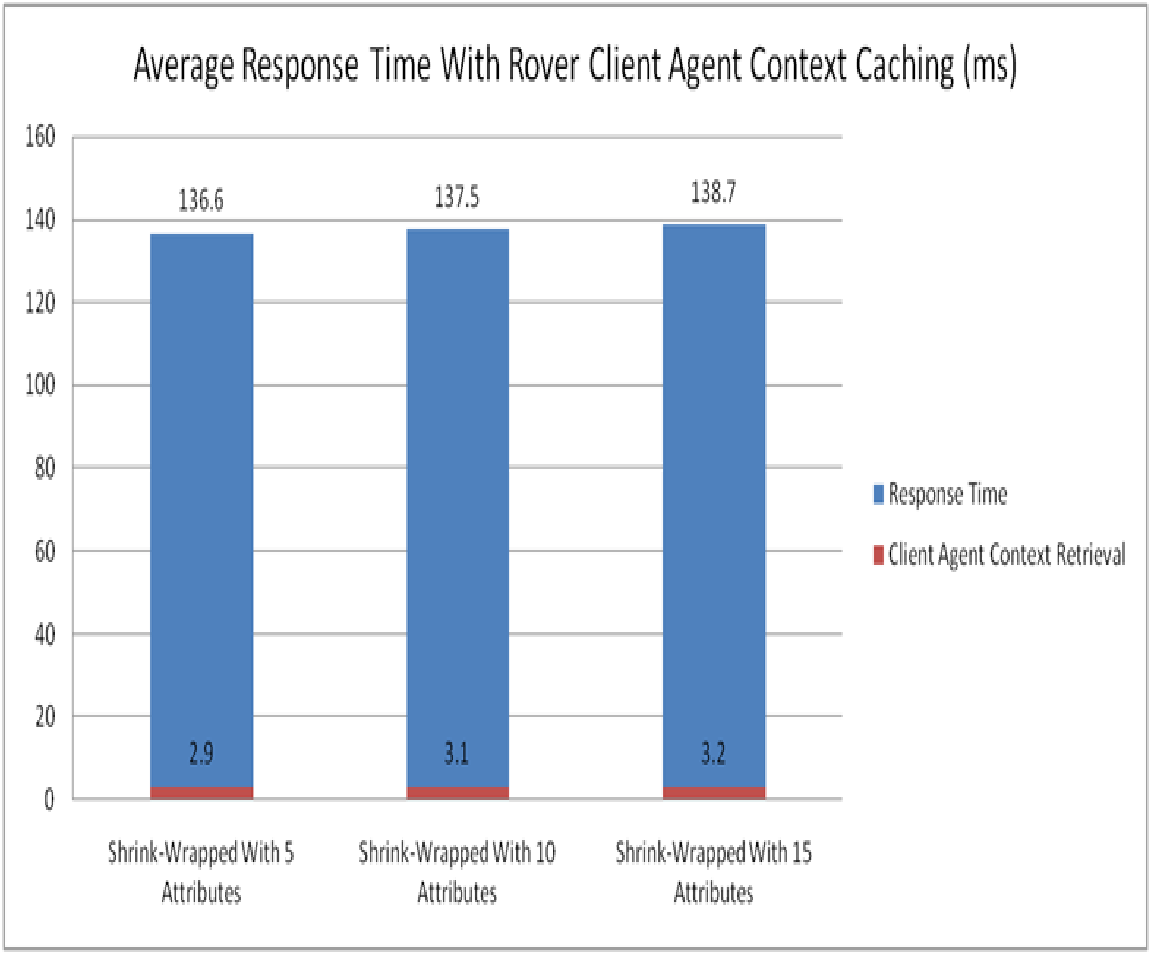


Figure 5.3: Response times with Rover Client Agent caching

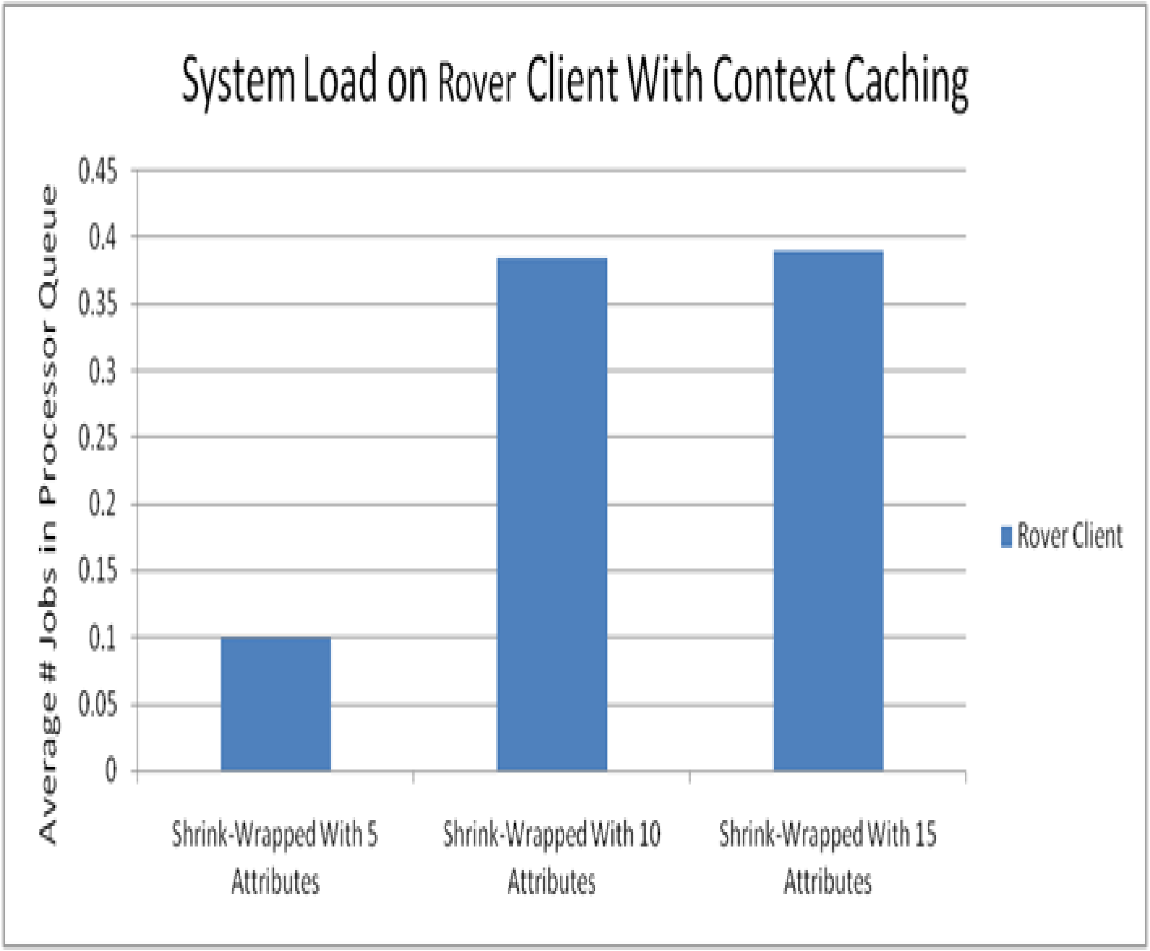


Figure 5.4: System load on Rover Client with context caching

Further improvement in response time could be achieved if the Rover Server pre-established a secure connection with the remote Context Server, because a majority of the time to retrieve context, approximately 6 milliseconds, was spent establishing the secure connection. Clearly, connection establishment time is not expended retrieving local context, and a connection is already established between the Rover Server and Rover Client when context is retrieved. As a majority of the time communicating with external context providers is spent establishing secure communication, when adding additional attributes it would be ideal if the context providers can be consolidated as much as possible to minimize that overhead.

5.2 Security Enhancements

There are no well-accepted metrics for measuring the effectiveness or functional quality of an access control service [25, 26]. Nonetheless, we discuss the security enhancements of shrink-wrapped access control with respect to well established security principles that deal with fundamentals concepts, not just a particular implementation.

The Principle of Least Privilege states that users should operate with the least set of privileges necessary to complete a job [45]. The intent of the principle is to limit the accidental and intentional improper use of privileges. This principle, like traditional security mechanisms, originated when users were typically computing in static environments. Shrink-wrapped access control presents another dimension on which to limit a user's privileges that is more aligned with users computing in dynamic situations - their context. This additional dimension allows not only a user's task, but a user's current situation to be taken into consideration to appropriately restrict privileges, further limiting the improper use of privileges.

There are two primary approaches to security in dynamic computing environments. One approach is to require systems to implement the most stringent security controls, based on the highest threat level a user may be exposed to in all possible scenarios. This approach is very rigid and has the following disadvantages:

- Security controls often reduce the functionality and usability of a system, and unnecessary and excessive deployment of security controls often reduces user acceptance of a system [23].

- This approach does not easily adjust to the fact that the assumed strength (and association with stringency) of security controls constantly change, for example because of newly discovered vulnerabilities, the development of more efficient algorithms, and advances in computer processing speed.

The other approach, involves adopting security to the changing situation. This is the approach shrink-wrapped access control takes and it facilitates another fundamental security principle that states that security in any system should be commensurate with its risks [1] [42]. With shrink-wrapped access control, security-relevant contextual attributes, which by definition allow estimating the likelihood of threats to a system being exploited, are utilized. Resources are categorized based on the impact of an exploitation, and there are different policies based on the current operating condition. If the strength of a security control changes, the likelihood associated with the value of the corresponding contextual attribute is simply updated to reflect the change. Based on these factors, shrink-wrapped access control allows adjusting security measures, which in the case of access control equates to allowing, denying, or revoking permissions, so that they are commensurate with risk.

Finally, shrink-wrapped access control allows the automatic enforcement of policies that were previously left up to the user to enforce. Removing the burden of

[1] Recall that risk is a function of the likelihood of threats to the system being realized and the resulting impact.

security enforcement from the user and putting it in the security service is a major benefit, as users are notoriously the weakest link in security.

5.3 Summary

In this chapter we presented our evaluation of shrink-wrapped access control. We presented a discussion of the security enhancements of shrink-wrapped access control. We also illustrated that it can provide access decisions well below established acceptable response times, further illustrating the feasibility of shrink-wrapped security. In the next chapter we present an overview of research in areas that are relevant to shrink-wrapped security.

Chapter 6

Related Work

In this chapter we present an overview of related work in the research areas that are primary to shrink-wrapped security. We include the following areas: utilizing more comprehensive context, focusing on security-relevant context, context-aware access control, and using logic programming in context-aware access control.

6.1 More Comprehensive Context

Since the early days of context-aware computing, researchers have realized that a more comprehensive notion of context is necessary. In one of the first papers to use the term “context-aware computing”, Schilit et al. state that contextual attributes other than location are of interest including: lighting, noise level, network connectivity, communication costs, communication bandwidth, and the social situation. In this research, the authors were focused on applications to enhance the quality of life of the intended users, such as proximate selection and automatic contextual reconfiguration [47]. Schmidt et al. also advocate a wider notion of context and argue that more than location is needed to approximate a situation, especially for “ultra mobile computing” where users operate their computing devices while on the move. This work stresses the importance of additionally considering the physical conditions in a given environment. Enhancing the quality of life of users through applications that perform adaptive user interfaces, context-aware communication, and proactive application scheduling was the focus of this work [51]. Additional researchers have advocated a broader concept of context as well [6, 52], but we are unaware of any that have specifically focused on context that is relevant to security.

6.2 Security-relevant Context

The importance of context that is relevant to security was realized as early as 2000 when Generalized Role-Based Access Control (GRBAC), one of the first incorporations of context-awareness into security, was proposed. GRBAC is an extension of role-based access control that includes a new type of role, an "environment role", to capture security-relevant context of the environment in which an access request was made [21]. Subsequently, other researchers have highlighted security-relevant context in their work [53, 18, 41, 24, 44]. For example, the Context-Aware Role-Based Access Control model (CGRBAC) provides access control for web services and also extends RBAC to include a role for capturing security-relevant context. The CGRBAC model addresses the unique access control requirements of global web services that are composed of atomic web services [24].

Approaches like those discussed in [53, 18, 13] particularly highlight the significance of context in security services as they are context-centric and completely replace attributes that are traditionally used in security-related decisions with context. This type of approach is useful when the identities of users are not known in advance or are not trustworthy. The Contextual Attribute-based Access Control (CABAC) model uses context to specify and enforce authorization policies for mobile users [18]. Toninelli et al. propose an approach similar to Role-Based Access Control (RBAC) that uses context to group users and assign permissions [53]. Bhatti et al. use context to determine the trust level of users that are not known a priori [13]. Undoubtedly, researchers have recognized the significance of security-relevant context, but the fact that the term had yet to be specifically defined left it ambiguous and made the systematic identification and use of such context nearly impossible.

Mostfaoui defined a security context:

A security context is a set of information collected from the users environment and the application environment and that is relevant to the security infrastructure of both the user and the application [39].

This definition was too vague for our objectives because it is not of much assistance when trying to determine what is and is not security-relevant context.

6.3 Context-Aware Access Control

There are existing context-aware access control systems. We contend that although they consider context, they are still relatively static due to a combination of factors, including: limited context use, infrequent context consideration, and lack of continuity in resource control. For example, in [19] the authors present implementation work, based on their previously proposed GRBAC model, in which they focus on securing smart homes. They use context to adopt security to changing conditions when requests are made. In [14], the authors extend their previous work on an XML-based RBAC framework to incorporate the use of context. They use context to establish a trust level for users that are not known in advance, and subsequently use these trust levels to control access to web services. In these approaches, context is only considered during the initial request to access a resource. This is not sufficient in highly dynamic computing environments because relevant context may change during a session and warrant a re-evaluation of access decisions.

There are context-aware access control systems that involve monitoring context during a session and adapting based on changes. For example, in [33] the authors present an authorization framework in the context of the Gaia operating system for active (smart) spaces. Gaia brings the functionality of an operating system to physical spaces. In their system, publish-subscribe event channels are used to control the system and disseminate information resources. They use cryptographic mechanisms to enforce dynamic authorizations by controlling the ability to send and receive messages via a secure distribution system. Users are provided symmetric keys to access encrypted resources that they have authorization to access. Key revocation is used to deny access when permissions change, so that future requests will be denied. In [53] the authors present an approach to securely share resources in ad-hoc networking scenarios in which users may not be known in advance or their identity may not be trusted. They use a resource-centric approach to context-aware access control, basing access decisions on the context of a protected resource. Policy reevaluation can be triggered by a change to resource context. Although these approaches are more dynamic than those that only consider context during the initial request for a resource, they do not deal with continuity in resource access and the control necessary to allow access to be revoked when appropriate [46].

6.4 Logic Programming

Previously, logic programming has been employed for context-aware access control in various ways. For example, in [43] the authors use first order logic to model context as first order predicates. For instance, *Location(Chris, entering, room 3231)* represents the situation where a user, Chris, is entering room 3231. They have integrated this technique into their smart space framework, GAIA, and use it to perform a set of actions when an associated context expression becomes true. They have also integrated their modelling approach into a security architecture, Cerberus, in which they focus on authentication and access control [7]. In Cerberus, policies are written as rules in first order logic. Loke introduced an extension to Prolog, which is a general purpose logic programming language, called LogicCAP (short for Logic programming for Context-Aware Pervasive applications)[35]. The extension is used to provide native support for context-aware applications in logic programming. A new operator, “in situation”, is introduced to facilitate reasoning about an entity being in a predefined situation. In [53] the authors propose a context-centric approach for access control in ad-hoc scenarios. Their approach is inspired by RBAC, but uses context instead of roles to provide a level of indirection between entities and permissions. They use a combination of description logic and logic programming to specify and evaluate access control policies. However, we are unaware of any previous work that utilizes *annotated* logic in context-aware security decision making. As a result, contextual attributes must be identified with a boolean variable instead of a range of values as we do here by leveraging an annotated logic. An extension to annotated logic programs, GAPs allow variables and evaluable functions to be used as annotations. As such, our use of GAPs allows an administrator more flexibility in deciding how various contextual attributes are combined, which can lead to the creation of policies that are not possible to express in other frameworks.

6.5 Summary

In this chapter we have provided an overview of related research areas. It was not our goal to provide an exhaustive survey of the state-of-the-art in context-aware security, but to focus on those areas that are key to shrink-wrapped security. We discussed research that focused on the necessity of using a more comprehensive set of context when developing context-aware systems, and approaches that highlight the importance of context that is specifically useful for context-aware security. We presented research focusing on context-aware access control, and using logic programming to incorporate context into security policies. In the next chapter we present our conclusions and future work.

Chapter 7

Conclusions and Future Work

In this chapter we present an overview of our research and discuss ways that it can be expanded with future work.

7.1 Research Overview

The mobile workforce, which consists of employees who use mobile computing devices to connect with the enterprise, and have highly dynamic computing environments, is rapidly increasing. To enable effective security for such users it is important to address the disparity between traditional, static approaches to security and the dynamic computing environment of this emerging user group. In this dissertation, we have presented our approach to addressing this need with a novel security paradigm, shrink-wrapped security, that involves utilizing context to provide a tight coupling between a user's current situation and security.

We presented the components necessary to realize shrink- wrapped security. We presented our approach to addressing a key challenge in context-aware system development by defining what constitutes relevant context for security services, and developing a taxonomy and corresponding guidelines to facilitate its identification. We presented a flexible technique that speaks to the fact that security is not binary in nature, to incorporate the use of security-relevant context into security policies.

Our technique is practical as it allows administrators to obtain the value of each attribute without explicitly including each attribute in a policy. We also presented a logical framework to facilitate the secure acquisition, monitoring and use of context, designed with the with the resource constraints of mobile devices in mind.

Finally, we presented the implementation and evaluation of shrink-wrapped access control. We detailed the extensions of the Rover system that were necessary to implement our framework and shrink-wrapped access control. We presented real-world usage cases that highlighted the interaction between a user and the system, and demonstrated the dynamicity of the system in various situations. Security enhancements were discussed and we illustrated the ability of our system to provide access decisions well below established acceptable response times, illustrating the feasibility of shrink-wrapped security.

In summary the contributions of this dissertation include the following:

- A novel security paradigm, shrink-wrapped security, which involves utilizing context to tightly fuse a user's situation and security, to address the security challenges associated with the highly dynamic computing environment of the emergent mobile workforce.

- A usable definition of security-relevant context, along with goal oriented guidelines and a corresponding taxonomy to facilitate the systematic identification of contextual attributes that are most pertinent to a security service. These contributions deal with a key challenge of context-aware system development - identifying relevant context.

- A context acquisition and management framework to facilitate the development and use of shrink-wrapped security services for the mobile workforce. The layered architecture of this framework supports secure context acquisition, utilization, and monitoring by security services and was designed with the resource constraints of mobile devices in mind.

- An approach based on logic programming to practically incorporate the use of security-relevant context into the security policies that govern security services. This technique is aligned with the shrink-wrapped security concept of utilizing a comprehensive set of relevant context, while remaining practical and manageable by abstracting relevant contextual attributes to a security level associated with the objectives of a security service.

- The implementation and evaluation of shrink-wrapped access control, which serves as a practical demonstration of the feasibility of shrink-wrapped security.

7.2 Future Work

We feel that our work on shrink-wrapped security has laid a strong foundation towards enabling appropriate security for mobile users with dynamic computing environments. This field has a large scope and there several ways that our research can be expanded and different areas that warrant further investigation. We present some of those areas in this section.

The current implementation of the resource container component of our architecture, which is used to maintain control of a resource on a user device, only works with Office documents and PDF documents. Making the resource container extensible by allowing the development of plugins for new data types at various levels of abstraction should be explored.

We currently obtain the likelihood associated with each value of a security-relevant contextual attribute from a survey of security experts. Refining these values by utilizing data from actual exploits is an area for future research. Improvements in forensics tools and utilizing sensors on devices and networks may facilitate the collection of useful data that will facilitate this process by providing information that allows determining which vulnerabilities contributed to an exploitation.

Machine learning techniques that could make a shrink-wrapped security system more 'intelligent' should be investigated. For example, machine learning could be used to facilitate automatically generating likelihood values for new attributes or new values of an existing attribute. Such techniques could also be used for determining maximum tolerable threat levels for new operating condition, resource category, access operation combinations, based on values for existing combinations.

Finally, this dissertation has focused on access control for the mobile workforce. Investigations into shrink-wrapping other security services, like authentication and encryption, in different domains, such as smart homes, are areas for future exploration. This would not require an adjustment to our framework, or guidelines for identifying relevant context, but the domain specific taxonomy would have to be updated to add additional security objectives, if necessary, and domain specific threats to those objectives would have to be identified. Of course, the implementation would be specific to the security service.

Appendix A

Repository of Security-Relevant Context

Name: Connection Encryption

Relevant Entity: Communication Mechanism

Affected Security Objective(s): Integrity - altering resource while in transit, Confidentiality- resource sniffing

Explanation: The connection encryption impacts the likelihood of a malicious party being able to access and/or alter information on the path to the user, resulting in a breach of integrity and confidentiality.

Value(s): Not Encrypted, Encrypted

Name: Networks Traversed

Relevant Entity: Communication Mechanism

Affected Security Objective(s): Integrity - altering resource while in transit, Confidentiality- resource sniffing

Explanation:

Value(s): No untrusted networks traversed, Untrusted networks traversed

Notes: The decision of whether a given network is trusted or not is organization-dependant. Some organizations consider any network outside of their domain untrusted

Name: Antivirus Status

Relevant Entity: Computing Device

Affected Security Objective(s): Confidentiality - unauthorized access to an authorized user's computing device, Integrity-altering resource once at user's computing device, Availability

Explanation: The antivirus software status may implicate the increased probability of the computing device being infected with malicious software. The antivirus status does not, however, indicate what type of malicious software the device may contain. For example, the device could contain malware that either steals, alters, or destroys information.

Value(s): Not present, Present but not up-to-date, Present and up-to-date, Present and up-to-date with on-access scanning enabled

Notes: Up-to-date can refer to software and signature version

Name: Firewall Status

Relevant Entity: Computing Device

Affected Security Objective(s): Confidentiality - unauthorized access to an authorized user's computing device, Integrity-altering resource once at user's computing device, Availability

Explanation: A firewall can help screen out malicious Internet traffic from reaching a user's device. The firewall status may implicate the increased probability of the computing device being infected with malicious software. The firewall status does not, however, indicate what type of malicious software may have reached the user's device. For example, the device could contain malware that either steals, alters, or destroys information.

Value(s): Not present, Present but not up-to-date, Present and up-to-date, Present and up-to-date with approved configuration

Notes: Approved configuration (e.g., list of ports that should be blocked) is organization-dependant

Name: Current Processes

Relevant Entity: Computing Device

Affected Security Objective(s): Confidentiality - unauthorized access to an authorized user's computing device, Integrity-altering resource once at user's computing device, Availability

Explanation: Extraneous processes may indicate that the computing device is infected with malicious software

Value(s): No extraneous processes, Extraneous processes

Notes: The determination of what is extraneous or not is organization-dependent. Techniques such as developing a profile for each user to determine if various processes are normal or not can be employed. The determination should be based on more than simply process name because some malicious software can masquerade as a legitimate process.

Name: Strength of Cryptographic Parameters [1]

Relevant Entity: Communication Mechanism

Affected Security Objective(s): Integrity - altering resource while in transit, Confidentiality- resource sniffing

Explanation: The strength of the cryptographic parameters, which can be determined by such things as the encryption algorithm and key length, implicate the likelihood that an unauthorized entity can successfully break the encryption to retrieve or alter the encrypted resources resulting in a breach of confidentiality and/or integrity

Value(s): Weak, Strong

Notes: The determination of weak or strong will vary with time depending on the emergence of new exploits and increases in computing capability

Name: Password Strength[1]

Relevant Entity: User

Affected Security Objective(s): Confidentiality - impersonating an authorized user to gain access, Integrity - impersonating an authorized user to make unauthorized alterations

Explanation: The strength of the password impacts the likelihood of an unauthorized user guessing or brute forcing the password to impersonate an authorized and gaining access to resources

[1] This contextual attribute is conditional, which means that its relevancy depends on the value of another attribute. For example, if the value of Connection Encryption = Encrypted, then the attribute Strength of Cryptographic Parameters is relevant, otherwise it is not.

Value(s): Weak, Medium, Strong

Notes: Strength determination is organization-dependant. There are several online password strength assessors

Name: Open Ports

Relevant Entity: Computing Device

Affected Security Objective(s): Confidentiality - unauthorized access to an authorized user's computing device, Integrity-altering resource once at user's computing device, Availability

Explanation: Open ports are a target for hackers and viruses that use port scanning to identify running services on a host to attempt to compromise it. The existence of open extraneous open ports may implicate the increased probability of the computing device being infected with malicious software

Value(s): Extraneous ports open, No extraneous ports open

Notes: The determination of what is extraneous or not is organization-dependant. It will depend on such things as what services they use

Name: Currency of System Patches

Relevant Entity: Computing Device

Affected Security Objective(s): Confidentiality - unauthorized access to an authorized user's computing device, Integrity-altering resource once at user's computing device, Availability

Explanation: A patch is a small piece of software that is used to correct a problem (often security vulnerability) with a software program or an operating system. If patches are not up-to-date, the system is vulnerable to the issues addressed in the patches that have been released. The currency of system patches may implicate the increased probability of the computing device being infected with malicious software. The currency of system patches does not, however, indicate what type of malicious software the device may contain. For example, the device could contain malware that either steals, alters, or destroys information.

Value(s): Patches not up-to-date, Patches up-to-date

Notes: Up-to-date doesn't necessarily refer to having the latest vendor released patches. For example, some organizations purposefully choose not to install some patches as they create problems

Name: File Sharing Settings

Relevant Entity: Computing Device

Affected Security Objective(s): Confidentiality - access to an authorized user's computing device, Integrity-altering resource once at user's computing device

Explanation: A user may have file sharing enabled on their device which allows them to share files with local or remote users. If such sharing is enabled, the level of sharing implicates the likelihood that some unauthorized user may access and/or alter a resource, resulting in a breach of confidentiality and/or integrity

Value(s): File-sharing disabled, Read-only file sharing of resource location, Full-access file sharing of resource location

Name: Device Authentication Requirements

Relevant Entity: Computing Device

Affected Security Objective(s): Confidentiality-unauthorized access to an authorized user's computing device, Integrity-altering resource once at user's computing device

Explanation: If an unauthorized party gains physical access to the device, the device authentication requirements impact the likelihood that they can gain unauthorized access to the content of the device. With this access they can also make unauthorized alterations to resources

Value(s): No device authentication, Authentication required on initial login, Authentication required on initial login and after x minutes of inactivity

Name: Co-location of People

Relevant Entity: Surrounding Environment

Affected Security Objective(s): Confidentiality - indirect access through proximity to an authorized user

Explanation: If the user is co-located with people lacking the proper authorization to access a resource, then it is possible for those people to obtain unauthorized access. For example a co-located person can look at the screen of the requesting user's computing device.

Value(s): Co-located with unauthorized user(s), Not co-located with authorized user(s)

Name: Co-location of Devices

Relevant Entity: Surrounding Environment

Affected Security Objective(s): indirect access through proximity to an authorized user

Explanation: If the user is collocated with certain devices, such as surveillance cameras or recording devices, then it is possible for an unauthorized party to gain indirect access to a resource

Value(s): Co-located with surveillance equipment, Not co-located with surveillance equipment

Name: Internet Threat Level

Relevant Entity: Surrounding Environment

Affected Security Objective(s): Confidentiality - unauthorized access to an authorized user's computing device, Integrity-altering resource once at user's computing device, Availability

Explanation: The threat level indicates the status of malicious traffic that threatens the global Internet and communications network. The value of the threat level impacts the likelihood of computing devices in general being infected with malicious software.

Value(s): Organization-dependant

Notes: Different organizations (e.g. Symantec, SANS Internet Storm Center) define various threat levels. The organization's Security Administrators can also define and maintain threat levels.

Name: Authentication Technique

Relevant Entity: User

Affected Security Objective(s): Confidentiality - impersonating an authorized user to gain access, Integrity - impersonating an authorized user to make unauthorized alterations

Explanation: The strength of the authentication technique implicates the ease with which an unauthorized user can impersonate an authorized user and gain access to resources and possibly make unauthorized modifications

Value(s): Password-based, Biometric-based, Certificate-based, Token-based, Multifactor

Name: Login Attempts

Relevant Entity: User

Affected Security Objective(s): Confidentiality - impersonating an authorized user to gain access, Integrity - impersonating an authorized user to make unauthorized alterations

Explanation: A large number of failed login attempts prior to successful login, may indicate that the user is being impersonated by an unauthorized user that will then be able to access resources and possibly make unauthorized modifications

Value(s): Below x, Above x

Notes: X is determined by administrators

Name: Duress Status of User

Relevant Entity: User

Affected Security Objective(s): Confidentiality - unauthorized access to an authorized user's computing device, indirect access through proximity to an authorized user, Integrity - altering resource once at user's computing device

Explanation: Whether or not a user is under duress implicates the likelihood of an unauthorized party gaining access to resources. If a user is under duress, by definition she is not acting according to her free will, she is being forced to do something. An unauthorized party can force an authorized user to request access to resources and subsequently access them (either indirectly through proximity or directly via the user's computing device) or force unauthorized modification.

Value(s): Not under duress, Under duress

Appendix B

Definition of Terms

Agent : A program that performs some information gathering or processing task in the background

Authorized : A system entity or actor that has been granted the right, permission, or capability to access a system resource

Control : An action, device, procedure, or technique that removes or reduces vulnerability

Permission : An authorized interaction that a subject can have with an object

Risk : The combination of the probability of an event and its consequence

Security policy : The set of rules and practices that regulate how an organization manages, protects, and distributes information.

Security service : A capability that supports one, or more, security requirements, such as confidentiality, integrity, and availability. Examples of security services are encryption, access control, and authentication.

Sniffing : monitoring or eavesdropping on electronic transmissions

Threat : A set of circumstances that has the potential to cause loss or harm. A potential cause of an unwanted impact to a system or organization

Vulnerability : A flaw or weakness in system security procedures, design, implementation, or internal controls that could be exercised and result in a security breach or a violation of the system's security policy

Appendix C

Survey Results

We conducted a survey of security professionals to determine how the values of a contextual attribute affect the likelihood of various security threats being exploited and the relevancy of each attribute. The group consisted of individuals with at least three years of experience doing security-related work from various sectors, including industry, government, and military. The following are the results of the survey.

Please select the sector you work in

Value	Count	Percent %
Industry	1	9.1%
Academia	1	9.1%
Government	1	9.1%
Military	5	45.5%
Other (please specify)	3	27.3%

Statistics	
Total Responses	11

Job Title

Count	Response
1	Assistant Professor
1	Associate
1	Chief, International Interoperability and Engagement
1	Cyber Security Analyst
1	Director IT Risk Management
1	Information Assurance Manager
1	Lead information security analyst
1	Network Engineer

Count	Response
1	Research Programmer
1	Telecommunication Systems Engineer
1	VP, Strategic Cyber Initiatives

Years of experience doing security-related work

Count	Response
1	10
1	11
1	12
1	15
1	28
1	3
1	4
2	6
1	7
1	9

Based on the following values of the connection's encryption, please rate the likelihood of a malicious party being able to access and/or alter information on the path to the user

	Low (0-10%)	Medium (>10 - 50%)	High (>50%)	Total
Not encrypted	0.0% 0	18.2% 2	81.8% 9	100% 11
Encrypted	90.9% 10	9.1% 1	0.0% 0	100% 11

Please rate the relevance of the connection's encryption status

Value	Count	Percent %
Weak relevance	1	9.1%
Medium relevance	1	9.1%
Strong relevance	9	81.8%

Statistics	
Total Responses	11

Based on the following values of the networks traversed, please rate the likelihood of a malicious party being able to access and/or alter information on the path to the user

	Low (0-10%)	Medium (> 10-50%)	High (> 50 â€“ 100%)	Total
No untrusted networks traversed	**45.5%** 5	**54.5%** 6	**0.0%** 0	**100%** 11
Untrusted networks traversed	**9.1%** 1	**27.3%** 3	**63.6%** 7	**100%** 11

Please rate the relevance of the networks traversed

Value	Count	Percent %
Weak relevance	1	9.1%
Medium relevance	4	36.4%
Strong relevance	6	54.5%

Statistics	
Total Responses	11

Based on the following values of the antivirus status of a user's computing device, please rate the likelihood of the computing device being infected with malicious software.

	Low (0-10%)	Medium (> 10-50%)	High (> 50 â€“ 100%)	Total
Not present	**0.0%** 0	**0.0%** 0	**100.0%** 11	**100%** 11
Present but not up-to-date	**0.0%** 0	**27.3%** 3	**72.7%** 8	**100%** 11
Present and up-to-date	**27.3%** 3	**72.7%** 8	**0.0%** 0	**100%** 11
Present and up-to-date with on-access scanning enabled	**54.5%** 6	**45.5%** 5	**0.0%** 0	**100%** 11

Please rate the relevance of the antivirus status

Value	Count	Percent %
Medium relevance	1	9.1%
Strong relevance	10	90.9%

Statistics	
Total Responses	11

Based on the following values of the firewall status of a user's computing device, please rate the likelihood of the device being infected with malicious software

	Low (0-10%)	Medium (> 10-50%)	High (> 50 â€“ 100%)	Total
Not present	**0.0%** 0	**27.3%** 3	**72.7%** 8	**100%** 11
Present but not up-to-date	**9.1%**	**45.5%**	**45.5%**	**100%**

	1	5	5	11
Present and up-to-date	27.3% 3	72.7% 8	0.0% 0	100% 11
Present and up-to-date with approved configuration	81.8% 9	18.2% 2	0.0% 0	100% 11

Please rate the relevance of the firewall status

Value	Count	Percent %
Weak relevance	1	9.1%
Medium relevance	3	27.3%
Strong relevance	7	63.6%

Statistics	
Total Responses	11

Based on the following values of the current processes running on a computing device, please rate the likelihood of the device being infected with malicious software

	Low (0-10%)	Medium (> 10-50%)	High (> 50 â€“ 100%)	Total
No extraneous processes	63.6% 7	27.3% 3	9.1% 1	100% 11
Extraneous processes	9.1% 1	72.7% 8	18.2% 2	100% 11

Please rate the relevance of the current processes

Value	Count	Percent %
Weak relevance	4	36.4%
Medium relevance	5	45.5%
Strong relevance	2	18.2%

Statistics	
Total Responses	11

Based on the following values of the strength of cryptographic parameters used to encrypt the connection, please rate the likelihood that an unauthorized entity can successfully break the encryption to retrieve or alter the encrypted resources

	Low (0-10%)	Medium (> 10-50%)	High (> 50 â€“ 100%)	Total
Weak	**0.0%** 0	**27.3%** 3	**72.7%** 8	**100%** 11
Strong	**100.0%** 11	**0.0%** 0	**0.0%** 0	**100%** 11

Please rate the relevance of the cryptographic parameters

Value	Count	Percent %
Weak relevance	1	9.1%
Medium relevance	1	9.1%
Strong relevance	9	81.8%

Statistics	
Total Responses	11

Based on the following values of the open ports of a computing device, please rate the likelihood of the computing device being infected with malicious software

	Low (0-10%)	Medium (> 10-50%)	High (> 50 â€“ 100%)	Total
Extraneous ports open	0.0% 0	36.4% 4	63.6% 7	100% 11
No extraneous ports open	72.7% 8	27.3% 3	0.0% 0	100% 11

Please rate the relevance of the open ports

Value	Count	Percent %
Medium relevance	5	45.5%
Strong relevance	6	54.5%

Statistics	
Total Responses	11

Based on the following values of the currency of system patches of a computing device, please rate the likelihood of the device being infected with malicious software

	Low (0-10%)	Medium (> 10-50%)	High (> 50 â€“ 100%)	Total
Patches not up-to-date	0.0% 0	9.1% 1	90.9% 10	100% 11
Patches up-to-date	54.5% 6	45.5% 5	0.0% 0	100% 11

Please rate the relevance of the currency of system patches

Value	Count	Percent %
Medium relevance	2	18.2%
Strong relevance	9	81.8%

Statistics	
Total Responses	11

Based on the following values of the file sharing settings of a computing device, please rate the likelihood of an unauthorized user accessing and/or altering a resource

	Low (0-10%)	Medium (> 10-50%)	High (> 50 â€“ 100%)	Total
File-sharing disabled	**72.7%** 8	**27.3%** 3	**0.0%** 0	**100%** 11
Read-only file sharing of resource location	**18.2%** 2	**63.6%** 7	**18.2%** 2	**100%** 11
Full-access file sharing of resource location	**0.0%** 0	**18.2%** 2	**81.8%** 9	**100%** 11

Please rate the relevance of the file sharing settings

Value	Count	Percent %
Weak relevance	3	27.3%
Medium relevance	4	36.4%
Strong relevance	4	36.4%

Statistics

Total Responses	11

Based on the following values of the authentication required to login to a computing device, please rate the likelihood of a user gaining unauthorized access to the content of the device.

	Low (0-10%)	Medium (> 10-50%)	High (> 50 â€“ 100%)	Total
No device authentication	**0.0%** 0	**0.0%** 0	**100.0%** 11	**100%** **11**
Authentication required on initial login	**9.1%** 1	**81.8%** 9	**9.1%** 1	**100%** **11**
Authentication required on initial login and after x* minutes of inactivity (*x is an organization-dependent number)	**54.5%** 6	**45.5%** 5	**0.0%** 0	**100%** **11**

Please rate the relevance of the device authentication requirements

Value	Count	Percent %
Weak relevance	1	9.1%
Medium relevance	3	27.3%
Strong relevance	7	63.6%

Statistics	
Total Responses	11

Based on the following values of the co-location of people, please rate the likelihood of an unauthorized user gaining access through proximity.

	Low (0 â€“ 10%)	Medium (>10 â€“ 50%)	High (>50%)	Total
Co-located with unauthorized user(s)	0.0% 0	45.5% 5	54.5% 6	100% 11
Not co-located with authorized user(s)	54.5% 6	36.4% 4	9.1% 1	100% 11

Please rate the relevance of the co-location of people

Value	Count	Percent %
Weak relevance	2	18.2%
Medium relevance	3	27.3%
Strong relevance	6	54.5%

Statistics	
Total Responses	11

Based on the following values of the authentication technique used by a user, please rate the likelihood of the user being impersonated

	Low (0 â€“ 10%)	Medium (>10 â€“ 50%)	High (>50%)	Total
Password-based	9.1% 1	54.5% 6	36.4% 4	100% 11
Biometric-based	81.8% 9	18.2% 2	0.0% 0	100% 11
Certificate-based	45.5% 5	45.5% 5	9.1% 1	100% 11
Token-based	63.6% 7	36.4% 4	0.0% 0	100% 11
Multi-factor	100.0% 11	0.0% 0	0.0% 0	100% 11

Please rate the relevance of the authentication technique

Value	Count	Percent %
Medium relevance	2	18.2%
Strong relevance	9	81.8%

Statistics	
Total Responses	11

Based on the following values for the number of login attempts made by a user, please rate the likelihood that the user is being impersonated

	Low (0 â€“ 10%)	Medium (>10 â€“ 50%)	High (>50%)	Total
Below x	**72.7%** 8	**27.3%** 3	**0.0%** 0	**100%** **11**
Above x	**18.2%** 2	**45.5%** 5	**36.4%** 4	**100%** **11**

Please rate the relevance of the number of login attempts

Value	Count	Percent %
Weak relevance	5	45.5%
Medium relevance	4	36.4%
Strong relevance	2	18.2%

Statistics	
Total Responses	11

Based on the following values of the password strength of the user, please rate the likelihood of the user being impersonated

	Low (0 â€“ 10%)	Medium (>10 â€“ 50%)	High (>50%)	Total
Weak	**0.0%** 0	**18.2%** 2	**81.8%** 9	**100%** **11**
Medium	**9.1%** 1	**63.6%** 7	**27.3%** 3	**100%** **11**
Strong	**81.8%** 9	**18.2%** 2	**0.0%** 0	**100%** **11**

Please rate the relevance of the password strength

Value	Count	Percent %
Medium relevance	2	18.2%
Strong relevance	9	81.8%

Statistics	
Total Responses	11

Based on the following values of the duress status of the user, please rate the likelihood of an unauthorized user gaining access to a resource by forcing an authorized user to make the requests

	Low (0 â€“ 10%)	Medium (>10 â€“ 50%)	High (>50%)	Total
Not under duress	**72.7%** 8	**18.2%** 2	**9.1%** 1	**100%** **11**
Under duress	**0.0%** 0	**36.4%** 4	**63.6%** 7	**100%** **11**

Please rate the relevance of the duress status of the user

Value	Count	Percent %
Weak relevance	4	36.4%
Medium relevance	3	27.3%
Strong relevance	4	36.4%

Statistics	
Total Responses	11

Based on the following values of the Internet threat level, please rate the likelihood of the user's computing device being infected with malicious software

	Low (0-10%)	Medium (> 10-50%)	High (> 50 â€“ 100%)	Total
Low/Green	**63.6%** 7	**27.3%** 3	**9.1%** 1	**100%** **11**
Medium/Yellow	**36.4%** 4	**54.5%** 6	**9.1%** 1	**100%** **11**
High/Orange	**9.1%** 1	**72.7%** 8	**18.2%** 2	**100%** **11**
Extreme/Red	**0.0%** 0	**36.4%** 4	**63.6%** 7	**100%** **11**

Please rate the relevance of the Internet threat level

Value	Count	Percent %
Weak relevance	6	54.5%

Medium relevance	5	45.5%

Statistics	
Total Responses	11

Based on the following values of for the devices a user is co-located with, please rate the likelihood that an unauthorized party can gain access to a resource via a co-located device

	Low (0-10%)	Medium (> 10-50%)	High (> 50 â€“ 100%)	Total
Co-located with surveillance equipment	**45.5%** 5	**36.4%** 4	**18.2%** 2	**100%** **11**
Not co-located with surveillance equipment	**45.5%** 5	**36.4%** 4	**18.2%** 2	**100%** **11**

Please rate the relevance of the co-location of devices

Value	Count	Percent %
Weak relevance	4	36.4%
Medium relevance	4	36.4%
Strong relevance	3	27.3%

Statistics	
Total Responses	11

Bibliography

[1] About wmi. http://msdn.microsoft.com/enus/library/windows/desktop/aa384642(v=vs.85).aspx. Accessed March 1, 2012.

[2] Office viewer component. http://www.ocxt.com/. Accessed July 23, 2012.

[3] Openssl - cryptography and ssl/tls toolkit. http://www.openssl.org/. Accessed July 27, 2012.

[4] Sqlite. http://www.sqlite.org/. Accessed July 27, 2012.

[5] United states code, title 44, subchapter 3542: Information security - definitions.

[6] G.D. Abowd and E.D. Mynatt. Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7(1):29–58, 2000.

[7] J. Al-Muhtadi, A. Ranganathan, R. Campbell, and M.D. Mickunas. Cerberus: a context-aware security scheme for smart spaces. In *Pervasive Computing and Communications, 2003.(PerCom 2003). Proceedings of the First IEEE International Conference on*, pages 489–496. IEEE, 2003.

[8] C.B. Almazan, M. Youssef, and A.K. Agrawala. Rover: An integration and fusion platform to enhance situational awareness. In *Performance, Computing, and Communications Conference, 2007. IPCCC 2007. IEEE Internationa*, pages 582–587. IEEE, 2007.

[9] M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007.

[10] Jakob E Bardram, Rasmus E Kjær, and Michael Ø Pedersen. Context-aware user authentication–supporting proximity-based login in pervasive computing. In *UbiComp 2003: Ubiquitous Computing*, pages 107–123. Springer, 2003.

[11] L. Barkhuus. Context information vs. sensor information: A model for categorizing context in context-aware mobile computing. *SIMULATION SERIES*, 35(1):127–133, 2003.

[12] Calum Benson, Adam Elman, Seth Nickell, and Colin Z Robertson. Gnome human interface guidelines (2.0), the gnome usability project. *Retrieved January*, 7:2010, 2004.

[13] R. Bhatti, E. Bertino, and A. Ghafoor. A trust-based context-aware access control model for web-services. In *Web Services, 2004. Proceedings. IEEE International Conference on*, pages 184–191. IEEE, 2004.

[14] R. Bhatti, E. Bertino, and A. Ghafoor. A trust-based context-aware access control model for web-services. In *Web Services, 2004. Proceedings. IEEE International Conference on*, pages 184–191. IEEE, 2004.

[15] Stuart K Card, George G Robertson, and Jock D Mackinlay. The information visualizer, an information workspace. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 181–186. ACM, 1991.

[16] G. Chen and D. Kotz. A survey of context-aware mobile computing research. 2000.

[17] K. Cheverst, N. Davies, K. Mitchell, and A. Friday. Experiences of developing and deploying a context-aware tourist guide: the guide project. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 20–31. ACM, 2000.

[18] M. Covington and M. Sastry. A contextual attribute-based access control model. In *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, pages 1996–2006. Springer, 2006.

[19] M.J. Covington, P. Fogla, Z. Zhan, and M. Ahamad. A context-aware security architecture for emerging applications. In *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*, pages 249–258. IEEE, 2002.

[20] M.J. Covington, W. Long, S. Srinivasan, A.K. Dev, M. Ahamad, and G.D. Abowd. Securing context-aware applications using environment roles. In *Proceedings of the sixth ACM symposium on Access control models and technologies*, pages 10–20. ACM, 2001.

[21] M.J. Covington, M.J. Moyer, and M. Ahamad. Generalized role-based access control for securing future applications. 2000.

[22] A.K. Dey and G.D. Abowd. Towards a better understanding of context and context-awareness. In *CHI 2000 workshop on the what, who, where, when, and how of context-awareness*, volume 4, pages 1–6, 2000.

[23] Kristian Fischer and Stefan Karsch. Modelling security relevant context an approach towards adaptive security in volatile mobile web environments. In *Proceedings of the ACM Web Science Conference*, 2011.

[24] S. Haibo and H. Fan. A context-aware role-based access control model for web services. In *e-Business Engineering, 2005. ICEBE 2005. IEEE International Conference on*, pages 220–223. IEEE, 2005.

[25] Ferraiolo D.F. Hu, V.C and D.R. Kuhn. Assessment of access control systems. Technical report, National Institute of Standards and Technology (NIST).

[26] Vincent C Hu and Karen Ann Kent. *Guidelines for access control system evaluation metrics*. US Department of Commerce, National Institute of Standards and Technology, 2012.

[27] RJ Hulsebosch, A.H. Salden, M.S. Bargh, P.W.G. Ebben, and J. Reitsma. Context sensitive access control. In *Proceedings of the tenth ACM symposium on Access control models and technologies*, pages 111–119. ACM, 2005.

[28] IDC. Worldwide mobile worker population 2011-2015 forecast. December 2011.

[29] G. Johnson, A. Agrawala, and E. Billionniere. A Framework for Shrink-Wrapping Security Services. In *Services Computing (SCC), 2010 IEEE International Conference on*, pages 639–640. IEEE, 2010.

[30] G. Johnson, P. Shakarian, N. Gupta, and A. Agrawala. Towards shrink-wrapped security: Practically incorporating context into security services. *Procedia Computer Science*, 5:782–787, 2011.

[31] G.M. Johnson. Towards shrink-wrapped security: A taxonomy of security-relevant context. In *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, pages 1–2. IEEE, 2009.

[32] M. Kifer and VS Subrahmanian. Theory of generalized annotated logic programming and its applications*. *the journal of Logic Programming*, 12(4):335–367, 1992.

[33] A. Lee, J. Boyer, C. Drexelius, P. Naldurg, R. Hill, and R. Campbell. Supporting dynamically changing authorizations in pervasive communication systems. *Security in Pervasive Computing*, pages 134–150, 2005.

[34] U. Lindqvist and E. Jonsson. How to systematically classify computer security intrusions. In *Security and Privacy, 1997. Proceedings., 1997 IEEE Symposium on*, pages 154–163. IEEE, 1997.

[35] S.W. Loke. Logic programming for context-aware pervasive computing: Language support, characterizing situations, and integration with the web. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 44–50. IEEE Computer Society, 2004.

[36] Robert B Miller. Response time in man-computer conversational transactions. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 267–277. ACM, 1968.

[37] Kazuhiro Minami and David Kotz. Secure context-sensitive authorization. *Pervasive and Mobile Computing*, 1(1):123–156, 2005.

[38] Ghita Kouadri Mostéfaoui, Mansoo Kim, and Mokdong Chung. Supporting adaptive security levels in heterogeneous environments. In *Computational Science and Its Applications–ICCSA 2004*, pages 537–546. Springer, 2004.

[39] G.K. Mostefaoui. Security in pervasive environments, what's next? *Security and Management Journal*, pages 93–98, 2003.

[40] Jakob Nielsen. *Usability engineering*. Access Online via Elsevier, 1994.

[41] S.H. Park, Y.J. Han, and T.M. Chung. Context-role based access control for context-aware application. *High Performance Computing and Communications*, pages 572–580, 2006.

[42] C.P. Pfleeger and S.L. Pfleeger. *Security in computing*. Prentice-Hall, 4th edition, 2006.

[43] A. Ranganathan and R.H. Campbell. An infrastructure for context-awareness based on first order logic. *Personal and Ubiquitous Computing*, 7(6):353–364, 2003.

[44] E. Sabbah, A. Majeed, K.D. Kang, K. Liu, and N. Abu-Ghazaleh. An application-driven perspective on wireless sensor network security. In *Proceedings of the 2nd ACM international workshop on Quality of service & security for wireless and mobile networks*, pages 1–8. ACM, 2006.

[45] Jerome H Saltzer and Michael D Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, 1975.

[46] Ravi Sandhu and Jaehong Park. Usage control: A vision for next generation access control. In *Computer Network Security*, pages 17–31. Springer, 2003.

[47] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pages 85–90. IEEE, 1994.

[48] B.N. Schilit and M.M. Theimer. Disseminating active map information to mobile hosts. *Network, IEEE*, 8(5):22–32, 1994.

[49] A. Schmidt. *Ubiquitous computing-computing in context*. PhD thesis, Lancaster University, UK, 2003.

[50] A. Schmidt. Potentials and challenges of context-awareness for learning solutions. *Proc. LWA*, pages 63–68, 2005.

[51] A. Schmidt, M. Beigl, and H.W. Gellersen. There is more to context than location. *Computers & Graphics*, 23(6):893–901, 1999.

[52] P. Tarasewich. Towards a comprehensive model of context for mobile and wireless computing. In *Proc. of AMCIS*, pages 114–124. Citeseer, 2003.

[53] A. Toninelli, R. Montanari, L. Kagal, and O. Lassila. A semantic context-aware access control framework for secure collaborations in pervasive computing environments. *The Semantic Web-ISWC 2006*, pages 473–486, 2006.

[54] A. Van Lamsweerde. Goal-oriented requirements engineering: A guided tour. In *Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on*, pages 249–262. IEEE, 2001.

[55] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems (TOIS)*, 10(1):91–102, 1992.

[56] M. Webster. Merriam-webster online dictionary. 2012.

[57] M. Weiser. The computer for the 21 st century. *IEEE pervasive computing*, 1(1):19–25, 2002.

[58] Patrick Wood. A survey of performance analysis tools. Technical report, Washington University in St. Louis.

[59] Christian J Wullems, Mark Hanmeng Looi, and Andrew J Clark. Towards context-aware security: An authorization architecture for intranet environments. 2004.

[60] K. Yue. What does it mean to say that a specification is complete? In *Proc. IWSSD-4, Fourth International Workshop on Software Specification and Design*, 1987.

[61] Guangsen Zhang and Manish Parashar. Dynamic context-aware access control for grid applications. In *Grid Computing, 2003. Proceedings. Fourth International Workshop on*, pages 101–108. IEEE, 2003.