Security Analysis of an Agent-Mediated Book Trading Application

Richard Ssekibuule and Jose Ghislain Quenum *
Faculty of Computing and Information Technology
Makerere University, Kampala, Uganda

Abstract

This paper presents a security analysis of an agent mediated application in an open distributed environment. We use a case study of a booktrading application that we implemented using AgentScape and JADE agent platforms. The paper analyzes whether security requirements, threats and countermeasures for an agent mediated application change when implemented on different types of agent platforms and presents countermeasures to generic and application specific threats.

Categories and Subject Descriptors: [Computer Science]: General Terms: Security Analysis, Agent-Mediated

IJCIR Reference Format:
Richard Ssekibuule and Jose Ghislain Quenum. Security Analysis of an Agent-Mediated Book Trading Application. International Journal of Computing and ICT Research, Special Issue Vol. 3, No. 1, pp. 67-76. http://www.ijcir.org/Special-Issuevolume3-numbe1/article7.pdf.

1. INTRODUCTION

Research and development in Agent system has seen tremendous progress in recent years leading to the development of several agent platforms such as NOMADS[Suri et al. 2000], AgentScape[Overeinder and Brazier 2005], Havana[Mahmoud and Yu 2004], JADE[Bellifemine et al. 2007] and Aglets[Lange et al. 1997]. In order for agents to execute tasks that have been assigned to them, they have to interact with other agents in the open distributed environment whose intentions could be malicious. Several researchers [Li et al. 2004; Farmer et al. 1996; Jansen 2000] have suggested solutions to different types of threats that can be anticipated in agent-based applications and platforms. However, most of these solutions are based on generic analysis of threats and security countermeasures. Consequently, such solutions do not address specific application security requirements and threats. To address this problem, we perform security analysis using an agent-mediated case study and propose generic countermeasures to security threats based on a concrete application. The study also proposes an approach for performing systematic security analysis for agent-mediated application based on our results and experiences. The paper also investigates whether security requirements and threats change when an application is implemented using different types of agent platforms. Section 2 presents the booktrading application that is used as a case study, stakeholders and assets are presented in section 3, security requirements and threat modeling are presented in sections 4 and 5 respectively.

2. THE BOOKTRADING APPLICATION

* Author's Address: Richard Ssekibuule and Jose Ghislain Quenum . Department of Computer Science Faculty of Computing and Information Technology, Makerere University, Kampala, Uganda. {rkayondo, joque}@cit.mak.ac.ug

"Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than IJCIR must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee."
© International Journal of Computing and ICT Research 2009.
International Journal of Computing and ICT Research, ISSN 1818-1139 (Print), ISSN 1996-1065 (Online), Special Issue Vol.3, No.1 pp. 67-76, October 2009.

The booktrading application comprises of booksellers and bookbuyer software agents that are ideally owned by separate individuals. The bookseller owners are responsible for maintaining bookstore operations which include stocking and setting prices for books. Bookbuyer software agents are responsible for buying books from bookseller agents on behalf of their owners. The implementation of the booktrading application was done in AgentScape[Overeinder and Brazier 2005] (version 0.9.1) and Java Agent Development Environment[Bellifemine et al. 2007] (JADE version 3.6) agent platforms. The purpose of implementing the application in two agent platforms was to investigate whether threats, security requirements and threat countermeasures for the application would change when implemented in either of the platforms. The bookseller interface is used by the bookseller owner to store information about books available for sale. In addition to the book title and year of book publication, the bookseller also records the first price of the book and lowest price at which he/she can sell a book. The first price is the best reasonable price that a seller finds competitive in the market. It is in the best interest of booksellers to keep information about the lowest price of the book secret from buyers and other sellers. Keeping this information secret prevents the bookseller from being exploited by buyers during negotiation. Additionally, the bookseller might be helped in keeping pricing information secret from other booksellers to avoid competition that may arise from other booksellers in the environment setting their prices based on what they know about the seller. The bookbuyer interface is used by the bookbuyer owner to send out requests to booksellers. Using the bookbuyer interface, the bookbuyer owner informs the agent of the best price at which they wish to buy the book and the maximum amount they are willing to pay.

The bookbuyer agent also implements a protocol for negotiation with the bookseller in case the desired book is found to be at a price higher than the bookbuyer's best price. The current implementation of the negotiation protocol always proposes a price that is half the sum of the best and highest price. Ideally this strategy should not be known to the bookseller, otherwise a bookseller agent can exploit this information to sell at a higher price to the bookbuyer than they should have done in case they didn't know the buyer's negotiation strategy. Figure 3 below presents an interaction protocol between the bookseller and bookbuyer for the booktrading application. The current implementation of the booktrading application does not support mobility of either the bookseller or the bookbuyer. The choice of having the bookseller stationary was a design decision made to support functionality for storing books in a MySQL[MySQL, AB] database, while the choice of having the bookbuyer stationary was due to limitations in inter-platform migration of JADE agents. The current version (3.6) of JADE does not have proper support for inter-platform migration. The mobility addon was developed for an older version of JADE and has not been since updated. Similar design decisions where taken with AgentScape, but the implementation limitations were different from those experienced with JADE. Nevertheless, security considerations for having the bookbuyer mobile were considered. The requirement of having the bookbuyer mobile could arise when it is deemed necessary to have the buyer migrate to the platform that has the desired book. Such a necessity would arise in case it is considered computationally expensive for the buyer to perform all tasks from the bookbuyer owner's platform.

3. STAKEHOLDERS AND ASSETS

This section presents stakeholders and assets in the booktrading application and agent platforms that were used in the implementation of the booktrading application. These stakeholders and assets are partly derived from the functional requirements of the booktrading application.

3.1 Booktrading Application Stakeholders

(i) ***Application creator:*** This is the individual or organization that developed both the agent bookseller and agent bookbuyer. (ii) ***Bookseller owner:*** The individual or organization that owns the bookseller agent. (iii) ***Bookbuyer owner:*** The individual or organization that owns the bookbuyer agent. (iv) ***Platform creator:*** The individual or organization that developed the agent middleware (here in referred to as the agent platform). (v) ***Platform owner:*** This is the stakeholder category that owns or administers the operating system (host) on which the agent platform is installed.

3.2 Booktrading Application Assets

Bookseller agent: This is the agent code that is responsible for handling tasks related to storing books in

the bookstore and interacting with bookbuyers.

Bookbuyer agent: The agent code that represents the human bookbuyer in booktrading tasks. The bookbuyer agents captures the title, the maximum age of the book, best and highest price that the agent owner would be willing to pay for the book.

Interaction protocol: The interaction protocol represents a set of rules through which messages exchanged between the agent buyer and agent seller are interpreted. In case the interaction protocol is is not followed, it is assumed that either party would not understand what they other is saying. The format of interaction messages were previous defined by Foundation of Intelligent Physical Agents (FIPA)[FIPA 2002]. The booktrading application extends the interaction protocol to include negotiation. The negotiation protocol is implemented by the bookseller and book-buyer agents to handle the logic through which they can agree on an alternative price that is different from the booksellers first price and the bookbuyers best price.

Agent platform: The agent platform provides the execution environment for both the bookseller and bookbuyer.

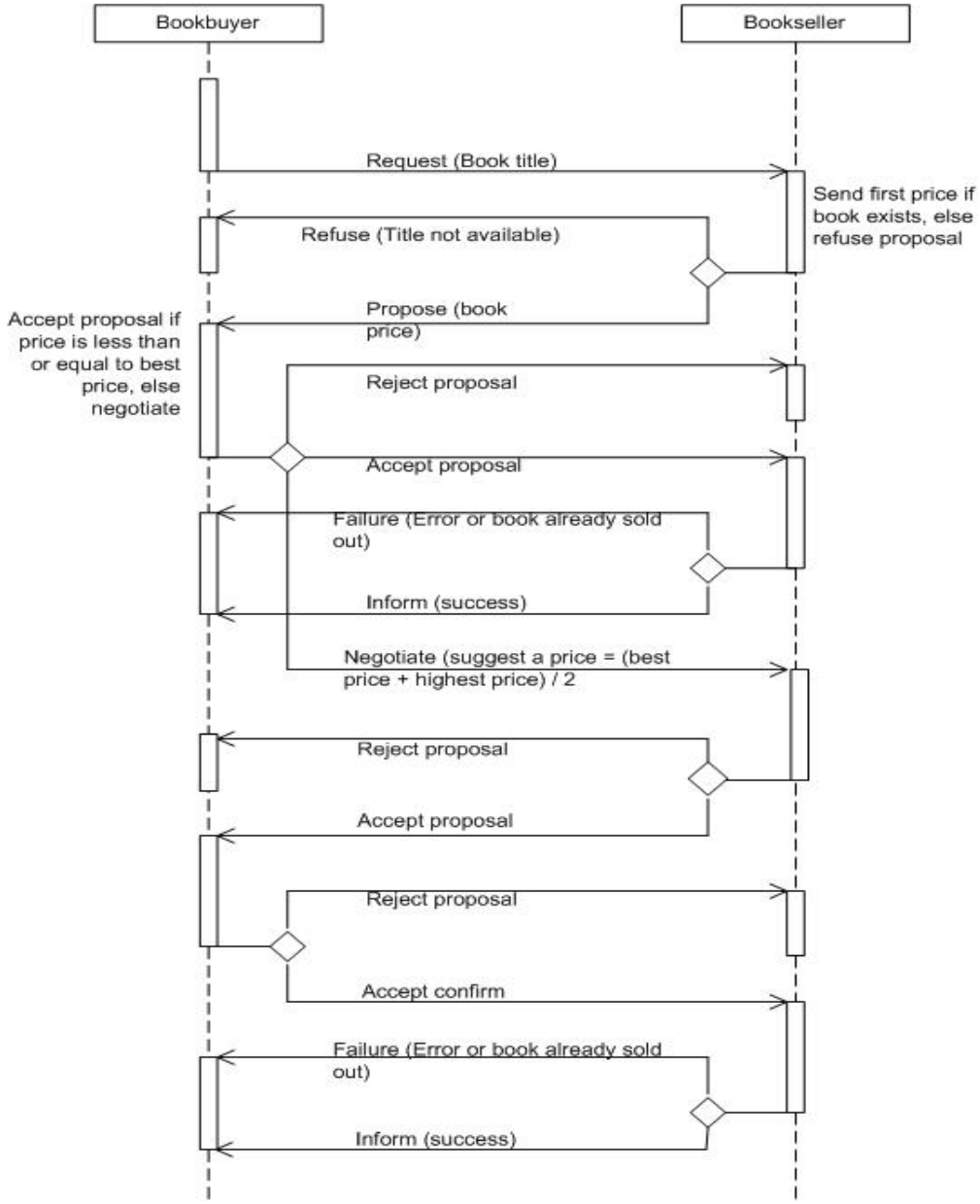

Fig. 1. *Interaction protocol for booktrading application*

3.3 Platform Assets

This section presents platform assets for AgentScape and JADE agent platforms. Separating the assets and possible attacks to the agent platforms created a basis for thinking about generic countermeasures for attacks on agent platforms.

3.3.1 *AgentScape Platform Assets.*

AgentScape system services: The component represents core kernel services for AgentScape. System services ensure that the right classes are used in the communicator. AgentScape system services

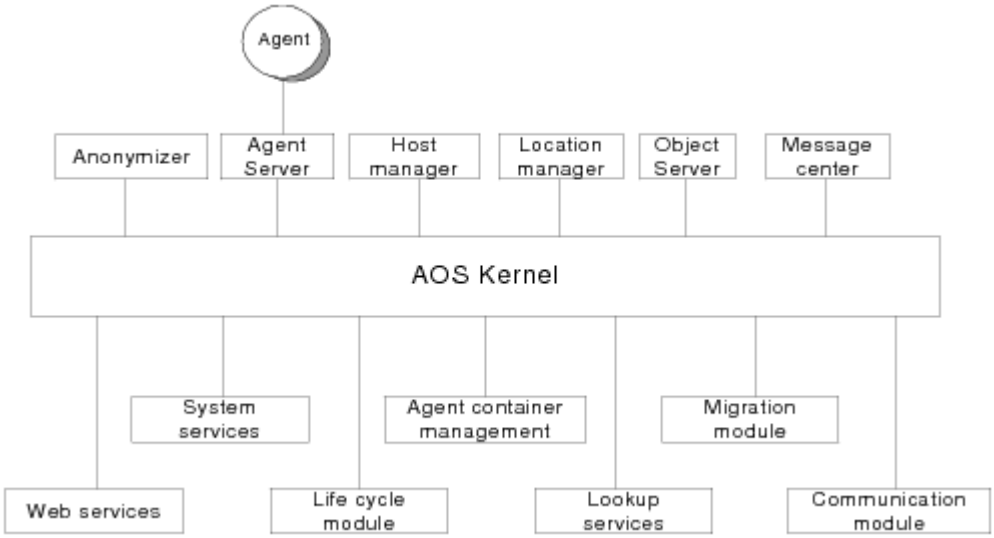

Fig. 2. *AgentScape Architecture*

provide lookup services through which agents and services are registered and discovered. They also provide a communication module through which messages are exchanged between agents and remote agent management platforms.

Agent server: Provides services for loading of agent code from the agent container, startup and termination of the agent. Facilitates agent access to kernel services, provides mechanism for making negotiation calls. The agent server also uses an agentwrapper to provide interaction between a running agent and agentscape middleware.

Host manager: AgentScape host manager provides agent container management for mediating access to agent stores. The hostmanager additionally facilitates data handling during agent migration. The host manager provides agent life cycle management services such as migration, suspending, stopping and running of an agent.

Location manager: Provides an agent management module that facilitates inserting new agents into a location and handling of migration requests from local agents and remote location managers.

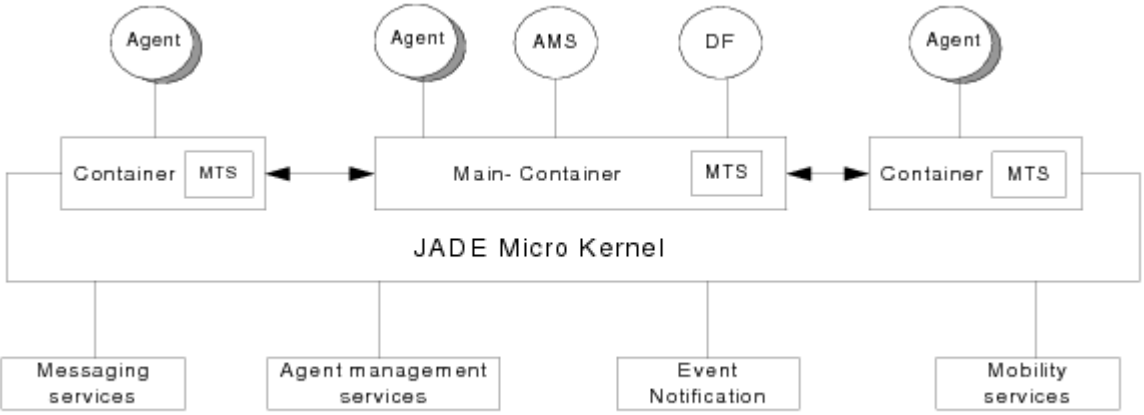

Fig. 3. *JADE Platform Architecture*

3.3.2 *Jade Platform Assets.*

JADE Core Base Service*:* Microkernel of the JADE system that provides a uniform mechanism for management and service discovery.

Agent Management Services: Agent management services define the agent, provide platform administration, status information for agents and a unique naming scheme.

Messaging Services: Microkernel service responsible for managing communication between agents and other entities in across agent platforms *Event Notification Services:* Represents events related to the agent life-cycle and configuration.

Agent Mobility Services: JADE microkernel provides an agent mobility service to facilitate migration within containers on the same platform and migration from one platform to another.

4. SECURITY REQUIREMENTS

This section covers security requirements for both the agent platform on which the agent-mediated application can be executed and security requirements for the booktrading application.

4.1 Application Security Requirements

We use the Confidentiality, Integrity, Authentication and Non-repudiation (CIAN) taxonomy to define security requirements for both the bookseller and bookbuyer agents. The Confidentiality, Integrity and Authentication (CIA) taxonomy has been reviewed by Howard et. al [Howard and Lipner 2006] in regard with risk analysis. This section presents a forth requirement of non-repudiation and brief definition of CIA components. ***Confidentiality****:* This is a security requirement that ensures information exchanged and stored in the system is accessed by only authorized users. ***Integrity****:* This requirement ensures that changes to information or application code are only done by authorized users. ***Availability****:* This is a security requirement that ensures that information and application resources can be accessed by all legitimate parties. The legitimate parties may include users and software processes. ***Non-repudiation****:* This security requirement ensures that all parties can be held accountable for their actions. It ensures that all actions taken cannot be denied at a later time.

4.2 Bookseller Security Requirements

This subsection presents security requirements for the bookseller agent application.

Confidentiality: The bookseller should not release buyers private information to the public and the information should be protected from attackers.

Integrity: Bookseller should prevent book reviews from being manipulated or changed by people who did not write them.

Availability: (i) The bookseller should not prevent some books from being available to some buyers for one reason or another. (ii) The bookseller should allow the buyer to choose a book based on attributes such as price, publication date, quality and relevance among others. (iii) The bookseller should not deny bookbuyers a chance to negotiate when requested. (iv) The bookseller should not disable possibilities of writing reviews and posting them from genuinely critical reviewers and buyers. (v) The bookseller should not prevent potential buyers from reading book reviews.

Non-repudiation: The bookseller should not be able to deny any information that is exchanged with the bookbuyer such as having received payment from bookbuyers or purchase orders.

4.3 Bookbuyer Security Requirements

This subsection presents security requirements for the bookbuyer agent application.

Confidentiality: (i) Should be able to migrate from one platform to another and perform computational tasks from local and remote hosts/platforms without leaving information traces for attackers. Such information could include platforms/hosts they have previously visited and budget information for their shopping. (ii) The bookbuyer should have control on the type of information that they have to submit to the operation environment. Withholding some information for privacy reasons should not be a cause for denying them access to requested services.
(iii) Bookbuyer should avoid forming a coalition in which it will be exploited. E.g exposing information it carries to malicious bookbuyers.

Integrity: (i) The bookbuyer should not be manipulated to buy a book that is not the best on offer. (ii) The information carried by the bookbuyer agent on behalf of the agent owner, should be protected from attackers that might want to change it.

Availability: (i) Bookbuyer should be able to find a book available for sale.
(ii) The bookbuyer agent should protect the information it carries and prevent it from being destroyed by attackers.
(iii) Bookbuyer should not be prevented from forming a coalition with other bookbuyers for mutual benefit.
Non-repudiation: The bookbuyer should not be able to deny that they ever ordered for a book.

4.4 Platform Security Requirements

The agent platform security requirements are derived from the CIAN acronym that was defined in subsection 4.1.
Confidentiality: In case of agent migration, the agent platform needs to protect the migration path of the agent. The migration path might contain information concerning platforms the agent has previously visited and intended destinations.
Integrity: (i) The agent platform is expected to maintain integrity of agents (agent code) and protect them from malicious attackers in the environment. It is important to note that malicious platforms might try to do otherwise. (ii) The agent platform should provide a mechanism for detecting compromised or malicious agents.
Availability: (i) The agent platform needs to ensure availability of messaging service for agent communication. Termination of the messaging service would prevent agents from communicating, while a comprised messaging service could yield unexpected and undesired results for the intentions of the communication. For example, attacker could delay messages for the intended destination whose requests might have been time bound. (ii) The agent platform should have policies for regulating access to system resources to prevent starvation of some agents by others that may intentionally or otherwise over consume system resources.
Non-repudiation: The agent platform needs to provide a mechanism through which agents can be accountable for the actions they perform when visiting platforms.

5. THREAT MODELING

We use the STRIDE[Howard and Lipner 2006] taxonomy to identify possible threats faced by the booktrading application and the agent platform on which the application is executed.
Spoofing Identity: This is a form of attack in which someone or an entity pretends to be someone else of another entity. For example agent X pretending to be agent Y.
Tampering : Tampering attack refers to unauthorized changing of software code or information.
Repudiation: This refers to a circumstance in which a software process or an individual deny responsibility for their actions.
Information disclosure: This refers to unauthorized access to information.
Denial of Service: This is a form of attack that denies legitimate access to resources such as information, storage space, processor and communication channels.
Elevation of privileges*:* This refers to a form of attack in which an entity with lower privileges gains unauthorized higher privileges.

A more detailed explanation of the STRIDE components was presented by Howard et. al [Howard and Lipner 2006].

5.1 Application Threat Model

The book trading application is an agent mediated application in which one agent acts as a seller and another agent as a buyer. The bookseller agent provides a variety of books for sale in a manner similar to bookstores such as amazon, but in this setting the bookseller expects the buyers to be software agents. The bookbuyer agents are supposed to search in the books catalogue and compare prices and other attributes such as publication date, relevance and book ratings on behalf of their owners. The bookbuyer agent is expected to perform these tasks for a range of booksellers that have books available for sale. This section covers possible goals of the attacker against the bookseller and bookbuyer agents and suggests possible countermeasures to the identified threats.

5.1.1 *Possible Goals of Attackers against Bookseller*

i. ***Spoofing Identity:*** An attacker could spoof the identity of a bookseller and requests payment from buyers for books that will not be delivered.
ii. **Tampering:** (i) The attacker may want to change information in the booksellers catalogue so that book attributes such as price, publication date, quality and relevance are not correct. These changes could lead bookbuyers into choosing items that they were not supposed to buy. (ii) An attacker could change book reviews and rating so that consumers will not buy books from that particular bookseller.
iii. ***Repudiation:*** A malicious bookbuyer could deny having requested or received a book from the

bookseller.

iv. ***Information Disclosure:*** (i) The attacker could compromise the negotiation logic implemented in the bookseller. E.g. if an attacker knows the price that a consumer wishes to pay for a product, the attacker could lower their prices to outcompete other sellers or simply to distract the buyer from making a genuine negotiation or purchase. (ii) An attacker may wish to access and log information concerning bookbuyers. The intention of this attack would be to compromise buyers privacy.

v. ***Denial of Service:*** (i) An attacker could block the messaging service between the bookbuyer and the bookseller. (ii) The attacker could try to remove items from the booksellers catalogue of books so that books requested by the book-buyer are not available. (iii) An attacker could block buyers from writing and sending reviews on books. Such an attack could negatively affect the bookseller if buyers wish to know whether the books on sale are good and price worthy.

5.1.2 *Possible Goals of Attackers against Bookbuyer:*

i. ***Spoofing Identity:*** An attacker could spoof the identity of a bookbuyer agent and purchases books that could be reputation damaging to the agent owner. This attack could be more severe in a general purpose e-commerce application where many types of products could be bought.

ii. ***Tampering:*** (i) The attacker could alter message responses from the bookseller to indicate to the bookbuyer that the requested book is not available, even when it is actually available. (ii) An attacker could change book reviews and rating so that consumers are lured into buying books that are not price worthy. (iii) An attacker could change information requests from the bookbuyer to indicate different requests to the bookseller from the ones submitted by the bookbuyer. Such attacks could lead the bookbuyer into getting invoices for books they did not order. Additionally, buyers could get false responses such as requested books not being available, even in circumstances where the books are available.

iii. ***Repudiation:*** A malicious bookseller could deny having received payment for a book. In such a case, the bookbuyer would end up losing money.

iv. ***Information Disclosure:*** (i) An attacker may have interest in accessing private information that is carried by a bookbuyer agent. (ii) An attacker could lure a bookbuyer into forming a coalition in which it would be exploited. E.g leaking information it carries to malicious bookbuyers.

v. ***Denial of Service:*** (i) An attacker could lure a bookbuyer into forming a coalition in which it would be exploited. (ii) The attacker could block the messaging service between the bookbuyer and the bookseller. (iii) The bookbuyer could be denied a chance of forming a coalition with other buyers by withholding coalition formation information. This attack could also affect the bookseller by not making a needed sale. (iv) An attacker could prevent potential buyers from reading book reviews. (v) An attacker could create a malicious bookstore to prevent a bookbuyer from finding a genuine book to buy.

5.1.3 *Application Specific Countermeasures.*

This section presents countermeasures for the security challenges that are likely to be faced by the agent bookseller and bookbuyer. The countermeasures are meant to prevent attackers' goals that were identified in subsection 5.1.1. The countermeasures are combined for attacks on the bookseller and bookbuyer agents, because these attacks are similar in nature. We also assume that safe coding procedures were followed for both the application and agent platform. When software security flaws such as buffer overflows exist in software, then authentication and authorization schemes can be subverted.

i. ***Spoofing Identity:*** Spoofing of an agent's identity can be prevented by providing an identity management system[de Groot and Brazier 2006] through which agents are assigned names (or identities) that are difficult to be changed by the agent or an attacker. Authentication and authorization systems such as kerberos[ste 1988] or message authentication codes [Kaliski and Robshaw 1995] can be used to authenticate agents' identity or their owners. In this setup an agent would be required to submit a message to the service provider indicating their identity and a small message encrypted by their private key. The service provider would then retrieve a public key (from the key-management system) that is needed to decrypt the short message. It is assumed with public-key cryptography that the private key of the agent is key secret.

ii. ***Tampering:*** Two things need to be protected against tampering. That is the information or data carried by agents and the agent code. The countermeasures available against these attacks fall into categories

of prevention and detection. Message authentication codes (MAC)[Kaliski and Robshaw 1995] and digital signatures on the agent code and data are used to detect any form of tampering on the agent code and data.

iii. ***Repudiation:*** Public-key digital signatures can be used to prevent repudiation by either a malicious bookseller or bookbuyer. In order to prevent repudiation, digital signatures would be required on messages from either the bookbuyer or bookbuyer. When a given agent (A) encrypts messages using their private key, those messages can only be decrypted by a corresponding public key that certainly belongs to the sending agent. The main challenge to this kind of solution is that security depends on the secrecy of the secret key.

iv. ***Information Disclosure:*** This countermeasure should protect information carried by the agent. Such information includes target book titles, best and highest price that the bookbuyer is willing to pay for the book. Confidentiality of this information can be provided by encrypting the information carried by agents.

v. ***Denial of Service:*** The agent platform and agent execution environment needs to provide strong authentication [ste 1988] and authorization for processes that access system resources. Authentication and authorization are useful in preventing non-authorized users and processes from accessing privileged resources that could be critical for correct functionality of the agent application. Apart from preventing users and processes from accessing privileged resources and services, authentication is useful for detecting users and processes that may breach the imposed restriction. In detecting which users or processes performed certain tasks, accountability can be achieved for all activities undertaken in the system. The concept of a trusted third party can be used to determine ratings of a bookseller before it can be considered by the bookbuyer for purchase of a book. A trusted third party would help in preventing malicious booksellers from participating in booktrading transactions.

5.2 Platform Threat Model

The platform threat model is based on the Java Agent Development Environment (JADE)[Bellifemine et al. 1999] and AgentScape[Overeinder and Brazier 2005] platforms whose assets were presented in subsections 3.3.2 and 3.3.1 respectively.

5.2.1 *Possible Goals of Attackers against Agent Platform*

i. ***Spoofing Identity:*** An attacker could launch counterfeit agents using the agent platform to participate in a transaction they are not supposed to be involved. For example, an agent Z (representing an attacker) could spoof the identity of agent X in order to perform actions privileged to X.

ii. ***Tampering:*** (i) An attacker could be interested in altering agent code through the agent platform so that the agent does not do what it is supposed to do. (ii) The attacker could use a weakness in the agent platform to reach and subvert the communication channel for agents. (iii) The attacker could use the platform to migrate an agent from a trusted platform to a compromised one.

iii. ***Information Disclosure:*** In circumstances where agent platforms keep a non-repudiable log of agent events, an attacker could be interested in knowing about action of a particular agent. An attacker accessing this information could violate agent's confidentiality requirements. Additionally, the attacker could use this information to launch other forms of attacks against the agent.

iv. ***Denial of Service:*** An attacker could be interested in subverting the Agent Management System (AMS), Directory Facilitator (DF) and Management services.

v. ***Elevation of Privileges:*** Exploiting a flaw in the agent platform to grant higher privileges to malicious agents. Such malicious agents could be interested in using free resources on platform hosts, or even overconsuming system resources in order to deny services to legitimate users

5.2.2 *Platform Specific Countermeasures.*

This section presents security techniques that are needed by the platform to achieve the security requirements indicated in section 4.4 and to prevent attackers from achieving their objectives stated out in subsection 5.2.1.

i. ***Spoofing Identity:*** Provide identity management: Only authenticated and registered agent owners are allowed to launch booksellers into the environment to prevent scenarios of malicious booksellers. A trusted third party can be used to verify and certify credentials agents.

ii. ***Tampering:*** As indicated in subsection 5.2.1 the attacker may want to change the agent code or migrate an agent to a malicious platform. The action of changing or tampering with agent code can be

prevented by code signing [Jansen 2000] with a digital signature. Agent migration to malicious platforms can prevented by use of security policies to authorize sensitive actions to be executed by only trusted parties.

iii. ***Repudiation:*** Combined with identity management, a non repudiable log kept by the agent platforms would be useful in tracking actions that were performed by various agents.

iv. ***Information Disclosure:*** Confidentiality of the agent logs resident on the agent platform can be achieve through encryption. However, encryption has to be applied with consideration for low resource platforms like mobile devices.

v. ***Denial of Service:*** Use of security policies for authentication and authorization of users and processes to key assets for the agent platform would prevent denial of service attacks.

vi. Elevation of Privileges: This attack can be stopped by preventing software security flaws such as buffer overflows and SQL injection. Language based security mechanisms such as static source code analysis to enforce safety properties of the programming language, sandboxing and proof carrying code can be used.

6. CONCLUSIONS AND REMARKS

In this paper, we present a systematic approach for performing a security analysis of an agent mediated application. We have presented the Confidentiality, Integrity, Availability and Non-repudiation (CIAN) framework through which security requirements for an application can be derived and combined it with STRIDE[Howard and Lipner 2006] to obtain possible attacker goals. The proposed countermeasures (presented in subsections 5.2.2 and 5.1.3) are clearly generic for any agent application and indicate generic assets that need to be protected in an agent mediated application.

Our results indicate that security requirements for the agent application did not change significantly when implemented with either AgentScape or JADE. In reference to subsection 3.3, the assets of the agent platforms fall in the categories of (i)*Agent Management Services*, (ii) *Directory Facilitator*, (iii) *Messaging Services*, (iv) *Mobility Services* and (v) *Event Notification Services*. This implies that these assets need to be protected irrespective of the agent platform against all possible forms of attacks for the agent environment to be considered secure. Furthermore, for any application to be considered secure, its security requirements have to be catered for in the implementation and assets protected.

6.1 Implementation Experiences and Issues

This section presents some key issues and experiences that were encountered during the implementation phase of the booktrading application on JADE and AgentScape platform.

The implementation challenges were different for the platforms (AgentScape and JADE) that were used. Intra-platform migration challenges were faced with JADE (version 3.5 and 3.6) mainly because the mobility add-on was implemented for lower version of JADE. In AgentScape, implementation challenges were faced with service discovery facility due to the nature of requirements that were imposed by the application. It is worthy noting that these middleware platforms (mostly especially

· AgentScape) are still under heavy development and most problems are being solved as they are reported by users.

7. FUTURE WORK

The information disclosure countermeasure needs to protect the bookbuyer against traffic analysis by an intelligent bookseller. In this example booktrading application, the bookbuyer negotiates by sending a second price that is half the sum of the best and highest price. Using traffic analysis, an intelligent bookseller could be able to generate these two (best and highest) prices of the bookbuyer. The bookseller knowing these prices puts the bookbuyer in a poor negotiation position. Furthermore, the countermeasure should protect the negotiation protocol from revealing negotiation strategy. Additionally, some malicious participants could start the negotiation protocol and then terminate it with the intentions of stealing information concerning pricing and introducing annoyance attacks.

8. REFERENCES

CISCO SYSTEMS, INC,. KERBEROS: An Authentication Service for Open Network Systems. Proc. *Winter USENIX Conference*, 1988.

BELLIFEMINE, F., CAIRE, G., AND GREENWOOD, D. 2007. *Developing Multi-agent Systems with JADE*. Springer.

BELLIFEMINE, F., POGGI, A., AND RIMASSA, G. 1999. JADE–A FIPA-compliant agent framework. *Proceedings of PAAM* 99, 97–108.

DE GROOT, D. AND BRAZIER, F. 2006. Identity Management in Agent Systems. *Proceedings of the First International Workshop on Privacy and Security in Agent-based Collaborative Environments (PSACE)*, 23–34.

FARMER, W., GUTTMAN, J., AND SWARUP, V. 1996. Security for Mobile Agents: Issues and Requirements. *Proceedings of the 19th National Information Systems Security Conference 2*, 591–597.

FIPA, F. 2002. Contract Net Interaction Protocol Specification.

HOWARD, M. AND LIPNER, S. 2006. The Security Development Lifecycle. *Microsoft Press Redmond, WA, USA, Chapter Risk Analysis*, 114–115.

JANSEN, W. 2000. Countermeasures for mobile agent security. Computer Communications 23, 17, 1667–1676.

KALISKI, B. AND ROBSHAW, M. 1995. Message authentication with MD5. *CryptoBytes (RSA Labs Technical Newsletter) 1, 1*.

LANGE, D., OSHIMA, M., KARGOTH, G., AND KOSAKA, K. 1997. Aglets: Programming Mobile Agents in Java. *Lecture Notes in Computer Science*, 253–266.

LI, X., ZHANG, A., SUN, J., AND YIN, Z. 2004. The Research of Mobile Agent Security. *Lecture Notes in Computer Science,* 187–190.

MAHMOUD, Q. AND YU, L. 2004. Havana: a mobile agent platform for seamless integration with the existing Web infrastructure. *Electrical and Computer Engineering, 2004. Canadian Conference*.

MySQL, AB. MySQL Database Server. *Internet WWW page, at URL: http://www. mysql. com (last accessed 5/21/2008)*.

OVEREINDER, B. AND BRAZIER, F. 2005. Scalable Middleware Environment for Agent-Based Internet Applications. *Lecture Notes in Computer Science 3732*, 675.

SURI, N., BRADSHAW, J., BREEDY, M., GROTH, P., HILL, G., JEFFERS, R., MITROVICH, T., POULIOT, B., AND SMITH, D. 2000. NOMADS: toward a strong and safe mobile agent system. *Proceedings of the fourth international conference on Autonomous agents*, 163–164.