

بسم الله الرحمن الرحيم





شبكة المعلومات الجامعية

التوثيق الالكتروني والميكروفيلم



جامعة عين شمس

التوثيق الإلكتروني والميكرو فيلم

قسم

نقسم بالله العظيم أن المادة التي تم توثيقها وتسجيلها
علي هذه الأقراص المدمجة قد أعدت دون أية تغيرات



يجب أن

تحفظ هذه الأقراص المدمجة بعيدا عن الغبار





بعض الوثائق الأصلية تالفة





بالرسالة صفحات لم ترد بالأصل



Ain Shams University

Faculty of Computer & Information Sciences

Computer Science Department



A Multi-Agent Based Framework for Managing Security Policies

Thesis submitted as a partial fulfillment of the requirements for the degree of Master of Science in Computer and Information Science.

By

Mohamed Al-Morsy Abd Al-Razek

B.Sc. in Computer Science.

Demonstrator in Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

Under Supervision of

Prof. Dr. Taha El-Aref

Professor of Computer Science
Faculty of Computer and Information Sciences,
Ain Shams University.

Dr. Hossam M. Faheem

Associate Professor of Computer Systems
Faculty of Computer and Information Sciences,
Ain Shams University.

Cairo 2009

ABSTRACT

The revolution of new technologies and internet has motivated organizations expanding their networks to provide more services to customers and employees. Although the benefits gained, organizational assets become more vulnerable. Security become an important concern in today networked environments. As the underlying environments are complicated and dynamic, Security solutions should be flexible, scalable, and can integrate to achieve organizational security needs.

Policy-based management, as one of the latest approaches for network, distributed systems, and security management, is a promising solution for the security management problem. It introduces security policies as abstraction of the business needs away from security solutions that implement the business needs. Security policy specification, security policy configuration, security policies conflicts detection, policy enforcement, and deployment are critical issues that need to be clearly defined in policy-based management. With the increase of the organization scale, security officers exert more efforts to cover the organizational security needs and impact of changes in business objectives, hence come the need to automated security policy management system.

The Proposed framework is a new idea in automating the security policy management process. The framework is based on mobile agents. It provides an XML/Ontology-based standard security policy language (SSPL), architecture for deploying policies based on SSPL, security solutions integration interface, and a set of helping tools for specifying

and managing policies. The SSPL is aimed to be a standard, formal and readable security policy specification language. It helps in eliminating problems of writing security policies in natural languages or the developed security policy specification languages. The XML technology gives SSPL the formalization (in syntax) while ontology adds the semantics and analyzing capabilities to SSPL.

The proposed architecture simplifies/automates the mapping phase among security policies and security solutions by introducing abstraction tier between policy development tier and security solutions tier and using mobile agents for communication among them.

The proposed security solutions integration interface is a software driver that defines and manages how security solutions integrate with the proposed framework. It acts as a communication channel between the framework and the security solutions deployed.

The framework eliminates ambiguities of using natural languages in specifying security policies, eliminates the needs to learn many of the existing security policy specification languages to be able to specify the whole organizational security policies, eliminates difficulties associated with configuring heterogeneous security solutions, and automates the impact of the changes in security policies configurations. The framework is designed to be as flexible and extensible to cover new policies, new policy domains, and capable to integrate with new security solutions seamlessly.

ACKNOWLEDGMENT

I would like to express my gratitude to my supervisor Dr. Hossam M. Faheem whose expertise, understanding, and patience, added considerably to my experience and finishing this work. Also, my most profound gratitude goes to my family who has been the timeless source of inspiration for me; especially my great wife, to whom this thesis is dedicated.

TABLE OF CONTENTS

| | |
|--|----|
| List of tables | i |
| List of figures | ii |
| List of symbols | iv |
| Chapter 1: Introduction | 1 |
| 1.1 Security Policy Overview | 3 |
| 1.1.1 Security policy Statement requirements | 4 |
| 1.1.2 Security Policy and the Abstractions Levels | 5 |
| 1.1.3 Security policy classification | 6 |
| 1.1.4 Security policy related challenges | 7 |
| 1.1.5 Security policy management lifecycle | 9 |
| 1.1.6 Developing organizational security policies | 10 |
| 1.2 Problem Definition | 14 |
| 1.3 Requirements | 16 |
| 1.3.1 Functional Requirements | 17 |
| 1.3.1 Non-Functional Requirements | 17 |
| 1.4 Motivation | 17 |
| 1.5 Objectives | 18 |
| 1.6 Thesis Outline | 19 |
| Chapter Summery | 20 |
| Chapter 2: Security Policy Management Related Work | 21 |
| 2.1 Security policy specification approaches | 21 |
| 2.1.1 “Non-Language based” policy specification approaches | 22 |

| | |
|---|----|
| 2.1.2 “Language based” security policy specification approaches | 24 |
| 2.2 Security policy management | 40 |
| 2.2.1 Policy management architectures | 40 |
| Chapter Summery | 43 |
| Chapter 3: Multi-Agents Systems | 44 |
| 3.1 Mobile agents | 44 |
| 3.2 Why Mobile agents in security solutions? | 46 |
| 3.3 The agent-oriented technology | 47 |
| 3.3.1 Agent-oriented software engineering | 48 |
| 3.4 Mobile agents standards | 51 |
| 3.4.1 MASIF standards for agent development | 52 |
| 3.4.2 FIPA standard for agent development | 53 |
| 3.5 Agent-based programming languages and platforms | 58 |
| 3.6 EtherYatri.NET framework | 59 |
| 3.6.1 Features of EtherYatri.NET | 59 |
| 3.6.2 EtherYatri.NET class library | 60 |
| 3.6.3 EtherYatri.NET and FIPA | 61 |
| Chapter Summery | 63 |
| Chapter 4: The Proposed Multiagent-Based Framework | 64 |
| 4.1 Technologies overview | 64 |
| 4.2 The Proposed framework architecture | 67 |
| 4.2.1 The core security policy tier | 68 |

| | |
|---|-----|
| 4.2.2 Communication between core policy tier and core abstraction tiers | 75 |
| 4.2.3 The core abstraction (server) tier | 78 |
| 4.2.4 Core security solutions tier | 82 |
| 4.3 The framework agents | 84 |
| 4.4 The framework deployment model | 86 |
| Chapter Summery | 88 |
| Chapter 5: The Proposed Standard Security Policy Language | 89 |
| 5.1 Desirable characteristics of security policy language | 90 |
| 5.2 The SSPL concepts overview | 91 |
| 5.2.1 Why XML? | 92 |
| 5.2.2 Why ontology? | 93 |
| 5.3 The SSPL specifications | 94 |
| 5.3.1 The SSPL syntax | 95 |
| 5.3.2 The SSPL semantics | 97 |
| 5.4 The SSPL engine | 98 |
| 5.5 Examples on SSPL | 101 |
| 5.6 The SSPL contributions and implementation | 85 |
| Chapter Summery | 107 |
| Chapter 6: Security policy management Case Studies | 108 |
| 6.1 Intrusion detection policy case study | 108 |
| 6.1.1 The working environment | 108 |
| 6.1.2 Developing the Intrusion detection policy | 109 |
| 6.1.3 Deploying the ESSIDS application | 114 |

| | |
|---|-----|
| 6.1.4 Reconfiguring the IDP | 115 |
| 6.1.5 Conclusion | 116 |
| 6.2 The mail server-DMZ case study | 117 |
| 6.2.1 The working environment | 117 |
| 6.2.2 Developing the Mail server policy | 117 |
| 6.2.3 Conclusion | 120 |
| Chapter 7: Conclusion and Future work | 121 |
| 7.1 Conclusion | 121 |
| 7.2 Future Work | 125 |
| References | 126 |

LIST OF TABLES

| Number | Page |
|--|------|
| Table 3.1: The EtherYatri.NET class library classes | 60 |
| Table 4.1: The framework agents' description | 85 |
| Table 5.1: Password policy statement | 101 |
| Table 5.2: Remote access policy statement | 103 |
| Table 5.3: Comparison between security policy languages | 106 |
| Table 6.1: The specifications of the working environment | 109 |

LIST OF FIGURES

| Number | Page |
|--|------|
| Figure 1.1: A sample of security policy. | 4 |
| Figure 1.2: Security Policies Management Process | 10 |
| Figure 1.3: Steps to develop organizational security policies. | 13 |
| Figure 2.1: Policy research areas | 21 |
| Figure 2.2: CIM representations for a policy rule | 24 |
| Figure 2.3: The IETF policy enforcement architecture | 40 |
| Figure 3.1: The MaSE Methodology | 49 |
| Figure 3.2: Shows FIPA Main Architecture | 53 |
| Figure 3.3: The AMS Reference Model | 54 |
| Figure 3.4: The EtherYatri.NET Architecture | 59 |
| Figure 4.1: The proposed framework architecture | 68 |
| Figure 4.2: The core security policy tier structure | 69 |
| Figure 4.3: The core security policy layers | 69 |
| Figure 4.4: The Policy-Pad | 71 |
| Figure 4.5: The Security policy Definition Screen | 71 |
| Figure 4.6: The dynamic GUI engine | 72 |
| Figure 4.7: Sample of the Dynamic GUI Engine | 73 |
| Figure 4.8: Sample of the Dynamic GUI Engine output | 75 |
| Figure 4.9: The Security Policy Agent trips | 78 |