JKAU: Comp. IT., **Vol. 1** No. 2, pp: 25-44 (2012 A.D. / 1434 A.H.)

DOI: 7193 / Comp. 1-2.2

Network Security Management Using Ontology-Based Mobile Agents

Abdullah Marich Ali, Mohamed Ashraf Madkour and **Omar Abdullah Batarfi**

Lecturer, CS Dept., Professor, IT Dept. and Assistant Prof., IT Dept.
FCIT, King Abdulaziz University, Jeddah, KSA

A_manbari@yahoo.com mamadkour@kau.edu.sa obatarfi@kau.edu.sa

Abstract. Automatic means to manage the security in moderate and large networks are of extreme importance to avoid error-prone manual techniques. This paper paves the way to develop an automatic network security management (NSM) system that is both flexible in deciding the system's objectives and efficient in using the valuable network bandwidth with a relatively low transmission overhead. The required flexibility and efficiency are obtained using mobile agents (MA) to collect the required security information from various network components and devices, and using ontology to specify the required security policies in such a way understandable by the MA's software.

A simplified NSM prototype is developed, implemented, and tested over a typical local area network to investigate the validity of the suggested ideas. The MA travels through the network and collects the necessary information using an ontology-based security policy. Next, it may either return back to the network administrator to let him decide and perform the suitable actions, or alternatively the MA may take the appropriate decisions.

This prototype is tested to examine its functionality and performance using a simple local area network consisting of three computers with different configurations. The developed MA was able to understand the ontology and move around the network. It has properly detected the components that are wrongly configured. It should be made clear that the design is scalable and can be directly applied to more computers in a local area network or even in a wide area network.

Keywords: Network security management, mobile agent, ontology, security policy.

1. Introduction

As the Internet evolves, more people and computational systems are connected and more information needs to be protected. In this context, the concerns about information security have increased inside organizations, mainly because information is their most valuable asset[1]. Typically a large volume of security data is produced by network security components such as firewalls (FW), intrusion detection systems (IDS), system logs (SL), antivirus software (AV), vulnerability scanners (VS), and *etc*. Tracing the attacks in all these overwhelming data is impossible; Also manipulation and management of such security information are even harder. Evaluating the vulnerability to attacks becomes increasingly more important to automate the network security management process[2]. Therefore the rapid growth in the size and complexity of organizational networks makes the current way of manual management infeasible.

Recent years have seen many tools developed to automate this process. There are also tools that scan networks and discover possible attack scenarios involving complex combinations of multiple vulnerabilities[3]. To automate the security administration, we need a security management system that deals with controlling access to resources and even alerting the proper authorities when certain resources are accessed. Its aims are to ensure the protection of resources by preventing unauthorized access to them and monitoring exceptional states such as unauthorized access attempts[4]. Also we need a network security management system to maintain the integrity, confidentially and availability of systems and services.

Securing a network involves protecting it against all possible attacks. But, in practice, it is not possible to have a completely secure network. So, applying security management is a two-fold activity: 1- The security architecture is to be deployed to protect networks by detecting attack; and 2- when attacks are detected the security architecture deals with these attacks in real time by taking security measures[5]. An efficient management system is needed to make use of different security mechanisms in large networks. All mechanisms have to be configured consistently according to a security policy. To reduce complexity, the administrator should not have to cope with details not important for him where the network is divided into several administrative domains which are managed rather independently from each other. Each domain maintains its own network access control policy (NAP). The enterprise-wide policy is a combination of all NAPs. It is enforced by different security mechanisms of which configuration can be derived from the global access policy automatically[4].

Nowadays, the above-mentioned security components (FW, IDS, SL, AV, VS, and *etc*) are popularly used to protect user's networks. These security components shield user's computer systems in different ways. However, there are many problems in the current use of these security components. First, each component only provides protection in one field; they lack communication and cooperation to each other. Second, the log styles of each component are always different, and it is difficult for the administrators to analyze and identify real risk from such huge amounts of various alert data. Last but not least, the dissimilar interface in operation and management of each device makes it harder for the managers to manipulate the security data. A promising solution is given in[6] which uses multilevel correlation in a Unified Platform of Network Security Management (UPNSM) which tries to solve all the problems mentioned above by collecting and intelligently parsing data from a vast array of event sources (such as IDSs, FWs, AV and VSs), and delivering a single view into the information that can be used by corporate stockholders across user's organization.

In view of the above-mentioned considerations, the present work investigates a simplified method to manage the network security in an administrative domain by collecting the security data from all relevant network security components in that domain using mobile agents. The security policies that describe what must be done is specified by ontology in a suitable ontology language readable and understandable by the MA software.

The rest of this paper is structured as follows. Section two reviews the existing techniques for NSM systems. The use of mobile agents for network security management is illustrated in Section three. Section four shows the development and implementation of the NSM prototype based on mobile agent technology with the security policies represented by ontology. Test results of the implemented prototype are given and discussed in Section five. The last section provides concluding remarks.

2. Network Security Management

NSM is the process of monitoring and testing the configuration of network components automatically verifying whether it obeys the security policies determined by the administrators or not. It helps the administrator to make a suitable decision depending on the current status of the network security by reconfiguring the network security components automatically or sending a message to a specified user to reconfigure the security depending on the policies. The major functions of network security management can be identified as follows[7]:

a) Collecting, storing and analyzing the appropriate security management information in order to detect intrusions at the network and host level.

b) Generating various levels of security alarms and routing them to the appropriate locations.

c) Managing and controlling the mechanisms for the collection, storage and examination of audit records and security logs.

d) Coordinating and controlling the access policies to network resources.

e) Managing and controlling the authorization, authentication and encryption processes.

f) Controlling the security of the network management procedures by monitoring the authentication, encryption and access control mechanisms of the network management protocol.

2.1 Essential Security Information

The Security Information defines largely four categories, which are system information, user information, group information, and file information. Collecting these types of information is an essential and basic functionality in any effective network security management system. System Information is the information that diagnoses its own system monitored by a manager. Typically it includes information about hardware, operating systems, programs that can be used as shells, and files to influence all users to use the system. User information and Group information have the basic information that helps a manager to find the weaknesses of user account and group accounts on the system, Furthermore; it includes the information about the related important files.

File information has the information that enables a manager to detect the abnormal contents and the variation of the important system files or the directory defined in advance[8].

2.2 Policy-Based Management

Quite often attacks are enabled by miss-configurations of networked devices generated by human errors. Policy-based network management has been proposed to cope with this problem where goals are expressed as high-level rules that are then translated into low-level configurations for network devices[9].

An example of a network policy could be: "If the network resources are low at the end of the quarter, then restrict the WWW traffic." The scope of this policy is the network and more precisely, the WWW traffic. The mechanism is bandwidth allocation, and the action is to limit the network resources used by the WWW traffic during certain periods. Policies are used for automated system management and controlling the behavior of complex systems. The use of policies allows administrators to modify system behavior without changing source codes or requiring the consent or cooperation of the components being governed[10]. By changing policies, a system can be continuously adjusted to accommodate variations in externally imposed constraints and environmental conditions. The adoption of a policy-based approach for controlling a system requires an appropriate policy representation and the design and development of a policy management framework[11, 12]. Policy-based management systems are best for large networks where large numbers of devices are easier to manage from a central location. The Internet Engineering Task Force Policy Framework (POLICY) Working Group has developed a policy management architecture that is considered the best approach for policy management on the Internet.

Several policy languages, such as Ponder, have been designed for policy-based management. However, the use of different policy languages may lead to some difficulties for implementing the security management system.

2.3 Ontology-Based Management

While the process of translating high-level rules policies into low-level configurations for network devices is clear, there is a lack of tools supporting this strategy. The use of an ontology-based policy representation approach mimics the behavior of expert administrators, without their mistakes. Normally administrators are given the high-level security goals and then, through their knowledge of network topology and security they adopt the best practice and derive the device configurations[9]. Ontology can be used as an alternative to management information because it has formal definition and can enhance semantic expressiveness[13].

Ontology is a data model that represents a domain and is used to reason about the objects in that domain and relations between them. Ontology provides a vocabulary for representing knowledge about a domain and describing specific situations in that domain[14]. In practical terms, ontology is a hierarchy of concepts with attributes and relations that define a consensus terminology in semantic networks of interrelated information units. Ontology is the key techniques used to describe the semantics of information exchange. It provides a shared and common understanding of a domain that can be communicated across people and application systems, and thus facilitate knowledge sharing and reuse.

The goal of ontology-based management is to improve the manageability of network resources through the application of formal ontology. Network administrators need more intelligent management systems that hide the underlying complexity of the network, allowing them to manage the infrastructure at an abstract level, focusing on what the expected behavior should be, instead of on how to specifically achieve it[15]. The advantages of the ontology-based management are: First, it reduces the impact of administrators' mistakes in the configuration workflow. Second, it strongly decreases the effort and the time to have a set of correct configurations (from days to minutes). Finally, it quickly adapts to changes in the methodologies.

Tsoumas[16] presents a security management framework of an arbitrary information system which builds upon security exploiting security knowledge from diverse sources. This framework links the high-level policy statements and deployable security controls. In[17] Blanco et al provide a systematic review and comparison of security ontologies to identify, extract and analyze the main proposals for security ontologies. They observed that the greatest part of the selected works is still at the early stages of development, and concluded that complete security ontology is still needed. In particular the reviewed works were not exhaustive because the ontologies do not define all possibilities of the security domain. Moreover they used few attributes to define concepts and inappropriate natural language expressions to define the attributes.

3. NSM Techniques

Many techniques are used in network security management. The following reviews the most common techniques.

3.1 Client-Server NSM

This technique has been studied by[8]. It uses three configuration elements: Security Agent (SA), Security Manager System (SMS), and Security Management Visualization System (SMVS). SA exists on each of the hosts where its function is monitoring the host, collecting the security information according to the SMS's security policies. This information is sent to the SMS when a request message is received from it. SMS plays a role of client for SA and a server for SMVS. It processes and saves the basic security information collected from SA and provides the results to SMVS. The SMVS sends the analyzed results through the Web to the security manager in an understandable, convenient, and easy to use form.

The disadvantages of this model can be stated as follows: First, the management may be stopped when the server goes down. Second, the load on the server becomes high because all the clients send the security information to it. These problems result from the process of management centralization.

3.2 Static Agents NSM

The static agent (SA) is an autonomous entity which observes and acts upon an environment and directs its activity towards achieving goals. Also, it may learn or use knowledge to achieve these goals. A hierarchal architecture of the SA-NSM is proposed and studied by[5, 18] based on a multi-agent system. It is viewed as a collection of autonomous and intelligent agents located in specific network entities named NSM hosts. These agents cooperate and communicate in order to perform intrusion detection tasks efficiently and achieve consequently better performance.

The structure is divided into two layers, manage layer and local layer. The manager layer manages the global security of network which is either local or distributed. Besides, the local layer manages the security of domain which consists of a set of hosts. This layer is composed of a group of agents (extranet, intranet and internal local agents) which have specific monitoring roles. The manager layer interacts with the local layer by sending goals, delegating specific monitoring and/or detection tasks and receiving pertinent reports and alarms.

This system has two main disadvantages. First, it does not provide the adaptation and flexibility features, which cannot be upgraded easily and cannot easily adapt their intrusion detection tasks to changes in networks and user behaviors. Second, it does not have the ability to learn a new attack.

3.3 Mobile Agents NSM

MA is a type of software agent, with the features of *autonomy*, *social ability*, security, intelligence, *learning*, and most importantly, *mobility*. Also it has the advantage of requiring a very low network bandwidth resulting in a reduced traffic for network security management. This technique has been studied by[19]. It uses mobile agents and policies in which the administrator determines the unified policy and the mobile agent monitors the network and takes the appropriate actions depending on the associated policy. Figure 1 shows the components of a mobile agent NSM. The first object is the Agent Manager, where it's main task is to manage and collaborate agents according to the agent control protocol, including: creating, running, hanging up, terminating agent, receiving the agent, protecting and verifying agent, creating a location transparence, and recording correlation information in audit database.

The abnormal information of the agent manager, including identity which cannot be authenticated, access of unmatchable data, actions that have important effects on the running system, other actions correlative to system security are stored in Audit Log. The Policy Server is an object that represents an aggregation of the policy Rules (security policy). The Security Policy Database stores security policy established by the system administrator or formed by online study function and finally the main task of the Mobile Agent is to monitor the security status of each device to enforce system security policy.

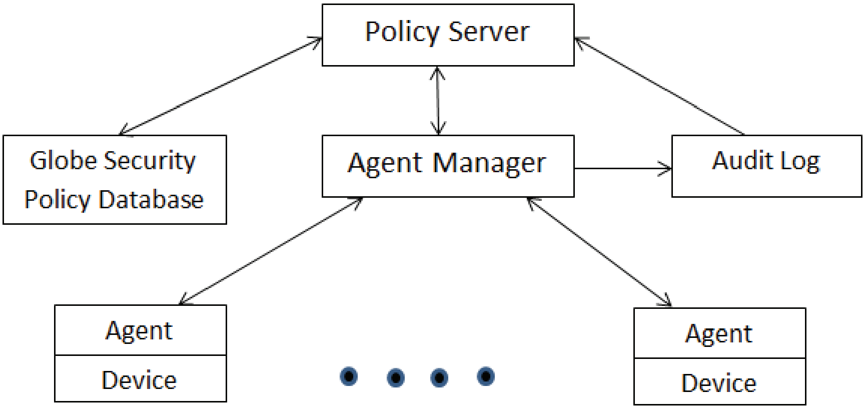


Fig. 1. Security framework based on mobile agent.

There are many advantages of this technique that lead to a high performance NSM. Most importantly is the *autonomy* where the MA does the task without direct intervention by human or other agents. Secondly, the *mobility* allows the MA to migrate from a host to another to collect security information and perform the suitable action. This property reduces the network traffic.

4. Ontology-Based Mobile Agent for NSM

The three techniques discussed in the previous section use policies to manage the network security. These policies are either represented by a database or by some policy language. Representing the policies by a database is not an effective way, because the database does not represent the policies in a semantic manner. Also, representing the policies using different policy languages may lead to difficulties for implementation of the management system.

The need for an ontology-based approach applied to security management has been proposed in[16]. Donner[20] defines a standards-based security ontology, which extends the Common Information Model (CIM) with ontological semantics. The integration of the ontology-based and policy-based approaches is proposed in[13]. The combination of mobile agent technology and the ontology representation of network security knowledge is a promising technique since it offers a lot of advantages, the most important of which is that ontology is machine readable and understandable. Consequently, this technique is chosen to be the focus of the presented work.

As a case of study on a sample network, this section introduces the development of a prototype NSM system using the mobile agent technique and ontology representation of the network security knowledge. It shows the architecture and interaction scenario of the developed NSM prototype. Various tools and libraries are used to realize the prototype and produce a working implementation. It is implemented using Java language together with the following standard tools: Jena[21], Aglet , JNIRegistry Tool, and the Altova Semantic Works. Samples of the implementation details can be found in[22].

4.1 Prototype Design

The architecture of the developed NMS prototype is shown in Fig. 2a and Fig. 2b. In this architecture, the network administrator (prototype user) starts the entire network security management operation from the NSM control center. Two dispatching modes are possible, namely sequential and parallel.

In sequential dispatching, the administrator creates an instance of the mobile agent, and sends it to travel through the network from one device to another until it finally returns back, with all collected data, to the control center. Alternatively, in parallel dispatching many instances of the mobile agent are created and each instance will be sent to a specific network entity to collect the local data and return back to the NSM control center.

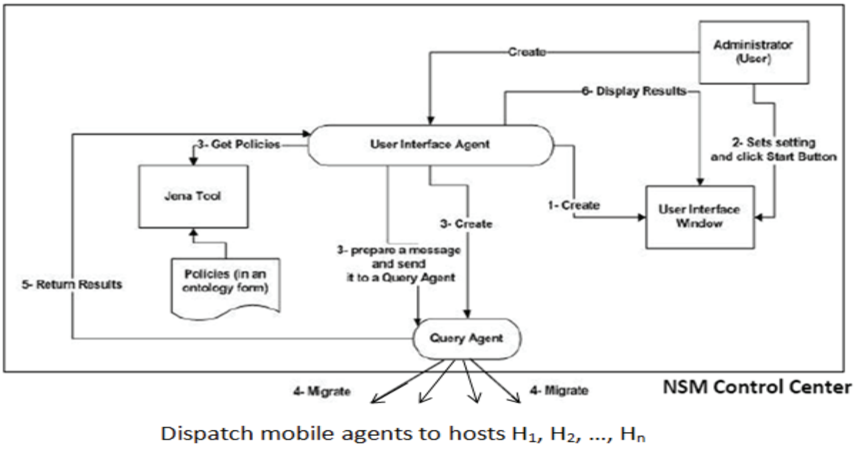


Fig. 2a. Architecture of the NMS control center.

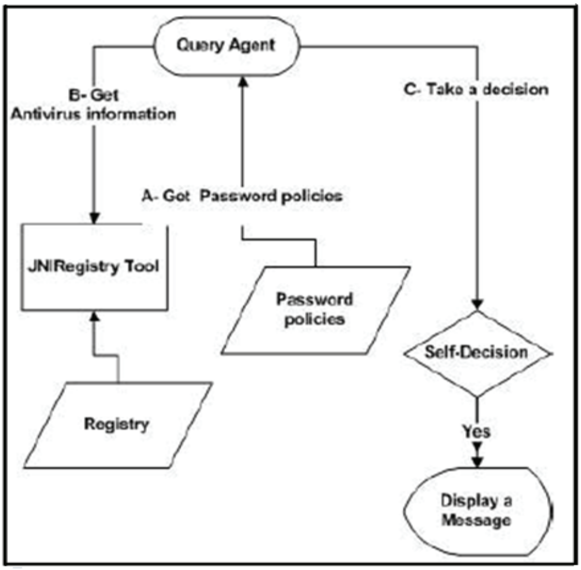


Fig. 2b. Architecture of a host in the NSM prototype.

Two types of actions are provided in this prototype depending on the type of the controlled entities, as explained next:

a) For the network devices such as routers, firewalls, and *etc*., the administrator would get all necessary information (based on the collected data) that allows him to take proper actions to reconfigure the network devices whenever necessary.

b) For every visited host, the mobile agent itself contains the necessary logic (intelligence) to examine the visited host and specify any required reconfiguration action in the form of a message displayed on the host monitor, so that the operator of that host will be informed what to do.

4.2 Interaction Scenario

The following steps show how to initialize the mobile agent and start its travel through the considered network to collect specified security information.

Step 1:

The administrator (user) runs the Aglet platform and creates the *User Interface Agent*, together with its *user interface window*. This window contains controls to set the following network security management parameters:

a) The addresses of the target hosts to be managed.

b) The type of the management (self-decision or gathering information).

c) The mode of MA dispatching (parallel or sequential).

To run the prototype, the administrator should select and set the above mentioned parameters and click the *Start* button.

Step 2:

When started by the administrator, the *User Interface Agent* will perform the following:

a) Reads policies that are represented by the ontology.

b) Creates the mobile agent, called the *Query Agent*.

c) Reads the type of the management.

d) Reads the mode of the dispatching, and acts as follows:

1. For sequential dispatching, it will send a message to the *Query Agent* containing addresses of the target hosts, policies, and the type of management.

2. For parallel dispatching, it will clone the *Query Agent* for every target host, and sends a message to every *Query Agent* clone that contains the same contents specified in step (d-1) above.

Step 3:

a) For sequential dispatching, the *Query Agent* dispatches itself to the first target and performs the following:

1- Reads the required information from the host based on the policies.

2- If the type of the management is self-decision, the *Query Agent* checks the collected information and finds out whether it conforms to the specified policies. If they are not conforming to the policies, it will show a message on that host describing the required reconfigurations.

3- If the type of the management is gathering information only, the *Query Agent* will store this information to be returned to the Administrator.

4- The *Query Agent* will dispatch itself to the next host and does the same actions 1, 2, and 3 as shown above. It will keep repeating dispatching itself to the next host until it eventually arrives back to the control center.

b) For parallel dispatching, every *Query Agent* will perform the above-mentioned actions at the visited entity and then it will dispatch itself back to the control center.

Step 4:

When the *Query Agent* arrives at the NSM control center, it will send a message to the *User Interface Agent* specifying the collected information or actions. The latter will display the received message on the *User Interface Window* at the control center.

5. Prototype Testing

The developed prototype enables the network administrator to manage the network security using the mobile agent and ontology-based querying policies. Mobile agent(s) migrate(s) to some hosts to retrieve the current configuration of the network security components.

When the MA visits an improperly configured host, it will autonomously take a corrective action by displaying a message on that host to guide its operator toward the required proper reconfiguration according to the security policy. On the other hand, when the MA visits a network device such as a router or a firewall, it will just collect all necessary security data and deliver it to the network administrator upon returning back to the control center. The administrator will consequently take the proper reconfiguration actions.

5.1 Experimental Example

The practical test described in this section is carried out over a simple LAN consisting of three computers. The considered security data is just two items as an example, namely to check the conformance with the following security policies: (a) antivirus protection policy, and (b) password policy. The three computers used in this experiment are configured as follows:

Host Name	**Configuration cases with respect to the specified policy**	
	Antivrus policy	**Password policy**
AbdallahHigh	Properly configured	Properly configured
AbdallahMedium	Improperly configured	Properly configured
AbdallahLow	Improperly configured	Improperly configured

5.2 Experimental Setup

To initiate the NSM prototype, the administrator will run the Aglet platform and create the *User Interface Agent.* When started, the *User Interface Agent* creates the **User interface Form** that is used by the administrator to run the experiment.

The **User interface Form** contains controls to set the required parameters needed for management such as the addresses of the hosts to be managed, the mode of dispatching agents (either sequential or parallel) and the type of management either for gathering information and making decisions or returning the gathered information to the administrator who, in turn, makes the decision. The following experiment is done using sequential dispatching and self-decision management.

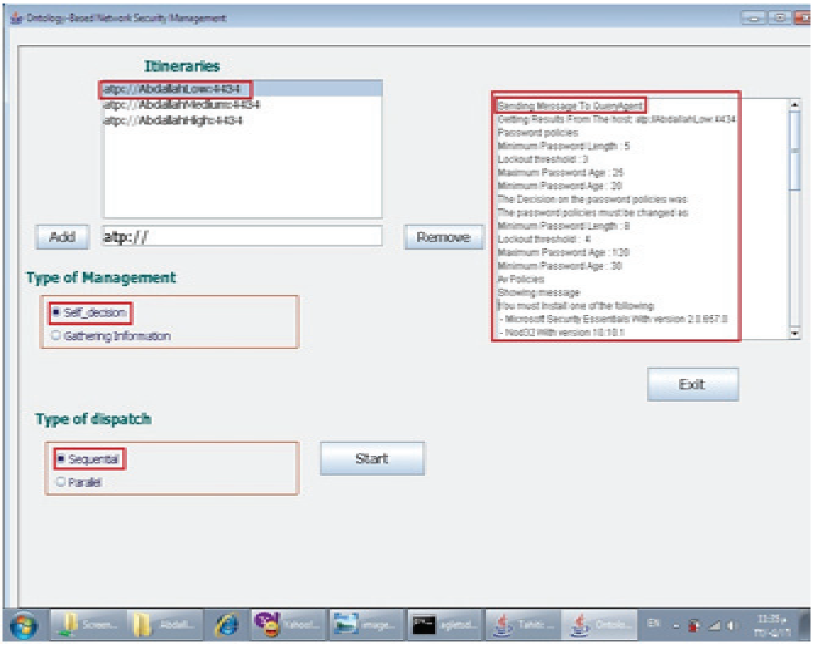
5.3 Experimental Results

Here we show the decisions taken by the MA upon visiting the improperly configured host '**Abdallah Low**'. Figure 3 shows (in the right window) both the collected data, and the policy information. The MA finds out that this host is improperly configured and decides to display the corrective actions. The MA will compose and display the following two messages on the visited host's screen instructing its user to do the necessary reconfigurations according to the policies as specified by the ontology.

Minimum password Length should be at least eight characters, maximum Lockout threshold should not exceed 4 trials, Maximum password age should be changed at most after 120 days, and Minimum password age should be changed at least after 30 days.

You must install either the "Microsoft Security Essentials with version (2.0.657.0)", or the "Nod32 with version (10.10.1)".

Similar results are obtained for the other two hosts specifying the reconfiguration message whenever necessary. The same experiment was repeated with parallel dispatch in which three separate MAs are generated and simultaneously dispatched to the three hosts. Similar results were obtained but of course in much less time due to the parallelism.



Itineraries

atp://AbdallahLow:4434
atp://AbdallahMedium:4434
atp://AbdallahHigh:4434

atp://

Type of Management

● Self_decision
○ Gathering Information

Type of dispatch

● Sequential
○ Parallel

```
Sending Message To QueryAgent
Getting Results From The host: atp://AbdallahLow:4434
Password policies
Minimum Password Length : 5
Lockout threshold : 3
Maximum Password Age : 25
Minimum Password Age : 30
The Decision on the password policies was:
The password policies must be changed as:
Minimum Password Length : 8
Lockout threshold : 4
Maximum Password Age : 120
Minimum Password Age : 30
Av Policies
Showing message
You must install one of the following
 - Microsoft Security Essentials With version 2.0.657.0
 - Nod32 With version 10.10.1
```

Fig. 3. The MA decision making in the improperly configured 'AbdallahLow' host.

5.4 Prototype Performance Evaluation

The system performance is examined using a standard 100 Mbps Ethernet. The prototype is run several times in four different modes and the number of hosts in the LAN is changed for each mode. All hosts were improperly configured. In each case we measured the total real time needed to complete the management and return back to the administrator. The obtained time values are given in Table 1.

Table 1. Total time needed to complete the management measured in milliseconds.

No. of Hosts	**Mobile Agent Mode**			
	Sequential + Self-decision	**Parallel + Self- decision**	**Sequential + Collecting Information**	**Parallel + Collecting Information**
1	235	250	219	235
5	1281	330	1140	300
10	2550	420	2300	390
15	3800	500	3490	484

Results show that in the sequential dispatch mode the management time is almost linearly proportional to the number of visited hosts. On the other hand a little increase takes place in the parallel dispatch mode. Ideally it should be constant for any number of hosts, but it seems that the queuing delays in the LAN correspond to the observed slight increase when there are more hosts. All results are more or less the same whether the mode was “self-decision” or “collecting information”. From these tests, we can conclude that the implemented NSM prototype is efficient and has a relatively good performance especially when using the parallel dispatch mode**.**

6. Conclusion

The present work presents a prototype implementation for a model to manage network security using mobile agents and ontology-based policies. The prototype aims to explore the effectiveness of that model in managing network security. As a sample case of study, the developed prototype is designed to examine security policies for the antivirus programs and Windows passwords policies. Excellent performance is achieved.

The developed system is flexible and scalable which is an advantage compared to the traditional policy-based security management that uses either the client/server paradigm or the multiple-static-agent’s paradigm. The prototype is tested using a standard local area network of up to 15 hosts, and the test results showed that it is feasible to use the mobile agent and ontology to provide a sound and efficient NSM system. In particular, the management of network security is done by specifying the policies and representing them by ontology. This permits the mobile agent to automatically accomplish the management tasks because the ontology is machine readable and understandable. Therefore the mobile agent and ontology make the NSM prototype sound and efficient.

Many tests are conducted to examine the performance of the developed NSM system. The system is set to collect security information from up to 15 hosts in a local area network, and the total management real time is measured. For the sequential dispatching, it is found that the management time is almost linearly proportional to the number of visited hosts, with a maximum value of 3800 milliseconds for 15 hosts. On the other hand, for the parallel dispatching the total management time is found to increase slowly with the number of hosts, due to the queuing delays in the LAN, with a maximum value of 500 milliseconds for 15 hosts. This gives an evidence of the inherent efficiency of the considered NSM model. A final comment on the design of the developed mobile agent is that it can be easily and directly expanded, by adding more ontology, to cover much more security aspects without losing its inherent efficiency.

References

[1] **Martimiano, L.A.** and **Moreira, E.S.**, An OWL-based Security Incident Ontology. *Proceedings of the Eighth International Protege Conference*, pp: 43-44 Poster, 18-21 July, Madrid, Spain, (2005).

[2] **Bhattacharya, S., Malhotra, S.** and **Ghsoh, S.K.**, A Scalable Representation towards Attack Graph Generation, *Proceedings of the 2008 1st International Conference on Information Technology*, 19-21 May 2008, Gdansk, Poland, pp: 1-4, (IT 2008).

[3] **Ou, X., *et al***, *Network security management with high-level security policies*, Technical report TR-714-04, Computer Science Dept, Princeton University (September 2004).

[4] **Moffett, J.**, Network Security Management, *IEE Colloquium on_Security and Networks*, **4**: 1-7 (1990).

[5] **Al-Hamami, A.H.** and **Hashem, S.H.**, A Proposed Multi-Agent System for Intrusion Detection System in a Complex Network, *The 2nd International Conference on Information & Communication Technologies: From Theory to Applications*, 24-28 April, Damascus, Syria, (ICTTA '06), **2**: 3552-3556, (2006).

[6] **Wu, Z., Xiao, D., Xiao, M.** and **Peng, X.**, Using Multilevel Correlation in a Unified Platform of Network Security Management: Design and Implementation, *2008 International Symposium on Electronic Commerce and Security*, IEEE, 3-5 Aug., pp: 402-406, (2008).

[7] **Apostolopoulos, T.K.** and **Daskalou, V.C.**, The role of the time parameter in a network security management model, *Proceedings of the Second IEEE Symposium on Computers and Communications*, 1-3 Jul, pp: 528-532, (1997).

[8] **Kim, S.C., Choi, Y.S.** and **Chung, J.W.**, Study of security management system based on client/server model, *IEEE International Conference on Communications*, (ICC '99), **2**: 1403-1408, (1999).

[9] **Basile C., Lioy, A., Scozzi, S.** and **Vallini, M.**, Ontology-based Security Policy Translation, *Journal of Information Assurance and Security*, **50**: 437-445, (2010).

[10] **Tonti G. *et al***, Semantic Web Languages for Policy Representation and Reasoning: A Comparison of KAoS, Rei, and Ponder, 2nd *International Semantic Web Conference*, (ISWC2003*)*, Sanibel Island, Florida, USA, 20-23 October (2003).

[11] **Johnson M., *et al.***, KAoS Semantic Policy and Domain Services: An Application of DAML to Web-Services-based Grid Architectures, *Proceedings of the AAMAS 03 Workshop on Web Services and Agent-Based Engineering*, July, Melbourne, Australia, (2003).

[12] **Uszok, A.**, KAoS policy and domain services: Toward a description-logic approach to policy representation, deconfliction, and enforcement, *Proceedings of the IEEE 4th International Workshop on Policy*, 4-6 June, Lake Como, Italy, pp: 93-96, (Policy 2003).

[13] **Xu, H. *et al.***, Towards Automation for Pervasive Network Security Management Using an Integration of Ontology-Based and Policy-Based Approaches, *3rd International Conference on Innovative Computing Information and Control*, 18-20 June, Dalian, (ICICIC '08), 87-87, (2008).

[14] **Fikes, R. *et al.***, Distributed repositories of highly expressive reusable ontologies, Intelligent Systems and their Applications, *IEEE* , **14**(2): 73-79, (Mar/Apr 1999).

[15] **Guerrero, A., Villagrá, V., López de Vergara, J.E., Sánchez-Macián,** and **A., Berrocal, J.**, Ontology-based policy refinement using SWRL rules for management information definitions in OWL. Proceedings 17th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM'2006), Dublin, Ireland, 23–25 October. ISBN 3-540-47659-8. *Published in Lecture Notes in Computer Science*, **4269**: 227-232, Springer Verlag (2006).

[16] **Tsoumas, B**., Towards an Ontology-based Security Management, *Proceedings of the 20th International Conference on Advanced Information Networking and Applications*, 18-20 April, Vienna, Austria, (AINA'06), **1**: 985-992, (2006).

[17] **Blanco, C. *et al.***, A Systematic Review and Comparison of Security Ontologies, *Third International Conference on Availability Reliability and Security*, 4-7 March, Barcelona , Spain, (ARES 08)**,** pp: 813-820, (2008).

[18] **Boudaoud K. *et al.***, Network Security Management with Intelligent Agents, Network Operations and Management Symposium, (NOMS 2000) *IEEE/IFIP*, pp: 579-592, (2000).

[19] **Tao J. *et al.***, The Research on Dynamic Self-adaptive Network Security Model based on Mobile Agent, *Proceedings of the 36th International Conference on Technology of Object-Oriented Languages and Systems*, TOOLS - Asia, pp: 134-139, (2000).

[20] **Donner, M.,** Towards a Security Ontology, *IEEE Security and Privacy*, **1**(3): 6-7 (May-June 2003).

[21] **Carroll, J. J. *et al.*,** Jena: Implementing the Semantic Web Recommendations, *Proceedings of the 13th international World Wide Web conference on Alternate track papers & poster, 17 – 22 May*, New York, New York, USA, (WWW Alt. '04), pp: 74-83, (2004).

[22] **Marich A.,** Ontology-Based Network Security Management Using Mobile Agent, *M.Sc. Thesis*, FCIT, King Abdulaziz University, (2011).

إدارة أمن الشبكات باستخدام تقنية الوكيل المتنقل المبني علي التوصيف الألي

عبدالله مارش علي، ومحمد أشرف مدكور*، وعمر عبدالله باطرفي*

قسم علوم الحاسبات وقسم تقنية الحاسبات، كلية الحاسبات وتقنية المعلومات*
جامعة الملك عبدالعزيز، جدة، المملكة العربية السعودية

المستخلص. إن استخدام الوسائل الأوتوماتيكية لإدارة أمن الشبكات المتوسطة والكبيرة الحجم لهو أمر شديد الأهمية لتجنب الأخطاء المتوقع حدوثها عند تدخل العنصر البشري في عمليات إدارة الشبكة. يمهد هذا البحث الطريق لتطوير منظومة أوتوماتيكية لإدارة أمن شبكات الحاسبات تتميز بالمرونة في تحديد الأهداف وأيضًا الكفاءة في استخدام سعات الاتصال بالشبكة عن طريق الإقلال من كميات الإرسال الخاصة بإدارة الأمن. ويتم ذلك بواسطة تخصيص وكيل متنقل لزيارة جميع الحاسبات وأجهزة الاتصال بالشبكة لتجميع وفحص البيانات اللازمة لعمليات إدارة الأمن لكل جهاز. ويستخدم التوصيف الآلي لتحديد سياسات الأمن للأجهزة المختلفة بالشبكة بحيث تتمكن برمجيات الوكيل المتنقل من فهم هذه السياسات والتصرف بموجبها.

لقد تم تطوير وبناء واختبار نموذج أولي للمنظومة المقترحة بالتطبيق على شبكة محلية لدراسة مدى قابلية الأفكار المطروحة للتطبيق العملي. يقوم الوكيل المتنقل في هذا النموذج برحلة للمرور على مختلف الأجهزة بالشبكة المحلية ومن ثم تجميع البيانات اللازمة من كل جهاز يزوره بناء على التوصيف الآلي لسياسات الأمن. يقوم الوكيل المتنقل بنفسه باتخاذ ما يلزم في بعض الحالات البسيطة، أما في الحالات التي تستلزم تدخل مدير الشبكة فإن الوكيل المتنقل ينشئ ويحتفظ بتقارير عن هذه الحالات لتسليمها إلى

مدير الشبكة في نهاية الرحلة ومن ثم يتخذ المدير ما يلزم من الإجراءات.

لقد تم اختبار النموذج الأولي على شبكة محلية مكونة من ثلاث حاسبات ذات إعدادات مختلفة من حيث متطلبات الأمن وذلك بهدف التحقق من سلامة أداء النموذج. وتبين من هذا الاختبار مقدرة النموذج على تفهم التوصيف الآلي والتجول في الشبكة المحلية المذكورة لأداء المهام المطلوبة طبقا لهذا التوصيف. ولقد تمكن النموذج من التعرف على مواطن الإعدادات المخالفة لسياسات أمن الشبكة المعطاة في التوصيف الآلي وتحديد الأجهزة المخالفة ومعرفة نوع المخالفات فيها.

جدير بالذكر أن منظومة إدارة أمن الشبكات المقترحة في هذا البحث تتميز بقابليتها للتطبيق العملي في شبكات تحتوي على أعداد كبيرة من الحاسبات والأجهزة وأيضًا تصلح للاستخدام في شبكات كبرى.