FRAMEWORK FOR SECURING MOBILE SOFTWARE AGENTS

Jeffy Mwakalinga and Louise Yngström

Department of Computer and System Sciences, Royal Institute of Technology, 164 40, Kista, Sweden
Tel: +468 161 721 Fax: +468 703 9025

jeffy@kth.se, louise@dsv.su.se

ABSTRACT

Information systems are growing in size and complexity making it infeasible for human administrators to manage them. The aim of this work is to study ways of securing and using mobile software agents to deter attackers, protect information systems, detect intrusions, automatically respond to the intrusions and attacks, and to produce recovery services to systems after attacks. Current systems provide intrusion detection, prevention, protection, response, and recovery services but most of these services are manual and the reaction time is usually from a number of hours to days depending on the complexity of the systems. There are efforts of using mobile software agents to provide these services automatically, thereby reducing reaction time, but the technology is not widely accepted due to security issues of mobile agents.

In this work we have created a framework for securing mobile software agents in information systems. Communication security between platforms, protection of the baggage carried by agents, and protection of agents are provided. The framework has five components: deterrence, protection, detection, response and recovery sub-systems. The framework has been partially implemented and has: an interface for administrators; monitored systems; NIST vulnerability database; patches' database; sensors; and Secure Mobile agents Run-Time System. This framework provides security for mobile agents at different levels and this increases trust in agents' technology. The response time, after intrusions are detected, is shortened.

KEY WORDS

Software mobile agents, deterrence, neural networks, immune system, genetic algorithm, and feedback mechanism.

FRAMEWORK FOR SECURING MOBILE SOFTWARE AGENTS

1 INTRODUCTIOIN

This work is aimed at studying the methodologies of securing mobile software agents. Software mobile agents are computer programs that perform tasks, like looking for the best airline ticket, buying shares in stock markets, and testing networks for vulnerabilities, on behalf of human beings. An agent consists of three main components [Cheng, 1997]: header, code, and a database. The header contains identity of the agent, agent attributes, signatures, travel paths, level of trust, ownership and other related information. The code section contains a system of programs performing the specific tasks of the agent. The database contains internal and the external information, collected while traversing in different environments.

This work is part of the investigation of creating of a generic security framework of survivable systems based on the Systemic-Holistic paradigm [Yngström, 1996] and the Immune system [Somayaji et al, 1997], where the general aim of the investigation is to identify features that protect living systems and may be used to secure information systems. Particular attention is paid to the immune system that protects people from different negative conditions in ever changing environments. The immune system uses cells, B-cells and T-cells to protect the body. In this work we use mobile agents in the place of cells in the immune systems.

According to [NIST, 2000] security requirements on agent frameworks include confidentiality, integrity, accountability, availability and anonymity. Confidentiality is required so that all the classified information can be kept secret at agent platforms and while being carried by the agents. Communications between agents and between agents and agent servers should also be confidential. The activities of agents should also remain confidential so the audit logs of their activities must be protected. Integrity of agents' code, state, internal data and collected data should be provided to ensure that unauthorized modification of code, state and data is not done. Agents should be able to detect when there is modification of their code, state and data. Accountability includes identification, authentication and audit of human users, agents and agent servers [NIST, 2000] . Audit trails of agents should also be kept to help tracing activities in case of errors. Agents and agent servers must authenticate each other before performing any transactions. Availability of information and services to mobile agents must be ensured. The agent servers must support simultaneous access, allocate resources fairly, be able to recover from different failures and so they should have fault-tolerance measures. Agent servers should scale and be able to handle requests from many agents. When the agent servers are not able to provide this service they should notify agents about it.

1.1 Related Work

1.1.1 A Distributed Intrusion Detection System Using Mobile Agents

This system [Kannadiga et al, 2005] has components: intrusion detection system console; the mobile agents dispatcher (MAD); and hosts. The system uses mobile software agents. Alerting agents (AA) reside in the IDS console and are used for receiving alerts that are generated by mobile agents (MA). Mobile agents (MA) are responsible for gathering and analyzing evidences of intrusions and attacks from different hosts. Each mobile agent is specific for a certain type of intrusions. Static agents (SA) reside in hosts with responsibility to monitor in the hosts. Static agents create a number of threads and each one is responsible for different kinds of attacks. MAD manages the dispatching of mobile agents to handle the requests that were generated by static agents. The requests are stored in a list called victim host list (VHL). This system has been implemented and it covers doorknob-rattling attacks, chain/Loop attack, distributed port scanning,

and distributed DoS attacks. In the doorknob rattling attacks attackers try to log in a system using a few common usernames and passwords. In he chain/lop attack, an intruder uses different machines to hide her identity and it is challenging to trace the origin of the intrusion. In distributed port scanning a number of distributed machines are used for scanning making it difficult to detect the origin of scanning. The system is effective in analyzing incidents but it lacks security of the individual mobile agents that are involved in performing different tasks.

1.1.2 A safe Mobile Agent System for Distributed Intrusion Detection

[Zhong et al, 2003] have created a system for distributed intrusion detection based on the mobile agents. The system has a manager, an assistant mobile agent, a response mobile agent, a host-monitoring agent, and three host-monitoring sub-agents. When an intrusion occurs the host-monitoring agent will send an alert to the manager. The manager will dispatch an assistant mobile agent to the different hosts to determine whether the intrusion is distributed or not. The assistant mobile agent will bring a report and the manager will analyse the report and then send the response mobile agent to all the hosts to fix the incident. The sub-agents are monitoring: network connections, size of packages, headers of packages and arriving times; different file operations and privilege operations. The results of the monitoring are sent to the intrusion analyser where they are interpreted in accordance to the interpretation trees. The security of mobile agents, confidentiality, authentication and integrity, is discussed but not in details. It is not described how the agents can be traced in case some problems occur during the traversing of agents. Baggage security is not discussed either.

2 SECURITY FRAMEWORK FOR AGENTS

This framework contains the following subsystems: agent generator and database, system manager, integrated security system, general database, special analysis, deterrence, protection, detection, response, and recovery a shown in figure 1. The agents are generated in the agent generator component. The system manager is responsible for the overall administration. The integrated security system is responsible for managing and providing security services, managing digital and attribute certificates and database services. The general database is keeping the main records of the system. Special analysis is used for analysing the different intrusions and abnormalities in the system. All the components communicate with other components in the system. The sub systems request mobile software agents from the agent generator. All agents are trained before being released into the real environment. In the first phase of the training, the agents pass a negative selection test [Kim, 2002]. The agents that pass this test go through the clonally selection test [Kim, 2002]. After this test the agents are ready to be deployed in the real environment and they are sent to the sub systems that requested the agents. When agents are released in the real environments they are monitored to record their activities. The features of the most successful agents, according to policy specified criteria, are recorded and used to improve the features of next generation of agents.

2.1 Register System

The function of this system is to verify the data of the entity with the administrator before issuing an ID, which is a mini certificate [Framework, 2004]. After verification of the data, a role is given to the entity, location of operation and other authorization parameters. If verification is not successful the entity is killed or sent to for analysis. If verification is successful the mini-certificate is issued. An encrypted authentication key is bound to the mini-certificate. An attribute certificate is issued and a public key certificate is issued. Attribute certificates are issued depending on the policy of the system and on the sensitivity of the environment. The entity is then registered into the database. Thereafter a status, a timestamp and agent's ID are recorded and a report is sent to the general DB. The message authentication code of the contents is created and encrypted; a denial of service cookie is created and signed. The level of security in an environment has a minimum and no

maximum. The owners of the system decide their maximum level of security depending on their environment.

2.2 Mobile Agents

There is a system of mobile agents that are used for providing different security services in the sub systems. These include: helper agents; authentication agents; confidentiality agents; authorization agents; Non-repudiation and integrity agents; and third-order feedback agents system. All agents have the capability to kill/terminate intruding programs or processes.

2.2.1 Helper agents

The helper agent is responsible for delivering messages among agents and subsystems. They take as input a message and verify the ID and signature of the message. If verification is not successful the message is deleted or taken for analysis. If verification is successful, the helper agent puts a status, a timestamp and the agent's ID. The message is then delivered to the requested destination and a signed and time stamped report is sent to the general database.

2.2.2 Authentication Agent

This agent is responsible for providing authentication security service in the system and sub systems. It verifies identities of entities in a system. It monitors the system and checks the identities of entities in the system, even as B-cells in the immune system check whether all cells that are in the human body are marked 'self'. It verifies the mini-certificate of the entity and checks with the general database to see if it is registered. If the entity is not registered or if the ID is not correct this agent kills the entity or sends it for analysis. If verification of the identity is successful then the agent puts the status, timestamp and agent's ID. The agent then sends it to the authorization agent and sends a report of the action to the DB.

2.2.3 Confidentiality Agent

This agent provides confidentiality security service in the system, deterrence, protection, detection, response and recovery sub-systems. It first verifies the ID of the entity or message to be encrypted. It then verifies the authenticity of the sender, the authorization that the sender has. If verification of the message and the sender is successful then the agent encrypts the message or entity, put a status, a timestamp and agent's ID. Then, it sends the message to the requesting system. It also sends a signed report to the general DB for audit purposes.

2.2.4 Authorization Agent

This agent provides the authorization security service. It first verifies the identity of the entity. Then it checks in the database whether this entity is registered. It then checks the signature. If verification is not successful this agent kills the entity or sends it for analysis in accordance to the specified policy. The agent also checks the attribute certificate of the entity. The attribute certificate has fields like roles, time and place of operations, delegation. The attribute certificate is also bound to the public key certificate. If the attribute certificate is not appropriate (expired, corrupted) the agent sends it to the registering system through the helper agent.

2.2.5 Non-repudiation and Integrity Agent

This agent is responsible for providing the non-repudiation and integrity security services in the system and sub systems. It takes as input an entity and it verifies the entity's ID. If ID verification is successful the agent signs or verifies the signature on the entity. If not the agent kills the entity or sends the entity for analysis via the helper agent. The agent put the status, a timestamp and agent's

ID. The agent sends the entity to the requesting system. It also sends a report of the action to the general DB.

2.2.6 Third Order Feedback Agents System

There are a number of agents in the third order feedback system [Schoederbek et al]. The detector agent notifies the entity the value that needs to be changed. The effector agent supplies the required change. The recall agent produces historical decisions that have been made in the past from the memory. The recombination agent produces a number of recombination from the memory and the selection agent selects those that can be further modified. The system of decision agents makes decisions based on the fuzzy logic controller and the Neural Network's Adaptive Resonance Theory (ART) [Dasgupta et al, 2001]. A report is then sent to the DB.

2.3 Deterrence System

This system is aimed at scaring away attackers from a system. It has sections: inputs; process; outputs; and a feedback mechanism. It takes as input traffic and processes it and the outputs are fed back and modified using the third order feedback system as shown in figure 1. The deterrence system classifies the inputs. It then kills the categories or allocates the different categories to the proper agents in the process section. This subsystem has a number of agents depending on the policy of the system.

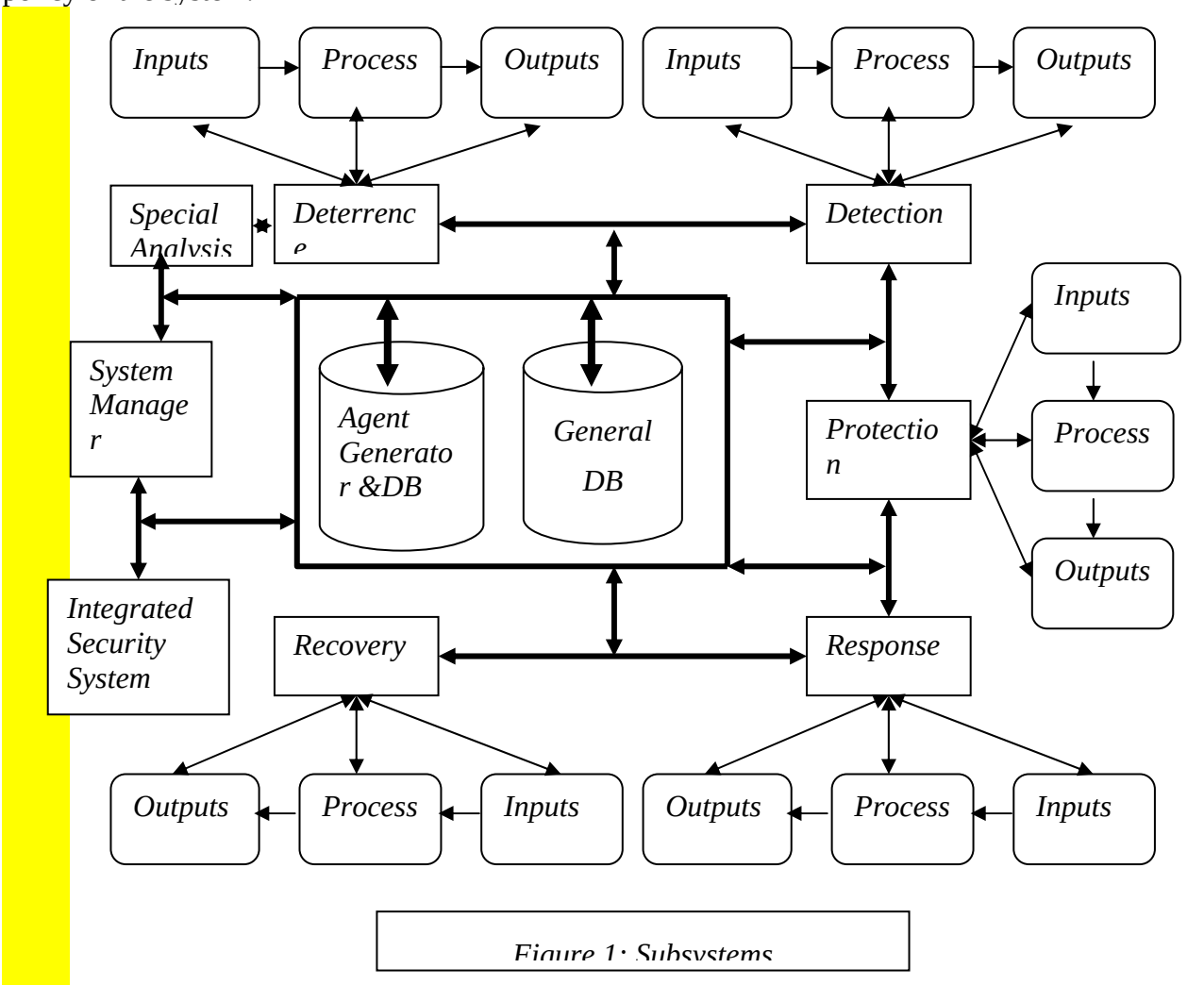

Figure 1: Subsystems

Every deterrence agent specializes in different types of surveillance. The agents can decide to: trace the surveillance or scanning efforts; kill them; reply nothing or with a legal action message; or send for analysis if the type of surveillance is unknown. The sub-system also has: helper agents;

authentication agents; confidentiality agents; authorization agents; non-repudiation and integrity agents; and third-order feedback agents system. These agents are present in all the sub-systems.

2.4 Protection Sub System

This subsystem is responsible for providing the security services authentication, confidentiality, integrity, authorization, and non-repudiation. It has inputs, processing, outputs and feedback mechanisms. These security services are provided with the help of the integrated security system and software mobile agents. The integrated security system manages public key certificates, authorization, directory services, and smart cards through certification, smart card, directory and authorization systems. Software mobile agents perform authentication, integrity, authorization, non-repudiation, and confidentiality security services. All the entities in a system are given identities even as all cells in the human body are marked as 'self'. Cells patrolling in the body continuously check for cells that are 'no-self'. The identities given to the entities in a system are in a form of a mini-certificate. The mini-certificate has the following fields: a unique ID; a group ID; Encrypted location; Encrypted authentication key; an encrypted key; Denial of service cookie; an encrypted certificate serial number of the entity; Encryption attribute certificate number; an encrypted attribute number of the entity; and a signature

2.5 Detection Subsystem

This system has inputs, processing, outputs and feedback mechanisms as shown in figure 1. In this subsystem we use the general mobile architectures. This sub system acts as a detection and prevention system since it detects and responds. The system takes as input traffic and classifies using the Adaptive Resonance Theory (ART) [Dasgupta et al, 2001] or kills the incoming traffic or closes the system. Agents make decisions basing on the fuzzy logic engine [Dasgupta et al, 2001] and genetic algorithms as described by [Pillai et al, 2004]

2.5.1 System Design for Network Intrusion Detection System using Genetic Algorithm

Genetic algorithms are based on the principle of survival of the fittest and the goal is to find a set of parameters or values that maximize a certain fitness function. An example of a fitness function could be $F(x) = y^3 + 2\cos(3y)$. There is a set of all possible y values. And the goal is to find a set of y values from this set that maximizes the fitness function. In the system [Pillai et al, 2004] a data set is created that specifies the normal and abnormal behaviors by analyzing traffic packets from a network sniffer, like a TcpDump or Win Dump, which records traffic [Pillai et al, 2004] . This dataset can have data like source IP, destination IP, source port, destination port, and the protocol used and an intrusion indicator [Pillai et al, 2004]. This data set is used for training a genetic algorithm. After training the data set can be updated and connections added [Pillai et al, 2004]. The rule set is established in the form "if {condition} then {act}" [Pillai et al, 2004] and thereafter a genetic algorithm rule set is created in GA format. In the initial stages the first part will function as search algorithm to get values for each rule to indicate the good rules. There after the genetic algorithm is used as a fitness function to determine the fittest rules. The fitness function used in [Pillai et al, 2004] is $F = a/A - b/B$ in which "*'a' contains the value that the specific rule carries for the number of correctly detected intrusions. 'b' contains the value that the specific rule carries for the number of false alarms. A is calculated by adding the value of correctly detected intrusions from all the rules. B is the total number of connections in the dataset*". *"When an intrusion occurs, it is notified by a response mechanism. The response mechanism is a pop up window indicating the rule, and a message notifying that an intrusion has occurred.* "[Pillai et al, 2004]

2.5.2 Neural Network Classifier

The Adaptive Resonance Theory (ART) [Dasgupta et al, 2001] is used for classifying network traffic into normal and abnormal and in training the monitoring agents. ART is neural network

classifier is an unsupervised neural network using competitive learning and it does not require human supervision. During the learning stage a knowledge base is established in which network, system, user, process normal behaviors are recorded. Any other behaviors that are not in the knowledge base are categorized as intrusions [Dasgupta et al, 2001]. The detection subsystem recognizes normal patterns and any other unknown patterns are regarded as malicious. In the ART Neural network there are two filters: one represents features; the other represents categories. The initial stage in the leaning process is concerned with parallel searching scheme that updates itself adaptively [Dasgupta et al, 2001]. During this period input traffic categories are assigned recognition codes. New networks are encoded by changing weights or long term memory traces and when self-learning is stable search automatically stops [Dasgupta et al, 2001].

2.5.3 Fuzzy Logic Controller

The decision agents are basing their decisions on the fuzzy logic controller [Dasgupta et al, 2001] and also from the agent generator library and database. Fuzzy logic is a concept in which objects or entities can partially belong to a set. The objects can for instance belong to a set A by 50%. The range of belonging is 0-100%. In classical sets an object or an entity is either inside or outside a particular set [MMDS, 2003]. The fuzzy logic is used in decisions because the differences between normal and abnormal behaviors in networks are not distinct but fuzzy [MMDS, 2003].

2.6 Response Subsystem

It has inputs, processing, outputs and feedback mechanisms as shown in figure 1. This sub-system is based on [Carver, et. al]. It receives alerts from the detection sub system. Interface agents are responsible for keeping history of false positives and negatives generated by each intrusion detection entity [Carver, et. al]. These agents transform IDS specific messages into a generic message format like the Knowledge Query and manipulation Language (KQML) or the Common Intrusion Detection Format [CIDF]. This history is used to create a confidence metric of each monitor. This metric and the intrusion reports are sent to the master analysis agent. The master analysis agent determines whether this intrusion is new or whether it is a continuation of the existing intrusion [Carver, et. al]. If the incident is new then a new analysis agent is generated to create a response plan for this new attack. If the intrusion is part of the existing one then the confidence metric and the intrusion report are sent to the agent handling the attack. To work out a plan for response the agent gets help from the Response Taxonomy agent, which is used to classify the attack [Carver, et. al]. The Policy Specification agent is also consulted to check the legal, ethical, institutional requirements and resource limitations. The decision made is sent to the Tactics agent by the analysis agent. The tactics agent species the action to be taken and then allocates the duty to the appropriate components of the Response toolkit. The logger agent record decisions made by the analysis and tactics agents [Carver, et. al]

2.7 Recovery System

This sub system is used for putting a system back to its normal state after an attack. It has the following agents: helper agents; authentication agents; confidentiality agents; authorization agents; Non-repudiation and integrity agents; and third-order feedback agents system; installation agents; cleaning agents; forensics agents; on-line back-up agents; off-line back-up agents.

3 SECURITY OF MOBILE SOFTWARE AGENTS

3.1 Security Services in the Generation of Agents

The Agent generator and creates agents. The agents are given identities. After generation the agents must be certified locally by the special analyzer, but in future there could be an independent body for certification of agents. Every sub system requests agents from the agent generator. The agent

generator and the requesting sub-system will mutually authenticate each other before communicating further. Every sub-system has many different agents for doing diverse kinds of tasks in this sub-system. To provide authenticity and integrity of agents the agent generator signs the agent. To provide confidentiality requirement, the agent generator seals the agent by using the public key of the special analyzer, which is acting as the agent certifier. The agent is then sent to the certifier. The certifier opens the message by the private key and verifies the signatures of the agent. The special analyzer checks whether the agent is behaving in accordance to the specifications. The certifier puts a trust level and its digital certificate [Cheng, 1997]. The analyzer signs the agent, protects it and sends it to the sub-system.

3.2 Protecting Agents, their Baggage and Securing Communication between Sub Systems

Agents carry baggage and this must be protected. During the handshake the sending and receiving sub systems exchange session secret keys to secure communication. When agents are moving between deterrence, detection, protection, response and recovery sub systems are protected in the following way. The session keys are used to protect the agent and the messages between these sub systems. The agent and the baggage are also signed by the sending sub system. The receiving sub system verifies the integrity and authenticity of the agent and messages by verifying the signature. Protecting agents when they are visiting a sub system is different from protecting agent servers [NIST, 2000] because agents don't have their own processors and they can't extend the home sub system, but have to rely on the environments provided for them there. The technology called Environmental Key Generation [12] is applied to protect all the executables if the environment is hostile. An agent generates a key and protects the executables if some environmental conditions are true. We also apply the Partial Result Encapsulation [NIST, 2000], in which the results from each visited sub system are encapsulated and signed. We also apply the sliding encryption [Young et al, 1997] in which the agent seals information every time it collects it. The agent can use the public key of the owner to seal the information, so that only when the agent returns home that the collected information is unsealed.

3.3 Protecting the Sub Systems

There are a number of technologies [NIST, 2000] for protecting agent servers. One technique is called State Appraisal [Farmer, 1996], which is a way of verifying the correct state of an agent before accepting the agent and before authorizing the agent to access objects. In this work agents are signed using the private keys of the agent generators and dispatchers. The sub-system opens the message using its private key. It verifies the signatures of the agent generator and of the special analyzer. The sub-system also verifies the authenticity of all the agent servers that were visited before the current sub-system in accordance to the Path Histories [Ordille, 1996]. The sub-system then notifies the agent generator and the certifier that it has received the agent. All the agent servers sign the information collected by the agent.

3.4 Sending an Agent for Cloning

If the agent is very successful in deterring, protecting, detecting intrusions and other tasks in accordance to the specified criteria the agent will be sent to the agent generator for cloning. In sending the agent the following procedure will be followed: the sending sub system and agent generator will authenticate each other before sending the agent; the sender creates a secret key and protects it with the public key of the agent generator, the sending sub system then signs it. When the agent generator receives the agent it will verify the signature of the sending sub system. The agent generator will then clone the agent and will send the agent back though the agent certifier. A copy of the agent is stored in the database of the agent generator.

4 PROTOTYPE OF THE SYSTEM

The architecture consists of the following Components: Interfaces for the administrators; Vulnerabilities database; patches database; a database for agents' actions log files; Monitored systems as shown in figure 8. Every interface has sub-systems: Certification system, Smart card system, directory system and authorization system. In the smart cards system one can create file systems in the smart, one can initialize the smart card, issue a smart and personalize a smart card. The authorization sub-system one can register different roles that exist in the system like administrator, normal user, security manager and so on. Another function is this system is to list roles that exist. It is also possible to remove roles whenever necessary. Another function is to register applications like databases and different programs. One can then list applications; one can assign roles to applications. If an application doesn't have a role to open a certain file access will be denied to this application. For viruses that come unnoticed to a system will not be able to run or open other files because they will not have been registered and will have no roles. Other functions in the authorization system include: assigning roles to users; removing roles from users; remove roles from applications. It is also possible to create access rules and list them. Other functions include creating a policy, removing a policy, creating and removing a policy set, exporting a policy or a policy set.

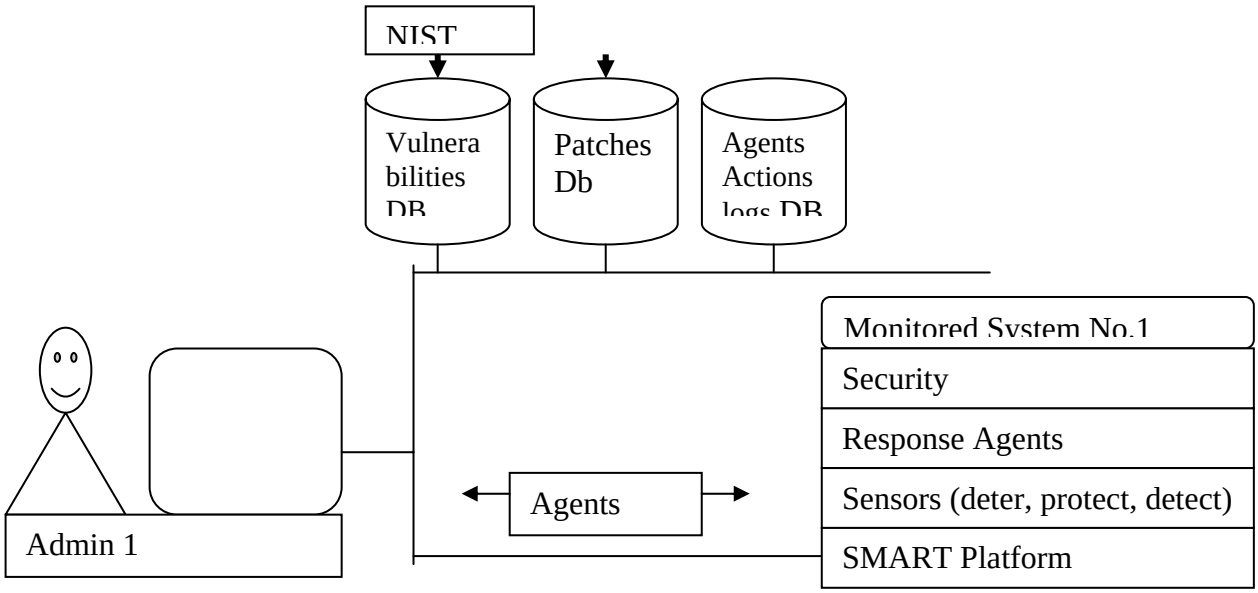

Figure 8: Architecture of the Mobile Agents Structure

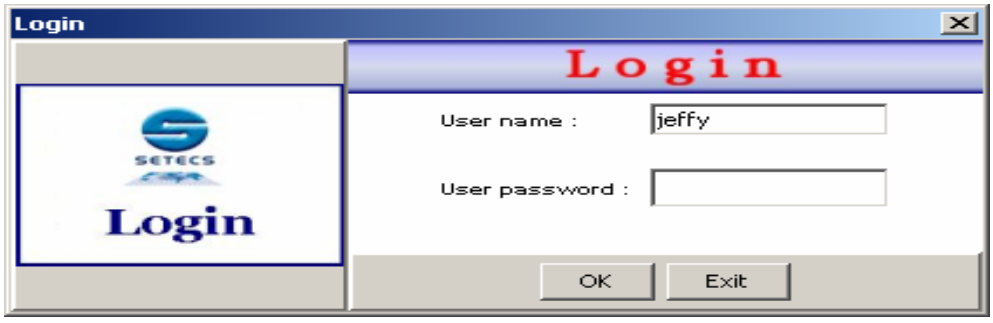

Figure 10: Login to the System

4.1 Interface

Every interface has sub-systems: Certification system, Smart card system, directory system and authorization systems. The functions in the interface include registration of users, servers, management of certificates and system administration [Muftic et al, 2001] as shown in figures 8, 9 and 10.

4.2 Vulnerability Database, Patches and Agent Logs

Most attacks are possible because of vulnerabilities in systems so if we can solve the vulnerability problem many attacks will not succeed. The vulnerability database contains vulnerabilities from the National Vulnerabilities database (NVDB) of the National Institute of Standards and Technologies (NIST). The database of vulnerabilities is downloaded from the NVDB and then it filtered from the XML format and converted to a relational database and stored and ready for querying. There is a set of mobile agents, which can access, query, read and dispatch data from the NVDB database. The patches database stores all the latest patches. All the actions of agents are logged in the agents' logs database. [Muftic et al, 2001].

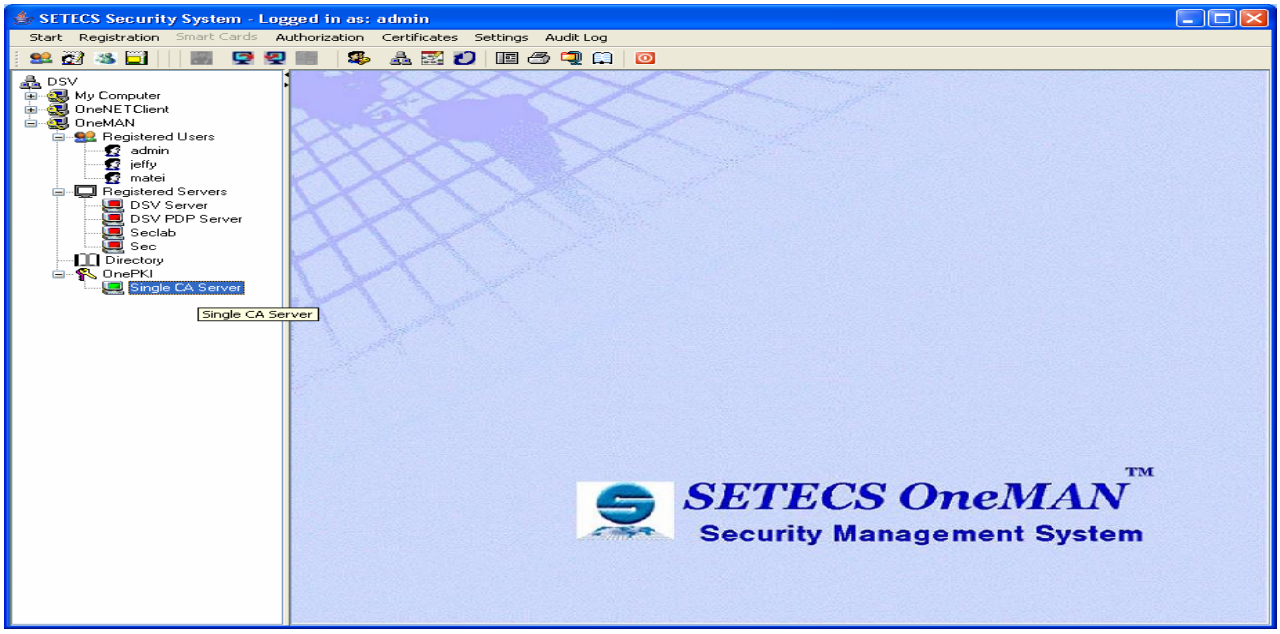

Figure 9: Interface of the system

4.3 The Monitored System

The monitored system consists of the following components: security, where different security services are performed; response agents; Sensors (deterrence, detection and protection); Secure Mobile agents Run-Time System (SMART) [Muftic et al, 2001]. When there is an alert or alarm from the sensors the response agents picks up the alerts. A secret key is used to protect the alert message using AES. This key is then protected by the system administrator's public key. A response agent is assigned the task to take the message to the interface. The message is signed by the SMART system send alert messages to the administration interface, which sends agents to take of the system via the SMART system, where an alert came from.

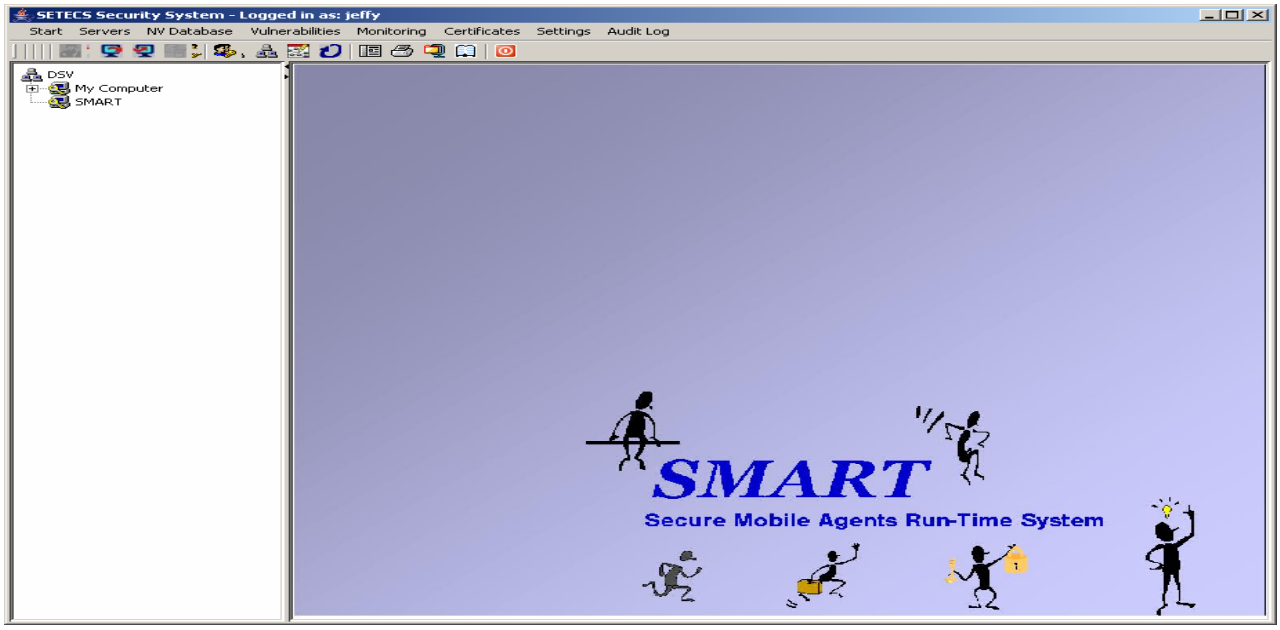

Figure 9: Interface of the SMART system

4.4 Conclusion

In this work we have created a framework for securing software mobile agents that perform different tasks in deterrence, protection, detection, recovery and response sub systems. The framework provides protection of sub systems, agents and their baggage, and communication security. The agents provide authentication, confidentiality, integrity, non-repudiation and authorization security services. The training of agents during their creation is based on the immune negative selection and clonally selection algorithms and genetic algorithms. A prototype has been created but it reflects only part of the framework that deals with maintenance. A conclusion of the general behavior can't be given from the prototype.

5 REFERENCES

[Bellifemine et al, 2005] F. Bellifemine, T. Trucco. Java Agent Development Framework: http://jade.tilab.com/index.html.(15-04-2005).

[Cheng, 1997] Yi Cheng. A comprehensive Security Infrastructure for Mobile Agents. ISRN SU-KTH/DSV/R—97/13--SE

[Dasgupta et al, 2001] Dasgupta D, Brian H. Mobile Security Agents for Network Traffic Analysis. IEEE Proceedings of DARPA Information Survivability Conference and Explosion II (DISCEX-II). 2001, Anaheim, California

[Eloff et al, 1995] J. H. P. Eloff and S.H. von Solms, Information Security – the next Decade, IFIP 1995, ISBN 0-412-64020-1

[Eloff et al, 2003] M.M. Eloff, J.H.P. Eloff. Information Security Management. A New Paradigm. Proceedings of SAICSIT, ACM, 2003

[Farmer, 1996] Farmer W, Guttman J, Swarup V. Security for Mobile agents: Authentication and State Appraisal. Proceedings of the 4th European Symposium on Research in Computer security, 96

[Grigoriadis, 2003] Anastasios Grigoriadis, Requirements for computer immune defense System based on body's immune System and DNA proofing. Masters thesis: Stockholm University, 2003.

[Framework, 2004] Mwakalinga, J, Yngström, L, Sketch of a Generic Security Framework based on the Paradigms of Systemic-Holistic Approach and the Immune System, Proceedings ISSA2004

[Jennings, 1999] N.R. Jennings. Agent-Based Computing: Promise and Perils. Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence. Stockholm, Sweden. Pp.1429-1436, 1999.
[Kannadiga et al, 2005] Kannadiga, P. Zulkenrnine, M. A Distributed Intrusion Detection System Using Mobile Agents, SNPD/SAWN'05, IEEE, 0-7695-2294-7/05
[Kim, 2002] Jung Won Kim, Integrating artificial Immune Algorithms for Intrusion Detection, Ph. D thesis, University of London, 2002
[MMDS, 2003] Dasgupta D, Gomez J, Gonzles F, Kaniganti M, Yallapu K, Yarramsettii R. MMDS: Multilevel Monitoring and Detection System. Intelligent Security Systems Research Laboratory. Division of Computer Science. University of Memphis, USA.
[Muftic et al, 2001] S Muftic, J Huang, O Gelbert, M Dean. Intrusion–Detection System based on Secure Mobile Agents Computer Science Department, The George Washington University Washington, DC, USA
[Muftic, 2006] Sead Muftic. Secure Mobile Agents Run-Time system (SMART). Report, Computer Science Department, George Washington University, Washington DC, USA
[NIST, 2000] Jansen, W. Karygiannis, T. National Institute of Standards and Technology Special Publication 800-19 – Mobile agent Security.
[Ordille et al, 1996] Ordille J. When agents Roam, Who Can You trust? Proceedings of the First Conference on Emerging Technology and Applications in Communications, Portland, Oregon, 96
[Pillai et al, 2004]. An Approach to Implement a Network Intrusion Detection System using Genetic Algorithms. M. M. Pillai, J. H.P. Eloff , H. S. Ventor. Proceedings of SAICSIT 2004, Pages 221 - 228
[Riordan et al, 1998] Riordan J. Schneier B. Environmental Key Generation Towards Clueless Agents. Vinga G (Editor). Mobile Agents Security, Springer-Verlag, Lecture Notes in Computer Science No. 1419, 1998.
[Roth et al, 1998] Roth V. Secure Recording of Itineraries Through Cooperating Agents. Proceedings of the ECOOP Workshop on Distributed Object Security and the 4th Workshop on Mobile Object Systems: Secure Internet Mobile Computations. INRIA, France 1998. URL:www.igd.fhg.de/www/igd-a8/pub/#Mobile Agents
[Schoederbek et al, 1998] Schoederbek P, Schoederbek C, Kafalas A. Management Systems, ISBN 0 07 709588 X
[Somayaji et al, 1997] A. Somayaji, S. Hofmeyr and S. Forrest. Principles of Computer Immune System, 1997 *New Security Paradigms Workshop, ACM p75-82*
[Yngström, 1996] Louise Yngström. A systemic-Holistic Approach to academic programs in IT Security, Ph. D thesis, Stockholm University / Royal Inst. of Technology ISRN SU-KTH/DSV/R--96/21--SE, 1996.
[Young et al, 1997] Young A, Yung M. Sliding Encryption: A cryptographic Tool for Mobile Agents. Proceedings of the 4th International Workshop on Fast software Encryption. FSE'97. 1997.
[Zhong et al, 2003] Zhong S., Song, Q., Cheng X., Zhang Y. A safe Mobile Agent System for Distributed Intrusion Detection, Proceedings of the Second International Conference on Machine Learning and Cybernetics. Nov 2003. 0-7803-7865-2/03, IEEE.

6 PERMISSIONS