A Network Security Architectural Approach for Systems Integrity using Multi Agent Systems Engineering

Gustavo A. Santana Torrellas

***Abstract*—Ideally, information systems security enables management to have confidence that their computational systems will provide the information requested and expected, while denying accessibility to those who have no right to it. The analysis of incidents resulting in damage to information systems show that most losses were still due to errors or omissions by authorized users, actions of disgruntled employees, and an increase in external penetrations of systems by outsiders. Traditional controls are normally inadequate in these cases or are focused on the wrong threat, resulting in the exposure of vulnerability. Security is a critical parameter for the expansion and wide usage of agent technology. A threat model is constructed and subsequently the basic techniques to deal effectively with these threats are analyzed. Then this paper presents a dynamic, extensible, configurable and interoperable security architecture for mobile agent systems. It is explained how this architecture can be used to tackle a big part of security threats. All the components of the security architecture are analyzed while we also argue for the benefits they offer.**

Key words
Mobile Agents, Security Architecture, Security Threats

I. INTRODUCTION

The primary goal of any enterprise-wide security program is to support user communities by providing cost-effective protection to information system resources at appropriate levels of integrity, availability, and confidentiality without impacting security, innovation, and creativity in advancing technology within the corporation's overall objectives. Ideally, information systems security enables management to have confidence that their computational systems will provide the information requested and expected, while denying accessibility to those who have no right to it. The analysis of incidents resulting in damage to information systems show that most losses were still due to errors or omissions by authorized users, actions of disgruntled employees, and an increase in external penetrations of systems by outsiders. Traditional controls are normally inadequate in these cases or are focused on the wrong threat, resulting in the exposure of a vulnerability.

People dealing with security have a hands-on experience with such issues. A secure system is a system that provides a number of services to a selected group of users and restricts the ways those services can be used. A security service is a software or hardware layer that exports a safe interface out of an unprotected and possibly dangerous primitive service. In order to build a security service we need a security architecture. Having analyzed the security needs of the Mobile Agent (MA) technology we propose in this paper a dynamic, extensible, configurable and interoperable security architecture for mobile agent systems. Software agents [1] are a rapidly multi-directional developing area of research since the early 90s. A general answer could be: Agents are software components that act alone or in communities on behalf of an entity and are delegated to perform tasks under some constraints or action plans. Mobile Agents shatter the notion of client/server model and eliminate its limitations. Standardization efforts and guidelines that boost the usage of agent technology exist in organizations such as the Object Management Group [11] and the Foundation for Intelligent Physical Agents [12]. Agents are computer and transport independent (they depend only on the execution environment) and therefore promote interoperability among systems and software. As the complexity of today's distributed computing environments continues to evolve independently, with respect to geographical and technological barriers, the demand for a dynamic, synergistically integrated, and comprehensive information systems security control methodology increases. Unfortunately, the prevalent attitude toward security by management and even some security personnel is that the confidentiality of data is still the primary security issue. That is, physical isolation, access control, audit, and sometimes encryption are the security tools most needed. While data confidentiality may be an issue in some cases, it is usually more important that data and/or process integrity and availability be assured. Integrity and availability must be addressed as well as ensuring that the total security capability keeps current with technology advancements that make it easier to share geographically distributed computing resources.

Proceedings of the 7th International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'04)
1087-4089/04 $ 20.00 © 2004 IEEE

Security environments have introduced significant opportunity for process reengineering, interdisciplinary synergism, increased security, profitability, and continuous improvement. With each introduction of a new information technology, there exists the potential for an increased number of threats, vulnerabilities, and risk. This is the added cost of doing security. These costs focus on systems failure and loss of critical data. These costs may be too great to recover with respect to mission- and/or life-critical systems. Enterprise-wide security programs, therefore, must be integrated into a systems integrity engineering discipline carried out at each level of the organization and permeated throughout the organization.

The purpose of this document is to provide an understanding of risk accountability issues and management's responsibility for exercising due care and due diligence in developing and protecting enterprise-wide, interoperable information resources as a synergistic organizational function.

II. THREATS IN A DISTRIBUTED AGENT ENVIRONMENT: UNDERSTANDING DISTRIBUTED PROCESSING CONCEPTS AND CORRESPONDING SECURITY-RELEVANT ISSUES

There are so many factors influencing security in today's complex computing environments that a structured approach to managing information resources and associated risk(s) is essential. New requirements for using distributed processing capabilities introduces the need to change the way integrity, reliability, and security are applied across diverse, cooperative information systems environments. The demand for high-integrity systems that ensure a sustained level of confidence and consistency must be instituted at the inception of a system design, implementation, or change. The formal process for managing security must be linked intrinsically to the existing processes for designing, delivering, operating, and modifying systems to achieve this objective.

Distributed systems are an organized collection of programs, data, and processes implemented in software, firmware, or hardware that are specifically designed to integrate separate operational systems into a single, logical information system infrastructure. This structure provides the flexibility of segmenting management control into domains or nodes of processing that are physically required or are operationally more effective and efficient, while satisfying the overall goals of the information processing community.

The operational environment for distributed systems is a combination of multiple separate environments that may individually or collectively store and process information. The controls over each operational environment must be based on a common integrated set of security controls that constitute the foundation for overall information security of the distributed systems.

The foundation of security-relevant requirements for distributed systems is derived from the requirements specified in the following areas:

- □ Operating systems and support software,
- □ Information access control,
- □ Application software development and maintenance,
- □ Application controls and security,
- □ Telecommunications,
- □ Satisfaction of the need for cost-effective security objectives.

Mobile code programming is by its nature a security-critical activity. In an agent based infrastructure the security implications are far more complex than in current static environments. In such an environment author of the MA code, the user, the owner of the hardware, the owner of the execution platform (even the execution place) can be different entities governed by different security policies and possibly competitive interests. In such a heterogeneous environment security becomes an extremely sensitive issue. We identify the threats that exist in an agent-based infrastructure. We can have: misuse of execution environment by mobile agents, misuse of agents by other agents, misuse of agents by the execution environment, misuse by the underlying network infrastructure. Mainly all security efforts target the first category and a big part of the second one. But misuse of agents by the host isn't touched almost at all. Our approach also provides protection for the two first categories and tries to provide some guarantees to the agent concerning the host code and execution environments.

MISUSE OF HOSTS BY MOBILE AGENTS

Malicious agents while visiting a host can: Destroy/reconfigure/change or even erase resources of the host. This affects all agents visiting the host that time. With various tricks or false language implementations [17] an agent can bypass authorization and authentication stages and obtain access to private data cause:

- Denial of Service attack. The agent overloads the host e.g. by consuming all network resources and then the host can't provide the expected services to the other agents.
- Eavesdrop. The agent can access sensitive information on the host e.g. the private key of the host, modify the security policy in order to obtain more access rights etc.
- Masquerade. The agent can pretend being someone else and therefore be objected to the wrong policy schemes.
- Violate non-repudiation. An agent can deny performing several actions to the node.

- Perform complex attacks. Here more than one agent co-operates in order to attack a host.

MISUSE OF AGENTS BY OTHER AGENTS

An agent can attack an other agent by changing agent's internal state, accessing/changing data of an agent (e.g. access the memory where agent keeps its own data), trapping an agent and changing its mission, stealing info, claim a false identity and in purpose damage agent's reputation, delay an agent in order to distract it from its goals etc

MISUSE OF AGENTS BY HOSTS

A host can have complete control over an agent. There for it can change his objectives, provide wrong execution and return wrong results, steal/change internal data e.g. electronic money or offers (if it is an auction agent), delete an agent or suspend it for enough time so that the operations the agent wanted to perform are not valid any more or have no meaning. E.g. an auction agent that has missed an auction can't really fulfill its goal. We mentioned above the main threats that exist in an agent-based infrastructure. Also all above mentioned security breakouts are performed when an agent is visiting a host. Contrary to popular belief agents don't transport themselves to the next host. So in any case the agent relies on the agency to transport its code safe and secure to the desired host!

MISUSES BY THE UNDERLYING NETWORK INFRASTRUCTURE

Threats exist also while the agent traverses the network from host to host. One external attacker could perform all kind of attacks such as agent deletion/alteration/copy & replay/stealing etc. in a network host. By misconfigurations the user that runs the agency allows others to access and modify the files that are stored on disk e.g. the policy files. Then another user could easily change the policy file and allow his agents to execute. Such kind of attacks can't be predicted by the designer of the agent platform and are also out of the scope of this paper.

In response to these threats and to assist Organisational agencies in mitigating associated risks, Information TS requires that agencies take steps necessary to initiate an enterprise-wide approach to:

- □ Ensure secure interactions between and among governmental agencies take place within a shared and trusted environment;
- □ Ensure secure interactions between and among security partners, external parties and general agencies utilize a common authentication process, security architecture, and point of entry;
- □ Prevent misuse of, damage to, or loss of IT hardware and software facilities;
- □ Ensure employee accountability for protection of IT assets; and
- □ Prevent unauthorized use or reproduction of copyrighted material by public entities.

III. DEALING WITH SECURITY RISKS

Having presented the threat model we will try here to see how we can deal with these problems. Traditionally, when talking about data security usually three security objectives are identified: confidentiality, integrity, and availability. To better suit the needs of electronic security management with all its legal aspects more security objectives have been identified. The most important one is accountability. In such a way the four main security requirements to be satisfied are:

- ***Confidentiality***
- ***Integrit.***
- ***Accountability***. **If the accountability of a system is guaranteed, the participants of a communication activity can be sure that their communication partner is the one he or she claims to be. So the communication partners can be held accountable for their actions.** Agents and platforms should audit their activities and be able to provide detailed info for debugging or security purposes. Every action should be uniquely identified, authenticated and audited.
- ***Availabilit***

Agent should also be able to monitor their services and actions in order not to be driven to endless loops. Several approaches have been developed in order to minimize security risks. We will not examine those approaches; instead we will focus on cryptography, signing and policy.

IV. THE SECURITY RISK ANALYSIS PROCESS

The methodology of security risk analysis also comprises a number of basic steps. These differ between authors, but in general include:

- *Asset Identification*. The asset identification phase should identify the resources that require protection. These will include: hardware, software, data, documentation, and computer services and processes. Financial values can be readily applied to some of these assets, but others are more difficult to price.
- *Vulnerability Analysis*. Having listed the assets of a computer system, the next stage is to determine their vulnerabilities. This stage is more difficult than the first, as it requires a degree of imagination to predict what damage might occur to the assets and from what sources [2]. The general aims of computer

security are to ensure data secrecy, data integrity and availability. System vulnerabilities are situations that could cause the loss of any of these qualities. A thorough understanding of the threats to the system is required if all the vulnerabilities are to be identified. Methodical and structured approaches are required if threat identification and vulnerability analysis is to be successful.

- *Likelihood Analysis*. The aim of likelihood analysis is to ascertain how often the system will be exposed to each of the vulnerabilities identified. Likelihood relates to the current security safeguards and the environment in which they are applied. Estimating the probability of exposure to a threat can be difficult. Sources of data for this estimation include: operations logs, local crime statistics and user complaints.
- *Countermeasure Evaluation*. All the analysis so far reflects the current situation. If, from this analysis, it is determined that the projected loss will be unacceptable, new or alternative countermeasures will have to be investigated. New controls will have to be identified, and their effectiveness evaluated.

I Cryptography

The basic purpose of cryptography and specifically encryption is to guard sensitive data against unauthorized access from non-intended recipients. Encryption techniques are used to acquire features such as:

- Data confidentiality and secrecy since all messages have to be decrypted in order to process the enclosed info.
- Data integrity, because if the cipher text has been tampered it won't be possible to decrypt correctly the original message
- Authentication. Non-repudiation. Public key technology can provide non repudiation of the recipient and its actions.

One-way hash functions, symmetric and public key cryptography belong to encryption techniques. Encryption is used in order to strengthen security. The user should be able to chose from a wide variation of encryption algorithms and have the ability to implement his own and make it available to other users.
The Component database that exists in our architecture ensures exactly that. Encryption guarantees authenticity, integrity, and secrecy of data and communication.

II Credentials and Authentication

Because agents are programs, they are intangible and live in a virtual world, we connect the trust model of such an infrastructure with the trust model of real world in order to make security critical decisions. That basically means that since every agent acts on behalf of a user or generally an entity we check to see if we trust that entity and indirectly trust the agent. The connection between those two worlds, the virtual one of agents and the real one is done via the digital certificates. A digital certificate is an object (file or message) signed by a certification authority that certifies the value of a person's public key. X509 [3] certificates of the International Standard Organization are the most popular, so we also adopt them in our design.

An agent is signed by one or more entities. Those entities can be either the creator of the code, the user that dispatched the agent (usually this is also the creator), a place of a host and generally any entity that holds a valid certificate. Signing an agent guarantees that i) the creator is the one claimed by the agent, ii) agent's code (at least the signed part) has not been tampered by a 3rd party during transportation. Signing doesn't guarantee that the agent will execute correctly (safety). Furthermore one place can encrypt the agent with the public key of the destination place (only the destination place has the private key to decrypt the agent), protecting in this way the agent while it traverses the net until it reaches the final destination. In order to ensure secure external communication we don't use any home grown solutions but instead we use the SSL (Secure Socket Layer) protocol [4]. TLS standard (Transport Layer Security) [5] is also another option.

Credentials also touch indirectly the "malicious host" problem. Since each place (or at least each agency) has its own certificate there is proof that this agent is mapped to a legal user who bears responsibility of the behaviour of the agency. An agent (based on a trusted host) prior to transportation can get next host's credential and decide whether to migrate and what to compute on the specific host. Furthermore it can ask the place to sign the results with its private key, so it can prove that those results were obtained during the execution on that specific place (repudiation problem). Non-changing parts of the agent should be signed for maximum protection.

III Means of authentication and authorization

In this section the actual methods or processes that are used to authenticate the identities of users are discussed. The authorisation of the user to gain access to services or resources can be carried out in a system after the user has been authenticated and his identity is resolved through the use of access control lists (ACL), to determine what the particular user is authorised to do. Authorisation is thus at maximum as accurate and correct as the process of authentication. A mechanism like the SPKI could be used, to avoid authentication of the user, but to still provide a reliable authorisation.
Some mechanism for implementing mobile authentication and authorization are:

- **Passwords.-** Passwords associated to user names (something that the person knows) are a simple way

of authentication. There are several authentication schemes that make use of passwords in combination with some other factor. A simple extension of passwords is one-time passwords.

- **Password with a token.-** Passwords can be used in combination with some physical object (something that the person owns). This concept has been extended with the use of integrated circuit cards (ICC) or smart card. A challenge and response method is used in authenticating the user. 'Synchronous one-time passwords' [5] is another similar technique.
- **Biometrics.-** Biometrics authentication techniques include fingerprint recognition, retinal scanning, hand geometry scanning and handwriting and voice recognition [5]. These techniques are all based on the physical properties of a person (something he owns / is).
- **Digital Signatures.-** When a PKI is put in place, digital signatures can be used to authenticate users. The following sequence of actions has to be carried out in order to authenticate a user by his digital signature:

    - The user requests access to the service or system
    - The system generates some data for the user to encrypt using his private key. Then the data is sent over to the user.
    - The user concatenates the data received from the system and a time stamp and encrypts the whole sequence. (N.B. It is a good practice that e.g. a time stamp is concatenated to the data, so that the data to be encrypted cannot completely be decided by some untrusted party. This is to avoid the possibility of a 'Chosen plain-text attack' as described in [12].) Then the encrypted data (the cipher text) is sent back to the system. Along with the encrypted data a link to the certificate (or the certificate itself) of the user is sent.
    - The system decrypts the received information with the public key of the user, found in the certificate.
    - The system verifies that the decrypted information is composed of the originally generated data and a valid timestamp. If this seems to be OK, the system has successfully authenticated the user.

Properties of a good authentication and authorization mechanism

This section lists the properties that a good identity authentication and authorization mechanism should possess. Some of the features listed are in contradiction of each other, but mostly it should be possible to reach an acceptable Level of compliance with each of the criteria:

- **Correctness**- The results of each individual instance of authentication or authorization carried out should be correct. If it is possible to authenticate the user, the result should always be that either it is found, that the user is who he claims or he is found to be a fraud. Based on this perfectly correct authentication it is further possible to authorize the user to access those services and resources that defined to be accessible for him or to the group or groups he belongs to. In practice it is impossible to get an absolute certainty in authentication. Only a reasonable Level of certainty can be gained.
- **Possibility to anonymity and privacy**- Identity authentication should only be done when absolutely necessary. Whenever authorisation is possible without the user's identity being revealed, it should be done that way.
- **Speed**- The process of authentication should be fast. The user shouldn't have to wait for the result for more than a second does or two.
- **Attack resistance**- The perfect mechanism of authentication should be resistant against any known or unknown types of attacks.
- **Inexpensiveness**- The mechanism shouldn't require extensive investments from either the users or the authenticators.
- **User friendliness**- The mechanism should produce as little overhead to the user as possible. It should also be as easy to use and understand as possible. In the optimum situation the user doesn't have to perform any actions in order to become authenticated. The user shouldn't be forced to carry around any extra equipment, magnetic or smart cards, lists of passwords or other physical objects in order to use the system.
- **Universality**- It should be possible for the user to use the same means or method of authentication in all services and everywhere.

IV Access Control Checks

Having successfully identified the agent is only the first step. Trust in the agent's credentials doesn't guarantee that it will behave legitimate nor execute correctly. Thus we monitor and authorize every call it makes to platform's resources. Any access to any resource e.g. network, file, system configuration etc is subject to an access control check. Therefore we need a policy and an enforcement manager to make sure that our policy is enforced. With this second level of check we provide fine-grained control customized per user or group. As users perform various activities not all of them have the same rights. The security is based in protection domains of Java. Those protection domains are defined by the internal agent id (not immutable) and/or by the signer(s) of the agent code (immutable). We can even require a combination of user identities in order to allow an agent to perform a task. A flexible policy scheme guarantees

exactly that. Although this second level provides some extra and selective security we understand its limits.

V. THE SECURITY ARCHITECTURE PROPOSED:

Secure systems and security applications produce their own special challenges to usability. Since secure systems have been traditionally difficult to use, some people are under the impression that usability and security are innately in conflict. Our model will have to provide a powerful tool for the definition of security policies, but power (that is expressivity) is useless if the user cannot easily figure out how he can employ it. For those reasons, the first requirement of our model is to provide a Graphical User Interface (GUI) that hides the complexity to the user. The GUI should provide the user with:

1 a way to define and modify the security policy
2 a tool to check the policy behaviour
3 a help to "debug" the security policy

Because information technology security planning is primarily a risk management issue, the architecture model, his policy and its associated standards and guidelines focus on the creation of a shared and trusted environment, with particular attention to:

- Common approaches to end-user authentication;
- Consistent and adequate network, server, and data management;
- Appropriate uses of secure network connections;

Approaches that try to incorporate security after the design phase have been proven to fail. The security architecture (Figure 1) for mobile agent systems tries to incorporate all above solutions to the threat model presented before and also to be as open as possible in order to integrate easily future solutions.

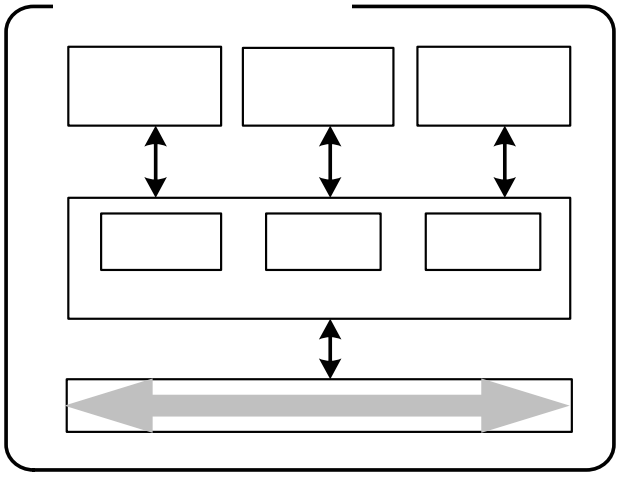

Fig 1.- **. Secure Agent Platform Architecture**

VI. CONCLUSION

Security architecture for agent based systems has been presented. This defensive model of design is focused on designing agent systems to be secure from the scratch. Adding security after the design phase has been shown to be difficult, expensive and inadequate. Security is not an explicitly called service and its treatment as such imposes further security risks in the infrastructure. We have showed that benefits such as simplicity, scalability, flexibility, interoperability, performance and safety have been addressed successfully. The components of the architecture have been analyzed and explained. Per identity/place security and customization as well as the rapid service creation is the main driving force for next generation mobile agent systems. In the future we intent to advance our approach. Our architecture tries to identify and prevent possible malicious agents. For the moment it can't handle collaborative attacks. Taking into account the tools provided (e.g. audit log, encryption tools, etc) one could implement stationary agents (guards) that reside on a place and based on intelligent internal strategy react to environment changes and try to track and eliminate collaborative attacks. Those guards could also work in collaboration thus providing a higher level of security to a number of hosts. As agent technology evolves and becomes more sophisticated a co-operative security infrastructure could be developed and deployed

VII. REFERENCES

[1] G. Santana and D. Higuera , A mobile user authentication protocol for personal communication networks, (Draft) Submitted to wireless and communications (WOC2001) , June 27-29,2001 Banff, Canada. IASTED International Conference

[2] V. Boyko and P.MacKenzie, Provably Secure Password – Authenticated Key Exchange Using Diffie-Hellman, Eurocrypt 2000, September 8 2000.

[3] William T. Polk, Bridge Certification Authorities Connecting B2B Public Key Infrastructures.

[4] M.J. Beller, et al, "Privacy and authentication on a portable communication network", IEEE Journal on Selected Areas in Communication, vol.11, August, 821-829, 1993.

[5] Liu Jianwei, Wang Yumin, "A User Authentication Protocol for Digital Mobile Communication Network", IEEE 1995.

[6]Internet Engineering Task Force (IETF), Request For Comment (RFC)2527, Internet X509 Public Key Infrastructure Certificate Policy and Certification Practices Framework, March 1999.

[7] Internet Engineering Task Force (IETF) , Internet Draft, Internet X509 Public Key Infrastructure Qualified Certificates Profile, August 2000.

[8]General Services Administration Access Certificates for Electronic Services (ACES) Certificate Policy, 3 September 1999.

[9] American Bankers Association (ABA) Trust ID Certificate Policy, December 13, 2000

[10] X509 Certificate Policy for the federal Bridge Certification Authority (FBCA), Draft Version 1.9, 27 March 2000