



Security in Mobile Agents



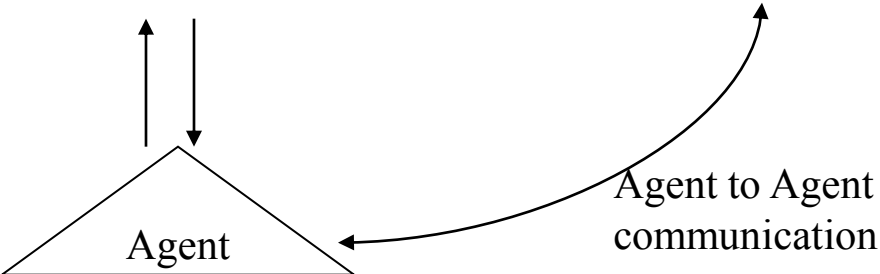
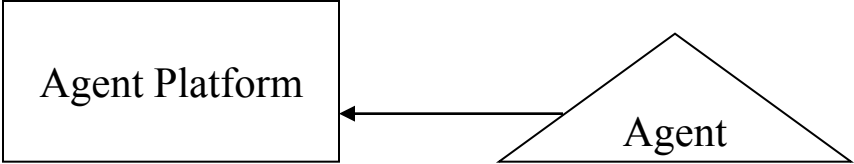
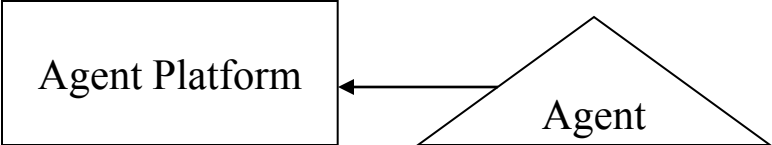
OUTLINE

- Introduction
- Advantages of Mobile Agent paradigm
- Applications of Agents
- Security Issues
- Classification of Security threats
- Security Measures
- Examples of Mobile Agent Systems

Introduction

- An agent is an autonomous software entity that can suspend its execution, transfer itself from one networked host to another and resume execution on the new host.
- They are programmed to perform certain “tasks “.In the process of performing the task , the agent may traverse numerous Hosts. The host provide an execution environment for the agent. The dispatcher of the agent is considered the Home platform
- A paper “*Is it an Agent, or just a Program? : A Taxonomy for Autonomous Agents*” by Stan Franklin and Art Graesser gives various classification of agents.

Simple Model of an Agent






Advantages of an Agent System


- Reduces network load and latency – There is usually no transmission of intermediate result. This conserves the network bandwidth.
- Asynchronous – Since the agents are autonomous, the mobile device that dispatches the agent need not be connected all the time.
- Fault Tolerant – If one of the host is down, the agent can be transferred to another host for execution. The agents can be programmed to adapt dynamically to the network conditions.
- Can be customized according to the needs.
- Can be deployed in a heterogeneous environment, as only the execution environment is of concern not the specifics of the Host Platform.

Applications of Agent Systems

Agents are used in numerous areas of information management, telecommunication systems, parallel processing, on-line auction, etc.

- E- Commerce – Agents act and negotiate on behalf of the user. Example: Auctions, service negotiations, etc.
- Personal Assistant – acts like a remote assistant. For example, can search and book airline ticket depending on a search criteria.
- Secure Brokering - Mobile agents meet and collaborate on mutually agreed secure host.
- Distributed information retrieval - Agent are dispatched to remote information sources and they can perform extended searches not restricted by working hours or connectivity.
- Information distribution – works on the push model. Software distributor may send software updates and versions to the user.

- 
- Parallel processing – agents can execute concurrently in a distributed system. A single task can be decomposed amongst multiple agents.
 - Network Security testing - Agents can be used in network intrusion detection.
 - Database searches – It is efficient to send out an agent to perform a specialized search on a large database than to move big chunks of data to the client using up enormous amount of bandwidth.



Security Issues in Agent Systems

The autonomous and mobile nature of the agents introduces new complexities and security issues such as

- In E-Commerce scenario, Agent may be carrying sensitive information like Social Security Number or Bank Account details. Agents must be secure and tamper-proof, and must not reveal information inappropriately.
- The Host platform should provide a safe environment for the agent to execute.
- A malicious agent may attack the Host and access sensitive data or tie up inordinate amount of resources causing Denial of Service to other applications or agents.
- Since agents traverse multiple hosts trusted to different extents, implementing any security measure is complicated.

Classification of Security threats in an Agent System

- Agents attacking Hosts - Malicious agents can steal or modify the data on the host. Lack of sufficient authentication and access control mechanisms lead to these attacks. If resource constraints are not set, they can also commit Denial of Service(DoS) attacks by exhausting computational resources and denying platform services to other agents.
- Hosts attacking the Agents – A malicious host can attack the agent, by stealing or modifying its data, corrupting or modifying its code or state, deny requested services, return false system call values, reinitialize the agent or even terminate it completely. It can also masquerade the agent by delaying the agent until the task is no more relevant. The Host may also analyze and reverse engineer the agent.
- Malicious Agent attacking another agent – A malicious agent may invoke public methods of another agent to interfere with its work.
- Attack by other entities – Some other entity in the network may manipulate or eavesdrop on agent communication.

Security Measures

Security in Agent System is based on the principle of trust. A set of security policies and protocols establish the trust relationship between the entities.

It is assumed that the agent trusts the Home platform that dispatches it.

Agent attacking the Host environment

- Traditional methods such as authentication, access control, sand-boxing techniques, cryptography can be used to secure the Host.
- ***Authentication and access control mechanisms*** – This is the first line of defense against a malicious agent. If the Host can authenticate the agent and in turn the device that dispatched the agent, it can apply authorization and access control.
- ***Safe Code Interpretation*** – Due to the necessity for the agents to run on heterogeneous computer, interpreted scripting or programming languages are used. This produces intermediate code that is executed by a virtual machine that sits on top of the native processor and OS. This virtual machine can enforce additional security.

- ***Path Histories*** - An agent could reach the host by making a number of hops. During this transit a malicious host could have morphed the agent into a malicious agent. By storing the log of the travel of the agent, the current host can determine the route taken by the agent. . Each host platform to which the agent travels to, appends a signed entry to the path. This entry indicates the hosts identity as well as the identity of the next host the agent intends to visit. The platform has to judge by looking at the log if the previous platforms can be trusted.
- ***State Appraisal*** – The author of the agent supplies a state appraisal function called maximum function. This function calculates, depending on the state of the agent, the maximum set of permissions to be granted to the agent. This function is packaged together with the agent. The user/owner of the agent also supplies another state appraisal function called the request function. This calculates the permissions the user wants the agent to have during execution. The host platform uses these state functions to verify the correct state of the agent and hence determines the privileges to give to the agent depending on its state. This ensures that the agent has not turned malicious due to alterations of its states.


Host Platform attacking the Agents

- Providing security against the attacks by the host is difficult due to the fact that the host needs to have the full knowledge of the code and the state in order to execute the agent. Traditional mechanisms are not sufficient to protect an agent from the attack of malicious hosts.
- **Mobile Cryptography** – Cryptography is used to maintain code and data privacy and integrity. Both code and data can be encrypted.

Encrypted Functions – For the host to execute the agent, it has to have full control over the code. As prevention, the function of the agent is encrypted according to some conversion algorithm. This encrypted function is implemented as a cleartext program. Even though the host is able to read the program it won't understand what the program does i.e. the “program's function”. The disadvantage of this technique is finding the encryption schemes to transform the arbitrary functions.

Encrypted Data - The agent data is encrypted and sent to host for computation. The data that the agent needs for its computation may have to be decrypted again and again at the host platform. For this reason, the agent will have to carry the decryption key making it that much vulnerable.

- ***Obfuscated code*** – A “blackbox” agent is generated from the agent specification wherein the agent’s code and data cannot be read or modified. Only its input and output can be observed. The algorithm that creates the agent is called “mess-up or obfuscating algorithm”. To prevent dictionary attacks the algorithm, that converts the agent specs into an agent, uses some random parameters. These parameters allow creation of number of different agents out of the same specification. The agents differ in code and data representation but give the same results.
- ***Secure Routing*** - An agent can be programmed to have a routing policy such that it migrates only to certain servers. Since a malicious host can tamper with the agent’s itinerary and also computation results, which can propagate, some fault tolerance is needed to ensure that the agent reaches its destination and perform its job correctly. Replication and voting can be used to achieve fault tolerance. The agent is replicated at each stage and run on hosts. The results from these computations are compared (i.e. voted). Then the correct result is sent out as output.



• ***Detecting attack using Dummy data*** –In this technique, dummy data items called *detection objects* are used. This dummy data is stored in the database of the agent and it will not be modified while the agent performs its functions. After the agents return, if the detection objects have not been modified, then one can have reasonable confidence that legitimate data also has not been corrupted. This technique requires that the dummy data should not adversely affect the results of the query.

• ***Using Trusted Hardware*** - This technique uses tamper proof trusted hardware to encapsulate the entire agent execution environment in which the agent executes, thus isolating the agent from the malicious host. The whole agent is not visible to the host environment. The agent in this system will interact with the Host environment through messages. Each Host in the Mobile Agent System is equipped with this hardware. The hardware can be in form of PC Cards, Smartcards, Integrated Circuits, etc. PC Cards are powerful and allow the whole agent code to be loaded into the card. Smartcards are limited in their capabilities. Only a part of agent code can be loaded on the card. The agent carries rest of the code along with it. The code that the agent carries is encrypted.

Examples of Mobile Agent Systems

- TACOMA(Tux) - Mobile Agent System. Operating System Support for mobility. Tromsø and Cornell Moving Agents (TACOMA) is a joint project between the University of Tromsø, Norway and Cornell University,USA. It is primarily focused on providing operating system support for agents.
- Telescript- developed by General Magic, includes an object oriented, type-safe language for agent programming.
- Agent TCL is a mobile agent system created at Dartmouth College. agents are written in the Tool Command Language (Tcl), which is an embeddable scripting language that is highly portable and freely available.
- Aglets is a Java based system developed by IBM. Agents are called aglets in this System.

References

[Che] Mobile Agents: Are they a good idea? , David Chess, Colin Harrison, Aaron Kershenbaum: <http://citeseer.nj.nec.com/chess95mobile.html>

[Funf] Protecting Mobile Web-Commerce Agents with Smartcards , Stefan Funfroken, Dept of CS, Darmstadt Univ of Tech. , Germany :
<http://citeseer.nj.nec.com/unfrocken99protecting.html>

[Hohl] Time Limited Blackbox Security:Protecting Mobile Agents From Malicious Hosts, Fritz Hohl:
<http://citeseer.nj.nec.com/hohl98time.html>

[Kar] Network Security Testing Using Mobile Agents, T. Karygiannis, karygiannis@nist.gov, NIST:
http://csrc.nist.gov/mobilesecurity/Publications/Agents_PAAM98.pdf

[Mea] Detecting Attacks on Mobile Agents, Catherine Meadows, Center for High Assurance Computing Systems, Washington DC:

<http://citeseer.ist.psu.edu/meadows97detecting.html>

[Mins] Cryptographic support for Fault-Tolerant Distributed Computing, Yaron Minsky et al : <http://citeseer.nj.nec.com/minsky96cryptographic.html>

[Ous] The Safe-Tcl Security Model, J.K.Ousterhout et al, Tech. Report SMLI TR-97-60, Sun Microsystems,1997:

<http://citeseer.nj.nec.com/ousterhout97safetcl.html>

[Sta]Is it an Agent, or just a Program? : A Taxonomy for Autonomous Agents, [Stan Franklin](#) and [Art Graesser](#) , [Institute for Intelligent Systems](#), [University of Memphis](#) : <http://www.cs.memphis.edu/~franklin/AgentProg.html>

[Tom] Protecting Mobile Agents against Malicious Hosts, Tomas Sanders and Christian Tschudin. Proceedings of Workshop on Mobile Agents and Security, number 1419 in LCNS, pages 44-60, 1997:

<http://citeseer.nj.nec.com/sander98protecting.html>

NIST Special Publication 800-19 – Mobile Agent Security:

<http://gunther.smeal.psu.edu/2276.html>

A Sanctuary for Mobile Agent, Bennet S. Yee:

<http://citeseer.nj.nec.com/142540.html>

Mobile agent security, *Niklas Borselius, Mobile VCE Research Group, University of London*: <http://citeseer.nj.nec.com/547211.html>

Agent Tcl: A flexible and secure mobile-agent system (1997) Robert S. Gray

<http://citeseer.nj.nec.com/12688.html>

Aglets - <http://www.trl.ibm.com/aglets/>

Mobile Agent Security and Telescript,

<http://citeseer.ist.psu.edu/tardo96mobile.html>