Software Technologies

Mobile Code

Sebastian Fischmeister
fischmeister@softwareresearch.net
University of Salzburg

Security Overview

- Elements in a security model:
  - host
  - user
- Traditional computer systems
  - host is trusted
  - user is untrusted
- Mobile agent systems
  - host is trusted / untrusted
  - user is untrusted / trusted

Security Overview

- Mobile agents extend the traditional view. Users can be trusted and the host may be malicious.
- Example:
  - the mobile agent search cheap hotels
  - the owner of the host wants to boost his sales
  - the owner modifies the host to attack the agent → always choose his offer
- Example II (real world):
  - you have a packet
  - you give the packet to someone else, who takes it into his home and locks you out
  - how do you make sure, he doesn't do anything illegal with the packet?

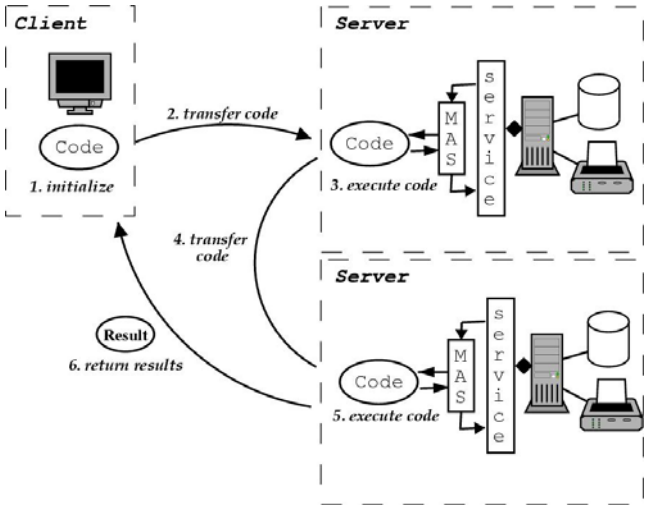
3

Security Overview

- **Confidentiality:**
  - secret information should be kept secret
  - agent contains information bought at other locations
- **Integrity:**
  - altering data must be detected
  - changing the best price, the agent has found so far
- **Authentication:**
  - map the real identity to the identity within the authorization system
  - an agent claims to be a service to gain access to other agents
- **Authorization:**
  - is the object allowed to perform the action
  - a role is not allowed to communicate with other agents, but finds a way to do so
- **Auditing:**
  - keeping track of the system
  - an agent misbehaves, this should be logged

4

Mobile Agents Revisited



5 © 2004 Sebastian Fischmeister

Mobile Agent Security Problems

- Masquerading
  - Agent poses as another agent to gain access to services or data at a host
  - Host assumes false identity in order to lure agents
- Denial of Service
  - Agents may attempt to consume or corrupt a hosts resources to preclude other agents from accessing the host's services
  - Hosts can ignore an agent's request for services or access to resources
- Unauthorized Access
  - Agents can obtain access to sensitive data by exploiting security weaknesses
  - Agent interferes with another agent to gain access to data

6 © 2004 Sebastian Fischmeister

Mobile Agents Security Problems

- Eavesdropping
  - With agents that are interpreted, the host can inspect their internal algorithms and data, such as the maximum price the agent's owner is willing to pay for item X
- Alteration
  - Hosts can change an agent's internal data or results from previous processing to influence the agent
- Repudiation
  - After agreeing to some contract, an agent can subsequently deny that any agreement ever existed or modify the conditions of the contract

7

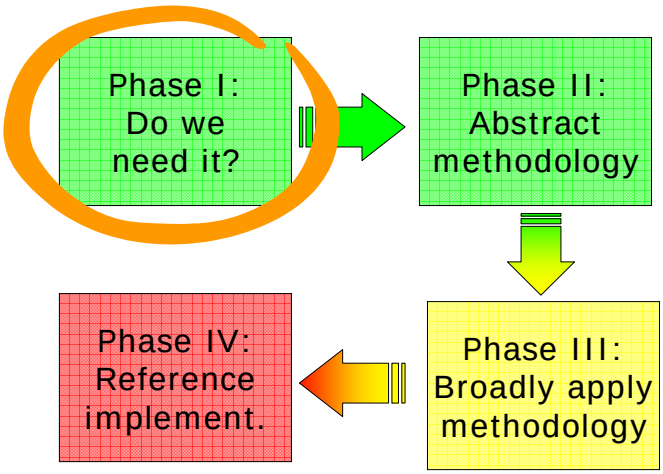
Example Methodology

Overview

- A security test, done in 1998
- We wanted to check the security system of available agent systems
- **Since then, many things have changed !**

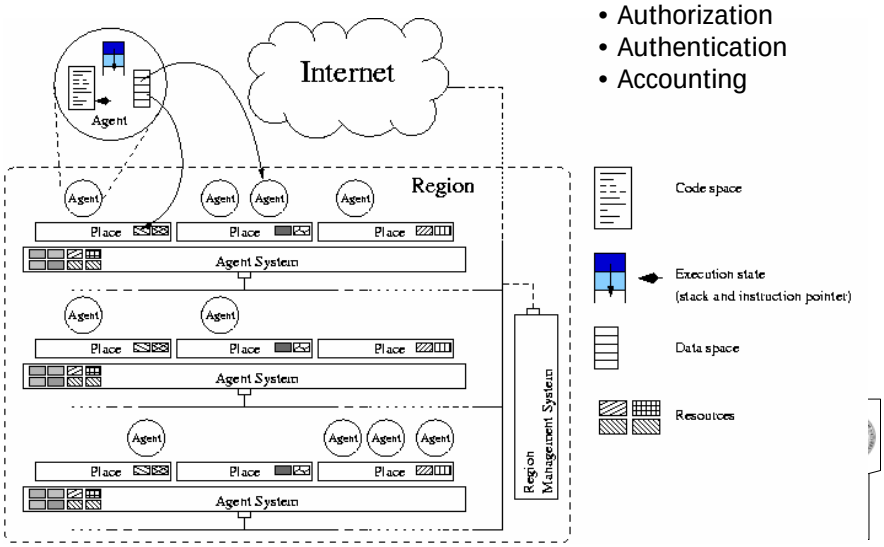
9

Context of the work



10

Model



11

Aglets

- Created by IBM, now open source (GPL)
- Evaluated 1.1b2

Model	Aglets
Mobile agent	Aglet
Place	Context
Place resources	Internal object
Agent system	Tahiti
System resources	Internal object
Region	missing

12

Aglets Security

- **Code repository attacks**
  - Use reflection classes
  - Create an exception, trace the stack, new classes
  - ➔ find potential holes *(static, public)*
- **Security policy attack.**
  - PolicyFileReader.getAllPolicyDB() *(public static)*
  - Permission *(public constructor)*
  - PolicyDB.add() *(public)*
  - ➔ the attacker gains the privileges of the user running the JVM
- **GUI attack**
  - Glitches in the implementation (window DOS)
  - ➔ lock down the graphical console

13

Jumping Beans

- AdAstra Engineering
- Evaluated version 1.1

Model	Jumping Beans
Mobile agent	Mobile application
Place	Agency
Place resources	Internal object
Agent system	*Agency*
System resources	*Internal object*
Region	Server

14

Jumping Beans Security

- **GUI attacks**
  - GUI in a separate thread *(windows are left open)*
  - Open a window of the size of the screen
  - ➔ no close operation, so shutdown the whole VM
- **Runtime system call attacks**
  - Incomplete implementation of SecurityManager
  - ➔ among others *System.exit()* works
  - Side effect, bypasses persistence manager, so the recovery mechanism is ineffective

15

Grasshopper

- GMD Fokus and IKV++
- Evaluated 1.2.2.3

Model	Grasshopper
Mobile agent	Service
Place	Place
Place resources	Agents
Agent system	Agency
System resources	Agents
Region	Region

16

Grasshopper Security

- **Policy attack**
  - Policy class is a singleton
  - Constructor is defined *public*
  - ➔ overwrite the policy, policy is lost
- **Trusted code base attacks**
  - Security manager uses trusted classes
  - *javax.swing.InternalFrame* (setCloseOperation, setClosed)
  - ➔ exit the JVM
- **GUI attacks**
  - checkAwtEventQueueAccess impl. missing
  - Send an “Alt-F4” event
  - Sniff for confirmation dialog and send correct response
  - ➔ exit the JVM
  - ➔ complete control of the GUI

Conclusions & Future work

- 100% success rate **(!!)** ➔ we need a general security testing methodology
- Obvious and crude mistakes (leaving security related methods empty)
- Recommendation:
  - firewall your systems
  - don’t tell anyone you’ve got a system running

- **Future work:**
  - Refine general attack model and testing methodology
  - Retest commercial systems using the defined testing methodology