Attribute Function: an enabler for effective inexpensive application-specific security decisions

Konstantin Beznosov
September 20, 2003
Overview

- overview of security enforcement in distributed applications
- problem motivation
- Attribute Function
- research plans
  - hypothesis
  - methodology
overview of security enforcement in middleware
decision-enforcement paradigm

- Access control
- QoP (secrecy, integrity)
- Audit

Subjects

Reference Monitor

Enforcement Function

Authorization Decisions

Decision Function

Objects
Existing Solutions Classification

<table>
<thead>
<tr>
<th></th>
<th>Decision Function</th>
<th>Enforcement Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>AD</td>
<td>AE</td>
</tr>
<tr>
<td>Middleware</td>
<td>MD</td>
<td>ME</td>
</tr>
</tbody>
</table>

Application space

Middleware Security

Middleware Space

Object
problem motivation
conflict of interests

Vendors:
stable
infrequently
changing
platform
security

Users:
security
decisions
based on
application-specific factors
application-specific factors

- Certain characteristic or property of an application’s resource
  - Produced, modified and processed in the course of normal application execution

- Examples
  - Bank account’s holders and their ranks
  - Phone numbers of telecom customer accounts
    - 5,000 changes/day with $10^6$ subscribers
main objective

Keep middleware security generic and yet allow for application-specific security policies

Approach: additional level of indirection 😊
- Separation of concerns
ADME – application decides, middleware enforces

- Could be inefficient on expensive to activate objects
- Vulnerable to deny of service attacks
- DF too complex for application developers to implement
Attribute Function
Proposed solution -- ADME/AF

- Object Attributes
  + Advantages of ADME
  + Separation of concerns
    - EF – middleware vendor
    - DF – authorization vendor
    - AF – application owner
- AF’s input:
  - Information for identifying the target’s state
Discussion

- Only information known before the object is called
- Not for all middleware platforms
- Not for all policies

+ Better tradeoff in responsibilities
+ Does not require application to implement either DF or EF

- non-middleware platforms?
- non-security policies
Research Plans
**hypothesis**

the attribute function allows effective use of application-specific factors in security policy decisions without expensive coupling between the decision function and the application.
methodology

comparative analysis of AF-based designs vs.

traditional methods
- mixing security and application logic
- back-doors

emerging approaches
- tool-based weaving using Aspect Oriented Software Development (AOSD) techniques
what should it be compared on?

- performance
- expressiveness of the supported security policies
- costs of application development, deployment, and maintenance
- degree of the separation of responsibilities among application, middleware, and security developers
- other?
experiment design

- Alternative designs of decision/enforcement functions
  - Mixed security and application logic
  - Security logic modularized and weaved using AOSD techniques
  - ADME/backdoor
  - ADME/AF

- Sample access control policies that require application-specific factors

- Sample application
  - Depends on the platform

- Experiment platform candidates
  - EJB
  - ASP.NET
  - CORBA ORB
**performance**

How fast security decisions are made and enforced

- Decisions/second (throughput) for 1 client
- Throughput = f(|client population|)
expressiveness

What types of security policies and application-specific factors can be supported by the decision logic?
How much effort is required to develop, deploy, and maintain secure distributed applications?

- changes to the application logic
- changes to the security policy
- replacement of a security policy with a different type

Effort measures

- changed lines of code? :)
- other metrics?
separation of responsibilities

How many interdependencies exist among application, middleware, and security developers throughout an application life-cycle?
Summary

Hypothesis

- Attribute function allows **effective** use of application-specific factors in security policy decisions **without** expensive coupling between the decision function and the application

Methodology

- Comparative analysis of AF-based design vs. other designs based on performance, cost, expressiveness, and separation of responsibilities
Who will be doing it?

- Looking for
  - faculty, and
  - bright, energetic, and enthusiastic graduate students
- to collaborate on this and similar projects

- Contact Konstantin Beznosov at beznosov@ece.ubc.ca