OpenPMF

Integrated IT Security

Dr. Ulrich Lang, CEO
About ObjectSecurity…

• IT security expertise
• Consulting services/solutions in IT security.
• Security specialist for complex, heterogeneous, networked environments
  – Middleware: EJB, CORBA, .NET, XML Web services, CCM
  – Model-Driven Architecture (MDA)
  – Security mechanisms: PKI, PMI, Firewalls, …
• Evolved from University of Cambridge (UK) research, founded in 2000

WWW.OBJECTSECURITY.COM
Security solutions for blue-chip customers

• Clients
  - Deutsche Telekom
  - General Electric
  - Agilent Technologies
  - US Naval Research Laboratory
  - Twinsoft
  - European Commission
  - Artechhouse Scientific Book Publisher

• Partners
  - Thales
  - Lucent
  - Intracom
  - US Naval Research Laboratory
  - Fraunhofer Gesellschaft FOKUS
  - Various Universities (e.g. Cambridge, London, Paris, Lille, Berlin)
ObjectSecurity - IT Security Expertise

- **Our approach:**
  - Complete organization-wide approach from business imperatives through to technology solutions
  - Unified approach to security problems
  - A complete solution from policy creation to technologies
- **Benefits:**
  - Security at a lower cost and with less effort
  - Greater flexibility and customization
  - Higher assurance
- **We do this for systems where other commercial solutions do not exist**
ObjectSecurity’s Expert Know-How

• Security architecture, policy design, risk analysis, policy integration
• Security policy and technology effectiveness analysis
• Integration of security products
• Security technology evaluation
• Applied research and development
Some of our projects

- Security consulting, development, applied R&D
  - Very complex, distributed environments:
    - Air traffic management
    - Defense communications
  - Very specific, distributed environments:
    - Geographical information system
    - Mobile telecoms application platform
  - More typical distributed environments
    - Secure mobile stock trading system
OpenPMF

• In a nutshell:
  – Technology framework
  – Open source software (tool kit)
  – Integration as a commercial service

• Purpose:
  – Add good security to distributed systems
  – Make distributed systems security manageable

WWW.OPENPMF.ORG
Legacy systems create problems

*Large enterprises use many separate, incompatible components (often legacy)*
Last decade: seamless, enterprise-wide integration of services and data (e.g. Web Services, EJB, CORBA, .NET, CCM, DCE)
Adding security has been in isolation

*Last decade: protection of information and services increasingly important; mostly “island solutions”*
ObjectSecurity’s solution: OpenPMF

Seamless, customized integration of security

Policy

- Specific Application
- Contractor Data Access
- PDA Application
- Web Server
- Firewall
- Data Mining Machine
- Legacy Back-end Data Server
- Customer Data Server AS/400
OpenPMF Main Principles

- Apply the OMG Model-Driven Architecture (MDA) approach to security
  - PIM: technology-unspecific policy
  - PSM: technology-specific policy
  - Implementation: enforcement

- Separation of functional and non-functional aspects

- Separation of policy definition, storage, evaluation, enforcement

- Flexible composition of simple concepts

- Small, well defined modules (-> assurance) to:
  - Describe, obtain, process security information
  - Evaluate policy
  - Trigger actions
OpenPMF Architecture
Policy Definition Language (PDL)

- Technology-independent language
- Technology-independent identifiers:
  - Initiator, intermediate, target, operation, action
- Hierarchies
- Clustering
- Delegation: weak and strong
- Arbitrary execution of predefined functions possible, for example logging or notification
PDL example

policy /OS [*, *] {
    // Admin allowed to write policy, bank server allowed to obtain policy
    policy /OS/Bank [/OS/Bank/Admin, /OS/Bank/Server] {
        // Simple rule
        (initiator.name == /OS/Director) & (operation.name == create) &
        (target.type == IDL:Bank:1.0) : allow;
        // All clients in group /OS/Accounting are allowed to open the account
        (initiator.group == /OS/Accounting) & (operation.name == open) &
        (target.type == IDL:Bank:1.0) : allow;
        // List of operations
        (initiator.group == /OS/Accounting) & (operation.name == {deposit, balance}) &
        (target.type == IDL:Account:1.0) : allow;
        // Again a simple rule
        (initiator.name == /OS/Director) & (operation.name == withdraw) &
        (target.type == IDL:Account:1.0) : allow;
        // Strong delegation
        (client.speaksfor.name == /OS/Director) &
        (initiator.group == /OS/Accounting) & (operation.name == withdraw) &
        (target.type == IDL:Account:1.0) : allow;
    };
};
Policy Repository

- Stores the entire security policy
  - Technology-independent rules
  - Consistent
  - Centralised
  - Optimised
  - Hierarchical (for separation of duties)

- Based on OMG Meta Object Facility (MOF)
  - UML model for policy structure
  - Automatic generation of the repository and XMI interchange
Policy Evaluation

• Interprets security rules

• Efficient runtime representation instantiated
  – At application startup (online repository)
  – At compile time (for embedded systems)

• Evaluators make decisions based on technology-unspecific attributes
  – obtained from Transformers
  – comparison done by Transformers

• Technology-independent, but programming language specific
Transformers

• Obtain attributes from platform and security mechanism
• Transform specific information to abstract identities
• Operations for the comparison of selector and obtained attribute
• Transformers have to be implemented once per security mechanism & platform (extensibility!)
• High flexibility and extensibility
  – Transformer can obtain arbitrary information
  – Transformers can be stacked
Adapter

- Adapter calls policy evaluator
  - Trigger evaluation of policy
  - Execute decision: Grant or reject invocation

- Integration into call chain platform specific, e.g.:
  - CORBA: Portable Interceptors
  - CCM: Component Portable Interceptors (COPI)

- Adapter has to be implemented once per platform
Central Management

• Central management (via management daemon) reduces costs
  – Users
    • Identities, roles,…
  – Applications
  – Policies
  – Configuration
  – Logging and auditing

• Integration with directory services
  – Already existing information, e.g. about users, can be reused

• Intrusion detection & prevention daemon
Technology Integration

• Some security infrastructure needed
  – Public Key Infrastructure
  – Privilege Management Infrastructure (ATLAS)
  – Directory Services (LDAP) for user data
  – protocol for delegation & authorisation token transfer, e.g. Common Secure Interoperability v2 (CSIv2)

• Current version tested with:
  – CORBA and CORBA Component Model (CCM)
  – Firewalls
  – EJB/Java

• Future: Web services, .NET
Technology Integration

• IIOP Domain Boundary Controller
  – Allows secure usage of EJB, CCM and CORBA over the Internet
  – Protects servers without self defense
  – Integration with packet filter

• Clusters and Grids
  – OpenPMF allows secure sharing of resources and information
    • Prototype: Office computers as number crunchers at night
Technology Integration

• Multiple Independent Levels of Security (MILS)
  – Separated nodes with different security levels running in OS “partitions”
  – OpenPMF used to control information flow between nodes
  – Mainly used by military applications
  – Civilian use: Damage restriction
Building Blocks for Distributed Systems

- Cross-platform security integration
  - Web Services,
  - .NET,
  - Enterprise Java Beans,
  - CORBA
  - CORBA Component Model,
  - MDA security modelling
  - Security technologies (firewalls, PKI, Privilege Management Infrastructure)
SecureMiddleware

- Project that integrates OpenPMF with Qedo CORBA Components
- First model-driven, component-based, secure application development and integration platform in the world
- www.securemiddleware.org
Conclusion

• OpenPMF benefits:
  – Makes security in complex, heterogeneous, networked IT environments manageable
  – Central administration
  – Flexible policies and consistent policies
  – Reliable policy definition and enforcement
  – Across differing technologies: organisation-wide security policy
  – Integrated validation, optimisation, intrusion detection possible
  – Easy extension to incorporate new security technologies and policy features
  – The effort for development and operation is reduced