The Cassandra System: Distributed Access Control Policies with Tunable Expressiveness

650.650: SPECIAL TOPICS IN MEDICAL SECURITY AND PRIVACY
What is Cassandra

• **Role-based** trust management system
  - Policy specification language based on Datalog with constraints.
• Used on a distributed network platform
• Uses five special predicates to express a wide range of policies.
  - Ex) Role hierarchy, role delegation, separation of duties, cascading revocation, automatic credential discovery, and trust negotiation.
• Cassandra policies are expressed in a language derived from Datalog with constraints.
• Cassandra supports credential-based authorization (between administrative domains), and rules can refer to remote policies (for credential retrieval and trust negotiation)
Datalogs and Their Purpose

- Datalog is a query and rule language for deductive databases that syntactically is a subset of Prolog. ([http://en.wikipedia.org/wiki/Datalog](http://en.wikipedia.org/wiki/Datalog))
- Query evaluation is usually done using bottom-up strategies.
- Recursive queries and clean semantics make it advantageous to use instead of SQL
  - Also well-suited for transitive policy idioms such as delegation or appointment.
- Two extensions that have been made to Datalog
  - Allow object-oriented programming
  - Allow disjunctions
Datalog Queries

- **Datalog query** – a finite set of rules of the form:

\[
R_0(x_1,\ldots,x_k) : \leftarrow R_1(x_{1,1},\ldots,x_{1,k_1}),\ldots, R_n(x_{n,1},\ldots,x_{n,k_n})
\]

- Where each \( R_i \) is either an input or a defined relation name. Including built-in relations such as \( +(x,y,z) \) which means \( x + y = z \).

- **head of the rule** – \( R_0 \)
- **body of the rule** – \( R_1,\ldots,R_n \)
Datalog Query Examples

Find the streets that can be reached from \((x_o,y_o)\).

\[
\text{Reach}(n) : \neg \text{Street}(n, x_o, y_o).
\]
\[
\text{Reach}(n) : \neg \text{Reach}(m), \text{Street}(m, x, y), \text{Street}(n, x, y).
\]

Which part of the rule is the head, which is the body?

Find the time to travel from \(x\) to \(y\).

\[
\text{Travel}(x, y, t) : \neg \text{Go}(x, o, y, t).
\]
\[
\text{Travel}(x, y, t) : \neg \text{Travel}(x, z, t_2), \text{Go}(z, t_2, y, t).
\]
Inspiration and Similar Systems

- **PolicyMaker/KeyNote**
  - First trust-management system.
  - Designed to be minimal and analyzable
    - Allows for large amount of responsibility is placed on the calling application, including policy enforcement, cryptographic verification and credential gathering.
  - The policy engine replies with a string; in the simplest case, this may be “grant” or “deny”.

Inspiration and Similar Systems (Cont.)

- Oasis, policy specification language
  - Role based trust management system
  - Datalog based rules specify which credentials are prerequisites for role activation and deactivation.
    - Ex) A hospital receptionist may assign the permission to read a patient’s record to the patient without being permitted to read the record herself. Appointment can thus be seen as a generalization of delegation.
Inspiration and Similar Systems (Cont.)

- **RT**
  - Authorization in collaborative environments should be based on authenticated attributes of the entities rather than on public keys.
  - The credentials in these languages can be translated into Datalog. In RT’s youngest offspring, RTC\textsubscript{1}, credentials are translated into Datalog with constraints.

- **SD\textsubscript{3}**
  - Datalog-based trust system, which allows for predicates to use public keys for each user.
Cassandra’s Vision

• British National Health Service’s plan
• Provide pre-defined constructs applicable for most real-world situations.
  ○ Have no explicit provisions for standard policy idioms
    ○ Including role hierarchy, separation of duties, or delegation
• Specifically designed for authorization purposes, not obligation policies specifying the automatic triggering of actions.
• Simplicity, flexibility, support for delegation of authority, scalability, and state awareness.
Becker, M. and Sewell, P.

- Uses a GUI to control policies, permissions, and roles.
- Network access is allowed only through an interface
  - Acts as a protective layer around the shared resources
- 3-level infrastructure and policies including 310 rules and 58 parameterized roles.
How Cassandra Works (Cont.)

- Entities interact by sending requests through GUI.
- The requester may submit a set of credentials to the service that may support the request.
  - Ex) to receive a discount, the requester may need to submit a student’s ID.
- The access control engine uses evaluation engine to query the policy.
- An authorization decision is based off of logical deduction.
How Queries Work
How Cassandra Works: High Level

- Access control is role based (RBAC)
  - Roles and actions are parametrized
- Uses automatic trust negotiation and credential discovery
- Each entity must run its own copy of the Cassandra service
- ACL cannot be identity based.
  - Collaborating entities may be mutual strangers at first.
- Authorization is based on credentials
  - OCaml prototype uses credentials without PKI or digital signatures
How Cassandra Works: Low Level

- A policy specification language is defines system security policy
- Expressiveness of a policy specification language can be adjusted by selecting an appropriate constraint domain
  - The expressiveness of Datalog\(_C\) depends on the chosen constraint domain \(C\).
- The access control engine handles the requests by invoking the policy evaluation engine
- Policy evaluations may trigger queries of remote policies over the network through Query Projection and Answer Propagation.
A Datalog rule: \[ p_0(e_0) \leftarrow p_1(e_1), \ldots, p_n(e_n), c \]

To constrain issuer and storage location of credentials parameter is written \( \text{loc}@\text{iss}.p(e) \)

Ex) \( \text{Alica}@\text{UCam}.\text{canActivate}(\text{Alice, Student(Maths)}) \) is a predicate asserting that Alice is Maths’ student.

The entity \( E_{\text{loc}} \) on the network protects its resources with a Cassandra policy rule

- \( E_{\text{loc}}@E_{\text{iss}}p_0(e_0) \leftarrow \text{loc}_1@\text{iss}_1.p_1(e_1), \ldots, \text{loc}_n@\text{iss}.p_n(e_n), c. \)

A rule with an empty body, called a credential rule

- \( E_{\text{loc}}@E_{\text{iss}}p_0(e_0) \leftarrow c \)

Uses automatic trust negotiation and credential discovery through \text{canActivate()}, \text{auth.canActivate()}, and \text{Cert[role]}().

Ability to limit the validity of a role

- \( \text{CurTime()} < 1 < T \)
Special Predicates

- They are used by the access control engine to make authorization decisions for the four types of requests: performing an action, activating and deactivating a role, and requesting a credential.

  1. permits(e,a) indicates that the entity e is permitted to perform action a.
  2. canActivate(e, r) indicates that the entity e can activate role r.
  3. hasActivated(e, r) indicates that the entity e has currently activated role r.
  4. canDeactivate(e1, e2, r) indicates that e1 can deactivate e2’s role r (if e2 has really currently activated r).
  5. isDeactivated(e, r) indicates that e’s role r shall be deactivated as a consequence of another role deactivation (if e has really currently activated r).
  6. canReqCred(e1, e2, p(~e)) indicates that e1 is allowed to request and receive credentials asserting p(~e) and issued by e2.

- Allows Alice and Bob to activate the administrator role if their user roles have been activated

  canActivate(e,Admin()) ← hasActivated(e,User()), e ∈ {Alice, Bob}
Concepts and Optimization Techniques

- Constraint > Expressive $\rightarrow$ policies and queries possibly uncomputable.
- Trade-off between Expressiveness vs. Decidability
  - Solution: Constraint compactness
    - Restricts domains to guarantee the termination of queries
  - Static ground analysis is used to restrict policies - Grounds variables.
- Standard SLD top-down resolution algorithm, known from Logic Programming (Prolog), is not suitable
  - May run into infinite loops even when the fixed-point semantic is finite.
- Cassandra uses a modified version of Toman’s memoing algorithm for evaluating constraint extensions of Datalog.
- Query Projection and Answer Projection
Query Algorithm Approaches

- **Bottom-up algorithms** are closely based on the fixed point semantics
  - Starts from basic facts, new derived facts are iteratively added until a fixed point is reached
  - The model can thus be pre-computed and reused.

- **Top-down algorithms** are based on some form of resolution and are goal-oriented
  - This information can be used to prune the search space for efficiency.
  - Disadvantages are that answers are not pre-computed, and standard top-down query evaluation algorithms such as Prolog-style SLD resolution are often not termination-complete
Query Evaluation

• Bottom-up evaluation is not suitable for Cassandra for several reasons:
  o The constraint domain $C_0$ may calls dependant on the environment
    o Ex) Current time $\rightarrow$ cannot be pre-computed.
  o Rule bodies can refer to remote predicates requiring a distributed form of bottom-up evaluation.
  o Requests made to the access control engine can modify policies.
    o Ex) a successful role activation request adds a hasActivated credential rule to the policy. The model would thus have to be re-computed after every such request.
Case Study Example

- The nation-wide Electronic Health Record (1998)
  - Broken into implementation phases extending into 2010
- Electronic Health Record (EHR)
  - Cradle-to-grave linking Europe, the United States, Canada and Australia clinical information across the entire health system
- Services Provided
  - The Care Record Service (CRS), or the Spine, containing the EHRs of all patients in the country.
  - Choose-and-Book, an electronic service for booking appointments and clinical referrals.
  - the Electronic Transfer of Prescriptions Service (ETP) will allow prescriptions to be sent electronically to patients’ pharmacies.
- The total number of records is expected to be in the order of $10^8$.

What are some security implications?
The Spine will be supported by the national *Patient Demographic Service (PDS)*.
- Single, comprehensive and consistent source of up-to-date demographic patient data (e.g. NHS number, name, address, preferred language).

**2010 NPfIT expectations:**
- 50 million patients
- 300 million GP appointments per year
- Annually 70 million inpatient episodes and out-patient hospital attendances
- 30 million other health episodes and encounters

Based on digital credentials and public key technology.
- Registration Authorities policies should be able to handle up to 200,000 users and NHS for 500,000 users

**Problems with the implementation**
- Deadlines, cost, compatibility for all GP and NHS, politics, public confidence

Still think the same security implications now?
Policy specification can be used in the NPfIT project for two purposes.

- As a communication aid for issues on information governance
  - Between the NHS and their suppliers.
- NPfIT would benefit from this approach for the following reasons:
  - A formal policy is unambiguous, precise and yet concise.
  - It is much more concrete and specific than current, natural-language specifications such as the Output Based Specification V.2 (OBS), but abstracts away irrelevant implementation details.
  - Policy rules are ideal for presenting alternatives and presenting differences.
  - As formal policies are machine-enforceable, it is easy to build a simulator application for stakeholders to explore alternative policies for compliance.
- The policy and the policy engine would act as a protective layer between the user interface and the restricted system functions and data.

Caldicott Guardians
Questions?
Citations


- *Datalog Queries*. [www.cse.unl.edu/~revesz/413/cdb04.ppt](http://www.cse.unl.edu/~revesz/413/cdb04.ppt)