Bootstrapping Privacy Compliance in Big Data Systems

Anupam Datta

Fall 2014
Privacy Compliance for Bing

Setting:

- Auditor has access to source code
The Privacy Compliance Challenge

- **Legal Team**: Crafts Policy
- **Privacy Champion**: Interprets Policy
- **Developer**: Writes Code
- **Audit Team**: Verifies Compliance

English Privacy Policy

Compliant?

Millions of Lines of Undocumented Code
A Streamlined Audit Workflow

Legal Team
Crafts Policy
Encode
Refine

Legalease
A formal policy specification language

Grok
Data inventory with policy labels

Audit Team
Verifies Compliance
Fix code

Potential violations
Update Grok
Annotated Code
Legalease Policy
Checker

Fix code
Update Grok
Annotated Code
Legalease Policy
Checker
A Streamlined Audit Workflow

**Workflow** for privacy compliance

**Legalease**, usable yet formal policy specification language

**Grok**, bootstrapped data inventory for big data systems

**Scalable** implementation for Bing

- **Legal Team**
  Crafts Policy

- **Legalease**
  A formal policy specification language

- **Grok**
  Data inventory with policy datatypes

- **Developer**
  Writes Code

- **Privacy Champion**
  Interprets Policy

- **Audit Team**
  Verifies Compliance

- **Checker**
  - Code
  - Fix code

- **Update Grok**
  Potential violations

- **Encode**
  Code analysis, developer annotations
A Streamlined Audit Workflow

Legal Team
Crafts Policy

Privacy Champion
Interprets Policy

Legalease
A formal policy specification language

Grok
Data inventory with policy datatypes

Code analysis, developer annotations

Checker

Annotated Code
Legalease Policy
Update Grok
Potential violations

Developer
Writes Code

Audit Team
Verifies Compliance

Fix code
Encode
Refine
Legalease: Syntax

Policy Clause $C ::= D | A$
Deny Clause $D ::= \text{DENY } T_1 \cdots T_n \ \text{EXCEPT } A_1 \cdots A_m$
| \text{DENY } T_1 \cdots T_n$
Allow Clause $A ::= \text{ALLOW } T_1 \cdots T_n \ \text{EXCEPT } D_1 \cdots D_m$
| \text{ALLOW } T_1 \cdots T_n$
Attribute $T ::= \langle \text{attribute-name} \rangle \ v_1 \cdots v_l$
Value $v ::= \langle \text{attribute-value} \rangle$
We will **not** use full IP Address for Advertising.
We will **not** use **full IP Address for Advertising**. IP Address may be used for **detecting abuse**. In such cases, it will not be combined with **account information**.
Designed for Usability

Exceptions
How legal texts are structured
One-to one correspondence

Local Reasoning
Each exception refines its immediate parent
Formally proven property

DENY Datatype IPAddress
  UseForPurpose Advertising
EXCEPT
ALLOW
  Datatype IPAddress:Truncated
ALLOW
  UseForPurpose AbuseDetect
EXCEPT
  DENY Datatype IPAddress,AccountInfo

H. DeYoung, D. Garg, L. Jia, D. Kaynar, and A. Datta, “Experiences in the logical specification of the HIPAA and GLBA privacy laws”
We will **not use full IP Address for Advertising**. IP Address may be used for detecting abuse. In such cases, it will not be combined with account information.
A Lattice of Policy Labels

- If “IPAddress” use is allowed then so is everything below it
- If “IPAddress:Truncated” use is denied then so is everything above it
Legalease Usability

Survey taken by 12 policy authors within Microsoft Encode Bing data usage policy after a brief tutorial

Time spent
2.4 mins on the tutorial
14.3 mins on encoding policy

High overall correctness
A Streamlined Audit Workflow

**Legal Team**
Crafts Policy

**Privacy Champion**
Interprets Policy

**Legalease**
A formal policy specification language

**Grok**
Data inventory with policy labels

**Developer**
Writes Code

**Audit Team**
Verifies Compliance

**Checker**
Legalease Policy
Potential violations

Code analysis
Developer annotations

Encode

Refine

Annotated Code
Update Grok
Fix code
A Streamlined Audit Workflow

Legal Team
Crafts Policy

Privacy Champion
Interprets Policy

Legalease
A formal policy specification language

Grok
Data inventory with policy labels

Code analysis, developer annotations

Checker
Legalease Policy
Potential violations

Update Grok

Audit Team
Verifies Compliance

Fix code
Map-Reduce Programming Systems

Scope, Hive, Dremel

Data in the form of Tables

Code Transforms Columns to Columns

No Shared State
Limited Hidden Flows

users =
    SELECT _name, _age FROM datasetAB

user_tag =
    SELECT GenerateTag(_name, _age)
    FROM users

OUTPUT user_tag TO datasetC
Grok

Purpose Labels
Annotate programs with purpose labels
Grok

Purpose Labels
Annotate programs with purpose labels

Initial Data Labels
Heuristics and Annotations

users =
    SELECT Name, Age FROM datasetAB
user_tag =
    SELECT GenerateTag(_name, _age)
    FROM users
OUTPUT user_tag TO datasetC
Grok

Purpose Labels
Annotate programs with purpose labels

Initial Data Labels
Heuristics and Annotations

Flow Labels
Source labels propagated via data flow graph

D. E. Denning. “A lattice model of secure information flow”
A Lattice of Policy Labels

- If “Profile” use is allowed then so is everything below it
- If “Name” use is denied then so is everything above it
Implicit flows

Beyond direct flows discussed in healthcare audit examples
Map-Reduce

Map
Operate on rows in parallel
eg. filtering

Reduce
Combine groups of rows
eg. aggregation

users =
    SELECT Name, Age FROM datasetAB

users_35 =
    SELECT _name, _age
    FROM users
    WHERE (_age > 35)

ages_35 =
    SELECT _age, COUNT(_name) AS Profile
    FROM users_35
    GROUP BY _age

OUTPUT ages_35 TO datasetC
## Combine Noisy Sources

<table>
<thead>
<tr>
<th>Carefully curated regular expressions</th>
<th>Expensive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leverages developer conventions</td>
<td>Low Noise</td>
</tr>
<tr>
<td>Significant Noise</td>
<td></td>
</tr>
<tr>
<td>Variable Name Analysis</td>
<td>Developer Annotations</td>
</tr>
</tbody>
</table>

| Very Expensive |
| Definitive    |
| Need very few of these |

<table>
<thead>
<tr>
<th>Auditor Verification</th>
</tr>
</thead>
</table>

---

24
Why Bootstrapping Grok Works

A small number of annotations is enough to get off the ground.

Pick the nodes which will label the most of the graph

~200 annotations label 60% of nodes
Scale

- 77,000 jobs run each day
  - By 7000 entities
  - 300 functional groups
- 1.1 million unique lines of code
  - 21% changes on avg, daily
  - 46 million table schemas
  - 32 million files
- Manual audit infeasible
- Information flow analysis takes ~30 mins daily

Fig. 9. Number of GRoK data flow graph nodes added each day
A Streamlined Audit Workflow

Legal Team
Crafts Policy

Privacy Champion
Interprets Policy

Legalease
A formal policy specification language

Grok
Data inventory with policy labels

Checker

Developer
Writes Code

Audit Team
Verifies Compliance

Code analysis
Developer annotations

Annotated Code
Update Grok

Fix code

Encode
Refine

Annotated Code Policy
Potential violations

Legalease Policy
A Streamlined Audit Workflow

**Workflow for privacy compliance**

**Legalease**, usable yet formal policy specification language

**Grok**, bootstrapped data inventory for big data systems

**Scalable** implementation for Bing
Reference

Policy Labels: Datatypes

Going down within a lattice:

- Finer specification

**Datatypes**

- TypeState specifies limited temporal properties

Diagram showing a lattice structure with nodes labeled `:Encrypted` and `:Truncated`, with attributes such as `Account Info`, `Unique ID`, `Location`, `IP Address`, and `Name`.
Policy Types: Concept Lattices

InStore Lattice

UseForPurpose Lattice

AccessByRole Lattice
Formal Semantics

\[ \exists_i D_i \text{ denies } T^G \]

ALLOW \( T^C \) \EXCEPT \( D_1 \cdots D_m \) \denies \( T^G \) (A2)

Based on Lattice Orderings on Policy Types
Formal Semantics

Recursively check exceptions
ALLOW clauses have DENY clauses as exceptions
Top Level clause determines Blacklist/Whitelist
Formal Semantics

Structural properties about semantics

Always returns an unambiguous answer
Weakening lattice orderings makes policy more permissive
Encoding

ALLOW

EXCEPT

DENY DataType IPAddress:Expired
DENY DataType UniqueIdentifier:Expired
DENY DataType SearchQuery, PII InStore Store
DENY DataType UniqueIdentifier, PII InStore Store
DENY DataType BBEPData UseForPurpose Advertising

DENY DataType BBEPData, PII InStore Store

DENY DataType BBEPData:Expired
DENY DataType UserProfile, PII InStore Store

DENY DataType PII UseForPurpose Advertising
DENY DataType PII InStore AdStore

DENY DataType SearchQuery UseForPurpose Sharing
EXCEPT

ALLOW DataType SearchQuery:Scrubbed

- "we remove the entirety of the IP address after 6 months"
- "[we remove] cookies and other cross session identifiers, after 18 months"
- "We store search terms (and the cookie IDs associated with search terms) separately from any account information that directly identifies the user, such as name, e-mail address, or phone numbers."
- "we do not use any of the information collected through the Bing Bar Experience Improvement Program to identify, contact or target advertising to you"
- "we take steps to store [information collected through the Bing Bar Experience Improvement Program] separately from any account information we may have that directly identifies you, such as name, e-mail address, or phone numbers"
- "we delete the information collected through the Bing Bar Experience Program at eighteen months."
- "we store page views, clicks and search terms used for ad targeting separately from contact information you may have provided or other data that directly identifies you (such as your name, e-mail address, etc.)."
- "our advertising systems do not contain or use any information that can personally and directly identify you (such as your name, email address and phone number)."
- "Before we [share some search query data], we remove all unique identifiers such as IP addresses and cookie IDs from the data."