Bootstrapping Privacy Compliance in Big Data Systems

Anupam Datta
CMU

Fall 2017
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
Interprets Policy

Preflight

Grok
Data inventory with policy labels
Code analysis
Developer annotations

Audit Team
Verifies Compliance

Legalease
A formal policy specification language

Checker
Annotated Code
Update Grok
Fix code

Encode
Refine

Potential violations
Update Grok
Annotated Code
Legalease Policy

4
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
- **Grok**, bootstrapped data inventory for big data systems
- Developer
  - Writes Code
- **Grok**, bootstrapped data inventory for big data systems
- Privacy Champion
  - Interprets Policy
- Audit Team
  - Verifies Compliance
Privacy as Restrictions on Personal Information Flow

Restrictions

Direct

Interference

Probabilistic Interference

Differential Privacy

Purpose & Role based

Temporal

EPAL
XACML
*-access control

Purpose → Planning

Jif,
FlowCaml,…

[Hayati & Abadi]

Information Flow Experiments

Differential Privacy

FOTLs
[Formal Contextual Integrity,
Reduce audit algorithm,
Basin et al.]

Grok + Legalease

Differential Privacy Experiments
A Streamlined Audit Workflow

Legal Team
Crafts Policy

Privacy Champion
Interprets Policy

Encode

Refine

Legalease
A formal policy specification language

Grok
Data inventory with policy datatypes

Annotated Code

Update Grok

Potential violations

Checker

Fix code

Code analysis, developer annotations

Annotate Code

Legalease Policy

Audit Team
Verifies Compliance

Developer
Writes Code

Fix code
### Specification: Legalease

Legalease: Syntax

Policy Clause \( C \) ::= \( D \mid A \)
Deny Clause \( D \) ::= DENY \( T_1 \cdots T_n \) EXCEPT \( A_1 \cdots A_m \)
Allow Clause \( A \) ::= ALLOW \( T_1 \cdots T_n \) EXCEPT \( D_1 \cdots D_m \)
Attribute \( T \) ::= \{attribute-name\} \( v_1 \cdots v_l \)
Value \( v \) ::= \{attribute-value\}
DENY Datatype IPAddress

UseForPurpose Advertising

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

H. DeYoung, D. Garg, L. Jia, D. Kaynar, and A. Datta, “Experiences in the logical specification of the HIPAA and GLBA privacy laws”
**Legalease: In Action**

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
Designed for Precision

\[
\begin{align*}
T^G \not\subseteq T^C \quad & \text{ALLOW } T^C \text{ EXCEPT } D_1 \cdots D_m \text{ denies } T^G \quad (A_1) \\
T^G \subseteq T^C \quad & \exists_i D_i \text{ denies } T^G \quad (A_2) \\
T^G \subseteq T^C \quad & \forall_i D_i \text{ allows } T^G \quad (A_3)
\end{align*}
\]

\[
\begin{align*}
\bot \in T^G \cap T^C \quad & \text{DENY } T^C \text{ EXCEPT } A_1 \cdots A_m \text{ allows } T^G \quad (D_1) \\
\bot \notin T^G \cap T^C \quad & \exists_i A_i \text{ allows } T^G \cap T^C \quad (D_2) \\
\bot \notin T^G \cap T^C \quad & \forall_i A_i \text{ denies } T^G \cap T^C \quad (D_3)
\end{align*}
\]

| Policy Clause $C$ | ::= | $D \mid A$ |
|-------------------|====|----------------|
| 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 |

**TABLE I**

Grammar for Legalease

**TABLE III**

Inference rules for Legalease
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.”
Designed for Expressivity (Google, October 2013)

ALLOW
EXCEPT
DENY $dataType$ PII UseForPurpose Sharing

EXCEPT
  ALLOW $dataType$ PII:OptIn
EXCEPT
  ALLOW AccessByRole Affiliates
EXCEPT
  ALLOW UseForPurpose Legal

DENY $dataType$ DoubleClickData, PII
EXCEPT
  ALLOW $dataType$ DoubleClickData, PII:OptIn

◁ “We do not share personal information with companies, organizations and individuals outside of Google unless one of the following circumstances apply:”
◁ “We require opt-in consent for the sharing of any sensitive personal information.”
◁ “We provide personal information to our affiliates or other trusted businesses or persons to process it for us”
◁ “We will share personal information [if necessary to] meet any applicable law, regulation, legal process or enforceable governmental request.”
◁ “We will not combine DoubleClick cookie information with personally identifiable information unless we have your opt-in consent”
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

Encode

Privacy Champion
Interprets Policy

Refine

Legalease
A formal policy specification language

Grok
Data inventory with policy labels

Code analysis
Developer annotations

Annotated Code

Update Grok

Checker

Legalease Policy

Potential violations

Fix code

Developer
Writes Code

Audit Team
Verifies Compliance
A Streamlined Audit Workflow

Legal Team
Crafts Policy

Privacy Champion
Interpret Policy

Legalease
A formal policy specification language

Grok
Data inventory with policy labels

Code analysis, developer annotations

Checker
Annotated Code
Legalease Policy
Update Grok
Potential violations

Developer
Writes Code

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
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
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

---

```sql
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

- Carefully curated regular expressions
  - Leverages developer conventions
  - Significant Noise
- Variable Name Analysis
- Expensive
- Low Noise
- Developer Annotations
- Very Expensive
- Definitive
  - Need very few of these
- Auditor Verification
Why Bootstrapping Grok Works

Pick the nodes which will label the most of the graph

~200 annotations label 60% of nodes

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

Static code analysis

Generate report

Manual Audit

files 25M+

schemas 300K+

positive patterns (40 taxonomy values, 400 patterns)

negative patterns (2500 total entries)

granular overrides (116 total entries)

--

DENY

Datatype

UniqueIdentifier

with PII

INSTORE

SELECT *

FROM (SELECT *

FROM Report

WHERE Taxonomy='ANID'

AND Confidence>='High'

) AS ID

INNER JOIN (SELECT *

FROM Report

WHERE TaxonomyGroup='PII'

AND Confidence>='High'

) AS P

ON ID.VC = P.VC

BingStore

Privacy elements

300K+

Audit candidates

10K+

teams 8

32
A Streamlined Audit Workflow

Legal Team
Crafts Policy

Privacy Champion
Interpret Policy

Legalease
A formal policy specification language

Grok
Data inventory with policy labels

Checker
Annotated Code
Update Grok
Potential violations

Developer
Writes Code
Code analysis
Developer annotations

Audit Team
Verifies Compliance
Fix code
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

TypeState
specifies limited
temporal properties
Policy Types: Concept Lattices

InStore Lattice

UseForPurpose Lattice

AccessByRole Lattice
Formal Semantics

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

ALLOW \( T^C \) EXCEPT \( D_1 \cdots D_m \) denies \( T^G \) (A_2)

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