Compliance in data retrieval systems

Peter Druschel
joint work with Eslam Elnikety, Aastha Mehta, Anjo Vahldiek, Deepak Garg
Compliance in data retrieval systems

Peter Druschel
joint work with Eslam Elnikety, Aastha Mehta, Anjo Vahldiek, Deepak Garg
Data retrieval systems

- Docs
- WWW
- Ads
- Mash-up of data sources

- Clicks, Posts, Queries
- Publishing, Blogging, Trading, Sharing
- Social networking
- Browsing, Searching
- Recommendations
- Advertising
Data retrieval systems

Characteristics:
- Many data types and items
- Many different policies
  - who has access for what purpose for how long
- Complex, dynamic code base

Publishing, Blogging, Trading, Sharing
Social networking
Browsing, Searching
Recommendations
Advertising

Clicks, Posts, Queries
Mash-up of data sources
Data retrieval systems

Google Hangouts And Google Talk Bug Resulting In Messages Going Out To The Wrong Recipients

Updated: Google says they've apparently resolved the problem. We'll update if we hear more about its cause.

A number of sources, including tipsters reporting directly to TechCrunch, as well as Google support forum posters and individuals on Twitter are reporting that Google's Hangouts service is having some issues.

Characteristics:

- Many data types and items
- Many different policies
  - who has access for what purpose for how long
- Complex, dynamic code base
Data retrieval systems

- Google Hangouts And Google Talk Bug Resulting In Messy Wrong Recipients
- TSA Leaks Sensitive Airport Screening Manual

Characteristics:
- Many data types and items
- Many different policies
- who has access for what purpose for how long
- Complex, dynamic code base

Sharing

- who has access for what purpose for how long
- Complex, dynamic code base
Data retrieval systems

Google Hangouts And Google Talk Bug Resulting In Messages Being Sent To Wrong Recipients

TSA Leaks Sensitive Airport Screening Manual

Facebook Apps Leak User Info [REPORT]

Characteristics:

- Many data types and items
- Many different policies
- Who has access for what purpose for how long
- Complex, dynamic code base
Data retrieval systems: Architecture

[Diagram showing the architecture of a data retrieval system with components like Docs, WWW, Ads, Search engine, Recommender, Indexer, Index, and Analytics. The diagram also includes a table with columns Name, Cookie, Cred, Settings, Friends, History, and Profile.]
Data retrieval systems: Architecture

More complex in practice due to distribution, parallel processing.
Data retrieval system: Policies

Front end

Docs
WWW
Ads

Account editor

Ad exchange

Search engine

Recommender

Index

Indexer

Name
Cookie
Cred
Settings
Friends
History
Profile

Analytics
Data retrieval system: Policies

Front end

Ad exchange

Search engine

Recommender

Index

Indexer

Account editor

Name | Cookie | Cred | Settings | Friends | History | Profile
--- | --- | --- | --- | --- | --- | ---

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public

Public
Data retrieval system: Policies

Private
Docs
WWW
Ads
Public

Private

Public

Front end
Search engine
Indexer
Index

Ad exchange

Recommender

Account editor

Name
Cookie
Cred
Settings
Friends
History
Profile

Analytics
Data retrieval system: Policies

Private

Docs
WWW
Ads
Friends
Public

Front end
Search engine
Recommender
Ad exchange
Indexer
Index

Account editor

Name | Cookie | Cred | Settings | Friends | History | Profile
--- | --- | --- | --- | --- | --- | ---

Analytics
Data retrieval system: Policies

- Private
  - Docs
  - WWW
  - Ads
  - Friends

- Public
  - Front end
  - Search engine
  - Recomender
  - Friends of friends

- Account editor
  - Name
  - Cookie
  - Cred
  - Settings
  - Friends
  - Index
  - Analytics

TCE Conference 2015, Technion, Haifa
Data retrieval system: Policies

- Private
- Public

- Docs
- WWW
- Ads
- Search engine
- Index
- Recomender
- Friends of friends

- Front end
- Account editor

- Friends
- Personalization only; history deleted after 48h

- Analytics

<table>
<thead>
<tr>
<th>Name</th>
<th>Cookie</th>
<th>Cred</th>
<th>Settings</th>
<th>Friends</th>
<th>History</th>
<th>Profile</th>
</tr>
</thead>
</table>

Expires after 2h
Data retrieval system: Policies

- Private
  - Docs
  - WWW
- Public
  - Ads
  - Friends
- Front end
- Search engine
- Ad exchange
- Recommender
- Friends of friends
- Account editor
- Analytics

<table>
<thead>
<tr>
<th>Name</th>
<th>Cookie</th>
<th>Cred</th>
<th>Settings</th>
<th>Friends</th>
<th>History</th>
<th>Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Staff access to client data audited
Expires after 2h
Personalization only; history deleted after 48h
Staff access to client data audited
Data retrieval system: Policies

Private

Content banned in certain jurisdictions

Docs
WWW
Ads
Friends
Public
Expires after 2h

Frontend

Search engine

Ad exchange

Friends of friends

Private

friends of friends

Staff access to client data audited

Front end

Search engine

Ad exchange

Friends

Public

Private

Friends

Public

Friends of friends

Personalization only; history deleted after 48h

Private

Friends of friends

Content banned in certain jurisdictions

Analytics

Profile

History

Staff access to client data audited

Account editor

Name
Cookie
Cred
Settings
Friends
History
Profile

Expires after 2h

Analytics
Data retrieval system: Policies

- Private
- Content banned in certain jurisdictions
- Other legal requirements: logging, retention

- Public
- Expires after 2h
- Staff access to client data audited

- Friends of friends
- Personalization only; history deleted after 48h

- Front end
- Search engine
- Ad exchange
- Indexer
- Recommender

- Account editor

<table>
<thead>
<tr>
<th>Name</th>
<th>Cookie</th>
<th>Cred</th>
<th>Settings</th>
<th>Friends</th>
<th>History</th>
<th>Profile</th>
</tr>
</thead>
</table>

| Analytics |
Data retrieval system: Policies

Challenges:
- Too many data items, too many policies
- Policy implicit in configuration, ACLs, enforcement logic
- Enforcement spread over many components, layers
- Complex, fast evolving software

Content banned in certain jurisdictions
Other legal requirements: logging, retention

Staff access to client data audited
Expires after 2h
Personalization only; history deleted after 48h
Staff access to client data audited
Expires after 2h
Content banned in certain jurisdictions
Other legal requirements: logging, retention
Outline

- Motivation
  - Data retrieval systems, policies
- Thoth: Distributed data compliance
  - Overview, threat model
  - Policy language, example policies
  - Enforcement
  - Type-constrained declassification
  - Partial policy evaluation
- Prototype and preliminary evaluation
- Related work and conclusion
Thoth: Distributed data compliance

- Declarative policy language
  - Confidentiality, integrity, provenance, declassification
Thoth: Distributed data compliance

- Declarative policy language
  - Confidentiality, integrity, provenance, declassification

- Policies attached to data conduits
  - Files, KVS tuples, pipes, network connections

- Distributed enforcement layer
  - Mediates process I/O at the OS level
  - Process-level information flow tracking/control
  - Interprets, enforces conduit policies
Challenges

- Low runtime overhead
- Work with existing software
- Easy to use
- Rely on a relatively small, stable trusted computing base
Thoth: Key ideas

- Process-level IFC
  - Language/runtime independent, efficient
  - Good match for distributed/parallel frameworks
Thoth: Key ideas

- Process-level IFC
  - Language/runtime independent, efficient
  - Good match for distributed/parallel frameworks

- Declassification, provenance policies
  - Avoid trusted declassifiers
  - Type-constrained declassification
Thoth: Key ideas

- Process-level IFC
  - Language/runtime independent, efficient
  - Good match for distributed/parallel frameworks

- Declassification, provenance policies
  - Avoid trusted declassifiers
  - Type-constrained declassification

- Policies propagated as taint
  - Enables partial policy evaluation to prevent taint accumulation
Thoth: Key ideas

- Transactions
  - Atomic commit if compliant, abort otherwise
Thoth: Key ideas

- Transactions
  - Atomic commit if compliant, abort otherwise
- User authentication
Thoth: Key ideas

- Transactions
  - Atomic commit if compliant, abort otherwise

- User authentication

- Untrusted code demonstrates policy compliance
  - Extended POSIX API
  - Concise policies
  - Simple policy language, interpretation
Threat model

- OS, storage, Thoth reference monitor trusted
Threat model

- OS, storage, Thoth reference monitor trusted
- Some implicit flows, side channels not considered
Threat model

- OS, storage, Thoth reference monitor trusted
- Some implicit flows, side channels not considered
- Some declassification policies assume applications do not actively encode private information in search results or profile vectors
Thoth: A distributed compliance layer

Bob

Thoth
OS
HW

Thoth
OS
HW

Thoth
OS
HW

Cred

Bob
Thoth: A distributed compliance layer
Thoth: A distributed compliance layer
Thoth: A distributed compliance layer
Thoth: A distributed compliance layer
Thoth: A distributed compliance layer

[Diagram showing a distributed system with multiple levels including Thoth, OS, and HW layers, with connections between them and a user interface for Bob.]
Thoth: A distributed compliance layer

Diagram showing a distributed system with multiple levels of Thoth, OS, and HW, interconnected with Bob and Cred.
Outline

- Motivation
  - Data retrieval systems, policies
- Thoth: Distributed data compliance
  - Overview, threat model
  - Policy language, example policies
  - Enforcement
  - Type-constrained declassification
  - Partial policy evaluation
- Prototype and preliminary evaluation
- Related work and conclusion
Thoth: Policy language

A policy has five rules of the form
permission :- condition
Thoth: Policy language

A policy has five rules of the form
permission :- condition

- permission = \{read, update, destroy, provenance, declassify\}
Thoth: Policy language

A policy has five rules of the form
permission :- condition

- permission = \{read, update, destroy, provenance, declassify\}
- condition is a boolean expression of predicates
Thoth: Policy language

A policy has five rules of the form
permission :- condition

- permission = \{read, update, destroy, provenance, declassify\}
- condition is a boolean expression of predicates
- predicates: relational/arithmetic, session, conduit, content, certificates (e.g. wall-clock time)
Thoth: Policy language

A policy has five rules of the form

```plaintext
permission :- condition
```

- permission = \{read, update, destroy, provenance, declassify\}
- condition is a boolean expression of predicates
- predicates: relational/arithmetic, session, conduit, content, certificates (e.g. wall-clock time)
- Declassification rule: c1 until c2
Thoth: Policy language

A policy has five rules of the form
permission :- condition

- permission = \{read, update, destroy, provenance, declassify\}
- condition is a boolean expression of predicates
- predicates: relational/arithmetic, session, conduit, content, certificates (e.g. wall-clock time)
- Declassification rule: c1 until c2
- Provenance rule: c1 since c2
Relational and arithmetic predicates

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eq(x,y)</td>
<td>(x = y)</td>
</tr>
<tr>
<td>neq(x,y)</td>
<td>(x \neq y)</td>
</tr>
<tr>
<td>lt(x,y)</td>
<td>(x &lt; y)</td>
</tr>
<tr>
<td>gt(x,y)</td>
<td>(x &gt; y)</td>
</tr>
<tr>
<td>le(x,y)</td>
<td>(x \leq y)</td>
</tr>
<tr>
<td>ge(x,y)</td>
<td>(x \geq y)</td>
</tr>
<tr>
<td>add(x,y,z)</td>
<td>(x + y)</td>
</tr>
<tr>
<td>sub(x,y,z)</td>
<td>(x - y)</td>
</tr>
<tr>
<td>mul(x,y,z)</td>
<td>(x \times y)</td>
</tr>
<tr>
<td>div(x,y,z)</td>
<td>(x / y)</td>
</tr>
<tr>
<td>rem(x,y,z)</td>
<td>(x \mod y)</td>
</tr>
<tr>
<td>concat(x,y)</td>
<td>(x \ll y)</td>
</tr>
<tr>
<td>varType(x,y)</td>
<td>(\text{type of } x = y)</td>
</tr>
</tbody>
</table>

Session predicates

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sKeys(x)</td>
<td>(x) is the session's client authentication key</td>
</tr>
<tr>
<td>sIPs(x)</td>
<td>(x) is the session's source IP address</td>
</tr>
<tr>
<td>lpPrefix(x,y)</td>
<td>(x) is a prefix of IP address (y)</td>
</tr>
<tr>
<td>time(t)</td>
<td>(t) is the current time</td>
</tr>
</tbody>
</table>

Conduit predicates

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cName(x)</td>
<td>(x) is the pathname of conduit</td>
</tr>
<tr>
<td>cId(x)</td>
<td>(x) is the id of conduit</td>
</tr>
<tr>
<td>cIdExists(x)</td>
<td>(x) is a valid conduit id</td>
</tr>
<tr>
<td>cCurrLen(x)</td>
<td>(x) is the current conduit length</td>
</tr>
<tr>
<td>cNewLen(x)</td>
<td>(x) is the new conduit length</td>
</tr>
<tr>
<td>clsIntrinsic</td>
<td>Is this conduit intrinsic?</td>
</tr>
</tbody>
</table>

Content predicates

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(c,off,len) says</td>
<td>(x_1, \ldots, x_n) is the tuple</td>
</tr>
<tr>
<td>((x_1, \ldots, x_n)) willsay</td>
<td>ditto for the update</td>
</tr>
<tr>
<td>((x_1, \ldots, x_n)) has</td>
<td>ditto for the update</td>
</tr>
<tr>
<td>each in ((c,off,len)) says</td>
<td>for each tuple in (c) at (off, len), assign</td>
</tr>
<tr>
<td>((x_1, \ldots, x_n)) {condition}</td>
<td>to (x_1, \ldots, x_n) and evaluate condition</td>
</tr>
<tr>
<td>each in ((c,off,len)) willsay</td>
<td>ditto for the update</td>
</tr>
<tr>
<td>((x_1, \ldots, x_n)) {condition}</td>
<td>of (c) in the current transaction</td>
</tr>
</tbody>
</table>

Provenance and declassification rules

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1 until c2</td>
<td>Condition (c1) must hold on the downstream flow until (c2) holds</td>
</tr>
<tr>
<td>c1 since c2</td>
<td>Condition (c1) must hold on the upstream flow since (c2) was true</td>
</tr>
<tr>
<td>isAsRestrictive(p1,p2)</td>
<td>The permission (p1) is at least as restrictive as (p2)</td>
</tr>
</tbody>
</table>

Table 1. Thoth policy language predicates and connectives
Example policy: Client access control

\[ \text{read} :\!-\! s\text{KeyIs}(k_{Alice}) \quad // \text{private} \]
Example policy: Client access control

\[
\text{read} :- \ s\text{KeyIs}(k_{Alice}) \quad // \text{private} \\
\quad \lor \ (s\text{KeyIs}(k_{Bob}) \quad // \text{friends} \\
\quad \quad \land \ (Alice.\text{friends}, \ off) \text{ says isFriend}(k_{Bob}))
\]
Example policy: Client access control

read :- sKeyIs(k_{Alice})    // private
    ∨ (sKeyIs(k_{Bob})      // friends
       ∧ (Alice.friends, off) says isFriend(k_{Bob}))
    ∨ (sKeyIs(k_{Charlie})  // friends of friends
       ∧ (Alice.friends, off1) says isFriend(k_{F})
       ∧ (F.friends, off2) says isFriend(k_{Charlie}))
Example policy: Client access control

\[
\text{read} : - \text{sKeyIs}(k_{Alice}) \quad // \text{private} \\
\lor \; (\text{sKeyIs}(k_{Bob}) \quad // \text{friends} \\
\quad \land (\text{Alice.friends, off}) \text{ says isFriend}(k_{Bob})) \\
\lor \; (\text{sKeyIs}(k_{Charlie}) \quad // \text{friends of friends} \\
\quad \land (\text{Alice.friends, off1}) \text{ says isFriend}(k_F) \\
\quad \land (\text{F.friends, off2}) \text{ says isFriend}(k_{Charlie}))
\]

“Only authorized users are able to access this data.”
Example policy: Client access control

\[
\text{read} \cdot s\text{KeyIs}(k_A) \quad \text{// private}
\]
\[
\bigvee s\text{KeyIs}(k_B) \quad \text{// friends}
\]
\[
\quad (\text{Alice.} \text{friends}, \ \text{off}) \ \text{say} \text{s isFriend}(k_B)
\]
\[
\bigvee s\text{KeyIs}(k_{\text{Charlie}}) \quad \text{// friends of friends}
\]
\[
\quad (\text{Alice.} \text{friends}, \ \text{off1}) \ \text{say} \text{s isFriend}(k_F)
\]
\[
\quad (F. \text{friends}, \ \text{off2}) \ \text{say} \text{s isFriend}(k_{\text{Charlie}})
\]

“Only authorized users are able to access this data.”
Example policy: Client access control

read :- sKeyIs(k_{Alice}) // private
\lor (sKeyIs(k_{Bob}) // friends
  \land (Alice.friends, off) says isFriend(k_{Bob}))
\lor (sKeyIs(k_{Charlie}) // friends of friends
  \land (Alice.friends, off1) says isFriend(k_{F})
  \land (F.friends, off2) says isFriend(k_{Charlie}))

update :- sKeyIs(k_{Alice})

“Only authorized users are able to access this data.”
Example policy: Client access control

\textbf{read} :- \texttt{sKeyIs}(k_{Alice}) \quad // \text{private}

\begin{align*}
\lor \quad \texttt{sKeyIs}(k_{Bob}) & \quad // \text{friends} \\
\land \quad (\text{Alice.friends, off}) & \text{says isFriend}(k_{Bob})) \\
\lor \quad \texttt{sKeyIs}(k_{Charlie}) & \quad // \text{friends of friends} \\
\land \quad (\text{Alice.friends, off1}) & \text{says isFriend}(k_F) \\
\land \quad (\text{F.friends, off2}) & \text{says isFriend}(k_{Charlie}))
\end{align*}

\textbf{update} :- \texttt{sKeyIs}(k_{Alice})

\textbf{destroy} :- \texttt{sKeyIs}(k_{Alice})

“Only authorized users are able to access this data.”
Example policy: Censorship

declassify :- ... until (... ∧ CENSOR)
Example policy: Censorship

declassify :- ... until (... ∧ CENSOR)

CENSOR:

sessionIPIs(ip) ∧ IPInRegion(ip, region) ∧
(region.BL, off) says cnd1 ∧
(region.BL, off+1) says cnd2 ∧
hash("foo.htm") > cnd1 ∧ hash("foo.htm") < cnd2

region.BL integrity policy requires sorted order
Example policy: Censorship

declassify :- ... until (... ∧ CENSOR)

CENSOR:

\[
\text{sessionIPIs}(ip) \land \text{IPInRegion}(ip, \text{region}) \land \n\text{(region.BL, off) says cnd1} \land \n\text{(region.BL, off+1) says cnd2} \land \n\text{hash("foo.htm") > cnd1} \land \text{hash("foo.htm") < cnd2}
\]

region.BL integrity policy requires sorted order

“Data derived from foo.htm must not be accessed by clients with an IP prefix whose blacklist mentions foo.htm”
More example policies

- **Mandatory access logging (MAL):** staff access of client data must be logged
  - Declassification into staff readable conduit if logged
  - Log has append-only integrity policy, auditor read only
More example policies

- **Mandatory access logging (MAL):** staff access of client data must be logged
  - Declassification into staff readable conduit if logged
  - Log has append-only integrity policy, auditor read only

- **Private history:** Client search history can be used for profile generation only
  - Private with declassification into a vector of m floats
More example policies

- **Mandatory access logging (MAL):** staff access of client data must be logged
  - Declassification into staff readable conduit if logged
  - Log has append-only integrity policy, auditor read only

- **Private history:** Client search history can be used for profile generation only
  - Private with declassification into a vector of m floats

- **Expiration:** Microblog posts searchable for limited period
  - Read requires certificate from trusted wall-clock source
Outline

- Motivation
  - Data retrieval systems, policy compliance
- Thoth: Distributed data compliance
  - Overview, threat model
  - Policy language, example policies
  - Enforcement
  - Type-constrained declassification
  - Partial policy evaluation
- Prototype and preliminary evaluation
- Related work and conclusion
Thoth: Policy enforcement algorithm

Process $p$ performs I/O on conduit $c$
Thoth: Policy enforcement algorithm

Process p performs I/O on conduit c

- If p is confined (I/O mediated by Thoth)
  - read: c.policy is added to p.taint
  - write: check c.update;
    foreach (c until c’) rule in p.taint
      check c’ OR (c AND p.declassify => (c until c’))
    <symmetric check for provenance rules>
Thoth: Policy enforcement algorithm

Process p performs I/O on conduit c

- If p is confined (I/O mediated by Thoth)
  - read: c.policy is added to p.taint
  - write: check c.update;
    
    \[
    \text{foreach (c until c')} \text{ rule in p.taint}
    \]
    
    \[
    \text{check c' OR (c AND p.declassify => (c until c'))}
    \]
    
    <symmetric check for provenance rules>

- If p is unconfined (unmediated external I/O)
  - Read: check c’s read permission
  - Write: check c’s update permission
Thoth: Policy enforcement algorithm

Process p performs I/O on conduit c

- If p is confined (I/O mediated by Thoth)
  - read: c.policy is added to p.taint
  - write: check c.update;
    
    foreach (c until c’) rule in p.taint
    
    check c’ OR (c AND p.declassify => (c until c’))
    
    <symmetric check for provenance rules>

- If p is unconfined (unmediated external I/O)
  - Read: check c’s read permission
  - Write: check c’s update permission

Ensures compliance regardless of policies on internal conduits
Example: Index/search flow
Example: Index/search flow
Example: Index/search flow

Private
Docs
WWW
Ads
Friends
Public
Public
Private
Public
Public
Public
Friends
Public
Front end
Search engine
Indexer
Index
read: false
declassify: false

<table>
<thead>
<tr>
<th>Name</th>
<th>Cookie</th>
<th>Cred</th>
<th>Settings</th>
<th>Friends</th>
<th>History</th>
<th>Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example: Index/search flow

- Private
  - Docs
  - WWW
  - Ads

- Public
  - Friends
  - Indexer
  - Index

Front end

- Search engine

- Indexer

<table>
<thead>
<tr>
<th>Name</th>
<th>Cookie</th>
<th>Cred</th>
<th>Settings</th>
<th>Friends</th>
<th>History</th>
<th>Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example: Index/search flow

- Private:
  - Docs
  - WWW
  - Ads

- Public:
  - Friends

- Front end
- Search engine
- Indexer

- Table:

<table>
<thead>
<tr>
<th>Name</th>
<th>Cookie</th>
<th>Cred</th>
<th>Settings</th>
<th>Friends</th>
<th>History</th>
<th>Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example: Index/search flow

Front end → Search engine → Indexer

Problem: unable to extricate search results
Declassifying search results

Each searchable source includes in its policy:

\texttt{declassify :- isAsRestrictive(read, this.read) until}
\texttt{(update :- CND_IDS ∨ CENSOR)}
Declassifying search results

Each searchable source includes in its policy:

\[
\text{declassify} \text{ :- isAsRestrictive(read, this.read) until } \\
(\text{update} \text{ :- CND_IDS} \land \text{ CENSOR}) \\
\text{CND_IDS} \text{: } \text{cCurrLenIs(currLen)} \land \text{cNewLenIs(newLen)} \land \\
\text{each in (this, currLen, newLen) says (cndId)} \\
\{\text{cIdExists(cndId)}\}
\]
Declassifying search results

Each searchable source includes in its policy:

\[
\text{declassify} \ :- \ \text{isAsRestrictive}(\text{read}, \ \text{this.read}) \ \text{until} \\
(\text{update} \ :- \ \text{CND} \_ \text{IDS} \ \land \ \text{CENSOR}) \\
\text{CND} \_ \text{IDS}: \ c\text{CurrLenIs}(\text{currlen}) \ \land \ c\text{NewLenIs}(\text{newlen}) \ \land \\
\text{each in (this, currlen, newlen) says (cndId)} \\
\{\text{cIdExists(cndId)}\}
\]

“Allows declassification into a list of valid conduit ids”
Example: Index/search flow

<table>
<thead>
<tr>
<th>Name</th>
<th>Cookie</th>
<th>Cred</th>
<th>Settings</th>
<th>Friends</th>
<th>History</th>
<th>Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bob</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example: Index/search flow

Front end → Search engine → Index → Indexer

Docs → WWW → Ads

read :- false
declassify :- ...

<table>
<thead>
<tr>
<th>Name</th>
<th>Cookie</th>
<th>Cred</th>
<th>Settings</th>
<th>Friends</th>
<th>History</th>
<th>Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example: Index/search flow

- Docs
- WWW
- Ads
- Settings
- Friends
- History
- Profile

Declassification into List of cnIds

<table>
<thead>
<tr>
<th>Cookie</th>
<th>Cred</th>
<th>Settings</th>
<th>Friends</th>
<th>History</th>
<th>Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bob</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

read :- false
declassify :- ...
Example: Index/search flow
Example: Index/search flow

read :- TRUE
declassify :- ... until (... ∧ CENSOR)
Example: Index/search flow

```
read :- TRUE
declassify :- ... until (... ∧ CENSOR)
```

Problem: FE process accumulates document-specific CENSOR taint
Partial policy evaluation

- Front end (FE) reads source doc with censor clause:
  
  sessionIPIs(ip) ∧ IPInRegion(ip, region) ∧
  
  (region.BL, off) says cnd1 ∧
  
  (region.BL, off+1) says cnd2 ∧
  
  hash(“foo.htm”) > cnd1 ∧ hash(“foo.htm”) < cnd2
Partial policy evaluation

- Front end (FE) reads source doc with censor clause:
  
  \[
  \text{sessionIPIs}(\text{ip}) \land \text{IPInRegion}(\text{ip}, \text{region}) \land \\
  (\text{region.BL, off}) \text{ says cnd1} \land \\
  (\text{region.BL, off+1}) \text{ says cnd2} \land \\
  \text{hash(“foo.htm”) > cnd1} \land \text{hash(“foo.htm”) < cnd2}
  \]

- Literal “foo.htm” would lead to FE taint accumulation!
Partial policy evaluation

- Front end (FE) reads source doc with censor clause:
  \[
  \text{sessionIPIs}(ip) \land \text{IPInRegion}(ip, \text{region}) \land \\
  (\text{region.BL}, \text{off}) \text{ says } \text{cnd1} \land \\
  (\text{region.BL}, \text{off+1}) \text{ says } \text{cnd2} \land \\
  \text{hash(“foo.htm”) > cnd1} \land \text{hash(“foo.htm”) < cnd2}
  \]
- Literal “foo.htm” would lead to FE taint accumulation!
- Solution: partial policy evaluation
Partial policy evaluation

- Front end (FE) reads source doc with censor clause:
  \[
  \text{sessionIPIs}(ip) \land \text{IPInRegion}(ip, region) \land \\
  (\text{region.BL}, \text{off}) \text{ says } \text{cnd1} \land \\
  (\text{region.BL}, \text{off+1}) \text{ says } \text{cnd2} \land \\
  \text{hash}("\text{foo.htm}") > \text{cnd1} \land \text{hash}("\text{foo.htm}") < \text{cnd2}
  \]
- Literal “foo.htm” would lead to FE taint accumulation!
- Solution: partial policy evaluation
- Thoth partially evaluates censor clause with ip binding provided by FE
Partial policy evaluation

- Front end (FE) reads source doc with censor clause:
  \[
  \text{sessionIPIs}(ip) \land \text{IPInRegion}(ip, \text{region}) \land
  (\text{region.BL}, \text{off}) \text{ says } \text{cnd1} \land
  (\text{region.BL}, \text{off}+1) \text{ says } \text{cnd2} \land
  \text{hash(“foo.htm”)} > \text{cnd1} \land \text{hash(“foo.htm”)} < \text{cnd2}
  \]
- Literal “foo.htm” would lead to FE taint accumulation!
- Solution: partial policy evaluation
- Thoth partially evaluates censor clause with ip binding provided by FE
  - Clauses that evaluate to TRUE are dropped
  - In common case, sessionIPIs(ip) remains
Outline

- Motivation
  - Data retrieval systems, policies
- Thoth: Distributed data compliance
  - Overview, threat model
  - Policy language, example policies
  - Enforcement
  - Type-constrained declassification
  - Partial policy evaluation
- Prototype and preliminary evaluation
- Related work and conclusion
Thoth: Implementation

Per-node enforcement

- Linux security module intercepts system calls
- Reference monitor
  - authenticates clients
  - tracks information flow
  - interprets policies, aborts non-compliant I/O transactions

![Diagram of Thoth components]

**Trusted components**

- Task\(_1\)
- Task\(_n\)
- Thoth LSM module
- Linux OS kernel
- Thoth Ref Mon
- Thoth Metadata and Log
- Thoth Global Policy Store
Prototype search engine

Lucene search core on Linux/Thoth, memcached
Prototype

- Dell R410 servers, Intel Xeon5650 2.66Ghz
  - 48GB memory, 2x 1Gb Ethernet
- OpenSuse Linux 13.1, kernel 3.13.1, x86-64
- OpenSSL v1.0.1k
- Thoth LSM module (3500 LoC), refmon (18610 LoC)
- Lucene v4.7 (with 50 LoC changed)
Results: Throughput

![Graph showing throughput comparison between Linux and Thoth for 1 engine/shard and 2 engines/shard.]

English Wikipedia, 220GB, 2 index shards of 29/26GB, in-memory
Policies: 30/50/20% private, public, friend-only
1% MAL accesses, 2250 files censored/region
Related work

- Grok [Guha et al. S&P14]: Policy compliance for Bing

**IFC:**
- Resin [Yip et al. SOSP09], Hails [Giffin et al. OSDI12], COWL [Stefan et al. OSDI14]
- Asbestos [Efstathopuolos et al. Eurosyst08], HiStar [Zeldovich et al. OSDI06], Flume [Krohn SOSP07]

**Declarative policies:**
- Datalog, LTL
- Nexus [Shieh et al. SOSP05], PCFS [Garg et al. S&P10]
Conclusion

- OS-level data compliance layer
- Rich declarative policies attached to conduits
  - confidentiality, integrity, provenance, declassification
- Compliance in data retrieval systems
- Modest overhead, room for further optimization
- Ongoing work:
  - Extend to SQL databases
  - Combine with static analysis
  - Extend to general data processing systems