Thoth: Comprehensive Policy Compliance in Data Retrieval Systems

Eslam Elnikety, Aastha Mehta, Anjo Vahldiek-Oberwagner, Deepak Garg, and Peter Druschel

Max Planck Institute for Software Systems
Data retrieval systems

collect, process, and serve data

- Searching, browsing, advertising
- Social networking
- Blogging, publishing, news
Data retrieval systems: many data sources

- msg & emails
- social net.
- streams
- WWW

- click history
- user profile

- Personalization
- Ad exchange

- Indexer
- Search engine

- index
Many policies

- Click history
- User profile

**Clients’ privacy preferences**

- Personalization
- Ad exchange

**Streams**
- Indexer
- Search engine

**Social network**
- Private
- Friends
- Friends of friends
- Public

**WWW**
- Msgs & emails

**Clients’ privacy preferences**
- Clients' privacy preferences
Many policies

- Personalization only; expire after 2 days

- Click history
- User

- Clients’ privacy preferences
- Providers’ privacy choices

- Personalization

- Audit staff access to client data

- Indexer
- Search engine

- Index

- Streams
- Social net.
- WWW

- Messages & emails
Many policies

- Click history
- User

- Clients’ privacy preferences
- Providers’ privacy choices
- Legal requirements

- Content banned in certain jurisdictions

- Indexer
- Index

- Personalization
- Ad exchange

- Other legal mandates: logging, retention

- Search engine

- Messages & emails
- Social network
- Streams
- WWW

Clients’ privacy preferences

Providers’ privacy choices

Legal requirements

Content banned in certain jurisdictions
Policy compliance challenges

- Many data items and complex policies
- Policy implicit in configuration files, code
- Enforcement spread over many components, layers
- Complex, fast evolving applications

**Goal:** Prevent *inadvertent* policy violations due to application bugs and misconfigurations
Outline

Motivation
• Data retrieval systems, policies
• Challenges and goals

Thoth: policy compliance layer
• Overview, threat model
• Policy language and examples
• Typed declassification
• Runtime enforcement

Prototype and evaluation
Thoth: overview

Policies attached to content

- click history
- user profile
- Ad exchange

Personalization

attached to internal content

Search engine

attached to sources and external connections

- messages & emails
- social network
- streams
- WWW

Policies attached to internal content

attached to sources and external connections
Thoth: overview

- Policies attached to content
- Process-level information flow control

- Click history
- User profile

- Personalization
- Ad exchange

- Track data flows that cross process boundary

- Indexer
- Search engine

- Policies-as-taint

- Streams
- Social net.
- Messages & emails
- WWW

- Index
Thoth: overview

- Policies attached to content
- Process-level information flow control

- Click history
- User profile

- Personalization
- Ad exchange

- Indexer
- Index
- Search engine

- Messages & emails
- Social network
- Streams
- WWW
Thoth: overview

Policies attached to content

Process-level information flow control

click history

user profile

Personalization

Ad exchange

Indexer

Search engine

msgs & emails

social net.

streams

WWW
Thoth: overview

Policies attached to content

Process-level information flow control

- click history
- user profile

- Personalization
- Ad exchange

msgs & emails
social net.
streams
WWW

- policy check
- index

Indexer
Search engine
Thoth: overview

Policies attached to content

Process-level information flow control

click history

user profile

Personalization

Ad exchange

msgs & emails

social net.

streams

WWW

Indexer

Search engine

policy check

index
Thoth: overview

Policies attached to content
- msgs & emails
- social net.
- streams
- WWW

Process-level information flow control
- click history
- user profile

Personalization
Ad exchange

Indexer
Search engine

Policies attached to content
Thoth: overview

Regardless of the **internal complexity**, correct **source policies** ensure policy compliance.
Thoth: threat model

- Thoth, OS, storage are trusted (but not applications)
- Covert and side channels are not a concern

**Pragmatic:**
- Provider is interested in policy compliance
- Goal is preventing inadvertent policy violations

**Guarantees:**
Application bugs and misconfigurations will not violate source policies
Thoth policy language overview

Declarative *data flow* policy language

Thoth policies
- can express confidentiality, integrity, declassification
- are attached to source content

Who can access? when? which state?

How can access conditions change along a data flow?
Example: client access control

Alice’s post accessible by herself, friends, friends of friends

read :-

sessionIs(Alice)  /* Alice */

/* Alice’s friends:
   x can access when in Alice’s friend list */

OR

sessionIs(x) AND "friend_list_{Alice, offset_{x}}" says x

/* Alice’s friends of friends:
   y can access when in x’s friend list, 
   and x is in Alice’s friend list */

OR

sessionIs(y) AND "friend_list_{x, offset_{y}}" says y AND "friend_list_{Alice, offset_{y}}" says x

“Only authorized users, as specified by Alice, can access the post.”
Example: client access control

Alice’s post accessible by herself, friends, friends of friends

read :-

sessionIs(Alice) /* Alice */

/ * Alice’s friends: 

x can access when in Alice’s friend list */

OR

sessionIs(x) AND “friend_list_Alice, offset_x” says x

/ * Alice’s friends of friends: 

y can access when in x’s friend list, 

and x is in Alice’s friend list */

OR

sessionIs(y) AND “friend_list_x, offset_y” says y AND “friend_list_Alice, offset_y” says x

“Only authorized users, as specified by Alice, can access the post.”
More example policies

- Personalization only; expire after 2 days
- Audit staff access to client data
- Content banned in certain jurisdictions
- Click history
- User profile
- Ads
- Search engine
- Ad exchange
- Social net.
- Streams
- Msgs & emails
- WWW
Thoth: Key ideas

- **Process-level** information flow control
  - Language independent, efficient
  - Good match for distributed computing frameworks

- **Policies-as-taint**
  - Enables taint reduction via policy comparison, partial evaluation

- **Policy-specified** declassification
  - No trust in application code for declassification

- **Typed declassification**
Typed declassification

Allows declassification for data of a specific type

**Example:** *Declassifying search results*

- **read:** Alice
- **read:** Bob
- **read:** no one
Typed declassification

Allows declassification for data of a specific type

**Example:** Declassifying search results

**ONLY_URLS**
- Search engine output must be list of urls

**read:** no one
**declassify:** ONLY_URLS

Search engine

**read:** anyone who can read the content

Other types: floats, integers
End-to-end policy enforcement

**Algorithm:** Process $p$ performs I/O on content with policy $pol$.

- **If $p$ is external** (access control)
  - **read:** check $pol$.read
  - **write:** check $pol$.update

- **If $p$ is internal** (flow control)
  - **read:** $pol$ is added to $p$.taint
  - **write:** check $pol$.update; for each policy in the taint; either
    - declassification conditions are met, or
    - $pol$ is at least as restrictive as the taint policy

*Ensures compliance regardless of internal policies*
Outline

Motivation
• Data retrieval systems, policies
• Challenges and goals

Thoth: policy compliance layer
• Overview, threat model
• Policy language and examples
• Typed declassification
• Runtime enforcement

Prototype and evaluation
Thoth Implementation

Per-node enforcement

- Linux security module (LSM) (3,500 LoC)
  - intercepts system calls (e.g., read, write, ..)
  - exposes Thoth API (e.g., set_policy)
- Reference monitor (19,000 LoC + OpenSSL)
  - authenticates users
  - evaluates policies
- Global policy store
Prototype search engine

Changes: 50 LoC (out of 300,000)
Performance evaluation

Setup
- 2-shard index, each shard hosted by a server
- No replication (2 servers), 2x replication (4 servers)

Dataset
- Wikipedia English articles; 15M documents
- Policies
  - 30% private; 50% public, 20% friends-only
  - Non-public documents allow staff access subject to logging
  - All documents have censorship clause

Workload
- Queries based on Wikipedia popularity access trace
Concurrent queries; clients saturate the search engine.

In-memory index and documents to fully expose Thoth overheads.

Overhead can be reduced significantly with:
- a more efficient I/O interception
- an in-kernel reference monitor

overhead:
- I/O interception + IPC to reference monitor
- policy evaluation
Search latency

<table>
<thead>
<tr>
<th></th>
<th>Average (ms)</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>Thoth</td>
<td>53.7</td>
<td>14.2%</td>
</tr>
</tbody>
</table>

Enforcing policy adds 6.7 milliseconds

- More performance numbers (indexing, microbenchmarks)
- Security evaluation

*Please see the paper!*
Policy compliance with Thoth: Contributions

- Declarative policies; directly attached to content
- Kernel-level compliance independent of application code
- Correct source policies ensure compliance regardless of the system internal complexity
Summary

- Declarative policies attached to conduits
  - Confidentiality, integrity, declassification

- Policy compliance despite application bugs or misconfigurations
  - Process-level IFC, policies-as-taint
  - Policy-specified/Typed declassification

- Efficient policy compliance
  - Low runtime overhead
  - Minimal application code changes

- Demonstrated utility with a distributed search engine

Questions?