UNIVERSITY AT ALBANY
State University of New York

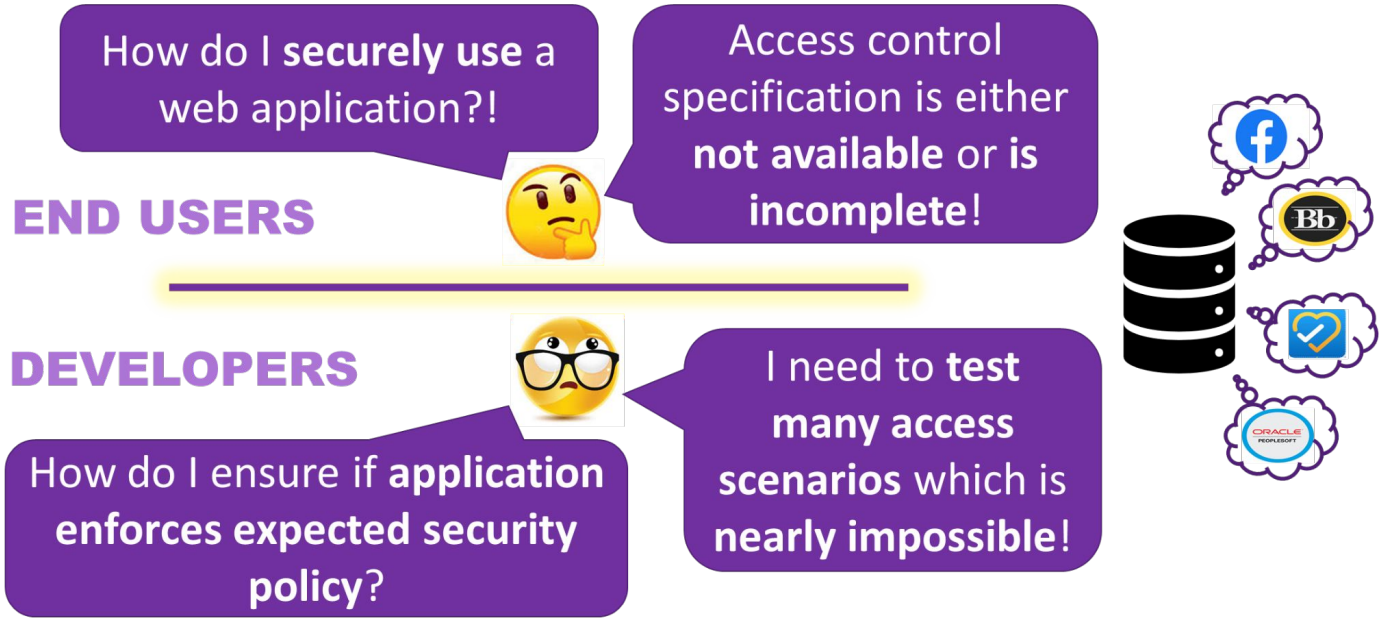
Towards Automated Learning of Access Control Policies Enforced by Web Applications

Padmavathi Iyer and Amir Masoumzadeh

28th ACM Symposium on Access Control Models and Technologies (SACMAT 2023)
Trento, Italy

Sponsored by NSF

Have you ever been overwhelmed with…



Our strategy to tackle such security/privacy concerns:

- Systematically infer access control policies enforced in web applications
- Inferred access control policies can be utilized by:
  - End users for understanding application's security behavior
  - Developers for validating implementation vs. specification

Previous works have …

- Focused on developing efficient algorithms for mining correct and concise access control policies
- Considered abstract systems with availability of inputs – low-level authorizations & application's data model

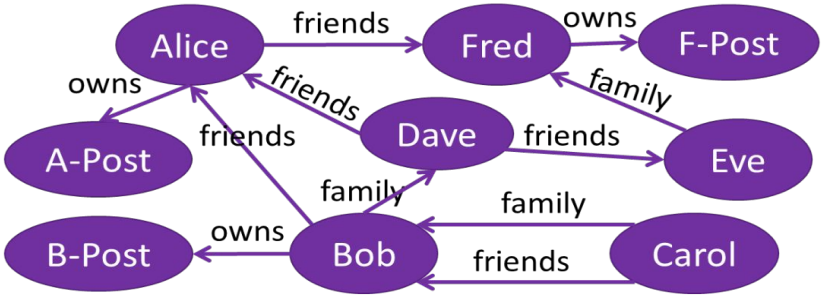
Relationship-Based Access Control Policy Mining

Low-level authorizations: Which user can access what resource

User	Resource	Decision
Alice	F-Post	Permit
Bob	A-Post	Permit
Carol	B-Post	Permit
Dave	F-Post	Deny
Eve	B-Post	Deny

+

Application's data model: System graph connecting users and resources



Policy Miner Algorithm →

- ✓ Users can access their own posts
- ✓ Users can access posts owned by their friends

Access Control Policy

High-level rules based on relationships between users and resources

Motivation

- Policy mining must be applicable to real-world systems
- Two inputs to mining process not readily available in real systems
- Need to infer the two mining inputs themselves

Challenges with concretizing policy mining

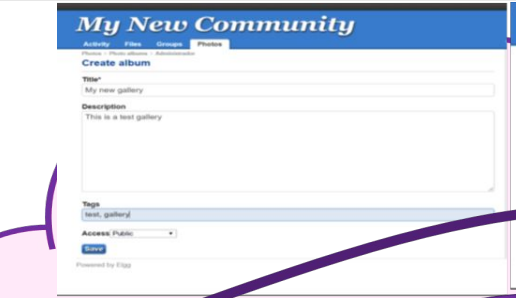
- Many real-world applications handle large amount and variety of data objects
  - *e.g., social networks support posts, comments, likes, and various content types like image and video*
- Different types of data objects may have different applicable policies
  - *Friends can see and comment on my posts*
  - *Friends of friends can only see my posts if they are also colleagues*

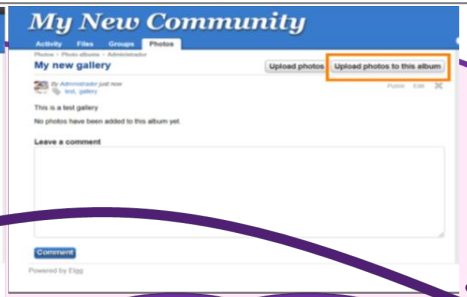
Contributions

- Inferring data model and low-level authorizations
  - Automating and observing user interactions with a web application
- Black-box view of application lets us observe its access control behavior as a whole
  - Helps overcome application's design complexities

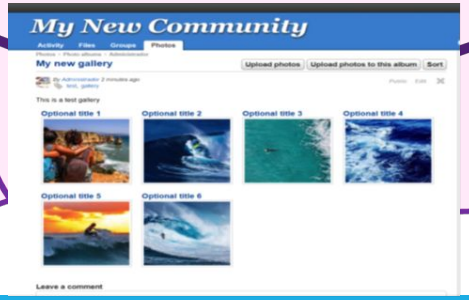
Our black-box inference strategy

By **observing client-server communications**, we can **identify information** about application users/resources…

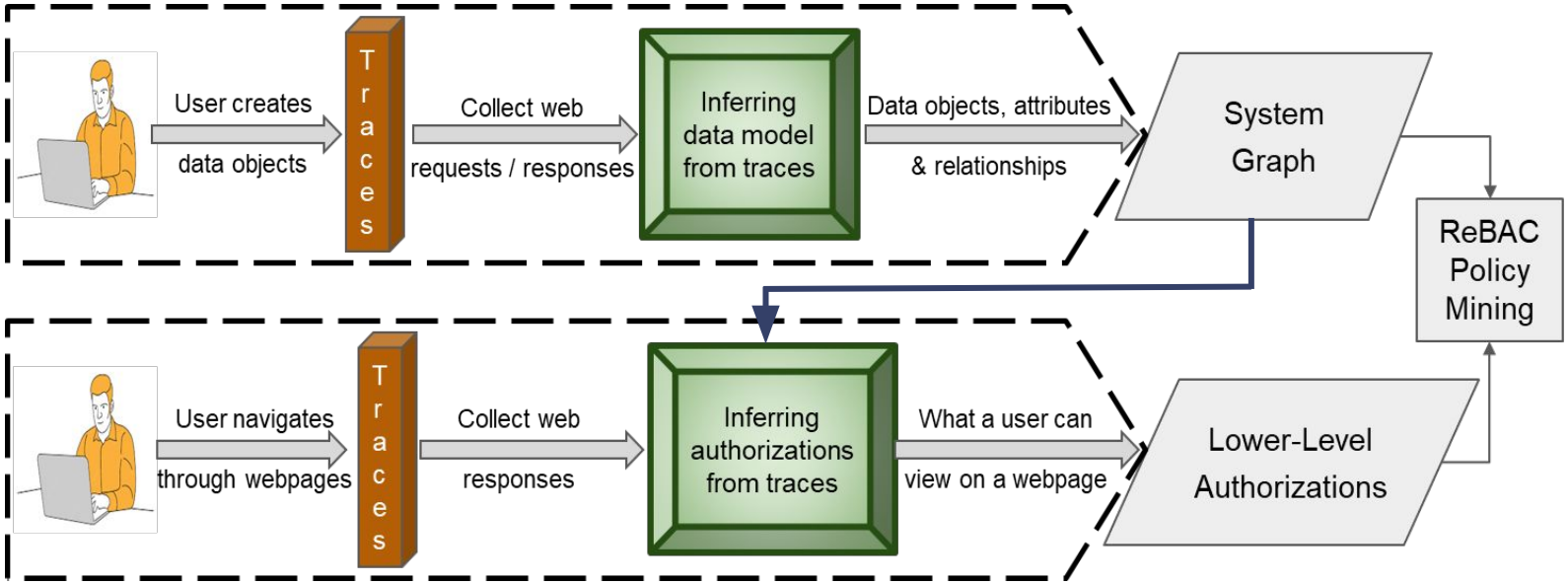




Client creates a photo album and server responds with page containing album's details



Applying ReBAC Mining to Infer Policies from Web Applications



Two-fold approach

Infer application's data model

Infer low-level authorizations using inferred data model

Two-fold approach - Infer Data Model

Infer application's data model

Identify object properties from client-server interactions

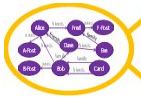
Cluster identified properties to prune spurious properties

Infer object relations using their distinguishing properties

Prune redundant object relationships using heuristics

Infer low-level authorizations using inferred data model

Two-fold approach - Infer Authorizations



Infer application's data model

Infer low-level authorizations using inferred data model

- Identify inferred data objects present in client-server interactions
- User permitted to view data model object & its properties if (s)he can view that object on any page of application; otherwise denied

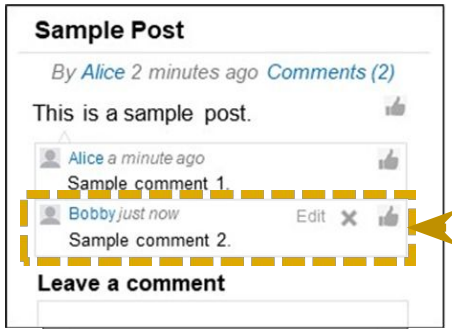
Inferring authorizations is not trivial!

A major challenge being Object-Reidentification!

- That is, how to correlate data elements viewed on a web page to data objects in inferred data model
- No abstract notion as data object; data objects usually characterized in terms of its corresponding attributes

Reidentifying Objects from Inferred Data Model

- Utilize unique properties of data objects to identify corresponding web page data with all those properties

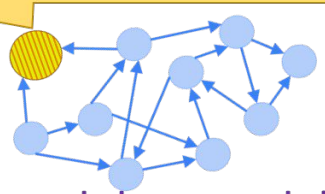


Carol views page corresponding to Alice's post

Re-identifying inferred data objects in page content viewed by Carol

Highlighted data object corresponds to second comment on current page

Inferred comment-object attributes:
Text = Sample comment 2.
Edit-URL = http://localhost/comment/edit/151
…



Inferred data model

Output Format

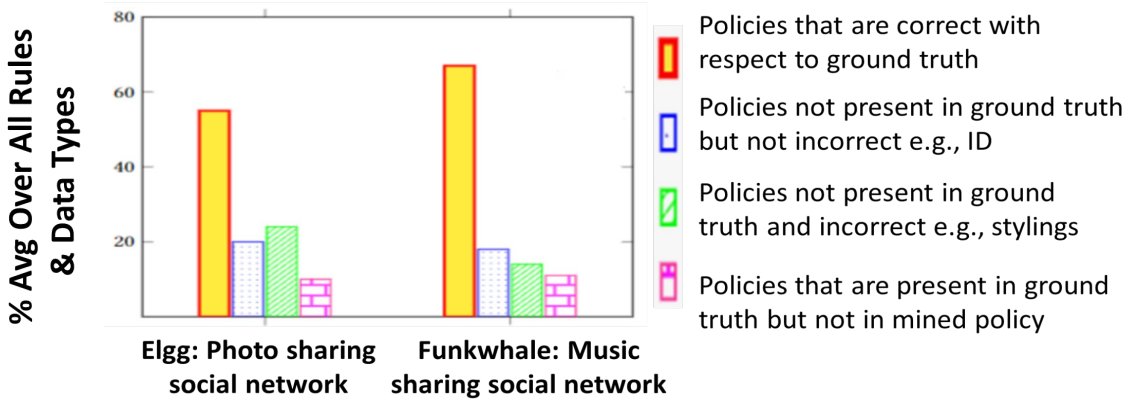
Object-Level Policy: Authorizations specified on data objects like posts and comments

- *e.g., users can access their own and their friends' posts*

Attribute-Level Policy: Authorizations specified on attributes of data objects like date-time, location, title

- *e.g., only users who own a post can see its location*

Experimental observations for mined attribute-level rules



- Ground truth based on what different users can see on their interfaces
- Able to mine most attribute-level policies in both applications
  - Failed in cases such as dates shown as “2 days ago” instead of absolute values
- Also mined policies for hidden attributes used by application to describe a data object
  - e.g., object-id present in the raw HTML to reference a post or a comment

Observations for mined attribute-level rules based on relationship patterns and data types

Application	Rule	Data Type	Mined	Ground-Truth (visible atts.)	$\|\Updownarrow\|$	$\|\Uparrow\|$	$\|\nearrow\|$	$\|\Downarrow\|$
Elgg	$[p]$	Post	27	19	16	6	5	3
	$[f, p]$	Post	23	15	12	6	5	3
	$[p, -b]$	Comment	20	12	11	3	6	1
	$[c]$	Comment	24	16	15	3	6	1
	$[f, p, -b] \wedge [f, c]$	Comment	20	12	11	3	6	1
Funkwhale	$[o, -i]$	Audio file	35	28	24	6	5	4
	$[w, -i]$	Audio file	29	23	20	6	3	3

Meaning of Relationships in Rules: b=belongs-to-post, c=owns-comment, f =friends-with, i=in-library, o=owns-library, p=owns-post, w=follows-library

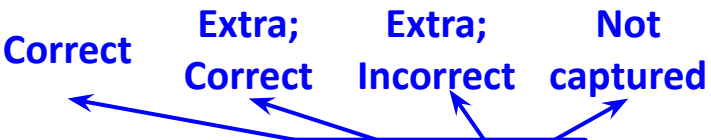
Observations for mined attribute-level rules based on relationship patterns and data types

Mined object-level rules

Application	Rule	Data Type	Mined	Ground-Truth (visible atts.)	|⇳|	|⇑|	|⇵|	|⇓|
Elgg	$[p]$	Post	27	19	16	6	5	3
	$[f, p]$	Post	23	15	12	6	5	3
	$[p, -b]$	Comment	20	12	11	3	6	1
	$[c]$	Comment	24	16	15	3	6	1
	$[f, p, -b] \wedge [f, c]$	Comment	20	12	11	3	6	1
Funkwhale	$[o, -i]$	Audio file	35	28	24	6	5	4
	$[w, -i]$	Audio file	29	23	20	6	3	3

Meaning of Relationships in Rules: b=belongs-to-post, c=owns-comment, f =friends-with, i=in-library, o=owns-library, p=owns-post, w=follows-library

Observations for mined attribute-level rules based on relationship patterns and data types



Application	Rule	Data Type	Mined	Ground-Truth (visible atts.)	⇳	⇑	⇵	⇓
Elgg	$[p]$	Post	27	19	16	6	5	3
	$[f, p]$	Post	23	15	12	6	5	3
	$[p, -b]$	Comment	20	12	11	3	6	1
	$[c]$	Comment	24	16	15	3	6	1
	$[f, p, -b] \wedge [f, c]$	Comment	20	12	11	3	6	1
Funkwhale	$[o, -i]$	Audio file	35	28	24	6	5	4
	$[w, -i]$	Audio file	29	23	20	6	3	3

Meaning of Relationships in Rules: b=belongs-to-post, c=owns-comment, f =friends-with, i=in-library, o=owns-library, p=owns-post, w=follows-library

Observations for mined attribute-level rules based on relationship patterns and data types

Application	Rule	Data Type	Mined	Ground-Truth (visible atts.)	$\|\Updownarrow\|$	$\|\Uparrow\|$	$\|\nexists\|$	$\|\Downarrow\|$
Elgg	$[p]$	Post	27	19	16	6	5	3
	$[f, p]$	Post	23	15	12	6	5	3
	$[p, -b]$	Comment	20	12	11	3	6	1
	$[c]$	Comment	24	16	15	3	6	1
	$[f, p, -b] \wedge [f, c]$	Comment	20	12	11	3	6	1
Funkwhale	$[o, -i]$	Audio file	35	28	24	6	5	4
	$[w, -i]$	Audio file	29	23	20	6	3	3

Meaning of Relationships in Rules: b=belongs-to-post, c=owns-comment, f =friends-with, i=in-library, o=owns-library, p=owns-post, w=follows-library

Observations for mined attribute-level rules based on relationship patterns and data types

Application	Rule	Data Type	Mined	Ground-Truth (visible atts.)	|⇳|	|⇑|	|⇵|	|⇓|
Elgg	$[p]$	Post	27	19	16	6	5	3
	$[f, p]$	Post	23	15	12	6	5	3
	$[p, -b]$	Comment	20	12	11	3	6	1
	$[c]$	Comment	24	16	15	3	6	1
	$[f, p, -b] \wedge [f, c]$	Comment	20	12	11	3	6	1
Funkwhale	$[o, -i]$	Audio file	35	28	24	6	5	4
	$[w, -i]$	Audio file	29	23	20	6	3	3

Meaning of Relationships in Rules: b=belongs-to-post, c=owns-comment, f =friends-with, i=in-library, o=owns-library, p=owns-post, w=follows-library

Observations for mined attribute-level rules based on relationship patterns and data types

Visible attributes of certain data object type to users satisfying the given object-level rule

Application	Rule	Data Type	Mined	Ground-Truth (visible atts.)	$\lvert\Updownarrow\rvert$	$\lvert\Uparrow\rvert$	$\lvert\nUparrow\rvert$	$\lvert\Downarrow\rvert$
Elgg	$[p]$	Post	27	19	16	6	5	3
	$[f, p]$	Post	23	15	12	6	5	3
	$[p, -b]$	Comment	20	12	11	3	6	1
	$[c]$	Comment	24	16	15	3	6	1
	$[f, p, -b] \wedge [f, c]$	Comment	20	12	11	3	6	1
Funkwhale	$[o, -i]$	Audio file	35	28	24	6	5	4
	$[w, -i]$	Audio file	29	23	20	6	3	3

Meaning of Relationships in Rules: b=belongs-to-post, c=owns-comment, f =friends-with, i=in-library, o=owns-library, p=owns-post, w=follows-library

Conclusion and Future Work

- Concretizing ReBAC mining to web applications
- Inferring data model and low-level authorizations by observing client-server interactions
- Experimented on two applications to show feasibility
- Future plans include:
  - Investigating further automation of trace generation
  - Experimenting on wider set of real-world applications