Role Discovery

Bill Horne, Prasad Rao, Rob Schreiber, Mehul Shah, Bob Tarjan (HP Labs)
Iver Band (HP IT)
Jason Rouault (HP Software)

2007 Interns:
Alina Ene (Princeton)
Nikola Milosavljevic (Stanford)
Outline

• Background
• Our Approach
• Complexity Results
• Lower Bounds
• Role Discovery Algorithms
• Results
• Next Steps
Role Based Access Control

• What is it?
  – An alternative to discretionary and mandatory access control, where users’ access to permissions is managed directly.
  – A role is a collection of permissions; users are assigned to roles

• Advantages
  – Aligned to business objectives of the organization
  – Rights defined once and applied to multiple recipients
  – Managing access changes for large groups of users
  – Managing individual user’s access as job roles change
What’s the problem?

- Migrating to RBAC is a huge challenge for large organizations
- The first step is *role engineering*
  - User Identification
    - Typically 10s of thousands in an enterprise
  - Resource Identification (e.g. applications)
    - Typically thousands
  - Constraint Analysis
    - e.g. segregation of duties
  - Design and Optimize
- This is a labor-intensive (expensive) process.
Role Discovery

- A bottom-up approach to discover roles that are implicit in an existing access control environment
  - Input: Existing access control rules
  - Output: A set of equivalent roles
- Goal:
  - Don’t replace role engineering
  - Provide tools to make the role engineering process more efficient
Benefits of Role Discovery

- Faster Results
  - Can help speed the role engineering process
  - Can migrate more of existing access controls to role based system
- Transparency
  - Provides the organization with a clear view of existing access controls.
  - Exposes “noise” in the system
- Lowers Risk
  - Lowers risk of business disruption and vulnerability introduction when role based system is deployed
Related Work

• Academic
  – Clustering (Schlegelmilch & Steffens 2005)
  – Complexity Results (Vaidya, et al 2007)
  – Merge and Split (Zhang, et al 2007)

• Commercial
  – Eurekify
  – Vaau
  – Bridgestream
Roadmap

• Background
• Our Approach
• Complexity Results
• Lower Bounds
• Role Discovery Algorithms
• Results
• Next Steps
Our Approach

Traditional Access Control

"bipartite" graph

users permissions

Role Based Access Control

"tripartite" graph

users roles permissions
Roles are bicliques

Therefore, discovering a set of roles to explain a set of access control rules is equivalent to covering the bipartite graph with a set of bicliques.
Two Goals

- Minimize total number of roles
  - Find the smallest biclique covering
- Minimize total number of edges
  - “Edge Concentration”
  - Find the biclique covering of minimum total order
Complexity Results

- Finding a minimum biclique cover is NP-complete (Orlin, 1977)
- Inapproximability (Simon, 1990)
  - The Minimum Biclique Cover problem is inapproximable in polynomial time within a factor $n^\delta$ for some constant $\delta > 0$, unless $P = NP$.
  - The Minimum Biclique Cover problem is inapproximable in polynomial time within a factor $n^{1-\varepsilon}$ for any constant $\varepsilon > 0$, unless $NP = ZPP$.
- Edge Concentration is NP-complete (Lin, 2000)
Roadmap

- Background
- Our Approach
- Complexity Results
- Lower Bounds
- Role Discovery Algorithms
- Results
- Next Steps
Lower Bound on Number of Roles

- **Max Independent Set**
  - Two edges \((a,b)\) and \((c,d)\) are independent if:
    - \(a, b, c,\) and \(d\) are distinct
    - not completely connected
  - Independent edges cannot be in the same biclique
  - N pairwise independent edges imply at least \(N\) bicliques in the cover

- **Finding the max independent set is also NP-complete**

- **Heuristic algorithm**
  - Run algorithm \(K\) times
    - Pick an edge randomly
    - Remove dependent edges
    - Iterate until graph empty
  - Choose largest independent set found
Lower Bounds on Number of Edges

- Only bound we know of is trivial
  - Total number of vertices (users + permissions)
Roadmap

- Background
- Our Approach
- Complexity Results
- Lower Bounds
- Role Discovery Algorithms
- Results
- Next Steps
Heuristic Algorithm for Biclique Cover

- Pick some node, n
  - e.g. a node with minimum degree
  - other ways to choose n are possible
- Find its set of neighbors, A.
- Find the intersection of A’s neighbors, B.
  - $n \in B$
- $(A,B)$ is a biclique, therefore a role
- Remove those edges from the graph and iterate.
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example
Edge Minimization

- For each role the number of edges is the sum of the number of users and permissions in the role.
Edge Minimization

\[
\text{#edges} = [(a+b) + (x+y)] \\
+ [(b+c) + (y+z)]
\]

\[
\text{#edges} = [(a+b) + x] \\
+ [(a+b+c) + y] + [(b+c) + z]
\]

Transform if \( y > a + b + c \)
**Edge Minimization**

\[
\text{Transform if } b > x + y + z
\]

\[
\text{#edges} = [(a+b) + (x+y)] + [(b+c) + (y+z)]
\]

\[
\text{#edges} = [a + (x+y)] + [b + (x+y+z)] + [c + (y+z)]
\]
Edge Minimization

- In certain degenerate cases, no increase in roles occurs

- Algorithm
  - Start with node minimization solution
  - Greedily substitute pairs of roles until no more gains possible
Roadmap

- Background
- Our Approach
- Complexity Results
- Lower Bounds
- Role Discovery Algorithms
- Results
- Next Steps
Application to Real World Problems

• HP IT Partner Connectivity
  - Allows external business partners to connect into internal HP systems
  - ~3,000 partner organizations
  - ~10,000 internal ipaddr/port pairs
  - ACLs on routers and firewalls

• Customer Application Entitlements
  - ~10,000 users
  - ~100 enterprise applications
  - ~1000s of finer grained permissions
  - Access control rules distributed across applications
## Sample Results

<table>
<thead>
<tr>
<th></th>
<th>dataset</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
</tr>
</thead>
<tbody>
<tr>
<td>#users</td>
<td>2044</td>
<td>3485</td>
<td>3477</td>
<td></td>
</tr>
<tr>
<td>#perms</td>
<td>1164</td>
<td>10127</td>
<td>1587</td>
<td></td>
</tr>
<tr>
<td>#edges</td>
<td>6841</td>
<td>185294</td>
<td>105205</td>
<td></td>
</tr>
<tr>
<td>role lower</td>
<td>453</td>
<td>390</td>
<td>172</td>
<td></td>
</tr>
<tr>
<td>bound</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>edge lower</td>
<td>3208</td>
<td>13612</td>
<td>5064</td>
<td></td>
</tr>
<tr>
<td>bound</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>role</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#roles</td>
<td>456</td>
<td>422</td>
<td>220</td>
<td></td>
</tr>
<tr>
<td>min</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#edges</td>
<td>4416</td>
<td>74568</td>
<td>8987</td>
<td></td>
</tr>
<tr>
<td>edge</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#roles</td>
<td>485</td>
<td>929</td>
<td>286</td>
<td></td>
</tr>
<tr>
<td>min</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#edges</td>
<td>3987</td>
<td>21968</td>
<td>8082</td>
<td></td>
</tr>
</tbody>
</table>
Next Steps

• The real problem is that most organization’s existing access controls are too complicated
• Discovered roles are difficult to interpret
• Possible Solutions
  – Approximate covers
  – Roles $\rightarrow$ Rules
    • Discovered roles are semanticless
    • Discover rules, based on user/permission attributes to describe roles
    • Dynamic roles
Thank you!

- Want to find out more?
  - william.horne@hp.com