Information Operation Across Infospheres: Assured Information Sharing

Prof. Bhavani Thuraisingham and Prof. Latifur Khan
The University of Texas at Dallas

Prof. Ravi Sandhu
George Mason University

August 2006
Acknowledgements

● Students
  - UTDallas
    ● Dilsad Cavus (MS, Data mining and data sharing)
    ● Srinivasan Iyer (MS, Trust management)
    ● Ryan Layfield (PhD, Game theory)
    ● Mehdi (PhD, Worm detection)
  - GMU
    ● Min (PhD, Extended RBAC)
● Faculty and Staff
  - UTDallas
    ● Prof. Murat (Game theory)
    ● Dr. Mamoun Awad (Data mining and Data sharing)
● Project supplemented by Texas Enterprise Funds
Architecture

Data/Policy for Federation

- Export Data/Policy
  - Component Data/Policy for Agency A
  - Component Data/Policy for Agency B
- Export Data/Policy
  - Component Data/Policy for Agency C
Our Approach

- Integrate the Medicaid claims data and mine the data; next enforce policies and determine how much information has been lost by enforcing policies.
- Examine RBAC and UCON in a coalition environment.
- Apply game theory and probing techniques to extract information from non-cooperative partners; conduct information operations and determine the actions of an untrustworthy partner.
- Defensive and offensive operations.
Data Sharing, Miner and Analyzer

- Assume N organizations.
  - The organizations don’t want to share what they have.
  - They hide some information.
  - They share the rest.
- Simulates N organizations which
  - Have their own policies
  - Are trusted parties
- Collects data from each organization,
  - Processes it,
  - Mines it,
  - Analyzes the results
Data Partitioning and Policies

- **Partitioning**
  - Horizontal: Has all the records about some entities
  - Vertical: Has subset of the fields of all entities
  - Hybrid: Combination of Horizontal and Vertical partitioning

- **Policies**
  - XML document
  - Informs which attributes can be released

- **Release factor:**
  - Is the percentage of attributes which are released from the dataset by an organization.
  - A dataset has 40 attributes.
    - “Organization 1” releases 8 attributes
    - RF = 8/40 = 20%
Example Policies

```xml
<?xml version="1.0"?>
<TEST_CASE>
  <BASE_POLICY_DIR>/data/policy/</BASE_POLICY_DIR>
  <!-- make sure to have different tc_id for the bundle -->
  <TC_ID>census_income_5</TC_ID>
  <TEST_CASE_DIR>testcases</TEST_CASE_DIR>
  <NUM_ORG>3</NUM_ORG>
  <RELEASE_FACTOR>5</RELEASE_FACTOR>
  <ATTRIB_XML>attributes.xml</ATTRIB_XML>
  <DATASET_BASE>/data/dataset/census_income/</DATASET_BASE>
  <MANDATORY_ATTRIB>income_type</MANDATORY_ATTRIB>
  <POLICY_XML>gen_org.xml</POLICY_XML>
  <ORG_PREFIX>org_</ORG_PREFIX>

  <!-- information about the dataset -->
  <DATASET_FN>census_income/census_income_50k.dat</DATASET_FN>
  <ARFF_PREFIX>census_income</ARFF_PREFIX>

  <!-- for each testcase bundle, used different test_case_id -->
  <TEST_CASE_ID>census_income_test_5</TEST_CASE_ID>
  <DATASET_PROCESSOR>
    <CLASS_NAME>processors.CensusIncomeProcessor</CLASS_NAME>
    <ATTRIB_FN>census_income/attributes.xml</ATTRIB_FN>
  </DATASET_PROCESSOR>
  <POLICY_DIR>census_policy</POLICY_DIR>
  <DELIM>,</DELIM>
  <TEMPLATE_FN>gen_template.xml</TEMPLATE_FN>
</TEST_CASE>
```
1. Load and Analysis.
   - loads the generated rules,
   - analyzes them,
   - displays in the charts.

2. Run ARM.
   - chooses the arff file
   - Runs the Apriori algorithm,
   - displays the association rules, frequent item sets and their confidences.

3. Process Data Set:
   - Processes the dataset using Single Processing or Batch Processing.
Extension For Trust Management

- Each Organization maintains a Trust Table for Other organization.
- The Trust level is managed based on the quality of Information.
- Minimum Threshold - below which no Information will be shared.
- Maximum Threshold - Organization is considered Trusted partner.
Role-based Usage Control (RBUC)

RBAC with UCON extension

- Users (U)
- Roles (R)
- Operations (OP)
- Objects (O)
- User-Role Assignment (URA)
- Permission-Role Assignment (PRA)
- Permissions (P)
- User Attributes (UA)
- Session Attributes (SA)
- Session Attributes (SA)
- Usage Decisions
- Authorizations (A)
- Obligations (B)
- Conditions (C)
RBUC in Coalition Environment

• The coalition partners may be trustworthy, semi-trustworthy, or untrustworthy, so we can assign different roles to the users (professor) from different infospheres, e.g.
  - professor role,
  - trustworthy professor role,
  - semi-trustworthy professor role,
  - untrustworthy professor role.

• We can enforce usage control on data by setting up object attributes to different roles during permission-role-assignment, e.g.
  - professor role: 4 times a day,
  - trustworthy role: 3 times a day
  - semi-trustworthy professor role: 2 times a day,
  - untrustworthy professor role: 1 time a day.
### Coalition Game Theory

**Players**

<table>
<thead>
<tr>
<th>P&lt;sub&gt;i&lt;/sub&gt;</th>
<th>( P_j )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tell Truth</strong></td>
<td>( A )</td>
</tr>
<tr>
<td><strong>Lie</strong></td>
<td>( B - M(p_j^i(\text{verify})) )</td>
</tr>
</tbody>
</table>

**Expected Benefit from Strategy**

- \( A \): Value expected from telling the truth
- \( B \): Value expected from lying
- \( M \): Loss of value due to discovery of lie
- \( L \): Loss of value due to being lied to
- \( p_j^i(\text{action}) \): Perceived probability by player \( i \) that player \( j \) will perform action
- \( \text{fake} \): Choosing to lie
- \( \text{verify} \): Choosing to verify
Coalition Game Theory

- Results
  - Algorithm proved successful against competing agents
  - Performed well alone, benefited from groups of likeminded agents
  - Clear benefit of use vs. simpler alternatives
  - Worked well against multiple opponents with different strategies

- Pending Work
  - Analyzing dynamics of data flow and correlate successful patterns
  - Setup fiercer competition among agents
    - Tit-for-tat Algorithm
    - Adaptive Strategy Algorithm (a.k.a. Darwinian Game Theory)
    - Randomized Strategic Form
  - Consider long-term games
    - Data gathered carries into next game
    - Consideration of reputation (‘trustworthiness’) necessary
Detecting Malicious Executables

The New Hybrid Model

What are malicious executables?

*Virus, Exploit, Denial of Service (DoS), Flooder, Sniffer, Spoof, Trojan etc.*

Exploits software vulnerability on a victim, May remotely infect other victims

**Malicious code detection: approaches**

**Signature based**: not effective for new attacks

**Our approach**: Reverse engineering applied to generate assembly code features, gaining higher accuracy than simple byte code features

Executable Files → Hex-dump → Byte-Codes → n-grams → Feature vector (n-byte sequences)

Select Best features using Information Gain → Replace byte-code with assembly code

Feature vector (Assembly code Sequences) → Machine-Learning

Malicious / Benign?