Less is More? Investigating the Role of Examples in Security Studies using Analogical Transfer

Hanan Hibshi
Apr 9, 2014

Joint Work with:
Ashwini Rao, Travis Breaux,
Jean-Michel Lehker, and Jianwei Niu
Introduction: Current Security Research

- User experiments: interviews, focus groups, surveys, etc.
- Purpose: evaluating a certain stimuli and its effects on security
  - Examples of stimuli: pop-ups, warnings, etc.
- Researchers may use a training phase to familiarize subjects with the tasks
- In the training phase, researchers rely on using related examples.
Motivation: Role of Examples in Security

- We use examples to teach and train users
- How do we choose examples
  - Number?
  - Type?
  - Does number and/or type have an effect?
- RQ: Can we “develop a theory” or “understand the science” behind choosing examples for security training?
Analogical Transfer Theories

• **Analogical transfer** is a problem solving strategy where a person applies knowledge gained from past examples to solve a problem\(^1\).

• Some mathematicians described that this is exactly how we solve problems.

Decomposing Problems

• **Schema**: structural details of the solution principle
  - *e.g. algorithm that the program implements*

• **Surface**: the superficial features of how we describe the problem
  - *e.g. language syntax of a computer program*
Different theories in research about role of surface and schema in stages.
- Surface only, schema only or both
- During Mapping, increase in schema information and decrease in surface along with increase in experience

In math: More experience, better problem solving by analogy
Analogical Transfer and Examples

- Schema induction: abstracting structural details from a set of examples
  - Schema induction DOES occur.
  - Unlikely to happen with one example
  - Minimum number of examples is disagreed upon

- Cognition load studies state:
  - Worked-out examples improve schema induction in novices
  - Individual differences can affect schema induction
  - Representation of a problem can affect cognitive load and hence, schema induction.
Applying Analogical Transfer to Security Learning

• Design a user study to test role of:
  • Surface
  • Schema.

• Two Hypothesis with two goals:
  • $H_1$: Identifying New Surface for a Schema
    • If users see more examples of the same schema, will they be able to extract the schema better?
  • $H_2$: Identifying New Schema
    • If users see examples of different schemas, would they be more able to identify new schemas?
Participants were randomly assigned to one of two groups:

- 3-Example: of same schema (H1)
- 2-Example: of different schema (H2)

The alphabets denote different schemas:

- A: Error Handling
- B: Checking input values
- C: Integer Security

The numbers denote different surfaces:

<table>
<thead>
<tr>
<th>Training Examples</th>
<th>3-Example Group</th>
<th>2-Example Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 (C)</td>
<td>A1 (C)</td>
<td></td>
</tr>
<tr>
<td>A2 (C)</td>
<td>B6 (C)</td>
<td></td>
</tr>
<tr>
<td>A3 (Java)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Problems</th>
<th>3-Example Group</th>
<th>2-Example Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>A4 (PHP)</td>
<td>A4 (PHP)</td>
<td></td>
</tr>
<tr>
<td>C7 (C)</td>
<td>C7 (C)</td>
<td></td>
</tr>
<tr>
<td>“A5”</td>
<td>“A5”</td>
<td></td>
</tr>
</tbody>
</table>
Choosing Examples and Problems

- Problems A4, A5 will evaluate different surfaces for same schema (H1)
- Problem C7 is a new schema (H2)
- Note that A4, and A5 has different representation of surface: (code vs. text)

<table>
<thead>
<tr>
<th></th>
<th>3-Example Group</th>
<th>2-Example Group</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training Examples</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1 (C)</td>
<td>A1 (C)</td>
<td></td>
</tr>
<tr>
<td>A2 (C)</td>
<td>B6 (C)</td>
<td></td>
</tr>
<tr>
<td>A3 (Java)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Test Problems</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4 (PHP)</td>
<td>A4 (PHP)</td>
<td></td>
</tr>
<tr>
<td>C7 (C)</td>
<td>C7 (C)</td>
<td></td>
</tr>
<tr>
<td>“A5”</td>
<td>“A5”</td>
<td></td>
</tr>
</tbody>
</table>
Code Snippet – A1

```c
/* function to copy network data to a file */
int copyData ( struct connection *c, FILE *fp) {
    char buf [200];
    readFromNetwork (c, buf , 100) ;
    writeToFile (fp , buf , 100) ;
    /* ... */
}
```

Training Problem A1

- **Question 1:** What is the problem, if any?
- **Question 2:** What are the consequences of the problem, if any?
- **Question 3:** What changes do you make to prevent the problem, if any?
```c
int getValueFromArray ( int *array , int len , int index )
{
    int value ;
    /* check that array index is less than array’s maximum length */
    if ( index < len ) {
        /* get the value at the specified array index */
        value = array [ index ];
    } else {
        /* if array index is invalid then output error message */
        printf ( "error : index %d invalid .\n", index );
        value = -1;
    }
    return value ;
}
```
Subjects

- We chose participants who took at least one security class.
- Recruited participants in security classes at different universities: CMU, NCSU, and UTSA
- We offered $5,$10 gift cards, depending on performance
Grading Responses

• Answers were open-ended, essay style.

• Two graders had an answer key to code the answers:
  • 0: unintended answer, 1: intended answer

• Grounded analysis for extra answers:
  • Incorrect
  • Extended-out-of-context
  • Extended-in-context

• Inter-rater Reliability: Cohen’s Kappa = 0.8
RESULTS OF STUDY
A total of 124 students
- 80 participants completed the study to the end.
- The 3-Example group had 54 participants (33 complete, 21 incomplete),
- The 2-Example group had 70 participants (47 complete, 23 incomplete)

Our major analysis is for the 80 completed responses. We discuss the drop outs separately.
Participants Education Level

- 52 students with graduate degree or higher
  - (20 in group 3-Example and 32 in group 2-Example)
- 16 with 3-4 years of college higher
  - (8 in group 3-Example and 8 in group 2-Example).
- 12 students had only 1-2 years of college experience
  - (5 in group 3-Example and 7 in group 2-Example)
## Computer Courses

<table>
<thead>
<tr>
<th>Course</th>
<th>3-Example</th>
<th></th>
<th>2-Example</th>
<th></th>
<th>Total</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#</td>
<td>%</td>
<td>#</td>
<td>%</td>
<td>#</td>
<td>%</td>
</tr>
<tr>
<td>Programming Languages</td>
<td>30</td>
<td>38%</td>
<td>45</td>
<td>56%</td>
<td>75</td>
<td>94%</td>
</tr>
<tr>
<td>Data Structures</td>
<td>26</td>
<td>33%</td>
<td>42</td>
<td>53%</td>
<td>68</td>
<td>85%</td>
</tr>
<tr>
<td>Computer Security</td>
<td>15</td>
<td>19%</td>
<td>25</td>
<td>31%</td>
<td>40</td>
<td>50%</td>
</tr>
<tr>
<td>Secure Programming</td>
<td>5</td>
<td>6%</td>
<td>6</td>
<td>8%</td>
<td>11</td>
<td>14%</td>
</tr>
</tbody>
</table>

* Percentages are calculated to the total number of participants in the sample (80)
## Industry Programming Experience

<table>
<thead>
<tr>
<th>Experience</th>
<th>3-Example</th>
<th>2-Example</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#</td>
<td>%</td>
<td>#</td>
</tr>
<tr>
<td>No Experience</td>
<td>3</td>
<td>4%</td>
<td>14</td>
</tr>
<tr>
<td>Less than 1</td>
<td>7</td>
<td>9%</td>
<td>13</td>
</tr>
<tr>
<td>One Year</td>
<td>2</td>
<td>3%</td>
<td>13</td>
</tr>
<tr>
<td>Between 1 and 3</td>
<td>3</td>
<td>4%</td>
<td>10</td>
</tr>
<tr>
<td>Three or More</td>
<td>2</td>
<td>3%</td>
<td>13</td>
</tr>
</tbody>
</table>

*Percentages are calculated to the total number of participants in the sample (80)*
Programming Languages Experience

- C/C++
- Java
- Perl, PHP, Python, Ruby
- SQL

Percentage of participants in total sample:

- 3 years or more
- Between 1 and 3 years
- 1 Year
- Less than One Year
- No Experience
Completion Time

- We recorded the overall completion time participants took to complete the survey (start to finish)
- Our sample does not have a normal distribution
- Used Wilcoxon Mann Whitney-U test ($p = 0.56$)
- Median (excluding 7 outliers out of the 80 values)
  - 3-Example: 23min
  - 2-Example: 22 min
- Limitation: we didn’t record time for each task.
### Performance Results

*Yes: Intended answer, No: Not-intended answer

<table>
<thead>
<tr>
<th></th>
<th>3-Example</th>
<th></th>
<th>2-Example</th>
<th></th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>More surfaces per schema</td>
<td>30</td>
<td>3</td>
<td>39</td>
<td>8</td>
<td>0.51</td>
</tr>
<tr>
<td>More Schemas</td>
<td>18</td>
<td>15</td>
<td>21</td>
<td>26</td>
<td>0.50</td>
</tr>
<tr>
<td>Different surface representation</td>
<td>11</td>
<td>22</td>
<td>14</td>
<td>33</td>
<td>0.80</td>
</tr>
</tbody>
</table>

- Fisher exact test was applied on the contingency tables.
- More surfaces didn’t improve performance of 3-Example group.
- More schemas didn’t improve the 2-Example group.
- Different representation dropped performance but no difference between the groups
  - We saw more “out-of-the-box” ideas for Problem A5 (text)
### Overall Performance

<table>
<thead>
<tr>
<th></th>
<th>3-Example</th>
<th>2-Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>No correct problem</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>One correct problem</td>
<td>11</td>
<td>17</td>
</tr>
<tr>
<td>Two correct problems</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>Three correct problems</td>
<td>3</td>
<td>7</td>
</tr>
</tbody>
</table>

- (p-value = 0.13, Fisher’s exact test)
### Performance and Experience

<table>
<thead>
<tr>
<th>3-Example</th>
<th>Test Problem A4</th>
<th>Test Problem C7</th>
<th>Test Problem A5</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Experience</td>
<td>Yes: 4, No: 0</td>
<td>Yes: 2, No: 2</td>
<td>Yes: 0, No: 4</td>
</tr>
<tr>
<td>Less than 1</td>
<td>Yes: 9, No: 0</td>
<td>Yes: 5, No: 4</td>
<td>Yes: 3, No: 6</td>
</tr>
<tr>
<td>One Year</td>
<td>Yes: 6, No: 1</td>
<td>Yes: 3, No: 4</td>
<td>Yes: 2, No: 5</td>
</tr>
<tr>
<td>Between 1 and 3</td>
<td>Yes: 6, No: 1</td>
<td>Yes: 4, No: 3</td>
<td>Yes: 4, No: 3</td>
</tr>
<tr>
<td>Three or More</td>
<td>Yes: 4, No: 2</td>
<td>Yes: 4, No: 2</td>
<td>Yes: 2, No: 4</td>
</tr>
<tr>
<td>2-Example</td>
<td>Test Problem A4</td>
<td>Test Problem C7</td>
<td>Test Problem A5</td>
</tr>
<tr>
<td>No Experience</td>
<td>Yes: 12, No: 1</td>
<td>Yes: 5, No: 8</td>
<td>Yes: 3, No: 10</td>
</tr>
<tr>
<td>Less than 1</td>
<td>Yes: 9, No: 2</td>
<td>Yes: 6, No: 5</td>
<td>Yes: 5, No: 6</td>
</tr>
<tr>
<td>One Year</td>
<td>Yes: 6, No: 2</td>
<td>Yes: 4, No: 4</td>
<td>Yes: 4, No: 4</td>
</tr>
<tr>
<td>Between 1 and 3</td>
<td>Yes: 6, No: 0</td>
<td>Yes: 3, No: 3</td>
<td>Yes: 0, No: 6</td>
</tr>
<tr>
<td>Three or More</td>
<td>Yes: 6, No: 3</td>
<td>Yes: 3, No: 6</td>
<td>Yes: 2, No: 7</td>
</tr>
</tbody>
</table>
## Drop Out

<table>
<thead>
<tr>
<th></th>
<th>3-Example</th>
<th>2-Example</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before treatment assignment</td>
<td>unknown</td>
<td>unknown</td>
<td>5</td>
</tr>
<tr>
<td>Before seeing examples</td>
<td>7</td>
<td>6</td>
<td>13</td>
</tr>
<tr>
<td>After 1st example</td>
<td>4</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>After 2nd example and before 3rd</td>
<td>2</td>
<td>N/A</td>
<td>2</td>
</tr>
<tr>
<td>After all examples and before questions</td>
<td>6</td>
<td>11</td>
<td>17</td>
</tr>
<tr>
<td>After 1st test problem</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>After 2nd test problem</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

- **Total drop out: 49 participants:**
  - 21 from the 3-Example, 23 from the 2-Example, and
  - 5 undetermined who dropped immediately after the consent
Important Observations

- We didn’t see a statistical difference but there are important observations:
  - Participants had similar backgrounds, and cognition studies has shown the effect of experience on analogical transfer
  - More surfaces for a given schema didn’t make a difference. In security, maybe it is not very helpful to show multiple surfaces of same schema (example: phishing?)
  - More schemas didn’t improve performance either, but we can’t rule out the learning effect.
  - Text representation affected both groups:
    - Different representation of schemas could make a difference
    - Future studies could help.
Threats to validity

• **Construct Validity**
  • Possible confounders: lack of randomization of test problems and examples, fatigue, learning effect.

• **Internal Validity**
  • Possible confounders: motivation level of participants, past exposure to similar training examples or test questions, and cognitive load induced by problem representation format

• **External Validity**
  • Our current results can’t be generalized.
Conclusions

• The goal of this study was to examine a possible way that we can improve security training and education.

• Further studies can be conducted to see how cognitive theories may apply to security learning and analysis.

• Security problems could be different than math!
  • Maybe the theory of analogical transfer will be different when applied to security.

• We need to think about the science behind the examples we choose!
Questions?

- Research funded by Army Research Office - (W911NF-09-1-0273)

- Hanan Hibshi
  - hhibshi@cmu.edu
  - http://www.cs.cmu.edu/~hhibshi/