Hidden in Plain Sight:

Automatically Identifying Security Requirements from Natural Language Artifacts

Maria Riaz, **Jason King**, John Slankas, Laurie Williams

Aug 28th, 2014

NC STATE UNIVERSITY
Real Research Group
Software Engineering @ NCSU

Agenda

- Motivation
- Research Goal
- Related Work
- Security Discoverer (SD) Process
- Security Requirements Templates
- Evaluation of SD Process
- Contributions

Motivation

Cert Research Report, 2010

- Security requirement among the lower 50% of prioritized requirements
- Difficult and expensive to improve security of an application once it is in operational environment

Building security in [McGraw06]

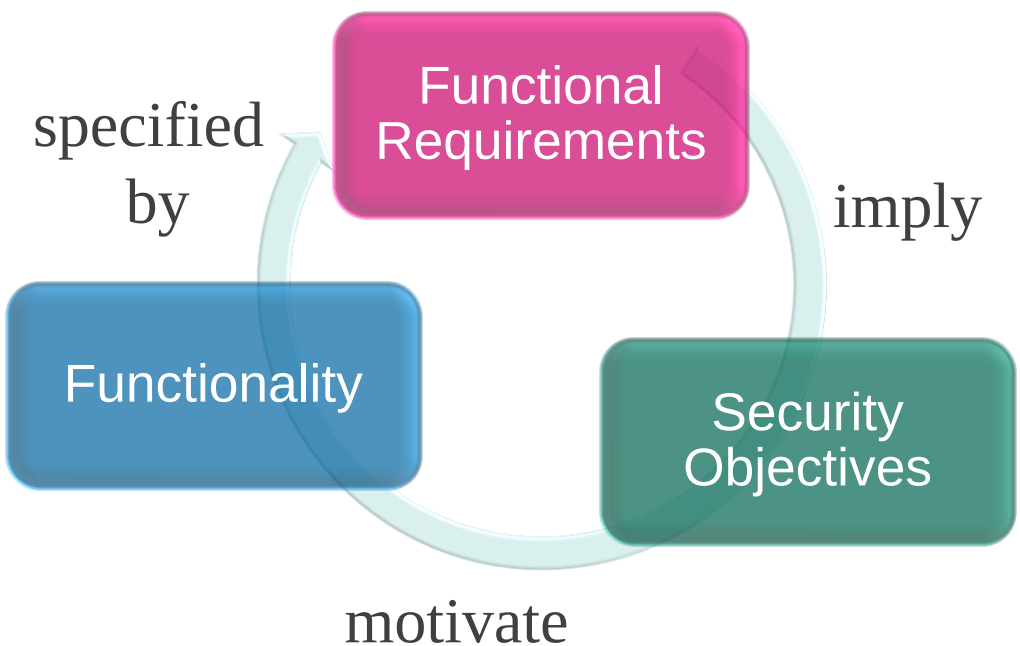
- Need to improve the quantity and quality of security requirements identified early on.

http://resources.sei.cmu.edu/asset_files/CERTResearchReport/2011_013_001_37704.pdf

Motivation

- Natural language requirements artifacts often ***explicitly*** state some security requirements.

- Additional sentences may have security ***implications,*** leading to additional requirements.



Research Goal

To aid requirements engineers in producing a more **comprehensive** and **classified set of security requirements** by:

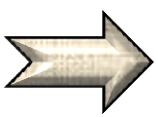
1) automatically ***identifying security-relevant sentences*** in natural language requirements artifacts, and

2) providing ***context-specific security requirements templates*** to help translate the security-relevant sentences into functional security requirements.

Overview

- Input: Natural language **requirements artifacts** (requirements specification, use case scenarios, user stories)

"HCPs can return to an office visit and modify or delete the fields of the office visit."



- Output: **Security requirements** for the system inferred from security-relevant sentences in the input

[ID & Authentication] Each user should be assigned a unique identifier that can be used for the purpose of authentication.

[Confidentiality] The system shall enforce access privileges that enable HCP to modify or delete office visit.

[Integrity] The system shall ensure that deletion of office visit is performed in accordance with the retention policy.

[Accountability] The system shall log every time HCP modifies or deletes office visit.

[Privacy] The system shall allow the owner of office visit to be notified when the office visit is modified or deleted by HCP.

Related Work

Identifying security requirements:

- Security requirements engineering [Square05]
  - Process for identifying security requirements
- Reusable security requirements and patterns [Toval02, Firesmith04, Schumacher06, Withall07]
  - Parameterized security requirements
  - Patterns for some aspects of access control and audit
- Organizational learning approach to security [Schneider12]
  - Reusing explicitly stated security requirements

Related Work

Natural language requirements classification:

- Automated classification of non-functional requirements [Cleland-Huang07]
  - Use of indicator terms; recall (81%); precision (12%);
- Automated extraction of non-functional requirements in available documentation [Slankas13-Nat]
  - Multiple algorithms; recall (54%); precision (73%);
- Access control policy extraction from unconstrained natural language text [Slankas13-Pass]
  - Sentence structure matching (k-NN classifier); Otherwise majority vote (naïve Bayes and SVM classifiers); recall (91%); precision (87%);

Security Discoverer (SD) Process

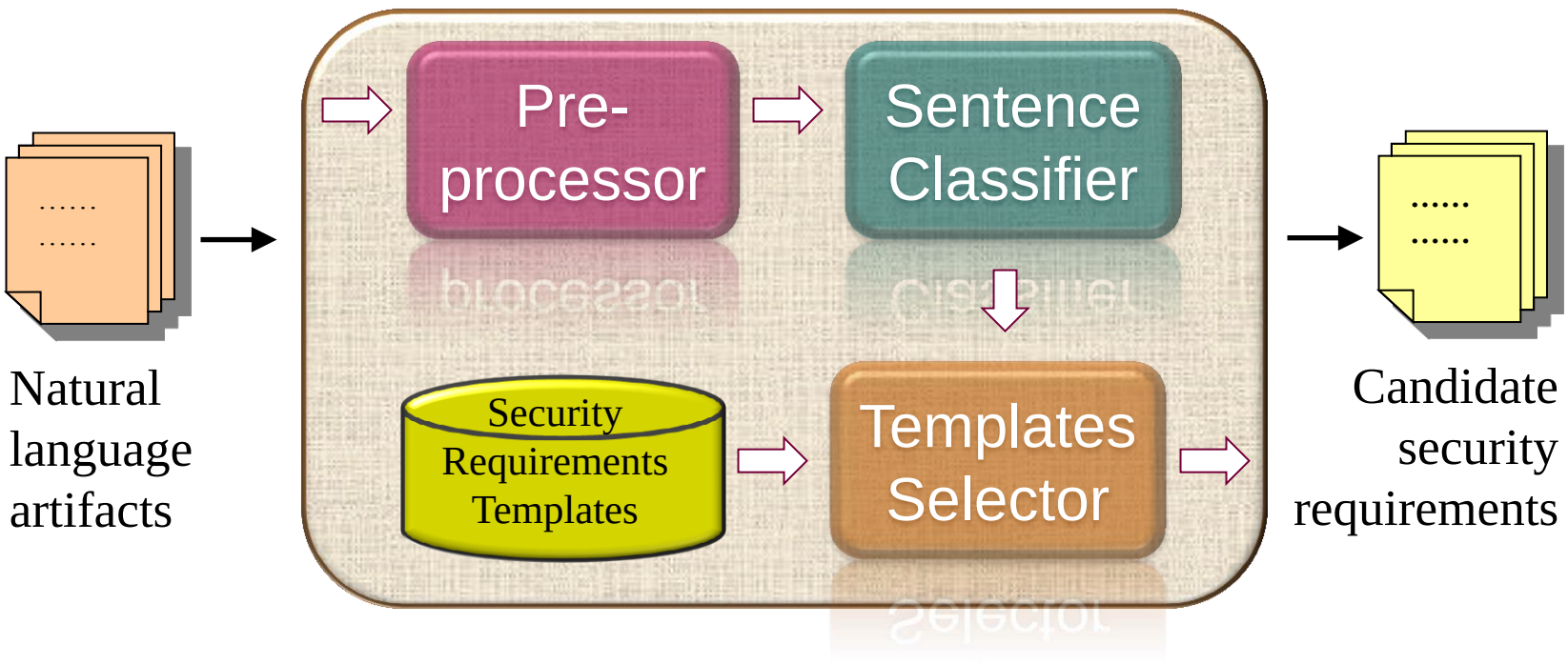
1-Parse Natural Language Requirements Artifacts

2-Identify Security-Relevant Sentences

3-Suggest Security Requirements Templates

4-Instantiate Selected Templates

5-Generate Security Requirements Document



SD Process
Pre-process Artifacts

Identify and parse individual sentences in natural language requirements artifacts

- ***Parts of speech tags***: can be used to instantiate templates or even group requirements by actors / resources / actions.

Example Sentence

"The **system** shall ***provide*** the **ability** to ***update*** a **patient history** by ***modifying***, ***adding*** or ***removing*** **items** from the **patient history** as appropriate."

nouns ***verbs***

SD Process
Security Objectives for Requirements Classification

Confidentiality (C)	• The degree to which the "data is **disclosed only as intended**“ [Schumacher06]
Integrity (I)	• "The degree to which a system or component **guards against improper modification or destruction** of computer programs or data." [FIPS-PUB-199]
Availability (A)	• "The degree to which a system or component is **operational and accessible** when required for use." [IEEE]
Identification & Authentication (IA)	• The need to establish that "a **claimed identity is valid**" for a user, process or device. [NIST-SP800-33]
Accountability (AY)	• Degree to which **actions** affecting software assets "can be **traced to the actor** responsible for the action“ [Schumacher06]
Privacy (PR)	• The degree to which “an actor can **understand and control** how their information is used.” [RE14]

SD Process
Security Objectives for Requirements Classification

Example Sentence

"The system shall provide the ability to **update** a **patient history** by **modifying, adding or removing items** from the patient history **as appropriate**."

Security Objectives

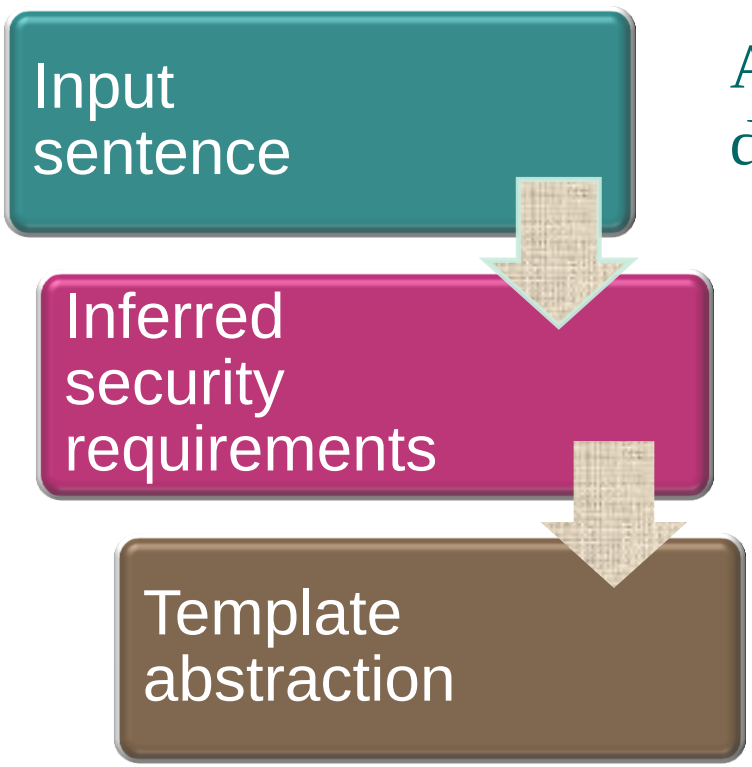
Confidentiality (disclosure)

Integrity (access / modification)

Accountability (trace actions)

Security Requirements Templates

Identifying common templates for specifying functional security requirements.



An HCP chooses to document an office visit.

The HCP may also add a patient referral.

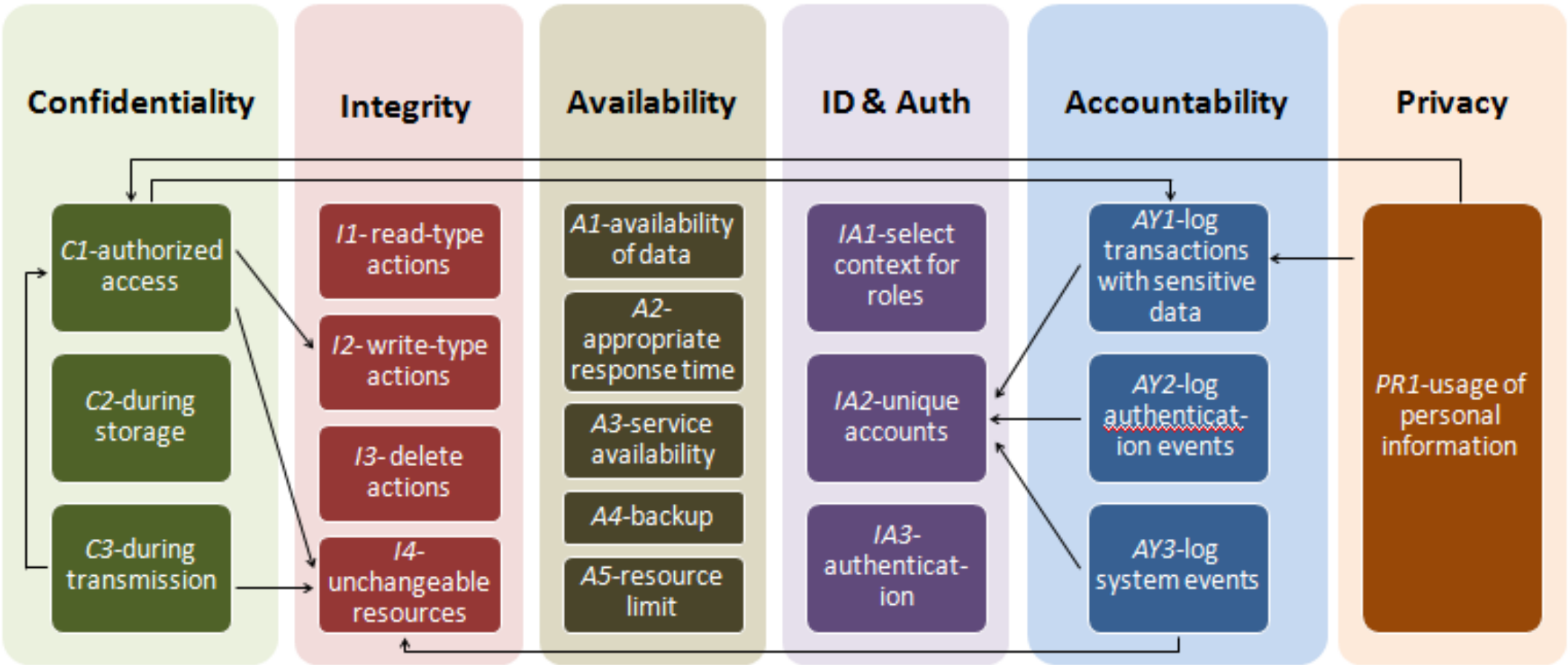
The system shall allow the owner of office visit to be notified when the office visit is documented by HCP.

The system shall allow the owner of patient referral to be notified when the patient referral is added by HCP.

"The system shall allow the owner of **<resource>** to be notified when the **<resource>** is **<action>** by **<subject>**"

Security Requirements Templates

- Extracted **19 context-specific** security requirements templates [*Empirically derived from security-relevant sentences*]



SD Process
Generating Security Requirements from Templates

Example Sentence

"The system shall provide the ability to **update** a **patient history** by **modifying, adding or removing items** from the patient history **as appropriate**."

Generated Security Requirements [Integrity-I2]

- The system shall ensure that all **mandatory information is provided** for the **<patient history>** before **<modifying, adding or removing items>**.
- The system shall have provision to **correct errors** in **<patient history>** if errors are detected.

……

[**see AY1**: Logging transactions with sensitive data]

SD Process Evaluation

Study Oracle for Supervised Learning

Sentences

Doc. ID	Document Title	# Total	# Explicit	# Implicit	# None
CT	Certification Commission for Healthcare Information Technology (CCHIT) Certified 2011 Ambulatory EHR Criteria	331	89 (27%)	236 (71%)	6 (2%)
ED	Emergency Department Information Systems Functional Document	2328	274 (12%)	1281 (55%)	773 (33%)
NU	Pan-Canadian Nursing EHR Business and Functional Elements Supporting Clinical Practice	264	41 (16%)	127 (48%)	96 (36%)
OR	Open Source Clinical Application Resource (OSCAR) Feature Requests	5081	174 (3%)	1172 (23%)	3735 (74%)
PS	Canada Health Infoway Electronic Health Record (EHR) Privacy and Security Requirements	1623	628 (39%)	67 (4%)	928 (57%)
VL	Virtual Lifetime Electronic Record User Stories	1336	185 (14%)	776 (58%)	375 (28%)
	Total	**10963**	**1391 (13%)**	**3659 (33%)**	**5913 (54%)**

https://www.cchit.org/ http://www.hl7.org/
https://www.infoway-inforoute.ca/ http://oscarcanada.org/
https://www.infoway-inforoute.ca/ http://www.va.gov/vler/

SD Process Evaluation
Security Objectives in the Study Oracle

Breakdown of security objectives in the oracle:

C	I	A	IA	AY	PR	None
27%	30%	~1%	~2%	34%	2%	54%

Frequently occurring groups of security objectives:

# (% sec-relevant)	Objective Groups
2232 (44%)	Confidentiality, Integrity, Accountability
702 (14%)	Integrity, Accountability
443 (9%)	Confidentiality, Accountability
106 (2%)	Confidentiality, Integrity
104 (2%)	Confidentiality, Identification & Authentication

SD Process Evaluation
Automatic Classification of Sentences

10-fold cross validation:

- Divide sentences in the oracle into 10 subsamples; Train on 9, test on the 10th, using each subsample once for validation.
- Each sentence used for both training and validation.

Supervised machine learning:

- **Naïve Bayes**: simple; does not consider sentence structure; needs small training set;
- **SMO** (*sequential minimal optimization*): train models for recognizing patterns in the input; less complex;
- ***k*-NN classifier**: simple; considers sentence structure; improves with larger training set;

SD Process Evaluation
Automatic Classification of Sentences

Correctly predicted and classified **82%** of security objectives for all the sentences (***precision***)

- ***18%*** *of the identified objectives an analysts examines would be* ***false positives***

Classifier	Precision	Recall	F Measure
Naïve Bayes	.66	.76	.71
SMO	.81	.76	.78
k-NN (*k*=1)	.80	.76	.78
Combined	.82	.79	.80

Identified **79%** of all objectives implied by sentences within the documents (***recall***)

- ***21%*** *of the possible objectives not found i.e.,* ***false negatives***

SD Process Evaluation
Automatically Suggested Templates

- In a separate user study, we evaluated the use of automatically suggested templates in generating security requirements:
  - *Found templates to be helpful in **considering more security objectives** as compared to a control group.*
  - *Found templates to be helpful in **identifying significantly more security requirements** (2-3 times) as compared to a control group.*

Contributions

- Facilitate security requirements engineering
  - *Set of context-specific security requirements templates*
  - *Tool-assisted process for generating requirements*
  - *Empirical evaluation of tool and process*
- A classified set of sentences for the healthcare domain

References

[Cleland-Huang06] J. Cleland-Huang, R. Settimi, X. Zou, and P. Solc, “Automated Classification of Non-functional Requirements,” *Requirements Engineering*, vol. 12, no. 2, pp. 103–120, Mar. 2007.

[Firesmith04] D. Firesmith, "Specifying Reusable Security Requirements," *Jornal of Object Technology,* vol. 3, p. 15, Jan-Feb. 2004.

[McGraw06] G. McGraw. “Software Security: Building Security In”, Addison Wesley Professional, 2006.

[Schneider12] Kurt Schneider, Eric Knauss, Siv Houmb, Shareeful Islam, and J. Jürjens, "Enhancing security requirements engineering by organizational learning," *Requirements Engineering,* vol. 17, pp. 35-56, 2012.

[Schumacher06] M. Schumacher, E. Fernandez-Buglioni, D. Hyberston, F. Buschmann, and P. Sommerlad, *Security Patterns: Integrating Security and Systems Engineering*. West Sussex: John Wiley & Sons, Ltd, 2006.

[Slankas13-Nat] J. Slankas and L. Williams, "Automated Extraction of Non-functional Requirements in Available Documentation", *1st International Workshop on Natural Language Analysis in Software Engineering* (NaturaLiSE 2013), San Francisco, CA.

[Slankas13-Pass] J. Slankas and L. Williams, "Access Control Policy Extraction from Unconstrained Natural Language Text", *2013 ASE/IEEE International Conference on Privacy, Security, Risk, and Trust* (PASSAT 2013), Washington D.C., USA, September 8-14, 2013.

[Square05] N. R. Mead, E. D. Houg, and T. R. Stehney, "Security Quality Requirements Engineering (SQUARE) Methodology," Software Engineering Inst., Carnegie Mellon University2005.

[Toval02] A. Toval, J. Nicolar, et al. (2002). "Requirements Reuse for Improving Information Systems Security: A Practitioner’s Approach." *Requirements Engineering* 6(4): 15.

[Withall07] Withall, S. (2007). Software Requirement Patterns, Microsoft Press.

Thank you!

mriaz@ncsu.edu jtking@ncsu.edu jbslanka@ncsu.edu