ASTQB
American Software Testing Qualifications Board, Inc.

Defining Solid Software Security Requirements

Randall W. Rice, CTFL, CTFL-AT, CMT, CTAL(Full), CTAL-SEC, CTAL-TAE, CFLBA

Director, ASTQB

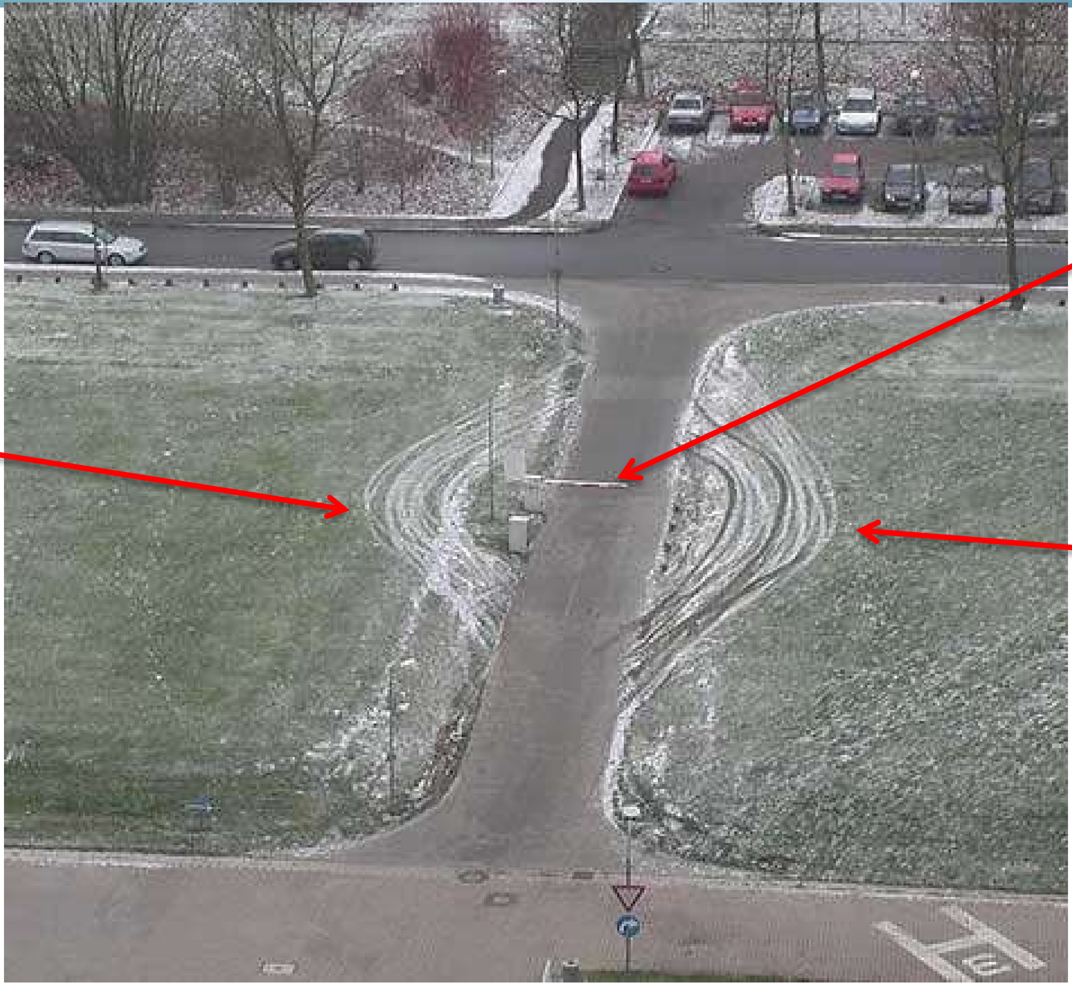
"If people don't know what they want, no development process - no matter how exact, how clever, or how efficient - will satisfy them." Gerald Weinberg

"There's no sense being exact about something if you don't even know what you're talking about." John von Neumann

The Challenges

- Security threats are ever-changing
  - Not just evolving
- There are 806 (at present count) common weaknesses (CWEs)
  - www.mitre.org
  - However, there is a "Top 25" list
    - https://cwe.mitre.org/top25/index.html

ASTQB American Software Testing Qualifications Board, Inc.


Attack #1 (such as SQL Injection)

Security Feature

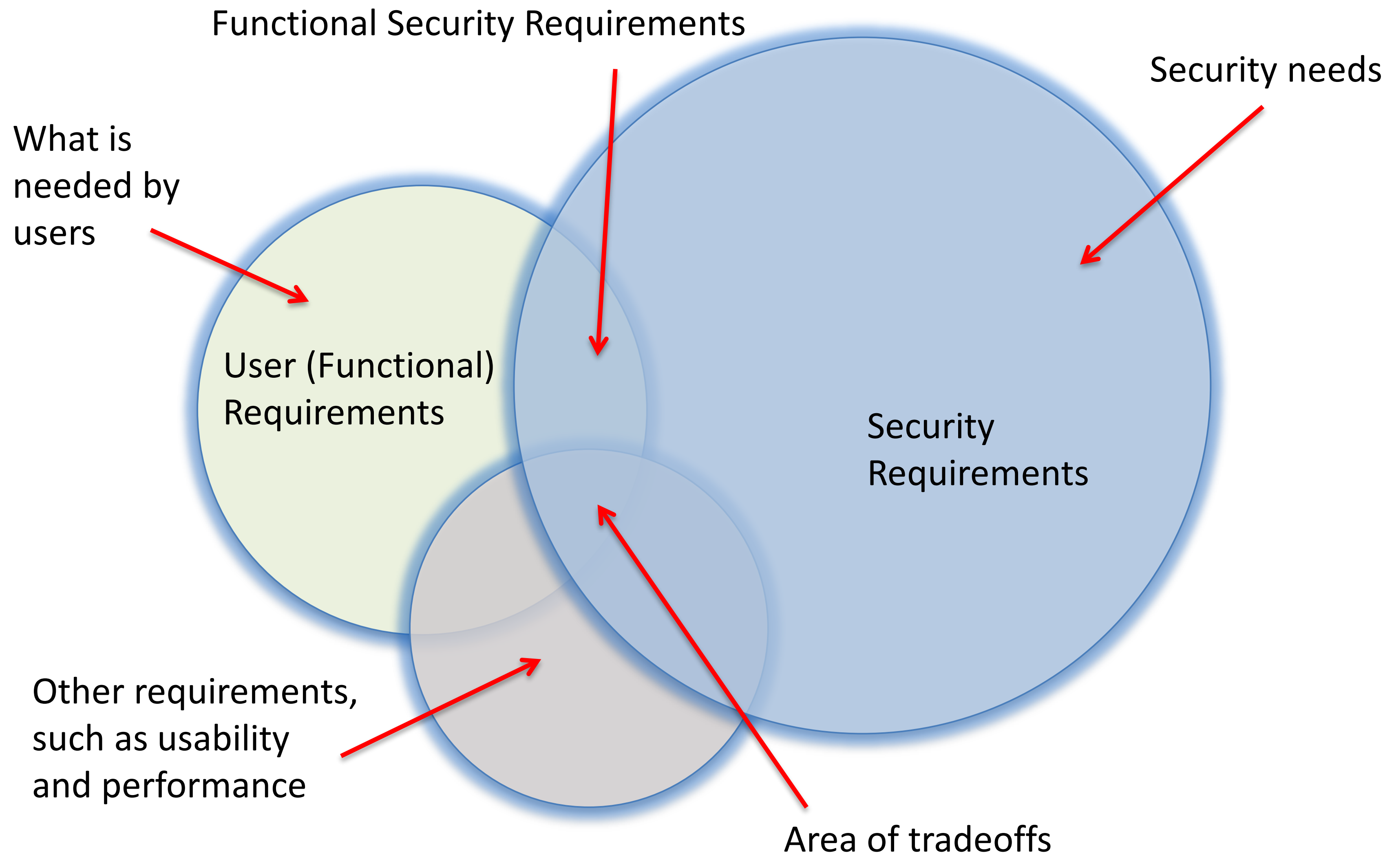
Attack #2 (such as Cross-site scripting)

Suggested Requirement: A permanent barrier of sufficient height and strength to prevent entry by vehicles must be installed around the complete perimeter on both sides leading to the security gate.

Other Challenges

- In recent years, the practice of defining good requirements has been de-emphasized due to agile and other light processes.
  - This has resulted in people needing to “re-learn” how to gather and write good user requirements.
  - User stories are generally inadequate to express the details that would normally be found in requirements.
    - The same holds true for use cases.

A Balancing Act


Functional Security Requirements
Security needs
What is needed by users
User (Functional) Requirements
Security Requirements
Other requirements, such as usability and performance
Area of tradeoffs

ASTQB
American Software Testing Qualifications Board, Inc.

What This Means

- Not all security requirements are functional
  - But some must be
- Security requirements must also be balanced with usability, performance and other attributes.
- The scope and boundary of all requirements can be somewhat fuzzy and changing.
- The customer often lacks the knowledge of what is truly needed for software or other IT security.

A Few Things to Understand About Requirements

- They are almost always *incomplete*.
  - Like tests, it is impossible to consider every need or problem.
- Most of the time and effort is (or should be) spent on *understanding* needs and problems.
- They often have various *origins*:
  - User, technical, legal, compliance, business, etc.
- The should describe *needs*, not solutions.
  - Solution is the implementation effort.

ASTQB American Software Testing Qualifications Board, Inc.

A Few Things to Understand About Requirements (2)

- They are *interim* work products, not the product itself.
  - "Means to an end"
- They should be *tested*.
  - With clear traceability to other related work items.
- There is a rich *body of knowledge* that few people read or follow.
- They will *change*!

A Few Guiding Principles For Security Requirements

- Security must be built in, not tested in or patched in.
  - This requires early and continual definition
- All classes of stakeholders must be involved.
  - Business, technical, security, testers, BAs
  - However technical guidance is needed
- Assets must be clearly identified and analyzed to determine sensitivity, risk and value.

Guiding Principles (Cont'd.)

- Past security issues must be incorporated.
  - Both internal and external learning
- New security issues must be incorporated
- Security requirements should be retained and shared between projects, where applicable.

The Nature of Software Security Requirements

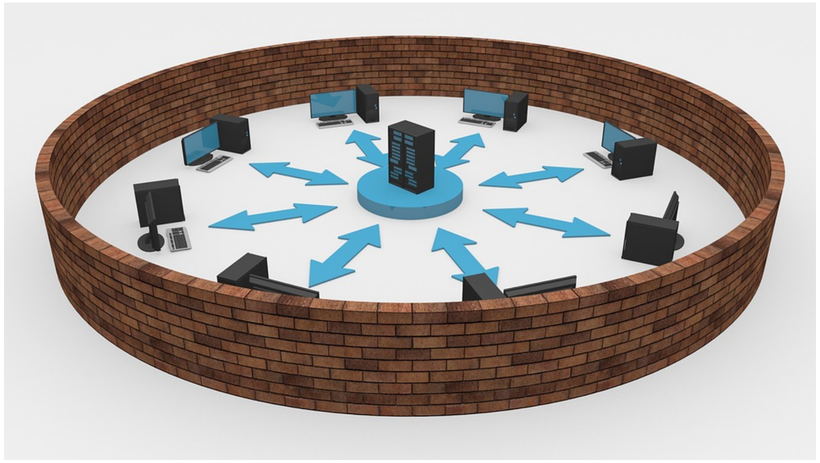
- Software security requirements can be seen in at least two contexts:
  - What the software must **include** to implement the needed levels of security
    - The scope of this can be somewhat managed
  - What the software must **exclude** to keep attackers from being successful
    - The scope of this can be almost impossible to manage completely due to the very high number of threats.

What is a Problem Frame?

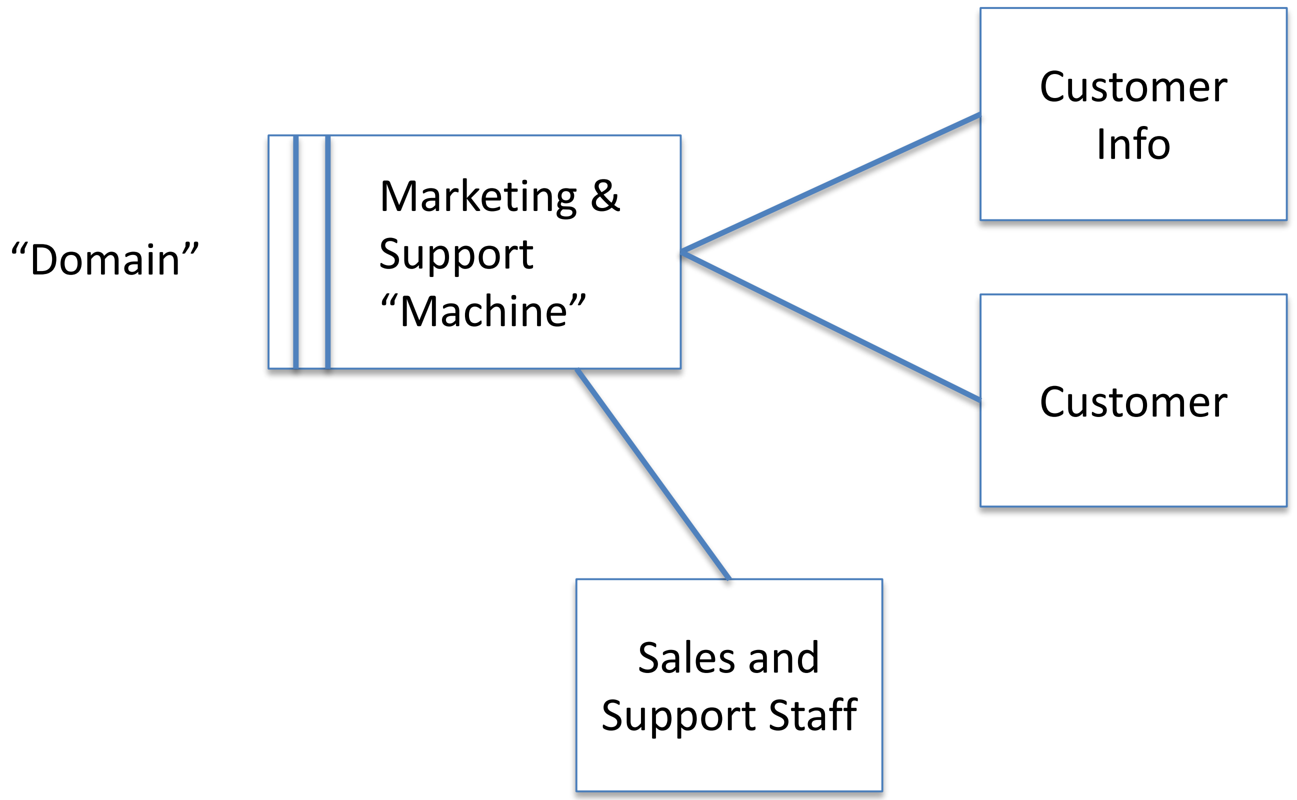
- It is a way to see typical patterns of software tasks.
- It "defines the shape of a problem by capturing the characteristics and interconnections of the parts of the world it is connected with, and the concerns and difficulties that are likely to arise.
- So problem frames help you to focus on the problem, instead of drifting into inventing a solution."
  - Michael Jackson

Example Problem Frame for Security

- An example problem frame for security is:
  - "Preserving the integrity and privacy of stored data"



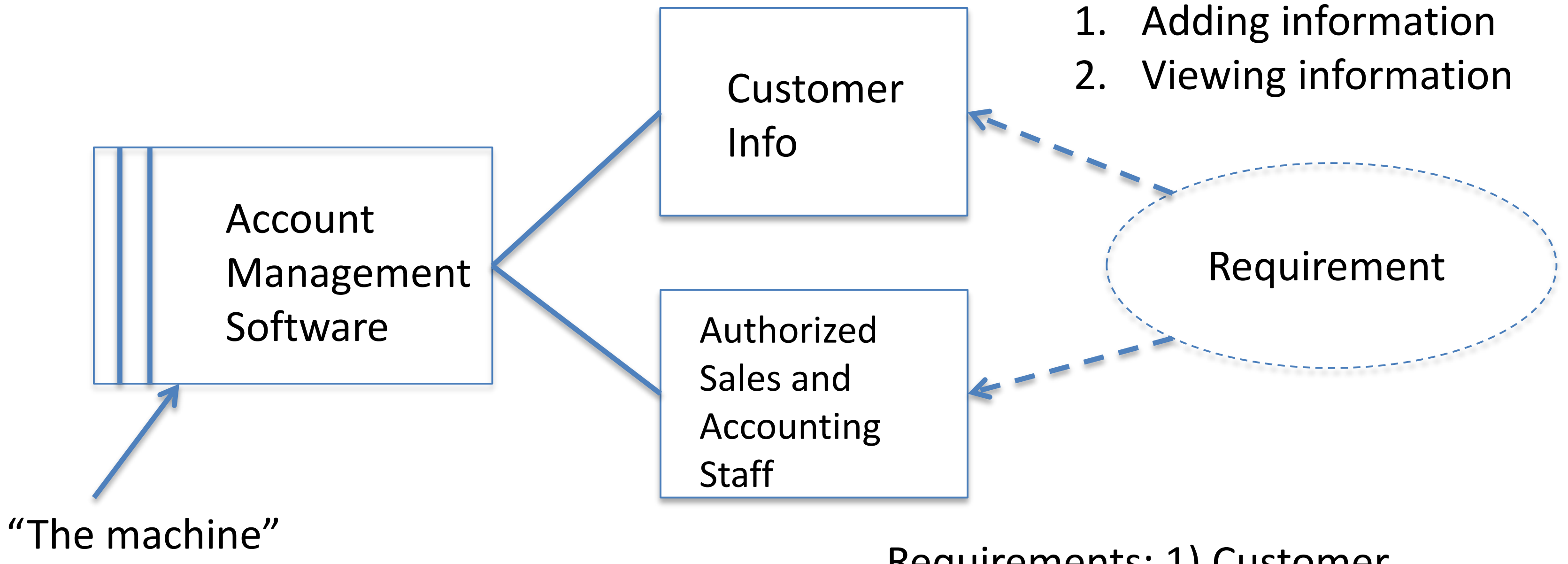
Context Model Example (General)



Frame Modeling Example (Security Context)

2 sub-problems:

1. Adding information
2. Viewing information



Account Management Software
Customer Info
Authorized Sales and Accounting Staff
Requirement
"The machine"

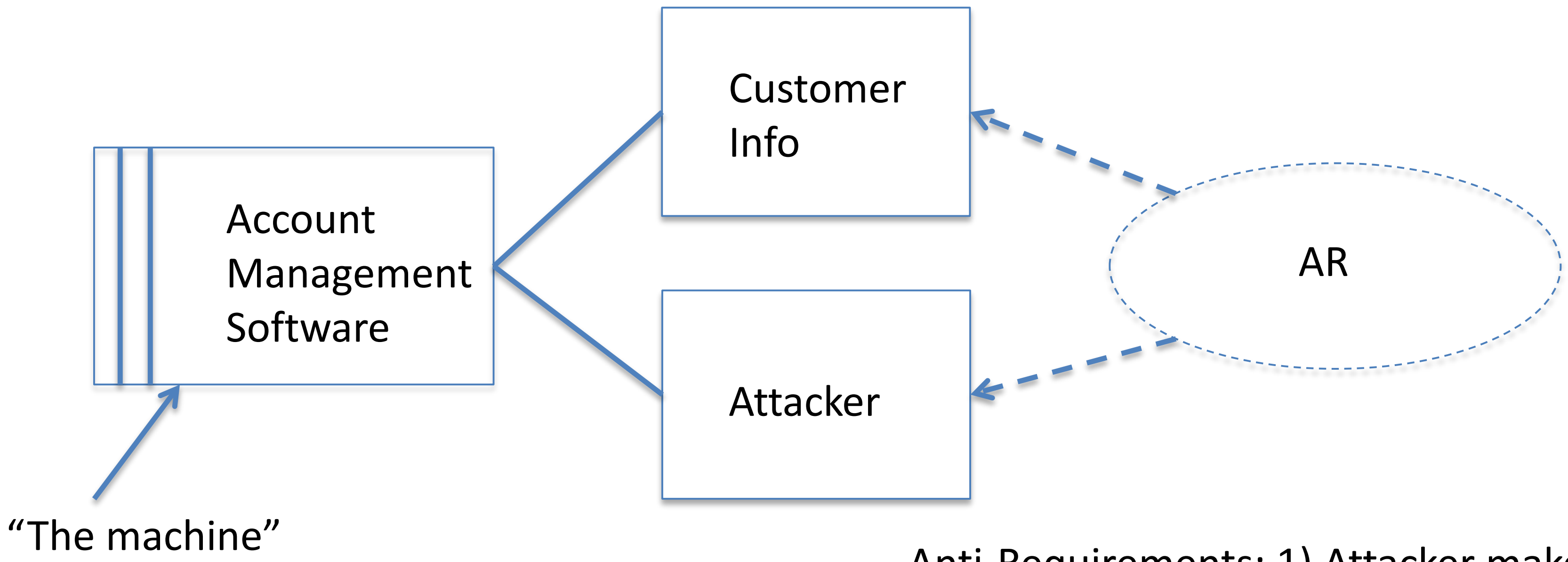
Customer info is the asset because it has value and has the need for confidentiality, integrity, availability, accountability and authenticity.

Requirements: 1) Customer Information must be editable and viewable only by authorized sales staff and authorized accounting staff. 2) Customer information must not be editable or viewable by non-authorized entities.

Security Catalog

- A security catalog contains security models for *threats* and the corresponding *security requirements*.
  - These models are modeled by security frames (what is needed) and abuse frames (threats)
- This catalog should be *generic*.
  - Not limited to any specific domain context
  - So, it can be customized and reused according to context of the software system to be modeled.

Abuse Frame Example



Customer info is the asset because it has value and has the need for confidentiality, integrity, availability, accountability and authenticity.

Anti-Requirements: 1) Attacker makes Modifications to customer information Without authorization. 2) Attacker Views customer information without Authorization. 3) Attacker obtains Copies of customer information without Authorization.

Eliciting Security Requirements

- Now, we need to express security requirements in tangible text and models.

Questions To Help Refine and Extend the Requirements

- Who are considered authorized users within the sales staff and accounting staff?
- How are those users granted authorization?
- Are there varying levels of authorization?
- Do the levels of authorization expire or require renewal?
- Which, if any, information may be viewed by customers (such as in a customer portal)?
- Are there any internal controls that must be considered and/or applied?
- Is it possible for this and other security requirements to be exploited for an attack?

Ways to Understand and Elicit Security Requirements

- Table-top exercises
  - 1 to 5 days in length
  - Red team vs. blue team
  - Identifies frames and abuse frames
- CWE List
- Interviews
- Brainstorming
- Requirements catalog
- Threat history
- Risk assessments
- OWASP Secure Coding Practices
  - https://www.owasp.org/images/0/08/OWASP_SCP_Quick_Reference_Guide_v2.pdf

Security Requirements Evaluation

- Now, we need to review and evaluate the security requirements for any ambiguities, conflicts or gaps.
  - Question assumptions
  - Verify constraints
- Reviews and Table-Top Exercises are good for this.
  - https://www.mitre.org/sites/default/files/publications/pr_14-3929-cyber-exercise-playbook.pdf

Security Requirements Testing

- This can take the form of:
  - Verification – Such as requirements-based testing and model-based testing
  - Validation – Real-world attack scenarios (white hat ethical hacking), Security testing (internal) not based on requirements, but traceable to requirements.
    - Hackers don’t care about your requirements, policies or procedures!

There Is More…

- In this session, we have only scratched the surface of problem frames.
- Hopefully, this process has provided an example of using problem frames and elicitation techniques to define and refine security requirements.
- There are many other attack vectors and threats, with more to come, so the target is constantly changing.

Helpful Resources

- Exploring Requirements by Weinberg and Gause
  - https://amzn.to/2kkJ87Q
- Managing Software Requirements by Leffingwell and Widrig
  - https://amzn.to/2s7Hwm7
- Problem Frames and Methods: Analysing and Structuring Software Development Problems by Michael Jackson
  - https://amzn.to/2GKnOkU
- Introducing Problem Frames
  - http://www.wirfs-brock.com/PDFs/Intro_Problem_Frames.pdf

Helpful Resources

- Table-Top Exercise Handbook
  - https://www.mitre.org/sites/default/files/publications/pr_14-3929-cyber-exercise-playbook.pdfOWASP Top 25 Secure Coding Practices
- Abuse Cases
  - http://www.jot.fm/issues/issue_2003_05/column6/
- NIST Risk Assessment Model
  - https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf
- ISTQB Advanced Security Syllabus
  - www.astqb.org

thank you

ASTQB
American Software Testing Qualifications Board, Inc.

Bio - Randall W. Rice

- 40 years experience in building and testing information systems in a variety of industries and technical environments
- ASTQB Certified Tester – Foundation level (CTFL), Advanced Level (CTAL – Full), CTAL-SEC, CTAL-TAE
- Director, American Software Testing Qualification Board (ASTQB)
- Chairperson, 1995 - 2000 QAI's annual software testing conference
- Co-author with William E. Perry, *Surviving the Top Ten Challenges of Software Testing*
- Principal Consultant and Trainer, Rice Consulting Services, Inc.

Contact Information

Randall W. Rice, CTAL

Oklahoma City, OK 73170

Ph: 405-691-8075

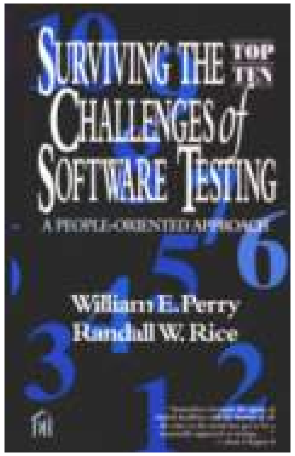
Web sites:

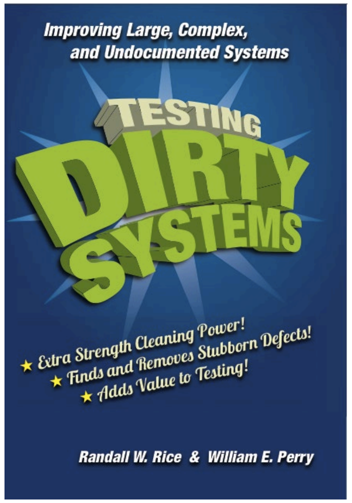
https://www.riceconsulting.com

https://mysoftwaretesting.com

Blog: randallrice.blogspot.com

e-mail: rrice@astqb.org


Surviving the Top Ten Challenges of Software Testing
A People-Oriented Approach
William E. Perry
Randall W. Rice


Improving Large, Complex, and Undocumented Systems
Testing Dirty Systems
★ Extra Strength Cleaning Power!
★ Finds and Removes Stubborn Defects!
★ Adds Value to Testing!
Randall W. Rice & William E. Perry