Using Malware Analysis to Identify Overlooked Security Requirements (MORE)

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Nancy R. Mead

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM-0004329

Topics

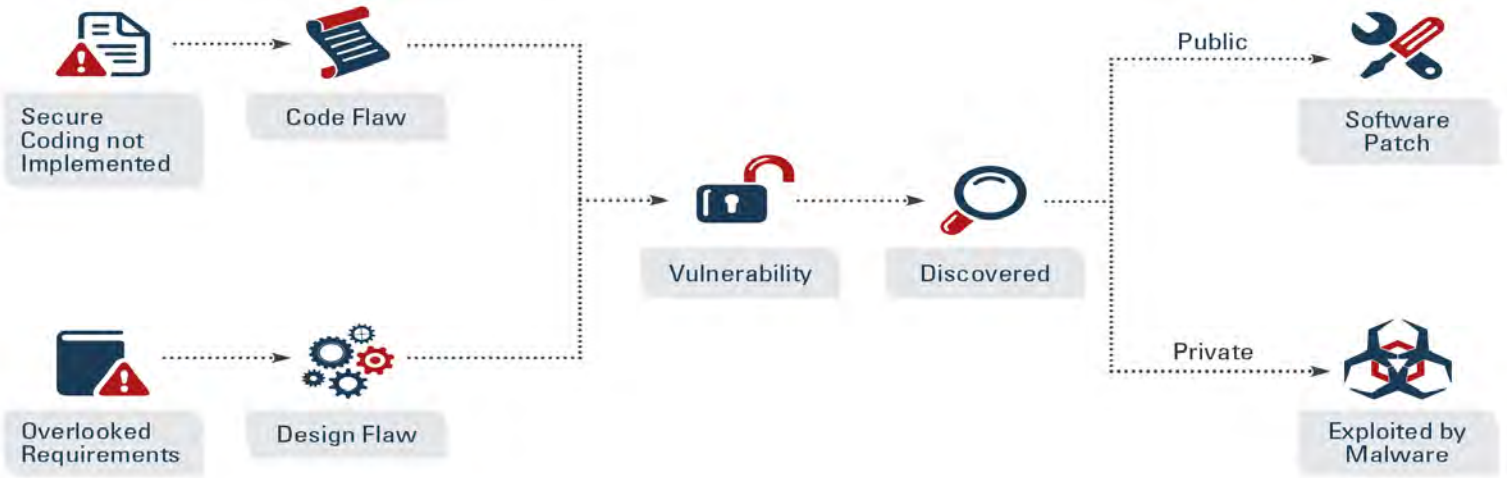
- Problem Statement
- Malware-Analysis-Driven Use Cases
- Process for Creating Use Cases
- Student Contributions to Research
  - Case Study
  - Tool Development
- Reflections
- Current Status and Future Work
- Questions

Problem Statement

Because malware analysis is not used to inform early lifecycle activities, such as security requirements engineering, our case studies show that new products are vulnerable to existing known malware.

- Operational techniques like malware analysis are typically used for patch generation; they don't usually get fed back into the development process.
- Security requirements developers tend to either start with a blank slate or with large databases of candidate requirements and use cases.
- Creating and prioritizing security requirements may be done without the insights gained from analysis of prior attacks, especially those that are specific to a particular domain.

Creating a Vulnerability



Code flaws result from a lack of secure coding.

Design flaws result from overlooked requirements.

An unknown amount of time is needed to discover a vulnerability:

- discovered in software
- discovered as part of a malware exploit

If discovered

- and made public → patch it!
- and kept private → exploit it!

Malware-Analysis-Driven Use Cases

Malware is already analyzed by domain expert.

- We start the process with the analysis results.

It's exploiting a vulnerability!

- Get the exploit details.

Design or code flaw?

If design, what requirements were overlooked that led to the flaw?

Create a use case from those requirements and add it to the database.

- Goal: Requirements should prevent this flaw from occurring again.

Process for Creating Use Cases

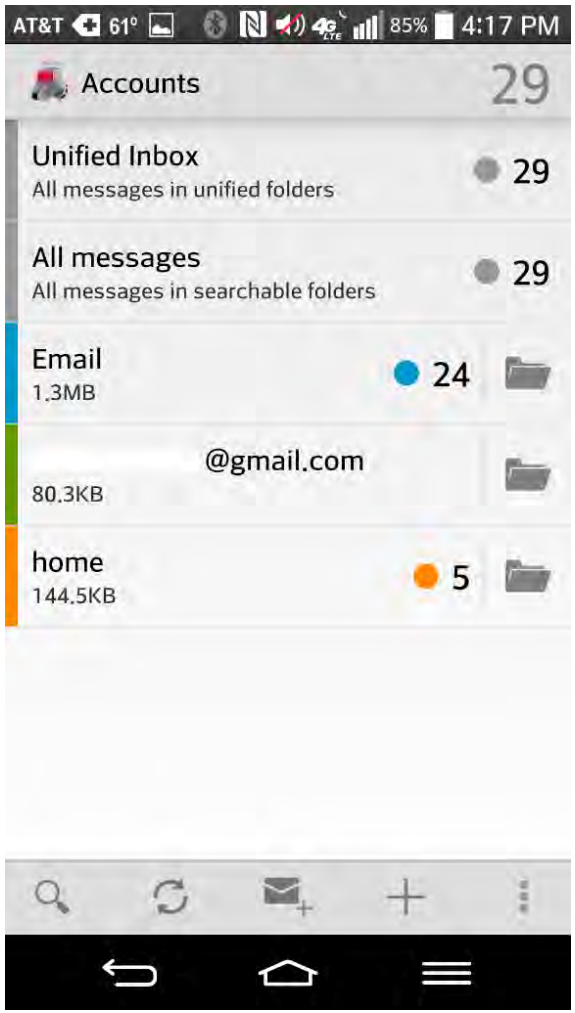
1. The results are obtained from completed static and dynamic analysis of a malicious code sample.
2. Analyses reveal the malware is exploiting a vulnerability from either a code flaw or a design flaw.
3. In the case of a design flaw, the exploitation scenario corresponds to a misuse case that should be described.
4. The misuse case is analyzed to determine the overlooked security requirement and its corresponding use case.
5. The security requirements statement and corresponding use case are added to a requirements database.
6. The requirements database is used in future software development projects. (Traceability is retained across the steps and the use of requirements from the database is tracked.)

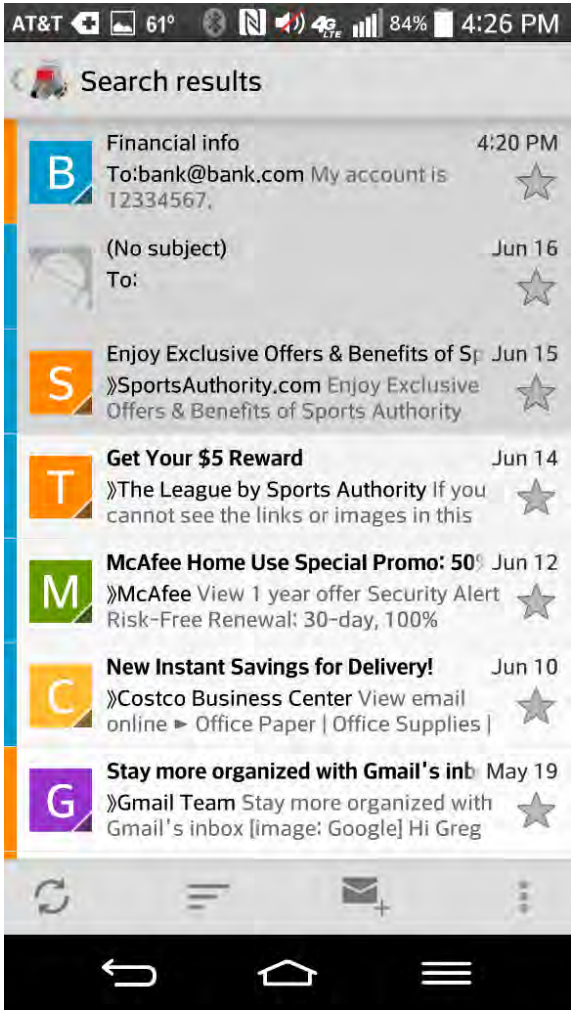
Case Study (with Greg Alice)

Application

K-9 Mail Application for Android

- open source
- compatible with IMAP, POP3, and Exchange 2003/2007
- provides searching and other common smartphone email client functionality
- user expectation of privacy and security





Vulnerability

DroidCleaner

- Trojan malware
  - claims to perform an Android tune-up
  - sends premium-rate SMS messages
  - uploads data from the Android External Storage area to hacker's servers

Exploitation Scenario

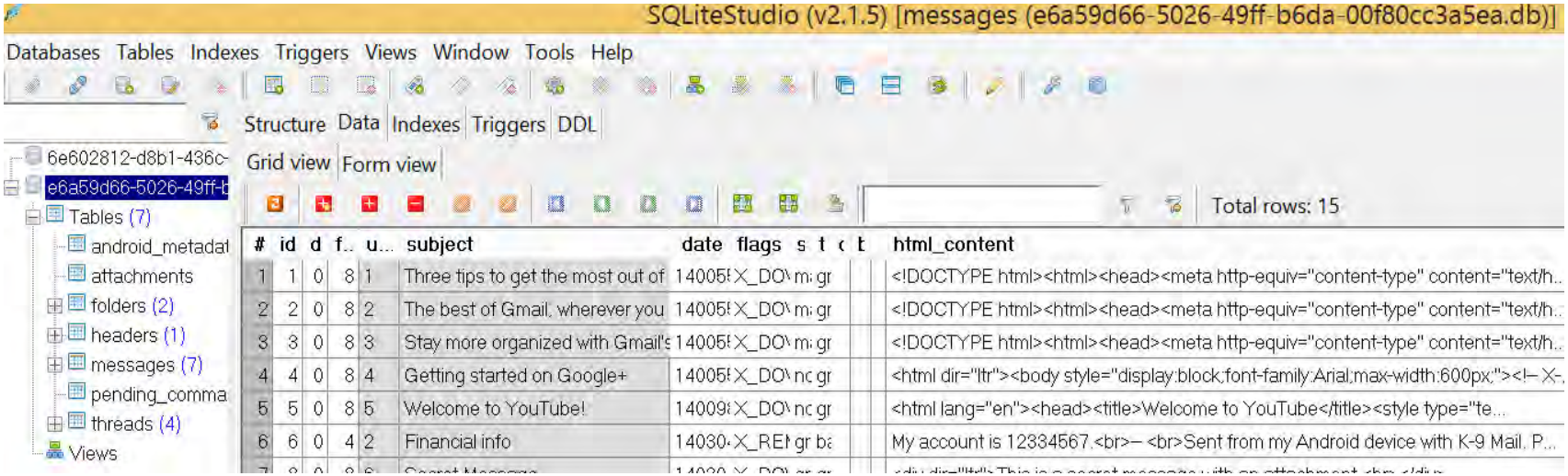
Trojan

- Social engineering to trick user into installing DroidCleaner:
  - Install software.
  - Grant access to external storage, internet access.

K-9 Mail configured to store email in External Storage.

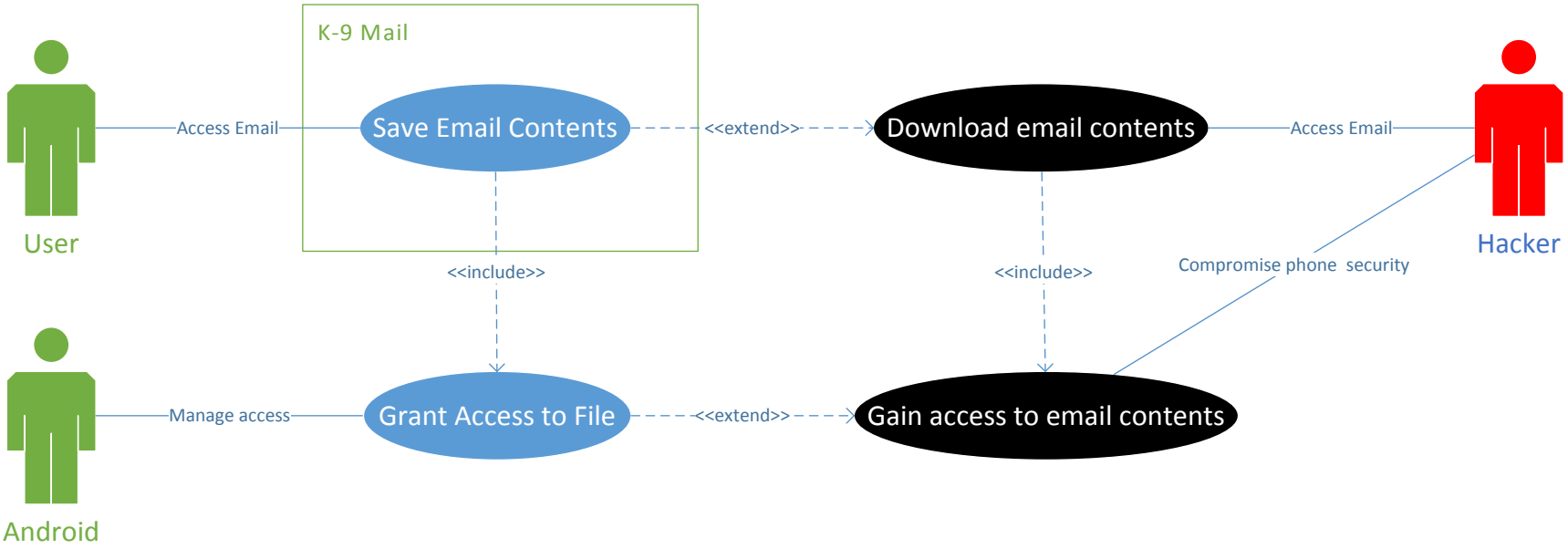
DroidCleaner uploads External Storage to hacker server.

The hacker examines contents; email contents are disclosed.



Misuse Case

Gain Access to Email Contents



New Requirement

Requirement Number: 1	
Requirement	1.1 Email contents shall be protected from unauthorized access. Email contents shall be stored in an area only available to the application (Android Internal Storage default configuration) and/or protected through encryption, which cannot be decrypted using data available in Android External Storage. 1.2 Processes with access to External Storage shall not have the ability to view K-9 Mail contents in clear text. If external storage is selected, a warning message or mitigation, such as encryption, is recommended.
Category	Data Protection
Priority	High
Cost	Medium
Misuse Case	MUC2
Rationale	Due to the high risk of data theft malware on Android, it is not safe to assume data kept on the phone is private; therefore, the email contents must be kept in a form that cannot be read, even if the hacker has access to the storage location.

Tool Development by Students

Tool Development: Student Teams

First team was a group of five CMU Master of Software Engineering (MSE) students, working on the project over the course of four semesters.

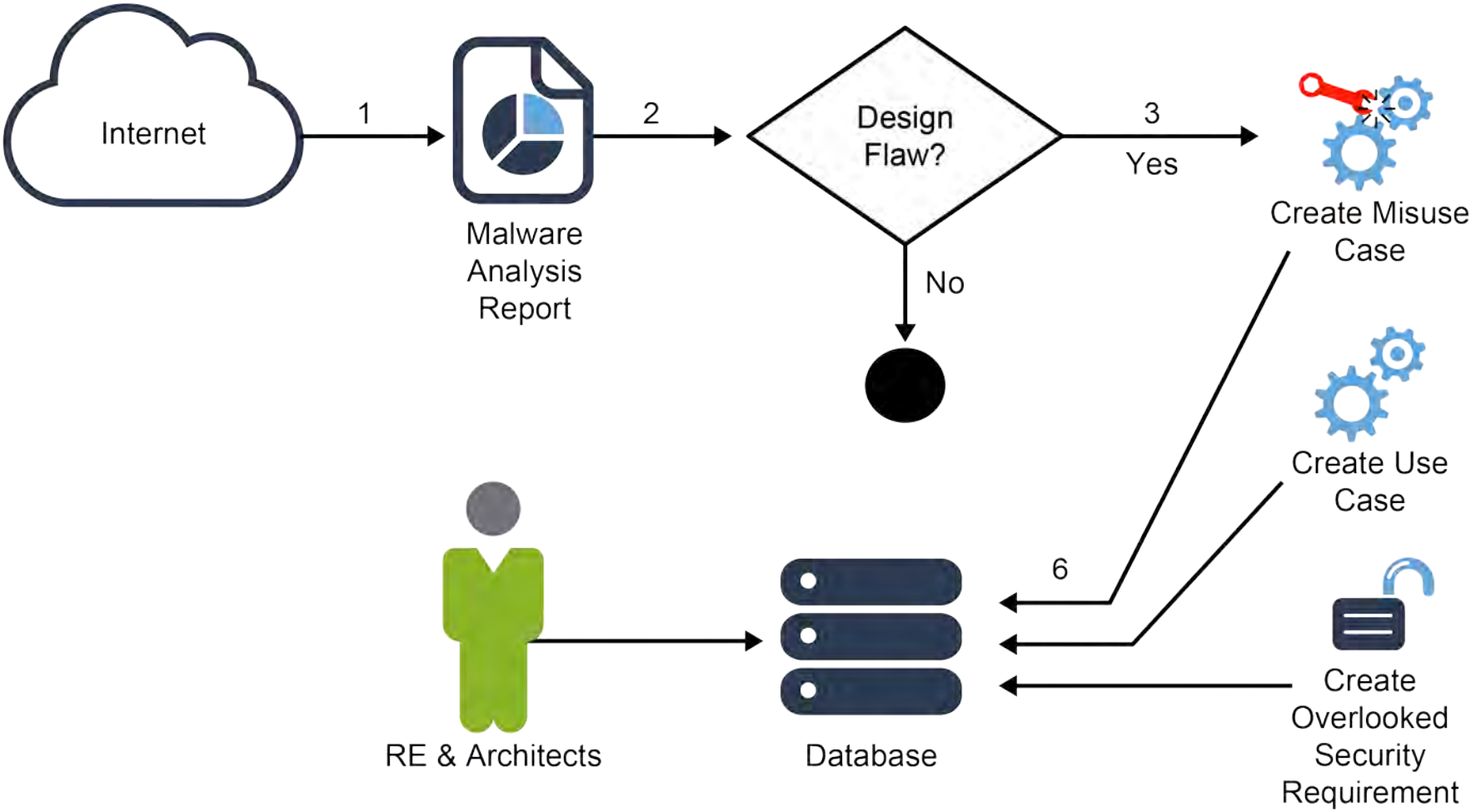
Second team is a group of three CMU Master of Information Technology (MITS) students, working on the project over the course of two semesters.

Tool Development: Initial MSE Team Goal

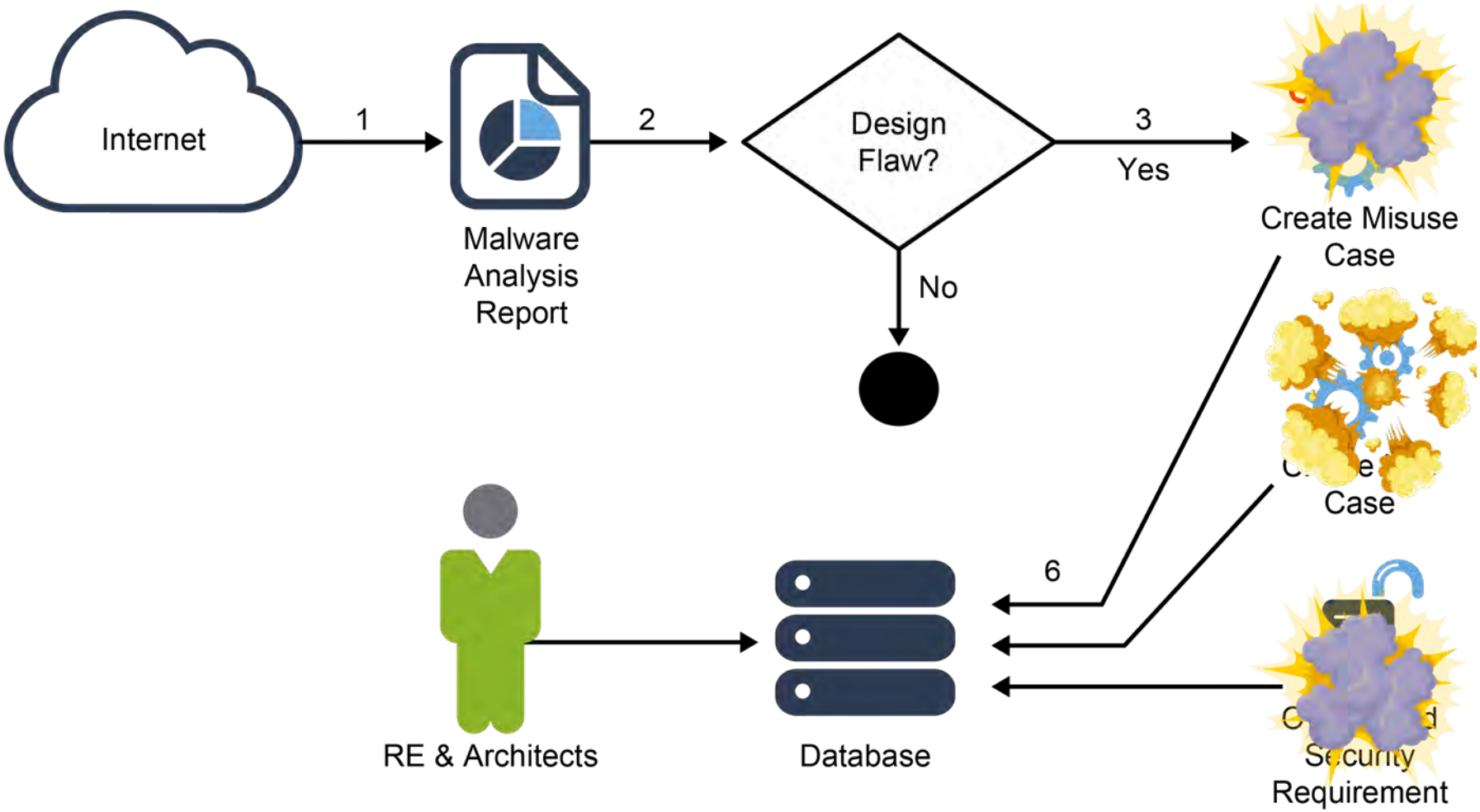
To provide a proof of concept by implementing and automating the solution

To develop a web application that reads malware analysis reports and creates a database of misuse cases, use cases, and overlooked security requirements (MUOs)

Initial Idea



Initial Idea



© Team Sentinels

Tool Development: Outcome

A tool to help report writers write more comprehensive reports and include the misuse cases, use cases, and overlooked security requirements from the start

Tool Development: MORE Tool

Two web-based applications

- Report Writer
- Security Requirement Finder (SERF)

User roles

- Public user
- Report writer
- Reviewer
- Administrator
- Super user

Reflection on the Project

Research contributions

- Initial research was done by me and Jose Morales (senior researchers), presentations helped to validate/refine ideas.
- Case study was developed by Greg Alice – Master's student with years of industry experience.

Tool/automation student contributions

- Master's students were able to contribute to tools development.
- Coaching is needed to understand methodology.
- For the most part, students are not prepared to do research.

Practical application of the method

- Gap between research and practice – the challenge is to find a forward-thinking organization that sees value and will fund pilot efforts.

Reflections from a Faculty Perspective

Student preparation

- For the most part, students with undergraduate degrees and a year or two of work experience are not prepared to do research in cybersecurity, or to fully specify a software product.
- Faculty effort is needed to coach them and set expectations; plan to spend a lot of time up front to get them up to speed.

Faculty Expectations

- Student artifacts are seldom complete products, more likely to be prototypes.
- Master's students with significant work experience or PhD students can contribute to research, as you might expect.

Current Status

Webpage cert.org/cybersecurity-engineering/research/security-requirements-elicitation.cfm

Release of MSE prototype tool source code on GitHub

Several industry case studies by U.K. students - Reports of prior successful attacks caused clients to assign higher priority to requirements for mitigating those attacks.

Paper and tool demo presented at Requirements Engineering Conference ESPRE Workshops. Additional presentations at CMU faculty seminars and SEI Software Solutions Symposium

MITS student team expanded tool database, currently working on bug fixes and enhancements

Future Work

Research activities

- Identify ways to use this method in threat modeling, in conjunction with the SEI's threat modeling project.
- Assess usefulness in other lifecycle activities (e.g., architecture and design).

Practical application of the method

- Apply this method to larger systems to increase the knowledge base.
- Work with organizations developing new systems or enhancing existing systems.

Tool/automation activities

- Revisit automated processing of malware reports.
- Revisit automated processing of CWEs in conjunction with Mitre reorganization of CWEs.

Other Related Work

Research

- threat modeling research (flyer available)

Education

- conference on Software Engineering Education & Training (Call for Submissions available)

Cyber Security Engineering Book

- planned book giveaway and deep discount order forms

Questions?

Contact Information

Nancy R. Mead

Fellow and Principal Researcher
CERT Division
Email: nrm@sei.cmu.edu

U.S. Mail

Software Engineering Institute
Customer Relations
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Web

www.sei.cmu.edu
www.sei.cmu.edu/contact.cfm

Customer Relations

Email: info@sei.cmu.edu
Telephone: +1 412-268-5800
SEI Phone: +1 412-268-5800
SEI Fax: +1 412-268-6257