Software Solutions Symposium 2017

March 20–23, 2017

Using Malware Analysis to Identify Overlooked Security Requirements (MORE)

Nancy R. Mead

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Using Malware Analysis to Identify Overlooked Security Requirements (MORE)
© 2017 Carnegie Mellon University

Software Engineering Institute | **Carnegie Mellon University**

Using Malware Analysis to Identify Overlooked Security Requirements (MORE)
March 2017
© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

2

Topics

Problem Statement

Malware-Analysis-Driven Use Cases

Process for Creating Use Cases

Case Studies

Tool Development

Current Status

Discussion and Future Work

Software Engineering Institute | **Carnegie Mellon University**

**Using Malware Analysis to Identify Overlooked
Security Requirements (MORE)**
March 2017
© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for
public release and unlimited distribution.

3

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
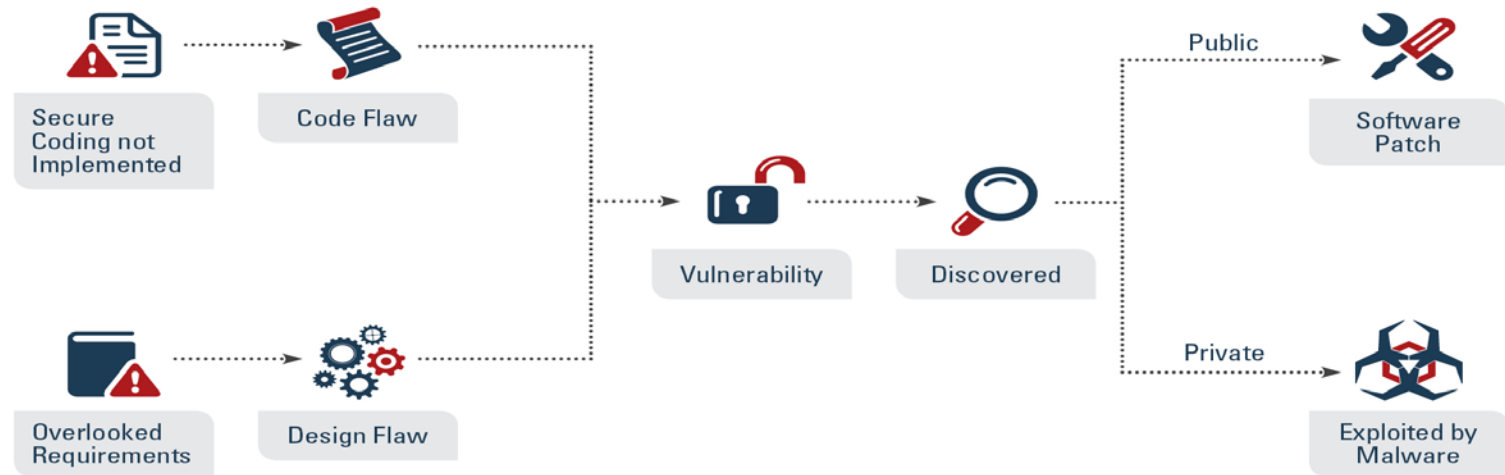
Code flaws result from a lack of secure coding.

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!

Software Engineering Institute | **Carnegie Mellon University**

Using Malware Analysis to Identify Overlooked Security Requirements (MORE)
March 2017
© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

5

Malware-Analysis-Driven Use Cases

Analyze Malware Sample → Exploiting Vulnerability → Determine Design Flaw → Determine Overlooked Requirements → Create Use Case → Add to Database

Malware 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.)

Software Engineering Institute | Carnegie Mellon University

Using Malware Analysis to Identify Overlooked
Security Requirements (MORE)
March 2017
© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for
public release and unlimited distribution.

7

Software Solutions Symposium 2017

Case Studies

Using Malware Analysis to Identify Overlooked Security Requirements (MORE)
© 2017 Carnegie Mellon University
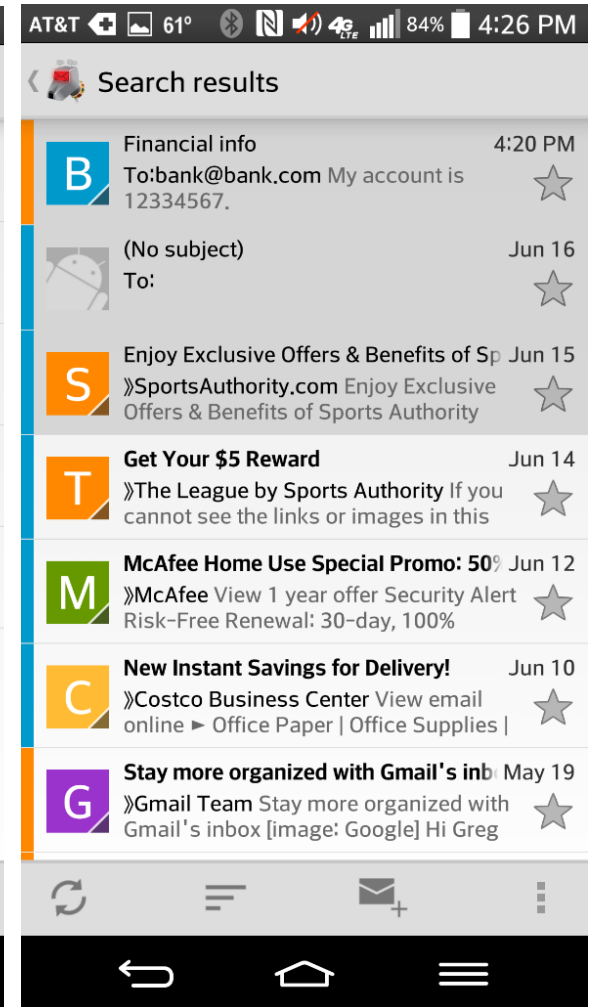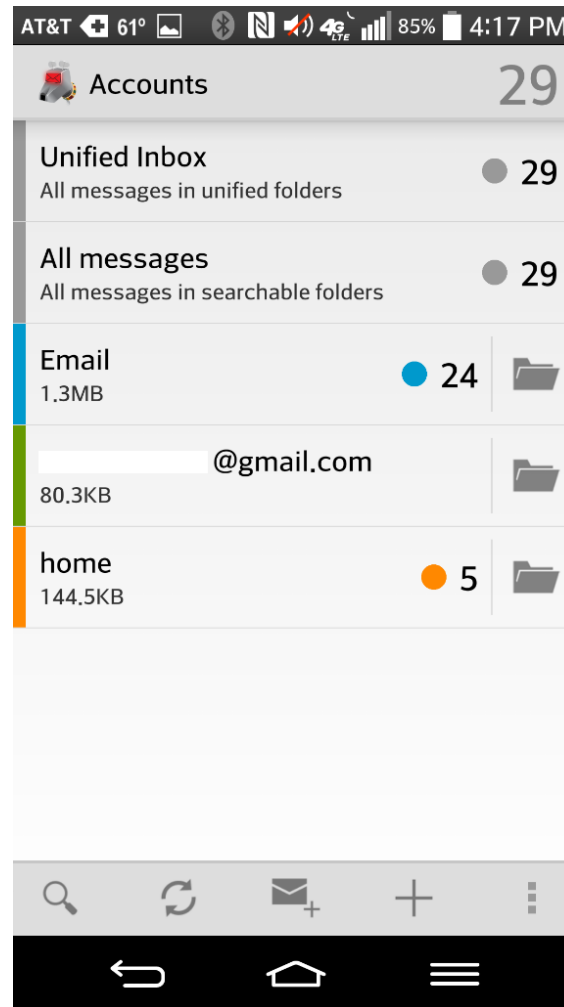
Application

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

1. Application installation
2. Malware download
   Malware
3. Malware infection
4. Information stealing

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.

Using Malware Analysis to Identify Overlooked Security Requirements (MORE)
March 2017
© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

10

Software Engineering Institute | Carnegie Mellon University
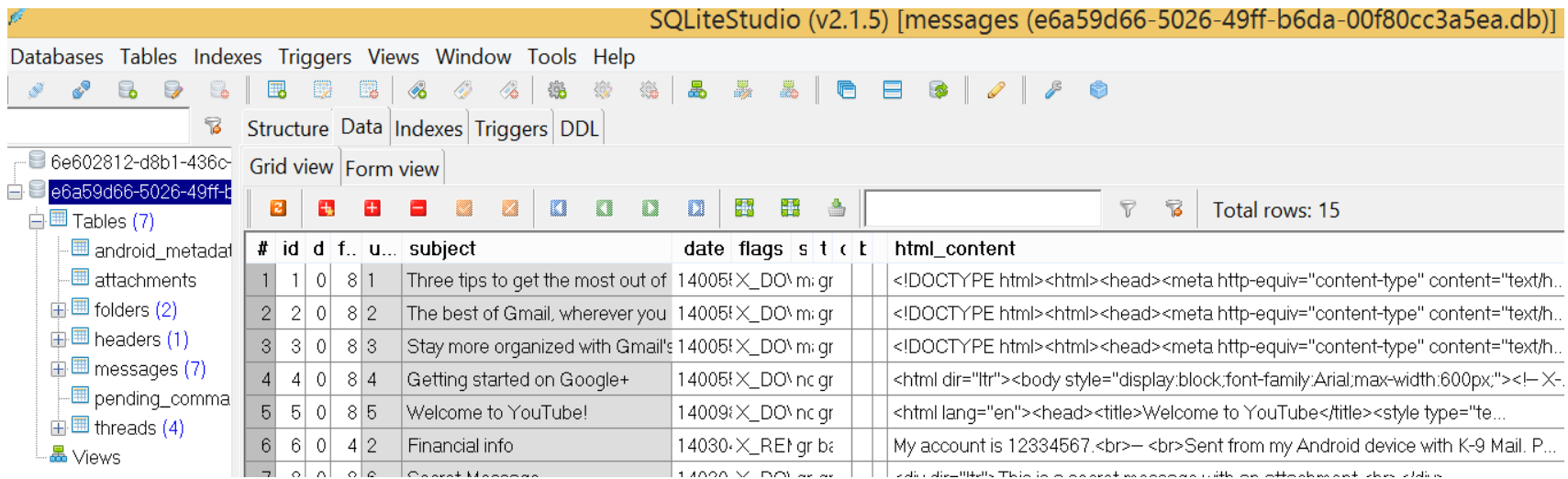
Exploitation Scenario

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.
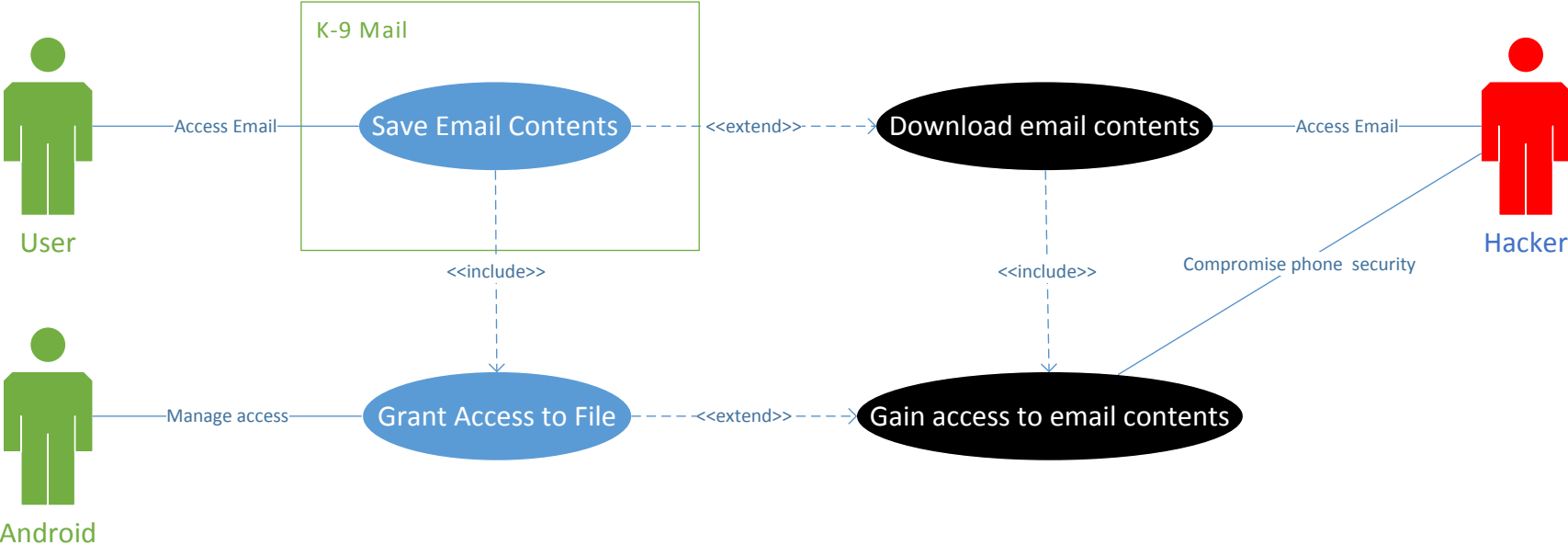
Using Malware Analysis to Identify Overlooked Security Requirements (MORE)
March 2017
© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

11

Misuse Case

Gain Access to Email Contents

Software Engineering Institute | **Carnegie Mellon University**

Using Malware Analysis to Identify Overlooked Security Requirements (MORE)
March 2017
© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

12

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.

Software Engineering Institute | Carnegie Mellon University

Using Malware Analysis to Identify Overlooked Security Requirements (MORE)
March 2017
© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

13

Software Solutions
Symposium 2017

Tool Development

Tool Development: Initial 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)

Using Malware Analysis to Identify Overlooked Security Requirements (MORE)
March 2017
© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

15

Software Engineering Institute | Carnegie Mellon University

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

Software Engineering Institute | **Carnegie Mellon University**

Using Malware Analysis to Identify Overlooked Security Requirements (MORE)
March 2017
© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

16

Tool Development: MORE Tool

Two web-based applications
- Report Writer
- Security Requirement Finder (SERF)

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

Using Malware Analysis to Identify Overlooked Security Requirements (MORE) March 2017 © 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

17

Current Status

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

Release of prototype tool source code on GitHub

Several industry case studies by U.K. students

Paper and tool demo presented at Requirements Engineering Conference ESPRE Workshops and CMU faculty seminars

Student team currently working on expanding tool database

Software Engineering Institute | **Carnegie Mellon University**

Using Malware Analysis to Identify Overlooked Security Requirements (MORE)
March 2017
© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

18

Discussion and Possible Future Work

Research activities
- Identify ways to use this method in threat modeling, in conjunction with the SEI 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.

Software Engineering Institute | **Carnegie Mellon University**

Using Malware Analysis to Identify Overlooked Security Requirements (MORE)
March 2017
© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

19

Contact Information

Nancy R. Mead

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

Web

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

U.S. Mail

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

Customer Relations

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

Software Engineering Institute | **Carnegie Mellon University**

Using Malware Analysis to Identify Overlooked Security Requirements (MORE)
March 2017
© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

20