Teaching Android Security: A Public Educational Activity of Vulnerable Android Applications

Daniel E. Krutz
Software Engineering Department
Rochester Institute of Technology
1 Lomb Memorial Drive
Rochester, NY, USA
{dxkvse}@rit.edu

Abstract

This tutorial presents a public educational activity to assist in the instruction of both students and developers in creating secure Android apps. These activities include example vulnerable applications, information about each vulnerability, steps how to repair the vulnerabilities, and information about how to confirm that the vulnerability has been properly repaired. Our goal is for instructors to use these activities in their mobile, security, and general computing courses ranging from the K-12 to university settings. A secondary goal of this project is to foster interest in security and computing through demonstrating its importance. All project activities may be found on the project website: **www.TeachingMobileSecurity.com** and more information about the activity may be found in an upcoming ACM InRoads article.

1 Project

1.1 Motivation

Developers frequently create vulnerable software for a wide range of reasons: ignorance of how to create secure apps, simple errors, or a lack of understanding of the importance of secure app development. We have created a public sample set of vulnerable Android apps in order to help educate developers to create secure apps and demonstrate the importance of secure app development.

There have been many recent publications and funded projects which address the deficiencies of security related activities which may be used in an educational environment [1, 3–5, 13]. Our project is distinct in that it exclusively focuses on mobile security.

1.2 Project Outline

Although the number of activities is growing, we currently have ten vulnerability exercises ranging from proper Intent protection to more complicated activities such as correct use of content providers. Each example contains a clear demonstration of the negative ramifications of the vulnerability, steps to repair the vulnerability, and posted actions to ensure that it has been resolved. The process outline of each activity is shown in Figure 1.

Figure 1: App Repair Process

Each of the exercises contains:

1. Mobile apps which contain well defined vulnerabilities.
2. Documentation about the adverse effects of the vulnerabilities and how they may be exploited.
3. Step by step documentation how to repair the vulnerabilities.
4. Instructions how to verify that the vulnerability has been repaired.
5. Examples of the apps which have already had the vulnerabilities repaired.

Activities begin with providing the user some background (when, why, and how the vulnerability may occur) about the specific vulnerability being targeted. Whenever possible, users are also provided with a real-world example of occurrences of the vulnerability such as where they occurred in specific apps. Also included are some basic reasons about why the vulnerability occurs and common developer mistakes which lead to the vulnerability.

1.3 Project Objectives

Creating accurate, robust activities can be a difficult and time consuming task for instructors. In order to alleviate some of these challenges, our goal is for instructors to use some of these activities in their mobile, security, and general computing courses.

1.4 Schedule

The following schedule will be used in our tutorial. All times are approximate, and our activity set has over ten possible exercises which may be done in this tutorial. However, we will select a few which will be the most beneficial for the attendees and may extend or shorten specific activities as time allows.

Table 1: Schedule

Length	**Activity**
30 min	Project Introduction
30 min	Machine setup
40 min	Exercise #1
40 min	Exercise #2
40 min	Exercise #3

2 Tutorial Information

1. **Duration:** Half-day (although it could be full day if needed). The activity-set is comprised of many exercises, so we would not expect to get through all of them in a single day. For this tutorial, we would focus on educating others about the activity-set, along with how to most properly use it in their classrooms.

2. **Expected background of the audience:** Audience will not need to have any background in mobile development, although some programming experience would be beneficial. Expected participants include computing educators; especially those who teach either mobile or security related courses. This tutorial would also serve as a beneficial exposure to the activity set to developers as well.

3. **Required Equipment:** The tutorial will require a projector be made available to the instructor, and each participant should bring their own Windows, or Mac laptop. A virtual machine pre-loaded with the appropriate software will be provided to each participant.

4. **Teaching Materials:** All existing teaching materials may be found on the project website: **www.TeachingMobileSecurity.com**.

3 Presenter Information

1. **Full Name:** Daniel E. Krutz: http://www.se.rit.edu/~dkrutz/

2. **Address:** One Lomb Memorial Drive, Rochester, NY 14623-5603 www.rit.edu, http://www.se.rit.edu

3. **E-mail:** dxkvse@rit.edu

4. **Institution:** Software Engineering Department, Rochester Institute of Technology

5. **Education:** PhD Computer Science - Nova Southeastern University 2013

6. **Related Works:**

   Projects related to Android apps and security:

   (a) **Darwin Project**[1]**:** Analyzes downloaded apps for a vareity of security and quality related metrics. To date, the project has analyzed over 70,000 Android apps. This project resulted in a team of my students winning 1st place in an IEEE student research competition [2].

   (b) **Androsec Project**[2]**:** Collects and analyzes Android version control repositories from F-Droid, an open source Android app repository. This project has already resulted in an MSR publication [11] and current research is being conducted using this data set.

   (c) **M-Perm**[3]**:** A tool for detection the permission gap in Android 6.0 and above apps.

   Daniel also has several recent pedagogically focused publications ranging from how to best instruct Deaf/Hard of Hearing students in computing to innovative activities in Software Security courses [6–10, 12, 14, 15]

[1] http://darwin.rit.edu
[2] http://androsec.rit.edu
[3] http://www.m-perm.com

7. **Background:** Daniel is a lecturer at the Rochester Institute of Technology in the Software Engineering department. He received his PhD in Computer Science in 2013 from Nova Southeastern University. Some of the courses he has taught include Introduction to Software Engineering, Engineering of Secure Software, Web Engineering, Foundations of Software Engineering (Graduate), and Research Methods (Graduate). Daniel's research interests include Mobile security and Software Engineering Education.

Acknowledgements

This work is partially sponsored by a SIGCSE Special Projects Grant.

References

[1] Y. Bai and X. Wang. Itseed: Hands-on labs for it security education (abstract only). In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE '14, pages 739–739, New York, NY, USA, 2014. ACM.

[2] F. Broderick. Undergraduate se students impress conference with research into android app vulnerabilities. https://www.rit.edu/gccis/news/team-presents-research-related-android-application-security.

[3] H. Chi and D. A. Rubio. Design insider threat hands-on labs. In *Proceedings of the 2014 Information Security Curriculum Development Conference*, InfoSec '14, pages 17:1–17:1, New York, NY, USA, 2014. ACM.

[4] W. Du. Seed: Hands-on lab exercises for computer security education. *IEEE Security & Privacy*, 9(5):70–73, 2011.

[5] M. Guo, P. Bhattacharya, M. Yang, K. Qian, and L. Yang. Learning mobile security with android security labware. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, SIGCSE '13, pages 675–680, New York, NY, USA, 2013. ACM.

[6] D. E. Krutz and M. Lutz. Bug of the day: Reinforcing the importance of testing. In *2013 IEEE Frontiers in Education Conference (FIE)*, pages 1795–1799. IEEE, 2013.

[7] D. E. Krutz, S. A. Malachowsky, S. D. Jones, and J. A. Kaplan. Enhancing the educational experience for deaf and hard of hearing students in software engineering. In *Frontiers in Education Conference (FIE), 2015. 32614 2015. IEEE*, pages 1–9. IEEE, 2015.

[8] D. E. Krutz, S. A. Malachowsky, and T. Reichlmayr. Using a real world project in a software testing course. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 49–54. ACM, 2014.

[9] D. E. Krutz and A. Meneely. Teaching web engineering using a project component. In *2013 IEEE Frontiers in Education Conference (FIE)*, pages 1366–1368. IEEE, 2013.

[10] D. E. Krutz, A. Meneely, and S. A. Malachowsky. An insider threat activity in a software security course. In *Frontiers in Education Conference (FIE), 2015. 32614 2015. IEEE*, pages 1–6. IEEE, 2015.

[11] D. E. Krutz, M. Mirakhorli, S. A. Malachowsky, A. Ruiz, J. Peterson, A. Filipski, and J. Smith. A dataset of open-source android applications. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, pages 522–525. IEEE, 2015.

[12] D. E. Krutz and J. R. Vallino. Experiencing disruptive behavior in a team using "moles". In *2013 IEEE Frontiers in Education Conference (FIE)*, pages 1492–1495. IEEE, 2013.

[13] L. Li, K. Qian, Q. Chen, R. Hasan, and G. Shao. Developing hands-on labware for emerging database security. In *Proceedings of the 17th Annual Conference on Information Technology Education*, SIGITE '16, pages 60–64, New York, NY, USA, 2016. ACM.

[14] M. J. Lutz, J. R. Vallino, K. Martinez, and D. E. Krutz. Instilling a software engineering mindset through freshman seminar. In *2012 Frontiers in Education Conference Proceedings*, pages 1–6. IEEE, 2012.

[15] S. A. Malachowsky and D. E. Krutz. A project component in a web engineering course. In *Frontiers in Education Conference (FIE), 2015. 32614 2015. IEEE*, pages 1–6. IEEE, 2015.