CSCE 713: Software Security
Mini Annotation Project

January 27, 2020

Date Due: Monday, April 27th at 8:00am

1 Overview

The mini annotation project is an opportunity for you to explore a topic in software security in greater-depth. In particular, you will read a *classic* paper related to software securtity, and annotate that paper to make it accessible to a broader audience.

More specifically, your mini annotation will give an in-depth account of the historical context of the work along with biographical accounts into the private lives of the author (or authors). Your annotation project should also help the reader understand the magnitude of the paper to the field of computer security. Finally, where appropriate, the annotation should include your personal thoughts.

2 Potential papers to annotate

You are free to consider any paper as long as it is arguably considered a classic paper in computer security[1]. Here are just a few thoughts on papers to potentially annotate.

- **Security Policy**
  - Do Bell and Lo La Padula. Secure Computer System: Unified Exposition and Multics Interpretation. ESD-TR-75-306. 1975.
  - Integrity Considerations for Secure Computer Systems. ESD-TR-76-372. 1977.
  - David Brewer and Michael Nash. The Chinese Wall Security Policy. IEEE Symposium on Security and Privacy. 1989.
  - Dorothy Denning. A Lattice Model of Secure Information Flow. Communications of the ACM, 1976.
- **Design Principles**
  - Jerome Saltzer and Michael Schroeder. The Protection of Information in Computer Systems. Communications of the ACM. 1974.
  - Jerome Saltzer, David Reed, and David Clark. End-to-end arguments in system design. ACM Transactions on Computer Systems. 1984.
- **Reliability**
  - Lloyd D. Fosdick and Leon J. Osterweil. Data Flow Analysis in Software Reliability. ACM Computing Surveys. 1976.
  - Robert E Strom and Shaula Yemini. Typestate: A Programming Language Concept for Enhancing Software Reliability. IEEE Transactions on Software Engineering. 1986.

[1] `https://www.sec.cs.tu-bs.de/~konrieck/topnotch/top100.html` is a list of the most highly cited papers in computer security.

- **Verification**
  - E.M. Clarke, E.A. Emerson, A.P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Transactions on Programming Languages and Systems.1986.
  - Dawson Engler, Benjamin Chelf, Andy Chou, and Seth Hallem. Checking System Rules Using System-Specific, Programmer-Written Compiler Extensions. USENIX Symposium on Operating Systems Design and Implementation. 2000.
  - Thomas Ball, Vladimir Levin, Sriram K. Rajamani. A decade of software model checking with SLAM. Communications of the ACM. 2011.

- **Programming**
  - Ken Thompson. Reflections on Trusting Trust. Turing Award Lecture. 1984.
  - William R. Bush, Jonathan D. Pincus and David J. Sielaff. A static analyzer for finding dynamic programming errors. Software–Practice and Experience. 2000.
  - Brian Hackett, Manuvir Das, Daniel Wang, and Zhe Yang. Modular Checking for Buffer Overflows in the Large. International Conference on Software Engineering. 2006.
  - Cowen, et al. StackGuard: Automatic Adaptive Detection and Prevention of Buffer-Overflow Attacks. USENIX Security. 1998.
  - David Wagner, Jeffrey S. Foster, Eric A. Brewer, and Alexander Aiken. A First Step Towards Automated Detection of Buffer Overrun Vulnerabilities. NDSS. 2000.

3 Completing your annotation project

(a) Select a classic paper in computer security. Your selected paper must be approved by me to make sure it fits the guidelines. Feel free to stop by my office hours or make an appointment. Discussing your paper choice should only take a few minutes. **Do this before 17 February.**

(b) Understand the historical context surrounding the paper that you have chosen. What event(s) shaped the paper's development? How did the paper change the field? **You should make use of the resources and expertise of the Texas A&M Libraries.** Library open reference hours are weekdays 10am – 10pm.

(c) Annotate the main results (especially the difficult concepts) of the paper so that they can be appreciated by a a broad range of readers interested in learning more about software security. Use the *Annotated Turing* by Charles Petzold as a guide.

(d) Submit a 10–20 page paper (in PDF) of your mini annotation by the deadline of Monday, April 27th at 8:00am. If you get on a roll and need more than 20 pages, that's fine. Just keep it below 30 pages. But, if you have less than 10 pages, then your annotation is probably insufficient.

4 Final thoughts

You should have fun and feel free to be creative with your mini annotation project. That is by far the best way to learn. Enjoy!