Security Research Methods 1

Reading papers …

- What is the purpose of reading papers?
- How do you read papers?

Understanding what you read

- Things you should be getting out of a paper
  - What is the central idea proposed/explored in the paper?
    - Abstract
    - Introduction
    - Conclusions

    These are the best areas to find an overview of the ***contribution***
  - How does this work fit into others in the area?
    - Related work - often a separate section, sometimes not, every paper should detail the relevant literature. Papers that do not do this or do a superficial job are almost sure to be bad ones.
    - An informed reader should be able to read the related work and understand the basic approaches in the area, and how they differ from the present work.

Understanding what you read (cont.)

- What scientific devices are the authors using to communicate their point?
- Methodology - this is how they evaluate their solution.
  - Theoretical papers typically validate a model using mathematical arguments (e.g., proofs)
  - Experimental papers evaluate results based on test apparatus (e.g., measurements, data mining, synthetic workload simulation, trace-based simulation).
  - Empirical research evaluates by measurement.
  - Some papers have no evaluation at all, but argue the merits of the solution in prose (e.g., design papers)

Understanding what you read (cont.)

- What did they find?
  - Results - statement of new scientific discovery.
    - Typically some abbreviated form of the results will be present in the abstract, introduction, and/or conclusions.
    - Note: just because a result was accepted into a conference or journal does necessarily not mean that it is true. Always be circumspect.
- What should you remember about this paper?
  - Take away - what general lesson or fact should you take away from the paper.
  - Note that really good papers will have take-aways that are more general than the paper topic.

The best papers are the ones that teach you something

Exercise

- Summarize the Thompson Article:
  - Contribution
  - Motivation
  - Related work
  - Methodology
  - Results
  - Take away

A Sample Summary

- Contribution: Ken Thompson shows how hard it is to trust the security of software in this paper. He describes an approach whereby he can embed a Trojan horse in a compiler that can insert malicious code on a trigger (e.g., recognizing a login program).
- Motivation: People need to recognize the security limitations of programming.
- Related Work: This approach is an example of a Trojan horse program. A Trojan horse is a program that serves a legitimate purpose on the surface, but includes malicious code that will be executed with it. Examples include the Sony/BMG rootkit: the program provided music legitimately, but also installed spyware.
- Methodology: The approach works by generating a malicious binary that is used to compile compilers. Since the compiler code looks OK and the malice is in the binary compiler compiler, it is difficult to detect.
- Results: The system identifies construction of login programs and miscompiles the command to accept a particular password known to the attacker.
- Take away: Thompson states the "obvious" moral that "you cannot trust code that you did not totally create yourself." We all depend on code, but constructing a basis for trusting it is very hard, even today.

Reading a paper

- Everyone has a different way of reading a paper.
- Here are some guidelines I use:
  - Always have a copy to mark-up. Your margin notes will serve as invaluable sign-posts when you come back to the paper (e.g., “here is the experimental setup” or “main result described here”)
    - Digitally: Zotero, Mendeley
  - After reading, write a summary of the paper containing answers to the questions in the preceding slides. If you can’t answer (at least at a high level) these questions without referring to the paper, it may be worth scanning again.
- Over the semester, try different strategies for reading papers and see which one is the most effective for you.

Reading a systems security paper

- What is the security model?
  - Who are the participants and adversaries
  - What are the assumptions of trust (trust model)
  - What are the relevant risks/threats
- What are the constraints?
  - What are the practical limitations of the environment
  - To what degree are the participants available
- What is the solution?
  - How are the threats reasonably addressed
  - How do they evaluate the solution
- What is the take away?
  - key idea/design, e.g., generalization (not solely engineering)
- Hint: I will ask these questions when evaluating course projects.