Appending Security Theories to Projects in Upper-division CS Courses

Vahab Pournaghshband

Network and Security Laboratory (Nets Lab)

Computer Science Department

University of San Francisco

vahab.p@usfca.edu

Software security attacks: *a pervasive problem*

- 2017: An attack was reported to a connected computer globally for every 39 seconds on average.
- 2020: Over 500,000 Zoom account credentials were hacked and made available on the Dark Web.
- 2020: Security flaw in WhatsApp was exploited, enabling hackers to install surveillance software on users' smartphones.
  - It may have impacted WhatsApp's 1.5 billion users.
- 2021: Colonial Pipeline attack
- *Teaching security is an effective approach to prevent security vulnerabilities.*

Challenges in Teaching Security

- ***Objective:*** Exposing students to security concepts can be incorporated into upper-division CS courses without increasing the required efforts normally needed by students as well as the instructor.
- ***Challenges*** in teaching security in non-security courses:
  - The instructors might not be security experts.
  - As the instructors rightfully claim, each one of these courses is already packed with concepts and materials developed toward that course, leaving not much room for other topics.
- *Our proposed approach addresses all of the above.*

Methodology

- ***Approach:*** Developing an additional design phase, that is security in nature, to an already existing course project.
- The project design phase is a combination of implementing both attack and defense mechanisms to the system implemented by the students in the original project.
- By appending a security phase to existing projects, our approach is easily adaptable to course projects currently employed in various upper-division courses.
- The project specification should be self-contained:
  - So it does not require in-class discussions on security topics.
  - It must include all the security terminologies, definitions, and preliminaries relevant to and required for the project.
  - The project specification also provides additional resources pertaining to security which enables students to conduct independent research in learning additional security concepts they may need in order to complete the project.

Rationale

- It allows the students to make design choices. This way, they understand why one approach or algorithm is better than the other.
  - e.g, they learn when to use public-key vs. private key encryption.
- It enforces the ‘secure first’ view.
  - We want the students to learn the principle of keeping security in mind from the beginning by understanding the complexity of adding it afterward.
- Design problems naturally lead to various solutions to the same problem based on inherently different design choices.
  - Through end-of-semester in-class presentations, the students learn about other attack prevention or mitigation approaches presented by their classmates.

Project Design Phase: Case Studies

- We developed and presented multiple design projects.
- These may be used as examples of how to add security components to already existing projects.
- Projects in a senior-level operating system course and a junior-level computer network course.
  - These courses were selected since they are presented to students who will be entering the industry workforce or transitioning to graduate computer science programs.

Case Study: Operating Systems

- Design and implement a Shell
- Attack and defend your shell
  - Buffer overflow attack
- Process-Overload Attack
  - DoS attack, e.g., fork() bomb
- Code Shell Injection
  - Safe implementation of variable shell arguments

Case Study: Computer Network

- Peer-to-Peer File Sharing
- Authentication and Authorization
  - Add an access control list support e.g., Apache-style ACL
- Transmitting Encrypted Files
  - Hiding file contents from network snoopers
  - Hiding the existence of a file from unauthorized peers

Other Examples

- OS: File Systems
  - Race condition attacks
  - File-system level encryptions
  - Denial-of-Service attacks on file systems
    - Free space illusion attack
    - Deep directory attack
    - Empyty file attack
- OS: Ramdisk
  - Encrypted Ramdisk
  - Partitioned Ramdisk

Conclusions

- Software security attack is a pervasive problem.
- Teaching security is an effective way to prevent software vulnerabilities.
- There are challenges in teaching computer security.
- We address the challenges by developing an additional design phase, that is security in nature, to an already existing course project.
- The project design phase is a combination of implementing both attack and defense mechanisms to the system implemented by the students in the original project.
- We successfully employed our proposed approach in two of our core CS courses: Operating Systems and Computer Networks.

Questions?