NANYANG TECHNOLOGICAL UNIVERSITY

Developing Secure Software (Review)

presented by

Li Yi
Assistant Professor
SCSE

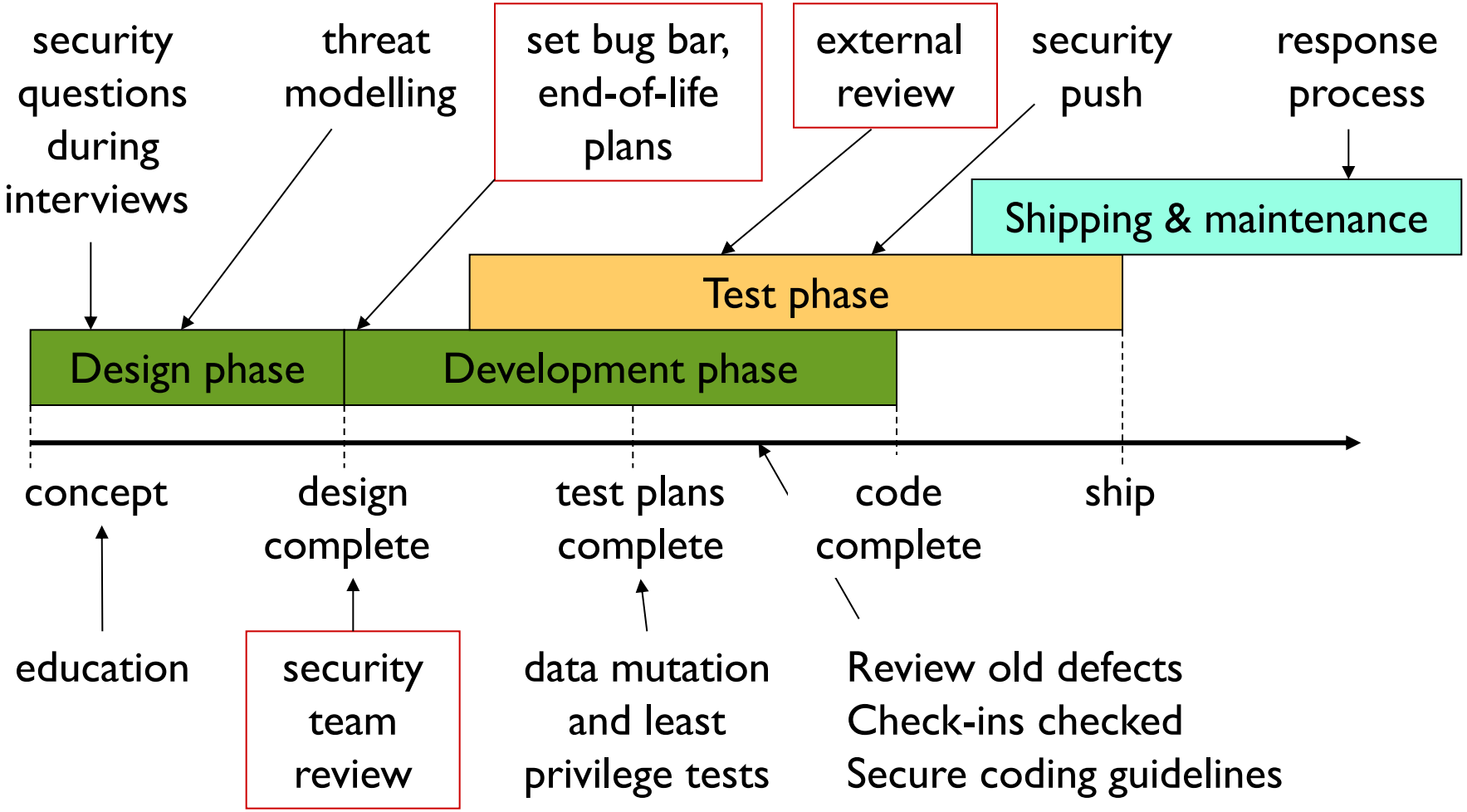
N4-02b-64
yi_li@ntu.edu.sg

Developing Secure Software

- Security is not an add-on feature
- You don't write software and then make it "secure" by adding a few security features
- You don't write software and then make it "secure" by removing vulnerabilities either
- Security is an ongoing concern throughout the software development life cycle:
  - Design/development – deployment – maintenance (patching, new releases, …)

From Howard & LeBlanc

Secure Software Development Life Cycle (SSDLC)



Summary

- As discussed throughout this course, there are various (often weird) ways to attack a software system
- Especially on the Internet, once a popular software is out there, various attackers spend time and resources, looking at it from various (often weird) angles and finding new ways to break into
- As defenders, we often cannot afford to spend as much time and resources to do those things; what we can do is
  - Security education and awareness
  - Follow security by design principles and standard secure coding
  - Perform rigorous security testing and deploy runtime verification systems if possible
  - Ship the product with secure defaults
  - Perform risk assessments and have adequate response protocols

SD^3 Principle

- Secure by design
  - Adhere to secure design and coding guidelines
  - Rigorous threat modeling, security testing
- Secure by default
  - Ship a product that is secure enough out of the box
  - keep the enabled feature set to a minimum; provide an easy mechanism for enabling the feature if it is required
- Secure in deployment
  - Educate the user how to use the system in a secure manner
  - A way to administer the application's security functionality
  - Create security patches asap

More on the Exam Paper

- 3 May 9 AM @ HALL F
- Structure similar to past year papers:
  - Five short-answer questions in 2 hours
  - Total 100 marks (20 each)
  - 60% of the final grade
  - Closed book
- To do well in the exam:
  - Basic security terminology and concepts
  - Read/write simple code snippets (C/C++, Java, SQL, JSP, Javascript, etc.)
  - Understand how different attacks work (environment, conditions, steps)
  - Know general defenses against each attack type (mitigations, bypasses, pros/cons)
  - Familiar with common security analysis and testing techniques

Course Contents

Black: not tested
Blue: basic understanding
Red: proficient

- Security fundamentals
- Measuring security (threat modelling)
- Stack overruns, integer overflows; DEP, ASLR, ROP
- Double-free vulnerabilities
- Object reuse, race conditions
- Meta characters, Unicode bugs
- Scripting languages: SQL injection, cross-site scripting attacks
- Generic countermeasures: regular expressions, taint analysis
- Security testing (fuzzing / symbolic execution)
- Generic countermeasures: type safety
- Secure software development

Good Luck!

See you at the final exam.