Homework 3 (10 points), SE 421X, due:2/2/2018
Schreck Mason

A. Software security has become critically important	1. Software is pervasive, it is used to automate factories, streamline commerce.
B. The state of software security is poor	3. New vulnerabilities are discovered every day. 5. Much of the activity that takes place under the guise of computer security isn't really about solving security problems at all. 4. Virus scanners, firewalls, patch management, and intrusion detection systems are all means by which we make up for shortcomings in software security. 6. The software industry puts more effort into compensating for bad security than it puts into creating secure software in the first place. 7. While developing software, we neglect to think about what would happen to our software if it were intentionally and maliciously attacked. 8. Many cybersecurity attacks are rooted in software.
C. Implementation bugs	9. Vulnerable APIs (e.g., gets(), strcpy(), and so on in C)
D. Code review process automated with a static analysis tool	10. Identify security bugs prior to the software's release.
E. A comprehensive approach to software security	11. Combine code review and architecture analysis.
F. Programming community tends to make the same security mistake	12. Buffer overflow was the number one cause of security problems cataloged by the Common Vulnerabilities and Exposures (CVE) Project [CWE, 2006]. 13. Morris worm occurred in 1988.
G. Defensive programming	14. Writing the program so it can cope with small disasters 15. Add code to check one's assumptions. 16. Makes bugs both easier to find and easier to diagnose
H. Why defensive programming is not enough for security	17. It does not guarantee secure software secure software. Instead of trying to compensate for typical kinds of accidents (on the part of either the programmer or the user), software security is about creating programs that behave correctly even in the presence of an adversary.(s16 LN1)
I. Secure feature	19. It is feature that is made secure
J. Security feature	18. It is a feature explicitly for security
K. Extreme Programming	20. Programmers write small tests (unit tests) even before the code is written.
L. Quality Assurance (AQ) is not adequate for security	21. It does not work for finding security defects that aren't related to security features.

Homework 3 (10 points), SE 421X, due:2/2/2018
Schreck Mason

M. A problem with penetration testing	22. Attackers have more hours to spend hunting for problems than the defenders have hours for testing.
N. Automated black-box texting	23. Find defects that do not require much meaningful interaction with the software being tested.
O. Fuzzing	24. It is likely to spend most of its time exploring a shallow portion of the program's state space.
P. Static analysis is well suited for security	25. It is well suited to security because many security problems occur in corner cases and hard-to-reach states that can be difficult to exercise by actually running the code. 26. By examining the code itself, it can often point to the root cause of a security problem. 27. It can find errors early in development.
Q. Problem with static analysis	28. It produces too many false positives.
R. Exploitability Gap	29. Ambiguous code makes it hard to come with a clear exploit.