Feedback — week 6 quiz

Help

You submitted this quiz on **Thu 4 Dec 2014 4:39 PM PST**. You got a score of **33.20** out of **39.00**. You can attempt again, if you'd like.

Question 1

What is *penetration testing*?

- A procedure for testing libraries or other program components for vulnerabilities
- A security-minded form of unit testing that applies early in the development process
- Whole-system testing for security flaws and bugs
- All of the above

Question 2

Which of the following are benefits of penetration testing?

- Compositionality of security properties means tested components are secure even if others change

- [ ] Full evidence of security: a clean test means a secure system
- [x] Results are often reproducible
- [x] Results are certain and not hypothetical

Question 3

What does it mean to "be stealthy" during a penetration test?

- ( ) Performing the tests from an undisclosed location
- ( ) Using encryption during tests to make the source of attacks impossible to determine
- ( ) Taking care to avoid activities during a penetration test that might attract attention, e.g., by operators or IDS services
- (•) Performing penetration testing without the target organization knowing

Question 4

What is a *web proxy*?

- ( ) A simulator for the web, for use when off-line
- (•) A piece of software that intercepts and possibly modifies requests (and responses) between a web browser and web server

- A piece of software that makes a web application look like a standalone application, making it easier to test
- An agent that makes decisions on the client's behalf when interacting with web applications

Question 5

What is **Nmap**?

- It is a network fuzz testing tool
- It is a suite of tools for scripting attacks: probe, construct, encode, inject, wait for response
- It is a map of the Internet
- It is a scanner which works by injecting packets to a range of addresses, and inferring what hosts and services might be at those addresses, based on the responses

Question 6

What is *ethical hacking*?

- Hacking into systems run by those whose ethics you disagree with
- Hacking systems (e.g., during penetration testing) to expose vulnerabilities so they can be fixed, rather than exploited
- A slang term for rapid software development, e.g., as part of hackathons

- ○ "Hacking" ethics so they justify unintended selfish behavior

Question 7

Which of the following statements describe *fuzz testing* (aka *fuzzing*)?

- ☐ It is a cost-effective replacement for functional testing
- ☑ It is a kind of random testing
- ☐ It is always purely black-box, in being indifferent to the software's functionality
- ☐ It is concerned with finding known-bad behaviors, like crashes and hangs

Question 8

Which of the following are true of *whitebox fuzzing*?

- [ ] **Radamsa** is (at least in part) a whitebox fuzzer
- [x] **American Fuzzy Lop** is (at least in part) a whitebox fuzzer
- [x] It takes into account the program's internals in some manner when deciding which inputs to choose
- [ ] It makes no sense to combine it with grammar-based fuzzing since the latter is just another way to consider the program's semantics

Question 9

Which of the following are true of *mutation-based fuzzing*?

- [ ] It works by making small mutations to the target program to induce faults
- [ ] It only makes sense for file-based fuzzing, not network-based fuzzing
- [x] It generates each different input by modifying a prior input
- [x] Each input may or may not adhere to a grammar, depending on the particular fuzzer

Question 10

Which of the following styles of fuzzer is more likely to explore paths covering every line of code in the following program?

```
int main(int argc, char **argv) {
  char buf[100];
  while (fgets(buf,sizeof(buf),stdin) != NULL) {
    int c = atoi(buf);
    if (c == 456799)
      printf("%s\n",(char *)c);
    else {
      int i = 0;
      for (i=0; i<c; i++)
        printf(".");
      printf("\n");
    }
  }
  return 0;
}
```

- Blackbox
- Generational
- Mutation-based
- Whitebox

Question 11

Which of the following are functions of a *network-based fuzzer*?

- [x] Acting as a "man in the middle"
- [ ] Acting as a server
- [x] Scanning a network address range
- [ ] Mutating network configuration files
- [ ] Acting as a client

Question 12

Suppose you want to use fuzzing on a program to try to find memory errors; which of the following statements are true?

- [ ] Compiling the program with address sanitizer (ASAN) will make errors *harder to reproduce*
- [x] Compiling the program with address sanitizer (ASAN) will make the source of a *memory error easier to find*
- [ ] Fuzzing doesn't find memory errors, it finds crashes and hangs
- [ ] You should not use a grammar-based fuzzer, because its adherence to the grammar means it will not find memory errors