This page is linked from http://breakingembeddedsoftware.com, Jon D. Hagar,
answers to exercises from his book "Software Test Attacks to Break Mobile and Embedded Devices"

Contents of this file (links)

Answers to Exercises from Chapter 1

1. Define three reasons why you test any piece of software

   Example Answer:

   - To show it works
   - To find bugs already present in it (break it)
   - To provide information on the qualities of the software e.g., performance, usability, compatibility (to hardware or something else), etc.
   - To provide information to developers or other stakeholders for decision making

2. Define three "bases" for testing

   Example Answer:

   a. Requirements
   b. Models
   c. Risks
   d. Mathematical combinations of input or output space
   e. Tester knowledge and exerience

3. Define three examples of embedded software devices

    Example Answer:
    a. Cell phone (mobile)
    b. “Smart” braking system in a car
    c. Flight avionics controller for an airplane
    d. “Smart” light switch with a microprocessor in it
    e. Medical insulin pump controlled by microprocessors

4. List the changes you want to make in

    Example Answer:
    a. Your testing – I want to test more efficiently and effectively so that I can provide the information of most value to the stakeholders who have interest in the information at the exact time they want the information.
    b. Your product testing – I want to find the bugs in the product. And by failing to find bugs when doing a series of “thinking” tests, provide confidence to the interest parties (stakeholders) that the product will satisfy users and customers.

Answers to Exercises from Chapter 2

1. You are hired onto a project. Management wants to improve testing on the project, which is part of why you have been hired. You find out the developers are not doing any developer level testing. List the reasons why they should improve their testing.
    Example Answer:
    a. The team is doing Agile, and one key ideal in Agile is that of Test Driven Development. The team should follow such important and accepted ideals.
    b. If a developer is not testing their code with some level of evidence provided then, they cannot provide information that the software is working as intended.
    c. Many of the most common embedded and mobile errors that escape into the field can be found by developer level evaluation activities such as: unit testing, static analysis, and peer reviews. To improve finding low level code errors, developer testing is a good idea.
    d. Errors are more cost effective to fix if found at the earliest point in time to when they are introduced.
2. The team has decided to use a static analysis tool and checks for a game app. Define which items you might check from table 2.2 and why.
    Example Answer:
    a. Uninitialized variable – the game has many variables for things such as counter, scores, players, and levels. These may cause the game to act different from device to device.
    b. Buffer overruns – overruns can use up (eat up) limited device memory.
    c. No return statement and unused code checks as these may indicate poor programmer practice.

d. **Note**: most other items in table 2.2 could be checked

3. Define a case where you can exercise a line of code, so you would have covered that line of code at a statement coverage level, but the line of code could still have an error in it.
   Example Answer:
   a. Line of code that could still be wrong even after coverage testing - $X = Y/Z$, where there is no guard that Z is not equal to zero, which could cause a divide by zero (undefined operation).

4. Define how your team will handle false positives in static code analysis.
   Example Answer:
   There are many possible ways to "handle" false positives, for example:
   a. The test team will analyze the SCA output and complete a first pass filter of messages before giving the messages to programmers so that they focus on a reduced set of "important" messages.
   b. Educate the programmer about how to look at the false positives during the code process by running "mini" SCA on each build and fixing problems as they code.
   c. Hire an outside company to run the SCA and do a first pass filter, before programmers get messages to address.
   d. As the project progresses, remove some message checks that are not finding bugs while producing large numbers of false positives.
   (**Note:** this comes at the risk of missing bugs)
   e. Have a local SCA expert analyze everything and fix bugs directly with the programmers.

5. Define who will use SCA on your team.
   Example Answer:
   a. The programmers (to find and fix bugs in their code before other testing)
   b. The testers (to find bugs that are hard to find in testing)
   c. Management (to understand code quality)
   d. Customers - regulators (to assure "rules" are being followed)

6. Can you think of a case where the low level developer Attacks 2 or 3 may not be worth doing?
   Example Answer:
   a. Consider a sample clock app with no user selection options, but just a cool looking display that uses system time and only a few lines of graphic code for "formatting". Unit testing in this case might not be of much value.
   b. Consider an initialization routine with no decisions and only required to set 10 variables to zero. Unit testing of this is of minimal value since a simple peer review visual inspection will be just as good.

7. Explain which items are needed for developer testing in the mobile or embedded environments.
   Example Answer:
   Items of higher value might include:

- Mock objects for hardware and software
- Software simulators
- Cross-compiler or CPU emulators
- Unit test tools that work with a CPU-bus monitor
- Unit test tools with instrumentation
- Configuration management tool with automated build and "kick off" features to support continuous integration, and metric analysis tool with reporting of coverage

8. Define why a tester would want SCA to be done.
   Example Answer:
   Testers should support and advocate SCA because it:
   a. Finds errors that are hard to find in testing
   b. Finds errors early
   c. Removes errors so testing can focus on "hard" bugs that SCA will miss
   d. Helps assess the overall quality of the code (i.e., do I need more or less testing?)

9. Explain two objections developers may have to SCA and what a tester can do to overcome these objections.
   Example Answer:
   a. Developers may feel like reviewing SCA output is a waste of their time since they won't be coding, or they don't like see false positives. To overcome these, a tester may need to educate the programmer, remove false positives, and/or turn in "real" bug reports to be fixed.

10. Define who can run developer level test attacks.
    Example Answer:
    a. Examples of who can run developer level attacks include: the developers, a second independent developer, an internal (to the company) independent tester, and/or an external (to the project) tester from another company.

Answers to Exercises from Chapter 3

1. List the environments that are different from a PC and how they could impact a smart phone with a touch screen display.
   Example Answer:
   a. Mobility – the device is actively moving around, which may impact networks, sensors, etc.
   b. Power Usage – the device runs on batteries and low/high display intensity may impact battery life.
   c. External environment factors – since the device can be used on the move, environmental factors such as outside lighting, water, heat, dust, noise (sound), electronic noise (e.g., EMI), and other factors will impact the device testing.

2. Define how long a long duration test should be run for a smart medical device used to record a person's heart and respiration rates.

   Example Answer:

   For this example, consider a mobile smart phone device that depends on batteries. Here are some long duration scenarios.

   a. Long run on batteries to drain them to "low" point and total discharge.
   b. Long run on battery and recharge (multiple cycles) over long time without turning device off.
   c. Long run with poor changing and/or no network connection.
   d. Long run for this device is defined as 1 week.
   e. Check for lost or corruption of data due to any of the above conditions (a-d).
   f. Are there heating or other usage problems seen during the above conditions?

3. Define where you have hardware-to-software or software-to-hardware interfaces that will need testing in a smart mobile phone with GPS, motion sensors, a touch screen, and Wi-Fi (or do this for your project's device).

   Example Answer:

   Internally to the smart phone, you will have the following interfaces to test:

   a. GPS to operating system to local app
   b. Motion sensor to operating system to local app
   c. Touch screen input sensors to operating system to local app
   d. Mobile cell phone data broadband connection to operating system to local app
   e. Wi-Fi data connection to operating system to local app
   f. Local app output to operating system to Wi-Fi connection
   g. Local app output to operating system to touch screen
   h. Local app output to operating system to cell phone data broadband connection

4. You are testing an embedded control system that has two processors and they reconfigure from one side to the other side when the hardware stops working. Make a list of attacks you might apply and why.

   Example Answer:

   Consider this example for a specific avionics device (e.g., aircraft or spacecraft) which has redundant computer processors that have fail-over logic to move from one computer side to other side computers when there is a computer hardware failure e.g., memory checksum failure. To get the software logic working for a fail-over, the hardware must fail, but this can be hard to create. Further, there is a series of cases to consider. Here, a tester might create a special software or hardware configuration where a reconfiguration request (e.g., interrupt) can be triggered to allow the fail-over logic to execute. Once this test support feature exists, then the cases to consider might be:

   a. Fail the software/hardware on Side 1 and watch for configure to Side 2
   b. Fail the software/hardware on Side 2, and then fail hardware on Side 1 and see what happens

c. Fail the software/hardware on both sides at the exact same time and see what happens
d. Fail the software/hardware on Side 2, the restore the hardware to being okay, then fail the hardware on Side 1 and see what happens
e. Fail the software/hardware on Side 1, reconfigure to Side 2, restore the failed hardware on Side 1, and then fail hardware on Side 2 to see what happens

5. You are testing a controller for a robotic car. How might a long-duration attack be applied?
   Example Answer:
   a. Example: In dealing with the software in a car control system, which interfaces with vehicle hardware (engine controller, fuel usage adjustments, fuel sensors, information system, warning systems, etc.), run the car until it almost runs out of gas, but refuel without turning the car off, and keep running while driving usage tests. Does the system behave properly (e.g., reset sensors, controller, reporting, warning, etc.)?
   b. Example: Also, run the car over long durations with such "refuels" as item a, and then test in a variety of environmental conditions changing: heat, altitude, speed, fuel, road conditions (snow, rain, dirt, mud), and different drivers.

6. You are testing a "fighting" robot for the games on your mobile device. Define which attacks you might apply and why.
   Example Answer:
   a. Attack 33 – test overall functionality of the game
   b. Attack 25 – focus on finding bugs in the game
   c. Attack 26 – this is a game and so needs game attacks
   d. Attacks 22 and 24 – find bugs in user documentation and online help files since new users will need these.
   e. Attack 20 – test the game in series of user stories for different players

   Note: other areas to consider might be: developer level attacks of Chapter 2, other performance Attacks of 6, or hardware/software interfaces and control attacks.
7. Conduct risk analysis for a smart medical device used to record a person's heart and respiration rates.
   Example Answer Risk factors to consider include:
   a. Heart rate – slow, fast, normal, strong, weak, drop outs, flutters, skips irregularities, etc.
   b. Respiration – slow, fast, normal, strong, weak, irregularities, etc. with changes including hold breath
   c. Test with different apps loaded on the phone/device
   d. Test with different device hardware and operating system configuration
   e. Is cell broadband, Wi-Fi, or both network configurations used in communications
   f. How much data is recorded with no network communications before data is "lost?"

Answers to Exercises from Chapter 4

1. A manufacturer of diesel tractors with GPS positioning controllers has implemented a series of software driven controls (engine, GPS, automatic positioning, farmer display). Why would they go to test places outside of the lab and their field in the mid–West? Give specific examples of where

they might go and why.

Example Answer:

a. Test to some place outside of the USA, since GPS is a US system and might function differently in other countries
b. The Farmer's display might be based in English and of less use to non-US farmers, so do testing in non-English speaking locations
c. The lab and mid-west are "flat", the controllers might behave different at altitude or on steep terrain

2. What communication channels can a smart cell phone have these days and can you list any protocols these channels might use?

Example Answer:

Wi-Fi (2 way)
Cell system broadband data network (2 way)
USB Communication port (2 way)
GPS signal (input only)

Note: each of these has industry standards which can be used in testing protocols.

3. Define the fault cases for a smart cell phone.

Example fault cases answers include:

a. Talk and messaging at same time on slow network
b. Talk and app use at same time on slow network
c. Talk and movement sensor data usage at same time in a very hot environment
d. Talk during movement with high-to-low/low-to-high network signal changes
e. Lots of apps in memory, while messaging and voice talk are used (i.e., memory used up and high rates of communication)

4. Define safing logic for a car.
    a. How would you test this car?

Example Answer:

The question here is what needs to be "safe" in a car. This depends on the car and what is "smart" as well as what poses threats to the user or the vehicle itself. The options include as examples: braking, engine controller, information used by the driver to make decisions, transmission controller, skid controllers, and most likely others. Each of these systems can pose a safety (to a human) or hazard risk (to the vehicle or environment). These risks may indicate safing logic and faulting cases which should have been considered in design and/or testing. To test these, I would start with planning by doing risk analysis and maybe a FMECA/FMEA. From these I would define and design test attack cases using a variety of techniques, e.g., mind map, scenario tests, combinatorial test, and controller test (attacks 4, 5,6,7, and 8) to name a few. This would provide some information to help in assessing if the safing logic was "good enough".

5. Define the different "users" of your embedded software system (or if you do not have this, define it for a car).

Example Answer: For a car, I might define users including:

a. A human driver
b. A human passage (may be different use/test cases that a driver)
c. The network the car is interfaced with (this could be several interfaces, e.g., cell, Wi-Fi, internet, servers, USB, etc.)
d. Software – apps in the entertainment system, comm apps, control apps, and all the embedded device apps

e. Hardware – every major system in the car, e.g., drive train, engine, braking, steering, speed control, etc.

Note: The trick to answering this question is don't think about just "human" users

6. Define the different interface points of your embedded software system (or if you do not have this, define it for your car's anti–lock braking system. You can research the basics of such a system on the Internet).

Example Answer:

Anti-lock breaking example might include: interface to the human driver via the break petal, interface to the engine controller, interface to the cruise control (may be part of engine system), interface to the brake hardware control, interface to sensors to determine "slippage" of wheel, interface to vehicles status control system (to display failures).

7. Define risk factors (hardware, software, faulting, etc.) for one or more of the following:

- Smart phone being used to monitor a person's heart rate

  Example Answer:

  Risk 1 : If a conflict in the software apps impacts the monitor feature, a heart failure can be missed. Attack 33

  Risk 2: If the connection interface(likely blue tooth) between phone and monitor sensor fails to follow IEEE standard, data gaps may exist which means doctors will not get a complete picture of heart functions. Attack 5

- Commercial passenger airplane flight control system

Example Answer:

Risk 1: If the flight control logic is not calibrated correctly, flight control surfaces will not be configured correctly resulting in plane performance issues up to and include crash

- An embedded device controlling the fuel rods in a nuclear plant

Example Answer:

Risk 1: If a radiation monitor device which contains hardware and software contains an interface problem between hardware and software, then a human may be over exposed to radiation resulting health impacts.

- Factory assembly line robotic welder embedded system

Example Answer:

Risk 1: since this is a PLC with a network connection, if this system is infected with a STUXNET type virus, then a security threat exists where the device could damage property or humans

Answers to Exercises from Chapter 5

1. Define attack tests for the following communication commands.
   - System "ok"
   - System "ready"
   - System "memory location corrupted"

   Example answers: I would apply Attack 14 pattern, where it has been customized into an exploratory test charter where each of these commands will be triggered and the response planned. This might have to be done using an automated computer script depending on the nature of the mobile/embedded device since these commands are "low level."

2. You are testing the embedded software for a car, which can have the following different features: engine type/size, number of car doors, entertainment systems, communication systems, and seat configurations. Define data values that you can think of to check that the software might have, and how you arrived at testing those values.

    Examples to list include:

    a. Number of doors (could be a variable depending on car type e.g., two door, three door, and four door)
    b. Configuration of entertainment/info system e.g., how many options exist
    c. Engine selection and options i.e., might need to be put into a table
    d. Vehicle sensor and device option configurations i.e., might need to be put into a table
    e. Weight of car (min and max value as well as overall stress value)
    f. Fuel options and configuration i.e., might need to be put into a table
    g. Sales location (U.S. or non U.S.)

3. You are charged with testing a car's anti-lock braking system (which has software). Define some simulation or stimulation input factors for your tests.

    Example answers include:

    a. Dry road
    b. Wet road
    c. Ice road
    d. Dirt road
    e. Variations of each of these: speed, direction (turning), transmission setting, different human users, etc.

4. You are testing a cell phone app that presents weather to its user based on communications from a server. Define some simulation or stimulation input factors for your tests.

    Example Answer includes:

    a. Slow network speeds
    b. User is moving (fast, slow, in/out of signal strength)
    c. Server is running full, out of date information, etc.
    d. Different locations (city, international, rural) of the device and user request.

Answers to Exercises from Chapter 6

1. You are trying to test interrupts in the following situations. Define how realistic it might be to test this way for each case that follows.

    a. Interrupt time test measures are accomplished by a human for a real–time system running audio controls for a car.
    b. Interrupt time is analyzed by automated comparison to a closed loop simulation, which is running at the same frequency as the processor running a car's braking system.
    c. Interrupt time is analyzed by automated comparison to a closed loop simulation, which is running at 2 times the frequency of the processor running a car's braking system.

    Example Answer:

    1a. Human time measure is likely "slow" (e.g., a stopwatch) and so any measurements of interrupt time processing may be inaccurate. But a "slow" response of an audio control is at least in part a human judgment factor of what is "good enough" from the performance time of an input to when the audio is adjusted. So a human "tester" can make a call of "good enough" interrupt processing here.

    1b. In this case, the timing analysis "tool" in the closed loop simulation is running at the

same speed as the system (braking) under test. This case has some realism problems because the "tool" system may not be "fast" enough to detect a problem that happened and the problem is "gone" before the tool finishes its processing cycle.
1c. This system has a little "less" of the problem that 1b has, but still has some risks to consider. An option might be to have triggers that can capture interrupts in real time as detected in the test environment

2. Define 3 possible time factor categories within a smart cell phone.
   Examples include:
   a. Real time for factors like GPS processing
   b. Alarm "sounding" for the user
   c. Time and date displays for the user
   d. Usage times (e.g., how much voice usage has occurred, but this may come from a server)
   e.
3. Define which performance criteria are important for a:
   - Smart phone
   - Car's anti-lock braking system

   Answer: Smart phone examples might include – user response time to input, memory performance usage, number of transactions per second, and number of apps that can be running concurrently.

   Answer: Braking examples might include - Detects environmental conditions within *.1 seconds* and changes braking system state conditions. Applies brakes within *.25 seconds* of "braking" command and/or pulses to anti-lock system at 5 times a second.

   Note: performance criteria include time and other physically measureable characteristics.

4. You are performance testing a new smart cell phone. What time-related tools would you consider and why?
   Example answers include:
   a. An ability to do real and simulated universal time code generator (external to device)
      i. If you can do item a, then have the ability to "turn clock time" ahead
   b. Performance analysis tool to show CPU and memory usage internal to device

5. Answer "device questions" from Attack 21 for a medical smart phone device. This device is a smart phone which monitors via a Bluetooth device (a sensor) a patient's heart rate and sends hourly text messages to the doctor.
   Example Answer:
   a. What things must work: sensor, Bluetooth connection, the software app, and the connection to the doctor's system? What else?
   b. Anything legal or contractually binding? Yes, this is a medical device and likely will need to comply with FDA regulations.
   c. Are there performance items that can kill? Yes, if the performance of the connections between the set of devices is not fast enough (not defined or ill defined), then a user may die as a result.
      Note: only the top three questions were answered here.

6. For the device in question 5, list the factors that might impact the software's performance.
   Example Answer:

Connection "speed" between the sensor, Bluetooth connection, the software app, and the connection to the doctor's system. A slow or missed connection in this sequence may pose a risk factor, so likely the timing of the integrated system under various performance loads and stress should be tested.

Answers to Exercises from Chapter 7

1. For the system you are testing, identify all user guidance information (software implementation) and then, conduct a risk analysis of what could be wrong with it.
    a. Or do the same for the "Winter Park Ski Area" app, which you can load on to your smart phone.
    b. Then conduct Attack 22 for 1 or 1a.

    1a: Example: "Live Pass" Tips can be wrong (wrong tips can create unhappy users who will not go to a ski area and/or spend money).

    1b: Example "Live Pass" has an on-line help menu (has getting started, sign-in, buy products, user twitter and Facebook, view and take photos, use the GPS map, track your runs/altitude, receive notifications, and about the ski area. These are in a scroll format, have web links, cover many topics including things like money, battery impact (power management), network data usage (messages, emails, etc.), all of which should be exploratory tested since they can impact smart phone devices.

2. Locate the "clock" app on a smart phone, and then define the alarms it uses in your system or do the same for a system you are responsible for testing which has alarms.
    a. What kind of alarms are they (short term, long term, sound, message, warning, or what)?
    b. Define the environments and conditions that might be important to test for the alarm.
    c. Define attacks for these alarms (input triggers, how to create, and expected results).
    d. Execute the attack.

    Example answer: This smart phone app ("clock") has a time function and allows the setting of multiple alarms, both short and long with a variety of sounds, levels, and features. It is intended to "alert" humans e.g., wake up, go to meeting, etc. This app also has a countdown stopwatch feature which alarms at zero. The test environment would be typical human user locations e.g., home, office, street, where ambient noise is low. The following attacks might be considered:
    - Normal wake up alarm at 7am
    - Alarm on stopwatch countdown
    - Vary Alarm times = 1 day, 1 week, 1 month, 1 year, AM and PM
    - Do a combinatorial attack 32 of all the option configurations
    - Stress case attack of loading 100s or 1000s of alarms
    - Check end/beginning of year
    - Check leap year
    - Check if device can be correctly moved around the "world" e.g., into different time zones and across date line.
    - To execute this attack, get your boss to send you around the world.

3. For a smart phone "tank" game (simulates a battle tank), identify the different possible users of the game's help file.

    Example Answer:
    a. Expert gamer

b. Newbie gamer
c. Gamer 5yr -10yr old
d. Gamer 11-13yr old
e. Gamer teen 14-19yr old
f. Gamer "old" (60+)
g. Gamer hacker (trying to find a security hole in the game to use in penetration)
h. Gamer "typical" player

Answers to Exercises from Chapter 8

1. Take out your smart phone and test your favorite app (or your project's app). Options to use for students include: a mobile travel app such as, Orbitz or Travelocity.
    a. Construct a storyboard
    b. Define the app's inputs
    c. Define the app's valid classes
    d. Define the app's invalid classes
    e. Construct a matrix of these (see Table 8.3 as an example)
    f. Determine minimum coverage for a set of tests
    g. Create data to cover these tests
    h. Run Attack 27

    Example Answer:

    The student should refer to Figure 10.2 for an example of a mind map story board. The beginning sample valid inputs would be: a userid and password as well as a trip plan (dates, travelers, and locations). Invalid input examples would be: bogus userids, password cracking, and strange trip planning (travel dates in the past, negative or hundreds of travelers, and fake locations). A table/matrix of these can then be created with minimum coverage mapping and data.

2. For your smart phone, find the "angry birds game" or pick your company's game. Load the app on to your device.
    a. Select and define your gamer role.
    b. Run Attack 26.
    c. Complete the checklists (see the checklist in Appendix F).

    Example Answer:

    This answer can vary a lot. I picked my gamer role as a 60-year old, non-computer user person. I ran Attack 26 and completed the check (parts of Appendix F). I determined the game "angry birds" is fun even for me as a non-computer user, but did need some help file reading.

3. Conduct Attack 25 on:
    a. Your project app, or
    b. The Orbitz travel app.

    No answer provided

4. Obtain the Orbitz app for your smart phone and conduct Attack 27.
    Show results?

    No answer provided

5. You are testing a violent war game for adult players. Who should NOT be included in the set of

"user" players and why?

Example Answer:

a. Children under age 13 should NOT be included as this would violate a “PG13” rating
b. Children ages 14-18 should be used to have “parental” control or input and so this might impact setting the test up (would the control work?)
c. People with strong “no violence” viewpoints should be included in the test team set up (beta testers), but the results might be to “filter,” since they will not like the game, but their feedback on help, warning, and controls could be important.

Answers to Exercises from Chapter 9

1. Without doing actual attacks, identify which attacks from this book and Table 9.1 that you might conduct on a mobile banking app (create an attack plan) and cite why you would run each attack.
   Answer: Attacks in Chapter 2 to be done by developers since this app involves a higher level of risk (banking transactions). Next, consider Attacks 32, 33, and 20, which would be done by a test team to assess functionality over a mathematically sound set of stories (extensively test functions and combinations). Finally, I would exercise all the security test attacks of Chapter 9, since this is a banking app (money is involved). Depending on what I learned during these tests, I would then consider modified test plans and attacks (exploratory testing).

2. Define how you would build a security test lab “sand box.”
   Answer: I would create a set of requirements for the lab. After this I would design, implement, and verify/validate the sand box facility to meet these requirements. Requirements would include:
   a. The lab shall have the ability to isolate physically the lab from the network/real world.
   b. The lab shall contain all production hardware.
   c. The lab shall be capable of loading configuration control software.
   d. The lab shall be capable of supporting operations/actions of human users:
      - Novice user
      - Skilled user
      - Expert user
      - Hacker user
      - Tester user
      - Admin user
   e. The lab shall have test support equipment:
      - Simulator
      - Test monitor computers
      - Test input generators

3. Define the spoofing vulnerabilities that a mobile map app might have.
   Example answer
   a. GPS spoof
   b. Network cell phone tower spoof
   c. Location on the map spoof

4. Build a risk matrix for Exercise 3.
   a. GPS spoof - Risk 2

b. Network cell phone tower spoof – Risk 1
c. Location on the map spoof – Risk 3

5. Research the offspring of Stuxnet and the kinds of systems (what types of embedded devices) they are targeting now. Make some notes to yourself about the characteristics of each one and take note of the tests that might find that "bug."

   Answer: The student is asked to do a search, for example Google and/or Wikipedia. These should return the answers which will change over time, so only a limited answer is provided here.

6. For mobile banking apps, define the "users" of the app, particularly identifying which ones might be or can become the "bad guy" hacker or cracker type users.

   Example answers:

   a. Bank customer
   b. Bank database server
   c. Bank web site server
   d. Local bank clients
   e. Remote transfer banks
   f. Penetration hacker

From a "bad guy" standpoint, the penetration hacker is the worst. But social engineering would teach us that a dedicated hacking attack would target items d and e too, looking for holes and backdoors. It might also be possible to hack into the bank's web server too and then, gain client information (item c), so this should be considered a risk. Finally, an assessment of the security of the bank's database server should be made. Of course, this assumes the bank customer themselves practices secure computing, which takes me back to where the hacker would start the penetration attacks (the customer side).

Answers to Exercises from Chapter 10

1. Do one of the following:

   a. Load your favorite app or the app you work on. Apply Attack 33. Draw a mind map, showing apps activities and links.
   b. Load a travel web app onto your smart phone, such as the Orbitz app. Apply Attack 33. Draw a mind map, showing app activities and links.

   Answer: Figure 10.2 did this for a "generic" travel site. Orbitz app mind map will be different, but similar. To do your test, each box and element of the mind map should be covered with one or more tests.

2. Produce a risk list for the software of Exercise 1. Define how you apply this to improve your testing/attack. (Refer to Appendix G for more on risk analysis.)

   Answer: Once the risk list is obtained and/or expanded, each risk is assessed and assigned a test priority. High priority risks should be tested first. As an improvement, if any time remains after testing high priority risks or if the stakeholder requests it (if time and budget are provided), lower priority risks can be tested. The involvement of stakeholders is an improvement over just doing requirements verification checking tests.

3. Produce a state chart for the software of Exercise 1. Define how you apply this to improve your testing/attack.

   Answer: Since this requires a picture, no direct answer is provided, however the text of this

chapter defines how to use state charts in attacks and testing.

4. Get a combinatorial tests tool (e.g., NIST or other tool) and load it on your computer. Define and apply combinatorial tests to the software of Exercise 1.
   Answer: since this is tool usage, no answer is provided.

Answers to Exercises from Chapter 11

1. You are field testing a car. List the environmental factors that you might try to find in the field to "stress" the car and how your mobile lab would account for these factors?
   Example Answer:
   a) Highway driving
   b) City driving
   c) Road conditions
   d) Different temperature conditions during driving (range -45c to 60c)
   e) Altitude (-100 to 14,000 feet)
   f) EMI
   g) Car conditions – working, failing, worn parts
   h) Different users – hardware and human
   i) Network conditions – hardware and air
   j) Systems I might evaluate include – info systems, diagnostic, in use safety, in use non-safety

   In my field lab, I would have the ability to record all of these. I would seek real world places where these can be encountered. I would do short and long duration runs. I would vary fuel, and human users, and even maybe the ability to introduce "hardware" ware. I'd go to "fun" places which might be different depending on the type of vehicle (commercial, consumer, luxury, basic, etc.). There is a lot more to testing cars, but this would be my start as I "learn."

2. You are working for a company testing new smart phones with 4–inch touch display screens. List the support items (hardware and/or software) you might want to have in your test lab environment to conduct your tests.
   Example Answer:
   - Many OS configurations
   - Many app configurations
   - A "test facility" where I can hook these up and monitor them, maybe with ability to automate
   - Test with real humans (can they read the display)
   - Tests in different environments – hot, cold, sun, bright light, city, country, mobile (walking, car, ship, other)
   - What are the display usages (risks) I should take in context e.g., medical, consumer, professional, etc.
   - Have we missed any environments (ask customers and stakeholders)?

3. You are testing an embedded pacemaker for humans, which has upload and download communication channels (wireless). Define the models and simulations you might want to have in your lab to test this software device.

Example Answer:

- Human body simulation model with
    i. Different heart rates
    ii. Many heart conditions
    iii. Different heart types
    iv. Heart size
    v. Medical conditions which impact heart
    vi. Model would need to be varied in real-time and simulated "real" conditions and reactions continuously
- I would want to have the ability to conduct an upload and download as needed.
- I might want to be able to upload "bogus" information.
- Can I hack the device with off-the-shelf tools (stuff from RadioShack) like a bad guy might and what damage can I do?
- Can I do long duration runs (days, months, years) in the lab with my simulations?
- What tools do I have to "probe" the device to see what is happening internally?
- Do my data/tests in the lab correspond to field installed usage?

4. You are testing an embedded process to control an airplane. List the kinds of tools for testing automation that you might want to consider and why.

    Example Answer:

    - SCA
    - Unit test tools (Attacks 2 and 3 support)
    - M&S
    - Automated GUI testing
    - Monitor and probe
    - Performance testing

5. You are being asked to define the test lab lifecycle for a new first of a kind "smart" electric car. List the stages your lab might evolve through and why.

    Example Answer:

    - 1) Rough prototype hardware and software lab – Testers would create such a lab for early assessment and feedback on the software system. It would not have all of the features and functions of a final test lab.
    - 2) Develop hardware - software lab – Testers would develop a lab, hardware and software, as a project inside of a project leading to a lab facility that would be used in actual production testing. This lab would support the lab development and verification of the lab leading to a final production version.
    - 3) Final production hardware in the loop software test lab – Testers would use and maintain this version of the test facility/lab to conduct testing. It will have "evolved" from items 1 and 2, which were not production grade to this final version.

Answers to Exercises from Chapter 12

1. Attack Planning—define a starting list of potential attacks for:
    a. Your project's device
    b. A travel web site app for cell phone

    Example Answer:

An example set of planned test attacks.

Attack 1: Static Code Analysis - My thinking here is any “important” code should have static analysis run; of course, this may depend on the types of code being done and tools available.

Attack 2: Finding White–Box Data Computation Bugs - The developer needs to check computations they are doing locally (on smart device), especially on financial and “travel” computations, though this may be more a web server side problem, so some checking may need to be done.

Attack 3: White–Box Structural Logic Flow Coverage - Ditto from Attack 2

Attack 14: Breaking Digital Software Communications - On such an app there will be real dependencies of many “features” with communication to the server. So a variation of the comm line (dip outs, weak signal, partial messaging, etc.) may be impact the app.

Attack 15: Finding Bugs in the Data - Here I am worried about “local” data and not some much the data coming from the server, since this should be “server” side testing.

Attack 16: Bugs in System–Software Computation - Here I am worried about the end-to-end computing between the smart app, comm lines, and the server. There are many layers where bugs can live impacting the overall user experience. I may need to define where to stop testing e.g., only the app, the smart device, or do I continue into the server layers. “It depends” is a factor here since the server may be tuned for a PC web browser, and not the web-app experience.

Attack 19: Finding Time Related Bugs – So we know users of smart apps, give only a short amount of time for the system to “perform” and maybe only give one to two chances before they delete the app. So time and performance (see Attack 21) may be very important for success of this app.

Attack 20: Time Related Scenarios, Stories and Tours - I would probably want a whole series of stories, users, tour, and usage scenarios. I would consider normal user, power user, and newbie users. I would use the checklist of Appendix F too.

Attack 21: Performance Testing Introduction - I would want to see the app under single user, multiple users, and huge numbers of users, including users of different types and styles (see attack 20). We may want to know how the app and server side perform as much as how the network performs. Successful performance testing must consider these.

Attack 22: Finding Supporting (User) Documentation Problems - So what does the on line (not help) information promise, which may help a potential customer to know what to download? What are the reviews (if any) saying? What other user documents are available? Is there a call-in phone help line script for customer support call center people to use (to make customers happy)? Examine all of these for errors.

Sub–Attack 22.1: Confirming Install–ability - Here I want to confirm the app will install and run on 80% of the market hardware, and OS versions (currently Android Samsung and IOS Apple). I may also want to see if the app conflicts with “standard” other apps, particularly competitor’s apps (when they are

installed). Because, I don't want my company's app *not* to work if some app that is likely to be present conflicts with mine.

Attack 25: Finding Bugs in Apps - This is an app, so I certainly will want to consider this attack, early in my attack execution cycle.

Attack 27: Attacking App–Cloud Dependencies - So what are the "cloud" dependencies here? For example, my user may book a flight on the phone, use a PC web browser to update things, and the app to check and get flight status. Is the data being shared, shared timely, and correct? I may need to do a little digging and exploration here.

Attack 28.1 Penetration Sub–Attacks: Authentication — Password Attack - Okay, so I am charged with evaluating the security of the app. Can it be cracked? What password strength is required? There is user data, financial info (credit cards?), and other personal/private info. How "safe" is all of this?

Attack 29: Information Theft—Stealing Device Data Here I am worried about local data kept on the smart device that might be hacked or stolen if a device is lost. I may not hit this area too hard initially, but as the device gains in usage, I'd worry about this because hackers go after higher use apps.

Attack 30: Spoofing Attacks - I would at least consider if spoofing is a possibility. To determine how far to take this attack, ask yourself about the context of the situation for this attack.

Attack 32: Using Combinatorial Tests - Such an app seems to have many possibilities for inputs, devices, options, and selection of scenarios. I would consider combinatorial tests once I have basic attacks (shown above) done, and I'm looking to "push" some testing.

Attack 33: Attacking Functional Bugs - I would do this early and often, probably with regression testing to make sure the needed features (and requirements) are working and continue to work. I would probably consider some form of automation in my lab, so that I can do these over and over. But, I would ensure that regression testing doesn't become "stale" due to repeating the same test over and over for no added value.

A long list to be sure and you might opt not to do all of these, but these are the ones I can think of. Plus, I would also do many of the "web attacks" found in Whittaker's books. We are talking money with this app, both for the customer and providers of travel services. Even small bugs, can impact the cash flow, hence, have a bigger plan.

2. Test Planning game
    a. Define characteristics you would test "Angry Birds" for (after 10 minutes stop).
    b. Define attack tests for each "Angry Birds" characterisitc (and how you evaulate these) in 10 minute cycles for each characteristic.

    Example Answer: Sample test attack sequence plan might be:

    - Do angry birds load?
    - Do angry birds launch?
    - Does the first display screen make sense?

- Do the “adds” and upgrade screen display (this is how money is made)?
- Can I select “poached eggs?”
- Does game launch?
- Can I play one cycle?
- Is score kept and reported?
- Can get to next level?
- Is (are) color(s) good?
- Does sound work?
- I would conduct Attacks 32 and 20. I would hit each feature/characteristic above in the attack and then I’d stress each of these.