TDT4237 Previous exams
Answers and hints

NOTE: this is not a complete solution proposal – just a list of what would be expected.

Exam 2006

Task 1
Answers: 1b, 2a, 3b, 4b, 5c

Task 2

a) Expected: a reference to the "fault model" for web applications. A discussion should include the three main parts – client, server, network – and potential attacks. Plus if discussion inlcudes how web applications differ from standalone applications.
b) Expected: name all the software security touchpoints as defined by McGraw and a brief (1-2 sentences) explanation of each touchpoint.
c) Not relevant. (Question from a guest lecture)
d) Expected: the priority list in the McGraw book. Should include: code that runs by default, code that runs in an elevated context etc.

Task 3
The taxonomy from Ch.12 in "Building Security In" (kingdom, phyla) should be used for classification.
Three main problems in the code:

- Exception handling – none at all in BuggyClass, in SubProcedure the catch clause is overly broad and the error message is written directly to UI -> information leakage.
  - Kingdom 5: Error handling. Phylum: Overly Broad Catch Block, Empty Catch Block/Lack of...
  - Solution/improvement: specific catch clause, write errors to a log only readable by sysadm.
- Lack of input-validation. None at all really.
  - Kingdom 1: Input validation and representation. Phylum: several may apply.
  - Solution: add input validation.
- race condition i SubProcedure. Same buffer is used to store both plaintext and encrypted text. Multithreading may cause the plaintext to be read before it is encrypted.
  - Kingdom 4: Time and state. Phylum: none of the ones in the book really fit, but (as stated in the book) this list is expandable..
  - Solution: atomic code/synchronized.

Note: You would have been expected to also include pseudocode demonstrate your solution.

Task 4

There is no one correct answer here – ”the sky is the limit”. But there should be a strong focus on security and security requirements should be specified in addition to other/regular requirements. A simple/overall risk analysis should be included (including business goals, business risks and technological risks). Plus if threat modeling (misuse cases/attack trees) is included

Task 5

Again there is no one correct answer. We assume that the risk analysis from Task 4 is used as a starting point along with the 24 known attacks from ”How to break..”. Plus if some attacks are not included – AND the student provides good reason why she chose not to include them. Also plus if some attacks are added that are not in the book – again IF a good reason for doing so is provided.

Exam 2007

Task 1

4 errors should be identified:

- ”security features – insecure randomness” to calculate loggId.
- ”error handling – overly broad throws” in writeToLog().
- ”input validation – log forging” no validation of what is written to the log.
- ”input validation – path manipulation” parts of the path to the logfile is added from input with no validation. Possible for an attacker to manipulate this path and control where the log is written to.

Task 2

a) Answers can be found in the lecture on Common Criteria and the ”Introduction to Common Criteria”

b) Attacks could include: SQL-injection, XSS, directory traversal, bypassing restrictions on input choices, tampering with hidden fields, CGI parameter tampering, URL jumping, buffer overflows, canonicalization, NULL-string attacks, command injection…

c) A security pattern is a standardized solution to a commonly occurring security issue in software. Expect the example pattern to be one of the patterns in ”Architectural Patterns for Enabling Application Security (Yoder, Barcalow 98)” – that is one of: Single Access Point, Check Point, Roles, Session, Full View With Errors, Limited View, Secure Access Layer.

d) Expected: an overview of the phases included in the RMF as defined by McGraw and a brief explanation of each phase. Also emphasis on iterations.

e) A race condition occurs when an assumption needs to hold true for a period of time, but actually may not. Whether it does is a matter of exact timing.
In every race condition there is a *window of vulnerability.*
Why a security issue?
- File access

- Time of check –time of use (TOCTOU)
    - A check precedes an action
    - The result of the check needs to be valid at the time of action
    - However, in the time between operations the result of the check may change

Task 3
The text specifically asks you to explain and *demonstrate* so it is not sufficient to say what you would have done – you are also expected to demonstrate how. A risk analysis is expected, and it is expected to include buffer overflows as the task explicitly says that C/C++ will be used. The answer should include: a requirements specification – including security requirements, a high-level design/system sketch, a risk analysis based on the requirements and the design, and also a ”mitigation strategy” for the identified risks. It is ok if all the phases are not complete (that is not possible given the time constraints of the exam) – the important thing is that all phases are included and some examples given for each phase.

Task 4
Do not expect the students to have extensive knowledge about threats specific to the health care domain, so whether the risks presented are realistic or not will not be part of the evaluation of the answer. What is expected: the use of misuse case and/or attack trees. Text only is not considered a compete answer.

Continuation exam 2007 (summer 2008)

Task 1

a) Expected: pseudocode demonstrating a simple sql-injection vulnerability and an explanation on how it could be exploited.
b) Expected: adding input validation the the pseudocode from a). The input validation should be specific either to the code an explain what it means to perform input validation for this sql injection vulnerability. Simply stating ”input validation” is not a complete answer. Should also include a general explanation of input validation (second part of question).
c) Expected: requirements, threat modeling, design.
d) Expected: challenges include complexity, code quality. Manual vs automatic: time consuming vs fast, skills required, false positives and noise.
e) This question is open to interpretation and opinions. In my opinion input validation is not an example of security functionality (you may use input val. for other reasons), but a strong argument would be awarded.
f) Expected: an explanation of when (final stages, production code and environment) and how (black box, outside in).

Task 2
To be able to perform a risk analysis, you would need to create requirements, security requirements and perform threat modeling. Also needs to identify business goals and risks. Expects all of these + the risk analysis to be explained (briefly with a focus on choice of method) and demonstrated. Does not expect a complete risk analysis (due to time constraints) but an emphasis on the risks of having a web-based system is expected.

Task 3
Part 1: should include a thorough discussion and reasoning for selecting methods to use. There is not necessarily one correct answer here, rather the strongest answer will be the one with the strongest reasoning behind the choices.
Part 2: Examples are not expected to be complete, but overly simple examples will not be sufficient to demonstrate the methods. A strength if there is a connection between the examples from the different phases so it is possible to see how they are related.
Part 3: expected to see some discussion on the challenges of being the one external person on the project and in charge of security that is often perceived as time- and resource consuming while not bringing any value. However this part of the question is no longer as relevant as the role of security lead was discussed in detail in the book that is no longer part of the curriculum.

Exam 2008

Task 1

a) White hat – someone who performs security testing on request from software owner.
Grey hat – someone who performs security testing on own initiative. May or may not alert software owner if errors are found.
Black hat – malicious hacker. Financial goal or to harm/take control over the system.
b) A metacharacter is a character with special meaning that may be interpreted differently depending on the receiver. Metacharacters are often exploited in injection attacks.
c) Two possible answers: the simple guide to risk analysis as presented on the lecture slides, or McGraw's RMF.
d) The caller cannot deny having sent the request. The service cannot deny having sent the response. The response is tied to the request. Digital signature over various contents of the message. Reject messages that do not adhere to policies.
e) Place some attack code somewhere in memory and overwrite the stack in such a way that control gets passed to the attack code.
f) Attack trees are used to analyse an attackers goal and potential ways for the attacker to reach that goals. Attack trees are used for threat modeling.

Task 2
The studenst should present the methods they choose to use and explain why. They should mention security reauirements, threat modeling, misuse cases, attack trees and preferably also

risk analysis. It is important that extensive examples are included to get a good score on this task.

Task 3

a) The answer should include: when (production), by who (external testers) and focus (outside in).

b) Expected to use experience from exercises. Should include a test plan. It is a bonus if the test plan is based on threat modeling and risk analysis.

Task 4

3 vulnerabilities should be discovered:
- both files are opened for reading and writing even if only reading is needed for the key-file. Classification: API abuse/privilege management
- no input validation (the comments clearly state that input is from a web form). Classification: Input validation and representation.
- it looks like the encryption key is stored in a file and that everyone uses the same key. Classification: Security features

1 vulnerability could be discovered:
- ”buffer” is reused and as this is Java code this could result in the wrong message being sent. Since all share the same key the receiver can still read it. Classification: Code quality