THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2009
Campus: City

COMPUTER SCIENCE
Software Security
(Time allowed: TWO hours)

NOTE: Attempt **ALL** questions in the 12-page script book provided, using approximately **25** words to answer each 5-mark question, **50** words to answer each 10-mark question, and approximately **75** words to answer each 15-mark question. Total possible: **100 marks.**

This is an ungraded sample exam, which should take you about 25 minutes to complete.
Sample answers from students are shown in blue.
Instructor's comments are shown in green.

A. Wang et al., in "On the Security of Delegation in Access Control Systems", define dynamic enforcement as follows.

> In dynamic enforcement, the initial state γ of the access control system is recorded. For every workflow instance X, the system maintains a list U_X of the participants for the instance. Every user who executed a step of X is added to U_X. When a user u requests to execute a step s, the system checks whether he/she needs to use a delegated privilege. If a delegated privilege r should be used by u to perform s, then both u and the delegator of the privilege are added to U_X. Note that if u has received r from multiple delegators, u has to specify the delegator of r for the execution of s. At the end of the instance, the system checks whether the users in U_X can complete the workflow in γ without delegation. If they can, then the execution of X is confirmed. Otherwise the system gives warning that users in U_X have enhanced their own power through delegation. The execution of X is rejected.

1. Briefly describe a workflow X which can be rejected (i.e. rolled back) safely. **(5 marks)**

To be rolled back, the workflow must to some extent be reversible. Since many operations that deal with data creation or modification (adding a record to a database, creating a file, modifying a file, etc. ...) are inherently capable of being rolled back, databases and file systems are designed to be, these operations are capable of being rolled back safely.

4 marks. This is a good discussion of a relevant issue, however the student hasn't considered the people involved in the workflow. When someone reads the contents of a file, then this read operation is essentially irreversible (because we can't reliably force people to "forget" what they have read), and this may be a breach of a confidentiality requirement. This issue was mentioned in our classroom discussion of this paper.

A workflow that results in the creation of a file could be rolled back safely via removal of said file, provided the creation and removal are atomic action performed in sequence. The atomic nature of these actions is important to the safe rejection. If actions were not atomic then the created file could become a dep[end]ency for other processes before its removal.

4 marks. This student has noticed that the system state must be rolled back. However their discussion seems to be limited to activity within a computer system, for example they refer to "processes" but not to workflows (which involve people). I'm left with the impression that this student is writing about the atomicity property of a database system, rather than about the security properties of an access

control system. The atomicity property of a correctly implemented database system *might* be a useful defense against a hostile user who is attempting to violate an integrity goal. Strictly speaking: preserving atomicity during a non-adversarial use of a database system is a functional goal, not a security goal.

2. Briefly describe a workflow *X* which cannot be rejected safely. **(5 marks)**

(Q1: If a user trys to read a file outside of there security level the execution can be stopped safely.) Similar to above but instead user writes to a file, if the previous state of the file was not stored it can't be rolled back.

1 mark. I'm having trouble understanding this student's answer. In their response to Q1, it seems they're assuming that an unauthorised user's attempt to read a file can be "stopped" before the user has been allowed to read the file. In Q2, it seems they're assuming that an unauthorised user's attempt to write a file cannot be prevented, and the file-write operation must be rolled back. It seems possible that this student has a good understanding of the relevant security issues, but they have not provided me with enough detail for me to be sufficiently confident of this to award them high marks. The apparent contradiction between the Q1 and Q2 responses suggests that this student does *not* have a good understanding. I'm awarding one mark for the appropriate mention of the user's security level.

An operation in which previously available information is either modified or is unrecoverable – probably the most extreme case would be device input, in which the rejection of either collection or storage would result in the input being unrecoverable under any circumstances.

5 marks. I'm a bit disappointed that this student seems to be considering only correctness (a functional property) rather than any explicit security property, however their reasoning is cogent and device I/O is a very important consideration in any rollback attempt.

3. Consider Lampson's "gold standard" for implementing security: authenticating principals, authorizing access, and auditing the guard's decisions. Which (if any) of these three mechanisms is implemented by dynamic enforcement, and which (if any) of these mechanisms are required by (but not provided by) dynamic enforcement? **(10 marks)**

The dynamic enforcement has implemented both authorizing access and auditing the guard's decisions. By checking for delegate[d] privilege, the system is author[iz]ing access – if the user has it. Auditing is done on the final step, where by the execution is checked (checking whether they should have allowed access.) However, authentication is not present in the system. It does not verify that user *u* is indeed the right person (or maybe it is done elsewhere?). It could be possible for user *m* to impersonate user *u*, who has the privilege to do something that *m* cannot.

6 marks. This student has correctly identified the authorization function of dynamic enforcement, with a brief discussion that gives me full confidence in their understanding. They have noted, accurately, that there is no mention of authentication in this passage; and they have indicated why authentication is necessary in any access control system. However, they have incorrectly identified part of the authorization mechanism of dynamic enforcement as providing an audit function.

Authenticating pri[n]cipals and authorizing access are both implemented by dynamic enforcement. Auditing of the guard's decisions is not provided by dynamic enforcement.

1 mark. If this were a series of three multiple-choice questions, then the student would have gotten 2/3 of the total marks. However this student has provided no explanations for their answers, making me wonder whether they are merely guessing. It is, of course, possible that this student actually understands the concepts of authentication, authorization, and auditing; and that they are able to apply these concepts to the quoted passage; but I see no indication of this understanding in their answer.

4. Consider the Transactional Memory Introspection (TMI) architecture for reference monitors, as described in the article by Birgisson et al. Is the TMI architecture suitable for implementing dynamic enforcement? **(5 marks)**

TMI enforces authorization policies by authorizing information leaving the system by creating a custom authorization module that implements the idea of dynamic enforcement we can implement a dynamic enforcement. This is because we are able to do some checks on the information as it leaves the system. I.e., is the information allowed to leave if there were not delegation taking place between the entities involved.

1 mark. I see no indication that this student understands dynamic enforcement, however they have shown a little ability to paraphrase the quoted paragraph.

With transactional memory introspection, each transaction in memory is checked against its previous states as in delegation in access control systems. Take for example the recording of a state before a workload is executed. Our workload can be likened to the transactions in TMI.

2 marks. I'm not confident that this student understands TMI. I am pretty sure – but I'm not certain – that they are using the word "workload" to refer to a workflow (or perhaps to a set of workflows? or perhaps to something else entirely?) in the dynamic enforcement mechanism discussed in the article by Wang et al. I might have given them only one mark for this answer, but this is a pretty hard problem and they have noticed, correctly, that the dynamic enforcement mechanism is transactional.

Transactional memory introspection allows us to ensure a consistent state for a system and the respective reference monitor. As we have already stated in (2) managing state may prove difficult for such a dynamic delegation system. I am willing to guess that by using transactional memory we can successful[ly] roll back these changes into a consistent state such that the limitation in (2) is no longer a problem.

5 marks. This student clearly understands TMI and dynamic enforcement. They correctly point out that TMI may offer an appropriate support for the rollback process of dynamic enforcement, and that there's some uncertainty about this level of support.

C. (Other questions). **[75 marks]**
