THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2008
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. Gurpreet Dhillon, in "Designing Information Systems Security…", is highly critical of the methodology that was used when specifying and designing an information system for a hospital. In the closing section of this article, Dhillon considers each of the primary modules of the hospital system in turn, noting how its *reference* (the system's actions, resulting from its actual use by its end-users) differs from its *usage* (the actions specified for the system).

Dhillon reports that the hospital's "Manage Pharmacy" module has the *usage* of transforming the maintenance of pharmacy records from a manual, paper-based procedure into a computer-based operation. The *reference* of this module is that it "Falls short of fulfilling the basic objective of pharmacy costing and drug utilization reviews." Security concerns may arise from any discrepancy between the usage and the reference. In the case of the Manage Pharmacy module, Dhillon concludes that there are "Threats of vulnerability to competitors and integrity problems of the pharmacy processes."

1. Critically and appreciatively discuss Dhillon's *reference/usage* analysis of a secure system, in the context of Lampson's assertion that "Studying a secure system involves three aspects: Specification/Policy … Implementation/Mechanism … [and] Correctness/Assurance."
(10 marks)

Student 1: Dhillon's reference/usage captures two important parts of a secure system. They are referred as Specification/Policy, and Implementation/Mechanism in Lampson's analysis. The usage or Specification/Policy here is the first step toward a secrue [sic] system. It defines what is considered to be secure in a particular system. Without a clear defined context, it is no point to work on a workable solution. After what needs to be done is defined, Implementation can be carried on to achieve such goals. Ideally the finished implementation can be carried on to its job and fulfil the goals. However it's most likely something is missing. Here, in Dhillon's framework, it would be considered as a failure. But Lampson's frameworks goes one step further. He adds Correctness/Assurance as a part of a secure system. Because security breach is anticipated, if such attack could be effectively detected and recovered from any loss possible, it should be considered as secure too. It's the outcome of the whole system as a whole that should be evaluated. And Dhillon gives up the chance to rescue the system in such situation.

This response clearly identifies an appropriate mapping of two elements of Dhillon's analysis onto the first two elements of Lampson's taxonomy. The student does not note any important differences between Dhillon and Lampson's first two elements. The student notes, correctly, that Dhillon's article does not consider the detection/recovery aspects of a secure system. The student incorrectly indicates that these types of defences are part of the Correctness/Assurance portion of Lampson's taxonomy, although they actually belong in his Implementation (where he identifies "recover" and "punish" as two of the five defensive strategies).

It can be difficult for an instructor to devise an appropriate way to award partial marks to a difficult question (such as this one). I'd probably award this answer about 8/10 marks, on the grounds that this student has successfully recognised a couple of Lampson's aspects in quite an unfamiliar setting, and that this student has also clearly identified a gap (relative to Lampson) in Dhillon's analysis of a secure system.

I don't expect students to be able to be able to quote the Lampson article in a closed-book exam, however Lampson's advice is on the development of security specifications is clearly inappropriate for complex information systems, with conflicting requirements by many different types of users – such as the hospital system described by Dhillon. Lampson asserts that "Each computer user must decide what security means. A description of the user's needs for security is called a *security policy*." I would hope (although this was a *very* difficult question) that a strong COMPSCI 725 student who had read Dhillon's article carefully, and who had then reviewed the Lampson article when preparing for the examination, would have noticed this contribution from Dhillon: that we should be very careful when developing our Specification. Otherwise, we run the risk of building a system that, at best, would correctly implement an inappropriate specification. Dhillon's table summarises some major disparities between what a specification "should say" versus what it actually does say. This is a major gap in Lampson's (otherwise quite strong) summary of a secure system, and is my primary appreciation of Dhillon's article. By the way, I think it very unlikely that Dhillon was the first person to identify and describe this security issue, even though his article was written in the mid-1990s (for otherwise his footnote #3 is inexplicable). The long publication delay *might* have been required under a contract this hospital trust insisted that Dhillon must sign before he was allowed to interview their employees. I have encountered similar advice elsewhere on many occasions. It seems to be a commonly held belief that our systems would be more secure if we put more effort into ensuring that these systems have implemented the correct type of security, than on ensuring that these systems have correctly implemented their security specification.

Student 2: There would be problems with its policy, if your records are held in a certain hospital, should other hospitals have access to it. How wide does the scope of the policy should extend, for example should it be all hospitals in a state?, maybe nation even world wide. Trying to implement such a broad system would be good in a world where everyone is willing to freely share public knowledge, but this might create more competitiveness among businesses, and when this happens, can we be assured that information being passed around is true.

This response does not make any clear references to the Dhillon article, other than in its use of the term "hospital". I am pleased that the student made an attempt to answer the question; I would expect them to get marks on other parts of this examination; and I would award 0/10 marks for this response.

Students 3 and 4 did not respond to this question.

0/10 marks

Student 5: Critically, Dhillon does not discussed about the Implementation.

0/10 marks. I wish this student had given some argument to support their assertion about Dhillon. I'd say that Dhillon's article carefully considers how users perceive the behaviour of the implementation: this is Dhillon's *reference* concept. However it might be that the student is considering the technical aspects of the implementation, *e.g.* what sort of cryptographic protocols it employs in order to implement a confidentiality or integrity goal. In the absence of any argument in favour of a non-obvious assertion, I won't award any marks. Some argument is always required to do well on my exam questions, for if all the assertions in a student's answer are obvious enough not to need any argument, then they are trivial enough not to be worth very many (if any) marks.

B. Hovav Shacham, in "… Return-into-libc without Function Calls (on the x86)", describes a vulnerability of the W⊕X defense. Shacham describes the W⊕X defense as an assurance that "no memory location in a process image is marked both writeable ("W") and executable ("X")." He notes that, in a system with this defense, "there is no location in memory into which the attacker can inject code to execute."

2. Discuss Shacham's attack, using terminology from COMPSCI 725. To receive full credit, you must indicate whether Shacham's attack is primarily on the Secrecy, Integrity, Availability, or Accountability goals of a secure system; and whether this attack can be used to compromise the other types of goals. You must discuss the defense (W⊕X) which this attack circumvents, indicating why it should be classified into one of the following categories: isolate, exclude, restrict, recover, and punish. Finally, you should consider defenses to Shacham's attack, in the context of your home PC, by indicating what you would do if you believed your PC would be the target of a Shacham-type attack. **(15 marks)**

Student 6: Shacham's attack of overwriting return addresses via a buffer overflow and returning into existing code is primarily an attack on the integrity of the system. This is due to the corruption of data. All other security goals can further be compromised given the attacker can now execute arbitrary code on the system. Secrets can be accessed, accesses to data/resources denied and logs changed.

The W⊕X defense restricts memory from being writeable and executable. This categorizes it as a restriction defense; attackers have no place to both write and inject their code which is necessary in a standard buffer-overflow attack. However Shacham's attack is not restricted by these limitations as it does not require the ability to write memory.

To defend from this attack, given that the root of the attack is still buffer overflows, it is still important to have all patches applied to software. Some software will always be insecure though so it is important to protect yourself with recovery schemes such as backups and to run vulnerable programs at a low user level so that when a process is attacked it has limited priviledges [sic] to perform further attacks.

This is a very nice response, one which clearly addresses all of the requirements. I am confident that this student has read Shacham's and Lampon's articles carefully, that they understand what they have read, and that they are able to apply this knowledge in a (slightly novel) setting. This is a fairly easy question. 15/15 marks.

Student 7 did not respond to this question.

0/10 marks.

Student 8: This attack is primarily intended to circumvent the goal of integrity – the architecture can be used against itself, thereby allowing the attacker to run arbitrary patterns of code – and potentially attacking the system secrecy, availability and accountability in the process.

This defines is primarily based on the 'isolate' strategy: in other words, bad code *is* allowed to exist, but the system keeps it separated from executable code, so that it can never (in theory) be used for any nefarious purposes.

The type of attack Shacham proposes is particularly problematic for code that is written in C and that executes, unconstrained, directly on hardware. The two primary strategies I would consider if I were concerned about this attack are:

- Try to avoid running C code, instead preferring managed [?] code (like Java or .NET) – this executes inside a virtual environment, so the scope for these attacks is lessened.
- If I had to run some arbitrary C program that I didn't trust, I would run it inside a virtual machine (like VMWare) to isolate it from the rest of the system. This would not stop the attack but would ensure it could not do any damage to my real computer or data.

This is a very good response. The student is not working very accurately with Lampson's definitions.

Lampson describes "restrict" in the following way: "let the bad guys in, but keep them from doing damage. This fine-grained strategy, also known as sandboxing, can be implemented traditionally with an operating system process or with a more modern approach that uses a Java virtual machine…"

Lampson's "isolate" strategy has a very narrow range of application: "keep everybody out". Lampson would reserve the use of the term "isolate" to extreme measures such as turning off a computer, or unplugging it from all networks ("air-gap security").

I'd deduct only one mark for the skewed usage of "isolate", because the student has provided a not-unreasonable alternative definition. 14/15 marks.

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