Software Security
415.725SC
Lecture 31: Sample Final Exam

Clark Thomborson
University of Auckland

Q1. Legal, Ethical and Conceptual Frameworks

- Consider the three goals of security, defined in Pfleeger's book: confidentiality, integrity, availability. Also consider his classification of assets: hardware, software, data.
- Which of Pfleeger's security goals, for which of Pfleeger's asset classes, are protected by the following clause of the Computer Science Department's Computer Systems Regulations of 17 September 1999:

  "No person shall … use a login name other than the one(s) assigned to that person by the Department or allow any other person to use that person's login name(s) to access one of the Department's computer systems without the express permission of the Director of that system."
- Explain your answer briefly (in approximately 50 words).

Student 1's Answer to Q1

- ***The users logon (data) (and password) is confidential as it is ONLY for a specified user. It is NOT for anyone else. It also has an integrity aspect where whoever has the logon can change the password portion of it.***
- ***Also there are weak links to other aspects of Swe [?] data integ/confid avail. Whoever shouldn't accesses the system has access to whatever's on the system.***

My Evaluation of S1/Q1

- *This student didn't realise that the clause covers hardware, as well as data; the student makes a brief reference to "SWe" which might be a non-standard abbreviation for software.*
- *Confidentiality is correctly noted, however the student didn't realise that the authorisation aspect of Pfleeger's "integrity" goal is the primary focus of the quoted clause.*
- *My evaluation: C+.*

Student #2 on Q1

- ***This covers confidentiality and integrity.***
  - ***Confidentiality because the users whose account is being used has a right to privacy in his or her home dir.***
  - ***Integrity because the account may have access to resources that the "hacker" should not have access to, the hacker may alter data structures in a way s/he shouldn't.***
- ***It covers all the classification of assets as using a given account may give unauthorized access to any of the items, hardware (changing h/w settings), software (running unauthorized s/w) and data (access to data).***

My Evaluation of S2/Q1

- *This student correctly identifies the goals and asset classes, in an answer that indicates good understanding.*
- *A minor defect in the answer is that it doesn't consider the context of the quoted clause, leading to a misplaced emphasis on confidentiality.*
- *It is reasonable to suppose that these regulations (and this clause in particular) were designed primarily to insure the integrity of the computer science department's assets, not to protect the confidentiality of our users – so confidentiality is a secondary goal, not a first-equal goal to integrity as is suggested in this student's response.*
- *My evaluation: A-.*

Q2: Cryptographic Authentication, e-Commerce, and Secure Communication

- The first step in Aucsmith's "Identity Verification Protocol" is $F_0 = (H_A == K^1_A[K^{-1}_A[H_A]])$. In this formula, the variables have the following definitions:
  - H_A is a hash value computed over the code of module A,
  - K^{-1}_A is a private key of the Integrity Verification Kernel (IVK) embedded in module A,
  - K^1_A is a public key of the IVK embedded in module A,
  - F_0 is a flag value indicating the success or failure of an operation, and
  - $K^{-1}_A[H_A]$ is the signature of module A under A's private key. This value was computed at compile-time and is stored in some secret fashion within module A.
- Which of the following phrases best describes this step in Aucsmith's IVP? a) A verifies self; b) A challenges E; c) F verifies self; d) E responds to A; e) E verifies A; f) A checks response.

Student #3's Answer to Q2

- ***e***
- *My evaluation would be deferred until I read this student's answer to the following question. My delay would give this student a second chance to explain their incorrect answer. The student is probably confusing the first step of Aucsmith's protocol with the fourth step, which is "E verifies signature of A and reports any failure* $F_2 = (H_A == K^{-1}_E[K^1_E[H_A]])$*." However I don't expect students to recall all details of a complex protocol, so I would be open to giving partial credit to an incorrect answer if it is accompanied by a plausible explanation.*

Student #4's Answer to Q2

- ***A verifies self.***
- *My evaluation: A.*
- *Note: your final exam script will show a total number of marks possible for each question. I would give equal weight to each of the five sections of this sample exam, and I would give less weight to a multiple-choice question like this one. So if this sample final were a 100-mark test, question 2 would be a 5-mark question and question 3 would be a 15-mark question.*

Q3. (Continuation of Q2)

- Briefly explain (in approximately 50 words) the individual calculations or function evaluations made during the first step in Aucsmith's IVP.

Student #3's Answer to Q3

1. ***encrypt*** H_A ***with securet [sic] key of*** $K^{-1}{}_A$***;***
2. ***decrypt the*** $K^{-1}{}_A[\ H_A\ ]$ ***with public key of*** $K^{1}{}_A$***;***
3. ***matching the result from previous two steps with*** H_A***. If they are matching, set*** F_0 ***to true otherwise set to false.***

Note: the calculation is given in Q2 as

$$F_0 = (H_A == K^{1}{}_A[\ K^{-1}{}_A[\ H_A\ ]\ ]\ )$$

My Evaluation of S3/Q3

- *This student has made a major error in their answer, incorrectly including a preprocessing step (of computing the signature of A under A) in their description of the operations that would be taken at verification-time in Aucsmith's IVP.*
- *The student has made a second major error, by not clearly identifying the (rather time-consuming) computation required to obtain the current value of* H_A *by computing a hash-digest over the current code and static data in module A.*

My Evaluation of S3/Q3 (cont.)

- *The student has also made a minor error of misspelling the technical phrase "secret key".*
- *There are several grammatical errors in this answer, however it is quite understandable except for the technical confusion that would arise if I were to translate the misspelled word "securet" into "security". So these grammatical errors would have no effect on the student's grade.*
- *My evaluation: D. This student is saved from an "F" grade on this question by their appropriate use of the descriptive terms "encrypt" and "decrypt" in their explanation.*

Student #4's Answer to Q3

a) Firstly A computes a hash of its own code as it is currently running.

b) $K^{-1}_A[H_A]$: this is a stored value. When the module was compiled by the distributer [sic] the hash was calculated and encrypted with A's private key to give this value.

c) $K^1_A [\ldots]$: A decrypts the stored hash value using its public key. This occurs at run-time. It therefore recovers the original hash value that was calculated at compile time.

d) $H_A = K^1_A [\ldots]$: The current hash and compiled hash are compared.

e) $F_0 = (\ldots)$: If they mismatch the flag is set to failure.

My Evaluation of S4/Q3

- *The student shows excellent understanding of the protocol, by describing each step in good detail without making any technical errors.*
- *I would not mark down the misspelled word "distributor" because it is not a technical error.*
- *My evaluation: A.*
- *I am pleased to report that most students did very well on this question in the sample exam.*

Q4. Protection of Hosts

- Briefly describe one assumption about computer system security, which is valid for non-mobile systems, but is violated for mobile code systems.

Student #5's Answer to Q4

- ***In a non-mobile system the host is trusted. This assumption cannot be made by mobile code in a mobile code system.***

My Evaluation of S5/Q4

- *The student identifies an assumption ("the host is trusted") that was not discussed in the article on mobile code security you read for this class, so I would examine the student's answer on its own merits to decide if it shows reasonable understanding of any other article or lecture material in this class.*
- *Other class readings have discussed reasons why hosts should NOT be trusted unconditionally in all respects, even in non-mobile systems.*

My Evaluation of S5/Q4 (cont.)

- *S5's assertion about security in non-mobile systems shows very little understanding of the course material, and is unaccompanied by any reasonable definition of what the student means by a "trusted" host.*
- *Furthermore, this student does not explain why adding mobility to code should destroy all our trust in a host (to the point that we cannot even make this assumption).*
- *My evaluation: D-. The student is saved from an "F" by their naming of an important issue in computer security, even though it is far from clear that they understand this issue.*

Student #6's Answer to Q4

- ***The user is the principle and all programes run by the user do only what the user wants. In non-mobile code systems code can run with the rights of the user and be executed safly. With mobile code systems the user may not know what a programme does (or that it is running at all) hence there must be a differnt principle (programmer / distributer / signing authority).***

My Evaluation of S6/Q4

- *This student shows an excellent understanding of the article by Chess on "Mobile Security".*
- *One of the main themes of the article was explaining why "Identifying Programs with Persons" is a problematic assumption in mobile code systems. This assumption is generally true in non-mobile systems though (with a few exceptions, such as programs that run as "setuid root" under Unix).*
- *Within the space available, and in s/his own words, this student does an excellent job of summarising Chess' discussion.*

My Evaluation of S6/Q4 (cont.)

- *This student barely misses an A+ on this answer, by misspelling the technical word "principal" as "principle". Confusing a "principle" with a "principal" is a common error in technical writing, so I wouldn't mark down harshly for this.*
- *Also "programmes" may be spelled as "programs" but never as "programes".*
- *I would not mark down for the misspelling of "safely" and "different", even though both were missing an "e", or for misspelling "distributor" because these words do not have sharp technical meanings and the student's meaning was crystal clear.*

“Principal” and “Principle”

- *“A principal in a security system” is an actor.*
- *“A security principle” is a rule or guideline.*
- *A principal has the right to act (within limits) in a security system.*
  - *A principal (if it is a human female) may act in accordance with her ethical principles.*
  - *A principal (if it is a software object) may act in accordance with its design principles.*
- *An operating system might be designed with the following security principle in mind:*
  - *The security rights of a principal must be checked before it is allowed to act.*

Another “Principal” Word

- *In ungrammatical sentences, the adjective “principal” (meaning primary, main, chief) could be confused with the nouns “principal” and “principle” discussed on the previous slide.*

Copy Detection

5. Consider the following images.

   Sketch the image that would result if the Checkerboard (Figure 1) is watermarked with the Copyright Notice (Figure 2), using the least-significant bit embedding described in the paper by Johnson and Jojodia.

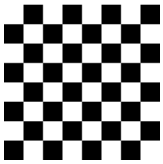

Figure 1. Checkerboard.

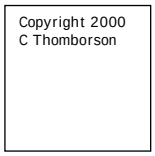

Figure 2. Copyright Notice.

Student #7's Answer to Q5

- ***Problem of depth of the cover image – assuming 256 level grey scale (8 bit). – 1 bit isn't much to encrypt in.***

  [The student's sketch shows the Copyright Notice overlaid on the Checkerboard.]

- ***-- except writing would distort the white. It would be really faint grey (slightly darker). Also the black checks would be slightly lighter as they have LSB set to 1, except where writing is – it would be full black.***

My Evaluation of S7/Q5

- *A+. This student shows an excellent understanding of the LSB embedding process.*
- *In addition, Student 7 correctly points out, and resolves appropriately, an ambiguity in my question.*
  - *The article discusses 8-bit and 24-bit images in detail, and mentions 4-bit images in passing.*
  - *As the student points out, 1-bit images are also possible, and my black/white image could indeed be encoded in 1 bit/pixel.*
- *The student's picture and explanation are accurate for 8-bit images.*

Student #8's Answer to Q5

- ***The sketched image depends on how many least-significant bits are used and how many bytes are used to present colors.***
- ***In this case for example, the figure 2: Copyright Notice consists of "Copyright 2000 Clark Thomborson" which is black and white which only need one bit to present its color or gray scale.***
- ***It nearly apply not changes to Figure 1 so the image will be the same as "Figure 1. Checkboard" as before.***

My Evaluation of S8/Q5

- *Student 8 has correctly pointed out an ambiguity in my question, however it is not at all clear what assumption they are making to resolve the ambiguity: have they assumed 1-bit or 8-bit pixels in the cover image?*
- *The student's last sentence is also ambiguous.*
  - *"[N]early apply not changes" implies that a small change has been made.*
  - *"[B]e the same as" means that no change has been made.*
  - *Perhaps the student is trying to say that the image "would appear the same to a human observer."*
  - *In their next answer, student #8 writes "the watermark is [undetectable] to our eyes" so I will use this statement to resolve the ambiguity of their answer to this question.*
- *The student's grammar is weak, but their meaning is clear except as noted above.*
- *Grade: A*

Question 6

- Characterise the watermarked image you constructed in your answer to the previous question, in terms of its fidelity, robustness, and security. Use the definitions of Matheson *et al.* for these terms.

Student #7's Answer to Q6

- ***Fidelity – is somewhat noticeable if you can see this kind of thing, being greyscale would not help. Probably most people would not see it, so the image would look fairly normal.***
- ***Robustness – something like JPEG would probably delete the LSB to improve compression. Would not be difficult to remove if you knew about it – simply replace LSB with noise or change all colours to be black or white.***

My Evaluation of S7/Q6

- *B+. The student has demonstrated an excellent memory and understanding of Matheson's definition of "fidelity".*
- *The student seems to understand the distinction between "robustness" and "security", even though the latter term does not appear in their answer.*
- *This student would have received an A evaluation on this question, if the last sentence in their answer were clearly intended as a discussion of the "security" of this watermark.*

Student #8's Answer to Q6

- ***In this example. Fidelity is not guaranteed in the case of only a few color or gray scale, however for over 250 colors or gray scale, we can think it maintains fidelity because the watermark is undectable to our eyes.***
- ***Robustness: obviously it is not robustness. The weakness results from the defects of the least-significant embedding algorithm itself.***
- ***Security: it is secure because it is hard to detect. It is no secure once it has been detected and it is ease to remove.***

My Evaluation of S8/Q6

- *A-. This student shows an excellent critical and appreciative understanding of the LSB embedding process, by applying Matheson's analytic framework accurately and fully.*
- *The answer is weak on "robustness", where the unhelpful word "obviously" and the vague "defects of the … algorithm itself" convey very little explanatory information.*
- *However the remainder of the answer is clear, direct and concise, even though it is somewhat ungrammatical.*

Copy Prevention

7. Consider the following assertion: “Any secrets carried in Java bytecode written today, could be easily attacked tomorrow by a reverse engineer who has access to the decompiler described by Proebsting and Watterson.” Make a brief argument for, or against, this assertion.

Student #9’s Answer to Q7

- ***We can view this assertion from two sides:***
  - ***it’s true: Because of the similarity between java bytecode and sourcefile, people make use of utility like javap to get the class file, and people have really developed many tools to reverse this classfile to sourcefile, such as tools in www.sourceagain.com or karokata.***
  - ***it’s not so true: In java 2, we’ve very rich security mechanism in security package, where we can sign the sourcefile to protect its copyright. We also have obfuscator to change the datastructure, variable, abstraction of java files. Actually, we can transplant every advanced security technology to java to protect the source files.***
- ***Finally: there’s no absolute secure system in the world, java is the most secure language among the existing ones. If there’s no important breakthrough in java security architecture, java can only patch the holes after the holes are found out.***

My Evaluation of S9/Q7

- *B. This student has demonstrated a very good understanding of the reverse-compilation process.*
- *They have correctly identified obfuscation as the main method of defense against decompilation.*
- *However, this student has incorrectly characterised the java2 code-signature techniques as providing a defense against a reverse engineer.*
  - *Signatures are a mechanism to demonstrate integrity and authenticity to the end-user.*
  - *A reverse engineer could easily remove or ignore signatures.*
  - *Indeed, a signature may assist a reverse engineer by assuring the integrity and authenticity of the code being attacked!*

Student 10's Answer to Q7

- ***The reverse engineering is developed very fast now. Sometimes we really see a lot of cases that the reverse engineer use the decompiler attack the java byte code but the security technical method is also developing such as java obfuscation watermark fingerprint etc.***
- ***Java bytecode is protected not only by the software control but also by the cyberlaw, by copyright, patent, ethical, moral etc.***
- ***So this assertion is not completely right.***

My Evaluation of S10/Q7

- *B+. This student has correctly named the relevant security techniques, however their answer is somewhat vague so I would not be confident enough of their understanding to assign an A grade.*
- *Indeed it seems likely to me that this student does not have a clear understanding of the difference between a preventative technical defense such as obfuscation, and a deterrent such as copyright.*
- *The question was whether the reverse engineer could attack "easily", not whether they'd be detected (e.g. by a watermark), or whether they'd escape punishment (e.g. for copyright violation) if their reverse-engineering were detected and proved in court.*

Bibliography Style

- You should include the following information for web material:
  - Name of webpage
  - Author's name (if available)
  - Date of publication (if available)
  - URL
  - Date you accessed it
  - Any other information that would help your reader find your reference.
- See http://www.unn.ac.uk/central/isd/cite/elec.htm for an acceptable style for online references, or copy the style from some paper we have read in our class.
- Example:
  - Touretzky, D. S. (2000) Gallery of CSS Descramblers. Available: http://www.cs.cmu.edu/~dst/DeCSS/Gallery, 18 October 2000.

Thesis Poster Competition

- Posters should be be on a zip disk ready for us to take to the printer by midday Monday November 20th.
- The value of the prizes is still to be negotiated with our sponsor. Last year the first prize was $1000 and the runner-up prize was $500.
- As a student in 415.725, you *may* be eligible to enter this competition. Expressions of interest should be sent by email to Dr Richard Lobb (richard@cs.auckland.ac.nz).

This Week

- By midnight Wednesday 18th October, send email to me containing the title and abstract of your term paper, on a single PowerPoint slide.
- Your term paper is due 4pm Thursday 19th October, in hardcopy submission to my mailbox.