System Security
CompSci 725 S2 16

First Set of Lecture Slides

Clark Thomborson

v1.01 of 2016-07-19: substitute “system security” for “software security”

Objectives

- Anyone who passes this class will be able to
  - give basic advice on system security, using standard terminology;
  - read technical literature on system security, demonstrating critical and appreciative comprehension; and
  - give an informative oral presentation on, and write knowledgeably about, an advanced topic in system security.

Assessment: 60% final exam

- To pass this examination, you must show good understanding of the required readings (approx. 300 pages)
- I'll administer a 20-minute "practice exam" (anonymous, ungraded!) in the 11th week.
  - I'll let you know how I'd mark some of your responses.
- You will be allowed two hours for your final exam.
  - Closed book exam, assessing *your* understanding of the articles you have read, and discussed, in this course.
  - My exam questions are based on our discussions… if you don't attend lectures, you won't hear our discussion.

Assessment: 25% written report

- Primary requirement: You must demonstrate your critical and appreciative understanding of
  - at least **three** professional publications relevant to software security.
  - At least **one** of your references must be a required reading for this course.
  - You must also cite and (at least briefly) discuss **any other** required class reading that is closely related to the topic of your term paper.
- Additional (form & style) requirements: see the next slide.
- I will publish your paper online, if you request this:
  - http://www.cs.auckland.ac.nz/courses/compsci725s2c/archive/termpapers
  - Your paper might be used by other scholars, see e.g. http://scholar.google.co.nz/scholar?hl=en&q=A+Taxonomy+of+Methods+for+Software+Piracy+Prevention&btnG=&as_sdt=1%2C5&as_sdtp=

Additional Requirements on Written Reports

- If you use someone else's words, you must put these in quotation marks and add a reference to your source.
  - I will report extensive plagiarism to the HoD, for possible disciplinary action.
- Use your own words, except when quoting definitions or other people's opinions.
  - Light paraphrase (i.e. changing a few words) of a declared source implies that you have a very poor understanding of the technical meaning of your source material.
  - Light paraphrase of an undeclared source is plagiarism – and it implies that you have tried to hide your plagiarism by paraphrasing. Declare your source!!
- Technical words must be spelled and used correctly.
  - You should use a spell-checker and a grammar checker (e.g. MS Word), however we will not mark you down for grammatical mistakes and spelling errors on non-technical words (if your meaning is clear).
- Your report *should* consist of eight to twelve pages of 12-point type with generous margins and 1.5 line spacing.
  - Enforcement is indirect. A longer paper takes much longer to write well. A shorter paper is unlikely to show strong critical and appreciative understanding.
- *Try to* match the style of one of the articles you read in this class.
- Reports are due at 4pm on Friday 13 October (the end of the 11th week) – so that you can have feedback before you sit your examination.

Assessment: 15% oral report

- During a lecture period, you will deliver an oral report on a technical article.
- Marking scheme:
  - 1 mark, for rehearsing your report at a tutorial the week *before* your presentation. (You must schedule this rehearsal with Andrew Colarik via Cecil – he'll tell you how to do this.)
  - 1 mark, for a title slide with your name and accurate bibliographic information on the article you're discussing in your oral report.
  - 2 marks, for your one-slide summary of the article. You may quote the topic sentence from the abstract of the article (if it has a topic sentence). Your summary must be appropriate for *your* presentation: it should mention the aspect you discuss in detail.
  - 1 mark, for delivering your report in 8 to 12 minutes.
  - Plus another 10 marks for:
    - identifying (2 marks) an aspect (e.g. a concept or a technical consideration) that is either discussed in the article, or which *should* have been at least mentioned in this article,
    - which is worthy (3 marks) of careful consideration by your classmates, and
    - which you adequately explain in one to four slides (5 marks).
- Note: the aspects selected by you, and your classmates, are examinable.
  - If you select a trivial aspect, you won't succeed in arguing that it is worthy of consideration.
  - If you select a complex technical concept, then you won't succeed in explaining it adequately.
  - Your most important task, when reading the article, is to decide "what would be a good focus for our attention the next time someone reads it?"
  - Try to persuade your classmates to read the article again, to learn more about what you have discussed!

Example of an Aspect

- In Abadi96, the authors assert (in Principle 3) that the omission of two names in Message 3 of the protocol of Example 3.1 has "dramatic consequences".
  - This article didn't adequately explain why these consequences are dramatic.
  - In my presentation, I'll explain this drama and why security professionals should learn how to avoid it.

An Aspect of Another Article

- In Birrell85, the author asserts that the use of CBC mode of DES encryption in their RPC protocol "reduces the probability of most undetected modifications to 2^{-64}."
  - The author reminds the reader that an attacker can guess a DES encryption key with probability 2^{-56}.
  - I'm confused by this: does Birrell believe that attackers will make random modifications, without even bothering to guess a key?
  - In my presentation, I'll discuss some other assertions in Birrell85 about the security of this RPC protocol, in an attempt to determine whether or not it should be considered a "secure protocol" or is merely a promising start on one.

A Temptation You May Feel

- You *might* be tempted to start reading other articles, to learn more about your "aspect" before finalising your oral presentation.
  - Resist this temptation!
  - Stay focussed on the article you're presenting!
  - As soon as you're done with your oral presentation, give in to the temptation – and you'll then be making an excellent start on your written report. We'll discuss this later…

Warning

- We will discuss vulnerabilities in widely-deployed computer systems.
- This is *not* an invitation for you to exploit these vulnerabilities!
- Instead you are expected to behave *responsibly*, e.g.
  - Don't break into computer systems that are not your own.
  - Don't attempt to subvert any security system in any other way, for example by taking over someone else's "digital identity".
  - Read & obey https://www.auckland.ac.nz/en/about/the-university/how-university-works/policy-and-administration/computing.html. (These are "soft" security controls: we will discuss some of these later in this course.)

Reading for Wednesday

- B. Lampson, "Computer Security in the Real World", *IEEE Computer 37:6,* 37-46, June 2004. DOI: 10.1109/MC.2004.17
  - Available to U of Auckland students on http://www.library.auckland.ac.nz/.
  - If you don't know how to use our University's library, see http://www.library.auckland.ac.nz/instruct/instruct.htm.

Lampson, "Computer Security…"

- "What do we want from secure computer systems?" Lampson says:
  - We want the same level of security as a "real-world system", *e.g.* the lock on the front door of our house.
  - Real-world security is just good-enough that the "bad guys" won't think the expected value of an attempted theft is worth the risk (expected cost) of punishment.
  - Better locks raise the cost of an attempted theft, and thus decrease its expected value to a "bad guy".
- Economic rationalism: We should buy a better lock only if our expected gain (= reduction in expected loss by theft) exceeds the cost of this lock.
- The cost of a lock includes its purchase, installation, periodic inspection or usage audit, key distribution and revocation, and operation (*e.g.* time to unlock and lock).

Who are “we”?

- Lampson identifies four different user populations in his threat analysis.
  - Users of internet-connected computers
    - Could be attacked by “anyone”
    - Could “infect others”
    - Could run “hostile code that comes from many different sources, often without your knowledge”
  - Laptop users
    - “Hostile physical environment”
  - “If you own content and want to sell it, you face hostile hosts”
  - Organizations trying to control access to “critical data”.

Who are “we”? (cont.)

- Consider: The users of a system rarely have administrative rights, especially in a corporate setting.
  - “What the users want” is not always the same as “what the administrator wants”.
  - “What the administrator wants” may not be the same as “what the CEO wants”.
  - “What the CEO wants” may be illegal, *i.e.* in conflict with “what the government wants”.
  - “What the customer wants” may differ from all of the above.
  - Any interested party may be unclear, or misinformed, about what they (or “we”) want!

Important Security Technologies

Do you know all of these? (If not, let's be sure to cover it in this course!)

1. Subject/object access matrix model [Lampson 1974]
2. ACLs [Saltzer 1974], [Denning 1976]
3. Information flow modelling [Myers & Liskov 1997]
4. Star property [Bell & LaPadula 1974]
5. Public-key cryptography [RSA 1978]
6. Cryptographic protocols [Abadi & Needham 1995]

Why Not Try for "Perfect Security"?

- Too complicated: can't understand all requirements; can't implement everything you understand; can't keep up with requirement changes; can't maintain.
- Security is only one of many design objectives.
  - Conflicts with features, usability?
  - Conflicts with performance?
  - Too expensive to specify, set up, maintain?
  - Difficult to justify expense, because security risks are impossible to assess accurately.
- Boaz Barak takes a contrary position, in his discussion of "fuzzy security" at http://www.math.ias.edu/~boaz/Papers/obf_informal.html.

Aspects of Secure System Design

- Specification/Policy
  - What is the system supposed to do?
- Implementation/Mechanism
  - How does it do it?
- Correctness/Assurance
  - Does it really work?

❖ Lampson takes a "computer science" viewpoint, emphasizing the technologies used in system design.

❖ The "information systems" viewpoint emphasizes policies, people, and whole-lifecycle processes.

Specification/Policy

- Secrecy (Confidentiality)
  - Unauthorized users cannot read.
- Integrity
  - Unauthorized users cannot write.
- Availability
  - Authorized users can read and write.

These are the "CIA" objectives.

  - The Unix filesystem has "x" and "d" bits, as well as "w" and "r" bits. Are "x" and "d" in the CIA?

- Accountability (Audit)
  - Administrative records of subjects ("who?") and objects ("to whom?").
  - Audit records may include actions ("did what?"), times ("when?"), authority ("who said it was ok?"), etc.

Implementation

- Code
  - "The programs that security depends on."
- Setup
  - "… all the data that controls the programs' operations: folder structure, access control lists, group memberships, user passwords or encryption keys, etc."

❖ Would you say this is a "computer science" viewpoint?

❖ What else would you include in implementation, from another viewpoint?

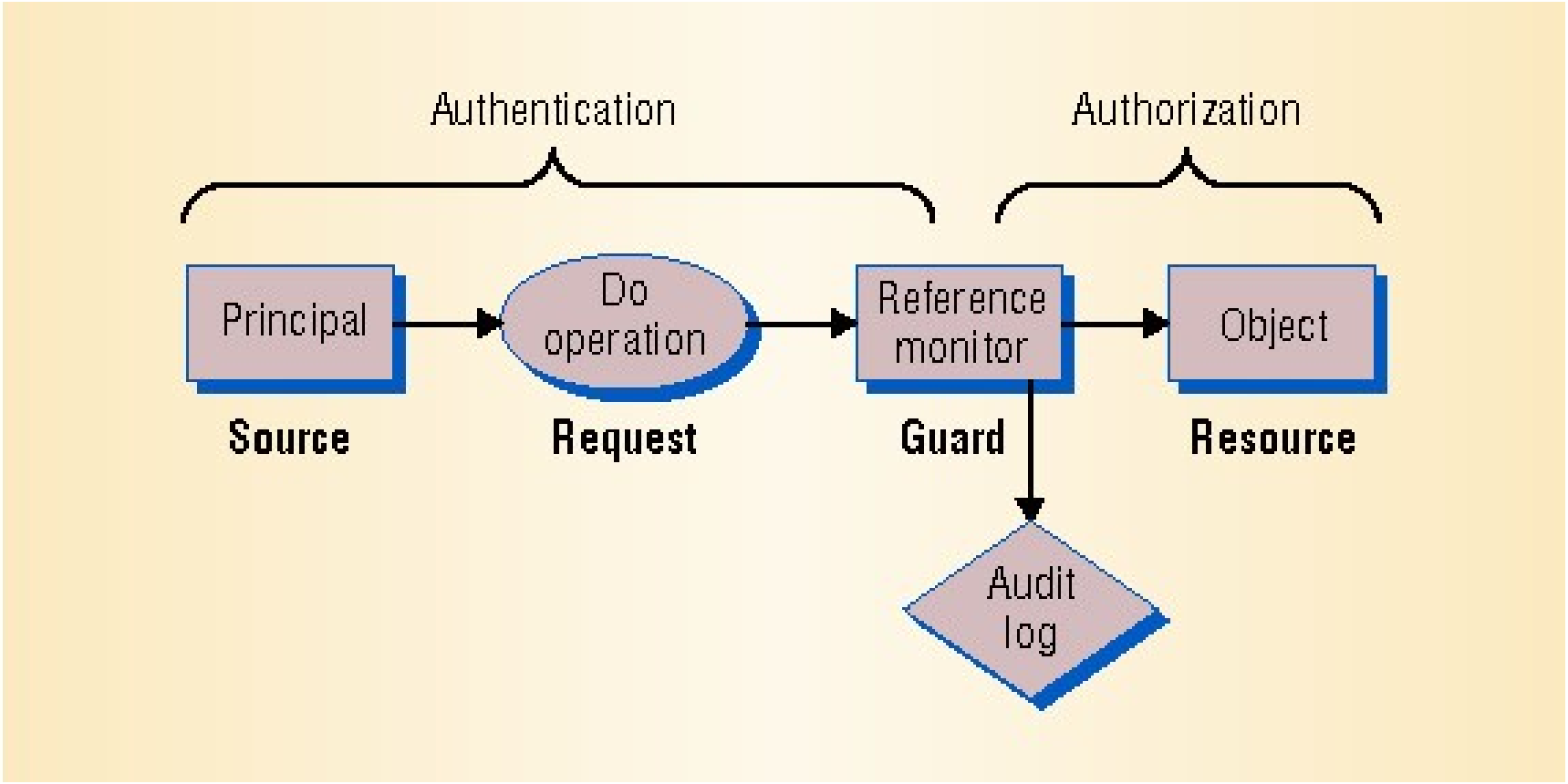
Vulnerabilities

- Programs
  - "Bad - buggy or hostile"
- Agents
  - "Bad – careless or hostile"
  - "Either programs or people, giving bad instructions to good but gullible programs"
- Agents
  - "Bad agents that tap or spoof communications"

❖ Is this a complete list? Are the distinctions clear?

❖ Can you draw a picture to illustrate these distinctions? (Subject, object, action, communication channel? Source, request, guard, resource, audit log?)

Figure 1. Access Control Model



Defensive Strategies

- Isolate: keep everybody out!
- Exclude: keep the bad guys out!
- Restrict: let the bad guys in, but keep them from doing damage! (Sandboxing.)
- Recover: Undo the damage!
- Punish: Catch the bad guys and prosecute them!

❖ Can you draw a picture to illustrate these strategies?

❖ The usual strategic taxonomy ("defense in depth") is "Prevent", "Detect", "Respond".

Information used by the Guard

- Authentication
  - Identification of the principal making the request
- Authorization
  - Policy on "who (= Principal or Subject) is allowed to do what (= Request or Action) to whom (= Object or Resource)"
- "Authentication" and "Authorization" are often confused in technical writing. Try to use them accurately!
- Many authors make a careful distinction between "identification" (e.g. a username) and "authentication" (e.g. a password).
  - Biometrics may be used either for identification (deciding who is trying to login) or for authentication (deciding whether the identification provided by the user is valid).
- Sometimes a distinction is made between the "Authorizing Subject" and the "Actor".
  - The Actor is delegated (by the Subject) to perform the Action.
- Design principle: Separate the guard from the object.
- Note: the Guard of Figure 1 doesn't check on what the Object does!
  - This security assurance (of "Object correctness") is sometimes ignored, or it may be handled by another Guard (not shown) which watches over Objects.

Information Flow Control

- Dual of Access Control Model
- "The guard decides whether information can flow to a principal."
- ❖ Can you draw a picture, like Figure 1, showing Information Flow Control?

- "Star property" (hierarchical security)
  - Principals at the center can "read everything" but "write nothing" outside the central ("top secret") domain.
  - Principals outside the center can "write everything" but "read nothing" in the central domain.

Assurance

- Lampson: "Making security work requires establishing a *trusted computing base."*
  - The TCB is the collection of hardware, software, and setup information on which a system's security depends.
- What else is required to make "security work"?

Simplifying Setup: Roles and ACLs

- Role-Based Security
  - Guard uses stereotypes when deciding whether or not to allow accesses by a "security principal".
  - Each process runs with (a subset of) the access rights of the login *x* that authorised the process to run. E.g.

    $$\text{role}(x) \in \{\text{Administrators, Users}\}$$

  - A simple role-based view of other principals p is

    $$p \in \{\text{Me, My group, The World}\}$$

- Access Control Lists
  - Guard looks for entry (*S,A,O*) in the ACL, when deciding if *S* is authorised to perform *A* on *O*.
- ACLs may become very large.
- Role-Based Security becomes difficult to design, manage and understand when there are many roles, many types of actions *A,* and many types of objects *O*.

Other Topics

- Distributed vs. Local Access Control
  - Access control is easiest on a standalone machine.
  - On distributed systems, communications between the Guard, Subject, Object and Actor must be either provably secure or trusted.
    - "Trusted" is not the same as "provably secure", for if there is no insecurity there is no need for trust.
- On pages 42-45, Lampson describes the concept of a "chain of trust".