Emily Ngo

Emn367

Ngo.emily@utexas.edu

Lecture 17

1. Yes, the BLP model is much stricter in terms of information flow and can be written as a NI policy since NI policies are more general with its information flow policies. Any MLS policy can be written as a NI policy.
2. A and B have completely different sets so neither can dominate one another, so they have no information flow with each other and there should be no arrows between the two.
3. Depending on how the policy is made, covert channels can still exist if they are not characterized in the "view" of the policy.
4. A can only be something that is lower or equal than B (in a BLP system) for subject A to interfere with B:
   A "low" ; B "low"
   A "low" ; B "high"
   A "high" ; B "high"

Lecture 18

1. NI itself is policy that can enforce mechanisms that may be anything, and the NI theorem refers to all subjects, states, and instruction sequences. While a policy can be restricted to one or two states or a single instruction sequence, NI can represent multiple workings of polices which is why it better represents a metapolicy.
2. L1, l2, l3, …, lk
3. Non-interference requires identifying all potentials channels of information which means modeling a realistic system would be highly difficult because realistic systems contains many potential channels. Also, non-interference captures deterministic systems better; however, most realistic systems are non-deterministic and random.

Lecture 19

1. Integrity in different contexts can refer to the truthfulness of a newspaper's source, or the correctness and pureness of a computer's database.
2. A freely available version is less regulated over the quality and content of the software being offered, such as including bloat ware or malware. If you purchased the software you are getting it from the makers of the software or trusted retailer and also you and have the retailer accountable for the quality of the software.
3. Separation of duty is having multiple subjects completing a function, and separation of function is not having a subject have multiple (complementary) roles in completing a function.

4. Auditing allows you to recover from bad changes by assessing or reviewing what changes are reported. This can be used to maintain the integrity of whatever system you are auditing for.
5. In a commercial setting there are many discretionary internal interactions that might not follow integrity principles which will raise concerns. Such as the first example, “Users will not write their own programs” this is a concern over Separation of Function, or the process of moving to production from development which is a concern over Auditing.
6. You made a purchase online that was 5 dollars but the website has an error while processing the transaction with your bank account number and charged you 6 dollars instead.

Lecture 20

1. A highly reliable piece of information with little sensitivity can be the Bill of Rights, because it doesn’t change and is available to almost everyone. Information that is not so highly reliable but has greater sensitivity can be a pending research paper. It’s only available to the people overseeing the research but the paper is due to change over the course of the research.
2. Expert: {Physics} dominates Student: {Physics} because an Expert is more credible than a Student and they both have equal sets.
   A Novice does not dominate an Expert, a Novice is less credible than an Expert.
   A Student is more credible than a Novice and {Art} contains the empty set so the Student does dominate the Novice.
3. High can only interfere with low. H|->L
4. Orthogonal issues means they are not related issues; they should be dealt with separately.

Lecture 21

1. Dual meaning BLP and Biba have analogous systems; they share concepts of the Simple Security Property and *-Property but the direction of information is reversed.
2. It is empty because L,{B,C} and L{A,B} cannot dominate one another because neither sets are supersets of the other. The Simple Integrity Property and Integrity *-Property requires dominance in order for a read or write to occur.
3. If the means is to protect both confidentiality then the answer is no for this case.

Lecture 22

1. The assumption is any read from a lower level can accidently corrupt the subject, so the subjects level decreases with the lowest item they’ve read.
2. No.
3. The Ring policy assumes that the subject can tell what bad information is.
4. Yes.

Lecture 23

1. No they are for separate security labels. One is a confidentiality category and the other is for integrity.

2. Downgrade allows objects to be moved from development to production, which addresses one of Lipner's concerns. In other words, downgrades allow you to change the type of an object.
3. No because in order to write, the confidentiality label for the system controller must be dominated by the confidentiality label for Development code. However SL{SP,SD} dominates SL{SD} , and both confidentiality and integrity policies must be fulfilled in order to write.
4. Weak tranquility

Lecture 24

1. To maintain the consistency among various components in the system. Mostly to maintain the integrity of a system and also verify it to transition from one consistent state to another.
2. Cash Balances, Checks
3. Office supply inventory, Printer usage
4. Certification rules confirm if the action is valid and the end result is valid, enforcement rules imposes the procedures for a system to achieve goals of certification rules.
5. A bank teller = user
   Deposits, Withdrawals, Wire, etc. = transaction procedure
   All customer bank accounts = CDI set

Lecture 25

1. It is a potential conflict of interest since both companies are Airline companies the consultant might carry proprietary information from American Airlines that United Airlines can exploit for an advantage in competition.
2. Yes, GM and Microsoft are not competitors.
3. GM, Microsoft, and all other company files excluding GM's competitors and Microsoft competitors are available for access.
4. BLP prevents information flow from high to low classes within a system; Chinese Wall policy prevents lateral information flow between classes of the same type (conflict classes).

Lecture 26

1. A subject can have multiple functions without having to assign them as individual permissions.
2. Active roles are roles currently taking place; authorized roles are roles that the subject could take on.
3. Role authorization imposes that a subject's active role must be authorized, and transaction authorization imposes if the subject can carry out the active role or not.
4. For larger organizations, if using standard access control policies for subjects that all need the same permissions you still have to set those permissions individually for each subject. Permissions are defined as read/write rather than specific actions like "create an account", so when the set of permissions allowed for a subject grows it is harder to distinct the function of that subject.

Lecture 27

1. Like before, a system can be very large and the matrix would get too big and sometimes empty for most subjects to be useful for reference.
2. Access control list, capability-based system, maintain set of rules.

Lecture 28

1. The sender and receiver must have some shared knowledge, so the receiver can interpret the answer.
2. To see if the capacity of the channel that is sending the message can handle it.
3. No communication can occur unless they have share knowledge; it is how the receiver interprets messages.
4. Sending more data just takes up bandwidth and won't be interpreted unless the receiver and sender have shared knowledge over the extra data.
5. 0 bits, the receiver already knows the answer regardless what the sender wants to send.

Lecture 29

1. N-bit binary number would take N-bits
   single decimal would take 4 bits
   two digit would take 7 bits
2. The uncertainty is the time of the attack, "dawn", which could range from dawn or dusk, any time of the day, or any time in the month… there is too much freedom in conveying when an attack can occur
3. 4 bits can represent 16 possible combinations of 1 or 0 which is sufficient for 16 different messages
4. 8 bits
5. Most scenarios deal with an unknown amount of possible messages that could be sent. Examples would be "The plant needs watering in the morning" morning is an unspecified amount of time that could be conveyed in many possible ways…

Lecture 30

1. A bit can be the numerical digit 0 or 1, or it can be bit can be a measure of information.
2. M0 000
   M1 001
   M2 010
   M3 100
   M4 011
   M5 101
   M6 110
   M7 111

3. In total to send 1000 messages it takes M10 which is 1 bit (information that is contained) * .995 * 1000 = 995 and all the other messages M0-M9; M11-M15 are the remaining (1000- 995) * 5bits (information that is contained)
4. Knowing the prior probability lets you identify which most occurring messages can be utilized to be encode more efficiently
   M0 ACSII value 49; 7 bits
   M1 ACSII value 50; 7 bits
   M3 ACSII value 51; 7 bits
   M4 ACSII value 52; 7 bits
   7bits * 1000 = 7000 bits, 7 bits per messages
5. The implication is that it is the best encoding, or the one that takes the least amount of bits per second to be interpreted.

Lecture 31

1. "2468"
2. 1 1
   2 11
   3 111
   4 1111
   5 11111
   6 111111
3. Non-uniquely decodable encodings can't differentiable between certain symbols if there is a shared prefix, so multiple WRONG decodings are possible and we don't want that
4. Lossless allows the possibility of decoding and then recoding to the original message (so information is never lost)
5. Morse code is not uniquely decodable, there is a shared prefix for E & S so if streamed endlessly without breaks we won't know if it is 3 E or 1 S

Lecture 32

1. Entropy of 8-die = -(log 1/8) = log 8 = 3
2. –(4/5 * log (4/5) + 1/5 * log (1/5)) = .722
3. It tells you the theoretically possible lower limit on encoding efficiency, which can tell you how efficient your current encoding is.

Lecture 33

1. You are increasing the variation of results
2. For 16 flips, (9/16 (probability) * 16 (total bits) * 1bit (each encoding)) = 9 bits for the first 9 flips; (3/16 (probability) * 16 (total bits) * 2 bits (each encoding)) = 6 bits for the next 3 flips; (3/16 (probability) * 16 (total bits) * 3 bits (each encoding)) = 9 bits for the next 3 flips; and the last flip is (1/16 (probability) * 16 (total bits) * 3 bits (each encoding)) = 3 bits. When you total the 16

flips you expect 27.

It's based on the probability * total number of flips * number of bits used in the encoding.

3.

Result	prob.	naive
1	6/20	000
2	6/20	001
3	3/20	011
4	3/20	101
5	1/20	110
6	1/20	111

4. $H = -((6/20)\log(6/20) + (6/20)\log(6/20) + (3/20)\log(3/20) + (3/20)\log(3/20) + (1/20)\log(1/20) + (1/20)\log(1/20)) = 2.295$

5.

Result	prob.	Code 1
1	6/20	00
2	6/20	01
3	3/20	100
4	3/20	101
5	1/20	110
6	1/20	111

6. Naïve encoding total bits out of 20 rolls= 60 bits = 3 bits per roll

   Code 1 encoding total bits out of 20 rolls= 48 bits = 2.4 bits per roll

   2.4 is closer to *h* which is 2.295