Name: Jessica Lucci
EID: jml3624
CS Login: jml3624
Email: jessicalucci14@gmail.com

Lecture 17

1. A system that complies with the BLP model does not necessarily comply with non-interference, as non-interference only knows to prevent certain subjects from interfering with one another. It doesn't by default prevent higher subjects from interfering with lower level subjects. It could be the case that a lower level subject is not allowed to interfere with a higher level subject in a non-interference policy, so a system in compliance with the BLP model would not necessarily be in compliance with the given non-interference policy.
2. Neither A nor B would be allow to interfere with each other, as neither of their labels dominates the other. Anything A does should not be visible to B and vice versa.
3. There are no covert channels within an NI policy, as NI states that, given a subject L, L's view of the system state from when the system is run normally to when the system is run with all the commands of the subjects not allowed to interfere with L are deleted, will look identical. This means that a subject within a NI policy enforced system will never see any information from a subject that it's not supposed to, as it'll appear that that information never even existed to said subject.
4. An NI policy A->B means that subject A is allowed to interfere with subject B. In a BLP system A could interfere with B if A was level "low" and B was level "high", if A was level "low" and B was level "low", and if A was level "high" and B was level "high".

Lecture 18

1. NI policies better mimic metapolicies than policies as they do not provide any concrete rules about which subjects can read/write which objects.
2. l1, l2, l3, lk
3. Proving NI for realistic systems is difficult for a variety of reasons including having to identify all potential channels of information within your "view" function (realistic systems have *many* of these channels), dealing with realistic timing channels is very difficult, and very few realistic systems are completely deterministic.

Lecture 19

1. Integrity can vary between different contexts (or sources) of the information being considered. For instance, the same piece of information coming from Wikipedia and from your biology teacher may change the meaning of that information dependent on which source, or what context, you read it from.

2. A company or individual might opt to buy commercial software rather than download a similar, free version, as the source of the code is validated and held to some sort of legal expectations (of code availability, workability, etc). A free version might be malicious, buggy, etc.
3. Separation of duty means that *several* different subjects must be involved to complete some sort of function, while separation of function means that a *single* subject cannot complete complementary roles within a process.
4. Auditing is important in dealing with integrity contexts, as it helps verify and make decisions about different information sources. If a system must have recoverability and accountability of an audit trail, its' history is essentially recorded, and decisions about that system and its' information can be derived from this history.
5. The underlying ideas that raise the integrity concerns of Lipner are ideas of commercial systems in relation to the product they provide to customers. For example, need to make sure system is always available, data is reliable, etc, in a way that maintains integrity.
6. A common scenario in which integrity would be more important than confidentiality would be in the commercial world. A company might value the integrity of their application much more so than the confidentiality, as if the application's integrity is damaged, the company is likely to lose many customers (and in turn money).

Lecture 20

1. A report of the score/highlight moments from last night's basketball game broadcasted on CNN has highly reliable information, but little sensitivity. A report on the United States government's top secret plan to invade Russia in the *National Enquirer* has high high sensitivity, but very low reliability.
2. (Expert: {Physics}) dominates (Student: {Physics}) as Expert hierarchically dominates Student, and the student's category Physics is a subcategory of the expert's categories of {Physics}. (Novice: {Physics, Art}) does not dominate (Expert: {Physics}) as novice is hierarchically lower than expert, and the expert's categories {Physics} is a sub-category of Novice's categories {Physics, Art}. Lastly, (Student: {Art}) dominates (Novice: {}), as Student is hierarchically greater than Novice, and Novice's categories {} is a sub-category of student's categories {Art}.
3. S_h -> S_l , such that h is a higher integrity level than l. Essentially, only subjects with high integrity are allowed to interfere with subjects with low integrity.
4. Confidentiality and Integrity are orthogonal issues in that they are issues independent of one another. A confidentiality policy will have nothing to do with an integrity policy and vice versa.

Lecture 21

1. Biba integrity is called the "dual" of the BLP model, as the BLP model, just like the Biba Integrity model, defines a Simple and *- Properties that define when subjects are allowed to read and write objects respectively.
2. The ACM for Subject 3 - Object 3 is empty as the neither the subject label nor the object label dominates one another (Since {A, B} is not a sub-set of {B,C} and vice versa).
3. In systems with both confidentiality and integrity systems, a subject must pass both confidentiality and integrity constraints to perform an operation on an object. So if a subject passes a confidentiality constraint but fails an integrity constraint for a particular object, that subject may not access that object.

Lecture 22

1. Biba's LWM Policy assumes that subjects are untrustworthy in that if they read a piece of lower-integrity information, their integrity must drop to match that object's integrity level. They are not capable of distinguishing "good" and "bad" information.
2. The subjects are not considered trustworthy in LWM policy, as their integrity level falls as soon as they read something of lower integrity.
3. Biba's Ring Policy assumes that subjects are capable of distinguishing "good" (high integrity) objects from "bad" (low integrity objects), so they won't be corrupted by reading a lower integrity piece of information.
4. The subjects are considered trustworthy in the Ring Policy, in the sense that they are able to filter out bad information (lower integrity objects) from good information (higher integrity objects).

Lecture 23

1. The SD and ID categories of Lipner's Integrity Matrix Model are not related, as SD is related to confidentiality, and ID is related to integrity, and confidentiality and integrity are orthogonal issues.
2. System controllers must be able to downgrade so they can move objects (code/software) from development to production. Without this ability, no objects could ever get through to production.
3. System controllers cannot modify as they don't satisfy the BLP requirement that the object's label ((SL, {SD})) dominates the subject's label ((SL, {SP, SD})).
4. Weak tranquility.

Lecture 24

1. The purpose of the four fundamental concerns of Clark And Wilson is to maintain consistency among the various components of the system state.
2. Some possible CDIs in a commercial setting could be things like bank balances, checks, etc. - things that are meant to be kept private.

3. Some possible UDIs in a commercial setting could be things like free candy, koozies, etc. that companies give away.
4. Certification rules are integrity-monitoring rules, while enforcement rules are integrity-preserving rules.
5. An example of a permission in a commercial setting would be a company manager (user), changing (TP) the salary (CDI) of an employee.

Lecture 25

1. This might potentially be a breach of contract as the two companies are directly competing companies.
2. The user would be able to access GM again, as Microsoft is not in the same conflict class as GM.
3. After a user access both GM and Microsoft, that user has access to the files of GM, Microsoft, Bank of America, Wells Fargo, and Citicorp.
4. The Chinese Wall Policy allows users to access any document that doesn't conflict with something they've already read, regardless of the document's security level.

Lecture 26

1. By associating permissions with roles instead of users, less permissions have to be administered/monitored (every person with role X has the same permissions), permissions can be more appropriate to organizations, and users can assume multiple roles.
2. Authorized are roles a user is allowed to assume, and active roles are the roles the user is currently occupying.
3. Role authorization means that a subject's active role must be an authorized role for that subject, while transaction authorization means that in order for a subject to execute a transaction, that transaction must be authorized for one of the subject's active roles.
4. Unlike RABC, Standard control policies force the subject to change identities in order to perform different roles, must track specific permissions of every individual, and have generic permissions (like "read a file" instead of "open an account").

Lecture 27

1. One wouldn't want to write out an explicit ACM for a system as the matrix would become impossibly large for any realistic system.
2. Access Control Lists store permissions with objects, Capability-Based Systems store permissions with subjects, and a set of system-defined rules will compute permissions on the fly.

Lecture 28

1. The receiver must have previous knowledge of what constitutes a "yes" or a "no" that corresponds with what the sender defines as a "yes" or a "no".

2. Quantifying information content of a message helps us to do things like create more efficient code (how efficiently can we transmit certain chunks of data) and gauge the capabilities of a system (maximum bandwidth).
3. Both the sender and receiver must have some shared knowledge of an encoding scheme, as the information the receiver gets would be virtually useless - the receiver could decode the information as they see fit, but unless that decoding matches what the sender intended, the information is useless.
4. The sender wouldn't want to send more data than the receiver needs to resolve uncertainty, as any additional information could potentially confuse the receiver, and unnecessarily clog up bandwidth.
5. No bits of information are needed to quantify the information content if the receiver knows the answer is "yes", as the receiver already knows the answer - they don't actually need any information to be sent to them.

Lecture 29

1. Each of the first three messages contain 1 byte of information each, as each message requires 4 bits to distinguish the message's ID.
2. The less things the receiver knows, the more things that must be clarified. For instance, if the receiver only needed to know if the attack was at "dawn" or "dusk", the sender would only have to send 1 bit. If the receiver needed to know if there was an attack at all, and if so what time, that would require more bits of information. So the greater the uncertainty, the greater the amount of information that must be sent to resolve that uncertainty.
3. 4 bits of information must be sent to send one of 16 messages, as it requires 4 bits to enumerate numbers 0 - 15.
4. A message from a space of 256 messages would have an information content of 2 bytes, as it takes 2 bytes to enumerate numbers 0-255.
5. In an ideal situation, a bit of information can reduce uncertainty by half. However, in realistic situations one bit of information usually resolves little (if none at all) uncertainty.

Lecture 30

1. The discrete connotation of bit refers to a binary digit, and the continuous connotation of bit refers to a quantity of information.
2. M0: 000, M1: 001, M2: 010, M3: 011, M4: 100, M5: 101, M6: 110, M7: 111
3. Since message 10 is only 1 bit, and is sent 995 times, message 10 contributes 995 bits per second. Then, since each other message is 5 bits long, and 5 of the non-10 messages will be sent, those 5 messages contribute (5*5) bits per second. Thus, overall we have 995 + (5*5) bits per second.
4. If the prior probabilities of messages are known, messages of high probability can be encoded to a very small number of bits, lowering the overall bandwidth used.

5. M0: 000001, M1: 000010, M2: 000100, M3: 001000, M4: 010000
6. If it's possible to find an optimal encoding, that means that every possible message, and the number of times each of those messages will be sent is known.

Lecture 31

1. "22"
2. 1: 0, 2: 01, 3: 010, 4: 0101, 5: 01010, 6: 010101
3. If an encoding is not uniquely decodable, multiple meanings can be gathered from the same message, and the true intent of the message cannot be known.
4. It is desirable for a coding scheme to be lossless, as there will be no translation errors between the original sequence of symbols and the transmitted, recovered sequence of symbols.
5. Morse code doesn't satisfy our encoding rules, at it fails to stream appropriately - there are breaks in the encoding between letters.

Lecture 32

1. 3
2. $-(⅕*(\log_2(⅕))+(⅘*\log_2(⅘)))$
3. Knowing the entropy of a language is important as it allows us to set a lower limit on encoding efficiency.

Lecture 33

1. Since H/T flips can be encoded as 0/1, if we performed 32 flips, we could expect to use 32 bits - one for each flip.
2. The total number of bits is 27 because there were 9 HH sequences which were encoded in one bit 0, 3 HT sequences which were encoded as two bits 10, 3 TH sequences which were encoded as 3 bit 110, and 1 TT sequence, which was encoded as 3 bit 111. This gives us 9 + (2*3) + (3 * 3) + 3 = 27.
3. 1: 000, 2: 001, 3: 010, 4: 011, 5: 100, 6: 101
4. 2.287
5. 1: 00, 2:01, 3: 100, 4: 110, 5: 101, 6: 111
6. This coding is more efficient than the naive encoding, as 1 and 2 are more likely to occur than any other number. Since we know they'll be rolled the most often, by giving them a smaller binary encoding, we save bits per second every time they're rolled.