WEEK TWO LECTURE QUESTIONS

Charu Sharma
Charu.sharma@utexas.edu
cs36739
Due: 6/19/2014

LECTURE 17

1. No, a noninterference policy can control information flow in any way and may not comply with BLP. BLP implies information flows only from low to high, while noninterference could control information flow in a different way too. A non interference policy could be specific to BLP, but it might not, since noninterference is more general than BLP.
2. A and B can both only interfere with themselves and not with each other.
3. No, any permission of information flow is explicit, and there is no transitivity in information flow. Additionally, the interferences such as references and modifications to the same attribute by functions that cause covert channels will not exist in noninterference policies.
4. B must be high level, and A must be low level.

LECTURE 18

1. Noninterference carries out information flow from low to high, mimicking the confidentiality metapolicy of BLP, and NI doesn't specify permissions or methods of enacting information flow from low to high. It is very abstract and is not system-specific or implementation-instructive.
2. I1,I2,I3,…,Ik
3. NI policy is difficult to implement fully, because realistic systems have many interferences, many of which are low level attributes. Additionally many of the interferences, such as encrypted files, are benign.

LECTURE 19

1. Integrity is important, because the subjects modifying objects should have some level of authority to do so. In terms of information dispersion, sources of information should have high authority. For instance, one should look for information in reputed newspapers such as the New York Times rather than erroneous news sources such as The Onion. In computing, you want to make sure that the subject writing to a file has the authority to do so to prevent malicious agents from writing to files.
2. A company would choose to pay for a commercial license, because the price guarantees integrity in that the creator of that software had integrity, while online, anyone could have written or modified the download, and there is no trustworthiness of the software.
3. Separation of duties makes sure that *different* subjects work together to complete a critical function, while separation of function ensures that one subject isn't completing complimentary roles in a critical process.
4. Auditing ensures that when security is breached, it can be detected, responsibilities can be appropriately assigned, and security policies can be reevaluated accordingly.
5. Commercial settings influenced Steve Lipner's integrity policy. He evaluates employee roles and departments in creating an integrity policy. Essentially, he didn't want customers writing their own software. He also didn't want programmers developing applications on production systems. Finally, a controlled mechanism has to work between the development and production spheres.
6. Integrity is more important in business relying on software for commercial practices, because the use of this software is fairly open, but its modification can have large scale negative consequences for a wide host of users.

LECTURE 20

1. Reputed newspapers are highly reliable with little sensitivity, because the information must be correct, but can also be read by nearly anyone. Suggestion boxes are something that is not so highly reliable as anyone can enter a suggestion, but has a high sensitivity, since only the right people should be able to read it to protect feelings and filter information in the box.
2. An expert in physics dominates a student in physics, because the expert has a higher level in physics than the student does, and is therefore a higher authority. A novice in physics and art does not dominate an expert in physics, because the expert has higher authority in physics than the novice does. A student in art dominates a novice in nothing, because the student has a higher level than the novice does in general, and therefore has a higher authority.
3. Don't allow information to "flow up" in integrity. Low level integrity subjects can't interfere with high level integrity objects.
4. Confidentiality and integrity are not related, though they can be handled analogously.

LECTURE 21

1. Biba Integrity is called the "dual" of the BLP model, because strictly speaking you handle integrity labels the opposite way that you handle confidentiality labels, but you do so with the same intent of keeping information flow valid. You just change the direction of arrows from Simple Security and *-Propety of BLP to create Biba Integrity.
2. They are empty because neither the subject nor object integrities dominate each other. Neither of the category sets is the subset of the other.
3. No, it must pass both BLP and Biba Integrity to gain access to the object.

LECTURE 22

1. Biba's Low Water Mark Policy assumes that a subject is only as trustworthy as the sources it reads from, since its level is the lower of its own integrity level and the object it reads from.
2. The subjects are NOT considered trustworthy under Biba's Low Water Mark Policy.
3. The Ring policy makes the assumption that the subject can read from low level objects without losing authority.
4. The Ring policy considers the subjects more trustworthy.

LECTURE 23

1. No, SL refers to system low confidentiality, while ISL refers to system low integrity. Integrity and confidentiality are orthogonal.
2. System controllers have to be able to downgrade, because they have to be able to move software objects from development to production, and thus have to be able to change labels appropriately.
3. Yes, they can.
4. Weak tranquility underscores the special downgrade permission for system controllers.

LECTURE 24

1. Authentication makes sure you know who each person is in a system. Audit makes sure we know what each person did in a system. Well-formed transaction ensures that users can only manipulate data in appropriately constrained ways. Separation o duty means that each user is associated with a valid set of programs they can run and prevent unauthorized modification, protecting integrity and consistency. These requirements aim to target the specific concerns of commercial environments.

2. Commercial software would be a CDI, since its integrity should be protected, so only authorized users can write to it. Bank balances would be CDIs, too.
3. A candy jar would be UDIs, since assets like those are open to everyone.
4. Certification rules look at a system and make sure that things are consistent at a specific time. Enforcement policies create a consistent environment within a system so that certification rules see a consistent environment.
5. An example of a permission in a commercial setting is that a customer user can write to their own files, but not to system software, so with a transaction write, a user, customer, and a set of objects, which would include personal files, but exclude system software, we have a permission.

LECTURE 25

1. The information the consultant read at American Airlines could help United Airlines learn information on American Airlines that better helps it maliciously plot against its competitor.
2. Yes, Microsoft and GM are not competitors, strictly speaking, so Microsoft and GM are not conflicting classes, and they can be moved between.
3. Only companies which are not direct competitors of the files you've accessed are available for access according to the simple security rule.
4. BLP keeps information flowing from low to high confidentiality levels, while Chinese Wall keeps information from flowing within conflict classes.

LECTURE 26

1. You can categorize users into roles, so that managing security is more feasible than handling each user alone. Instead you handle them in groups organized by the common function users carry out.
2. Authorized roles are a set of roles an individual is allowed to fill out at various times, while active roles are a set of roles an individual is filling out at a specific time.
3. Role authorization ensures that a subject's active role is an authorized role for the subject. Transaction authorization says a subject can execute a transaction only if the transaction is authorized for one of the subject's active roles.
4. RBAC is more flexible than standard access control policies, which is harder to administer, have inappropriate permissions, have stiff function roles, and have little room for transition without identity change.

LECTURE 27

1. In realistic systems, most subjects don't have any access to most objects, so it would be inefficient to store the entire ACM explicitly.
2. Storing permissions with objects is called an access control list (ACL), and creates a list of subjects with access to a particular object. Storing permissions with subjects is a capability-based system, and creates a list of objects and accesses for a particular subject. One can also maintain a set of rules to compute access permissions on the fly.

LECTURE 28

1. The Sender has to have either a "yes" or a "no", and the Receiver must know that the Sender has one of those two possibilities, but not which. The Sender wants to efficiently transfer that data to resolve the Receiver's uncertainty. The Sender will send one bit of information indicating a "yes" or a "no" with a 1 or a 0.
2. We want to quantify the information so that we can determine bandwidth or the amount of information sent in a unit of time.

3. The Sender must have a way of transmitting information, the Receiver must know what information types to expect from the Sender, and the knowledge must be shared between the Sender and Receiver for them to effectively communicate with one another.
4. The Sender wants to communicate information with the Receiver as efficiently as possible and without leaking more information than necessary.
5. 0 bits of data are necessary since the answer is already known.

LECTURE 29

1. An n-bit binary number contains n bits. A single decimal digit contains 3-4 bits. A two digit decimal number contains 7 bits.
2. It depends on the Receiver's level of uncertainty because we don't know if the Receiver knows the day of the attack, what the attack is, what precise dawn refers to, etc., and depending on that information, the amount of information in the message could vary.
3. 4 bits are necessary to get down to one message each time you split possibilities (turning it into a search problem), so 4 bits must be transmitted to send the message.
4. 8 bits are contained.
5. In realistic systems, we often don't know what information *could* be transmitted, so we can't reach the ideal binary search tree encoding.

LECTURE 30

1. A bit is either a discrete binary digit (0 or 1), but there is also a continuous entity, bit, which is a quantity of information.
2. 000, 001, 010, 011, 100, 101, 110, 111
3. 995 out of 1000 instructions will take one bit, and the other 5 instructions will take 5 bits. Therefore, 995*1+5*5 is the number of bits for 1000 instructions.
4. Instructions with high prior probability can take fewer bits, so that only instructions with lower prior probability take more bits. Most of the time, we will be able to use fewer bits, bringing us to a higher efficiency. Prior probability gives us this information since it tells us how often each message appears in an arbitrarily long sequence of messages.
5. 000, 001, 010, 011
6. We would have to find the best encoding for a particular language, using the fewest number of bits on average to transmit messages in that particular language.

LECTURE 31

1. "24268422248"
2. 0, 1, 10, 100, 101, 110
3. If the string is not uniquely decodable, then the Receiver might read the string two or more different ways, and the communication between Sender and Receiver would not be effective.
4. If it is not lossless, then the entire original sequence of symbols won't be recovered from the transition, and the message will have transformed between the Sender and the Receiver, forming ineffective communication.
5. Morse code is not streaming, because there is a break between each letter in the code, giving too much information away.

LECTURE 32

1. –(8*(1/8 log 1/8))=3
2. –((4/5 log (4/5) + (1/5 log 1/5))
3. The entropy of a language is the fewest number of bits that can be used to transmit a message. It is the ideal encoding.

LECTURE 33

1. We are now flipping 2 coins, so we multiply the likelihood of the first coin outcome by the likelihood of the second, getting HH = ¾ * ¾ = 9/16, HT = ¾ * ¼ = 3/16, TH = ¼ * ¾ = 3/16, and TT = ¼ * ¼ = 1/16.
2. If we flip the coin 32 times, we get 9*1 + 3*2 + 3*3 + 1*3 = 9+6+9+3 = 27
3. 000, 001, 010, 011, 100, 101
4. Log(20)-.6-.9log(3)
5. 1: 00, 2: 01, 3: 10, 4:110, 5: 1110, 6: 1111
6. 2*(.3)+2*(.3)+2*(.15)+3*(.15)+4*(.05)+4*(.05)=.6+.6+.3+.45+.2+.2=1.9+.45=2.35, which is 2.35 bits per message on average, which comes out better than 3 bits per message in the original, naive encoding.