**Week 1 Questions**

**Name: Ridwan Hoq**
**EID: rmh2376**
**CS Login: ridwan**
**Email: ridwanhoq@gmail.com**

**Lecture 1**

1. Security is relevant to my identity (credit card numbers, social security numbers), my well being (driving, violence).
2. Both of those things have items that I need to protect against threats that could damage them or take them away.
3. Yes, but it was “user error”. I once forgot to lock my parents’ van and a GPS system was stolen from it.
4. Not sure how likely my laptop is infected with malware. It is likely there is some piece of software that is installed that I did not explicitly install, but I am not sure what the chances are that it is malicious.
5. My laptop requires a password on login. Additionally, I’m running avast antivirus to protect against malware.
6. To some degree. However, due to the ever changing nature of malware, it is unlikely that I am even close to being 100% protected.
7. I don’t think that the risk of cyber attackers is so high that we need to worry about if the US will exist tomorrow. However, we still ought to take appropriate measures to protect our national interests.
8. It is important to learn computer security because people depend heavily on computer systems today. There are many critical tasks being entrusted to computers and we must ensure that those tasks are not being interfered with by malicious attackers.

**Lecture 2**

1. Security is hard because humans design how systems should be secured. Even if a computer executes a security policy perfectly, that security policy could still be flawed.
2. Not really. You might be able to detect all the locations in your program where “bad things” can happen, but in an entire computer system, there’s so many things that can go wrong without you knowing about it.
3. The defender has to cover all his bases when protecting his system, when an attacker only needs one exploit to be successful.
4. Morris and Chang are basically saying the same thing: computers are, by design, inherently insecure. The only way to attain perfect security is to just not use them.
5. Security can be trade off for usability/simplicity because of all the necessary hoops one must go through to ensure that the system is safe. A good example of this is two factor authentication. If a user is trying to log in to a system that requires 2 factor authentication, they must have their mobile phone nearby to login, which is somewhat annoying and makes the process more complex.

**Lecture 3**

1. Risk is the possibility that a particular threat will adversely impact an information system by exploiting a particular vulnerability.
2. Yes I agree since there is no way to ensure that a computer system can be fully secure.
3. 
    a. A risk I accept: driving with other people on the road
    b. A risk I avoid: driving drunk
    c. A risk I mitigate: getting injured while driving by using a seatbelt
    d. A risk I transfer: taking a cab
4. Annualized loss expectancy is a way to calculate an exact cost for the potential loss for a risk. Using the values for a potential loss and the likelihood of that loss occurring, the potential cost is calculated. It's not a very effective method of calculating security expenditure, because the ALE can be effectively the same, but have very different risks.
5. Factors for risk assessment take the following into account: technical risks (will the system break?), economic (will we lose money?), or even psychological (will this risk prevent taking more risks?).

**Lecture 4**

1. Slide 2 consists of goals that computer security tries to accomplish, while slide 3 contains mechanisms for achieving said goals.
2. For me, confidentiality is key. There are a couple of places where I have saved my payment information online. If that data were to be accessed by someone else, I'd probably lose money to them using my payment information.
3. Grouping and categorizing information is done to ensure the proper individuals have access to the right information. You would group information by the level of confidentiality that each piece of information requires.
4. Authorization might change over time if a certain individual requires more/less information if their job changes.
5. Reliability and security are related in the sense that a system must be available to use at any time. If an attacker manages to take down the system by using up all the resources a system has, then the system can be considered insecure. Therefore, a system must be reliable to be considered secure.
6. Authentication and non-repudiation are important in the commerce context. A user must ensure that only he can access his payment information (authentication) and the vendor must ensure that a customer can't refuse to pay for a product after purchasing it (non-repudiation).

**Lecture 5**

1. A military database system would likely have a metapolicy focused on confidentiality since there is likely to be important secrets contained within the database. But a wireless phone system would probably have a metapolicy concerned with availability since a wireless phone user needs to be able to make a call at any time.
2. A metapolicy is very broad and doesn't provide specific guidance since it is the overall security. A policy is necessary to nail down the specific and enforceable guidelines for developers and system users.
3.

    a. Faculty/staff may not use student SSNs in documents/files
    b. Documents containing SSNs must be destroyed unless necessary
    c. Documents containing SSns must be kept in secure storage
4. It is possible. For example, if a stakeholder demands that a service must require a simple registration that doesn't require email verification, it could conflict with a security policy that requires email validation.
5. The metapolicy is likely ensuring confidentiality.
6. The statement means that to implement an overall metapolicy, the developer must understand what it intends to accomplish so that they can come up with specific policies to implement the metapolicy. If the developer doesn't understand the metapolicy, then it is likely that the policies that the developer comes up will not achieve the goals of the meta policy.

**Lecture 6**

1. Military security is mostly concerned with confidentiality because much of the military's plans must be kept away from the enemy in order to not undermine military operations. Of course, integrity and availability are important as well. If the enemy were to obtain access to military plans, they could overwrite them in order to cause confusion and ineffectiveness. Additionally, if a military's communication system were to become unavailable, then coordination and planning would become impossible.
2. The major threat in the MLS scenario is that an unauthorized actor might gain access to confidential information.
3. The proviso is there to ensure that the policy is focused on one aspect of security. A different metapolicy can be enforced with other policies.
4. The labels are split into two parts. One part is a linearly ordered set of levels associated with a piece of information. The other part places the piece of information into a category that doesn't have any particular order.
5. The labels are assigned by those who actually know about the contents of the information being labeled. The system design is not concerned with the content of the information, but rather securing the information.
6.
    1. The cafeteria is serving chopped beef on toast today (Unclassified: {Personnel, Janitorial})
    2. The base softball team has a game tomorrow (Unclassified: {Personnel})
    3. Col. Smith didn't get a raise (Confidential: {Personnel})
    4. Col. Jones just got a raise (Confidential: {Personnel})
    5. The British have broken the German Enigma codes (Secret: {Crypto})
    6. The Normandy invasion is scheduled for June 6 (Top Secret: {Personnel})
7. See above
8. If a document contains information that has various levels of security levels, then it makes sense to use the highest level of information because a lower level of security should not be able to read the document. However, a higher level of security should be able to view content with a lower level security with no problem. If a document pertains to

multiple to multiple groups, then it makes sense to apply both categories since both groups will need access to the information of function effectively

**Lecture 7**

1. Labels are “affixed” to humans in the sense that a label indicates what type of information that the person has access to.
2. Labels for documents indicate the sensitivity of the information contained. Labels for individuals indicate the clearance to view certain classes of information.
3. Documents are files on the computer. Humans are users on the computer.
4. It makes sense because it minimizes risk in a secure system. If a subject gets more access to information than is needed, it is introducing an unnecessary risk into the system that that subject might compromise the system.
5. 
    a. Both clearance and sensitivity have the same group (Crypto). The clearance of the individual (Secret) has a higher level than the sensitivity of the document (Confidential).
    b. The clearance of the individual (Secret) has a lower level than the sensitivity (Top Secret).
    c. The clearance of the individual (Secret) has a much higher level than the sensitivity (Unclassified). Additionally, the document doesn’t specify a category so any category can access it.

**Lecture 8**

1. The vocab terms were introduced to reduce ambiguity. Subjects and objects can have the same labels so it important to draw the distinction between the information and the actors in these scenarios.
2. Dominates is ordered pair. Proof:
    a. Reflexive (a<=a) : Level 1 “dominates” Level 2 if Level 1 = (Secret: {Crypto}) and Level 2 = (Confidential: {Crypto}). Reflexive because both Levels have {Crypto}.
    b. Transitive (if $a \leq b$ and $b \leq c$ then $a \leq c$):Level 1 “dominates” Level 2 if Level1 = (Secret: {Nuclear}) and Level 2 = (Unclassified:{})
    c. Antisymmetric (if $a \leq b$ and $b \leq a$ then $a = b$):
3. It’s not a total ordering because it is possible that neither labels could dominate each other.
4. For Label 1 to dominate Label 2, security level of of Label 1 must be equal or higher than Label 2 and Need to Know properties of Label 2 must contain Need to Know properties of Label 1.
5. The Simple Security property says that a subject can only be granted read access if the subject dominates the object.
6. Only if implies that it is a necessary condition but not a sufficient condition. There may be other conditions that need to be fulfilled to gain access.

**Lecture 9**

1. Simple Security isn’t enough to ensure confidentiality because it doesn’t codify write access. Information can flow from a higher level to a lower level if a subject can copy the

contents of a higher level object to a lower level object if the Simple Security is the only property implemented.

2. We need constraints on write access because sensitive information can be written to a lower security level. Maybe the user itself is trusted, but the programs that are run on behalf of the user might be programmed to overwrite lower security level documents on accident.
3. Computers are not to be trusted since they follow instructions blindly without necessarily knowing whether or not its actions are correct.
4. The *-Property states that a subject attempting to gain write access to an object can only do so if the level of the subject is dominated by the level of the object.
5. A subject must have the exact same security label to both read and write access.
6. The General would have to log out of his Top Secret account and log in to his Unclassified account to communicate orders to a private that has an Unclassified security level.
7. It is a problem, but these set of rules aren't concerned with violations of integrity. Perhaps, we should enforce a restriction of only writing to the level at which they have read access.

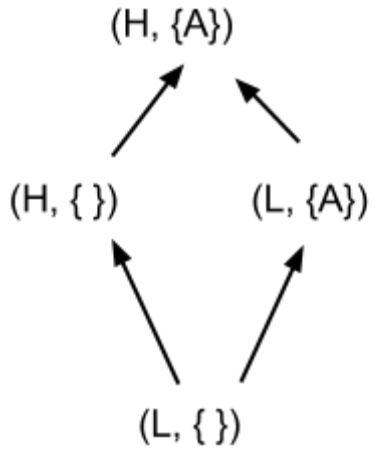
**Lecture 10**

1. Changing a subject's level up (in terms of weak tranquility) would almost always be bad because it would violate the rule of " no reading up" since the subject would now be able to read information it previously didn't have access to. However, changing a subject's level down is more ambiguous. If the subject has residual information that it had access to, then it could "write down" which would be bad. If a subject is stateless and has no residual information, then it might be ok to lower that subject's level.
2. Strong tranquility is very inflexible and overly restrictive to be used realistically. It's pretty likely that a subject or object might need to change labels, but strong tranquility would prohibit such changes.
3. Lowering the level of an object may be dangerous because its contents could be read by lower level subjects which would breach confidentiality.
4. Downgrading should be allowed from higher Level of to lower Level to keep the system secure.

**Lecture 11**

1. For a library system to have subjects have only read access to objects, but not write access, you would have to give subjects a higher security level than the objects. This follows the concept of "read down" and "no write down".
2. It would cause a whole bunch of unnecessary computations when, instead, permissions could be computed when performing an action.

**Lecture 12**




1.
2. The GLB is the lowest point in the lattice. THE ULB is the highest point in the lattice.
3. The upward flow of the information is actually the metapolicy for BLP because BLP wants to ensure confidentiality. Thus, if information can only flow upwards, that means that higher level information is being protected from lower level subjects trying to access it.

**Lecture 13**

1. In slide 1, information is permitted to flow from L to H. This reinforces BLP's metapolicy of "read down" and "write up" because the information can be transmitted upwards (writing only to a higher level) but cannot be transmitted downwards (reading only lower levels).
2. READ satisfies BLP because it requires that object O must be dominated by subject S. WRITE satisfies BLP because it requires that subject S must be dominated by object O.
3. CREATE satisfies BLP because it ensures that the object created can be written and read by subject S since the object O is created at the same level. DESTROY satisfies BLP because it ensures that the subject performing the DELETE must be dominated by object. DELETE is essentially a type of WRITE operation.
4. The high level subject must either perform (or not perform) the CREATE operation so that the low level subject can attempt to access it.
5. The DESTROY operation is there so that the set of instructions can be performed repeatedly to transmit bits through the covert channel repeatedly.
6. No they are not. Since the low level subject can always write up, it doesn't matter which subject created the object.
7. SL does the same thing in both cases to see if a bit of information has been transmitted by SH. It must do the same thing in both cases for there to be a covert channel.
8. SL does different things in both cases to transmit the bit of information. The difference in actions is what allows SL to detect that a bit has been transferred. It must do the same thing in both cases for there to be a covert channel.
9. If SL can detect varying results based on SH then confidentiality has been breached because that means that there is a flow of information downwards (which is explicitly prohibited by BLP).

**Lecture 14**

1. Two humans talking over coffee is not a covert channel because that is a normal means of communication within a system.
2. It is not a covert channel because in both branches of program execution, the SL will not be able to read what the SH wrote. Therefore, no information is being transmitted downwards.
3. The bit of information resides within the type of error message returned when SL attempts to access a high level resource.
4. The bit of information resides within the duration of the process: either the process takes no time (0) or the process takes up the entire time allocated (1).
5. The bit of information resides within the the last read sector.
6. The bit of information resides within the variable h since the variable l is dependent on the value of the variable h.
7. A termination channel would have a low bandwidth because if the program doesn't terminate, the receiver would have to wait until the program terminated to receive the next bit which could take a while.
8. The high level process would have to be able to modulate the power consumption of a device and the low level process would have to be able to detect how much power was consumed by the high level process.
9. Power channel might exist in a processor chip that measures voltage across a resistor.

**Lecture 15**

1. Covert channels, even though they usually have low bandwidth, can transfer thousands of bits per second at no performance cost on modern computer processors.
2. Because there may not be a way to eliminate the covert channel without affecting the actual function of the program.
3. One could eliminate the channel by modifying the system implementation, reduce the bandwidth by introducing noise into the channel, or just monitor the channel to see if it is being used for exploitation.
4. An example of a covert storage channel is as follows: a root user on a Unix system and a regular user can both access a certain location. The root user creates a file in that location to transmit a bit or doesn't create a file in that location to not transmit a bit. The regular user also attempts to create the file. If the file didn't exist previously, then the regular user will have created the file. Then the regular user tries to read that file. If it was created by the root user, then it will not have permission to read it, so the bit will be transferred. If it was created by the regular user, then the bit will not be transferred.
5. The root user can read in information from a secure location and either write or not write a file that can be accessed by the regular user. Depending on what the regular user observes, it can record what happens. At that point, it will have a record of the information that the root user was communicating.

**Lecture 16**

1. CREATE modifies the state of existing for a certain file. Therefore, it is an M rather than a R.

2. If an R and an M are in the same row, it indicates a potential covert channel because that indicates there is a type of resource or state that can be observed and changed by a user/process. Having a shared resource that can be observed and changed is a prerequisite for a potential covert channel.
3. If an R and an M are in the same column, it doesn't indicate that there is a potential covert channel because each command has different effects on different resources/states. The resource/state must be shared between commands for there to be a potential covert channel.
4. You'd want to create an SRMM table to identify potential covert channels.