CS 591 Bonus Homework
Kamakshi Nagar
Due 10pm, Friday June 14, 2019

Topic : Self-driving or remotely-piloted long-haul delivery truck

Security Catechism (Threat Analysis)

Give short answers (1–2 sentences, up to a full paragraph) for each of the questions below.

1. What is it that you are trying to build?

   We are trying to build a secure system for self-driving car.
   Self-driving meaning a computer program or AI will operate the car and will give instructions for directions, speed and other details on the road.
   Security system for self-driving car will have three notions of security.
   - Confidentiality – Secret should be secrets, no one who is not authorized should be able to access
   - Integrity – adversary should not be able to modify the system
   - Availability – It should be available all the time

2. What does it do? (That is, what is it supposed to do?)

   The security system is supposed to verify the instruction for Authenticity, Authorize and Audit. Make sure that the source, destination and direction is valid and comes from authorized entity.
   - Authentication – identify authorized user
   - Authorization – resources are only accessible to authorized program or users
   - Audit – keep track of what happened and who did what?

3. How does it do that?

   - Authenticity – have stronger login management system(hard and slow encryption - decryption techniques), protect against spoofing i.e. adversary(hacker, passerby or teenager) is not pretending to be an authorized user, check signature, voice recognition
   - Authorization – have an access control mechanism for authorized users and programs with their resources thus total isolation of resources (such as memory, files etc.).
   - Audit – make sure no harmful instructions were given ex. User didn't tell the car to drive into a wall

4. If you designed your system naively, without security in mind, what are some assumptions that it might make? (List 3–4 major assumptions that could be dangerous.)

   1) All instructions are coming from trusted entity. For example, if there radio plays a song that has name of two cities , the car might start driving in that direction.
   2) There is no back-door entry meaning there are no vulnerability in the code. It does not have any hardware or software bugs.

3) The keys(physical or software), passwords will not be shared/not get stolen by/from trusted user.
4) All instructions will be performed independently meaning total isolation of programs and resources.
5) System will have infinite memory, and will be available all the time. For example, information of pedestrian, bus , bicycles etc. will never overwhelm the system.

5. What happens if each of your assumptions above is violated?

Consequences of some of the assumptions can be quite serious.
If system runs out of memory leading entire system to crash, it can be dangerous and might kill people.
Instruction isolation should be handled properly. When you ask to turn off the car and open the doors, it should finish the first task before finishing the second task otherwise it would not be safe.
Unauthorized user may steal the car or login credentials.
Attacker can install the virus or worms causing undesired operations.

6. How could an attacker violate each of the assumptions? Then, what would the attacker be able to do?

We have studied lot of attacks in the class such as Format string attack , Buffer overflow, code injection, denial of service attacks, password hacking, identity theft, mimicry attack, length extension attack, dictionary attack, etc. If the system is designed without keeping the security in mind an attacker can perform any of the above attack and will break the system.

An unauthorized attacker can leak confidential information such as passwords, insurance information, trip details of last user. He/She can inject malicious code (viruses or worms) which might propagate to other systems and can affect them as well. Denial of service attack will make system unavailable. Buffer overflows can also lead to confidential data leak like destination addresses(example school, office address, preferred routes etc.).

- Mitigations

1 Finally, for each of your problem scenarios identified in the previous section, give one or more ways you could improve your design to protect against attacks.

To protect against malware we can install Antivirus, Firewalls and Security Patches.
For Identity and access management we can have secure hash, HMAC, XTS.
We can also have confidentiality and integrity policies like BIBA, BLP, Access Control list, capability system.
Process memory isolation can be provided by Address space randomization or installing a patch like KAISER to avoid users reading kernel memory.