CSE 565 A Computer Security (Fall 2025)

Assignment 2 (Total Marks: 100)

Name: Sri Charan Reddy Teegala
Email: teegala@buffalo.edu
UBID: teegala
UB Number: 50681752

Academic Integrity Statement:
I, Sri Charan Reddy Teegala, have read and understood the course academic integrity policy.
(Your assignment will not be graded without filling your name in the above AI statement)

1. [10 points] How to authenticate a human to a machine?

Authenticating people is very different when compared to machines and programs because of the difference in capabilities. The following approaches are used for authenticating humans:

- Something you know:
  - Most common kind of authentication
  - Passwords are used to access different systems in our daily life.
  - People always tend to choose passwords that are easy which means it is easy to guess i.e., passwords are one of the biggest practical problems faced.
  - We can use Length, Character set, randomness as factors to decide the strength of a password.
  - Passwords are hashed to store them in system's databases using a key which secures them from attacks.
  - Examples: Passwords, PINs, Security questions.

- Something you have:
  - Authentication based on something people have with them physically.
  - Removes the problem of forgetting the password.
  - But whenever we need to authenticate, we need to carry an object with us.
  - Usually, a PIN is added to ensure two factor authentication, one more layer for attacker to deal with.
  - Examples: Smart cards, Mobile phones, hardware tokens, access cards.

- Something you are:
  - Authentication based on something that is unique and intrinsic to the person.
  - Reliable compared to card-based authentication as they are difficult to lose.
  - But the biometric sensors are expensive and not very accurate compared to password based.
  - Easy to hijack the system connected to the biometric
  - Examples: Retinal Scan, Fingerprint reader, voice print

In summary, the combination of something you have and something you know is better way to go about authenticating users. Passwords will be in use for a long time though.

2. [20 points] This question explores the strength of different text-based password options. We constrain our discussion to 8-character passwords. Compute the number of possible passwords in each category below

   i. [5 points] Passwords composed of any combination of the 96 printable ASCII characters.
   No. of characters = 8
   There are 96 combinations per character i.e.,
   The number of possible passwords will be 96^8

   ii. [5 points] Passwords composed of lowercase letters only.
   No. of characters = 8
   There are 26 (lower case letters) combinations per character i.e.,
   The number of possible passwords will be 26^8

   iii. [10 points] Passwords are composed of lowercase or uppercase characters, where at least one character has to be a lowercase letter and at least one character has to be an uppercase letter.
   No. of characters = 8
   There are 52 (lower case + upper case letters) combinations excluding the ones with all lower case or all upper case.
   Total number of passwords will be 52^8 in which there will be 26^8 passwords having all uppercase and 26^8 having all lowercase. Therefore, using exclusion principle:
   the number of possible passwords will be $52^8 – 2 * 26^8$

3. [10 points] What is the dictionary attack? Why can we use salt to make the dictionary attack more difficult to launch?

   Dictionary Attack:
   - Systems store passwords by hashing them using a hash function i.e., storing y = h(password).
   - User can verify entered password by hashing, even if the attacker obtains password file, he should figure out the hash method for obtaining the passwords.
   - In this attack, Attacker pre-computes h(x) for all the x in a dictionary of common passwords.
   - If attacker gets access to password file containing hashes passwords, he only needs to compare hashes to precomputed dictionary.

   Using salt for prevention:
   - For every given password, we choose a random salt s to compute y by hashing both password and s together.
   - Salt s is not a secret key because it is stored in the password file itself to make the verification easier.
   - So, the attacker must recompute the dictionary with the salt (which is random every time) for each user i.e., it will be a lot harder compared to hashing without salt.

   $$y = h(\text{password}, s)$$

4. [10 points] What are the 2-factor Authentication and the single sign-on?

   Two-Factor Authentication:
   Security mechanism that requires users to provides two different types of verification factors to prove one's identity before accessing an account or system. It adds an extra layer of security.
   Examples:
   - ATM: Card and Pin.
   - Credit card: Card and signature.
   - Smart Card with PIN

   Single Sign-on:
   Authentication process that allows user to log in once and gain access to multiple related applications/services without logging in again for each one. It improves user convenience and centralizes authentication.
   Examples:
   - UB SSO for services like UB Global, UB Learns, Student HUB.
   - Google Ecosystem

5. [10 points] What are the differences between ACLs and Capabilities?

   Access Control List:
   - Created by splitting the access matrix column-wise.
   - ACL is defined object-wise (resources).
   - Lists various subjects along with the rights of an object.
   - Each object has a security attribute that identifies its access control list.
   - Protection is data oriented.
   - Convenient to change rights to a resource.

   Capabilities List:
   - Created by splitting the access matrix row-wise.
   - Capability List is subject wise (users, processes, and procedures.
   - Lists various objects along with rights permitted on them for a subject.
   - Each subject has a list of rights that it has on various objects.
   - Protection is user oriented.
   - Easy to add or delete users but difficult to implement.

6. [10 points] Why do we need mandatory access control (MAC)?
   - In mandatory access control, users are granted privileges which they cannot control or change.
   - Adding MAC to Operation systems is important to deal with host compromise.
   - Network based attacks target the hosts which is the main cause of many serious security problems.
   - Therefore, by strictly defining rules for every user ensure that there is no data leakage.
   - A system-wide security policy is enforced that restricts access rights of the users which cannot be modified by themselves.
   - Widely used for military applications.

7. [20 points] This question is about RBAC. Suppose there is a medical office with two divisions: surgery and radiology that treat separate sets of patients. Everyone who works at the office is entitled to have access to some basic resources. There are also nurses who are given privileges specific to their jobs, doctors (a doctor is either a surgeon or a radiologist, but not both), and there are administrators with access to financial and other information that is not necessarily accessible to medical personnel.

   Design an RBAC system for the office. Your specification needs to include (i) the set of roles in the system and the role hierarchy if a hierarchy is used, (ii) what privileges are associated with each role (provide text description, it doesn't need to be specific), (iii) if an individual can be assigned more than a single role, specify when this is the case and what roles are assigned (otherwise, we assume each employee is assigned a single role according to their job description), (iv) optionally, anything else you think is relevant to the RBAC design (e.g., specifying constraints). Note that the medical office specification above may not be complete with respect to the exact privileges and roles and as a system designer, you have the flexibility to specify it the way you think it should be set up. Explain your design choices and explicitly state any assumptions you make.

   Given a scenario to design an RBAC for a medical office consisting of two divisions: surgery and radiology that treat separate set of patients.

   i. Roles and Role Hierarchy
      - Employee: Baseline role assigned to everyone in the office.
      - Nurse: Access to medical records and nursing functions.
      - Doctor (Abstract role): represents medical professionals.
        - Surgeon: Doctor who performs surgeries
        - Radiologist: Doctor who interprets imaging results.
      - Administrator: Manages Financial and HR data.

      Role hierarchy is established so that higher roles inherit permissions from lower ones. In this case, medical staff and administrator inherits employee role and doctor role is further extended to Surgeon and Radiologist ensuring each doc is specialized in only one area.

   ii. Privileges assigned to each role
      - Employee: Access to general office systems like mail, announcements.
      - Nurse: View and Update patient records within their division and handle patient schedules.
      - Doctor: Can diagnose patients, add prescriptions and access division specific medical records.
      - Surgeon: Has access to surgical scheduling, operation reports and room data.
      - Radiologist: Can view, upload image data such as MRIs
      - Administrator: Manages billing, HR records, financial reports.

   iii. Role assignment rules.
      - In this system, each employee is assigned to a single primary role based on their job title. All roles inherit the Employee role so that basically

means everyone has basic privileges. A doctor (abstract) can be assigned to either surgeon or radiologist role.

iv. Design Rationale and Assumptions.

- Roles such as admin and medical staff (nurses and doctors) are mutually exclusive to prevent conflicts or unauthorized access to data.
- The clear separation between administrative and medical roles will ensure confidentiality and prevents misuse of information.
- Each role is assigned only the permissions necessary for performing its duties minimizing risk.
- Hierarchy helps to maintain operational efficiency.

8. [10 points] What is the SQL injection attack? Give an example to explain such an attack.

SQL Injection Attack:
Injection of malicious SQL code into application that allows the attacker to view or modify a database. SQLi can have significant negative impact on an organisation because it would expose all the private information about its customers. The impacts of a SQLi can be:

- Exposes sensitive company data.
- User privacy and Data integrity is compromised.
- Attacker gains administrative/general access to your system.

Example:
Let's say there is a web application that authenticates users as follows:

```
$input_uname = $_GET[‘username’];
$input_pwd = $_GET[‘Password’];

$sql = "SELECT * FROM credential WHERE name= ‘$input_uname’ and
Password=‘$input_pwd’ ";
$result = $conn-> query($sql);
```

Attacker can enter some snippet like ***’ OR ‘1’=‘1*** as username leaving the password empty then the *WHERE* condition in SQL always return true i.e., the user will always be authenticated and given access to the data.
Therefore, attacker could exploit similar injections to extract sensitive tables and modify data or escalate access.

References:

https://www.cs.cornell.edu/courses/cs513/2005fa/NNLauthPeople.html

Week-4_Class-1-2_Authentication.pptx

Week-5_Class-1_Authorization_I.pptx

Week-5_Class-2_Authorization_II.pptx

http://geeksforgeeks.org/operating-systems/difference-between-access-control-list-and-capability-list/

https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/sql-injection-attack/

Week-6_Class-1_Database_Security.pptx