ECE458/750: Computer Security
Assignment 1

Prepared by Tanmayi Jandhyala*

June 14, 2025

Problem 1. *Force a website logout by tampering with the authentication cookie (20 points)*

Like most websites, the University of Waterloo uses cookies to authenticate you on each page you visit. When you first log in they set a cookie and that cookie is then sent along with each request for a new page. Go to a University page that requires a login to view it and use your browser's developer tools to interact with the cookies. The assignment was tested using `https://outline.uwaterloo.ca` but any University page requiring a login should work. Answer the following questions.

a) What is the name of the cookie that stores login information?

b) Describe two ways you can tamper with the cookie and cause the website to log you out. The two ways do not need to be fancy or complicated, but they do need to be different.

c) Advertisers use cookies to "track" usage habits across multiple website visits. The tracking works very similar to how the University uses cookies to manage logins. Try opting out of trackign and then examining the opt-out cookie that was set. How is the opt-out cookie similar and different from your answers to question 1.b?

d) In the lectures on Access Control we discussed *capabilities* as a way to manage access rights. How is a cookie that is used to maintain login information similar and different from a *capability*?

Problem 2. *Authentication (15 points)*

a) Explain how 2-factor authentication increases the security of password-based authentication.

b) Entering a username and password frequently can be annoying for users. Assuming that the user has logged in using a username/password in the last few days, suggest three different ways in which users can be authenticated by the system without requiring them to login again. Justify your answer by explaining why each suggested approach securely satisfies authentication.

c) The BBC wants to create an authentication mechanism that works well for children who are too young to create or remember passwords. It would be expensive to send every child a physical token like a YubiKey, and most young children do not have cell phones. For privacy reasons they also do not want to record biometrics about the children. To solve the problem, they decide to ask each child to draw a picture on a piece of paper as part of the account creation process, the child then shows the picture to the camera which records it. To login in the future, the child only needs to show the camera the same drawing.

   i) What kind of authentication is the above described approach using drawings? (Something you ....)

   ii) Is the above approach better or worse than passwords for the intended audience? Justify your answer.

*Based on an earlier assignment by Kami Vaniea and Cameron Hadfield

iii) What might a backup authentication scheme look like for this approach?

Problem 3. *Access Control (15 points)*

a) Describe a situation were the Bell-LaPadula Information Control Model is more appropriate than Biba model.

b) Describe a situation were the Biba Information Control Model is more appropriate than Bell-LaPadula model.

c) A new app wants to help people create, modify, and share images such that the image authors can see all the modifications. So if a user Alice creates an image, shares it with Bob, and then Bob modifies the image, Alice can see that modification. If Bob then shares his modification with Charlie who again modifies the image, then Alice, Bob, and Charlie can all see Charlie's modifications. In other words, each image version has a list of authors associated with it and every author has read rights to the image version, but not necessarily modification rights. New authors also cannot see older versions, just newer versions. Images can only be modified if they are explicitly shared with someone.

What would be the most appropriate access control approach to use in the above system and why? Think about issues such as: number of files the system has to manage, defaults, permission delegation, and density of a theoretical access control matrix.

Problem 4. *Cryptography. (25 points)*

(a) Consider the below advice in the context of using encryption to protect "data in motion" like Internet network connections, as well as "data at rest" such as database storage.

(i) Why do cryptologists recommend changing these encryption keys from time to time?

(ii) Should old keys be completely discarded? Explain your answer.

(b) *Scenario:* Alice wants to send an encrypted message containing a user ID (`user42`) to Bob over a public channel. The ciphertext is visible to anyone observing the communication.

You are allowed to use online tools, but you are responsible for verifying their accuracy.

(i) Encrypt the userID using a Caesar cipher with key = 3 (a → d). What is the resulting ciphertext?

(ii) Encrypt using Vigenère cipher with key = `SECRET`. Try encrypting a second message (e.g., `user43`) with the same key. What do you observe in terms of how the ciphertext changes when part of the plaintext is changed?

(iii) Now encrypt the same message using AES in ECB mode with any bit key. (You can use a tool like CyberChef. Search for 'AES Encrypt' and set to ECB mode. You may use an online tool such as random.org to generate random bytes for the key.) Using a fixed key length like 128 bits does not make AES insecure and AES-128 is still considered secure against known practical attacks today. Its 128-bit key space means that brute-forcing would take 2^{128} operations, which is infeasible in simple computational environments.

Provide the key you selected and the encrypted message.

(iv) State two situations in which using a shorter key length, such as a 128-bit key length, could be security problematic compared to usin ga longer key length. (*Hint: Think about different threat models, different capabilities of attackers, and the range of security needs of system designers.*)

(v) Encrypt the following three strings using AES in ECB mode and record the ciphertext lengths (in bytes or hex digits). Use a tool like CyberChef again and ensure the mode is set to AES-ECB. Use a consistent key.

- `this_is_user42`
- `this_is_user42_and_friends`
- `this_is_user42_and_many_more_friends`

What do you observe about the ciphertext hex characters?

(c) Alice and Bob want to communicate using public/private key encryption. They have already (securely) swapped keys. For the following story state what word or words goes in each of the numbered blanks:

Alice first _(1)_ using _(2)_ key. She then _(3)_ using _(4)_ key. She then takes the message and sends it to Bob using a normal email. Bob gets the message and first he _(5)_ using _(6)_ key. He then _(7)_ using _(8)_ key. Bob now has the plaintext message and high certainty that it came from Alice.

(d) In the above story Alice had to make a choice about which order to sign and encrypt the message. Why is it important that she do signatures and encryptions in the order you specified above?

Problem 5. *Introduction to password cracking on Linux (25 points)*

In this problem, you will learn a bit about how Linux stores passwords for authentication purposes. For all the questions in this problem, we will be referring to the `root` password from the provided a1-shadowfile.txt on the course website:

(a) Find the line of a1-shadowfile.txt that is associated with the root user. Using the information in that line, answer the following:

- What algorithm was used
- What salt was used
- What is the user's hashed password

(b) Why does Linux use a salt when storing passwords? Think about the definition of security, how does using a salt make the system more secure?

(c) What is root's plaintext password? What method did you use to find it? *HINT: For simplicity, it is one of the top 10 passwords given on the slides.*

1 Unmarked Extra Questions

All questions in this section are valuable for a deeper understanding of the cryptographic ideas used in security, but your answers will not be marked for points. You are encouraged to complete these exercises for extra learning opportunities.

Problem 6. *MD5 Hashing and its Problems*

MD5 is a hash function widely used in cryptographic applications and was found to be vulnerable to computationally feasible attacks wherein an adversary could break MD5's property of `collision resistance`, which is a necessary property of cryptographic hash functions.

To complete this segment, students should follow the SEED lab on MD5 Collisions: https://seedsecuritylabs.org/Labs_20.04/Crypto/Crypto_MD5_Collision/.

Problem 7. *Password Cracking cont'd*

`John the Ripper` is a well-known password-cracking tool[1], commonly used for cracking Linux account passwords stored in the shadow file.

We provided a shadow file to run John the Ripper against, in `shadowfile.txt`.

Problem 8. *Symmetric-Key Encryption*

This section asks you to perform several activities with basic encryption. We recommend that you complete the Secret-Key Encryption SEED lab Tasks 1-5 which will help you understand how to answer the questions in this section.
SEED lab: https://seedsecuritylabs.org/Labs_20.04/Crypto/Crypto_Encryption/
Encrypted file on Learn: a1-ciphertext.txt

1. Perform Frequency Analysis on the file provided in the assignment folder on Learn: `a1-ciphertext.txt` and answer the following questions:
   (a) What are the 5 highest frequency 1-grams, 2-grams and 3-grams of the text?
   (b) Based on your analysis, what is the key used in the substitution?
   (c) The original text was taken from the web. What website was it most likely copied from, provide the URL.
2. Please define the key differences between ECB mode and CBC mode encryption. What are the performance implications of this difference?
3. Using AES-128, and encrypting a 1000-byte file, how many bytes can be recovered if the 100th byte is corrupted under the following modes:
   (a) ECB
   (b) CBC
   (c) CTR

[1] "openwall/john." Openwall, May 14, 2024. Accessed: May 14, 2024. [Online]. Available: https://github.com/openwall/john