Assignment 2

Name: **Tanmayi Jandhyala** Waterloo ID: **20877641**

WhatIAM id: tjandhya@uwaterloo.ca

Answer 1: Access Control

1.

(i) Alice Zander's abilities depend on the Clearances they have. Let us denote that clearance as C(s) and the objects that Alice wants to access as C(o).

For the file 'flag.txt', the sensitivity level and components being (Secret, { Administration, Development}), C(s) $\geq_{dom}$ C(o). So, going by the 'no read-up' of ss-property, Alice would be able to access 'flag.txt' for performing a read action.

For writing to report.txt, it is required that the objects of report.txt should have a greater <dominance of clearance> as compared to the clearance that Alice has. But in the scenario that I am faced with, Alice/C(s) has more security clearance than the file 'report.txt' as a whole does: C(o) $\leq_{dom}$ C(s) This would mean that if write access is provided to Alice for report.txt, they can post confidential information about, say, the Administration compartment to Marking and someone else with clearance to it at Top Secret level could read that information. This is why, as per the *-property, Alice cannot write to 'report.txt'.

The answer is (a), Alice can only read to flag.txt.

(ii) Bob Yves' accessing abilities would depend on them having a security clearance that is greater than or equal to 'flag.txt' in order to read the file. Since flag.txt is in the same sensitivity level that Bob has access to, but not all the compartments match, Bob can still not read the file. So (a) would be untrue.

In order for Bob to be able to write to 'report.txt', the object which is the file must have a clearance that is greater than that of Bob's. This is true for the given case, so Bob can append or write to 'report.txt'. This is essential in considering whether an attacker can target Bob's system to gain access to write to upper level files, as the model does not validate integrity property.

The answer is hence only (b), Bob can only write to 'report.txt'.

(iii) Carol Xavier's security clearance matches with that of 'flag.txt', which makes the ss-property of C(s) $\geq_{dom}$ C(o) render true. This means that Carol can read to the file flag.txt.

For Carol to be able to write to ‘report.txt’, they have to have their clearance lower than or equal to that of the object’s. Since this is also true, Carol can write to the file report.txt as well.

The answer is (c), Carol can read flag.txt and write to report.txt.

This could make a potential attacker target Carol as they seem to have both permissions necessary to make an attack.

(iv) David Wheeler has an unclassified sensitivity level, which means their access rights can be compartmentalized but there are no rules to prevent such a right from disseminating from its assigned compartment. It can be assumed that such a role is at the lowest level in a lattice and this means that David can obviously not read ‘flag.txt’. However, being unclassified, David does not hold any information that is confidential. This would mean that by the rules of the Bell La Padula model, they can write to the ‘report.txt’ file any information from and to the (Accounting, Marketing} compartments.

The answer is (a), David can read flag.txt file.

(v) Since Eve has a sensitivity level of ‘Classified’, they would not be able to read ‘flag.txt’, which holds a sensitivity level of ‘Secret’. It is irrespective of whether they share the same compartments with the file object, the security model does not allow Eve to read the file, going by the ss-property.

In order for them to write to ‘report.txt’, they would need to have compartments that match with the file ‘report.txt’. Since that is true for this scenario, and both of them have {Administration, Development, Support} roles, it is possible for Eve to write to ‘report.txt’ in these compartments.

The answer is (b), Eve can write to report.txt.

(vi) Frank has the sensitivity level of Unclassified with access to no compartments. There is not a chance they can read anything in the flag.txt file, and even though they can perform a write operation because they are at a lower sensitivity level, they do not have access ti any compartments from which they can obtain information to write.

This renders Frank to (d), perform neither read or write actions to any of the file objects.

2.

The original integrity level of Carol Xavier is (Medium Integrity, { Administration, Development }).

(a). Read is performed from File 1, of High Integrity. Since integrity of the file is greater, Carol can perform read to this file.

(i) This triggers for the integrity of the subject (Carol) to **remain the same**, as they are already at a lower integrity level than the file. Read can be performed from both compartments {Administration, Development}, as subject and object both have access to it.

(ii) The integrity level of the object, File 1 is changed from High Integrity to Medium Integrity in order for the read action to be performed.

At this point: I(s) and I(o) both are of Medium.

(b) For Write to be performed to File 2,

(i) **Integrity** of the subject, Carol, **remains unchanged** as they already are on the same level as file 2.

(ii) Integrity of the object, File 2 also **remains unchanged**. However, the write action can happen only to the {Administration} compartment, keeping the original nature of file 2 intact.

At this point again, both subject and object have the same level of Integrity- Medium.

(c) In order for Carol to write to File 3,

(i) Writing up to File 3 does not render a change in the integrity level of Carol's system- it **remains as Medium Integrity.**

(ii) The sensitivity level for file 3 changes from High Integrity to Medium Integrity, {Administration, Development})

(d) For reading File 4 which has a lower level of integrity,

(i) Carol's level of integrity **changes from Medium to Low Integrity**, and they can read only from the {Development} compartment.

(ii) The integrity level of File 4 **remains the same**: Low.

(e) For writing to File 2

(i) Carol's system's integrity **remains unchanged**, because they are already at the lower level.

(ii) File 2's integrity changes f**rom Medium to Low** to match with the subject's integrity.

(f) For reading File 5, the level of integrity of Carol has to be lowered.

(i) Carol's level of integrity **changes** to (Untrusted, ϕ) to match with the File's.

(ii) File 5's remains unchanged.

Answer 2:

1. Password Policy rules and their possible problems:

Rule 1: While it is important to let users know of the minimum and maximum length of passwords so that they do not give too short passwords, keeping a cap on the maximum length that a password should be is not in best security practice. It is recommended to have a maximum password length of 64 characters [1]. This is because longer passwords, or essentially, passphrases, have the potential to make brute-force attacks harder to implement, hence improving on security. This, of course, comes with the assumption that such passphrases are easier to remember, and not too generic.

Rule 2: This rule is fairly good to have, because it pushes the user creating the password to include several other characters that make it more complex. Expectedly, "complex" passwords are harder to carry out simple dictionary attacks against. However, such passwords are harder to remember, so users can end up using the same password across multiple platforms.

Rule 3: It is actually proven [1] that asking users/employees to change passwords frequently can result in them following low security practices like changing up just one number or just another character. This does not serve the purpose that this rule intends to have, and it is recommended [1] that strong passwords changed annually will suffice the purpose of maintaining security.

2. Bcrypt utilizes Blowfish, which is a symmetric-key block cipher, which produces a 32-bit output. This is said to be particularly fast, but it slows down when utilized in password hashing (bcrypt). This is because, when generating new keys, it requires up to 4 kilobytes of preprocessing text, which it iterates variable times, making the process extremely slow. Adding more iterations can make it slower. It is beneficial because it would take an attacker a long time to carry out a brute-force/dictionary attack (where the password string is pre-existing) due to the process of changing the key requiring more computational effort. All of this makes bcrypt a good choice to hash passwords for CHOWN.

3. (i) The string appears to be generated by a cryptographic hash, like the MD5.
(ii) There is actually no way to "reverse" or "decrypt" a cryptographic hash by definition, but on the internet, there is a dictionary of the most frequently used passwords that can give results, and fortunately the question had one such highly secure password's hashed output: password123. Since hash functions are not really reversible, companies with the intent to "safely" store passwords, have made repositories of commonly used passwords like "password", "qwerty123", etc. in the form of their hashes, called "rainbow tables". The presence of such datasets is how an MD5 hash was searchable on the Internet.
(iii) For obvious reasons, the MD5 is not a great hash function at all. Although it is true that hash functions cannot technically be reversed, due to the available dictionary for MD5 online, carrying a brute force attack is easier. Plus, MD5 is too fast in this regard, a strong hash should take an attacker more time. Also because these passwords are not salted before they are hashed, the rainbow table exists, which again makes this function insecure.

4. SMS- based 2FA can have the below issues:

- Although it can be argued as being user-friendly, SMS messages are still in plaintext and not encrypted. They can be retrieved by third-party apps simply by querying.
- Because they are in plain text, they are also susceptible to social engineering attacks. Unauthorized attackers can act as parties that offer the company's users

of certain services that they know that they are subscribed to, and can be scammed for money or data by orchestrating social engineering attacks like email or telephone scams.

- If a malicious user can obtain access to a victim user's mobile phone, they can gain access to a scary amount of sensitive information that can be authorized by 2FA SMS. For example, certain payment gateways use APIs to autofill passcodes that appear on a mobile phone.
- Further, in case the SIM card/number is changed, logging in to the service that requires SMS-based 2FA gets difficult and might involve a process to obtain authentication with the help of a third party.

Answer 3: Firewall Rules for CHOWN

The below ways is how I would have the firewall at CHOWN:

- Denial is set by default

- Allowing HTTP and HTTPS connections for all systems in CHOWN

ALLOW all => 129.34.156.0/24 FROM PORT 443 TO all BY TCP

ALLOW 129.34.156.0/24 => all FROM PORT all TO 443 BY TCP

ALLOW all => 129.34.156.0/24 FROM PORT 80 TO all BY TCP

ALLOW 129.34.156.0/24 => all FROM PORT all TO 80 BY TCP

- Allowing for the Matrix server to be accessible from anywhere (for HTTP and HTTPS):

ALLOW all => 29.34.156.48 FROM PORT 80 TO all BY TCP

ALLOW 29.34.156.48 => all FROM PORT all TO 80 BY TCP

ALLOW all => 29.34.156.48 FROM PORT 443 TO all BY TCP

ALLOW 29.34.156.48 => all FROM PORT all TO 443 BY TCP

- Having CHOWN's Matrix communicate with Root's Matrix server:

ALLOW 29.34.156.48 => 243.82.77.124 FROM PORT all TO 8448 BY TCP

ALLOW 243.82.77.124 => 29.34.156.48 FROM PORT 8448 TO all BY TCP

ALLOW 243.82.77.124 => 29.34.156.48 FROM PORT all TO 8448 BY TCP

ALLOW 29.34.156.48 => 243.82.77.124 FROM PORT 8448 TO all BY TCP

- Enabling Carol to work remotely:

ALLOW 129.34.156.78 => all FROM PORT all TO 22 BY TCP

ALLOW all => 129.34.156.78 FROM PORT 22 TO all BY TCP ACK

- Blocking incoming traffic from suspicious host:

DROP 84.71.99.0/24 => all FROM PORT all TO all BY BOTH

- Allowing for DNS lookups

RFC 1034 accords that on the internet, queries are carried in UDP datagrams or over TCP connections [2]. Although it might be the case that larger information is exchanged for TCP, DNS lookups mostly use UDP.
ALLOW all => 243.82.76.43 FROM PORT [1700-1750] TO 53 BY UDP
ALLOW 243.82.76.43 => all FROM PORT 53 TO [1700-1750] BY UDP

References

[1] "NIST Password Guidelines: The New Requirements You Need to Know", National Institution for Standards in Technology, Retrieved November 1, 2022. Available: https://www.auditboard.com/blog/nist-password-guidelines/

[2] *DOMAIN NAMES - CONCEPTS AND FACILITIES,* RFC 1034, Retrieved November 4, 2022. Available: https://www.rfc-editor.org/rfc/rfc1034.html