Solver: Aryani Paramita

Email Address: ar0003ta@e.ntu.edu.sg

1. (a)
   (i) Weak Authentication: Claimants authenticate themselves by **giving up** the secret. Such as key in password to the system without further follow up. In this mechanism, only claimant need to prove themselves, but not verifier. Therefore it is a weak authentication that is prone to phishing activities.
   (ii) Strong Authentication : Authentication mechanism to check both claimant and verifier. Strong authentication mechanism **would not** allow claimant to prove the knowledge by giving up the secret and typically is executed by challenge-respond protocol

   (b) Hash → all passwords in the folder is hashed therefore, it would be hard to retrieve the password although hash value is obtained.
   Salting → before hash is computed, salt is added to the password, therefore it would further prevent rainbow table attack.
   Access Restriction → Only root could access this folder.

   (c) Case sensitive alphabets = 52 chars | 0-9 => 10 chars | Special Characters = 5 chars
   Total combination = $(52+10+5)^8 = 67^8 = 4.06 \times 10^{14}$

   (d) $\log_2(67^8) = 48.52$

   (e) D
   A
   A
   C
   D

2. (a)

	fyp_report.doc	fyp_code.c	fyp_code.exe	fyp_present.ppt	fyp_results.txt
Alfred	{R,W,O}	{R}		{R}	
Betty	{R}	{R,W,O}	{R,E,O}	{R}	
Carol	{R}	{R}		{R,W,O}	
David					{R,W,O}

   (b)

	fyp_report.doc	fyp_code.c	fyp_code.exe	fyp_present.ppt	fyp_results.txt
Alfred	{R,W,O}	{R}		{R}	
Betty	{R}	{R,W,O}	{R,E,O}	{R}	
Carol	{R}	{R}		{R,W,O}	
David	{R}		{R,E}	{R}	{R,W,O}

   Fyp_results.txt | David:{R,W,O}

   (c) 110 100 000

   (d) chmod 605 fyp_code.exe

   (e) chmod 4605 fyp_code.exe

   (f) umask
   Umask 077

3. (a)(i)To call the program and suply"%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s" as the argument.

(ii) The program allow us to print the content of the stack. Since we have transferred the retrieved secret into variable secret (int) ,which is located in the stack, printing the stack will caused the secret to be exposed.

(b) (i)printf("%s", str);

(ii)The existing code will enable attacker to print the content of the stack in the program, therefore secret is easily exploited. However, by adding %s in printf method will prevent such attack, thus it will be more secure.

(c) No. Since the secret is obtained via printing the stack, stack guard enabled compilation will not prevent attacker to print the stack. Therefore, stack guard enabled will not help in this situation.

4. (a) (i) Without *- property S3 could easily observe from O3 and alter O2 at the same time, (in this case is to copy the information content of O3 to O2). Then, S1 could read O2 which also contains information from O3.

(ii) With *-property and assumption that S3 does not have capability to memorize information, S3 could not observe O3 and alter O2 at the same time. Thus, this mechanism will prevent S1 to obtain the information in O3.

(b) In a corporation, an employee which is the owner of the file could create an important file. However, once the employee leave or move to another department, he should not be able to edit or even view the document. In a unix system, an owner could easily incorporate chmod to grant himself access over the document. Therefore, there is a need to transfer ownership of files when there is a change in organization. However in windows, the transfer does not need to be done as the system still allow ex-employee to be the owner yet restriction is imposed to the owner.

(c) A malicious app could simply set up the same intent as the legitimate application. When the intent is triggered, the malicious app, typically via social engineering will lure the user to choose malicious app over the legitimate app. In this situation, the data will be passed to the malicious application.

Thank you and all the best for your exams! ☺