IERG4130 Midterm

Name: YU Sihong

SID: 1155141630

Q1

- which: Separation of privilege;
- why: because give the full privilege to user directly is risky and may do harm to the system file easily. Separation of privilege is able to reduce the risk. And a normal user can already accomplish most of the work. Typing "sudo" also make user be aware of that they are doing something risky.

Q2

- Problem: His computer was possibly effected by some malicious software, i.e. virus or ransomware.
- What to do: Download and keep the official protection software open in the computer and the browser, and don't open, click, download things on the unfamiliar website, or in virsual machine in advanced.

Q3

- NO
- why: a true random number generator basically is based on hardware, it costs more to build such generator than the software one. And it is not convenient for every user to hold such hardware in their daily life. And a pesudo generator nowadays is good enough. The true random numbergenerator loses in the trade-off of cost and performance.

Q4

- Some opening ports of the system
- As a system can have many ports, the user may forget to close some of these after using it, some the port may be detected and exploited by worm. And if the port do not need to login or do not be protected by firewall, the worm will be able to get into the system and read the files inside or even run their injected code to harm the CIA of that system.

Q5

- first to authentication;
- because authentication is to identify the objects whereas authorization is to give privileges that objects. It is risky to directly give privileges to someone without identifying. And it is also possible to exposure some useful message to the un-identified object during communication.

Q6

- validation is to check if the user input is in some pattern (legal) or not, and reject the illegal inputs directly. Whereas santization trys to understand the ambiguous inputs from user, and just do substitution for the inputs.
- validation is better.
- validation is more straight-forward and easier to implement and perform better. Because it is true that you never know what the user will input and hard to underatand and handle all the ambiguity of user. And if the input is still not the legal input, it will do harm to your system.

Q7

- random initial vector
- they are all need to initialize randomly, and the more random they are, the more secure they will be. The key point to improve their security is never reuse the initialized value.

Q8

- OFB
- In the noise communication channel channel, it often loses texts. The CBC is more sensitive to it, because every next block is depent on the privious block. And if some texts are lost, CBC will stop until it receives the text successfully. Whereas the OFB can generate the feedbacks in advanced. If some texts are lost but the keys are recieved, it is able to en/decrypt the existing texts and go back to the lost ones in the future. In sum, the OFB will perform better in such noise channel. And in terms of the implementation part, they do not have much difference.

Q9

- CTR
- Because it donot need feedback and only needs a counting number, CTR has no inter-dependence and is robust to the noisy channel. And CTR has some data can be pre-calculated so it is good for parallelism and burst communication at high spped.

Q10

- Frequency based attack
- To Find the vulnerability, which is the probability of words like "AA", and so that it is able to predict some values of key.
- Maintance a single IC, because the whole ciphertext is just the substitution of plaintext.

Q11

- 3
- 3^1024 mod 7 = (3 mod 7 * 9 mod 7 * ...)^x * (...) mod 7 = ((3*2645*1) mod 7 * (3264) mod 7 mod 7) = 6 * 4 mod 7 = 3
- 3^x mod 7 has a cycle of 3 2 6 4 5 1, and 1024 mod 6 is 4, so the answer is above

Q12

- Because it do not manage the process of sending the key and can not comfirm who will hold the key eventually.