IS Department

MILITARY COLLEGE OF SIGNALS, NUST

Computer Security

BESE-14 (Network Security Stream)

**Exam:** Mid-Term **Instructor:** Asad Raza

**Type of Paper:** Regular **Total Marks:** 18

**Semester:** 6th **Time Allowed:** 1 hr 15 min

**Note:** The purpose of providing Solution is that you can compare your answer with this solution and don’t come to the teacher arguing over wrong solutions. I will only entertain arguments if have missed anything in your paper which you have written but it hasn’t been marked

**Question No 1. To be solved on question paper (3+ 2 Marks)**

Give the summary of 6 messages exchanged in Kerberos and also the contents of Tickettgs and Ticketv.

1. **C to AS ……………………………………………………**
2. **AS to C ……………………………………………………**
3. **C to TGS ………………………………………………….**
4. **TGS to C ……………………………………………………**
5. **C to V ……………………………………………………**
6. **V to C ……………………………………………………**
7. ***Tickettgs =***

1. ***Ticketv =***

**Solution : Full marks will be given only if all the contents and structure of messages is correct neglecting lifetime / time stamp fields**

* (1) C to AS *IDc||IDtgs||TS*1
* (2) AS to C E(*Kc*,[*Kc,tgs*||*IDtgs*||*TS*2||*Lifetime*2||*Tickettgs*])
* (3) C to TGS *IDv*||*Tickettgs*||*Authenticatorc*
* (4)TGS to C E(*Kc,tgs*, [*Kc,v*||*IDv*||*TS*4||*Ticketv*])
* (5) C to V *Ticketv*||*Authenticatorc*
* (6) V to C E(*Kc,v*, [*TS*5 + 1]) (for mutual authentication)

**Question No 2: (2 Marks)**

Assume that the Bell-LaPadula security model has been implemented in a system. Alice has a ‘*secret’* clearance and Bob’s clearance is ‘*classified’*. Which of the following operations are not allowed, assuming that both Alice and Bob operate at their highest clearance level?

* 1. Bob reads a document written by Alice.
  2. Alice sends Bob a document that she has written.
  3. Alice reads a document with the label ‘secret’.
  4. Bob reads an unclassified document and sends it to Alice.

Solution :

1. Bob reads a document written by Alice. (Not Allowed )
2. Alice sends Bob a document that she has written. (Not Allowed )
3. Alice reads a document with the label ‘secret’. (Allowed)
4. Bob reads an unclassified document and sends it to Alice. (Allowed )

**Question No 3. (4 Marks)**

How does a window store user passwords? Also explain how NT and LM Hashes are generated?

**Solution**

**Part 1 Ans (2 Marks )**

* Windows does not store passwords in clear text. Passwords go through a process to generate an LM hash and an NT hash. Hashes are one-way mathematical functions and hashed passwords cannot be reverse engineered. Windows store these hash passwords in SAM file which is located in System32 directory.

**Part 2 (1 + 1 )**

**Lan Manager or LM Hash Generation**

1. Convert password to upper case

2. Pad the plaintext with null characters to make it 14 bytes long.

3. Split into two 7 character (byte) chunks

4. Use each 7 byte chunks separately as keys to DES -encrypt an ASCII constant string

5. Concatenate the two cipher texts from step 4 to product the hash.

6. Store the hash in the SAM file

**NT Hash Generation**

* 1. Take the Unicode mixed‐case password and use the Message Digest 4 (MD4) algorithm to obtain the hash.
  2. Store the hash in the SAM file.
  3. The LanMan hash is only stored for passwords that are 14 characters or fewer
  4. The NT Hash handles passwords up to 128 characters in length

**Question No. 4 (2 Marks)**

When a file object has an implicit label and two hard links from different directories then it may have two labels. Either support or refute this state with arguments.

Solution

Please read the solution . But a file object cannot have two labels in any case. So those students who have supported this argument, their answer is incorrect and cannot be awarded any mark.

There are two cases which apprise 1. When the system is on DG/UX system 2. When it is mounted on DG/UX ..Whoever has written it will be awarded full marks …

**Question No 5 : (3+2 Marks)**

One of the main problems with password authentication is that we commonly have to authenticate ourselves to several different systems. Review this list of good advice on password choice and handling

* Don't use only letters.
* Make passwords long.
* Avoid names, words and other common patterns
* Change regularly.
* Don't write them down.
* Don't tell anyone
* Do not reuse passwords from one system on another system.

1. Which of these principles would you choose to relax when the number of passwords becomes too many to handle securely? Motivate your answer with arguments.
2. Suggest a strategy that the system administrator could use to alleviate the problem.

**SOLUTION (Part a)**

Of all of these pieces of advice, there are some that we can discount immediately since it is very unlikely that their relaxation would help the situation at all. These are *Don't use only letters*, *Make* *password long*, and *Don't tell anyone*. The only reason that the relaxation of any of these might help is if it were to make a large number of password easier to remember. I contend that none of these three make any significant difference. Surely a password can be just as memorable if it has, for example, some punctuation marks in it. Assuming that you have some system to remember passwords by, longer passwords are surely just as memorable; it is just that they take more time to type in. Telling someone does not seem to have any advantages relevant to this situation. No exam answers had any convincing arguments that any of these factors would make a difference.

Using names or commonly used patters might make it easier to deal with many passwords. Perhaps we could use a word or a name that is easy to associate with the systems that we are logging on to.

However, we know that any kind of pattern that can be used to create a dictionary of possible passwords can very effectively reduce the search space for a program that can test its way to correct passwords, making such passwords very weak. We can assume that the negative effects of using words will outweigh any advantages.

Relaxing the advice to change regularly might be considerable help. If we change passwords half as often we are effectively reducing the number of passwords we need to remember by half. The threats naturally increase. There is always a threat that a password that is used often might be discovered by someone eavesdropping, either in terms of watching the keys we hit, installing a keylogger, or listening on unencrypted communication lines. It seems that all of these threats are fairly linear, so if a password is kept for twice as long, we could assume that the threat is doubled. Depending on how likely these threats are, we might assume that a doubling of the a minimal threat is still minimal, and worth the risk. However, there is another kind of attack to worry about: the discovery of encrypted passwords. There is a rule that says that passwords should be changed every third month (ok, not always third month, but I believe this to be the most common interval), and this is presumably based on the idea that after that time even difficult passwords can be cracked.

Changing at around this time would reduce the chance that a difficult password is compromised, and limit the time that a cracked password can cause any damage. The relationship between time and security may not be so linear. Perhaps there is a significant increase in the risk after a certain amount of time that has to do with how difficult passwords usually are, and how powerful current password cracking programs are. It is in any event difficult to quantify these risks, so if one claims that the increased risk of not changing passwords so often is worth taking, then it is difficult to argue either for or against. But the answer would be correct if someone chooses to relax it .

“Do not reuse passwords from one system on another system”. I was surprised that some of

Students proposed reusing passwords on several systems. This could no doubt be a very

effective way to reduce the number of passwords, ultimately leaving us with possibly only one

password to remember. Some students claimed that if the password was a good one, then its

strength would mean that it could be used on several systems. I suggest that this is a serious

misconception.

The problem with reusing passwords has nothing to do with the password's strength.

We usually cannot determine or control the security levels of the systems including how they handle our passwords, so if we reuse a password its level of security is determined by the weakest of the systems we log on to. In one fell swoop we can drastically reduce the security of all of the systems we log on to. In the worst case scenario we might use the password on an insecure web-site with an unscrupulous webmaster who deliberately saves all the passwords used at that site to attack other systems.

It might possibly be ok to reuse passwords under the assumption that you have good knowledge of the security of each and every system that you reuse it on. None of the students who suggested

password reuse as a strategy wrote about these necessary assumptions. In general, we must assume that this is a very poor strategy.

If we relax the advice not to write passwords down we have an effective method for keeping track of many different systems. However, it could potentially be compromising since the security of all the systems we use is now dependent on how the written down passwords are kept. Note however that we still have the level of security under our control, so if we can find a method that we can reasonably assume is at least as secure as the most secure system on our list we will not have to compromise security at all. It would not be a good idea to create a list where potential attackers can get at it (on a post-it note on your computer screen? in your desk drawer?), but other methods may be more secure.

Keeping a list in your wallet would normally mean that it would be physically under

you control in all situations where an attacker might be at large. Your wallet might be stolen but that kind of attack is usually noticeable, so it gives you the opportunity to have your accounts closed or your passwords changed before too much damage occurs. You can take certain measures to improve security, such as inventing a personal system for encoding them so that even if the text is seen by an attacker, the actual password would not be revealed. Devices such as PDAs often allow a text to be protected by a password, which could give you an extra method of protection. A disadvantage of writing passwords down is that when you need to remember them you have to retrieve your list. This involves a certain amount of inconvenience, and also gives an attacker the chance to eavesdrop either your method or your actual password.

However, on balance, it would seem easiest to argue for a relaxation of “Don't write them down”. **The answer is correct if someone has chosen to relax this with supporting arguments**

Note that question about what a system administrator could to do to alleviate the problem, “the

problem” refers of course to “the number of passwords becomes too many to handle securely”.

Some students' answers included such methods as checking for easy passwords, forcing password changes after a set period of time, creating randomly structured passwords for the user, etc. These may be methods to ensure the security of password based authentication, but they are hardly solutions to the problem of passwords becoming too many to handle.

I hope that you remember what I told you in class. Please read the question carefully and don’t jump into writing the solution.

One clear answer to the too many passwords problem is for system administrators to install a Single Sign-On system. Kerberos is one such system that we have briefly covered during the course. A single (hopefully strong) password gives a user access to tickets that in turn give access to a number of services that would otherwise all require individual passwords.