Exercises with solutions, Set 3

EDA625 Security, 2017

Dept. of Electrical and Information Technology, Lund University, Sweden

Instructions

These exercises are for self-assessment so you can check your knowledge during the course. You do not need to submit the answers.

- These exercises are based on material in Chapters 11-15 (except 14 and 15.5) and course slides.
- Some exercises may require you to look for information outside the textbook as well.
- Exam questions can be based on some, part of, or none of the exercises.
- The teacher(s) is/are happy to help you out if you run into problems.

Exercise 3.1

a) Describe the information flow in the Bell-LaPadula security model.
b) Assume that process A has a higher security level than process B. Moreover, process A is not considered trusted and the security level of an object cannot be changed. How is it possible for process A to send information to process B without violating the access rules of the model?

Exercise 3.2

a) What is a replay attack?
b) How can a replay attack be prevented?
c) Explain how Kerberos prevents replay attacks.

Exercise 3.3

Following the unexpected failure of the xor-based three-way protocol from exercise 1.6 the same person has now invented a new protocol. The new protocol is based on the Encrypted Key Exchange (EKE) protocol designed to resist offline dictionary attacks. In the original protocol Alice uses her password to encrypt a public key and the result is sent to Bob. Bob then encrypts the session key using Alice's public key. The result is then again encrypted using the shared password. The observation leading to the modification is that symmetric algorithms are much faster than asymmetric ones, so why waste any time? Instead of encrypting a public key with the password Alice now encrypt a 128 bit AES key using the password. Subsequently Bob decrypts the AES key using the shared password, encrypts the session key using AES, and encrypts the result using the password. Analyze this protocol with respect to offline dictionary attacks. Is it still as resistant as the original protocol? NOTE: As with the original protocol we have to assume that anyone can intercept data encrypted with the session key and that this data has some redundancy.

Exercise 3.4

a) What parameters are assumed to be known to the man-in-the-middle attacker in the Diffie-Hellman protocol?
b) Show how the attack works using a numerical example. Explain clearly who chooses and has access to the different parameters.

Exercise 3.5

Give at least 3 important differences between TCSEC and Common Criteria.

Exercise 3.6

The Clark-Wilson model is closely connected to object oriented languages and their security model. Explain how Clark-Wilson is relevant in connection with Java.

Exercise 3.7

In the Bell-LaPadula security model, two different security levels are used for subject. The maximum level $f_S(s)$ is used to control read access and the current level $f_C(s)$ is used to control append access. Since we always have that $f_C(s) \leq f_S(s)$ it would be possible to read an object o while at the same

time appending to an object o', with $f_O(o') \leq f_O(o)$. The downward information flow made possible here is solved by adding an extra restriction to the *-property.
a) Describe this extra restriction to the *-property.
b) What if we instead modify the ss-property, and instead of using $f_S(s)$, we use $f_C(s)$ to control read access? Then it would not be possible to have read access to an object with higher security level than objects we have append access to. Analyse this alternative approach.

Exercise 3.8

Operating system A has been evaluated according to the Orange Book and has received the classification C2. Operating system B has been evaluated using Common Criteria and has received level EAL4. What can you say about the functionality and assurance for OS A and OS B?

Exercise 3.9

a) Describe at least three different firewall functions?
b) What is DMZ?

Exercise 3.10

A VPN solutions is, loosely defined, a secure connection between a client machine/network and a server (gateway) in another network designed such that applications in the client do not have to be aware of the presence of the VPN (their code is not affected by it). **a)** Explain why TLS cannot directly be used for a VPN?
b) Describe a way to design a system for a secure VPN connection that uses TLS.
c) Explain briefly why IPsec has not the same problem as TLS. Also indicate what is additionally needed in a IPsec based VPN.

SOLUTIONS EXERCISE SET 3

Solution Exercise 3.1

a) No read-up and no write-down.
b) Process A can be temporarily downgraded to a lower security level and write information to an object at that level. After writing, the security level of process A can be restored to $f_S(A)$. The information can then be read by process B.

Solution Exercise 3.2

a) Eve records some encrypted and/or authenticated communication between Alice and Bob and uses it to later fool one or the other of them. For example, if Alice tells Bob to buy 500 Volvo stocks and Eve later replays the conversation with Bob, he will buy more stocks than he should and Alice will get the blame. Note that Eve doesn't have to know *what* she is replaying, she just has to do it and hope for some (disastrous) result.
b) Alice and Bob could make sure that each conversation is unique (not the information in it, of course, but the session setup, encryption, etc.). Some tools would include nonces or timestamps.
c) When Alice contacts the server she uses a nonce. The server responds with a newly created session key. The fact that it is new can be verified since the nonce is included in the response. Furthermore, when contacting Bob with the session key data, Alice includes a time stamp which he can compare against his own clock.

Solution Exercise 3.3

Let's understand the scheme: Alice and Bob have a shared key, S. Alice creates a key pair and sends $X = E_S(PK)$ where PK is her public key. Bob knows S so he can find PK. He chooses a session key SK and sends $Y = E_S(E_{PK}(SK))$ to Alice. Since she knows both S and her private key, she can find SK. Eve can only observe X and Y. (Some observations: Bob needs to trust that Alice never uses an old key pair while, similarly, Alice needs to trust that Bob knows how to generate good session keys.)

The modified scheme is as follows: Alice generates a symmetric key AES and sends $X' = E_S(AES)$ and Bob responds with $Y' = E_S(E_{AES}(SK))$. Eve observes X' and Y'.

Doing a dictionary attack, Eve tries out some possible values for S. She calculates $AES' = D_S(X')$ and $SK' = D_S(D_{AES'}(Y'))$. She uses SK' to decrypt some session data. Since we assume some redundancy in the data sent between Alice and Bob (for example, they might be communicating in a human language), Eve will realize when she has made a correct guess of S and she now has the session key SK'.

Would the same attack apply to the original protocol? Well, Eve could guess S and calculate $PK' = D_S(X)$ but she'll have problems calculating $D_S(D_{PK'}(Y))$ since the public key is of no use for *decrypting*!

Solution Exercise 3.4

a) The man-in-the-middle needs to know p and g, but this is a fair assumption. Alice and Bob can either a) use a predetermined parameter set[1] which has to be assumed public by Kerckhoff's principle, or b) start their conversation by negotiating p and g. This negotiation can be eavesdropped.

[1] In SSH, there are many variants of Diffie-Hellman, but one option that all clients have to support is `Diffie-Hellman-group1-sha1` which uses a quite scary $p = 2^{1024} - 2^{960} - 1 + 2^{64}\lfloor 2^{894}\pi + 129093 \rfloor$ and a somewhat simpler $g = 2$.

b) We'll use $p = 11$ and $g = 2$. a, b, x, z used below will be picked randomly. Alice chooses $a = 4$ and sends $y_a = g^a \bmod p = 5$. She thinks that she sends it to Bob, but Eve intercepts it, picks $z = 3$ and sends $y_z = g^z \bmod p = 8$ to him. Bob cheerfully picks $b = 7$ and sends $y_b = g^b \bmod p = 7$ to Eve, thinking she is Alice. Eve picks $x = 9$ and sends $y_x = g^x \bmod p = 6$ to Alice.

Alice will compute $A = y_x^a \bmod p = 9$ and Bob will find $B = y_z^b \bmod p = 2$. Clearly, they do not share keys. However, Eve calculates $A' = y_a^x \bmod p = 9$ and $B' = y_b^z \bmod p = 2$. She will now be able to relay Alice's and Bob's "secret" communications while recording everything!

We see that Alice chooses a while Bob chooses b, just as intended, but that Eve selects x and z. Alice will have access to a and A and Bob to b and B. Eve will have access to x, z, A and B.

Solution Exercise 3.5

An answer might include the following:

- The TCSEC was a US standard while the Common Criteria is an international one (however, only EAL1 through EAL4 assurances are recognized in all countries; higher levels of assurance have to be accepted in each country that the product targets).
- Evaluation according to TCSEC was free of charge, while a Common Criteria evaluation can be quite costly.
- TCSEC does not separate functionality and assurance, while Common Criteria somewhat separates this by using Protection Profiles.
- TCSEC primarily focuses on operating systems while Common Criteria can be used to evaluates several other types of security products.

Solution Exercise 3.6

In Clark-Wilson the objects can only be manipulated through a restricted set of programs. This relates directly to the way Java has objects that have data, some of which private, and the have a set of methods that operate on the data.

Solution Exercise 3.7

a) The extra restriction says that if we have access to several files, and this access sometimes includes append and sometimes includes read, then the security level of all files that we can append to must dominate the security level of all files we have read access to.
b) Let us first consider only the read access. Since you will always be able to read files with security levels that are dominated by your maximum level, f_S, there is no motivation to use f_C.

If we instead consider append access, then this restriction in the ss-property is motivated since you will never be allowed to read objects with security level that dominates objects that you can append to. However, the result would actually be a little more restrictive than needed. Consider the case where we have 4 classifications. For somplicity we call them 1, 2, 3, 4 where 4 is the highest security level (system high) and 1 is the lowest (system low). Assume that user Alice has $f_S(Alice) = 4$ and that she downgrades her current level to $f_C(Alice) = 1$. In this case, reading an object with $f_O(o) = 2$ while at the same time appending to an object with $f_O(o) = 3$ would not risk any information flow downwards. This is also not violating the original security properties. However, with the proposed property this situation would be forbidden..

Solution Exercise 3.8

Since the Orange book does not separate functionality and assurance it is possible to determine both functionality and assurance for OS A. At least to a certain extent. It has e.g., discretionary access control implemented for individual users and it implements an audit function. However, we do not know if it has mandatory access control. It is possible that it implements some functionality for higher evaluation classes but not all. For OS B we know the assurance level, but we do not have any information about the functionality of the product. To know this we additionally have to look at the Protection Profile(s) that it is evaluated against.

Solution Exercise 3.9

a) Packet filter that filter on protocol types and routing information, stateful packet filters that also understand the requests and replies in the protocols, circuit-level proxy, application-level proxy. Proxies are non-routing filters. From the outside one sees only the proxy (server).
b) See page 332 in Course book.

Solution Exercise 3.10

a) An application must be programmed to use TLS. Hence TLS is not transparent from application point of view and therefore cannot directly be used as a VPN solution.
b) The trick is to introduce a virtual NIC (such as supported in Unix/Linux and Windows). The normal application use this virtual NIC for their data transport. This NIC however is not connected to the Internet, instead it is connected with a special VPN application that uses TLS and that uses a normal NIC that is connected to the Internet.
c)IPsec lies below the TCP layer in the communication stack between the TCP/UDP layer and the IP layer. Therefore applications that use TCP/UDP are not effected by IPsec and thus IPsec can directly be used in a VPN. However, since IPsec lacks a key establishment protocol a IPsec based VPN also requires a protocol to setup the keys for IPsec, e.g. IKEv2.