Final exam in

Computer Security EIT060

Department of Electrical and Information Technology
Lund University

March, 7, 2012, 14-19

- You may answer in either Swedish or English.
- If any data is lacking, make (and state) reasonable assumptions.
- Use legible hand writing. If your answers cannot be read, you will receive zero points on that problem.
- Only pencil and eraser is allowed.
- Grading is done as follows.
  Grade 3 = 20–29 points,
  Grade 4 = 30–39 points,
  Grade 5 = 40–50 points.

Problem 1. What is the difference between access control lists and capabilities?

Answer
An ACL describes what subjects can do with a particular object. This information is stored together with each object. Capabilities describe what a particular subject can do with objects and this information is stored with the subject. (3 points)

Problem 2. A (long) listing of the information about the /tmp directory in a typical Linux distribution is shown below.

```
drwxrwxrwt 16 root root 4096 2012-03-07 14:00 tmp/
```

Why is there a `t` in the permissions list? What would be the difference if it is replaced by an `x`?

Answer
The `t` is called sticky bit. When this is set on a directory, only the owner of the file and the owner of the directory is allowed to delete a file in the directory. If the bit is not set, anyone with write access to the directory would be allowed to delete it. (3 points)

Problem 3. How does the Orange Book, ITSEC and Common Criteria differ in their treatment of functionality and assurance of products.

Answer

In the evaluation classes in the Orange Book, there is no separation of functionality and assurance. Each class defines what functionality and what level of assurance is needed for that specific class. ITSEC completely separated the two concepts. One document was used to describe the functionality, and then this functionality could be evaluated at different levels depending on the assurance needed. Common criteria is not fully as flexible as ITSEC but also uses levels of assurance. The functionality is typically given by conforming to a predefined protection profile that states functionality requirements for a specific type of product, e.g., an operating system. (3 points)

Problem 4. What is a replay attack? How does the SSL handshake protocol protect against replay attacks?

Answer

In a replay attack, the attacker records messages and later replays them. The attacker may not be able to read the messages, if they are encrypted, but can just replay the encrypted data. If the replay goes unnoticed, the attack succeeds. In the SSL handshake, both the client and server send random numbers in the hello messages. In the finished message, previous handshake messages are hashed and encrypted. Since the random numbers are included someone replaying would not be able to produce the correct finished messages and a replayed message would not be valid. (3 points)

Problem 5. In the original Encrypted Key Exchange protocol (EKE), a public key is encrypted using a password. This will protect against offline dictionary attacks on the password. Breaking this scheme will require breaking either the symmetric key, or finding the private key corresponding to K_a. But if the security does not depend on the strength of the password, it should be ok to just omit the use of the password, shown in the modified EKE below. This is of course not true, why?

Original EKE	**Modified EKE**
1. $A \rightarrow B: \ eP(K_a)$	1. $A \rightarrow B: \ K_a$
2. $B \rightarrow A: \ eP(eK_a(K_s))$	2. $B \rightarrow A: \ eK_a(K_s)$

Answer

In the modified protocol, the authentication provided by the password is removed. Thus, anyone can send a public key, claiming the be A, or send a response, claiming to be B. This can be realized in e.g., man-in-the-middle attack or in a spoofing attack. Such an attack on the original protocol gives an attacker only one password guess so the password does matter for these types of attacks. (3 points)

Problem 6. Why is the ESP header not encrypted in IPsec transport mode?

Answer

Because the SPI is stored in the header. This is an index pointing to the the algorithms and keys used in encryption. (3 points)

Problem 7. Explain the idea behind SQL injection attacks. What is the main implementation mistake that allows the attack to succeed?

Answer
Often, web applications make database queries in order to get wep page content or to check e.g., login credentials. If the queries are based on user generated input, and the applications does not properly validate this input, it can be possible to manipulate the queries. Depending on the exact situation, the manipulated query can e.g., extract data from the database that was not intended to be extracted, or it can bypass password verification checks. (3 points)

Problem 8. An intrusion detection system (IDS) is based on *misuse detection* or *anomaly detection*. A *honeypot* has many similarities with an IDS. Describe when false positives and false negatives occur for each type of IDS and a honeypot.

Answer
In misuse detection, false negatives occur when the database of signatures does not include the actual intrusion signature. False positives occur when a signature matches with a behaviour that is actually not an intrusion. In anomaly detection, false negatives occurs when the intrusion is too close to the baseline for an alarm to be generated. False positives occur when the normal behaviour differs too much from the baseline and an alarm is generated. A honeypot should have no false positives or false negatives since all access are by definition an attack and everything can be monitored. (Variants including mistakes as a reason for false positives in honeypots would also be ok as answers.) (3 points)

Problem 9. The Biba security model uses the notion of *clean* and *dirty* subjects and objects. Explain what is meant by this.

Answer
We trust clean subjects to *provide* information that is more correct than dirty subjects. We trust clean objects to *contain* information that is more correct (or trustworthy) than dirty objects. Note that clean and dirty are relative notations. (3 points)

Problem 10. In an RSA scheme, the public modulus is given by $n = 391$ and the public exponent is $e = 7$. Break this RSA scheme by finding the private exponent d. Recall that $d \cdot e \equiv 1 \bmod \phi(n)$.

Answer
By factoring n we get $p = 17$ and $q = 23$. Thus, $\phi(n) = 16 \cdot 22 = 352$. With $d \cdot 7 = k \cdot 352 + 1$, and testing different k we find that there is a solution when $k = 3$, namely $d = (3 \cdot 352 + 1)/7 = 151$. (3 points)

Problem 11. The Diffie-Hellman protocol is probably the most well-known key agreement protocol. Variants of the protocol has been implemented in numerous applications.

a) Describe the basic Diffie-Hellman protocol.

b) The basic protocol is vulnerable to a man-in-the-middle attack. Describe this attack.

c) Describe one way to fix this vulnerability.

Answer

a) In the protocol, A and B decides on a prime p and a number g with certain properties. A secretly generates a number a and sends $g^a \bmod p$ to B. B similarly generates b and replies with $g^b \bmod p$. Now both can compute $g^{ab} \bmod p$ which is the shared secret.

b) An attacker can intercept the message from A and generate his own secret exponent x and forward $g^x \bmod p$ to B. In the same manner, $g^y \bmod p$ is sent to A. Now the attacker has one secret shared with A and one secret shared with B.

c) The problem is that A and B can not be sure of whom they are talking to as there is no authentication of messages. This can be fixed by using e.g., a MAC which requires a pre-shared key, or by digitally signing messages, which requires a certificate signed by a trusted third party. (5 points)

Problem 12. Passwords are treated and stored in Windows and Linux systems in similar ways, but there are some important differences. In addition to this, there are differences between different Windows configurations and also between different Linux configurations. We are of course here only interested in differences related to the security of passwords.

a) What is the difference of storing the hash in /etc/passwd and /etc/shadow?

b) Windows Vista and Windows 7 by default only store the NTLM hash. Previous versions additionally stored the LM hash. What is the difference?

c) How does the use of salt differ between Windows and Linux?

d) Why does a random unique salt for each user protect against time-memory-tradeoff attacks?

Answer

a) The file /etc/passwd is readable for everyone, while /etc/shadow is only readable for root. Thus it is more difficult to get the list of hashed passwords for users if it is stored in /etc/shadow.

b) The LM hash is contructed by separating the password into two independent chunks of 7 characters and additionally converting all characters to upper case. This significantly reduces the entropy of the passwords, making the LM hash much weaker than the NTLM hash.

c) Windows (local logon) does not use salts at all while they are used in Linux. Correct answers for Windows domain logon will also give full points.

d) The purpose of the tables is to invert the hash funcion. When constructing the tables all possible inputs to the hash function have to be tested (but not all are stored). The salt is a part of the input to the hash function but since the tables are precomputed we do not know the salt. Thus all possible salts have to be tested. If the salt is relatively small it can be included in the precomputation, but if it is large we have to construct one table for each salt (or a small number of salts). Thus, in the context of this attack the salt adds as much uncertainty of the input as the password itself. (5 points)

Problem 13. Not only clients need to be authenticated, but also the servers. This can be done in several ways.

a) What is it called when *both* entities in the communication are authenticated?

b) How is the network authenticated in UMTS?

c) How is the server authenticated in SSL when RSA is used as handshake method?

Answer

a) Mutual authentication

b) The network is authenticated by computing a MAC based on a *sequence number* known to both the USIM and the network. There is also a random number involved but this alone is not sufficient for authentication as it is generated by the network.

c) The server is authenticated by making it decrypt the premaster secret with its private key. Without the private key, the session key can not be computed. Presenting a certificate is not enough for authentication, but can be seen as a part of it. There is nothing secret in the certificate. (5 points)

Problem 14. Explain shortly the following five terms:

a) Controlled invocation

b) HMAC

c) Chinese wall model

d) Spoofing attack

e) Canary value

Answer

a) A less privileged process (or user) is allowed to perform actions requiring higher privileges. This is accomplished by carefully making sure the process only does exactly what it should be allowed to do.

b) A message authentication code based on a hash function.

c) A security model focusing on avoiding conflict of interest in e.g., consulting firms.

d) When an attacker tries to disguise as another user or process.

e) A value placed before the return address in order to detect buffer overflow attacks on the stack. (5 points)
