

Network Security IT-653

Assignment 3 by Staffan Lundström

1.1 SSL is susceptible to man-in-the-middle attack. Consider the following situation:

A is client and B is the server. C is the man in the middle. During the handshake A starts with `client_hello`. C intercepts it and changes the key exchange algorithm parameter in the `CipherSuite` so that Anonymous Diffie-Hellman is the only option supported by the client. Then C passes the message to B with the same nonce as A used.

C sends to `server_hello` and has no other choice than to choose Anonymous Diffie-Hellman. Since these two hello messages are not encrypted I guess that C also can see the random number.

B does not have to send his certificate. C intercepts B's `server_key_exchange` message. C notices the MAC that B sends during the `server_key_exchange` message. He changes the Diffie Hellman public key to his own, and calculates the MAC using his own private key. Then C sends the message to A. No `certificate_request` is sent. Last in this sequence is the `server_done` message.

A responds with a `client_key_exchange` message. C changes the client's public Diffie-Hellman parameters before he passes it to B.

After the last messages are sent C has succeeded with his attack and controls the whole communication.

This bug can be fixed by not allowing Anonymous Diffie-Hellman as key exchange algorithm.

1.2 Replay attack.

I have not found any replay attack that SSL is susceptible to. The nonces are the main component that protects against replay. There is no trivial way to prove that it is secure and therefore no proof is given.

2.1 I assume that it is the Needham-Schroeder public key authentication protocol that the question is about, since as far as I know the Needham-Schroeder secret key authentication protocol does not have seven messages.

Message 7 is not necessary for sharing the session key. But it is necessary for B to ensure that he's not exposed to a replay attack. A sends B the nonce it has just received, encrypted in B's public key, proving that the communication is fresh and that it is indeed A that is communicating. It protects against replay and

masquerading since only A could decrypt the nonce sent in message 6. In case of an attack, the intruder does not know the content of the message he has replayed.

- 3.1 First of all I have had very good experiences with Kerberos as a user. It's used at my home university and the system works exactly as you want it to work from a user's point of view. It's transparent and it's definitely not security by restriction. As a user you don't notice Kerberos at all except for the request of password.

The most serious drawback I can find is that the protocol does not rely on any formal analysis. So there are no guarantees that there are no flaws. The evolution of the Needham-Schroeder protocol shows how difficult it is to anticipate all possible attacks. The protocol is much more difficult to analyze than the public key authentication protocols.

I don't have any objections against this centralized design with a third party that also knows the secrets. It's an excellent solution in some environments. It's not a good design for all environments but all systems can be misused by using them in situations where they are not intended to be used.

- 3.2 Two enhancements in Kerberos 5 compared to Kerberos 4

i) *Encryption system dependence*

Version 4 requires the use of DES. DES is not considered to be secure any more. In version 5 any encryption technique can be used

ii) *Password attacks*

Both versions are vulnerable to password attacks. The message from the AS to the client includes material encrypted with a key based on the client's password. It can be captured, especially if the users are not so careful about the choice of password. But version 5 provides a mechanism known as preauthentication, which makes password attacks more difficult.