Identity Management Systems

Assignment I

In this assignment you will implement a simplified version of the CHAP - Challenge Handshake Authentication Protocol, based on a shared secret (password), which allows a client program to authenticate a user against a server program, without ever sending the secret over the network. The implementation can be achieved in the system and programming language of your choice.

The Protocol should contain at least the following steps:

1. The server application knows a set of user credentials (usernames and passwords) which should be stored in a file, with the passwords stored as digests (e.g.. MD5).
2. The client application prompts the user for their username.
3. The client opens a connection to the server (using for instance the socket API), sends the username in cleartext and waits for a response from the server.
4. The server checks if the username is included in the store and if so, retrieves te corresponding password digest. If the user is unknown, it reports an error to the client "Unknown User".
5. If the username is found, the server sends a challenge to the client, consisting in a 64 bit (8 bytes) nonce N, which can be obtained by concatenating the current time in milliseconds with a random number (see a possible example in the references).
6. The client application prompts the user for their password and calculates its digest (e.g. MD5), thus obtaining Dp = MD5(P), which is 128 bits (16 bytes) long.
7. The client performs the concatenation of the nonce N with the password digest Dp, thus obtaining N || DP and calculates the response R = MD5(N || DP), which it sends back to the server (16 bytes). The client then waits for a response from the server.
8. The server performs the same operation than the client, concatenating the user password digest Dp (which was retrieved from the storage) with the previously generated nonce N, obtaining R' = MD5(N || DP).
9. If R == R ', the client possesses the correct password and responded directly to the server (since the nonce is valid), thus proving the user's identity. Therefore, the server sends an "Authentication Success" message to the client.
10. If R != R´ either the user did not provide the correct password, or the nonce was not valid, and the server sends an "Authentication Failure" message to the client.

Note: Alternately, the more secure SHA256 hash function can be used, with the difference that the digests will be 256 bits (32 bytes) long.

References:

[1] CHAP - Challenge Handshake Authentication Protocol:
https://technet.microsoft.com/en-us/library/cc775567.aspx
http://www.ietf.org/rfc/rfc1994.txt
http://netlab.ulusofona.pt/im/teoricas/IM-02-IdMgmt.pdf

[2] Client / Server examples using sockets:
http://netlab.ulusofona.pt/im/praticas/ClientServer/

[3] Nonce generation example:
http://netlab.ulusofona.pt/im/praticas/nonce.java

[4] MD5 Digest example:
http://netlab.ulusofona.pt/im/praticas/md5.java