CPSC 526 - Assignment 2

Artem Golovin
30018900

March 2, 2020

Question 1: Password Management (20 marks)

Implemented in the current directory.

Question 2: Kerberos Forward Secrecy (10 points)

1. What are the long-term secrets in Kerberos?

   - K_C, a long term key for the client C, derived from the user's password.
   - K_{TGS}, a long term key for the Ticket Granting Service, known by Key Distribution Center (KDC) and TGS.
   - K_V a long term key for the network service V, known to V and TGS, each V has their own key.

2. What are the short-term secrets in Kerberos?

   - $K_{C,TGS}$, a short term session key for the communication between Client C and Ticket Granting Service, created by KDC, known to Client C and TGS.
   - $K_{C,V}$, a short term session key for the communication between Client C and Network service V, created by TGS, known to client C and TGS.

3. For each secret (long-term and short-term), assume that the secret is leaked to a passive adversary who has previously recorded all network traffic. What new information does this adversary learn that was not already known?

An adversary now will know the long-term secrets, since those keys are usually protected more than the short-term keys. Given that, an attacker will be able to do the following:

- K_C - once an attacker has this key, the communication between client and any other service can be compromised.
- K_V - in combination with K_C, this will give the attacker the knowledge of $K_{C,V}$.
- K_{TGS} - with this information, an attaker can learn $K_{C,TGS}$

4. Explain why Kerberos does not have forward secrecy. Be specific about what data needs to be compromised and what the consequences of it are.

   Forward secrecy is a feature that guarantees that the session keys will not be compromised if the private keys is compromised. That means, for every session initiated by users, a new unique session key would need to be generated.

   Kerberos aims to be fast, adding forward secrecy would slow down the algorithm as it will introduce additional overhead. From Kerberos RFC, " Applications requiring perfect forward secrecy must exchange keys through mechanisms that provide such assurance, but may use Kerberos for authentication of the encrypted channel established through such other means."

   Leaking K_C (q. 2.3) would compromise the communication between client and TGS and any network service V. If an active eavesdropper records the traffic, they could decrypt the message now that they know the secret key.

5. Augment Kerberos to have forward secrecy for the actual communication between Alice and Bob. You only need to specify the message sequences for the parts of the Kerberos protocol that you change (i.e., if Alice's communication with the TGS is unchanged then you do not need to specify it).

   To achieve perfect secrecy, Kerberos needs to be integrated with Diffie-Hellman Key Exchange.

   Before communication is established, the client C and the Authentication Server AS need to agree on some large prime number p and a generator g. C picks a secret integer a and AS picks a secret integer b.

(a) $C \rightarrow AS$: $\{C, TGS, g^a \pmod p, T_C\}_{K_C}$

(b) $AS \rightarrow C$: $\{TGT, g^b \pmod p, T_{AS}, lifetime\}$

(c) ... The rest of Kerberos

The steps are performed for every possible communication. C and AS share secret g^{ab}, thus providing Kerberos with perfect forward secrecy.