Jeff Kremer

11524078

CptS 427

Homework 3

**General Questions**

1. Password Entropy
   1. log2(26) \* 10 = 4.7 \* 10 = 47 bits of entropy
   2. log2(67) \* 8 = 6 .1 \* 8 = 48.8 ~ 49 bits of entropy
   3. log2(10) \* 6 = 3.32 \* 6 = 19.92 ~ 20 bits of entropy
   4. Human- Based entropy to == 49 bits of entropy
      1. first character = 4 bits; total entropy = 4 bits, total chars = 1
      2. next 7 characters = 2 per character; total entropy = 18 bits, total chars = 8
      3. chars 9 – 20 = 1.5 per character; total entropy = 34.5 bits, total chars = 20
      4. chars 21+ = 1 bit per char; total entropy = 49.5 bits, total chars = 35
      5. assuming all lowercase chararcters: **35 characters to be equivalent to b)**
      6. Assuming uppercase and special characters: (+6 entropy): **29 characters**
2. A salt is random data that is used in addition to a password when generating a password hash. Hashes do not need to be memorized, and thus dramatically increase the size of a hash table required to succesfully brute force a password, without placing additional burdens on the users.
3. System A will be much easier to crack, as 256 rainbow tables will be required to account for all salts. This is less than the Unix 12-bit salts. System B’s 32-bit salt is not feasible to crack using rainbow tables due to requiring 2^32 full rainbow tables, the amount of storage to be required for this is extremely large.
4. V does not provide a random value to C to randomize the signed hash. An Eavesdropper or Man-in-the-middle could read/intercept the response, and in the future log into the system because the signed hash C sends to V never changes.