Research Week 3

Chaz Davis

Question

An early attempt to force users to use less predictable passwords involved computer-supplied passwords. The passwords were eight characters long and were taken from the character set consisting of lowercase letters and digits. They were generated by a pseudorandom number generator with 2^{15} possible starting values. Using the technology of the time, the time required to search through all character strings of length 8 from a 36-character alphabet was 112 years. Unfortunately, this is not a true reflection of the actual security of the system. Explain the problem.

answer

Analyze the actual security of the computer supplied passwords: In the early days to select strong passwords, the computers use a psuedo random number generator to generate passwords which are 8 characters in lengths using 36 character alphabets. the number of possible strings that can be generated using 36 characters with a length of eight characters is 36^8 which is aproximately equal to 2^{41} But the psudorandom generator can produce 2^{15} passwords. as a result, the number of passwords that needs to be looked at to hack is only 2^{15}. Therefore, this system does not provide much security.

Question

A phonetic password generator picks two segments randomly for each six-letter password. The form of each segment is CVC (consonant, vowel, consonant), where V = < a, e, i, o, u > and C = V . What is the total password population? What is the probability of an adversary guessing a password correctly?

part a: answer

the password generator generates 6 letter passwords randomly which consist of two segments. the segment is

- The segment is of the form CVC (consonant, vowel, consonant) consist of 5 letters consist of 21 letters

- 6 letters the 6 letter password is of the form CVCCVC
- The total number of passwords that a phoetic password generator can generate is $21x5x21x21x5x21$ which is equal to: 4,862,025.

part b: answer

The probability to guess a password correctly is calulated using the following formula:

$$\text{Probability of correct number of guesses} = \frac{1}{\text{Total password population}}$$

Here, the total password population is 4,862,025; substitute it in the above formula to find the probability.

$$\text{Probability of correct password guesses} = \frac{1}{4,862,025} \approx 2x10^{-7}$$

So, the probability to guess a correct password by an adversay is approximately equal to $2x10^{-7}$

Question

It was stated that the inclusion of the salt in the UNIX password scheme increases the difficulty of guessing by a factor of 4096. But the salt is stored in plaintext in the same entry as the corresponding ciphertext password. Therefore, those two characters are known to the attacker and need not be guessed. Why is it asserted that the salt increases security?

answer

Salt: An extensively employed password security is the usage of a salt value and hashed passwords. This password scheme is found on all UNIX systems as weall as on other operating systems.

the salt has certain features that increase the security of the password scheme

Different salt values for the same password Salt averts duplicate passwords in the password file Even when two end-users pick the same password, those passwords will be given different salt values. For this reason, the hashed password of the two end-users will be different Increase the difficulty of guessing a pasword Salt significantly increases the difficulty of dictionary attacks If salt is of "b" bit length, then the number of possible passwords is amplified by a factor of "2^b".

So, for those reasons, it is understood that the salt increases security

Question

Assuming you have successfully answered the preceding problem and understand the significance of the salt, here is another section. Wouldn't it be possible to thwart completely all password crackers by dramatically increasing the salt size to, say, 24 or 48 bits?

answer

Protecting the security of the system completely from password crackers does not depend on the salt size but depends on the user population size. Using the hash function of the cipher password and the randomly generated salt and both are stored in the password file. Increasing the size of salt leads to resolve problems of two users having the same salt. *Because, if the salt is the same for more users then the attacker needs to do seperate encryptions for each password of the user* So, increasing the sie of salt through bits, such as 24 or 48, does not protect the security of the system completely from all password crackers.

Question

A relatively new authentication proposal is the Secure Quick Reliable Login (SQRL) described here: https://www.grc.com/sqrl/sqrl.htm. Write a brief summary of how SQRL works and indicate how it fits into the categories of types of user authentication listed in this chapter.

answer

Question

In general terms, what are four means of authenticating a user's identity?

Something the individual **knows**: a password a Personal Identification Number (PIN) answers to prearranged questions Something the individual **possesses** (these are known as tokens): Electronic key-cards smart cards physical keys Something the individual **is**(static biometrics): Fingerprint retina face Something the individual **does**(dynamic biometrics): voice recognition handwriting characteristics typing rhythm

answer

Question

What are two common techniques used to protect a password file?

answer

Using a salt this is stored in plain text with the hash from (salt + password) **Password File Access Control** The hashed passwords are kept in a seperate file from the user IDs refferred to as shadow password file. Only priveledged users have access to this file.

Question

Explain the difference between a simple memory card and a smart card.

answer

Memory Card stores but does not process data **Smart Card** Has a micrprocessor, different types of memory, I/O ports, etc. May also have a cryptocoproccessor and an embedded antenna.

Question

Define the terms false match rate and false nonmatch rate, and explain the use of a threshold in relationship to these two rates.

answer

False Match Rate It measures the percent of invalid inputs which are incorrectly accepted. **False Non Match Rate** It measures the percent of valid inputs which are incorrectly rejected.

By moving the threshhold, the probabilities can be altered but note that a decrease in false match rate doesn't necessarily result in an increase in false non-match rates, and vice versa

Question

Describe the general concept of a challenge-response protocol.

answer

The host generates a random number r and returns it to the user (=challenge). In addition, the host specifies two functions, a hash function h() and another f() to be used in the respomse. The user calculates $f(r', h(P'))$, where r' = r and P' is the user's Password. When the response arrives, the host compares the incoming result to the calculated $f(r, h(P))$ and if it matches the user is authenticated. Advantages: Only the hashes of the passwords have to be stored and they do not have to be transmitted directly, so it cannot be captured during transmission.

bonus data

Linux Encryption Method -bit salt to mod DES into one way hash zero value repeated 25 times output translated into 11 character sequence