PASSWORD SCHEME FOR UNIX

HASHES

UNIX PASSWORD SCHEME Module 2

SUMMERY

Matched Branch:	0
Partially Matched Branch:	3
Mismatched Branch:	6
Extra Branch:	26
Total Match:	18.18 %
Concept Match:	18.75 %
Link. Phr. Match:	16.67 %
Misconception:	92.31 %
Hierarchy Match:	8.86 %

Grade: 11.55%

is comprised of

use of hashed passwords by

SALTS

Loading

Verification

has the purpose of

illustrated by

depicted by

increases the difficulty of offline dictionary attacks

prevents duplicate passwords from being visible in the password file

password and salt serve as inputs to a hashing algorithm to produce a fixed-length hash code and is designed to be slow to execute in order to thwart attacks

hashed password is then stored, together with a plain text copy of the salt, in the password file for the corresponding user ID

to load a new password into the system, the user selects or is assigned a password, which is is combined with a fixed-length salt value

in older implementations, the salt value is related to the time at which the password is assigned to the user, while newer implementations use a pseudo random or random number

salt and user-supplied password are used as input to the encryption routine, and if the result matches the stored value, the password is accepted

when a user attempts to log on to a UNIX system, the user provides an ID and a password, then the operating system uses the ID to index into the password file and retrieve the plain text salt and the encrypted password

does so by

salt changes the hash values of the same password

difficulty in discovering whether a person has used the same password on two or more systems