Lab 5 Report  
Passwords

# Task 1: Password Files

1. What error did you receive when you tried to view the shadow password file without privilege?

TBD – Permission denied

1. As recorded in item #1 above, you received an error when you tried to view the shadow password file. Why is this error a good thing?

TBD- This error is a security measure to prevent unauthorized access to sensitive password hash data, which could be used for malicious purposes if compromised.

1. What is the hash algorithm that was used to store your password information in the password file?

TBD - SHA-256

1. What salt value was used to generate your stored hash value?

TBD- 7voVz3cj

1. When was your password chosen, as reported by the chage command?

TBD- jul 07,2018

1. In item #5 above you recorded the date/time when your password was selected. Why does the system need to keep track of this information?

TBD - The system monitors this to implement rules like password expiration, which can improve security by necessitating regular password changes.

# Task 2: Dictionary Attacks

1. By examining the passwords in htpasswd-sha1, which users have the same password?

TBD - Alice and Dave have the same password which is awesome.

1. List the usernames and passwords of any accounts in htpasswd-sha1 that were cracked when using tinylist.txt as the dictionary.

TBD-

Username - password

Alice – awesome

Dave - awesome

1. List the usernames and passwords of any accounts in htpasswd-sha1 that were cracked when using biglist.txt as the dictionary.

TBD-

Username - Password

Carol – 2cute4u

Alice – awesome

Dave – awesome

Bob – passweod1

Eve – zaq12wsx

1. Record the displayed statistics when you performed a dictionary attack using biglist.txt as the dictionary.
   1. Number of words tried: TBD - 2198690
   2. Number of passwords found: TBD - 5
   3. Number of seconds: TBD – 5.258
2. When a dictionary attack fails to crack one or more passwords (as was the case in the above cracking attempts), what can be said about those passwords?

TBD – those are strong passwords.

# Task 3: Considering Execution Times

1. Record the displayed statistics when you performed a dictionary attack using biglist.txt on htpasswd-md5.
   1. Number of words tried: TBD- 2198690
   2. Number of passwords found: TBD – 5
   3. Number of seconds: TBD – 5.224
2. Item #12 recorded the time it took to perform a dictionary attack on a file of MD5 digests. Assume there is a hash algorithm called APR1 that is simply 1000 iterations of MD5. If this password file indicated that APR1 had been used instead of MD5, approximately how many seconds would it have taken? Show your work.

TBD - If APR1 takes 100 times longer than MD5, it would take 5.224 seconds \* 100 = 522.4 seconds.

1. Item #12 recorded the time it took to perform a dictionary attack on a file of MD5 digests. If this password file also contained salt values that were used in the creation of the MD5 digests (i.e., it contained the following: username, salt, digest), roughly how many seconds would it have taken? Why?

TBD - Pre-computed dictionary attacks are far less successful when salt values are used since they guarantee that hashes for even identical passwords are distinct. Because each password hash needs to be attacked separately, the processing time would increase dramatically.

1. Record the output data when using biglist.txt on htpasswd-sha512.
   1. Number of words tried: TBD- 2198690
   2. Number of passwords found: TBD – 5
   3. Number of seconds: TBD -6.028
2. Referring to the times recorded in #12 and #15, if a system was using MD5 as the hash function for storing password information, but then switched to SHA512, by what percentage would it slow down a dictionary attack (or a brute force attack)? Show your work.

TBD - The percentage slow down = ((6.028 - 5.224) / 5.224) \* 100 = 15.4% slowdown.

1. Review the results of the spreadsheet when 10,000,000,000 passwords/sec was entered. From the point of view of a computer security officer, what conclusions or observations can be made?

TBD - This information would highlight the impossibility of brute-forcing vital hash functions, such as SHA-512, at a practicable rate and emphasize how crucial it is to use robust hash algorithms when storing passwords.

1. Record the output data when pre-hashed passwords are used to crack htpasswd-sha1.
   1. Number of words tried: TBD- 3693
   2. Number of passwords found: TBD - 5
   3. Number of seconds: TBD – 0.103
2. Explain why crackPre.py did not try all the words in the dictionary.

TBD - Once it had discovered every password, the script probably terminated, or it may have streamlined the process by omitting known to be unsuccessful hashes.

# Task 4: Personal Experimentation

1. Record your observations and conclusions from your personal experimentation.

TBD – we need a bigger list and stronger cracker

1. What did you learn from this lab exercise?

TBD – how to do your brut force and set it up.

1. How could this lab exercise be improved?

TBD – teach how to attack how to make a script in any language and how to modify it to make it suitable for particular tasks.

Here are all the supporting screen short.































