Files Included

SHA-512 Files:

**MeetingRoomClientNew.java** – set to work with hashed passwords for login and reservations

**MeetingRoomServerNew.java** – set to work with hashed passwords for login and reservations

**PasswordHashing.java** – Hashes the LoginCredts.txt passwords with SHA-512 and writes them to HashedPasswords.txt

**PasswordHashingSalt.java** – Hashes the LoginCredts.txt passwords with SHA-512 and salt, then writes them to HashedPasswords.txt

**Top1MilPasswordHash.java** – Hashes the dictionary password list with SHA-512

**Top1MilPasswordHashandSort.java** – Hashes the dictionary password list with SHA-512 and then sorts the hashed passwords

**HashedPasswordChecker.java** – Matches the hashed dictionary passwords with the hashed passwords used for the logins

**BinaryPasswordSearch.java** – Searches for matching hashed passwords with the hashed dictionary passwords using a binary search after they have been sorted

**DictionarySaltSearch.java** – Uses the salt for each of the user passwords and hashes every dictionary password with that hash looking for a result, repeats for each salt

**LoginCreds.txt** – contains the username and password for approval to make reservations

**PrivateKey.txt** – for protocol exchange

**PublicKey.txt** – for protocol exchange

**HashedPasswords.txt** – SHA-512 hashed passwords and usernames with the hashed passwords

**MeetingTimes.txt** – for reservation process

**HashedandSalted.txt** – passwords that have been hashed and salted

**top-1million-password-list.txt** – password list

**top-1million-password-list-hashed-txt** – SHA-512 hashed words

**top-1million-password-list-hashed-sorted.txt** – Hashed and then sorted hashes for the hashed passwords

Table Comparing Times

|  |  |  |
| --- | --- | --- |
| SHA-512 | No Salt | Salt |
| Hashing Time | 44 milliseconds | 87 milliseconds |
| Dictionary Attack | 17 milliseconds | 92898 milliseconds |
| Binary Dictionary Attack | 16 milliseconds | N/A |

Matched Passwords from the list:

user10:gandalf

user12:135790

user19:chester

user18:qwerty

user43:admin10

user45:password1!

gunnar:student

user49:popcorn6

user37:chester12

admin:password

user38:password9

user23:password6

user5:123456

user29:dozen12

Total time taken to search for the 50 passwords: 15 milliseconds

Number of matched passwords: 14

Salted Dictionary Attack:  
Match found: admin - password

Match found: gunnar - student

Match found: user5 - 123456

Match found: user10 - gandalf

Match found: user12 - 135790

Match found: user18 - qwerty

Match found: user19 – chester

Match found: user23 - password6

Match found: user29 - dozen12

Match found: user37 - chester12

Match found: user38 - password9

Match found: user43 - admin10

Match found: user45 - password1!

Match found: user49 - popcorn6

Time elapsed: 92898 milliseconds

Total Matches found: 14

It seems that for the most part when adding special characters it makes the password much stronger, the combination of words and numbers and special characters is quite uncommon.

**Bonus Points:  
Files Included:**

**MeetingRoomClientBcrypt.java** – client program for login and reservation, works with bcrypt hashing

**MeetingRoomServerBcrypt.java** – server program for login and reservation, works with bcrypt hashing

**PasswordHashBcrypt.java** – program that hashes the user login passwords using bcrypt

**PasswordHashBcryptDictionary.java** – program that hashes the password files from the 1 million list using the salt that was used for the PasswordHashBcrypt.java passwords stored in HashedPasswords.txt

**BcryptSearch.java** – program to search for matching bcrypt password hashes from the 1 million passwords list and the hashed passwords for logins

**LoginCreds.txt** – contains the username and password for approval to make reservations

**PrivateKey.txt** – for protocol exchange

**PublicKey.txt** – for protocol exchange

**HashedPasswordsBcrypt.txt** – bcrypt hashed passwords

**MeetingTimes.txt** – for reservation process

**top-1million-password-list.txt** – password list

**top-1million-password-list-bcrypt.txt** – bcrypt hashed password list that uses the salt from the bcrypt hashing of the HashedPasswordsBcrypt.txt

(1)

bcrypt hashing of the password list time: approximately 3.93 hours

Dictionary attack search using bcrypt hashed passwords:

(2)

I will not try to use the dictionary attack for this case using salt because if each hashed password from the login credentials had its own unique salt, it would take approximately 50x as long because the entire dictionary password list would have to be hashed for each password in the hashed passwords list. Even using a supercomputer this type of attack on bcrypt hashing with unique salts would take a long time, but with my personal computer this could potentially take as long as 200 hours.