Members of Group:

Fatih ATALAY 192010020008

Ertuğrul KÖKSAL 192010020006

BNC TEST

In our application we tested the processing power specifically. We wrote a program that challenges the device to perform some complex math and then time how long that takes.

First test of us is Secure Hash Algorithm 1, also known as SHA-1, is a type of "cryptographic hash algorithm". Data can be encrypted with this tool and then stored in an array. A challenging method is used to the provided array to determine the position. The objective of the cryptographic function is to encode this data in a difficult-to-decode manner.

MD5 is our other test. This is another significantly flawed cryptographic hash method that is even less safe than SHA-1.

Brute Force is our final test. This implies that the computer makes educated assumptions about each combination before arriving at the right answer. It's referred to as a "brute force attack." Additionally, some hackers will grab your password in this manner. We determined how long it will take to crack the password and assess its strength.

Since this is a benchmark test app, we want to know exactly how long the tests takes to complete. The best approach to achieve this is to use “System.nanoTime” to obtain a timestamp. Although it doesn't provide us with the precise time, it gives us access to the internal clock that is the most accurate and runs in nanoseconds. This 'clock' doesn't reflect actual time. It is still possible to use it to measure time, which is what we're attempting to do in this instance. Total time is the time at the end of the function, minus ‘time started’.

We specifically timed how long it would take to encrypt a short string 20,000 times in nanoseconds. Our test string was MD5 encrypted by the “MessageDigest” class. This just involved taking time (tsLong = System.nanoTime()), repeating “computeSHAHash” 20,000 times, taking the new time, and figuring out the difference before showing it in the widget titled "result." By dividing the total number of nanoseconds by 100,000,000 and rounding to the nearest whole number, we also calculated a "score.". We applied this to all tests we did and then we sum the scores we got. Using this value, we subsequently created the total score.

In this application we also used the handlers. Because of the handlers are used to control several threads and enable communication between them and the rest of the activity. The time before and after the loop is executed are simply obtained, and the producing variable is then changed into a message that is sent to the handler.

Same result pictures

 