ATTACK

Here is the procedure used to deduce a password with a timing attack:

For each index of the hidden password, randomly select a lowercase letter as a guess and sample the time it takes for the password check to return. Do this for many iterations so that variances may be averaged out. Each letter will have a running total of the number of samples, sum of the times, and the sum of the squares of the times.

Once all the samples have been taken for the index, calculate the confidence interval for all 26 possible different characters by using the running totals for that character. If a character's interval has no overlap with all other intervals, we can say with high probability that the character at this index made the password check take longer and thus is closer to the true password. We then fix that character at that index and proceed to the next index. If all intervals overlap, the number of iterations needs to be increased to shrink the width of the intervals.

Once this is done for all indices, the check_password function will return true indicating that the correct password has been found.

ASIDE: Guesses are sampled in random order instead of sampling the same guess back-to-back because of compiler optimizations. These optimizations can skew the data so they must be avoided. By making the next sampled guess random, the compiler is unable to deduce any patterns and thus cannot make any optimizations.

COMPILATION

Source code is in main.cpp and may be compiled with "g++ main.cpp". Then run "./a.out". The default password that is checked is called "password" and may be changed if needed.

STATISTICS

The statistics for the runtime of password_check for each letter of each index for the password "password" is in the file called stats.txt.