Homework: Timing Attack

In this assignment, you are tasked with thinking like an attacker. To that end, you've come to understand there is a web service that is vulnerable to a timing attack. This implementation is *particularly obvious* and should be relatively easy to break.

Through other means you have learned three vital pieces of information to assist in gaining access to the system.

1. The alphabet of possible characters is only 0 through 9. Thus a password may be "1234" or "543224" or any other combination of these digits.
2. The entropy of the password is somewhere between 35 and 45 bits. Using what you know about entropy, determining a reasonable range of password lengths is possible.
3. The format of the message is a POST with a "pwd" parameter.
    a. A cURL versions to do this would look something like this:
    `curl -X POST https://qrxjmztf2h.execute-api.us-west-2.amazonaws.com/prod -d '{"pwd":"1111"}' -v`
    b. If the body is in the wrong format we'll get a 422 returned
    c. If the pwd is wrong, we'll get a 401
    d. If the pwd is right, we'll get a 200
4. A *test* endpoint that behaves exactly the same way as the puzzle can be found at URL: with a *correct* password of "42answersall"
    a. Consider using this to test good/bad letters to see the response time
    b. Consider running many tests per letter (in my solution, I used a threadpool to make ~10 requests simultaneously to gather timing information)
    c. Test URL:
    `curl -X POST https://qrxjmztf2h.execute-api.us-west-2.amazonaws.com/prod/example -d '{"pwd": "42answersall"}' -v`

Recommendations of approach:

- You might sketch out your approach knowing what you do about timing attacks prior to attacking the system.
- You might consider gathering some metrics about what time differences look like assuming good and bad responses.
- You might consider trying to determine the key length first.

HOST: https://qrxjmztf2h.execute-api.us-west-2.amazonaws.com/prod
DEMO: https://qrxjmztf2h.execute-api.us-west-2.amazonaws.com/prod/example