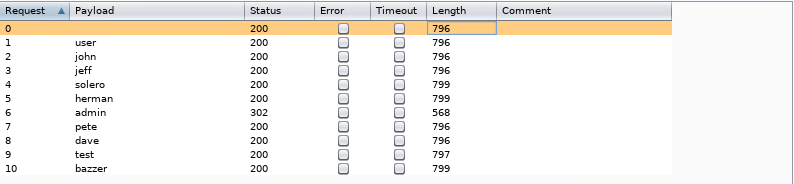
Lab 06 - Attacking Authentication

10 Points

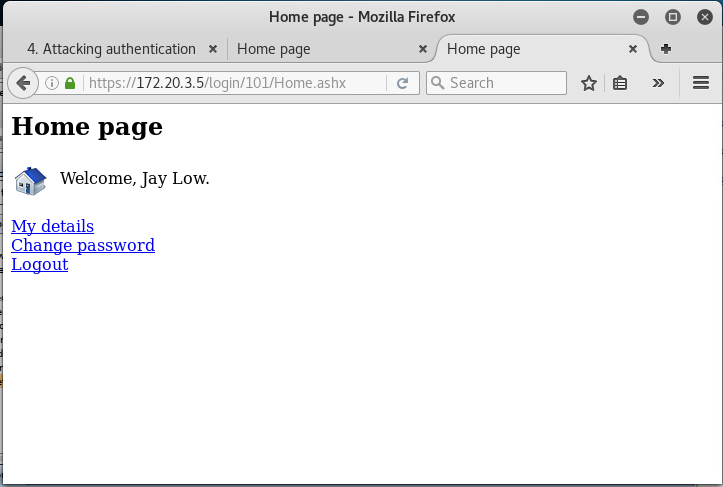
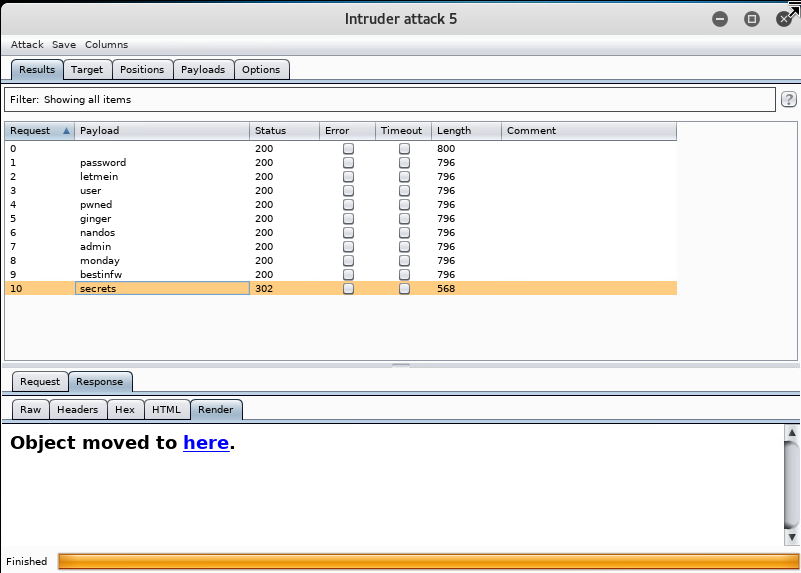
Authentication is the cornerstone for most applications to protect or partition resources. In this lab, we’ll be looking at ways to attack this process. As with the last lab, these can be challenging - start early and let me know if you run into any issues!

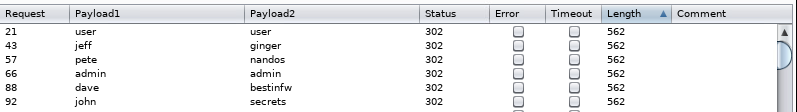
Complete the following tasks in **Lab 4. Attacking Authentication** on Pablo:

1. **[2 points]** What are valid usernames? What is a valid password? What were you able to log into the system with?  
   valid usernames:

**User, John, Jeff, Pete, Dave**

**I got in with:  
User: john  
Pass: secrets**

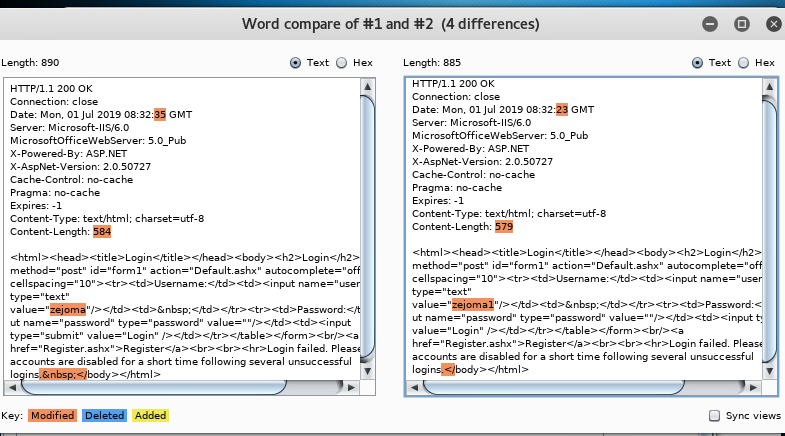


1. **[1 points]** How many more attacks were required? Screenshot showing all valid usernames and passwords  
   It tested 100 combinations, The last hit was on #92 
2. **[7 points]** Complete tasks **a - g.**

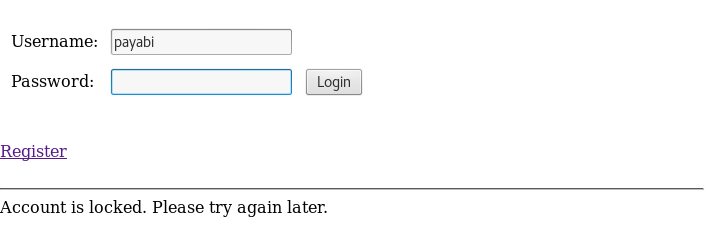
**Deliverable:**

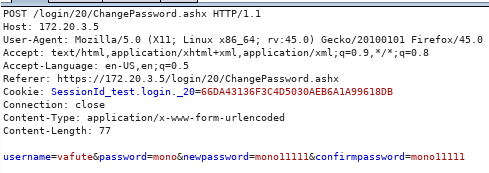
Turn in a Word Document or PDF that contains screenshots for each task as well as an explanation of what you are doing.

**Hints**

1. Username Enumeration
   1. Use the source!   
      Here I am comparing the source response to two login attempts with what would seem to be identical notices of failure… one has a valid username and one does not, however neither have a valid password (for username enumeration)  
      valid usernames have a whitespace character at the end of the notification, making this identiable by looking at difference in response content length
   2. Look at response headers…  
      **Valid username:**  
      https://gyazo.com/7d0dbbedcdbc1906d1dbeb49bb9d20e5.png  
      **Invalid username**:  
      https://gyazo.com/08fee9e29135d9a267cb43ddebb435ed.png

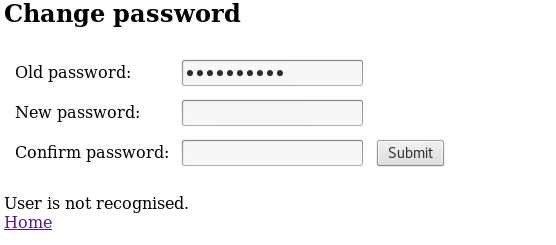
In this one we do the same thing, but the responses are much more obvious…

* 1. When does an account lock?  
     Lockout after like 5-6 failed attempts, and only if the username is valid… we could automate a clusterbomb, with 6 junk passwords… any lockout responses are valid usernames  
     
  2. Change your password...

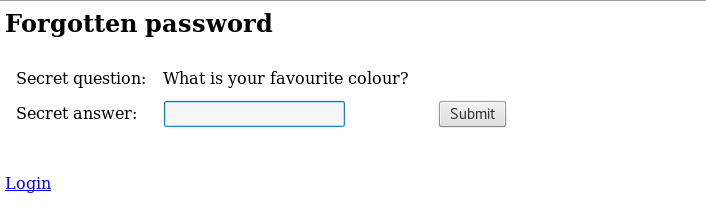
here the vulnerability lies in the server sending the uername as a variable in the header…  
  


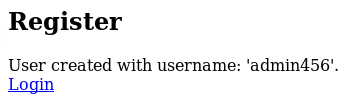
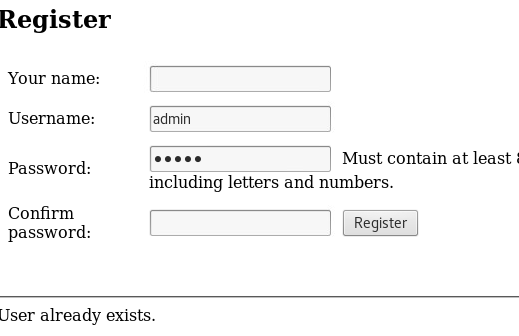
Valid username, invalid ‘old password’

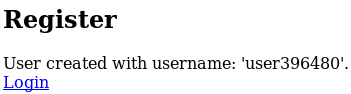


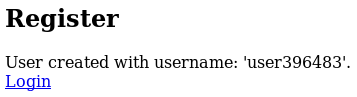
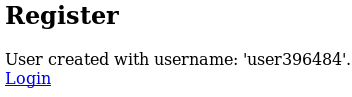
Invalid username invalid ‘old password’:  


* 1. Did you forget your password? When would a link show-up?

A link for “did you forget your password shows up (here at least) on any failed login, whether the uername is correct or incorrect…  
however… from there it asks for your username. On valid username you get the security question:  
  
  
on an invalid username you get  


* 1. Did you register for an account already taken?  
     When registering, you are allowed to choose your username…  
     if the username is not currently being used you get:  
       
     if the username is taken you get :  
       
     **USER ALREADY EXISTS**
  2. Is there a pattern?

Created a number of users: landing me on this page with these assigned usernames:  


The usernames are simply adding 1 to the appended number.