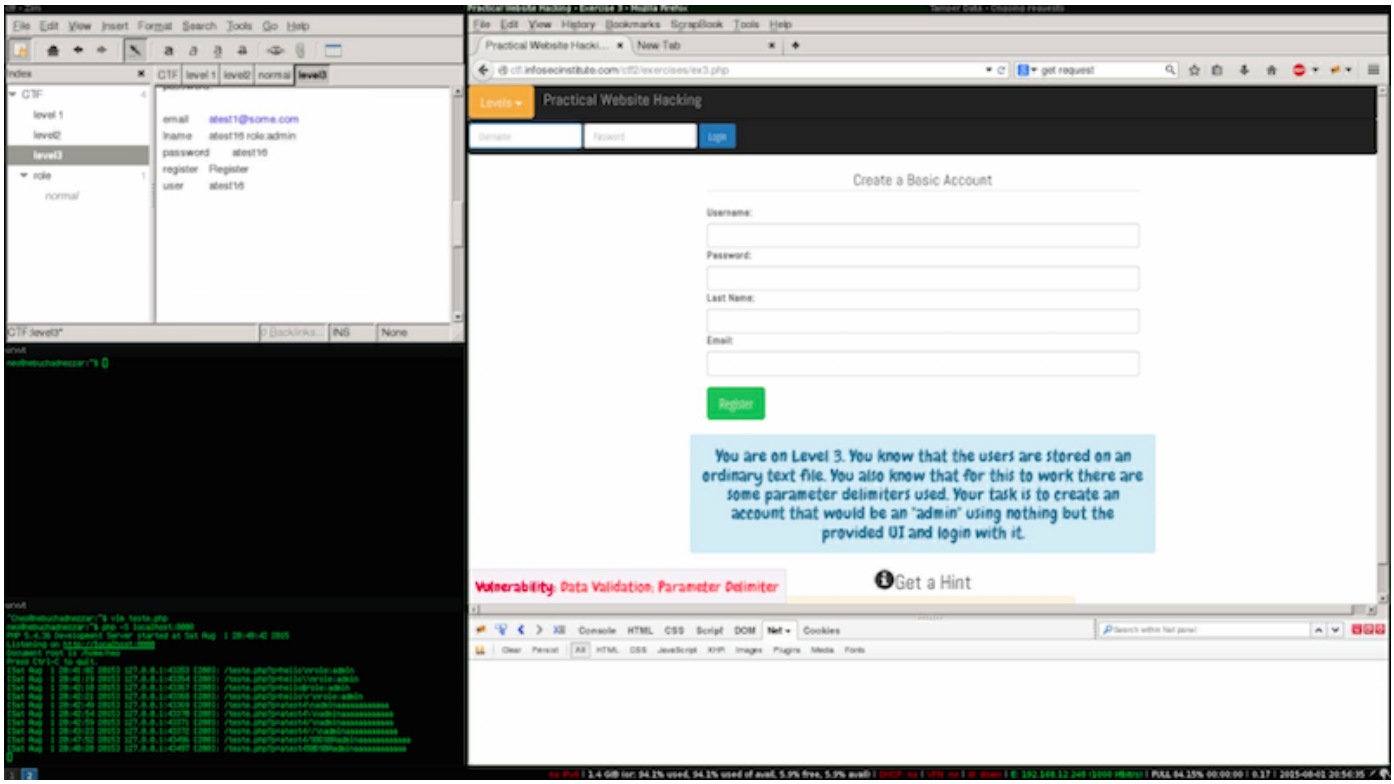
Infosecinstitute CTF 2 - LEVEL 3

This will be solution for Level 3 Practical Web Hacking CTF #2.

In this level we are told to attept a privilege escalation, the objective is to register a user with an ADMIN role, and exploit a Data Validation.

We are told that the information is saved in text file, from this information we can assume that the several fields are some home separated by different chars souch as $ # etc.

We start by looking at the HTML document, and search for any hidden fields that might represent the role, since nothing useful was found i proceeded to create the regular user this allowed me to gather information about how the privileges are represented in the text file.



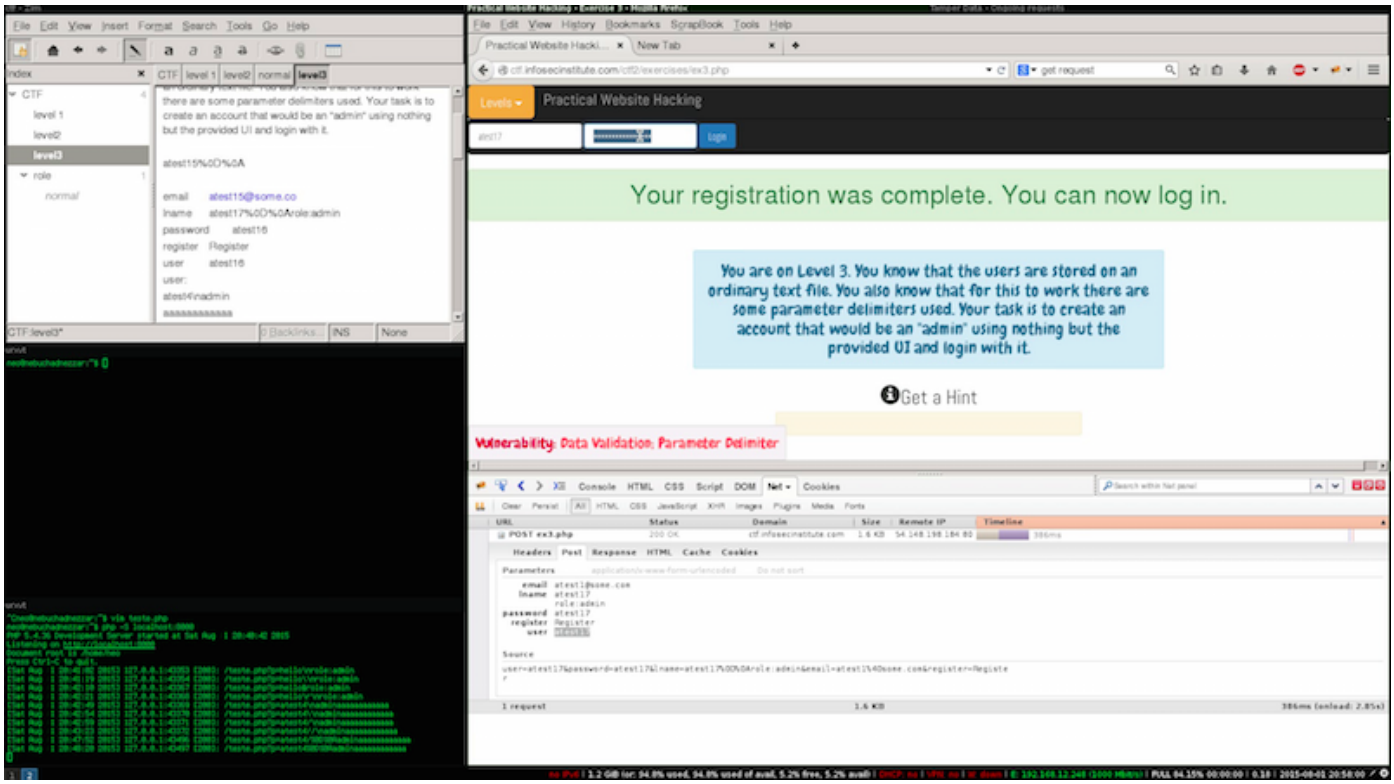
Their are many ways to implement this functionality the ones i suspected were:

1. The field and Value separated by **":"** following the same logic used in the page after login.
2. Use a different char to separate both.

The second attempt would probably less likely as it could lead changes in the order that they are supposed to be saved, for this reason i tried the first option.

Tasks

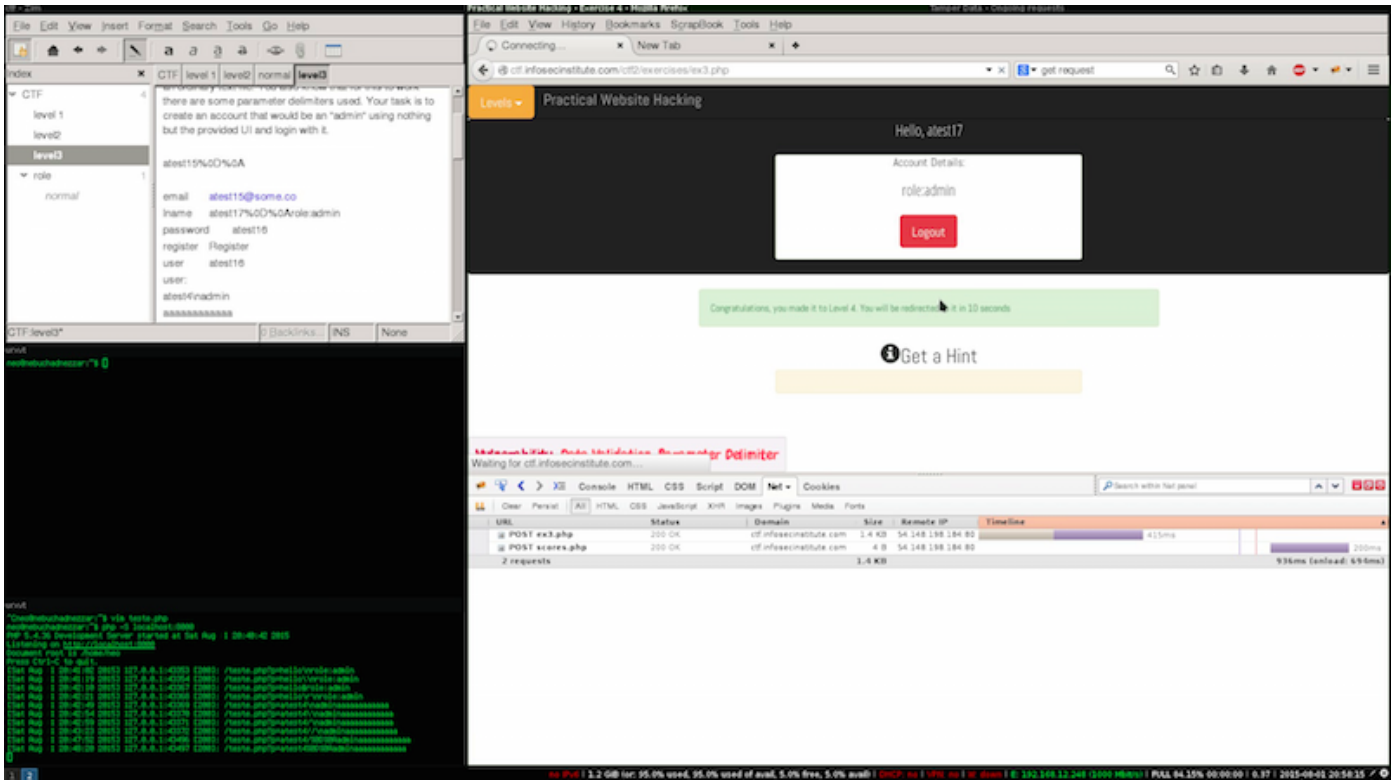
1. Find the field that is written before **role**.
2. Inject a new line char followed by the admin role value.



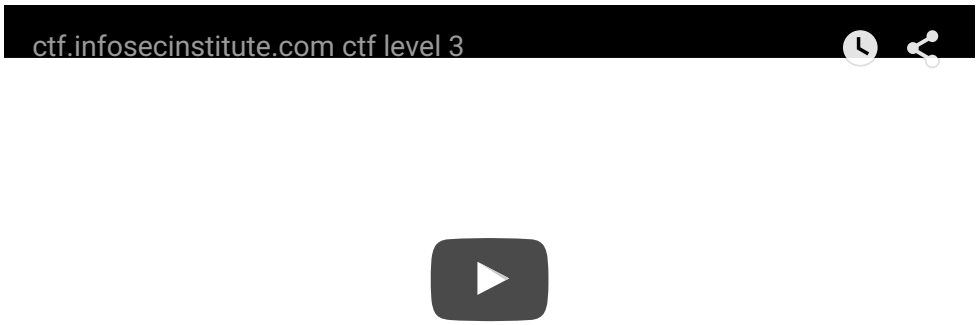
Injecting new lines from Http request parameters to php is can't be done directly, for this to work i need to find a way to send the "value" in a way that php could see it as a new line. and the answer was %0D%0A this will be automatically translated to newline followed by role:admin.

Now that we know what to send to the server we need to find out where to place it, this process requires a careful examination of the registration form, where i could verify that username field couldn't be used not only due to size limitations but because if the role was to be located between the username and the password by injection our payload in the text file, would either break the script or the password would be admin or empty.

The only fields that are not showed\used are the lastname and e-mail address, in order to exploit the registration from we can use Tamper data to change the values before they are sent to server this ensures that the chares are sent without being encoded.



Video



0x4E0x650x6F
0x4E0x650x6F