**PICOCTF 2017**

**Cryptography**

**Keyz –** ssh login using keys

If not using password then the server requires the user’s public key saved within their .ssh directory. The user then only needs to login remotely by typing: ssh [username@server\_addr](mailto:username@server_addr). The user will be automatically be authorized to that server since the server has the user’s public key which it uses to combine it with the user’s private key to authenticate the machine.

>> Firstly generate new ssh public key in your local machine. This will be saved under /root/.ssh/. Type the following:

$ ssh-keygen -t rsa -C "[your\_email@example.com](mailto:your_email@example.com)"

>> Now within the picoctf web-shell create a text file called ‘authorized\_keys’ and change the permission to : chmod 600. Copy your local machine public key and paste it inside this file.

>> Now from your local machine type the following and find the flag:

$ ssh [username@shell2017.picoctf.com](mailto:username@shell2017.picoctf.com)

ComputeAES

Encrypted with AES in ECB mode. All values base64 encoded

ciphertext = V3Vqirostg6qW26sle5mnyrwEYSrteN6oHkilO50e9dFkN+0JhC3yu0LcQNw/hXU

key = r7y1dhmTvjQrcra7A1UQFw==

**Master Challenge**

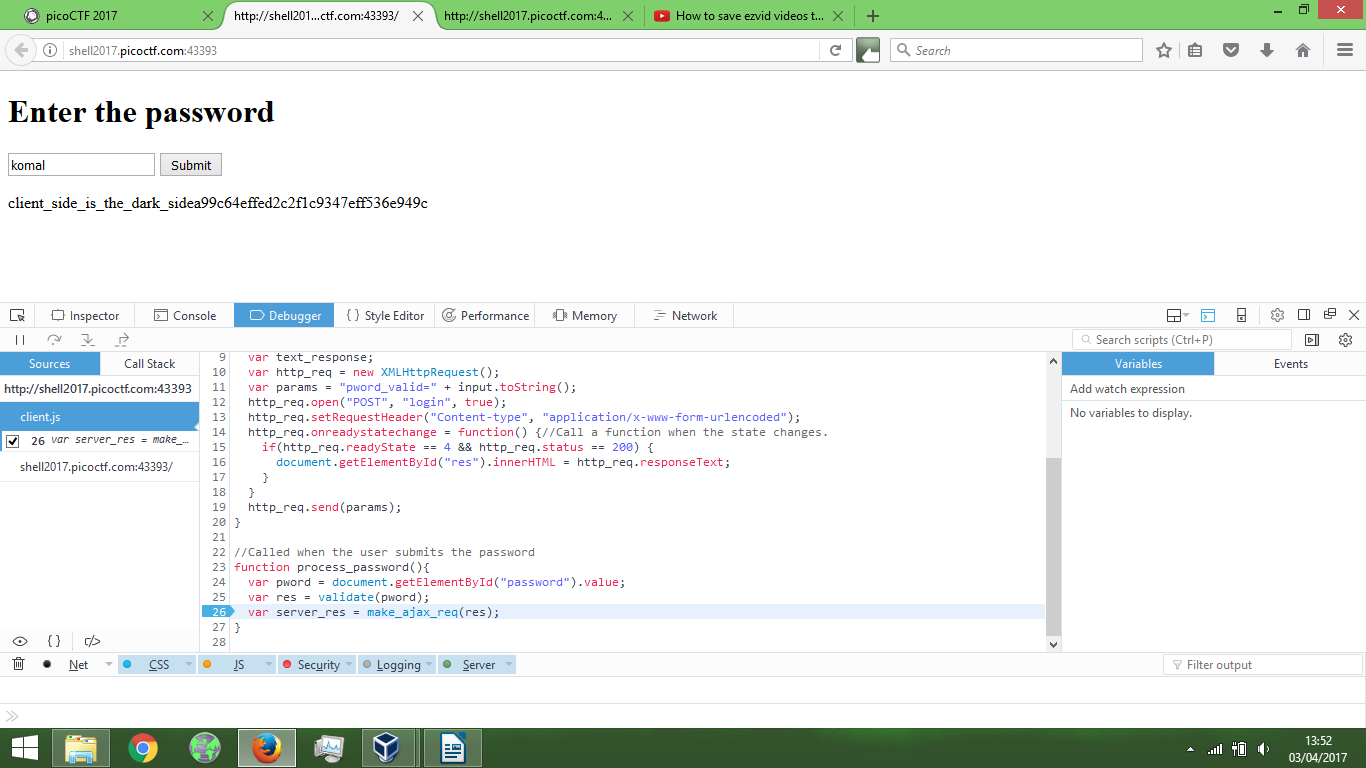
**Level 1 –** Bypass client side authentication

>> Right click on the source code page to view the JavaScript password process functions.

>> Go back to the main website and hit f12 key to inspect under the Debugger mode.

>> Add a breakpoint under the process\_password() function where it says

var server\_res = make\_ajax\_req(res);



>> Now enter any random characters in user field and hit ‘submit’. On the right hand side of the debugger window you will see the values passed to the variables. Over there change ‘res:false’ to ‘res:true’. Click on play button on the left hand side of the debugger window. You have now bypassed the client side authentication

