Chad Ballay

CYBR350-342N

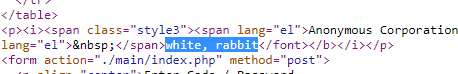
Week 7

08/16/2020

OWASP Ecuador Chapter

# Challenge 001

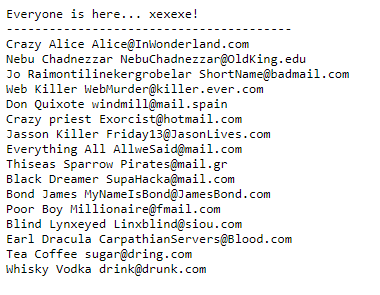
This looked weird. Tried it for the username/password and it worked.



Since the goal mentioned emails the first thing I did was click on Mailbox Special Clients’ Mailbox link. Once there I didn’t see much until I noticed the url for the image had “secret\_area” as part of its path.



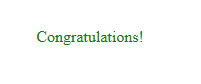
On a hunch I started trying various extensions and was able to load mails.txt



The scenerio’s flavor text points to Friday the 13th for a hire date. This is a big assumption but I think this is the email address.

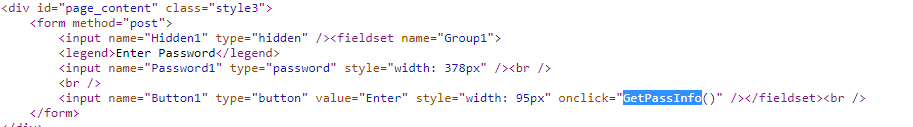


Tested out sending an email to that address and got the Congratulations screenshot.

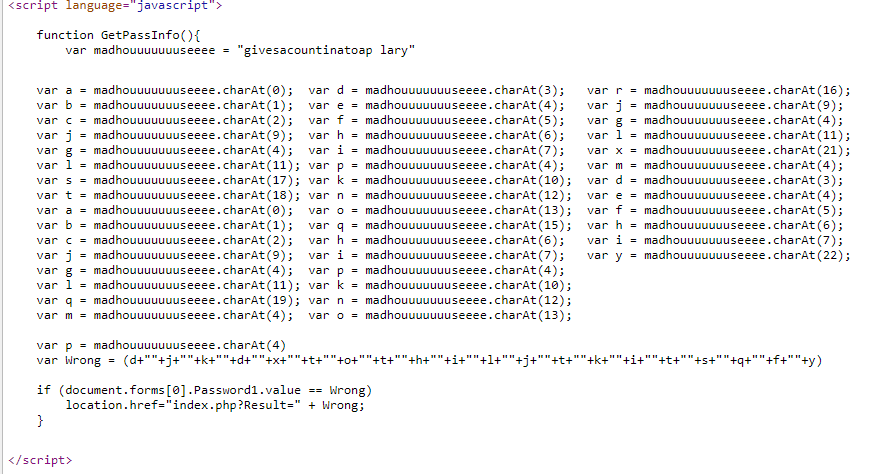


# Challenge 002

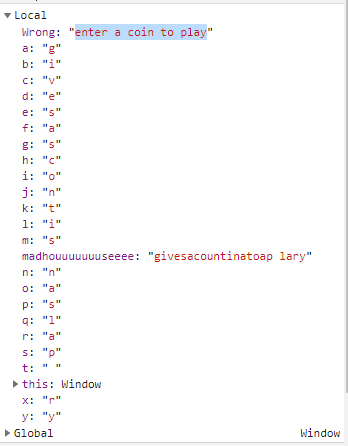
This one was odd in that it only required a password. No userid. Looking at the page’s code it loads a function for validation.



Reading the source of the page further down shows the contents of the function. The code is a mess of obfuscation so I’m trying to figure it out by setting a breakpoint and walking through it.



Setting a breakpoint got this in the execution.

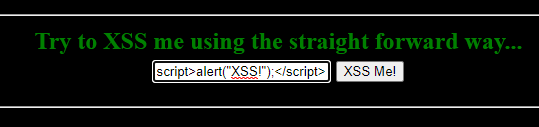


Keyed that in and loaded the congratulations page.

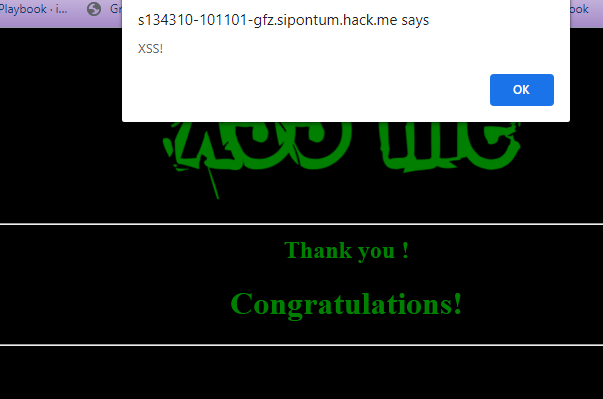


# Challenge 003

This one ended up being really simple. Put in a script tag with an alert function call.

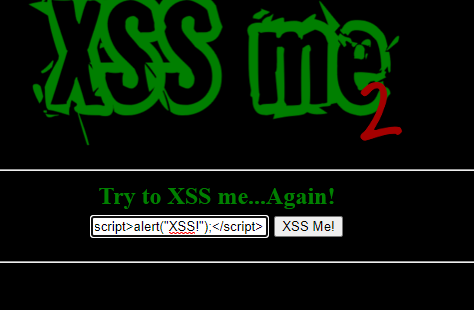


Yielded the success and got the congratulations.

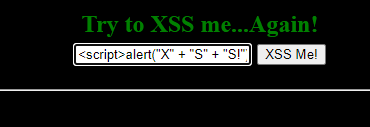


# Challenge 004

Variation of Challenge 003. Tried to reuse the same script tag and alert function but didn’t work.



Figured if they are looking for specific strings maybe I could play around with it. Tried concatenating characters instead of one string.

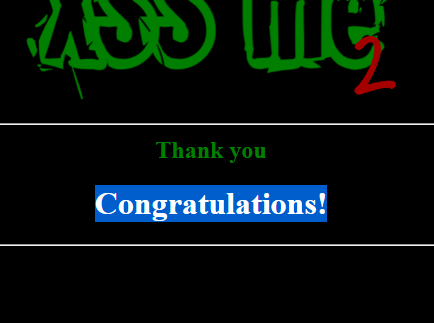


That didn’t work so I tried referencing the letters by their unicode number instead of directly.

<script>alert(String.fromCharCode(88,83,83,33))</script>

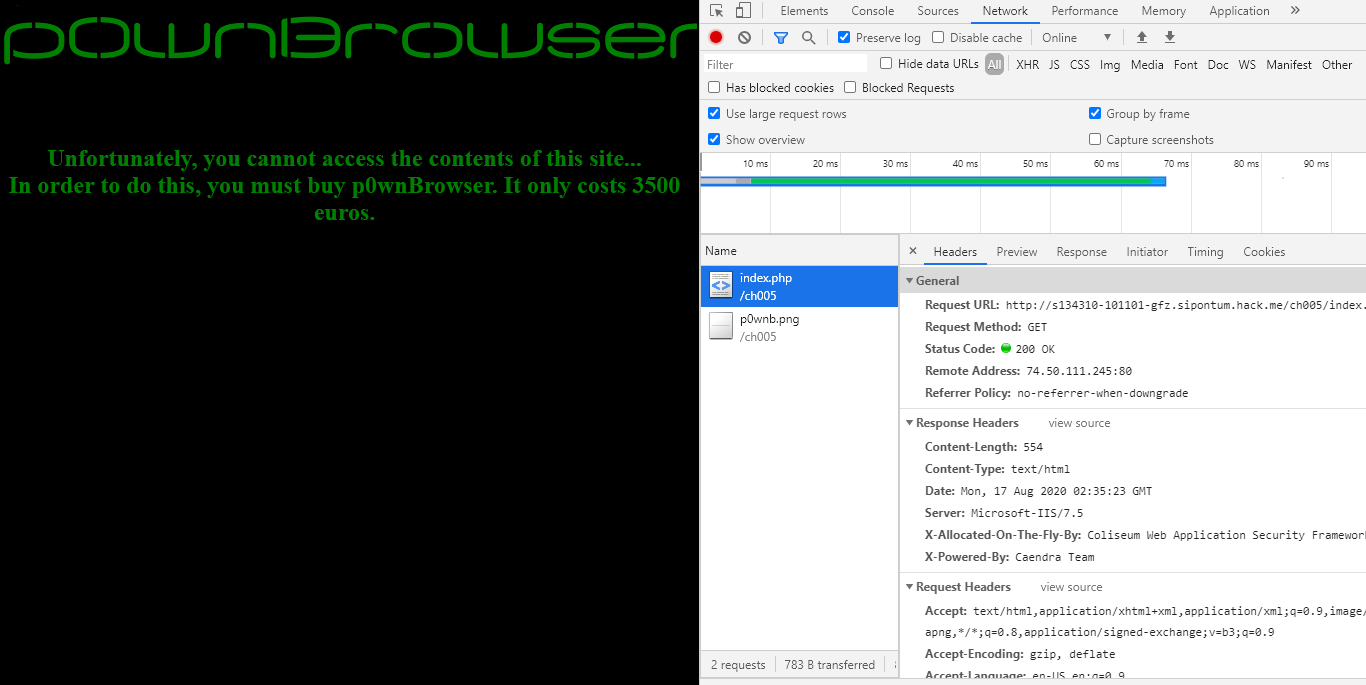


And it worked.

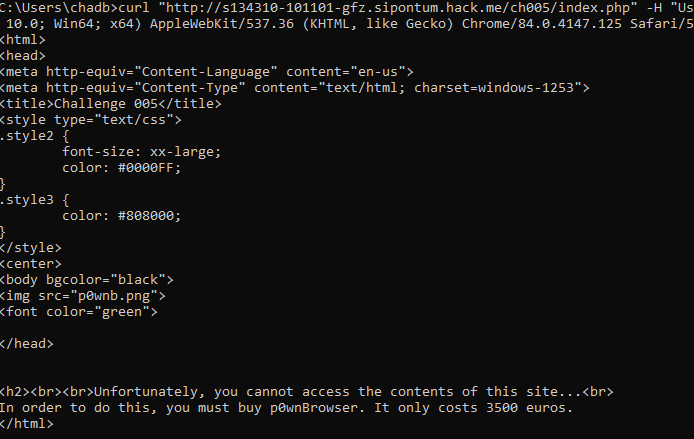


# Challenge 005

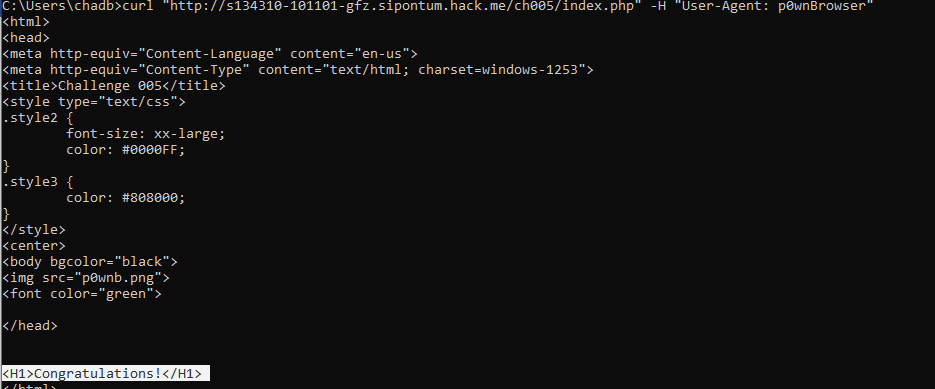
Loaded the page and got the boilerplate message about needing the right browser.



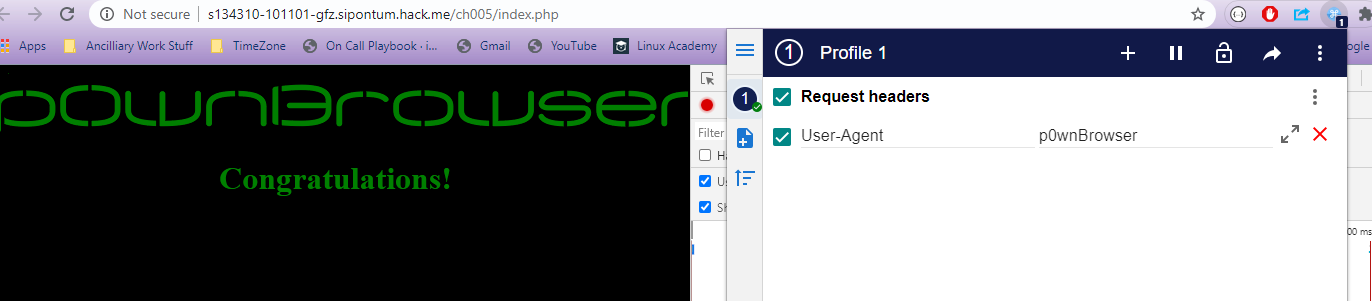
Started wondering how it would know what browser I was using? Pulled the request and started stripping out what I could.



Still no dice. Decided to just set the User-Agent to p0wnBrowser and it worked.



Used a Chrome plugin to do it in the browser to be sure.

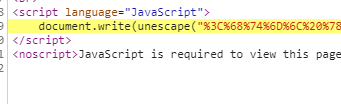


# Challenge 006

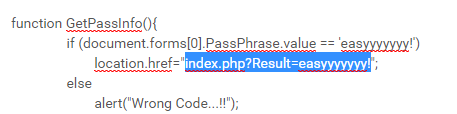
Loaded the page and my first question was what was up with the capitalization and punctuation errors.



Looking at the source there was a big long URL encoded string.



Plugged that into the first google utiltiy I could find. This yielded some really obfuscated code. Reading through that I found this snippet.

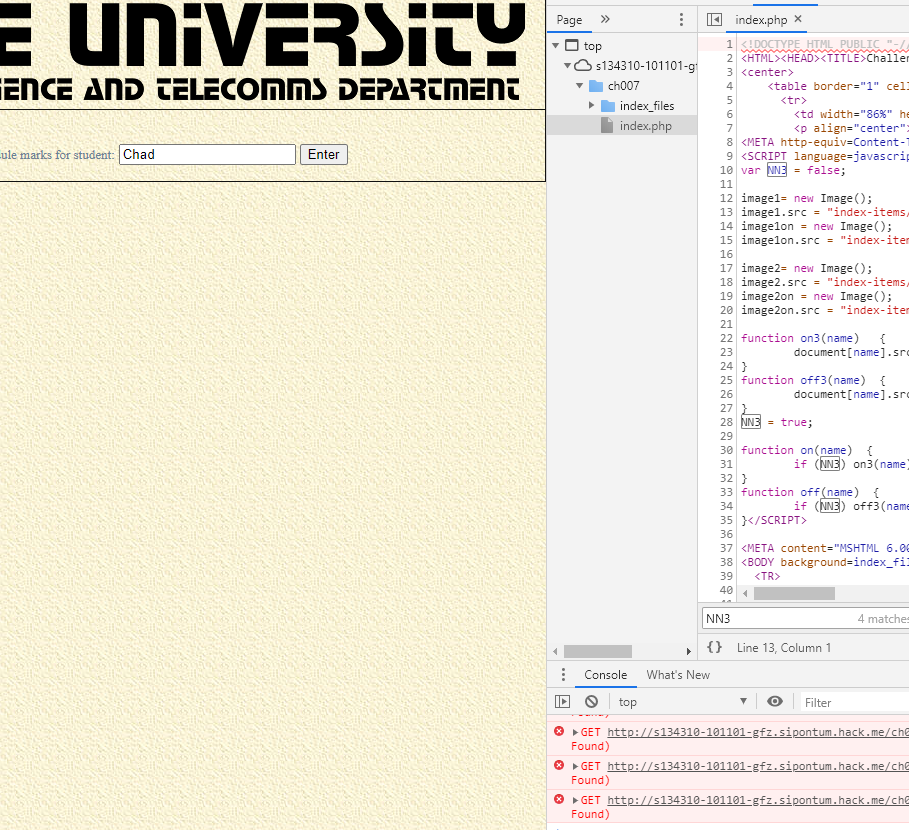


Loaded that URL and got the congratulations.



# Challenge 007

I can see where there is all sorts of oddness around some javascript in the code. I can also see where the image URL’s are returning 404’s.

I can see where there are directories that have content being served out but I can’t do a directory listing. I can't figure out how to proceed on that though.

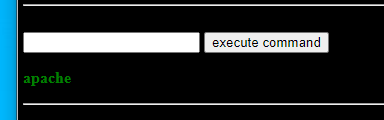


# Challenge 008

Loaded page and ran ‘Help’ like recommended.

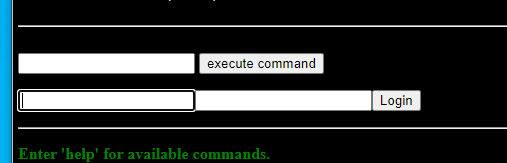


Whoami and Id let me know that we were running as Apache.

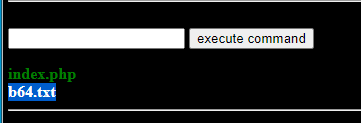




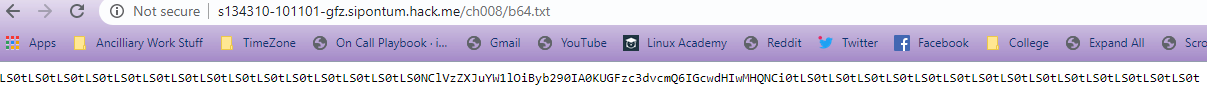
Su didn’t really help since I don’t have the password for root.



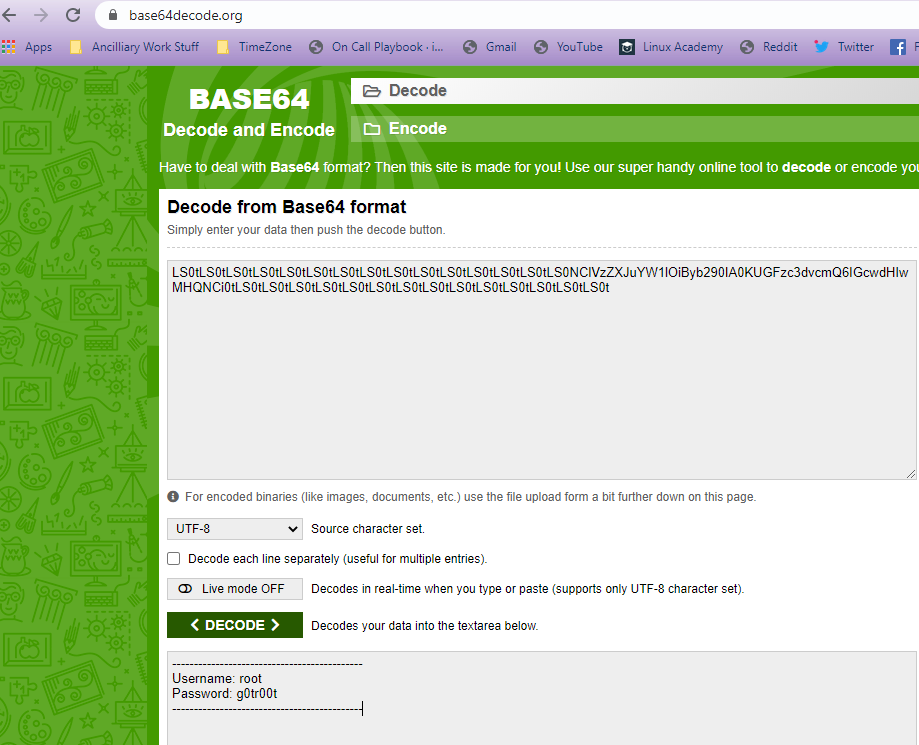
Ls showed that there was some other file in the directory.



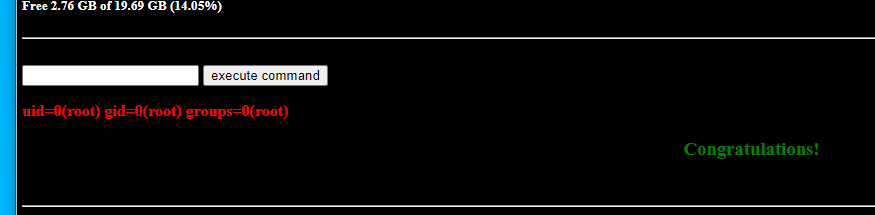
Loaded the file and got a string of text. I’m assuming it is base 64.



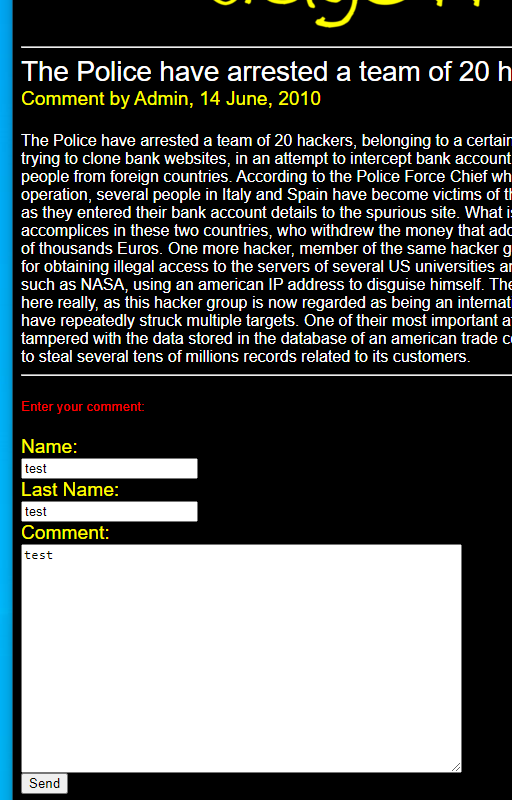
Which then yielded this.



Tried su again but using those credentials.

Challenge 009

Loaded the page and tried to send dummy data.



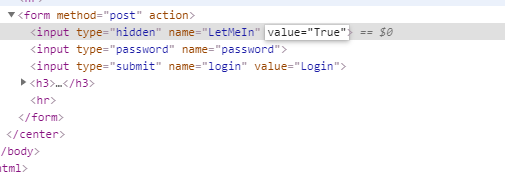
Gave this URL as a response.



No idea after this point. I’ve tried various names/text/page arguments, and nothing is jumping out at me.

# Challenge 010

Loaded the page and looking at the source code saw that a hidden value was set to False. Changed it to True.



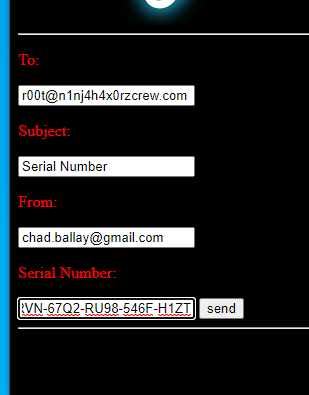
Got a string of URL encoded text.

%53%65%72%69%61%6C%20%4E%75%6D%62%65%72%3A%20%54%52%56%4E%2D%36%37%51%32%2D%52%55%39%38%2D%35%34%36%46%2D%48%31%5A%54

Decoded that into this string.

TRVN-67Q2-RU98-546F-H1ZT

Hitting ok on the alert loads a second page with a form for sending an email.



Hit enter and click through a false alert. Got the congratulations.

