

Assignment 5

Made By: Karim Salem 1001619

1st Attack:

Checked through source code till I reached the path below:

Juice-shop/lib/insecurity.js

master juice-shop / lib /Go to file

bkimmminich Fix further TypeScript issuesbc55299 on 1 FebHistory

..		
startup	Fix further TypeScript issues	3 months ago
accuracy.ts	Fix further TypeScript issues	3 months ago
antiCheat.ts	Fix further TypeScript issues	3 months ago
botUtils.ts	Bump copyright notice to include 2022	4 months ago
insecurity.js	Bump copyright notice to include 2022	4 months ago
logger.ts	Bump copyright notice to include 2022	4 months ago
utils.ts	Fix further TypeScript issues	3 months ago
webhook.ts	Refactor exports into TypeScript syntax	4 months ago

180 lines (160 sloc)6.72 KBRawBlame

```
1  /*
2   * Copyright (c) 2014-2022 Bjoern Kimminich & the OWASP Juice Shop contributors.
3   * SPDX-License-Identifier: MIT
4   */
5
6  /* tslint node: true */
7  const crypto = require('crypto')
8  const expressJwt = require('express-jwt')
9  const jwt = require('jsonwebtoken')
10 const jws = require('jws')
11 const sanitizeHtml = require('sanitize-html')
12 const sanitizeFilename = require('sanitize-filename')
13 const z85 = require('z85')
14 const utils = require('./utils')
15 const fs = require('fs')
16
17 const publicKey = fs.readFileSync('encryptionkeys/jwt.pub', 'utf8')
18 module.exports.publicKey = publicKey
19 const privateKey = '-----BEGIN RSA PRIVATE KEY-----\nMIICXAIBAAQBgQDhwL Ee9wgTxcBc7+RPhdbbBebq3bs4k0P0IGzqlpKv3K1xxd81m28EaM48KUqysIa+ndv3NAn2RxCd5ubVdJ3cK43z06ko8TFEzx/65g
20
21 exports.hash = data => crypto.createHash('md5').update(data).digest('hex')
22 exports.hmac = data => crypto.createHmac('sha256', 'pa4qacea4VK9t9mGv7yZtwmj').update(data).digest('hex')
23
24 exports.cutOffPoisonNullByte = str => {
25   const nullByte = '\x00'
26   if (utils.contains(str, nullByte)) {
27     return str.substring(0, str.indexOf(nullByte))
28   }
29   return str
```

Trying commenting md5

Customer Feedback

Author

***der@juice-sh.op

Comment *

md5

Max. 160 characters

3/160

Rating

CAPTCHA: What is 5-3-2 ?

Result *

0

Submit

Attack succeeded

You successfully solved a challenge: Weird Crypto (Inform the shop about an algorithm or library it should definitely not use the way it does.)

2nd Attack:

Searching till I found file package.json.bak

localhost:3000/ftp

~ / ftp

quarantine

coupons_2013.md.bak

incident-support.kdbx

suspicious_errors.yml

acquisitions.md

eastere.gg

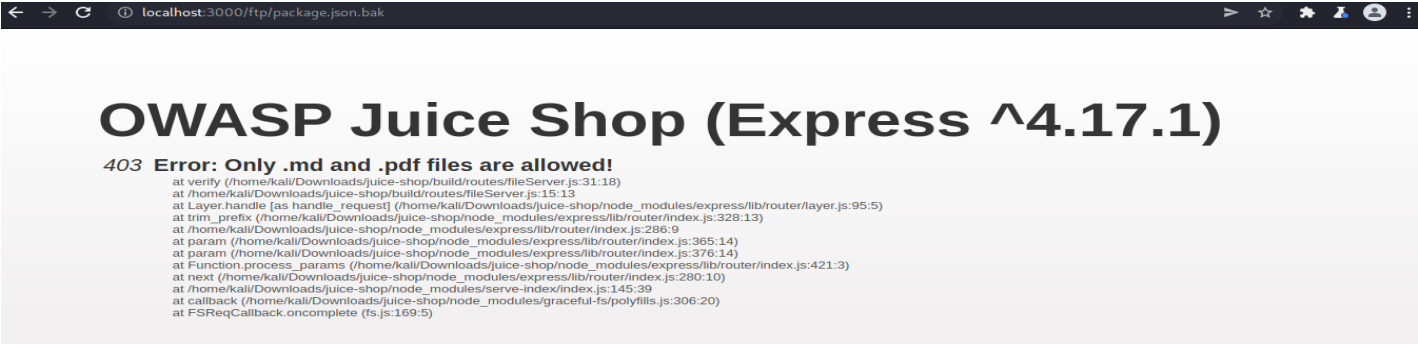
legal.md

announcement_encrypted.md

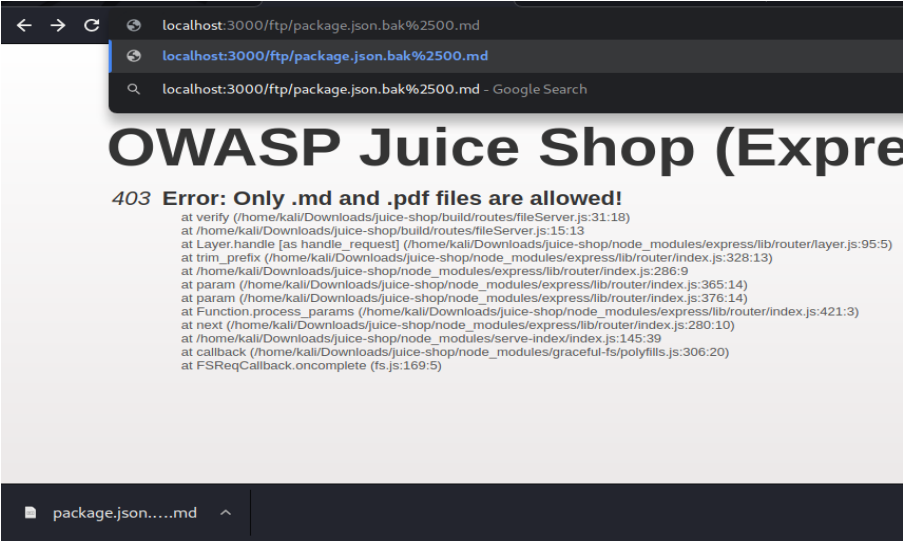
encrypt.pyc

package.json.bak

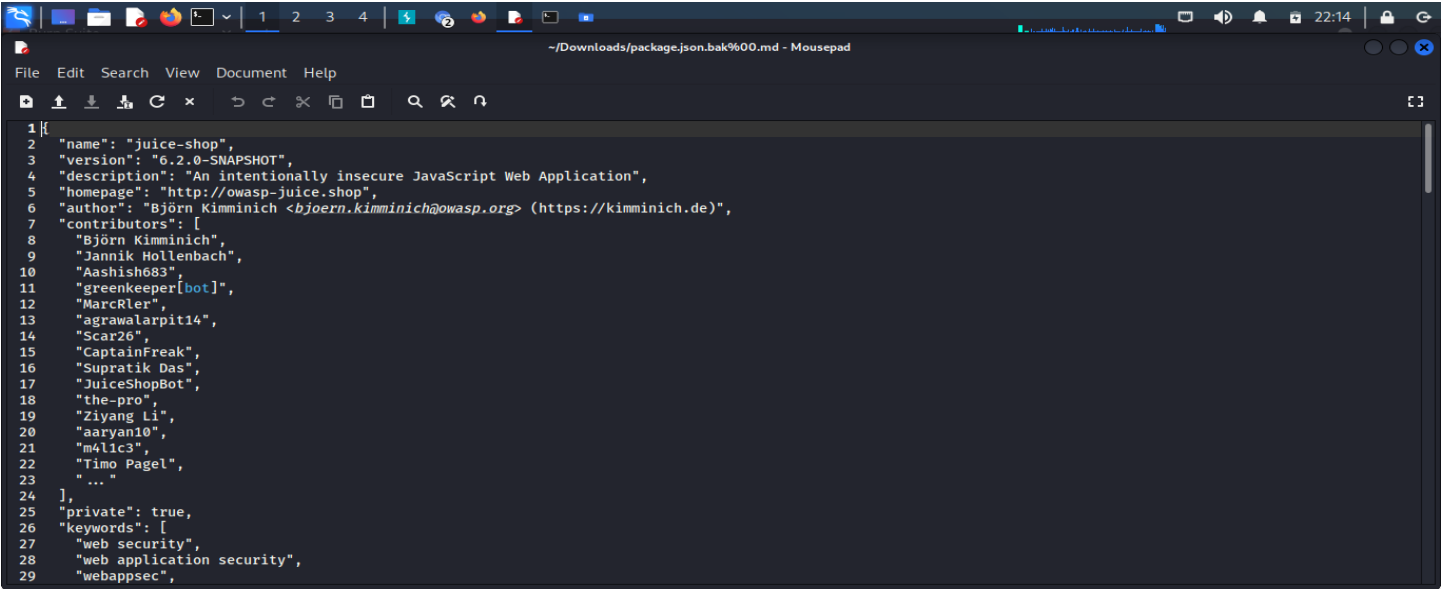
package.json.bak



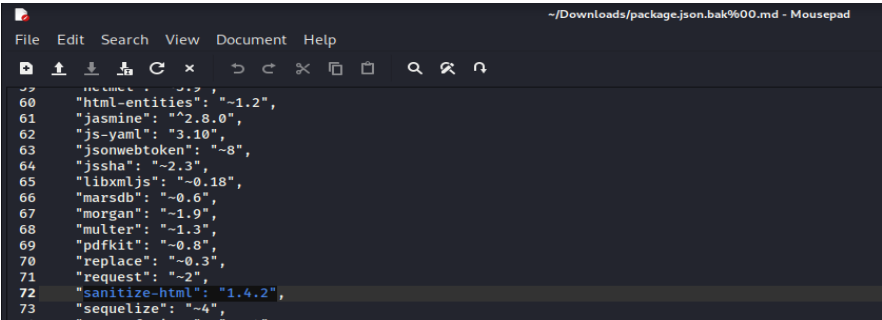
Accessed the file by downloading .md version.



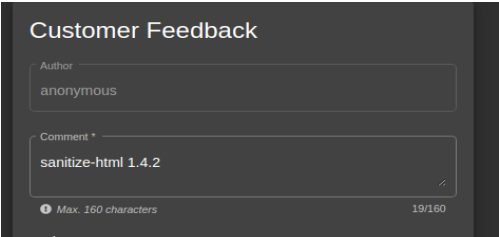
The file contents:



Found librariy's version



Tried adding library with its version

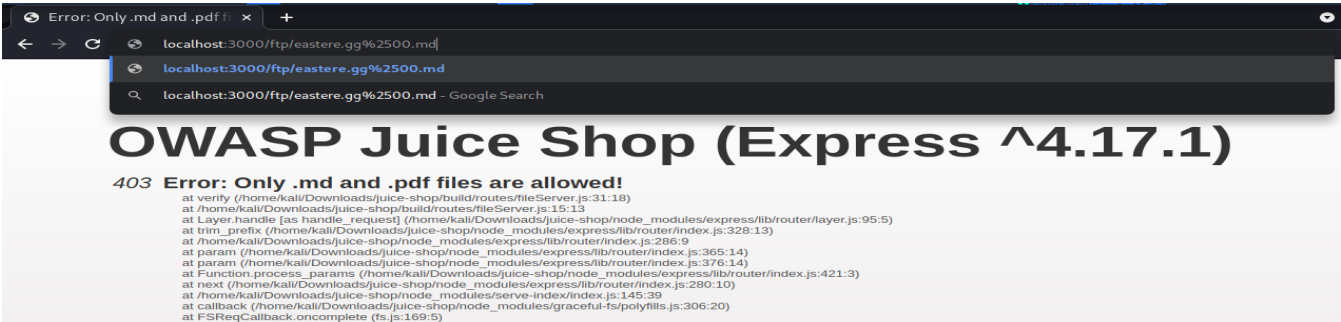


Attack succeed.

You successfully solved a challenge: Vulnerable Library (Inform the shop about a vulnerable library it is using. (Mention the exact library name and version in your comment))

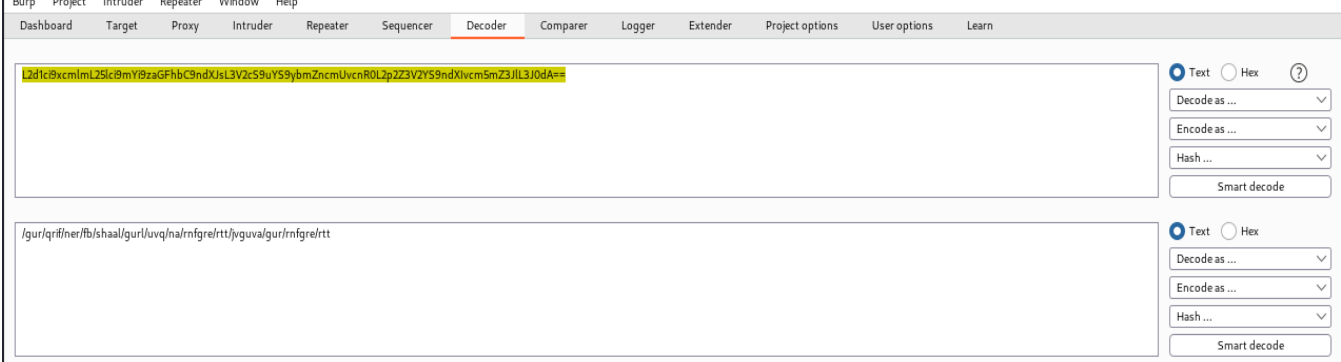
3rd Attack:

You can use easter egg which I got in assignment 4 or you can download as md.

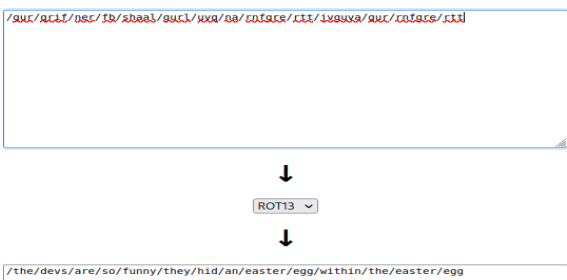


then decoded line 12 as base-64.

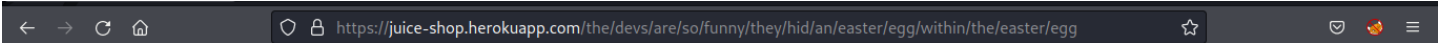
```
1|Congratulations, you found the easter egg!"
2|- The incredibly funny developers
3|
4|...
5|
6|...
7|
8|...
9|
10|Oh' wait, this isn't an easter egg at all! It's just a boring text file! The real easter egg can be found here:
11|
12|L2d1ci9xcmImL25lc9mY9zaGFhbC9ndXJsL3V2cS9uYS9ybmZncmUvcnR0L2p2Z3V2YS9ndXlvcnM5mZ3JlL3J0dA==
13|
14|Good luck, egg hunter!
```



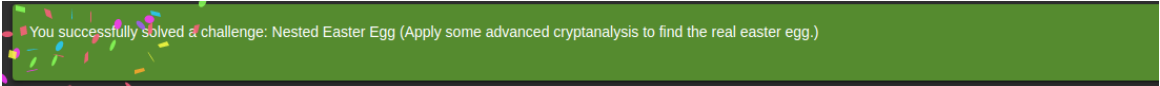
then decode as rot-13.



copy the url after juice-shop url.



Attack succeeded.

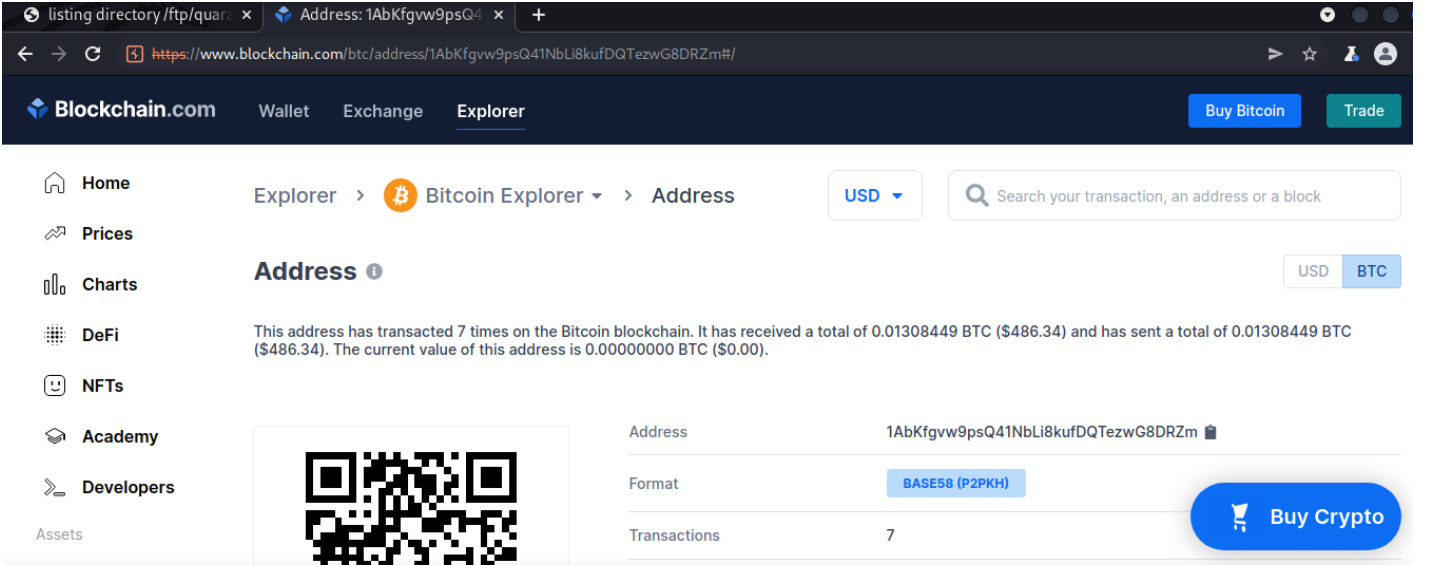
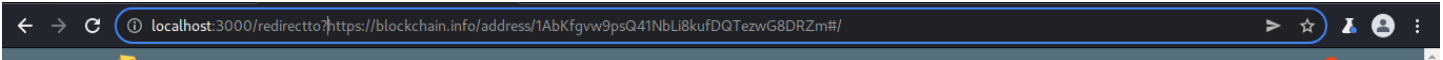


4th Attack:

Getting the allowed redirect list from source code.

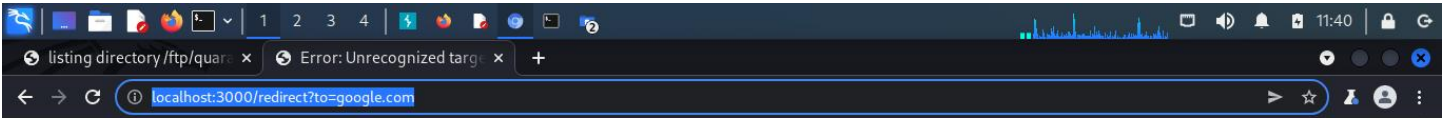
```
94     return undefined
95 }
96
97 function isValidFormat (coupon) {
98     return coupon.match(/^(JAN|FEB|MAR|APR|MAY|JUN|JUL|AUG|SEP|OCT|NOV|DEC)[0-9]{2}-[0-9]{2}/)
99 }
100
101 // vuln-code-snippet start redirectCryptoCurrencyChallenge redirectChallenge
102 const redirectAllowlist = new Set([
103     'https://github.com/bkimminich/juice-shop',
104     'https://blockchain.info/address/1AbKfgvw9psQ41NbLi8kufDQTezwG8DRZm', // vuln-code-snippet vuln-line redirectCryptoCurrencyChallenge
105     'https://explorer.dash.org/address/Xr556RzuwX6hg5E6pkybbv5RanJo2N17kM', // vuln-code-snippet vuln-line redirectCryptoCurrencyChallenge
106     'https://etherscan.io/address/0x0f933ab9fcaa782d0279c300d73750e1311eeae6', // vuln-code-snippet vuln-line redirectCryptoCurrencyChallenge
107     'http://shop.spreadshirt.com/juiceshop',
108     'http://shop.spreadshirt.de/juiceshop',
109     'https://www.stickeryou.com/products/owasp-juice-shop/794',
110     'http://leanpub.com/juice-shop'
111 ])
112 exports.redirectAllowlist = redirectAllowlist
113
114 exports.isRedirectAllowed = url => {
115     let allowed = false
116     for (const allowedUrl of redirectAllowlist) {
117         allowed = allowed || url.includes(allowedUrl) // vuln-code-snippet vuln-line redirectChallenge
118     }
119     return allowed
120 }
121 // vuln-code-snippet end redirectCryptoCurrencyChallenge redirectChallenge
122
123 exports.roles = {
124     customer: 'customer',
125     deluxe: 'deluxe',
126     accounting: 'accounting',
127     admin: 'admin'
128 }
129
130 exports.deluxeToken = (email) => {
131     const hmac = crypto.createHmac('sha256', privateKey)
132     return hmac.update(email + this.roles.deluxe).digest('hex')
```

trying redirecting.



We use both our own cookies and third-party cookies on our websites to enhance your experience, analyze our traffic, and increase site security. [Manage preferences](#) [Accept all](#)

tried redirecting out of the list produced an error for an unrecognized url

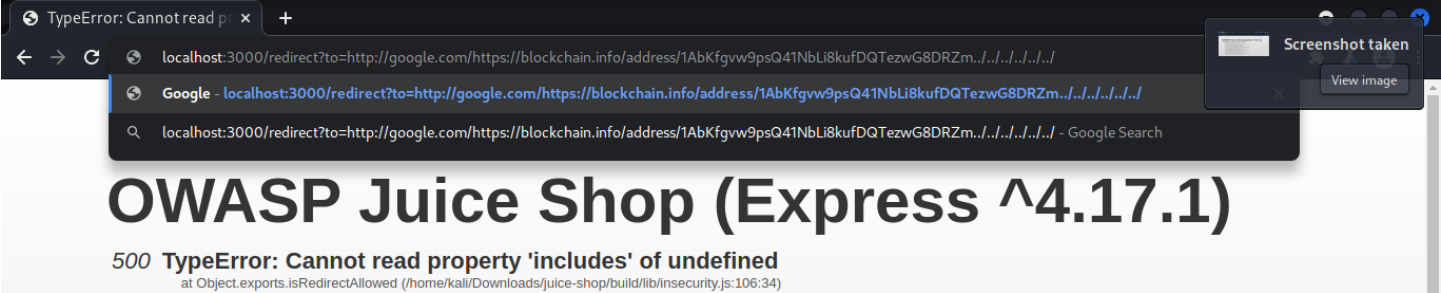
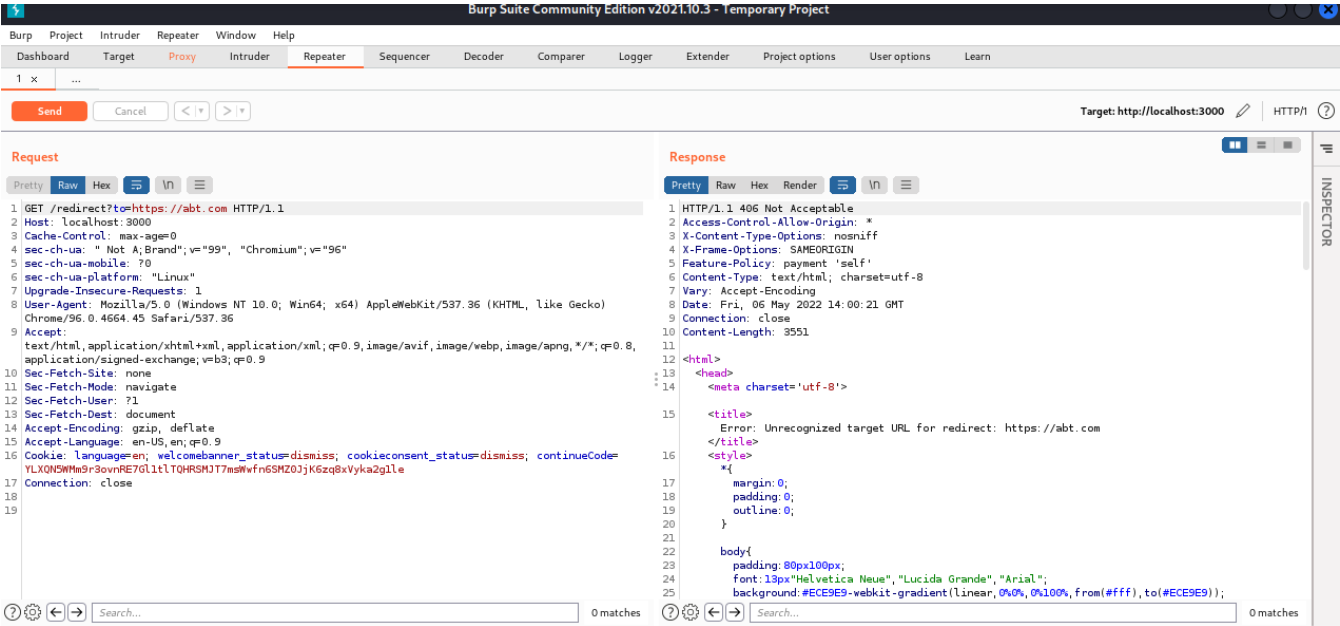


OWASP Juice Shop (Express ^4.17.1)

406 Error: Unrecognized target URL for redirect: google.com

```
at Layer.handle [as handle_request] (/home/kali/Downloads/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at next (/home/kali/Downloads/juice-shop/node_modules/express/lib/router/route.js:144:13)
at Route.dispatch (/home/kali/Downloads/juice-shop/node_modules/express/lib/router/route.js:114:3)
at Layer.handle [as handle_request] (/home/kali/Downloads/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at /home/kali/Downloads/juice-shop/node_modules/express/lib/router/index.js:286:15
at Function.process_params (/home/kali/Downloads/juice-shop/node_modules/express/lib/router/index.js:346:12)
at next (/home/kali/Downloads/juice-shop/node_modules/express/lib/router/index.js:280:10)
at /home/kali/Downloads/juice-shop/build/routes/verify.js:133:5
at Layer.handle [as handle_request] (/home/kali/Downloads/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/home/kali/Downloads/juice-shop/node_modules/express/lib/router/index.js:328:13)
at /home/kali/Downloads/juice-shop/node_modules/express/lib/router/index.js:286:9
at Function.process_params (/home/kali/Downloads/juice-shop/node_modules/express/lib/router/index.js:346:12)
at next (/home/kali/Downloads/juice-shop/node_modules/express/lib/router/index.js:280:10)
at /home/kali/Downloads/juice-shop/build/routes/verify.js:69:5
at Layer.handle [as handle_request] (/home/kali/Downloads/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/home/kali/Downloads/juice-shop/node_modules/express/lib/router/index.js:328:13)
at /home/kali/Downloads/juice-shop/node_modules/express/lib/router/index.js:286:9
at Function.process_params (/home/kali/Downloads/juice-shop/node_modules/express/lib/router/index.js:346:12)
at next (/home/kali/Downloads/juice-shop/node_modules/express/lib/router/index.js:280:10)
at logger (/home/kali/Downloads/juice-shop/node_modules/morgan/index.js:144:5)
at Layer.handle [as handle_request] (/home/kali/Downloads/juice-shop/node_modules/express/lib/router/layer.js:95:5)
```

tried to redirect to several urls to do the attack.



Attack succeeded as it’s redirected me to google page



You successfully solved a challenge: Allowlist Bypass (Enforce a redirect to a page you are not supposed to redirect to.)

5th Attack:
Checked github repo.

https://github.com/juice-shop/juicy-chat-bot/blob/master/index.js

Booking.comTripAdvisorInternational Journ...https://forms.gle/kL...Editorial Manager®Minimizin

```
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
class Bot {
  constructor (name, greeting, trainingSet, defaultResponse) {
    this.name = name
    this.greeting = greeting
    this.defaultResponse = { action: 'response', body: defaultResponse }
    this.training = {
      state: false,
      data: trainingSet
    }
    this.factory = new VM({
      sandbox: {
        Nlp: NlpManager,
        training: this.training
      }
    })
    this.factory.run(ctx)
    this.factory.run(`trainingSet-${trainingSet}`)
  }

  greet (token) {
    return this.render(this.greeting, token)
  }

  addUser (token, name) {
    this.factory.run(`users.addUser("${token}", "${name}")`)
  }

  getUser (token) {
    return this.factory.run(`users.get("${token}")`)
  }

  render (statement, token) {
    return statement.replace(/<bot-name>/g, this.name).replace(/<customer-name>/g, this.factory.run(`currentUser("${token}")`))
  }

  async respond (query, token) {
    const response = (await this.factory.run(`process("${query}", "${token}")`)).answer
    if (!response) {
```

logged in as admin by sql injection.

Login

Email *

admin@juice-sh.op'--

Password *

Forgot your password?

Log in

Remember me

User Profile



Email:
admin@juice-sh.op

Username:
e.g. SuperUser


Set Username

File Upload:
Choose File No file chosen

Upload Picture

To perform the attack add this code by sql injection.

User Profile



Email:
admin@juice-sh.op

Username:
admin"); process=null;users.addUser("1337","te

Set Username

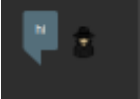
File Upload:
Choose File No file chosen

Upload Picture

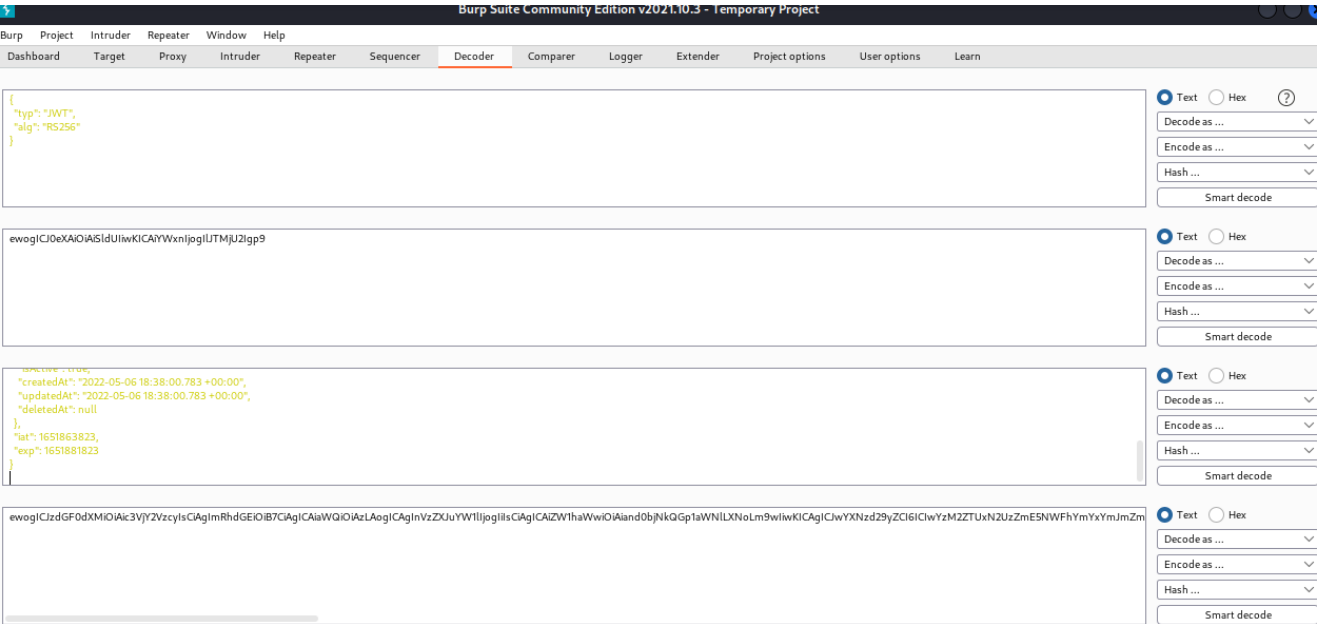
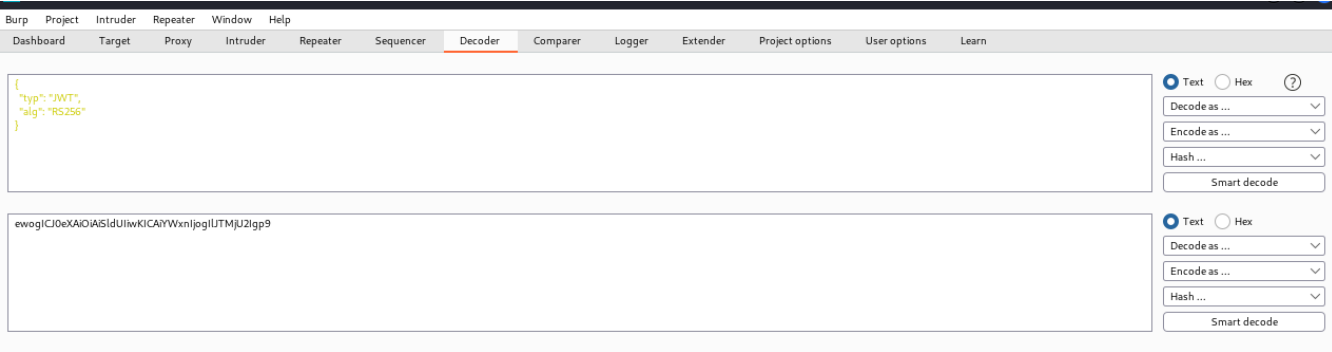
or

Image URL:
e.g. https://www.github.com/assets/29272622/avatars/1337/1337.png

Attack succeeded.



start encoding the new code to base-64 to get the new bearer.



copy new bearer and replace the old in the request.

Attack succeeded

