

Laboratory: small-data-leak

Description:

I do not know what is wrong /user?id=. It's not working at all. All I know is that an attacker is asking us for a ransom...

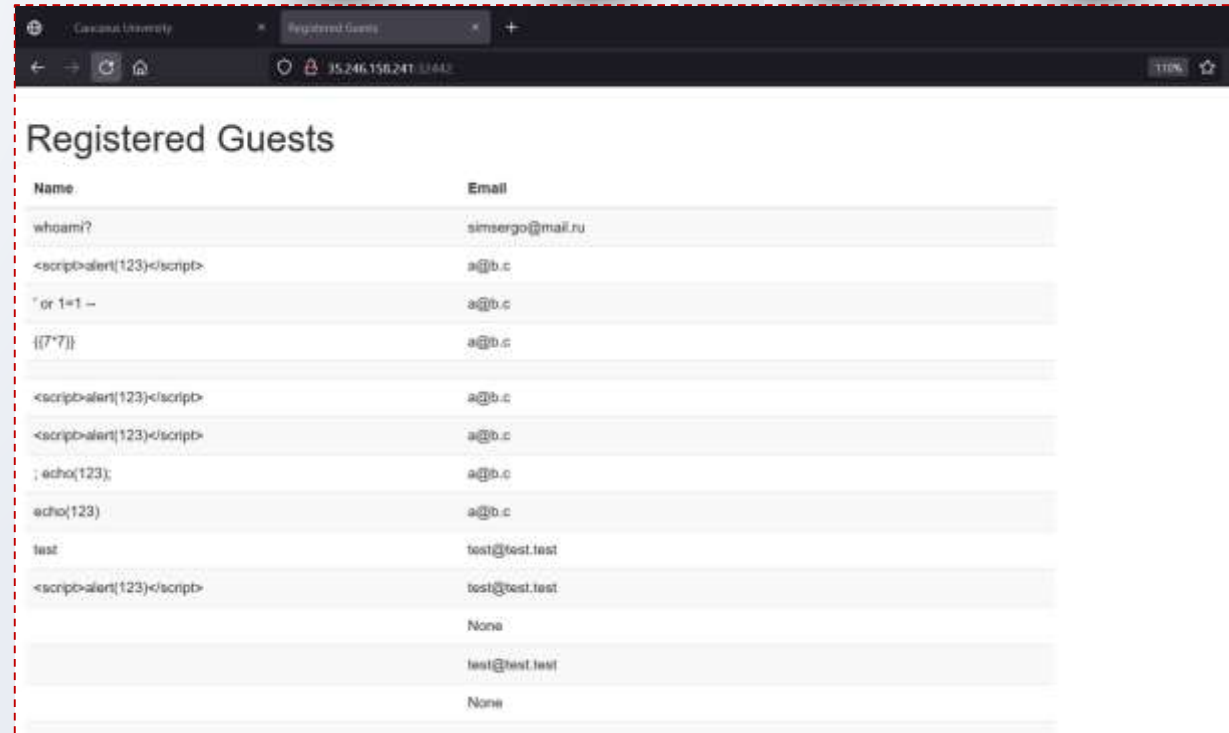
Flag format: CTF{sha256}

Level: Medium

Server: 35.246.158.241:32442

Hints:

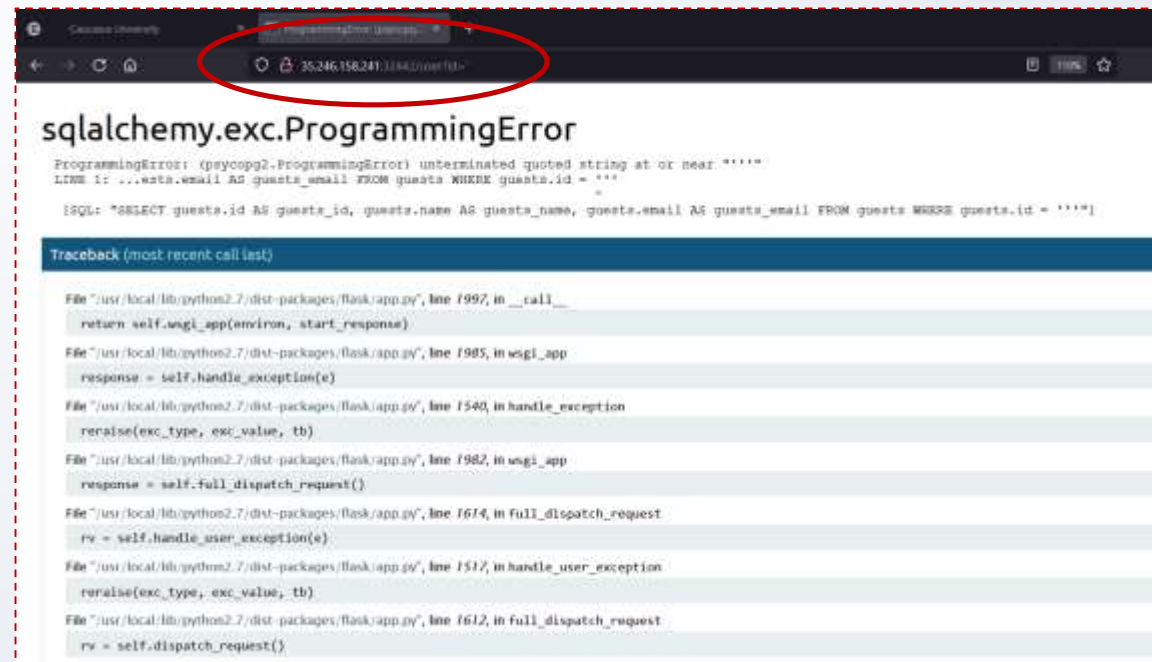
- Hint 1: **sqlmap** is your friend



Name	Email
whoami?	simsergo@mail.ru
<script>alert(123)</script>	a@b.c
" or 1=1 --	a@b.c
{{7*7}}	a@b.c
<script>alert(123)</script>	a@b.c
<script>alert(123)</script>	a@b.c
; echo(123);	a@b.c
echo(123)	a@b.c
test	test@test.test
<script>alert(123)</script>	test@test.test
	None
	test@test.test
	None

Laboratory: small-data-leak

Try to use hint from the description: `/user?id=`



```
sqlalchemy.exc.ProgrammingError
ProgrammingError (psycopg2.ProgrammingError) unterminated quoted string at or near ''''
LINE 1: ...ests_email AS guests_email FROM guests WHERE guests.id = ''
                                ^
SQL: "SELECT guests.id AS guests_id, guests.name AS guests_name, guests.email AS guests_email FROM guests WHERE guests.id = ''"
```

Traceback (most recent call last)

```
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1997, in __call__
    return self.wsgi_app(environ, start_response)
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1985, in wsgi_app
    response = self.handle_exception(e)
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1540, in handle_exception
    reraise(exc_type, exc_value, tb)
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1982, in wsgi_app
    response = self.full_dispatch_request()
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1614, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1517, in handle_user_exception
    reraise(exc_type, exc_value, tb)
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1612, in full_dispatch_request
    rv = self.dispatch_request()
```

Based on the error above, it was concluded that SQLAlchemy is a python library used for interacting with SQL databases.

Since the SQL vulnerability seems pretty basic, we will use SQLmap to obtain the injection point.

Diagram of a hydrogen atom (H) showing the nucleus (p+, n0) and the electron (e-) orbiting in a shell. The diagram is labeled with H, p+, n0, e-, and V.

<http://sqlmap.org>

```
sqlmap: error: missing a mandatory option (-d, -u, -l, -m, -r, -g, -c, --list-tampers, --wizard, --update, --purge or --dependencies). Use -h for basic and -hh for advanced help
```

```
(kali㉿kali)
$ sqlmap -hh
```

```
$sqlmap -u http://35.246.158.241:31089/??????? -D ????????????
```

\$????????????????

Laboratory: ping-station

Description:

Just another ping service to audit.

Flag format: ECSC {sha256}

Level: Easy

Server: 35.246.134.224:31532

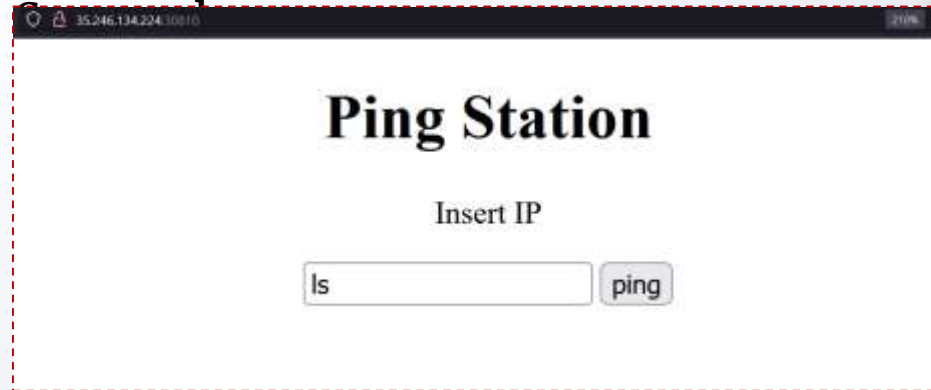
Hints:

- **Hint 1:** Regex
- **Hint 2:** Command injection



Laboratory: ping-station

Try Only



A screenshot of a web browser window showing a web application titled "Ping Station". The browser's address bar displays the URL "35.246.134.224:30810". The application has a large heading "Ping Station" and a sub-label "Insert IP". Below this is a text input field containing the text "ls" and a button labeled "ping". The entire application area is enclosed in a red dashed border.

Gives the error:

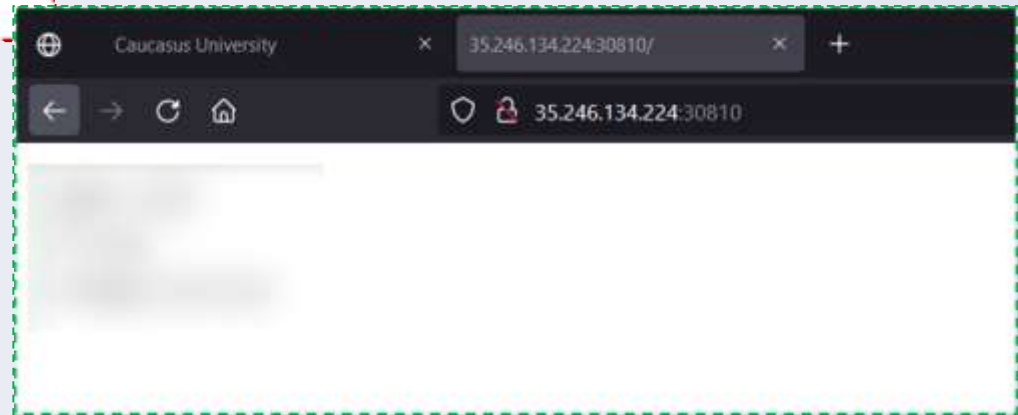


Laboratory: ping-station

Try yourself ...



Result:



Laboratory: ping-station

Try to read the content of ????????????



Result:

