AUBH - Challenge 2
Zain Aqeel
Zain Al-Hashemi
Ruqaya Alasoomi
Mohamed Mansoor
Mazen Emad

Severity for vulnerability:

Low	Medium	High	Critical

<table>
<tr><th></th><th>Severity</th><th>Vulnerability definition</th><th>Identification method</th></tr>
<tr><td>1</td><td>Low</td><td>

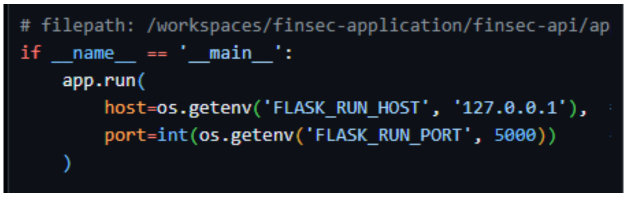
Title: B104 - Hardcoded Bind All Interfaces

Description: Possible binding to all interfaces (host='0.0.0.0').

Affected Locations:
./finsec-api/app/app.py:28:17
./finsec-api/finsec_api/run.py:6:17

Impact:
Binding to all interfaces (0.0.0.0) exposes the application to external networks, which can be a security risk if not properly secured. This is acceptable in development but should be avoided in production unless necessary.

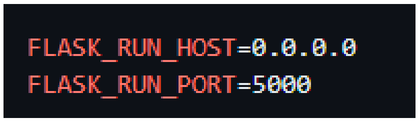
Fix
Use an Environment Variable for the Host: Modify the code to use an environment variable for the host. Default to 127.0.0.1 (localhost) for development. Update the app.run() call in app.py:

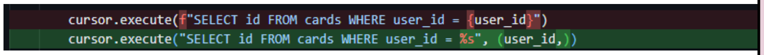


```
# filepath: /workspaces/finsec-application/finsec-api/ap
if __name__ == '__main__':
    app.run(
        host=os.getenv('FLASK_RUN_HOST', '127.0.0.1'),
        port=int(os.getenv('FLASK_RUN_PORT', 5000))
    )
```

Set Environment Variables: For development, you can set the environment variables in a .env file or directly in the terminal

</td><td>Bandit Scan (bandit -r .)</td></tr>
</table>

		
		`FLASK_RUN_HOST=0.0.0.0`
		`FLASK_RUN_PORT=5000`	

2	Medium	**Title:** B608 - Hard Coded SQL Expressions
		
		Description: Possible SQL injection vector through string-based query construction.
		
		Affected locations:
		./finsec-api/finsec_api/add_test_data.py:28:25
		
		Impact:
		Using string interpolation to construct SQL queries can lead to SQL injection vulnerabilities if user input is not properly sanitized. This is a critical security risk.
		
		Fix
		Use parameterized queries to safely pass variables into SQL statements. This prevents SQL injection by ensuring that user input is treated as data, not executable code.
		
		Update the Code: Replace the string interpolation with a parameterized query:
		
		`cursor.execute(f"SELECT id FROM cards WHERE user_id = {user_id}")`
		`cursor.execute("SELECT id FROM cards WHERE user_id = %s", (user_id,))`
		%s is a placeholder for the parameter.
		(user_id,) is a tuple containing the value to be substituted into the query.
			Bandit Scan (bandit -r .)

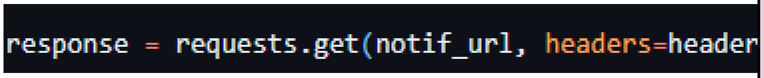
3	Medium	**Title: B113 - Requests Without Timeout** *(TOTAL 15 issues and 15 fixes)* **Description**: Calls to requests without a timeout. **All Affected Files:** test_analytics.py test_api.py test_notif.py Test_notifications.py **Affected Lines:**	Bandit Scan (bandit -r .)

./finsec-api/finsec_api/test_analytics.py:11:15
./finsec-api/finsec_api/test_analytics.py:42:19
./finsec-api/finsec_api/test_analytics.py:62:19
./finsec-api/finsec_api/test_api.py:16:15
./finsec-api/finsec_api/test_api.py:38:15
./finsec-api/finsec_api/test_api.py:61:15
./finsec-api/finsec_api/test_notif.py:15:15
./finsec-api/finsec_api/test_notif.py:32:15
./finsec-api/finsec_api/test_notif.py:48:15
./finsec-api/finsec_api/test_notif.py:71:15
./finsec-api/finsec_api/test_notif.py:90:23
./finsec-api/test_notif.py:15:15
./finsec-api/test_notif.py:32:15
./finsec-api/test_notif.py:48:15
./finsec-api/test_notif.py:71:15
./finsec-api/test_notif.py:90:23
./finsec-api/test_notifications.py:16:21
./finsec-api/test_notifications.py:35:29
./finsec-api/test_notifications.py:52:24
./finsec-api/test_notifications.py:79:31
./finsec-api/test_notifications.py:102:33

Impact:
When making HTTP requests without specifying a timeout, the program can hang indefinitely if the server does not respond. This can lead to denial-of-service vulnerabilities or unresponsive applications.

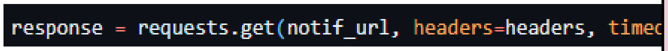
Fix:
Add a timeout parameter to all requests calls. The timeout value should be appropriate for your use case (e.g., 5 seconds).
For a requests.get or requests.post call: response = requests.get(url, timeout=5) # Add a timeout of 5 seconds

Before:

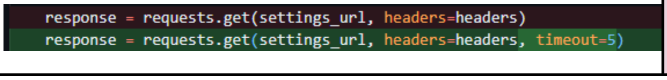


```
response = requests.get(notif_url, headers=header
```

After:

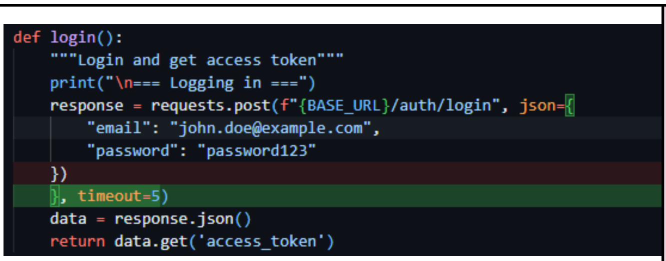


```
response = requests.get(notif_url, headers=headers, timeo
```



```
    response = requests.get(settings_url, headers=headers)
    response = requests.get(settings_url, headers=headers, timeout=5)
```

<table>
<tr>
<td></td>
<td></td>
<td>



```
def login():
    """Login and get access token"""
    print("\n=== Logging in ===")
    response = requests.post(f"{BASE_URL}/auth/login", json={
        "email": "john.doe@example.com",
        "password": "password123"
    })
    }, timeout=5)
    data = response.json()
    return data.get('access_token')
```

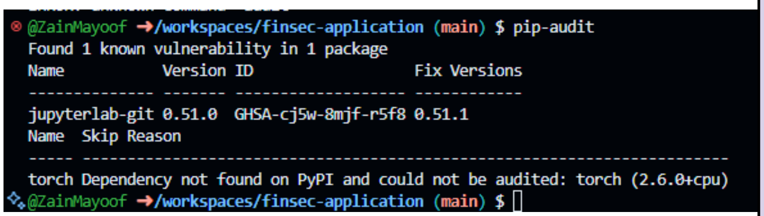
</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Medium</td>
<td>

Title: Vulnerable dependency

Description:

pip install pip-audit

Pip-audit

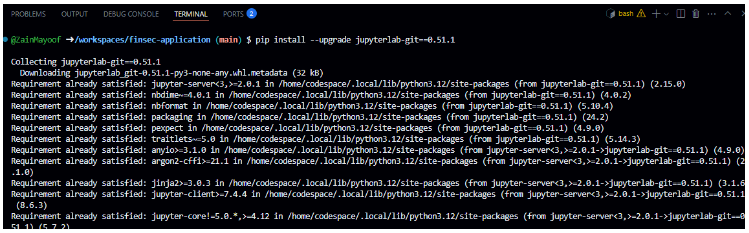


```
⊗ @ZainMayoof ➜/workspaces/finsec-application (main) $ pip-audit
  Found 1 known vulnerability in 1 package
  Name           Version ID                  Fix Versions
  -------------- ------- ------------------- ------------
  jupyterlab-git 0.51.0  GHSA-cj5w-8mjf-r5f8 0.51.1
  Name  Skip Reason
  ----- --------------------------------------------------------------------------
  torch Dependency not found on PyPI and could not be audited: torch (2.6.0+cpu)
  @ZainMayoof ➜/workspaces/finsec-application (main) $ 
```

This means the version of jupyterlab-git you are using has a known vulnerability, and it can be resolved by upgrading to version 0.51.1 or later.

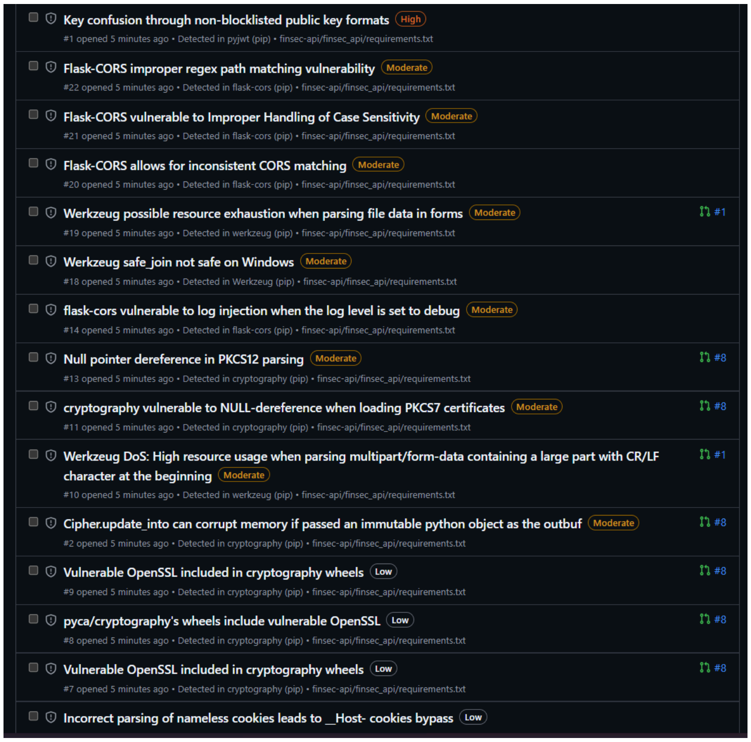
Fix:

pip install --upgrade jupyterlab-git==0.51.1



```
@ZainMayoof ➜/workspaces/finsec-application (main) $ pip install --upgrade jupyterlab-git==0.51.1

Collecting jupyterlab-git==0.51.1
  Downloading jupyterlab_git-0.51.1-py3-none-any.whl.metadata (32 kB)
Requirement already satisfied: jupyter-server<3,>=2.0.1 in /home/codespace/.local/lib/python3.12/site-packages (from jupyterlab-git==0.51.1) (2.15.0)
Requirement already satisfied: nbdime~=4.0.1 in /home/codespace/.local/lib/python3.12/site-packages (from jupyterlab-git==0.51.1) (4.0.2)
Requirement already satisfied: nbformat in /home/codespace/.local/lib/python3.12/site-packages (from jupyterlab-git==0.51.1) (5.10.4)
Requirement already satisfied: packaging in /home/codespace/.local/lib/python3.12/site-packages (from jupyterlab-git==0.51.1) (24.2)
Requirement already satisfied: pexpect in /home/codespace/.local/lib/python3.12/site-packages (from jupyterlab-git==0.51.1) (4.9.0)
Requirement already satisfied: traitlets~=5.0 in /home/codespace/.local/lib/python3.12/site-packages (from jupyterlab-git==0.51.1) (5.14.3)
Requirement already satisfied: anyio>=3.1.0 in /home/codespace/.local/lib/python3.12/site-packages (from jupyter-server<3,>=2.0.1->jupyterlab-git==0.51.1) (4.9.0)
Requirement already satisfied: argon2-cffi>=21.1 in /home/codespace/.local/lib/python3.12/site-packages (from jupyter-server<3,>=2.0.1->jupyterlab-git==0.51.1) (2
.1.0)
Requirement already satisfied: jinja2>=3.0.3 in /home/codespace/.local/lib/python3.12/site-packages (from jupyter-server<3,>=2.0.1->jupyterlab-git==0.51.1) (3.1.6
Requirement already satisfied: jupyter-client>=7.4.4 in /home/codespace/.local/lib/python3.12/site-packages (from jupyter-server<3,>=2.0.1->jupyterlab-git==0.51.1
 (8.6.3)
Requirement already satisfied: jupyter-core!=5.0.*,>=4.12 in /home/codespace/.local/lib/python3.12/site-packages (from jupyter-server<3,>=2.0.1->jupyterlab-git==0
51.1) (5.7.2)
```

</td>
<td>Dependency Vulnerability Scanning (pip-audit)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>22 Issues</td>
<td></td>
</tr>
</table>



□ Key confusion through non-blocklisted public key formats High
#1 opened 5 minutes ago • Detected in pyjwt (pip) • finsec-api/finsec_api/requirements.txt

□ Flask-CORS improper regex path matching vulnerability Moderate
#22 opened 5 minutes ago • Detected in flask-cors (pip) • finsec-api/finsec_api/requirements.txt

□ Flask-CORS vulnerable to Improper Handling of Case Sensitivity Moderate
#21 opened 5 minutes ago • Detected in flask-cors (pip) • finsec-api/finsec_api/requirements.txt

□ Flask-CORS allows for inconsistent CORS matching Moderate
#20 opened 5 minutes ago • Detected in flask-cors (pip) • finsec-api/finsec_api/requirements.txt

□ Werkzeug possible resource exhaustion when parsing file data in forms Moderate #1
#19 opened 5 minutes ago • Detected in werkzeug (pip) • finsec-api/finsec_api/requirements.txt

□ Werkzeug safe_join not safe on Windows Moderate
#18 opened 5 minutes ago • Detected in Werkzeug (pip) • finsec-api/finsec_api/requirements.txt

□ flask-cors vulnerable to log injection when the log level is set to debug Moderate
#14 opened 5 minutes ago • Detected in flask-cors (pip) • finsec-api/finsec_api/requirements.txt

□ Null pointer dereference in PKCS12 parsing Moderate #8
#13 opened 5 minutes ago • Detected in cryptography (pip) • finsec-api/finsec_api/requirements.txt

□ cryptography vulnerable to NULL-dereference when loading PKCS7 certificates Moderate #8
#11 opened 5 minutes ago • Detected in cryptography (pip) • finsec-api/finsec_api/requirements.txt

□ Werkzeug DoS: High resource usage when parsing multipart/form-data containing a large part with CR/LF character at the beginning Moderate #1
#10 opened 5 minutes ago • Detected in werkzeug (pip) • finsec-api/finsec_api/requirements.txt

□ Cipher.update_into can corrupt memory if passed an immutable python object as the outbuf Moderate #8
#2 opened 5 minutes ago • Detected in cryptography (pip) • finsec-api/finsec_api/requirements.txt

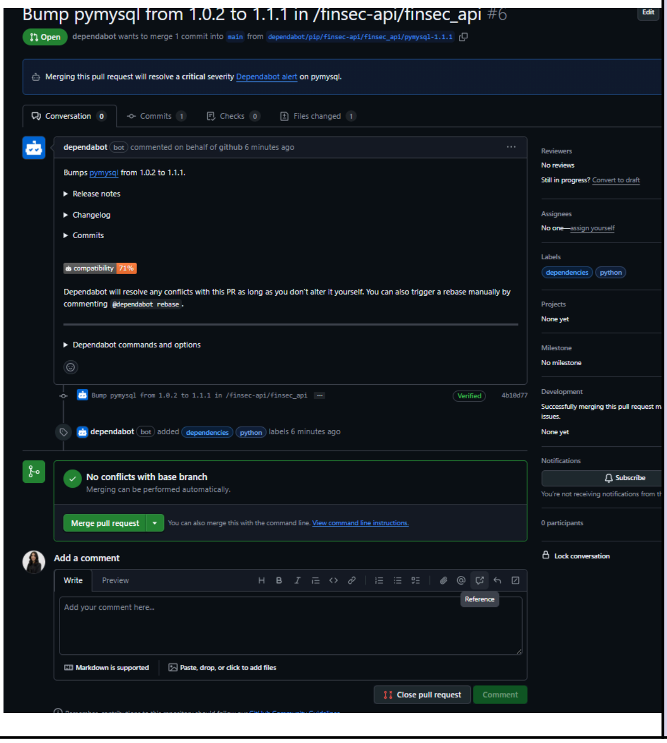
□ Vulnerable OpenSSL included in cryptography wheels Low #8
#9 opened 5 minutes ago • Detected in cryptography (pip) • finsec-api/finsec_api/requirements.txt

□ pyca/cryptography's wheels include vulnerable OpenSSL Low #8
#8 opened 5 minutes ago • Detected in cryptography (pip) • finsec-api/finsec_api/requirements.txt

□ Vulnerable OpenSSL included in cryptography wheels Low #8
#7 opened 5 minutes ago • Detected in cryptography (pip) • finsec-api/finsec_api/requirements.txt

□ Incorrect parsing of nameless cookies leads to __Host- cookies bypass Low

1. PyMySQL through 1.1.0 allows SQL injection if used with untrusted JSON input because keys are not escaped by escape_dict.
   Solution: bumppymysql from 1.0.2 to 1.1.1.



Bump pymysql from 1.0.2 to 1.1.1 in /finsec-api/finsec_api #6

Open dependabot wants to merge 1 commit into `main` from `dependabot/pip/finsec-api/finsec_api/pymysql-1.1.1`

Merging this pull request will resolve a critical severity Dependabot alert on pymysql.

dependabot bot commented on behalf of github 6 minutes ago

Bumps pymysql from 1.0.2 to 1.1.1.

- Release notes
- Changelog
- Commits

compatibility 71%

Dependabot will resolve any conflicts with this PR as long as you don't alter it yourself. You can also trigger a rebase manually by commenting `@dependabot rebase`.

- Dependabot commands and options

`Bump pymysql from 1.0.2 to 1.1.1 in /finsec-api/finsec_api` Verified 4b10d77

dependabot bot added dependencies python labels 6 minutes ago

No conflicts with base branch
Merging can be performed automatically.

Merge pull request — You can also merge this with the command line. View command line instructions.

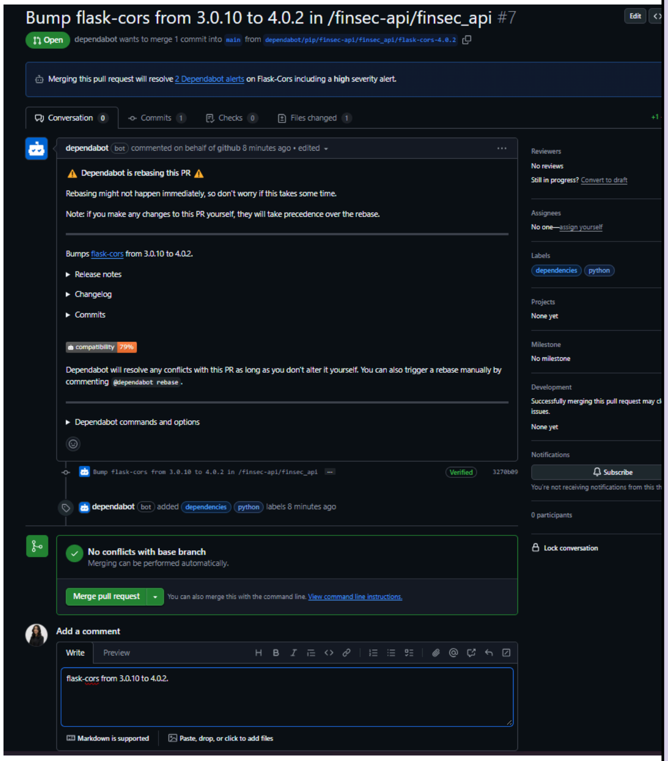
Add a comment

Add your comment here...

Markdown is supported | Paste, drop, or click to add files

Reviewers: No reviews
Still in progress? Convert to draft
Assignees: No one—assign yourself
Labels: dependencies python
Projects: None yet
Milestone: No milestone
Development: Successfully merging this pull request m issues.
None yet
Notifications: You're not receiving notifications from th
0 participants

2. Bump flask-cors from 3.0.10 to 4.0.2 in /finsec-api/finsec_api #7



Bump flask-cors from 3.0.10 to 4.0.2 in /finsec-api/finsec_api #7

Open dependabot wants to merge 1 commit into `main` from `dependabot/pip/finsec-api/finsec_api/flask-cors-4.0.2`

Merging this pull request will resolve 2 Dependabot alerts on Flask-Cors including a high severity alert.

dependabot bot commented on behalf of github 8 minutes ago • edited

⚠ Dependabot is rebasing this PR ⚠

Rebasing might not happen immediately, so don't worry if this takes some time.

Note: if you make any changes to this PR yourself, they will take precedence over the rebase.

Bumps flask-cors from 3.0.10 to 4.0.2.

▶ Release notes

▶ Changelog

▶ Commits

compatibility 79%

Dependabot will resolve any conflicts with this PR as long as you don't alter it yourself. You can also trigger a rebase manually by commenting `@dependabot rebase`.

▶ Dependabot commands and options

`Bump flask-cors from 3.0.10 to 4.0.2 in /finsec-api/finsec_api` Verified `3270b09`

dependabot bot added dependencies python labels 8 minutes ago

No conflicts with base branch
Merging can be performed automatically.

You can also merge this with the command line. View command line instructions.

Add a comment

flask-cors from 3.0.10 to 4.0.2.

Reviewers
No reviews
Still in progress? Convert to draft

Assignees
No one—assign yourself

Labels
dependencies python

Projects
None yet

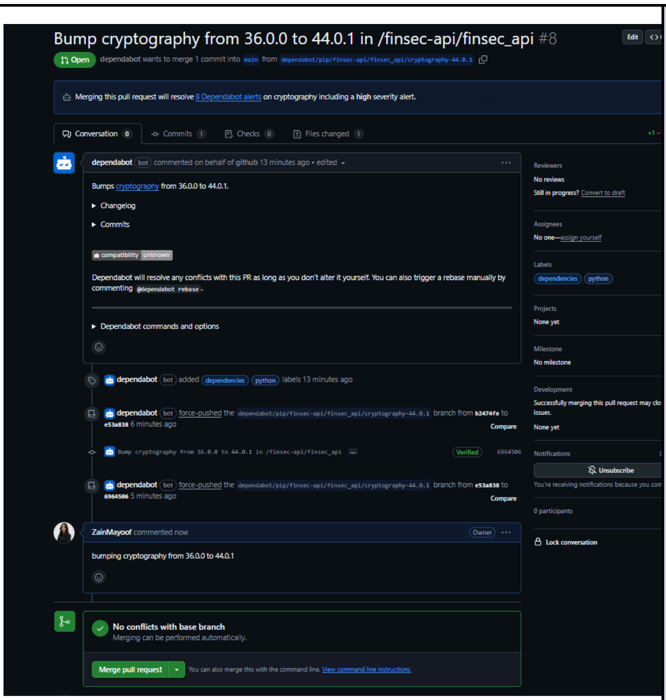
Milestone
No milestone

Development
Successfully merging this pull request may cl
issues.
None yet

Notifications
You're not receiving notifications from this th

0 participants

3. Python Cryptography package vulnerable to Bleichenbacher timing oracle attack #12



Bump cryptography from 36.0.0 to 44.0.1 in /finsec-api/finsec_api #8

Open dependabot wants to merge 1 commit into `main` from `dependabot/pip/finsec-api/finsec_api/cryptography-44.0.1`

Merging this pull request will resolve 8 Dependabot alerts on cryptography including a high severity alert.

dependabot bot commented on behalf of github 13 minutes ago • edited

Bumps cryptography from 36.0.0 to 44.0.1.

▶ Changelog

▶ Commits

compatibility unknown

Dependabot will resolve any conflicts with this PR as long as you don't alter it yourself. You can also trigger a rebase manually by commenting `@dependabot rebase`.

▶ Dependabot commands and options

dependabot bot added dependencies python labels 13 minutes ago

dependabot bot force-pushed the `dependabot/pip/finsec-api/finsec_api/cryptography-44.0.1` branch from **b2474fe** to **e53a838** 6 minutes ago

```
Bump cryptography from 36.0.0 to 44.0.1 in /finsec-api/finsec_api
```
Verified 6964506

dependabot bot force-pushed the `dependabot/pip/finsec-api/finsec_api/cryptography-44.0.1` branch from **e53a838** to **6964506** 5 minutes ago

ZainMayoof commented now

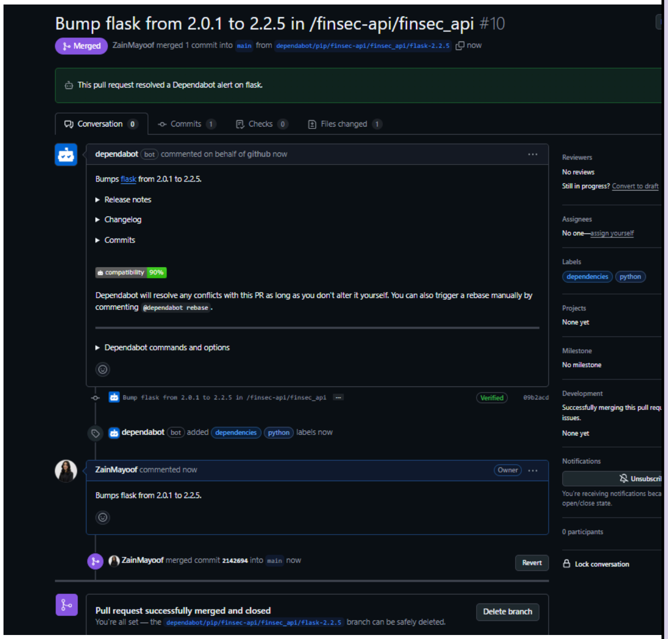
bumping cryptography from 36.0.0 to 44.0.1

No conflicts with base branch
Merging can be performed automatically.

You can also merge this with the command line. View command line instructions.

Reviewers: No reviews
Still in progress? Convert to draft
Assignees: No one—assign yourself
Labels: dependencies python
Projects: None yet
Milestone: No milestone
Development: Successfully merging this pull request may clo
issues.
None yet
Notifications
You're receiving notifications because you com
0 participants

4. Flask vulnerable to possible disclosure of permanent session cookie due to missing Vary: Cookie header #6



Bump flask from 2.0.1 to 2.2.5 in /finsec-api/finsec_api #10

Merged ZainMayoof merged 1 commit into `main` from `dependabot/pip/finsec-api/finsec_api/flask-2.2.5` now

This pull request resolved a Dependabot alert on flask.

dependabot bot commented on behalf of github now

Bumps flask from 2.0.1 to 2.2.5.

▶ Release notes

▶ Changelog

▶ Commits

compatibility 90%

Dependabot will resolve any conflicts with this PR as long as you don't alter it yourself. You can also trigger a rebase manually by commenting `@dependabot rebase`.

▶ Dependabot commands and options

```
Bump flask from 2.0.1 to 2.2.5 in /finsec-api/finsec_api
```
Verified 09b2acd

dependabot bot added dependencies python labels now

ZainMayoof commented now

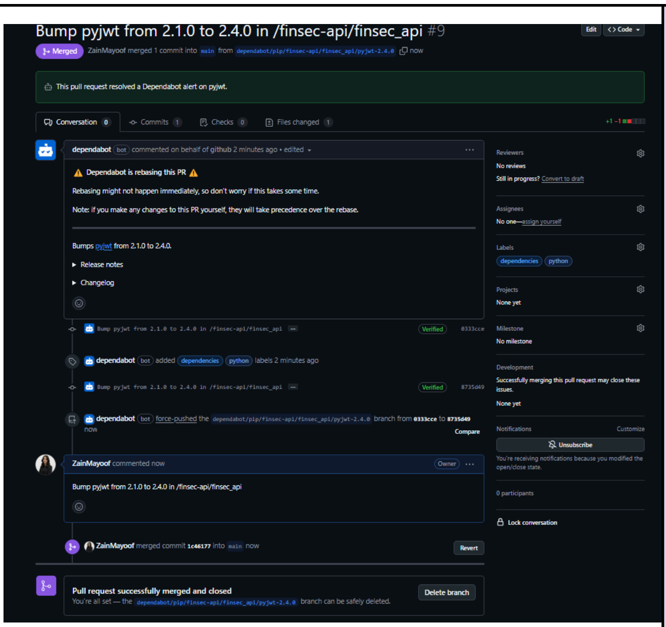
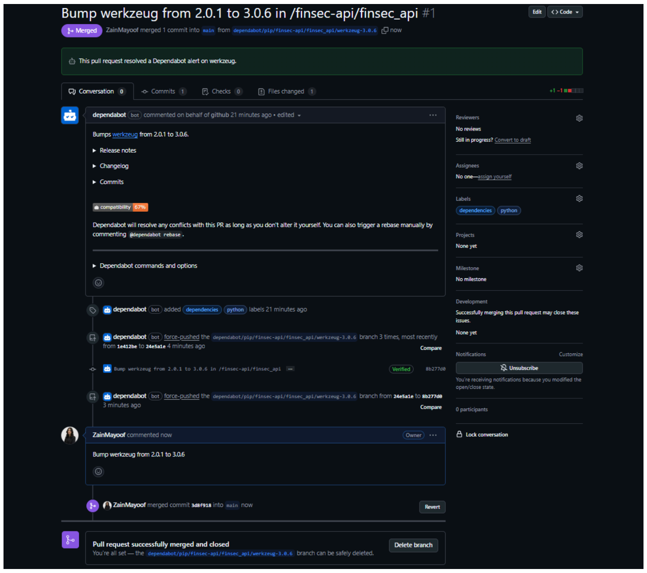
Bumps flask from 2.0.1 to 2.2.5.

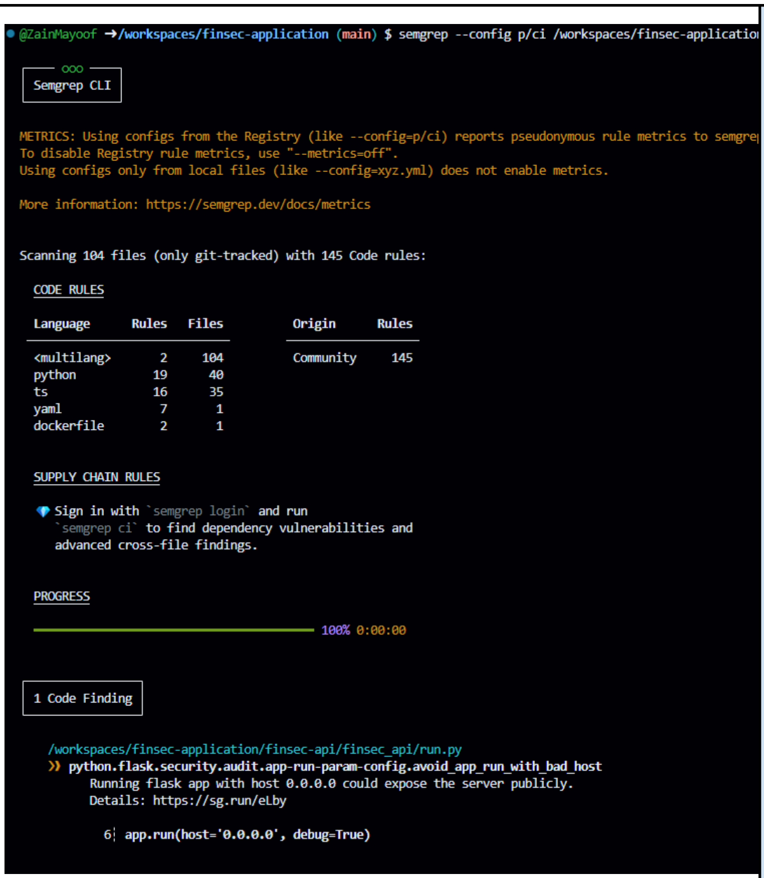
ZainMayoof merged commit **2142694** into `main` now

Pull request successfully merged and closed
You're all set — the `dependabot/pip/finsec-api/finsec_api/flask-2.2.5` branch can be safely deleted.

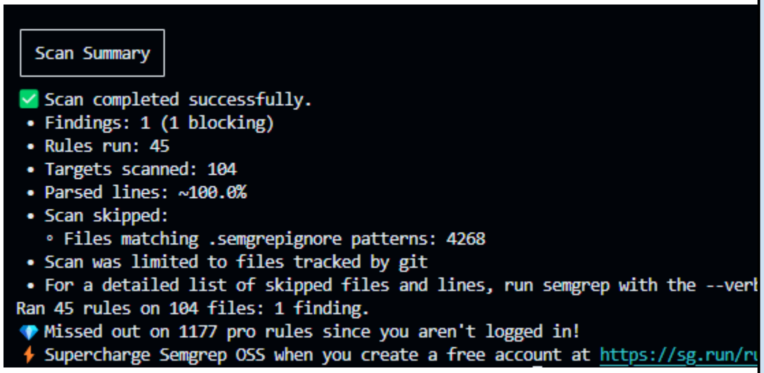
Reviewers: No reviews
Still in progress? Convert to draft
Assignees: No one—assign yourself
Labels: dependencies python
Projects: None yet
Milestone: No milestone
Development: Successfully merging this pull requ
issues.
None yet
Notifications
You're receiving notifications beca
open/close state.
0 participants

5. Bump pyjwt from 2.1.0 to 2.4.0 in /finsec-api/finsec_api #9

		 Bump pyjwt from 2.1.0 to 2.4.0 in /finsec-api/finsec_api #9 ZainMayoof merged 1 commit into `main` from `dependabot/pip/finsec-api/finsec_api/pyjwt-2.4.0` now This pull request resolved a Dependabot alert on pyjwt. dependabot bot commented on behalf of github 2 minutes ago • edited ⚠ Dependabot is rebasing this PR ⚠ Rebasing might not happen immediately, so don't worry if this takes some time. Note: if you make any changes to this PR yourself, they will take precedence over the rebase. Bumps pyjwt from 2.1.0 to 2.4.0. ▸ Release notes ▸ Changelog `Bump pyjwt from 2.1.0 to 2.4.0 in /finsec-api/finsec_api` Verified 0333cce dependabot bot added dependencies python labels 2 minutes ago `Bump pyjwt from 2.1.0 to 2.4.0 in /finsec-api/finsec_api` Verified 8735d49 dependabot bot force-pushed the `dependabot/pip/finsec-api/finsec_api/pyjwt-2.4.0` branch from 0333cce to 8735d49 now ZainMayoof commented now Bump pyjwt from 2.1.0 to 2.4.0 in /finsec-api/finsec_api ZainMayoof merged commit 1c46177 into `main` now Pull request successfully merged and closed You're all set — the `dependabot/pip/finsec-api/finsec_api/pyjwt-2.4.0` branch can be safely deleted. Reviewers: No reviews Still in progress? Convert to draft Assignees: No one—assign yourself Labels: dependencies python Projects: None yet Milestone: No milestone Development: Successfully merging this pull request may close these issues. None yet Notifications You're receiving notifications because you modified the open/close state. 0 participants Edit Code Merged Conversation 0 Commits 1 Checks 0 Files changed 1 +1 −1 Compare Owner Revert Delete branch Customize Unsubscribe Lock conversation 6. Bump werkzeug from 2.0.1 to 3.0.6 in /finsec-api/finsec_api #1  Bump werkzeug from 2.0.1 to 3.0.6 in /finsec-api/finsec_api #1 ZainMayoof merged 1 commit into `main` from `dependabot/pip/finsec-api/finsec_api/werkzeug-3.0.6` now This pull request resolved a Dependabot alert on werkzeug. dependabot bot commented on behalf of github 21 minutes ago • edited Bumps werkzeug from 2.0.1 to 3.0.6. ▸ Release notes ▸ Changelog ▸ Commits compatibility 67% Dependabot will resolve any conflicts with this PR as long as you don't alter it yourself. You can also trigger a rebase manually by commenting `@dependabot rebase`. ▸ Dependabot commands and options dependabot bot added dependencies python labels 21 minutes ago dependabot bot force-pushed the `dependabot/pip/finsec-api/finsec_api/werkzeug-3.0.6` branch 3 times, most recently from 1e412be to 24e5a1e 4 minutes ago `Bump werkzeug from 2.0.1 to 3.0.6 in /finsec-api/finsec_api` Verified 8b277d0 dependabot bot force-pushed the `dependabot/pip/finsec-api/finsec_api/werkzeug-3.0.6` branch from 24e5a1e to 8b277d0 3 minutes ago ZainMayoof commented now Bump werkzeug from 2.0.1 to 3.0.6 ZainMayoof merged commit 3d8f918 into `main` now Pull request successfully merged and closed You're all set — the `dependabot/pip/finsec-api/finsec_api/werkzeug-3.0.6` branch can be safely deleted. Reviewers: No reviews Still in progress? Convert to draft Assignees: No one—assign yourself Labels: dependencies python Projects: None yet Milestone: No milestone Development: Successfully merging this pull request may close these issues. None yet Notifications You're receiving notifications because you modified the open/close state. 0 participants Edit Code Merged Conversation 0 Commits 1 Checks 0 Files changed 1 +1 −1 Compare Owner Revert Delete branch Customize Unsubscribe Lock conversation	
5	Medium	**Title**: Insecure Host Binding (Flask app exposed)	Semgrep scan



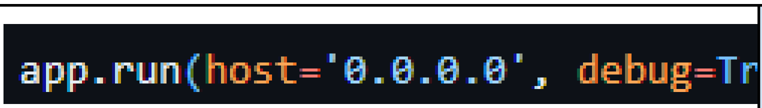
```
● @ZainMayoof ➜/workspaces/finsec-application (main) $ semgrep --config p/ci /workspaces/finsec-applicatio

┌──── ○○○ ────┐
│ Semgrep CLI │
└─────────────┘

METRICS: Using configs from the Registry (like --config=p/ci) reports pseudonymous rule metrics to semgre
To disable Registry rule metrics, use "--metrics=off".
Using configs only from local files (like --config=xyz.yml) does not enable metrics.

More information: https://semgrep.dev/docs/metrics


Scanning 104 files (only git-tracked) with 145 Code rules:

  CODE RULES

  Language      Rules   Files          Origin      Rules
 ─────────────────────────────        ─────────────────
  <multilang>       2     104          Community     145
  python           19      40
  ts               16      35
  yaml              7       1
  dockerfile        2       1


  SUPPLY CHAIN RULES

  💎 Sign in with `semgrep login` and run
     `semgrep ci` to find dependency vulnerabilities and
     advanced cross-file findings.


  PROGRESS

  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 0:00:00


┌────────────────┐
│ 1 Code Finding │
└────────────────┘

    /workspaces/finsec-application/finsec-api/finsec_api/run.py
    ❯❯ python.flask.security.audit.app-run-param-config.avoid_app_run_with_bad_host
          Running flask app with host 0.0.0.0 could expose the server publicly.
          Details: https://sg.run/eLby

            ┆ app.run(host='0.0.0.0', debug=True)
```



```
┌──────────────┐
│ Scan Summary │
└──────────────┘
✅ Scan completed successfully.
 • Findings: 1 (1 blocking)
 • Rules run: 45
 • Targets scanned: 104
 • Parsed lines: ~100.0%
 • Scan skipped:
   ◦ Files matching .semgrepignore patterns: 4268
 • Scan was limited to files tracked by git
 • For a detailed list of skipped files and lines, run semgrep with the --verb
Ran 45 rules on 104 files: 1 finding.
💎 Missed out on 1177 pro rules since you aren't logged in!
⚡ Supercharge Semgrep OSS when you create a free account at https://sg.run/ru
```

Description: Running an app with host='0.0.0.0' could expose the server publicly.
Rule:
python.flask.security.audit.app-run-param-config.avoid_app_run_with_bad_host

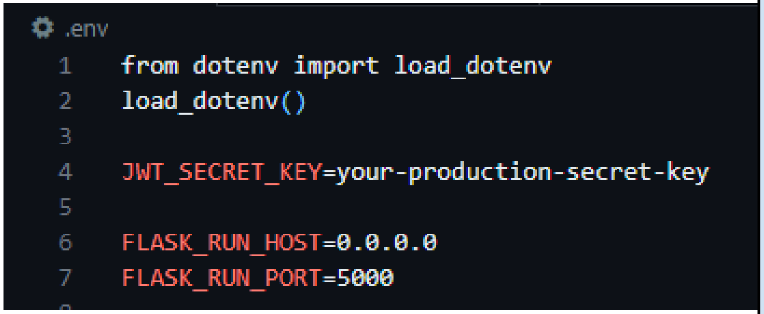
Affected locations:
File: run.py



```
app.run(host='0.0.0.0', debug=Tr
```

Impact:
Binding the Flask app Sem0.0.0.0 exposes it to all network interfaces, making it accessible from external networks. This is acceptable in development but should be avoided in production unless properly secured.

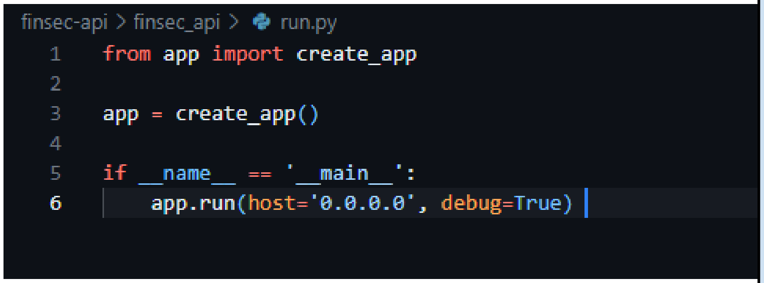
Fix:
Use Environment Variables for the Host:
Update the app.run() call to use environment variables for the host and port.
Default to 127.0.0.1 for development
Set Environment Variables:
Add the following to your .env file (already present from the last bandit scan fixes



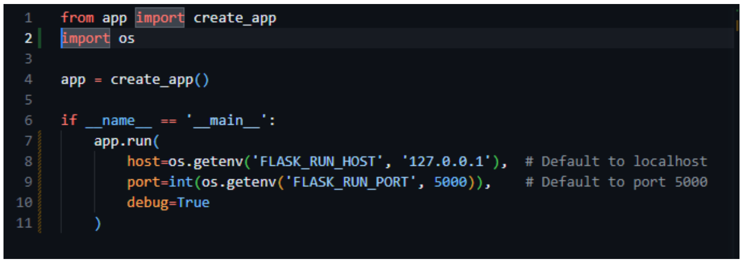
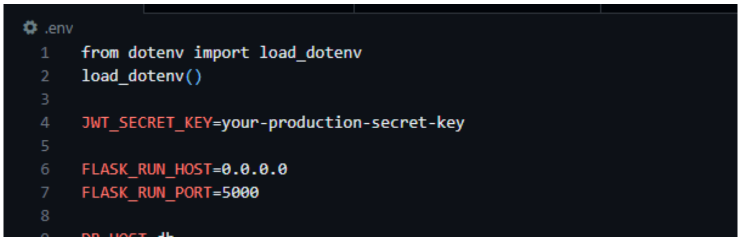
```
from dotenv import load_dotenv
load_dotenv()

JWT_SECRET_KEY=your-production-secret-key

FLASK_RUN_HOST=0.0.0.0
FLASK_RUN_PORT=5000
```

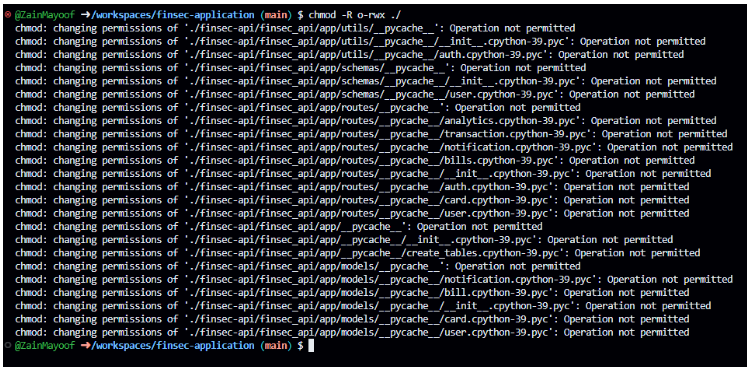
):

Before:



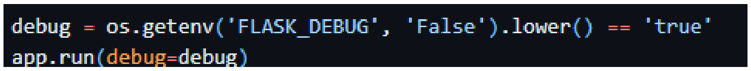
```
from app import create_app

app = create_app()

if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=True)
```

		after:
		
		`from app import create_app`
		`import os`
		
		`app = create_app()`
		
		`if __name__ == '__main__':`
		`    app.run(`
		`        host=os.getenv('FLASK_RUN_HOST', '127.0.0.1'),  # Default to localhost`
		`        port=int(os.getenv('FLASK_RUN_PORT', 5000)),    # Default to port 5000`
		`        debug=True`
		`    )`
		
		.env
		`from dotenv import load_dotenv`
		`load_dotenv()`
		
		`JWT_SECRET_KEY=your-production-secret-key`
		
		`FLASK_RUN_HOST=0.0.0.0`
		`FLASK_RUN_PORT=5000`
		
		`DB_HOST=db`	

6	High	**Title:** CWE-732 Overly Permissive File Permissions via chmod -R 755
		
		Description:
		chmod -R 755 ./ recursively sets read, write, and execute permissions for the owner and read and execute permissions for everyone else on all files and directories in the project.
		
		Impact:
		May expose sensitive files to unauthorized users in shared or multi-user environments.
		
		Fix:
		Remove others' access: chmod -R o-rwx ./
		
		`@ZainMayoof →/workspaces/finsec-application (main) $ chmod -R o-rwx ./`
		`chmod: changing permissions of './finsec-api/finsec_api/app/utils/__pycache__': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/utils/__pycache__/__init__.cpython-39.pyc': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/utils/__pycache__/auth.cpython-39.pyc': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/schemas/__pycache__': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/schemas/__pycache__/__init__.cpython-39.pyc': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/schemas/__pycache__/user.cpython-39.pyc': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/routes/__pycache__': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/routes/__pycache__/analytics.cpython-39.pyc': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/routes/__pycache__/transaction.cpython-39.pyc': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/routes/__pycache__/notification.cpython-39.pyc': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/routes/__pycache__/bills.cpython-39.pyc': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/routes/__pycache__/__init__.cpython-39.pyc': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/routes/__pycache__/auth.cpython-39.pyc': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/routes/__pycache__/card.cpython-39.pyc': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/routes/__pycache__/user.cpython-39.pyc': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/__pycache__': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/__pycache__/__init__.cpython-39.pyc': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/__pycache__/create_tables.cpython-39.pyc': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/models/__pycache__': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/models/__pycache__/notification.cpython-39.pyc': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/models/__pycache__/bill.cpython-39.pyc': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/models/__pycache__/__init__.cpython-39.pyc': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/models/__pycache__/card.cpython-39.pyc': Operation not permitted`
		`chmod: changing permissions of './finsec-api/finsec_api/app/models/__pycache__/user.cpython-39.pyc': Operation not permitted`
		`@ZainMayoof →/workspaces/finsec-application (main) $`
			Set up

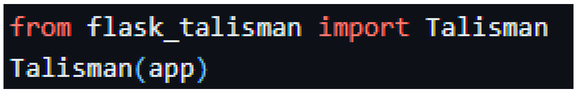
7	High	**Title:** CWE-200 - Exposure of Sensitive Information to an Unauthorized Actor **Description:** Exposed Database to All Interfaces	Set up

		Database user finsec_user is allowed access from %, exposing it to all network interfaces.
		
		Affected Location:
		/workspaces/finsec-application/finsec-api/init.sql
		/workspaces/finsec-application/finsec-api/init-root.sql
		
		Fix:
		Restrict access to specific IPs or localhost:
		BEFORE:
		`CREATE USER IF NOT EXISTS 'finsec_user'@'%' IDENTIFIED BY '${FINSEC_PASSWORD}';`
		After remove %:
		`CREATE USER IF NOT EXISTS 'finsec_user'@'localhost' IDENTIFIED BY 'finsec_password';`	

8	Critical	**Title**: CWE-489 - Debug Mode Enabled in Production
		
		Description:
		The Flask application is running with debug=True, which enables the interactive debugger and detailed error pages. This can expose sensitive environment variables, stack traces, and code, making it a serious risk if left enabled in a production environment.
		
		Affected Location:
		/workspaces/finsec-application/finsec-api/app/app.py
		/workspaces/finsec-application/finsec-api/finsec_api/run.py
		
		Fix:
		Disable debug mode in production:
		
		`debug = os.getenv('FLASK_DEBUG', 'False').lower() == 'true'`
		`app.run(debug=debug)`
			Misconfiguration

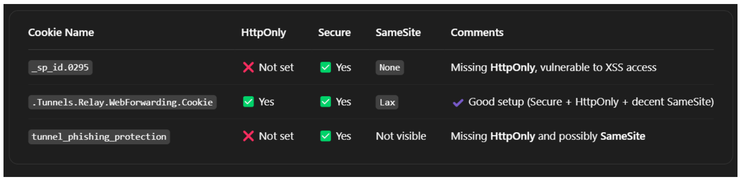
9	Critical	**Title**: CWE-303 - Authentication Bypass or Logic Flaw After Failed Login **Description:**	Misconfiguration

		When logging in with invalid credentials, the application correctly displays an error ("no user ID found"), but still navigates the user to the authentication-protected area. This implies that access control is not properly enforced after login validation fails. **Impact:** ● Authentication bypass — users may access sensitive or protected resources without valid credentials ● Could lead to privilege escalation or data exposure if protected endpoints/pages are accessed ● Undermines the entire authentication system, making the app trivially exploitable	

10	Medium	**Title:** CWE-693 - No Security Headers
		
		Impact: No security headers to protect against common attacks.
		
		Affected Location:
		/workspaces/finsec-application/finsec-api/app/app.py
		Fix:
		Use Flask-Talisman to set security headers
		
		`from flask_talisman import Talisman`
		`Talisman(app)`
			Misconfiguration

11	High	**Title**: CWE-1004, CEW-1004 Missing Secure Attributes on Session Cookies **Description:** The application does not explicitly configure cookies to be secure. Flask's JWTManager may set cookies for JWTs, but without explicitly setting the secure and httponly flags, the cookies could be vulnerable. Why not having http is bad: javascript can access the cookie, and with utilizing xss an attacker can steal cookies Why not having same site is bad: Your cookies are sent with cross-site requests — like if your site receives a	Cookies

<table>
<tr><td></td><td></td><td>POST request from another site.Makes your app vulnerable to CSRF (Cross-Site Request Forgery), where an attacker tricks a logged-in user’s browser into making a malicious request on another site.


<table><tr><th>Cookie Name</th><th>HttpOnly</th><th>Secure</th><th>SameSite</th><th>Comments</th></tr><tr><td>_sp_id.0295</td><td>✗ Not set</td><td>✓ Yes</td><td>None</td><td>Missing **HttpOnly**, vulnerable to XSS access</td></tr><tr><td>.Tunnels.Relay.WebForwarding.Cookie</td><td>✓ Yes</td><td>✓ Yes</td><td>Lax</td><td>✓ Good setup (Secure + HttpOnly + decent SameSite)</td></tr><tr><td>tunnel_phishing_protection</td><td>✗ Not set</td><td>✓ Yes</td><td>Not visible</td><td>Missing **HttpOnly** and possibly **SameSite**</td></tr></table>
Impact:
Cookies being accessible via JavaScript, making them vulnerable to theft via Cross-Site Scripting (XSS).
Cookies being sent with cross-origin requests, making the app vulnerable to Cross-Site Request Forgery (CSRF).
When these flags are not set:
● Attackers can steal session tokens via XSS (no HttpOnly).
● Session tokens can be intercepted over HTTP (no Secure).
● CSRF attacks can be performed using existing cookies (no SameSite).</td><td></td></tr>
<tr><td>12</td><td>High</td><td>**Title:** CWE-384 - Session Fixation

Description: The application does not generate a new session ID upon successful login. This allows an attacker to set a known session ID (like via a phishing link), and if the user logs in without the session ID being regenerated, the attacker can hijack that session.

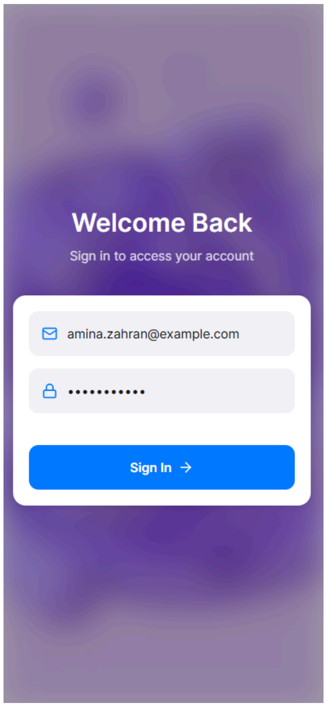
Impact:
The attacker sends a link to the victim with a predefined session ID. The victim logs in. Because the session ID remains unchanged, the attacker can now use the same ID to impersonate the victim. This undermines the integrity of session-based authentication and can lead to full account takeover.</td><td>Cookies</td></tr>
<tr><td>13</td><td>High</td><td>**Title**: CWE-613 - Session Not Invalidated on Logout

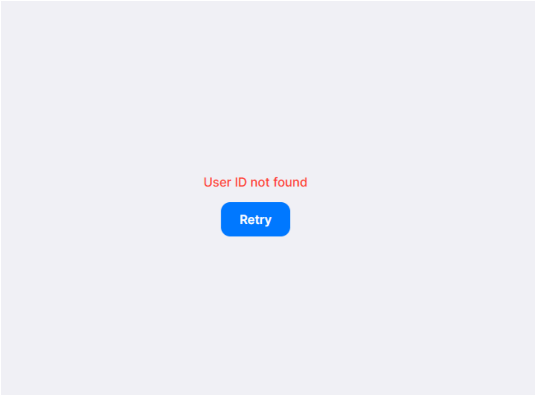
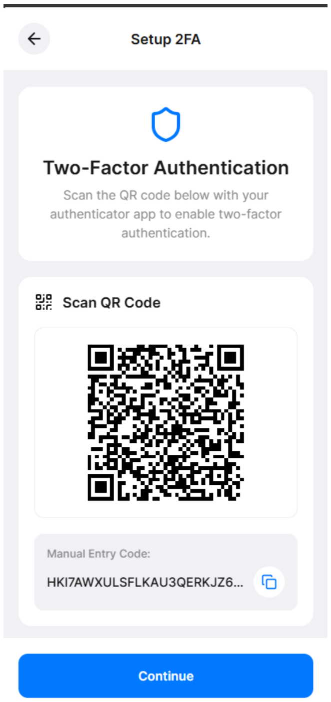
Description:

The application does not destroy or regenerate the</td><td>Cookies</td></tr>
</table>

		session ID upon logout. This means: ● The same session ID remains valid after logout. ● If an attacker had previously captured or predicted the session ID (via XSS, MITM, or session fixation), they can reuse it to re-authenticate without credentials — even after the legitimate user logs out. **Impact:** ● Persistent session IDs = persistent access. ● This leaves logged-out users vulnerable to session hijacking. ● If session IDs aren’t tied to state or expiration, an attacker could log in indefinitely using the same token.	
14	Medium	**Title**: CWE-251 Inadequate Password Complexity Requirements **Description**: The application currently allows users to create passwords that are too basic and easily guessable. A strong password policy is not enforced, allowing passwords that may: ● Lack uppercase/lowercase diversity ● Have no special characters ● Contain whitespace ● Be short or dictionary-based I**mpact:** ● Brute-force and dictionary attacks become significantly more effective. ● Increases risk of account takeover, especially if combined with reused credentials or data from breaches. ● Weak passwords undermine other layers of security (like MFA or rate-limiting) if not properly enforced.	

15	Critical	**Title**: CWE-319 Exposure of Full Credit Card Number and Expiration Date **Description**: The application displays full credit card numbers and expiration dates within the user interface or API responses. This behavior violates PCI DSS (Payment Card Industry Data Security Standard) and poses a major risk of financial fraud and identity theft. **Impact:** ● If unauthorized access occurs (via XSS, compromised account, or insecure endpoints), attackers can: ● Steal cardholder data ● Conduct fraudulent transactions ● Violates compliance standards (e.g., PCI DSS) leading to potential fines or legal action ● Exposing such data unnecessarily increases the attack surface and makes your system a target	
16	High	**Title:** CWE-532 Exposure of MFA Secrets and OTPs via Console Logging **Description:** The init_db.py script prints MFA secrets and one-time passwords (OTPs) to stdout, which can be unintentionally captured by: ● Container logs (e.g., Docker) ● CI/CD pipelines (e.g., GitHub Actions, GitLab) ● Logging infrastructure (e.g., ELK Stack, CloudWatch) This poses a significant risk as OTPs and MFA secrets can be used to bypass two-factor authentication, especially if an attacker has access to logs. **Impact:** ● Compromise of MFA mechanisms: An attacker can register the MFA secret in their own authenticator app. ● Bypass of multi-factor authentication, leading to full account takeover. Logs may persist in: ● CI/CD history ● Log aggregators ● Developer consoles or shared environments	

17		**Title**: CWE-303: Incorrect Implementation of **Issue**: Faulty Authentication Logic — Redirect on Failed Login **Description** When loggin in with the credentials given, the system displays ‘no user ID found’ however the user is then navigated to the authentication page **Impact**: Potential unauthorized access if session or access tokens are still being issued or not cleared  Welcome Back Sign in to access your account amina.zahran@example.com ••••••••••• 	

		 User ID not found  Two-Factor Authentication Scan the QR code below with your authenticator app to enable two-factor authentication. Manual Entry Code: HKI7AWXULSFLKAU3QERKJZ6... 	

	High	Title: CWE-789 - use of hard-coded credentials In the test_api.sh script, sensitive login credentials are hardcoded directly into the request payload, for example: -d '{"email": "john.doe@example.com", "password": "password123"}' This practice exposes secrets in plaintext, making them vulnerable to: ● Accidental exposure via version control (e.g., GitHub) ● Log file leaks from CI/CD pipelines or terminal history ● Insider threats or unauthorized access from team members or attackers with repo access Impact: ● If leaked, hardcoded credentials can grant unauthorized access to user accounts or APIs ● Can lead to account takeover, privilege escalation, or data breaches ● Violates best practices for secure secret management Fix: -d "{\"email\": \"${LOGIN_EMAIL}\", \"password\": \"${LOGIN_PASSWORD}\"}"	