# OWASP Top 10 Toepassing – Werkplaats (WP2)

Student: (vul naam/studentnummer in)

Datum: (vul datum in)

## 1) Kloon de repository

Gekloonde repository:

https://x-access-token:ghs\_qpspTv1bms1b0UGeff9EmBrPLs1WaU1sHiuH@github.com/Joshua-1078191/wp2-2024-mvc-1b4-04\_notfound

## 2) Reproduceerbare kwetsbaarheid

Kwetsbaarheid gereproduceerd door in te loggen als niet-admin en een admin-gebruiker aan te maken via de route voor nieuwe redacteuren.

Bewijs: SQL toont dat gebruiker attacker@example.com met is\_admin=1 is aangemaakt.

## 3) OWASP Top 10 categorie

A01:2021 – Broken Access Control

## 4) Kwetsbare code (regels)

Link check\_login (kwetsbaar): https://x-access-token:ghs\_qpspTv1bms1b0UGeff9EmBrPLs1WaU1sHiuH@github.com/Joshua-1078191/wp2-2024-mvc-1b4-04\_notfound/blob/31c042595a1ca01ffddd8632fa6237a8e7e65c4b/app.py#L40-L45

check\_login (voor fix):

35 database\_generator.generate\_database()
36
37 with app.app\_context():
38 init\_db()
39
40 def check\_login(require\_admin = False):
41 if 'user\_id' not in session:
42 return redirect('/index/login')
43 if require\_admin and session.get('is\_admin'):
44 abort(401)
45 return None
46
47 @app.route('/')
48 def main():
49 return render\_template("homepage.jinja")
50
51 @app.route('/index/login', methods=['GET', 'POST'])
52 def login\_route():
53 if 'user\_id' in session:
54 return redirect('/')
55

Link admin-route (kwetsbaar): https://x-access-token:ghs\_qpspTv1bms1b0UGeff9EmBrPLs1WaU1sHiuH@github.com/Joshua-1078191/wp2-2024-mvc-1b4-04\_notfound/blob/31c042595a1ca01ffddd8632fa6237a8e7e65c4b/app.py#L478-L486

nieuwe\_redacteuren (voor fix):

472 @app.route('/redacteurs/lijst\_redacteuren', methods=['GET', 'POST'])
473 def lijst\_redacteuren():
474 if result := check\_login(): return result
475 users = Users(database\_path)
476 return render\_template("redacteurs/lijst\_redacteuren.html.jinja", editors=users.get\_all())
477
478 @app.route('/redacteurs/nieuwe\_redacteur', methods=['GET', 'POST'])
479 def nieuwe\_redacteuren():
480 if result := check\_login(): return result
481
482 users\_model = Users(database\_path)
483
484 if request.method == 'POST':
485 nieuwe\_redacteur\_id = users\_model.register(
486 email=request.form.get('email'),
487 password=request.form.get('password'),
488 username=request.form.get('name'),
489 is\_admin=bool(request.form.get('is\_admin', False))
490 )
491 if nieuwe\_redacteur\_id:
492 flash('Redacteur succesvol toegevoegd!', 'success')

## 5) Link naar kwetsbare code (commit)

Commit (kwetsbaar): 31c042595a1ca01ffddd8632fa6237a8e7e65c4b

## 6) Uitleg waarom kwetsbaar

De functie check\_login controleerde admin-rechten omgekeerd: bij require\_admin=True werd juist geblokkeerd wanneer de gebruiker admin was, en toegestaan wanneer de gebruiker geen admin was. Bovendien riep de route voor het aanmaken van een nieuwe redacteur check\_login() zonder require\_admin=True aan. Hierdoor kon een niet-admin een account met admin-rechten aanmaken (privilege escalation).

## 7) Nieuwe code (fix)

Link check\_login (fix): https://x-access-token:ghs\_qpspTv1bms1b0UGeff9EmBrPLs1WaU1sHiuH@github.com/Joshua-1078191/wp2-2024-mvc-1b4-04\_notfound/blob/3a8ff3241c78dd50c0e3679641c51e68f5acb5b7/app.py#L40-L45

check\_login (na fix):

40 def check\_login(require\_admin = False):
41 if 'user\_id' not in session:
42 return redirect('/index/login')
43 if require\_admin and not session.get('is\_admin'):
44 abort(401)
45 return None
46
47 @app.route('/')

Link admin-route (fix): https://x-access-token:ghs\_qpspTv1bms1b0UGeff9EmBrPLs1WaU1sHiuH@github.com/Joshua-1078191/wp2-2024-mvc-1b4-04\_notfound/blob/3a8ff3241c78dd50c0e3679641c51e68f5acb5b7/app.py#L478-L486

nieuwe\_redacteuren (na fix):

478 @app.route('/redacteurs/nieuwe\_redacteur', methods=['GET', 'POST'])
479 def nieuwe\_redacteuren():
480 if result := check\_login(True): return result
481
482 users\_model = Users(database\_path)
483
484 if request.method == 'POST':
485 nieuwe\_redacteur\_id = users\_model.register(
486 email=request.form.get('email'),

## 8) Commit ID(s) nieuwe code

Commit (fix): 3a8ff3241c78dd50c0e3679641c51e68f5acb5b7

## 9) Waarom dit de kwetsbaarheid verhelpt

De fix past deny-by-default toe voor admin-acties: wanneer require\_admin=True en de sessie niet admin is, volgt een 401. Daarnaast is de admin-route expliciet beschermd met check\_login(True). Dit volgt OWASP A01-aanbevelingen: expliciete autorisatie checks op elke gevoelige route, principle of least privilege, en systeematische controle vóór uitvoering van gevoelige acties.

## Gebruik van tools

Reproductie is geautomatiseerd met curl en verificatie met SQLite queries. (Optioneel) OWASP ZAP of SQLMap inzet kan worden toegevoegd.