WESTERN SPRINGS COLLEGE

Cyber Security Risk Assessment

WSCW Tuckshop Web Application

therustymate@gmail.com

2025-08/22

ASSESSMENT SCOPE

1.1 Assessment Scope - Teacher Panel & Student Sign In Portal

The following features were included within the scope of the scan:

- ***.[REDACTED].site**

The following APIs were identified during assessment:

- **/api/v1/order**
  - /api/v1/order/carts
  - /api/v1/order/orders
  - /api/v1/order/admin/orders
- **/api/v1/auth**
  - /api/v1/auth/google/login/token
- **/api/v1/user**
  - /api/v1/user/tickets
  - /api/v1/user/logs
  - /api/v1/user/current-user/update
  - /api/v1/user/current-user/info
- **/api/v1/product**
  - /api/v1/product/products
  - /api/v1/product/menus
  - /api/v1/product/active-menus
- **/api/v1/token/refresh**
  - /api/v1/token/refresh

1.1 Assessment Scope - Result

Severity	Score
Critical vulnerabilities	**1**
High vulnerabilities	**0**
Medium vulnerabilities	**2**

1.2 TOP 3

Feature	Critical	High	Medium
/api/v1/user/admin/users	1		
/api/v1/user/tickets			1
/			1

Detailed results of vulnerability assessment and explanations are provided on [ASSESSMENT SCORE] section.

ASSESSMENT SCORE

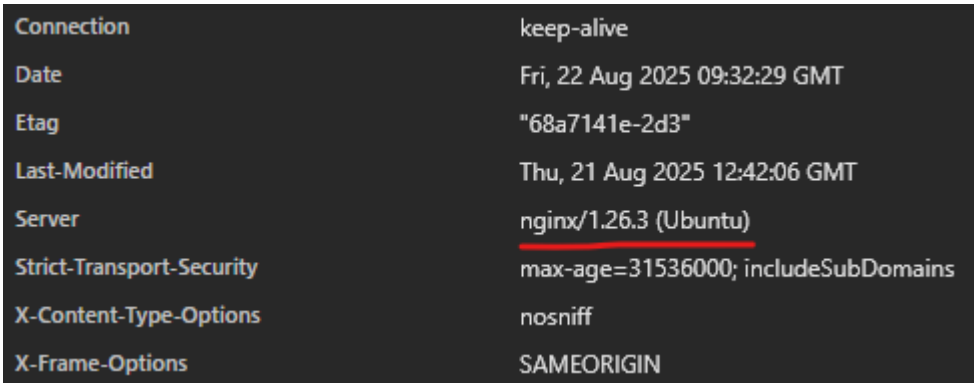
Assessment Area	Why it matters	Score
Cross Site Scripting (XSS)	Attackers can inject malicious scripts, leading to data theft or session hijacking.	100/100
SQL Injection	Allows attackers to manipulate the database, exposing or modifying sensitive data.	100/100
Broken Authentication	Password leakage, authentication token takeover, and vulnerable session management allows attackers to access another user's account.	100/100
Security Misconfiguration	Security settings are incorrect or misconfigured which potentially leads to exploits.	100/100
Directory Traversal	Allows attackers to steal sensitive information by navigating the file system on the server.	100/100
Information Disclosure	Exposure of sensitive or confidential data to unauthorized individuals due to a system vulnerability or security flaw	70/100
Privilege Escalation	An attacker exploits vulnerabilities to gain unauthorized higher-level access and control within a system or network.	50/100
Remote Code Execution (RCE)	Attackers could execute remote code on the server.	95/100

- Server Information Disclosure at /
- Order Object Information Disclosure at /api/v1/user/tickets
- Privilege Escalation at /api/v1/user/admin/users/{UUID}

Overall: 89%

Server Information Disclosure

The current server software version and operating system are being displayed without filtering. Although no critical vulnerabilities exploitable in this software have been identified at present, it is necessary to configure the system to prevent version disclosure during scans in order to avoid potential issues in future operations. (The server software version is highly sensitive information commonly used in cyber attacks. It is recommended to hide it, even forcibly if necessary.)



Connection	keep-alive
Date	Fri, 22 Aug 2025 09:32:29 GMT
Etag	"68a7141e-2d3"
Last-Modified	Thu, 21 Aug 2025 12:42:06 GMT
Server	nginx/1.26.3 (Ubuntu)
Strict-Transport-Security	max-age=31536000; includeSubDomains
X-Content-Type-Options	nosniff
X-Frame-Options	SAMEORIGIN

Nmap Scan Result:

PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	OpenSSH 9.9p1 Ubuntu 3ubuntu3.1 (Ubuntu Linux; protocol 2.0)
80/tcp	open	http	nginx 1.26.3 (Ubuntu)
• http-server-header: nginx/1.26.3 (Ubuntu)			
443/tcp	open	ssl/http	nginx 1.26.3 (Ubuntu)
• http-server-header: nginx/1.26.3 (Ubuntu)			

Mitigation Recommendations

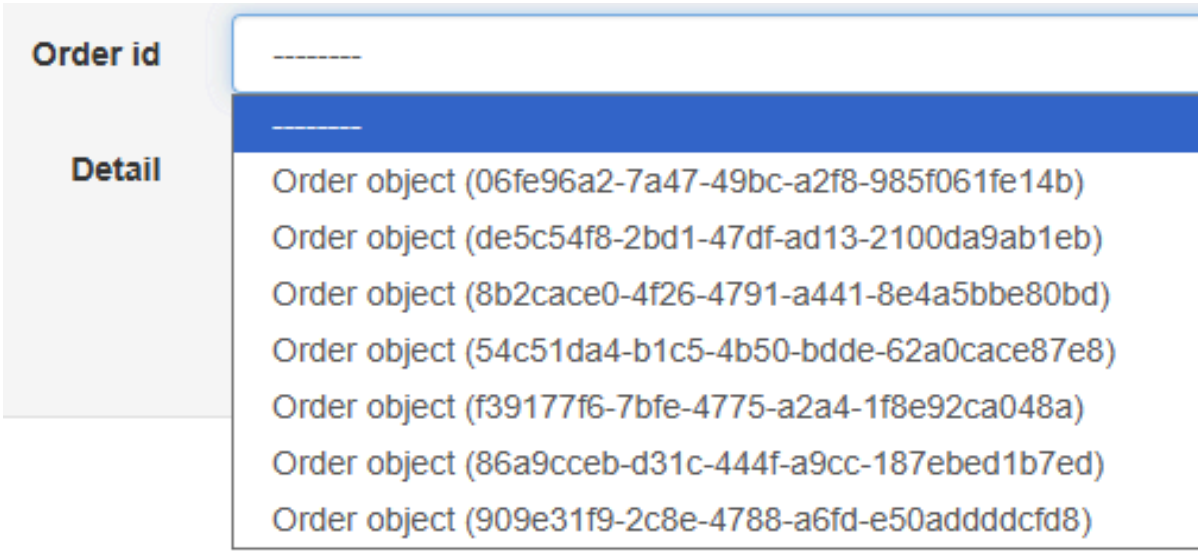
To resolve this issue, `server_tokens` must be set to **off** in the nginx configuration file within the http, server, or location block:

```
http {
    server_tokens off;
}
```

Serverfault: https://serverfault.com/questions/214242/can-i-hide-all-server-os-info

Order Object Information Disclosure

The Django Default API Router is currently exposing the UUIDs of orders requested by other users, including the current user, at `/api/v1/user/tickets`. This information could potentially be leveraged for further attacks. This attack can be carried out with basic user privileges (Student level or higher).



Mitigation Recommendations

Disable the use of Django API Default Browser and replace it with a Simple Router. Replace it using the following code:

```
router = routers.SimpleRouter()
```

Additionally, add the following settings to your Production Settings file to configure the API mapping and prevent endpoint listings:

```
REST_FRAMEWORK = {
    'DEFAULT_RENDERER_CLASSES': (
        'rest_framework.renderers.JSONRenderer',
    )
}
```

StackOverflow:
https://stackoverflow.com/questions/42829782/hide-django-rest-framework-routers-api-view-page

Privilege Escalation

A user with the **Website Technologist** role can perform a privilege escalation attack by sending a `PUT` request to `/api/v1/user/admin/users/{UUID}` and manipulating the **Role** parameter to gain **Administrator** privileges. Additionally, they can **forcibly downgrade the privileges of existing administrators**. This attack grants a user with the Website Technologist role additional configuration privileges (e.g., product creation/modification/deletion, menu creation/modification/deletion, server log access, etc.).

Mitigation Recommendations

Add an additional permission check in the server-side user role management logic to ensure that a user cannot assign a role higher than their own to another user.

CONCLUSION

Considering the structural characteristics of the school, the Assistant account is accessible to an unspecified number of users (students). **However, there does not appear to be any realistic hacking vulnerability that would allow one to directly obtain the next-level permission of a Website Technologist**. Nevertheless, if an account with that permission is compromised through social engineering techniques or unknown vulnerabilities, **it could ultimately lead to full control of the website due to the API's flawed access control**.