Security Vulnerability Report – Stored XSS via CSV Import (Update Existing Records) in ERPNext / Frappe Framework

Severity: High

CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:H/VA:N/SC:N/SI:N/SA:N

Description

A Stored Cross-Site Scripting (XSS**)** vulnerability was discovered within the CSV import mechanism of ERPNext when using the **"**Update Existing Records**"** option.

An attacker can embed malicious JavaScript code into a CSV field, which is then stored in the database and executed whenever the affected record is viewed by a user within the ERPNext web interface.

This exposure may allow an attacker to compromise user sessions or perform unauthorized actions under the context of a victim's account.

Affected Environment

- Repository: https://github.com/frappe/frappe_docker
- Deployment Method: Default Docker setup
- ERPNext / Frappe Version: last version on 16 - 17 Nov 2025
- Browser Used During Testing: Google chrome Version 142.0.7444.135 (Official Build) (64-bit)

Technical Details

Attack Vector:

A crafted CSV file containing specially formatted HTML/JavaScript is imported into a target Doctype or content in malicious file.

```
<script>alert(document.cookie)</script>
```

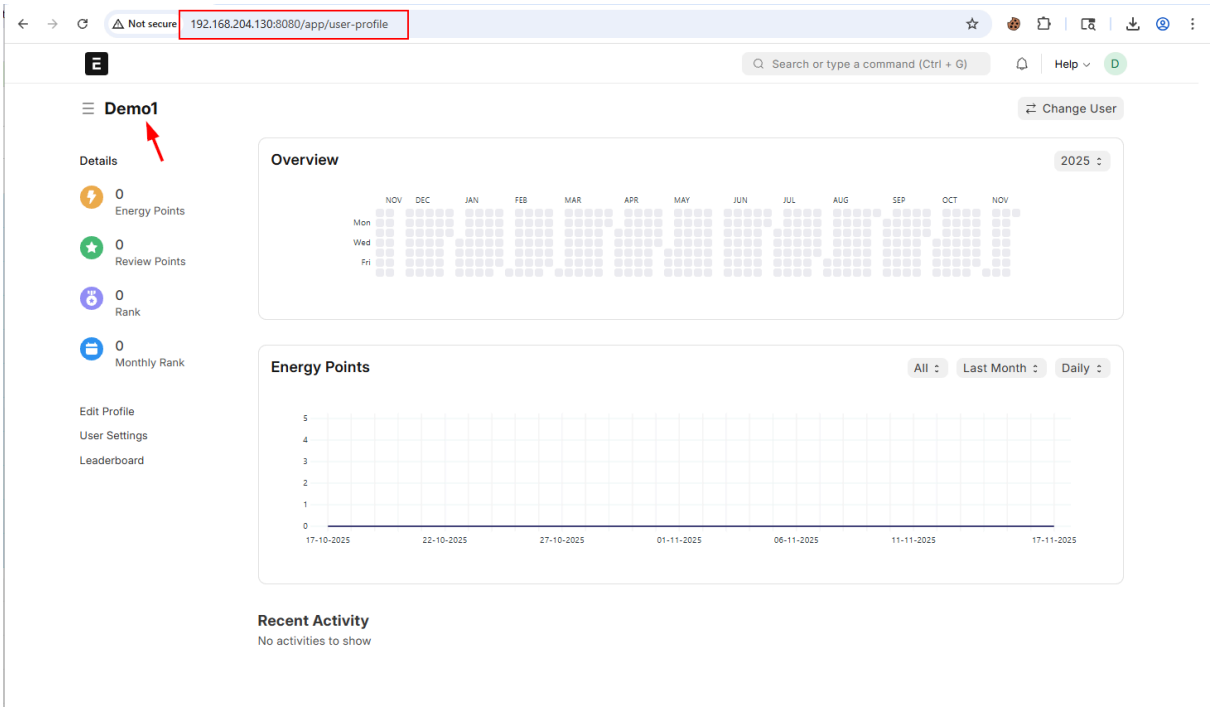
Impact:

Successful exploitation results in Stored XSS, which may lead to:

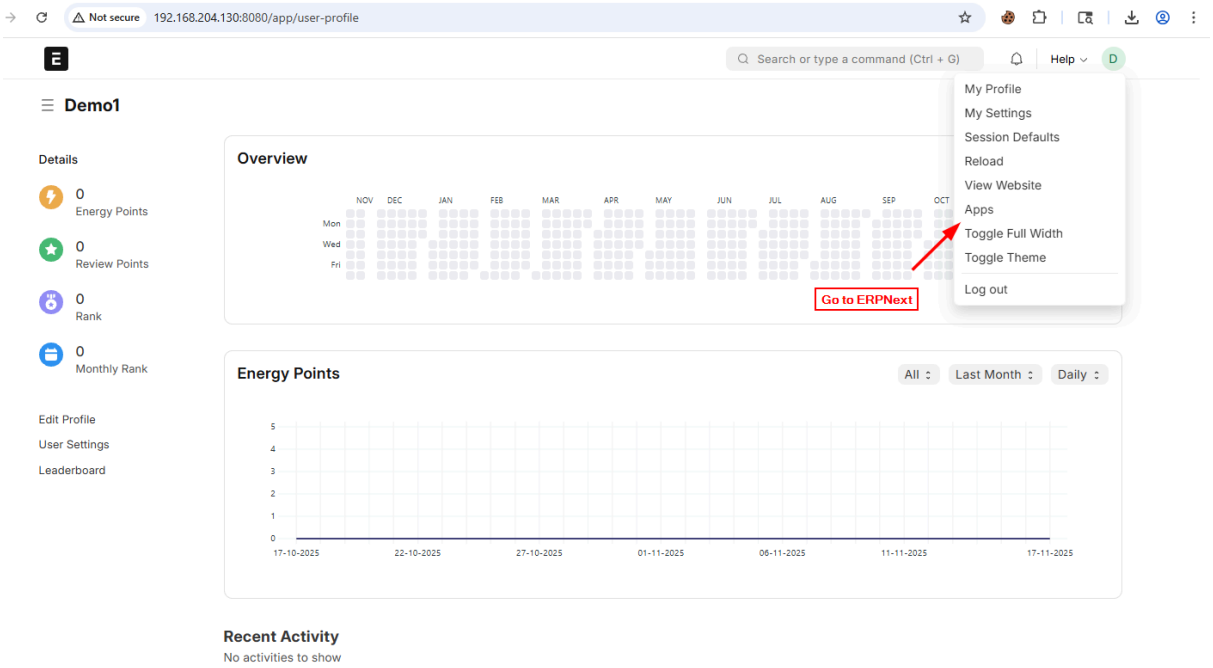
- Execution of arbitrary JavaScript in a victim's browser
- Privilege escalation through session hijacking
- Unauthorized data manipulation
- Potential takeover of administrative accounts
- Compromise of ERPNext workflows and user integrity

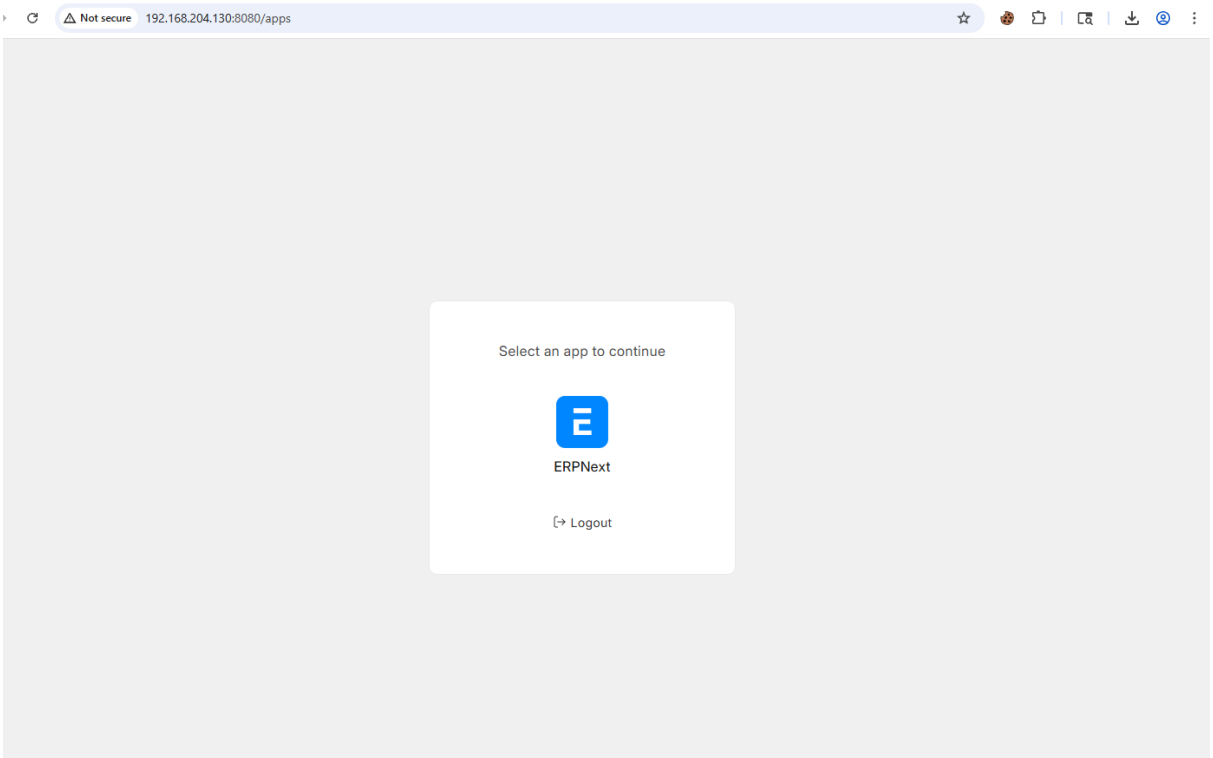
Proof of Concept

1. We have logged into the system.

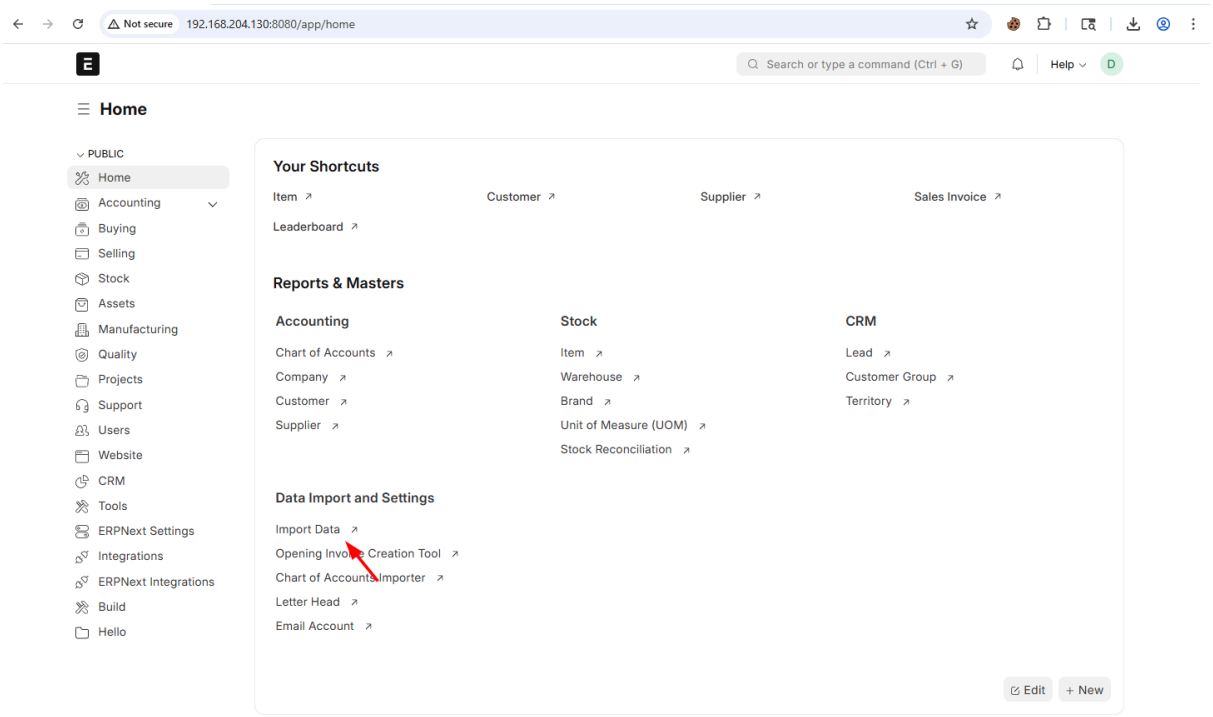


2. We navigated to the application menu to be redirected to ERPNext.

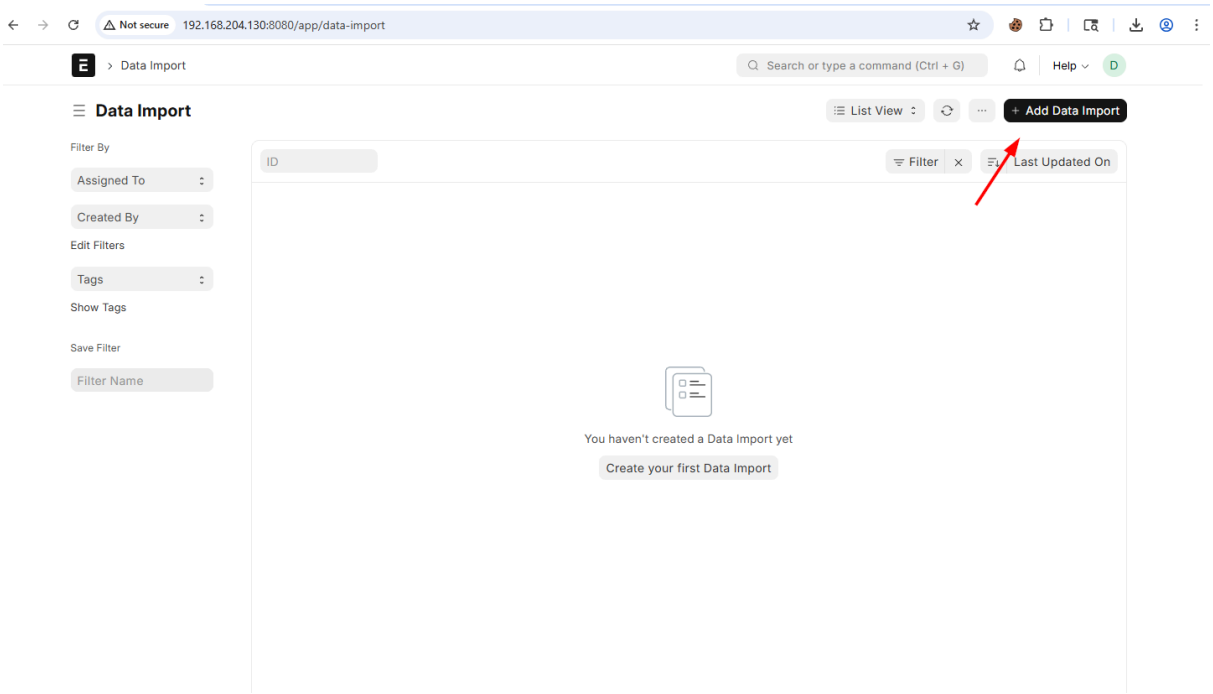




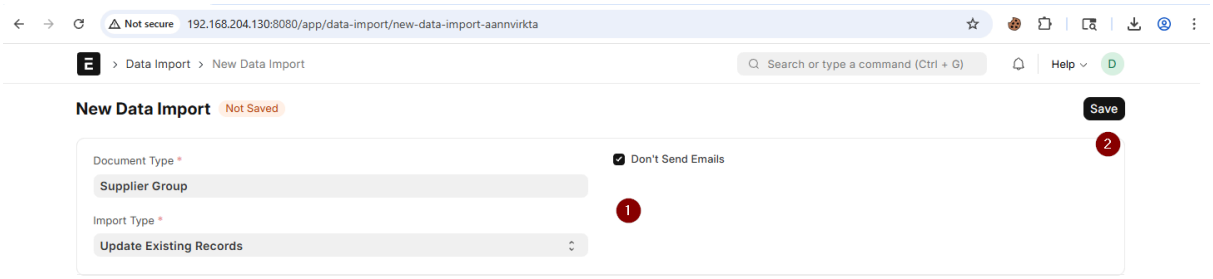
3. Then we proceeded to the ‘Import Data’ menu.



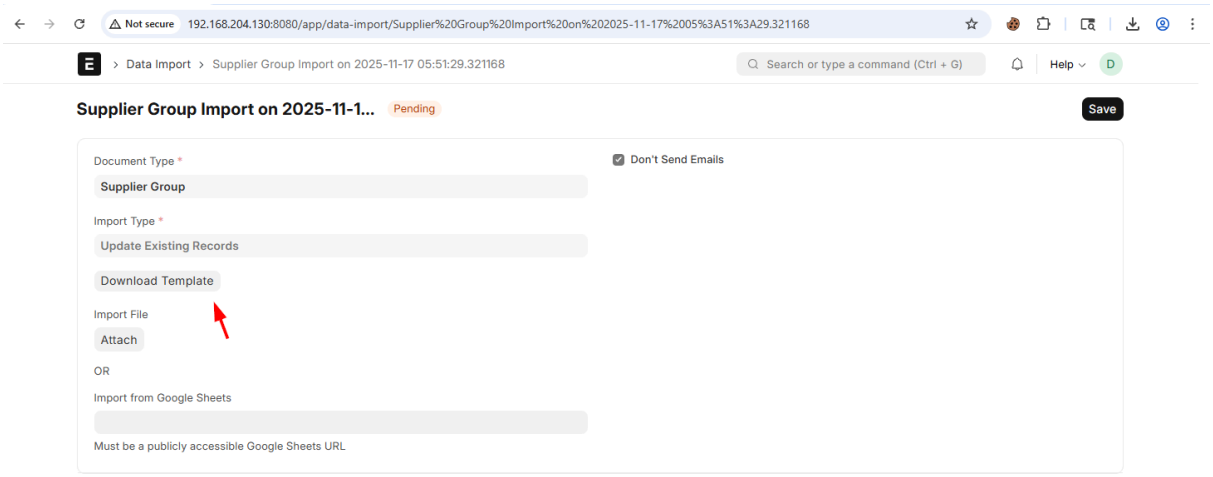
4. Add a data import.



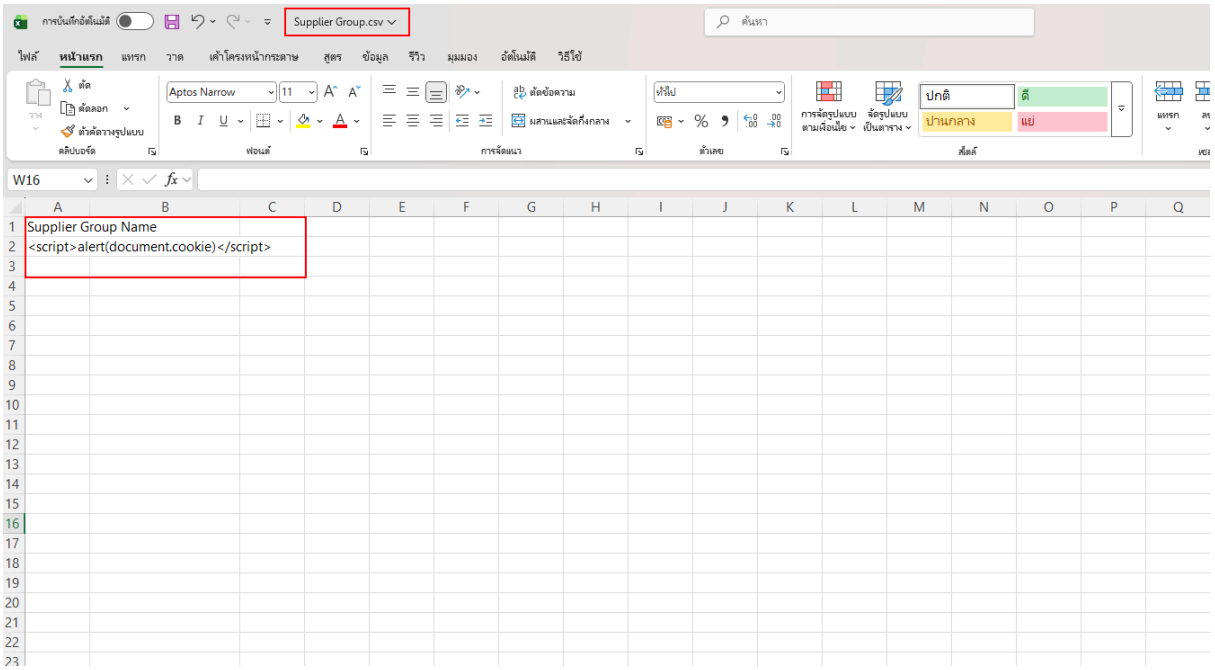
5. Next, select the data type as *Supplier Group* and set the insertion type to *Update Existing Records*, then click *Save*.



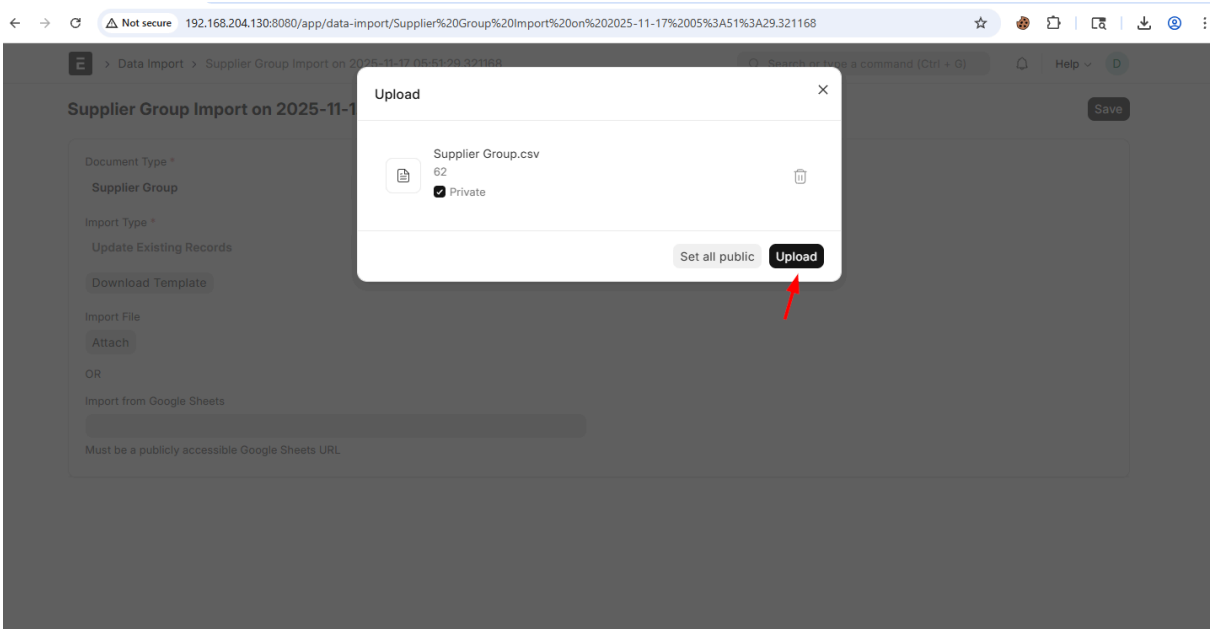
6. We downloaded the example template to review the structure.



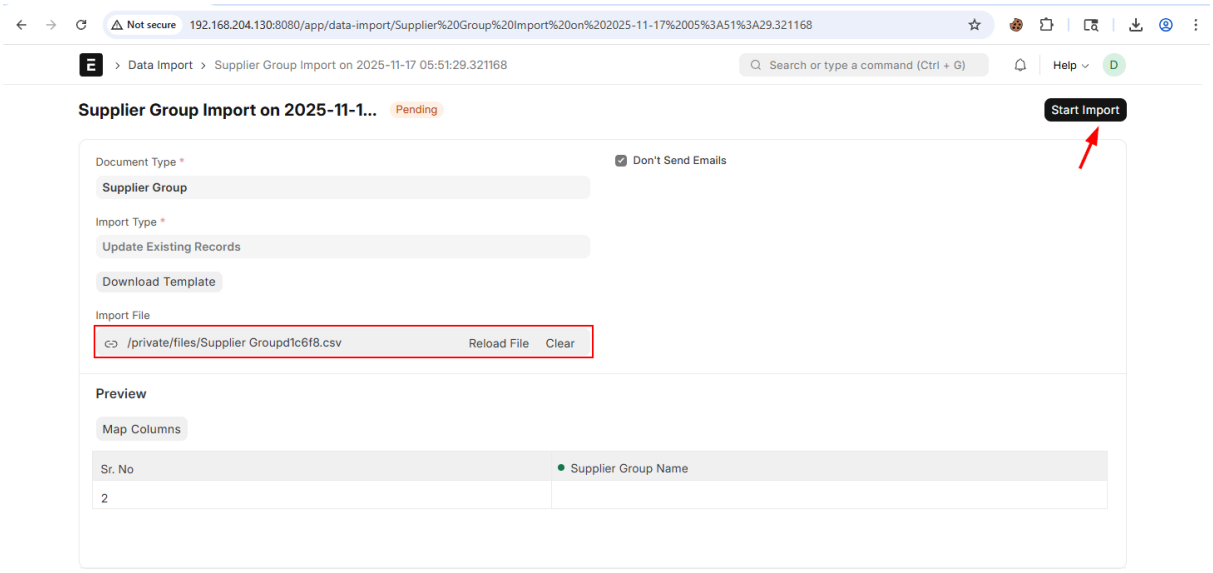
7. We attempted to insert a malicious script (XSS) into the CSV file.



8. Then we clicked ‘Upload’ and the file was successfully uploaded.



9. We clicked ‘Start Import’.



10. The website executed the malicious script.