# XSS Attack Demonstration – Cyber Security Project

## 📍 Before Fix – Vulnerable Version

In this version, the system is vulnerable to a stored XSS attack.  
A malicious client was added with the following name:

<script>alert('XSS')</script>

When fetching clients via /list-clients, the malicious script is returned as-is:

{  
 "name": "<script>alert('XSS')</script>",  
 "sector": "Finance"  
}

This would result in a JavaScript alert being executed in a real browser.

## ✅ After Fix – Protected Version

In the secured version, html.escape() is applied before returning user input.  
The same input now produces a safe output:

{  
 "name": "&lt;script&gt;alert('XSS')&lt;/script&gt;",  
 "sector": "Finance"  
}

This ensures that the code is rendered as text and not executed.

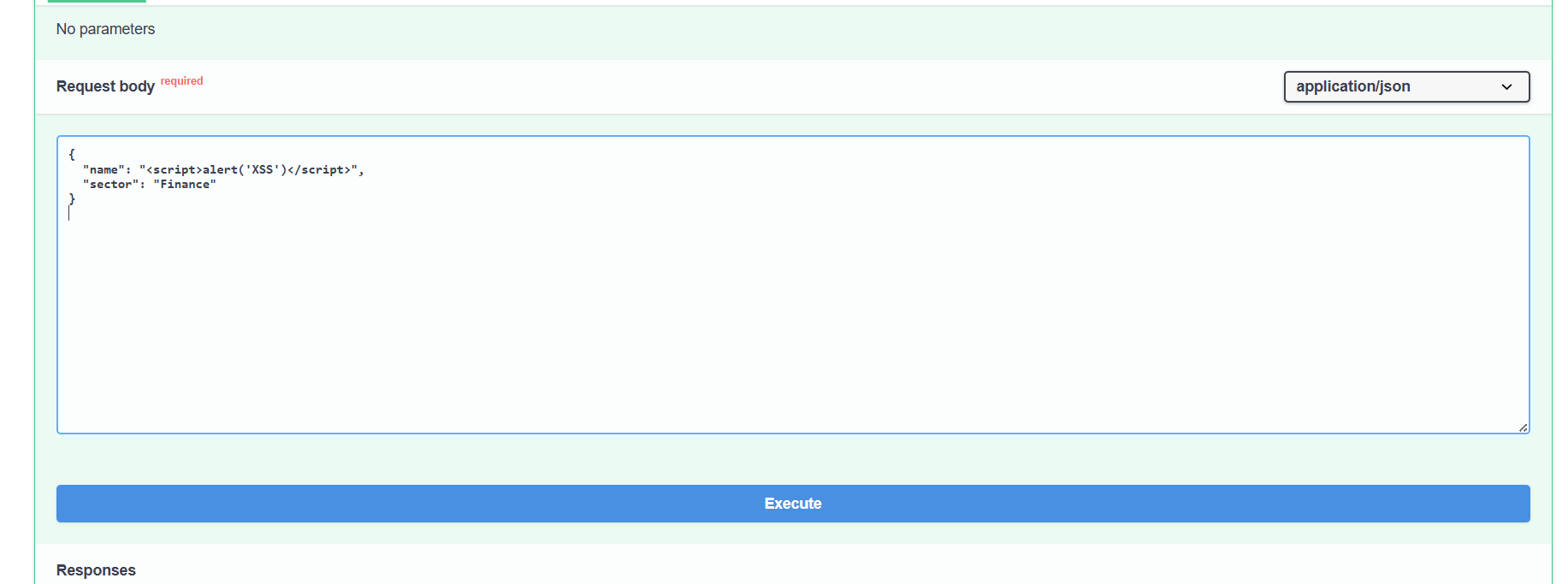
## 🧠 Summary

This demonstration proves how unsanitized user input can lead to a critical XSS vulnerability.  
Using html.escape() or output encoding effectively neutralizes the threat.

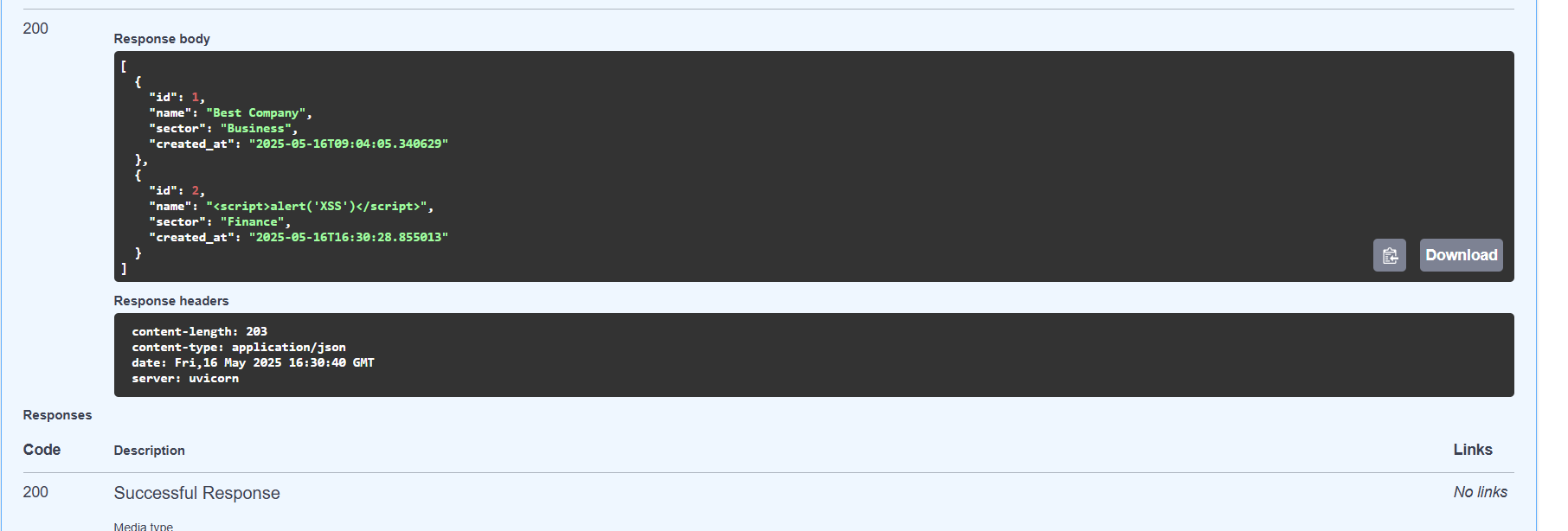
## 📸 Screenshots

The following images demonstrate the XSS attack in both vulnerable and protected states.

1️⃣ Submitting a malicious client with XSS code:



2️⃣ Response from vulnerable system (before escaping):



3️⃣ Response from protected system (after applying html.escape):

