Parameter Tampering

What is Parameter Tampering

Parameter tampering is the act of modifying values sent between the client and server to manipulate application behaviour.
For example, on an e-commerce website, if a product is priced at ₹500, an attacker could change that value to ₹1 through intercepted requests.

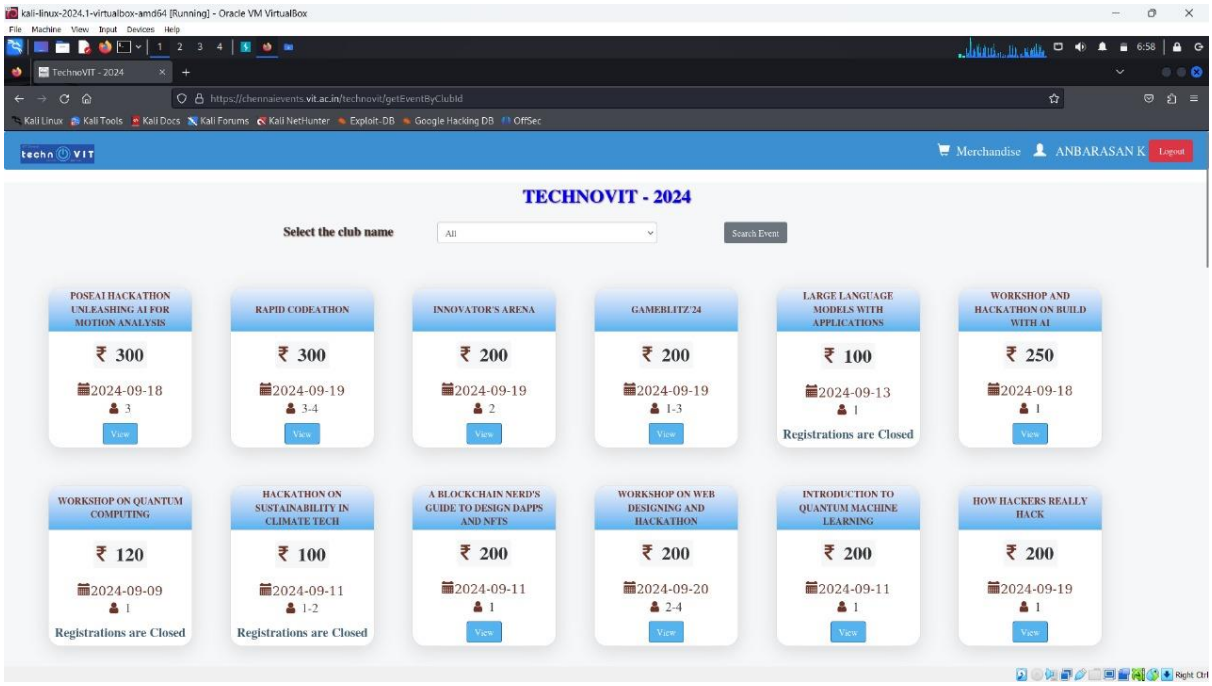
Real Bug Discovery (College Event Example)

I initially discovered this vulnerability and tested it using a friend's account. The exploit worked successfully, allowing me to manipulate the payment amount. However, I did not capture a screenshot at the time for proof of concept. Four days later, I attempted the same process using my own account with the intent of documenting it, but by then the bug had been patched. The transaction failed and displayed an error message indicating an "incorrect hash parameter," confirming that the vulnerability had been fixed.

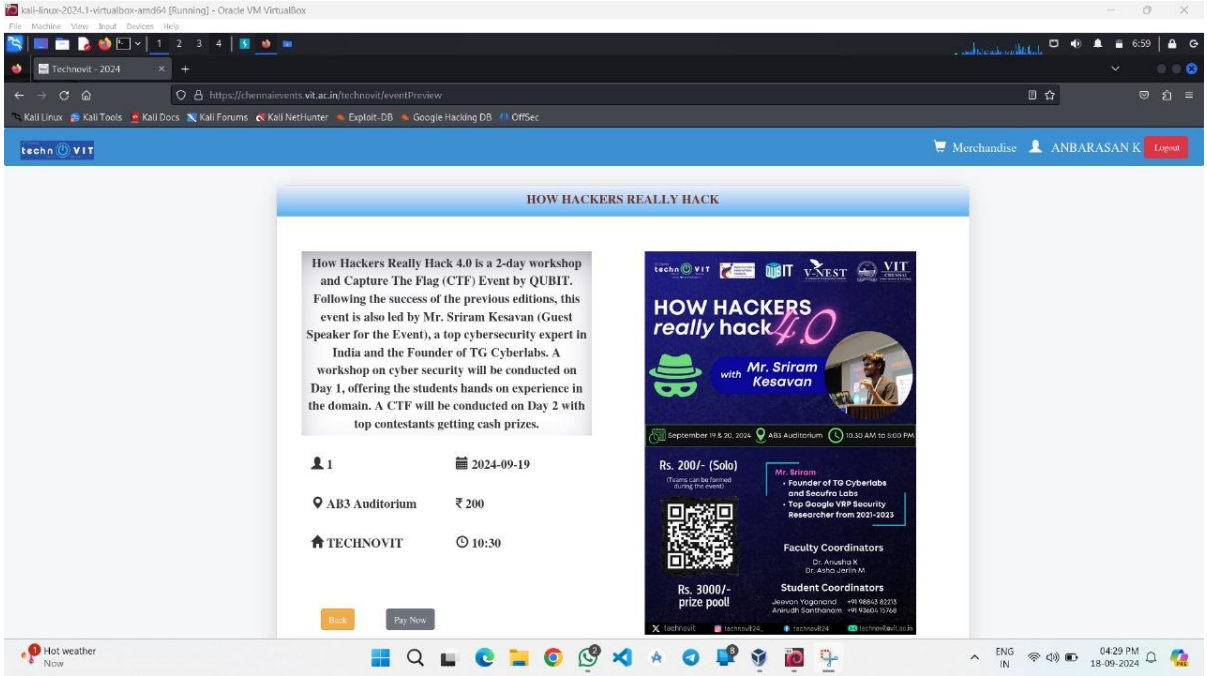
Although the vulnerability has since been fixed, I will walk through the steps I initially performed by logging in with my own account. While I was unable to capture a full proof of concept due to the patch, I can still demonstrate the manipulated result — specifically, the receipt showing the altered payment value.

Step-by-Step Walkthrough

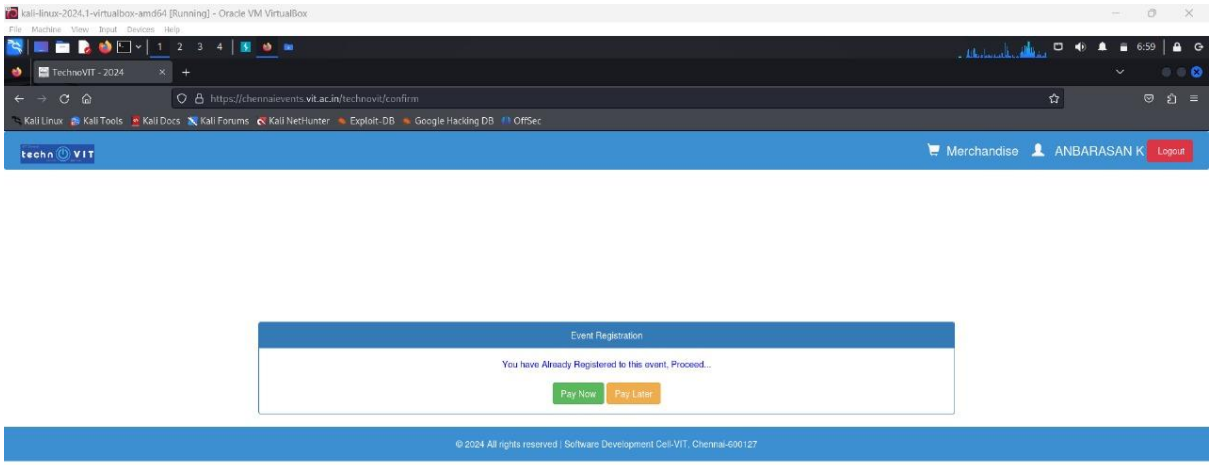
1. Actual website with my account logged in

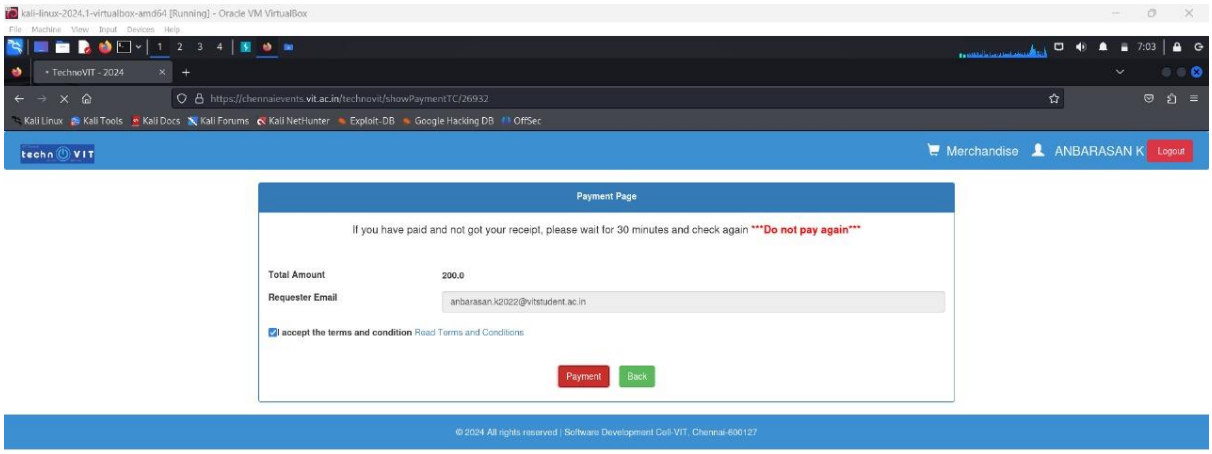


2. This was the targeted event which I will perform the parameter tampering



3. Way to payment

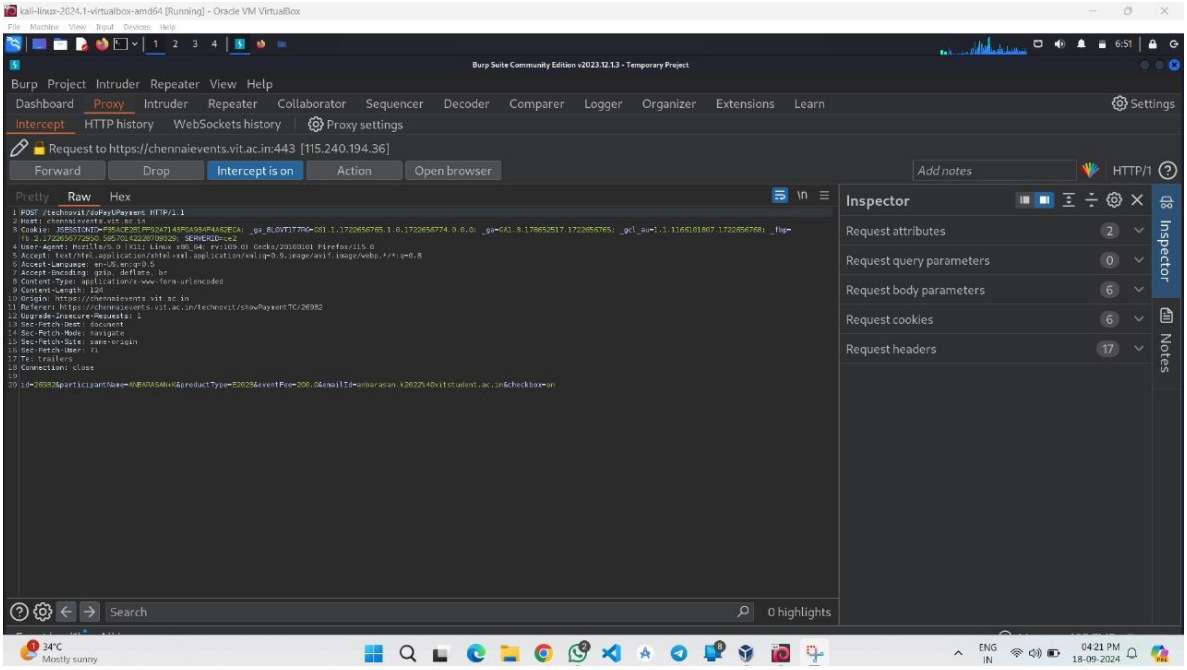




Tool Used: Burp Suite

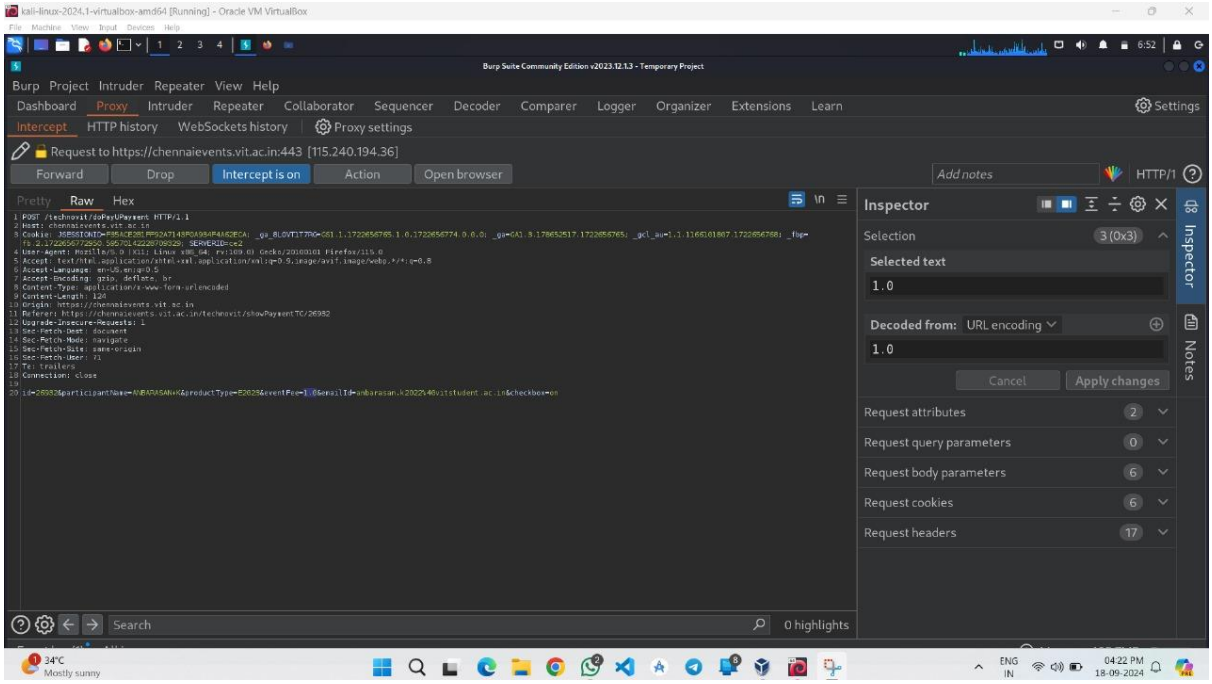
All request interceptions and parameter modifications were performed using **Burp Suite**, a powerful web vulnerability scanner and proxy tool that allows real-time interception and modification of HTTP requests and responses.

4. This is how the actual web page looks in backend, using Burp Suite we can see backend of the web page



In this at the last line there is a variable named eventFee with value 200.0 (eventFee=200.0), this is the value we are looking for

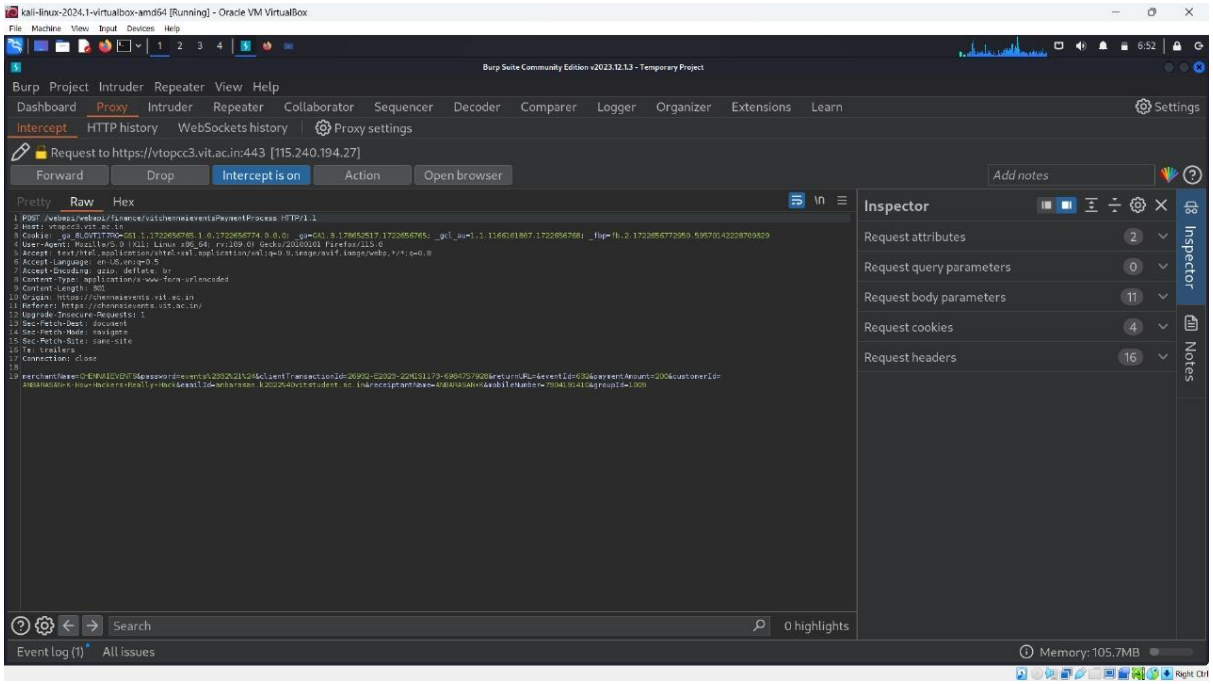
5. Now we are going to manipulate this value



The value is changed from eventFee=200.0 to eventFee=1.0

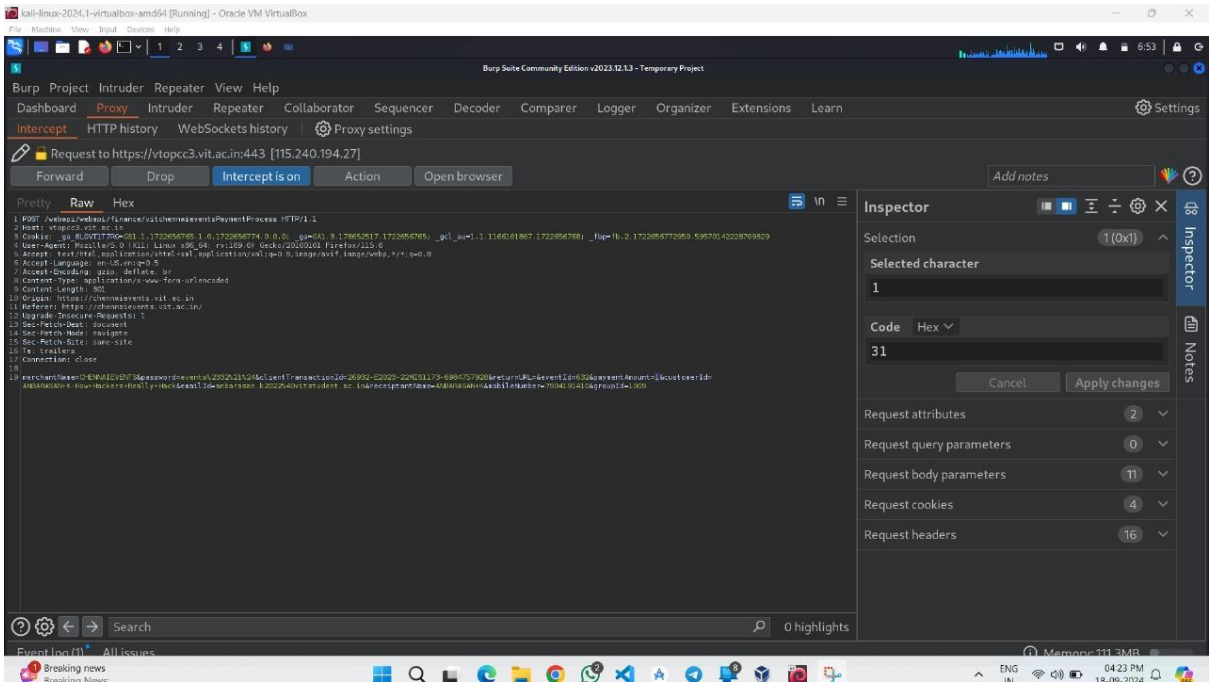
6. Similarly, we need to change the value for the upcoming pages

(Actual page)



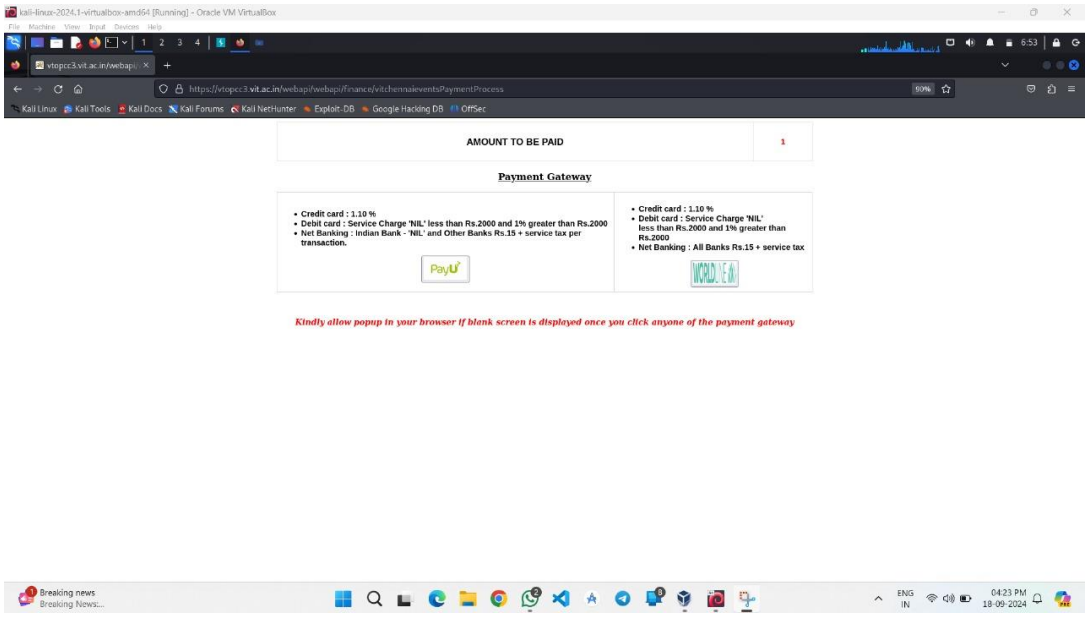
paymentAmount = 200

(Manipulated page)



PaymentAmount = 1

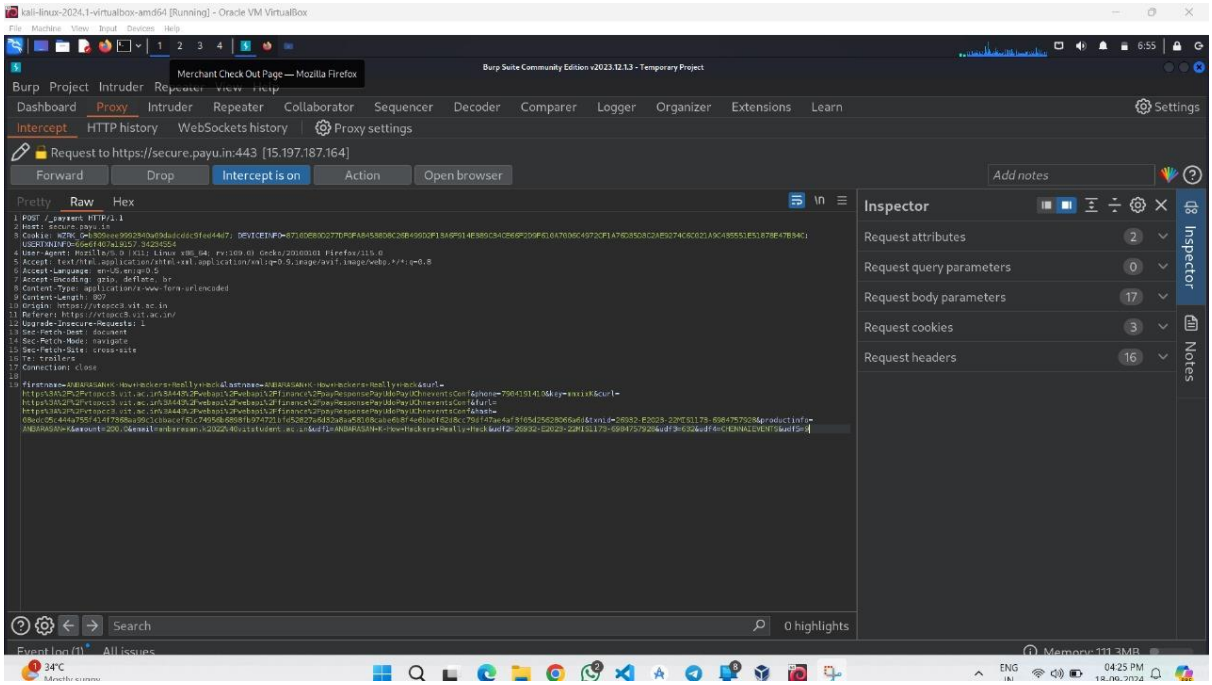
7. By doing these steps we have finally made the changes in the frontend of the web page



AMOUNT TO BE PAID 1

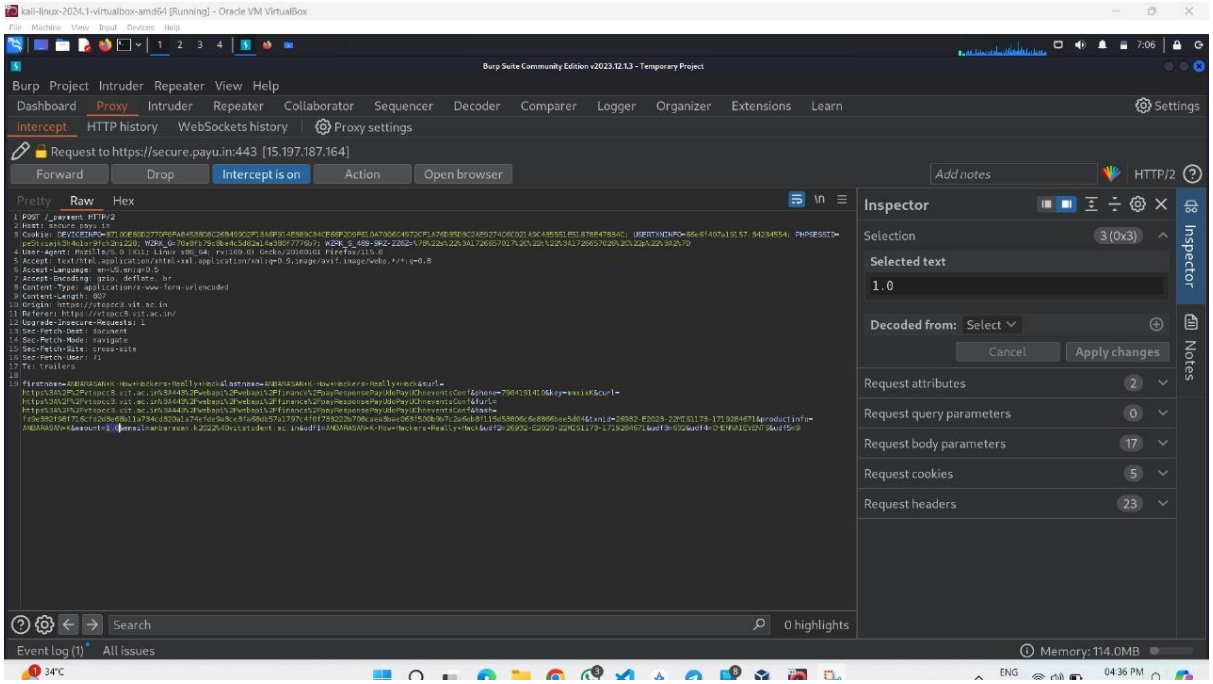
8. Now we need to select the payment gateway and do same steps as we have discussed

   (Actual page)



amount=200.0

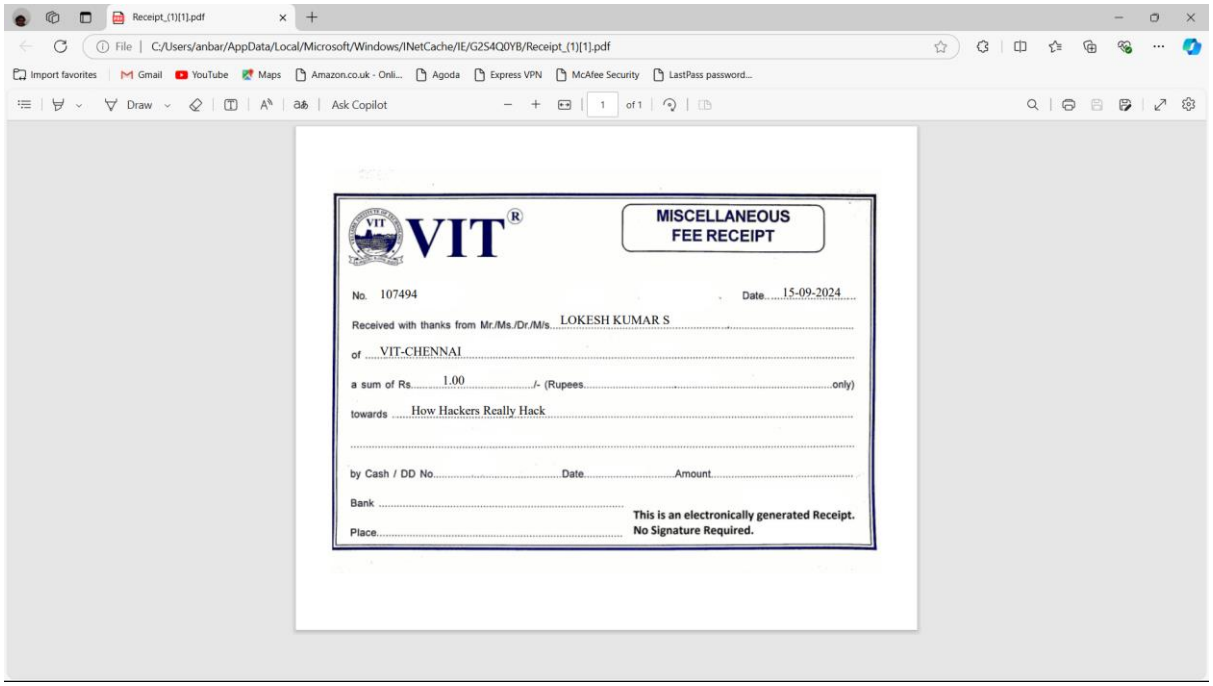
(Manipulated page)



amount = 1.0

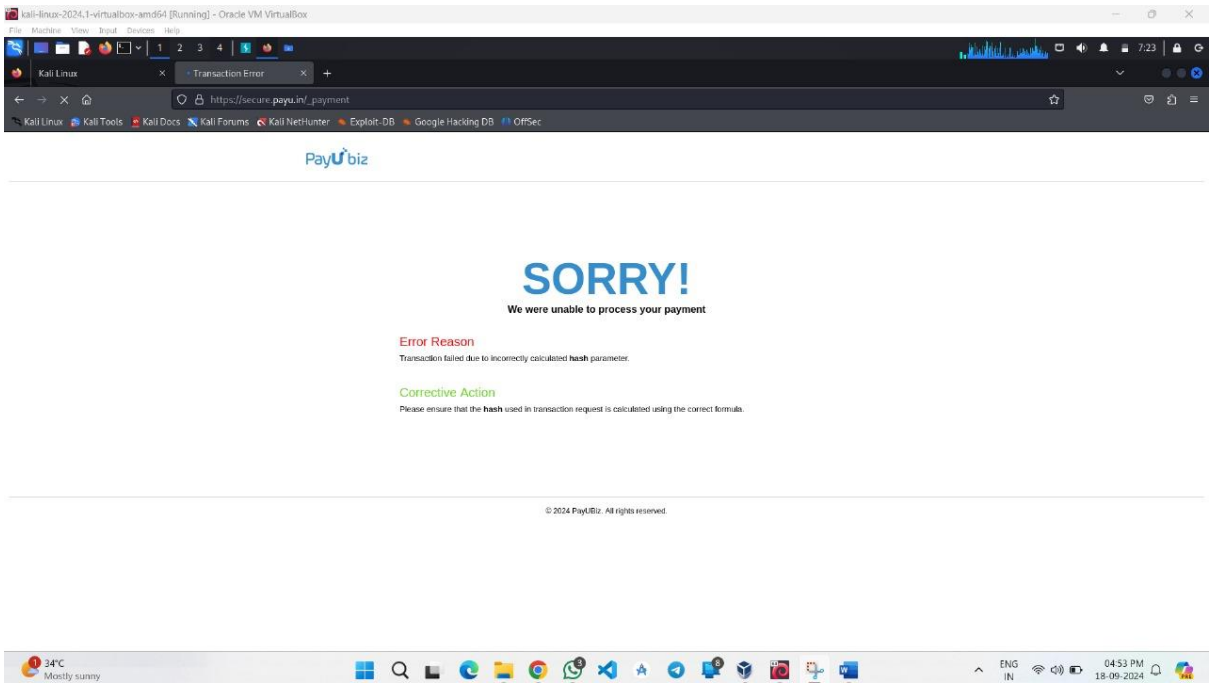
Result / Receipt

Finally, the amount has changed and we can pay for 1 rupee instead of 200 rupees and this is the generated receipt for the event



Fix Observation

As I mentioned early when I tried this bug after 4 days the bug does not work and I got an error message too



This means they have fixed the bug and this bug will no longer works in this site

Conclusion & Takeaways

This real-world experience provided a valuable insight into how insecure client-side parameter handling can lead to serious vulnerabilities like parameter tampering. By manipulating values in payment requests, I was able to modify the transaction amount successfully, demonstrating how critical it is to implement robust server-side validation.

Although the issue was later fixed by the developers, this situation emphasized the importance of continuous security testing and proper handling of sensitive data.