Bypass testing in Web Application (Client Side)

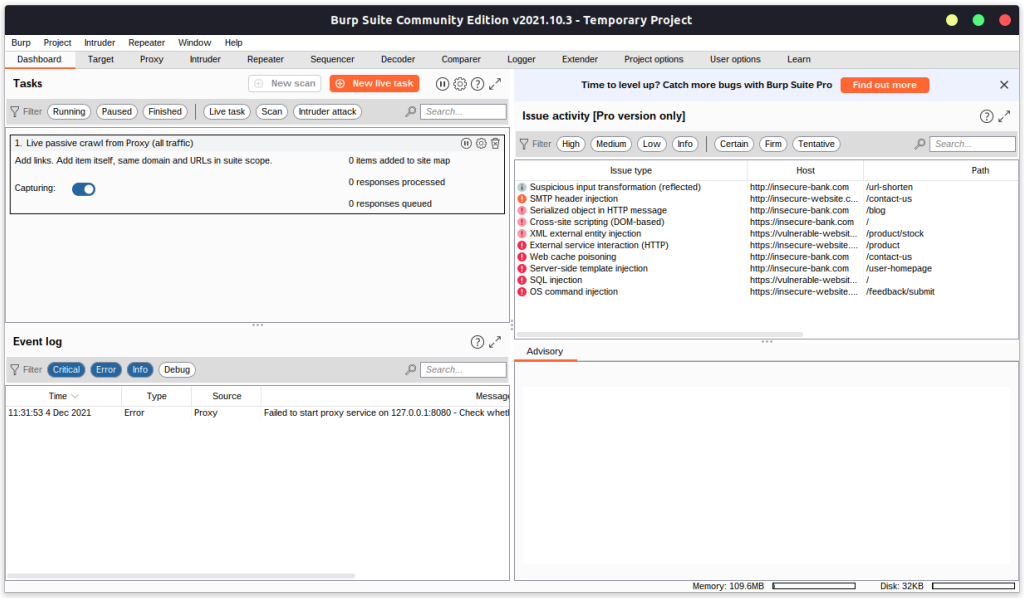
Client Side Bypass Testing

Bypass testing is the act of circumventing the checks and validations at the client side to send the unexpected input values to the server and observe it's behavior.

Tools Used

Bypass Test Tool: BurpSuite

BurpSuite: Burp will be used to capture the response sent by WebGoat and circumvent the client side javascript validations.

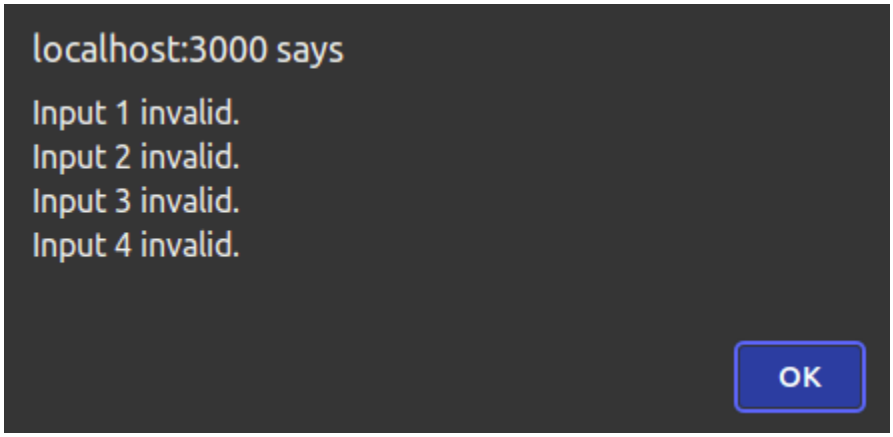


Input Validation Test Cases

Test Case 1: Input string of length 3
Test Case 2: Input small alphabetic character string [a-z]
Test Case 3: Input digit character string [0-9]
Test Case 4: Input only capital alphabetic character string [A-Z]

Test Cases	Valid Inputs	Invalid Inputs
Test Case 1	abc	abcasd
Test Case 2	string	string12
Test Case 3	1234	1234s
Test Case 4	STRING	STRING12

Client Side Response before Bypassing the input validations -



Bypassing the Input Validation -

Now we will intercept the request sent by the client to the server after the validation of the inputs at the client side.

Intercepting the request -

First we will send the valid inputs to bypass the client side javascript validations and then change the intercepted request to send the invalid inputs.

Intercepted Request -

```
POST /form HTTP/1.1
Host: localhost:3000
Content-Length: 51
Cache-Control: max-age=0
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Upgrade-Insecure-Requests: 1
Origin: http://localhost:3000
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apr
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:3000/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close

input1=abc&input2=string&input3=12345&input4=STRING
```

Now we can modify the fields to send the invalid inputs to the server

Previous Inputs in the Request -
input1=abc&input2=string&input3=12345&input4=STRING

Modified Request with Invalid Inputs -
input1=abcasd&input2=string12&input3=12345s&input4=STRING1

Now send the modified request to the server and that's how we can bypass client side validations.