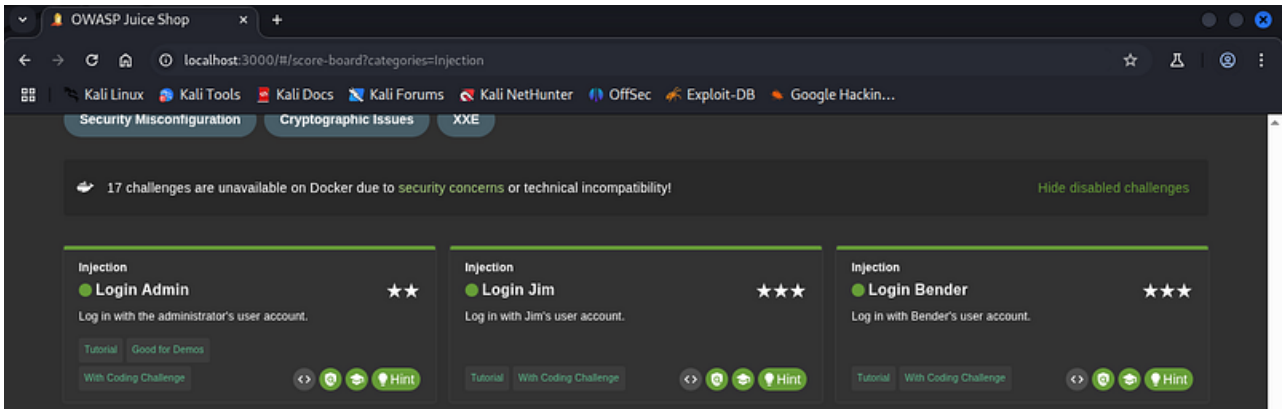
OWASP Juice Shop Challenges Writeup

Category

SQL Injection

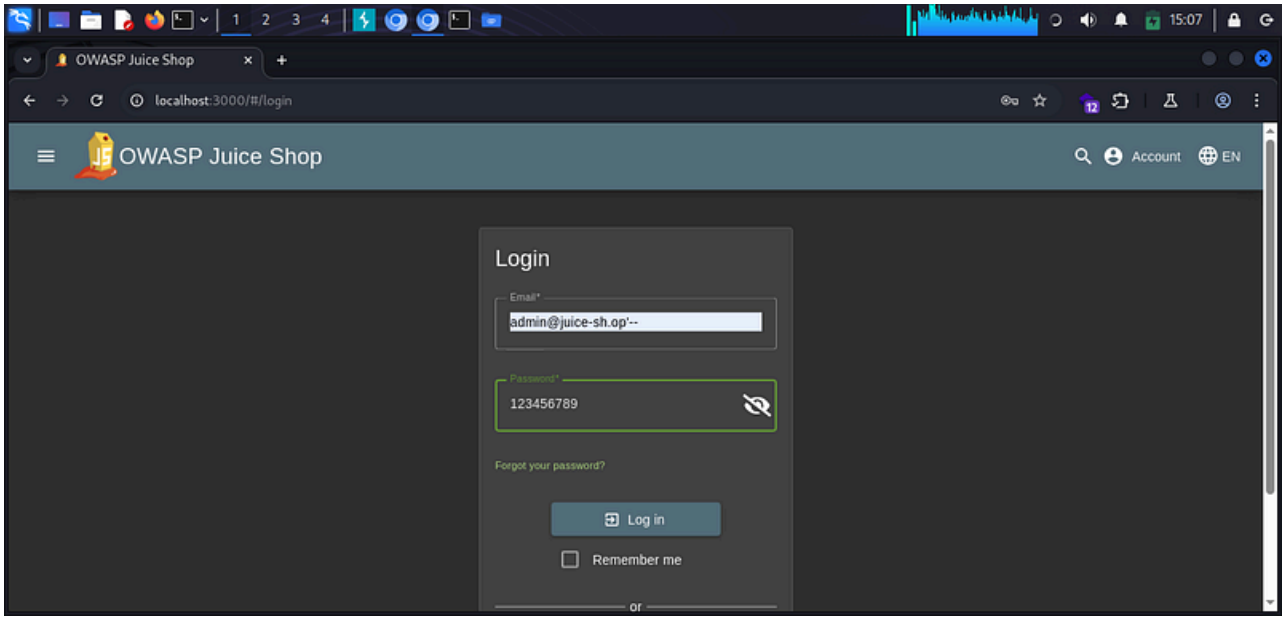


Tested the login page for SQL injection by using a SQL comment ('--). I found usernames while browsing product **reviews** on the site, used one of those usernames on the login form as:

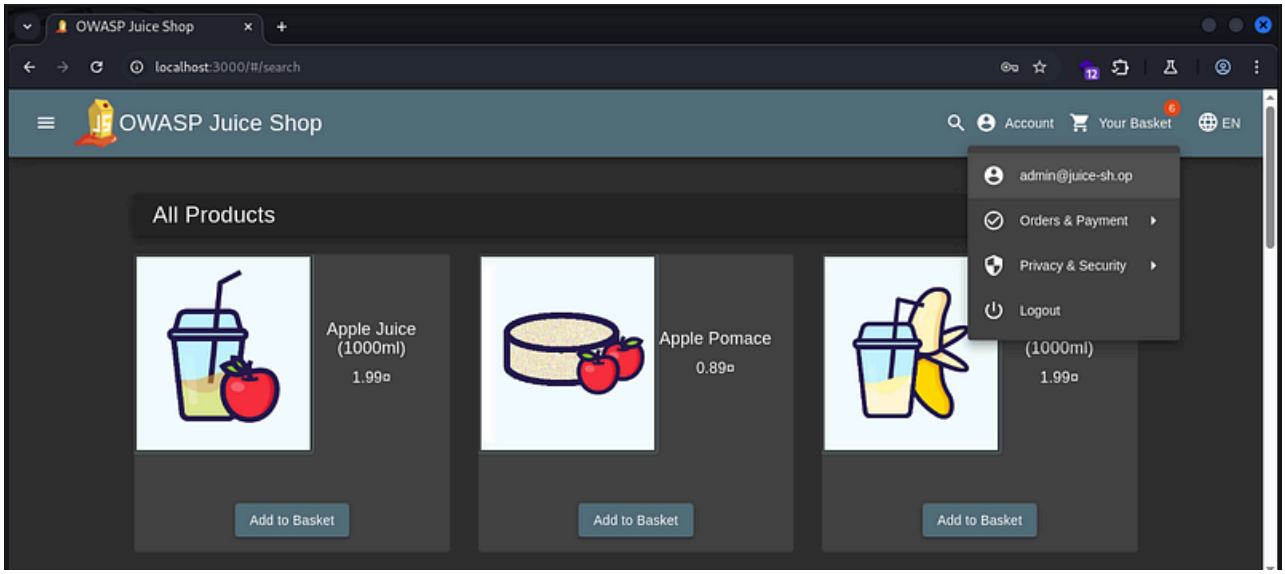
<found_username>'--

-- starts a SQL comment, so the backend query becomes something like:

SELECT * FROM users WHERE username='found_username'--' AND password='...';

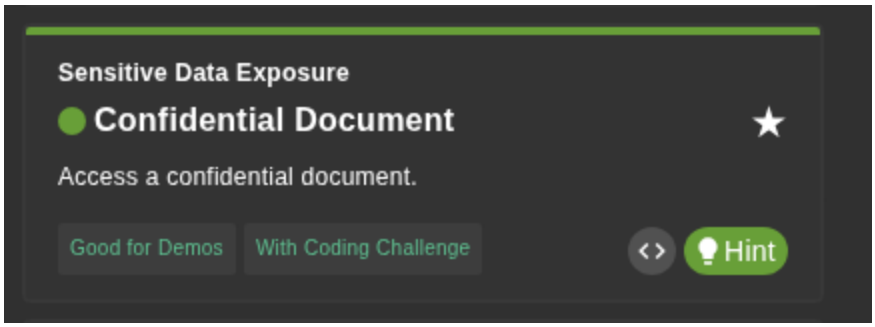


so you can enter anything in the password field)—the AND password=... portion is commented out, so the password check is skipped. Successful login occurred using the commented username input.

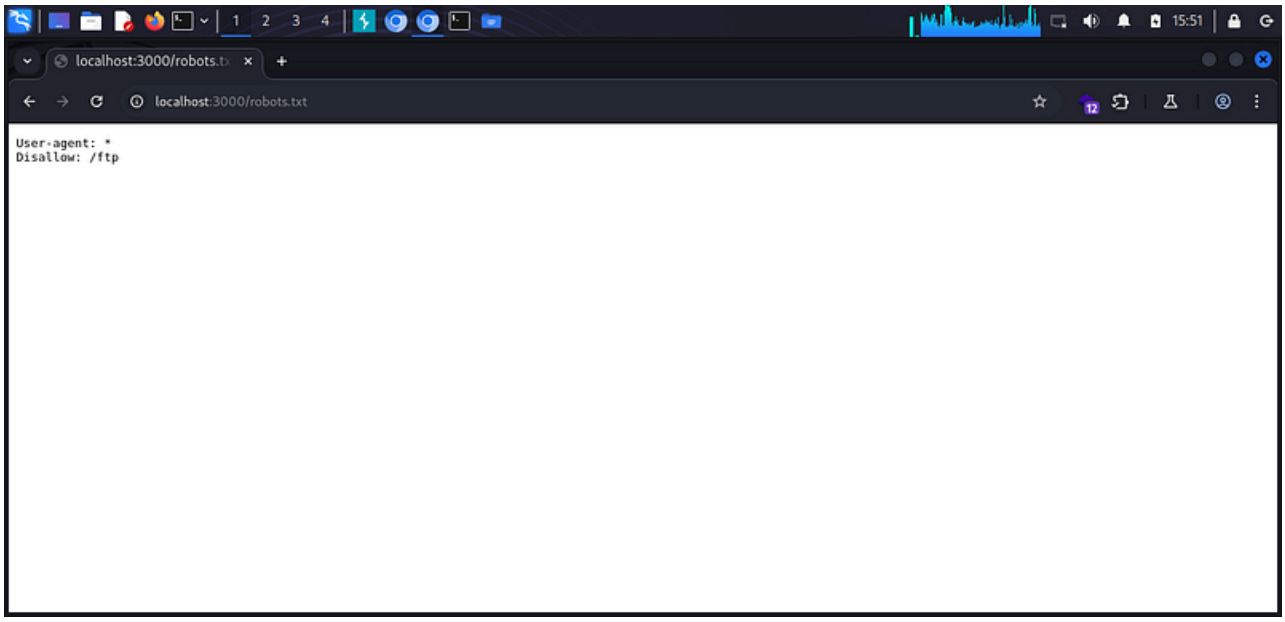


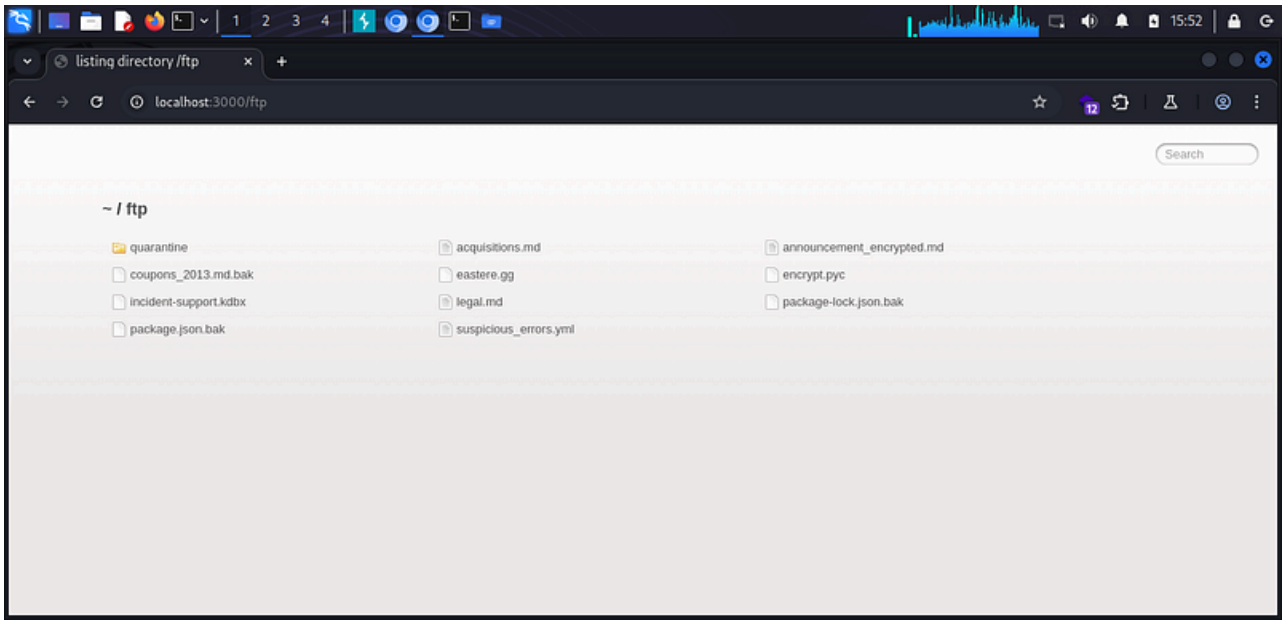
Category

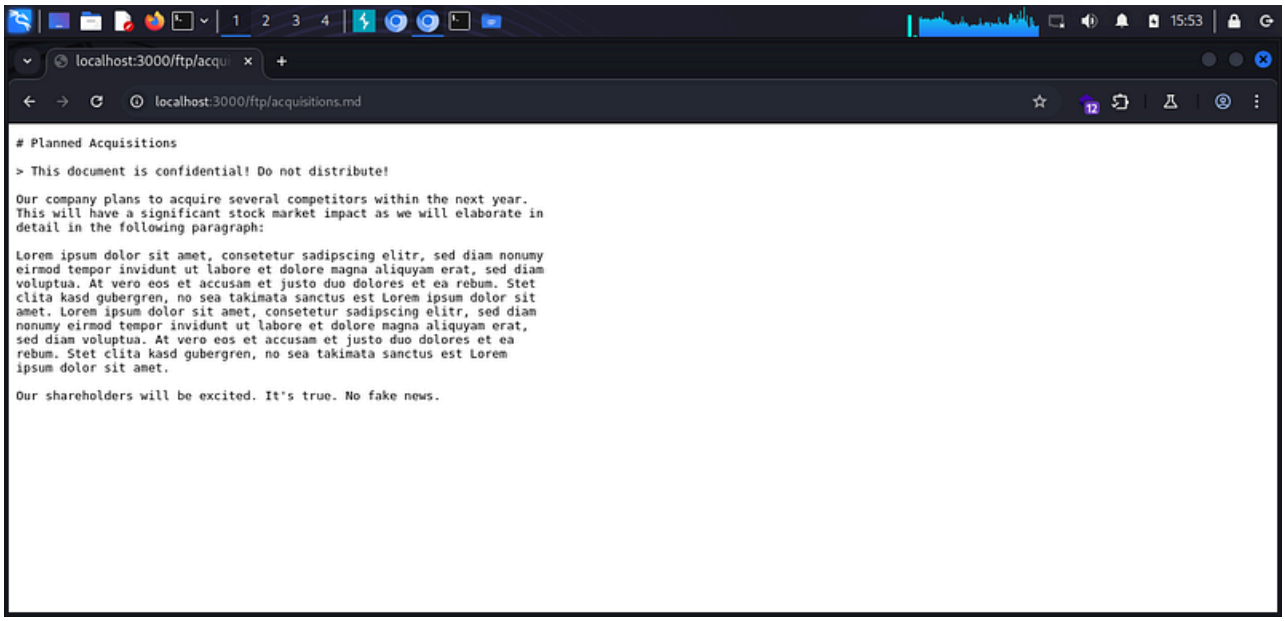
Sensitive Data exposure



robots.txt is a public file that tells search engines which paths to avoid crawling—it can accidentally reveal hidden or sensitive endpoints, so it's a quick thing to check. I found a reference to /ftp there, visited /ftp, and inspected the directory contents. Inside I discovered an **acquisition** file that contained sensitive information and was retrievable . I was able to access/download that file.

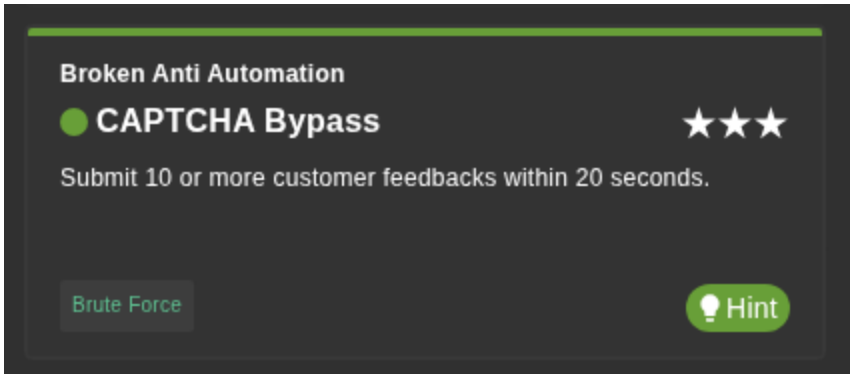

localhost:3000/robots.txt
localhost:3000/robots.txt
User-agent: *
Disallow: /ftp


listing directory /ftp
localhost:3000/ftp
Search
~ / ftp
quarantine
acquisitions.md
announcement_encrypted.md
coupons_2013.md.bak
eastere.gg
encrypt.pyc
incident-support.kdbx
legal.md
package-lock.json.bak
package.json.bak
suspicious_errors.yml


localhost:3000/ftp/acquisitions.md
Planned Acquisitions
> This document is confidential! Do not distribute!
Our company plans to acquire several competitors within the next year.
This will have a significant stock market impact as we will elaborate in
detail in the following paragraph:
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy
eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam
voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet
clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit
amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam
nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat,
sed diam voluptua. At vero eos et accusam et justo duo dolores et ea
rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem
ipsum dolor sit amet.
Our shareholders will be excited. It's true. No fake news.

Category

Broken Anti Automation _Brute Force



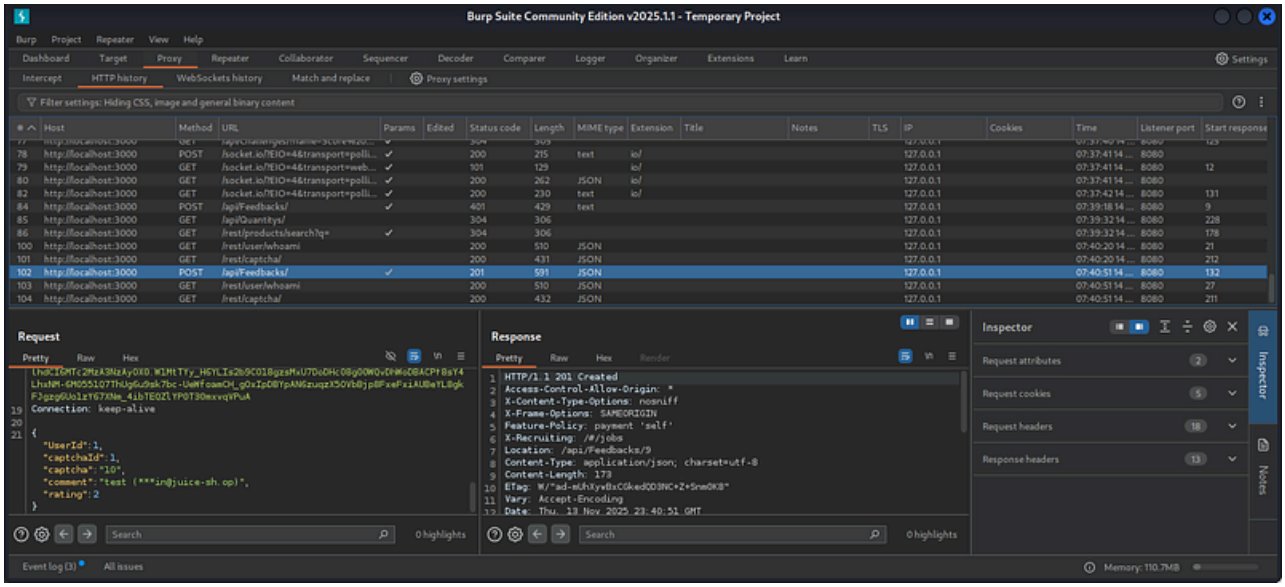
CAPTCHA Bypass Submit 10 or more customer feedbacks within 20 seconds.

Tool used: Burp Suite + Intruder

I navigated to the Customer Feedback section and submitted a message to capture the request.

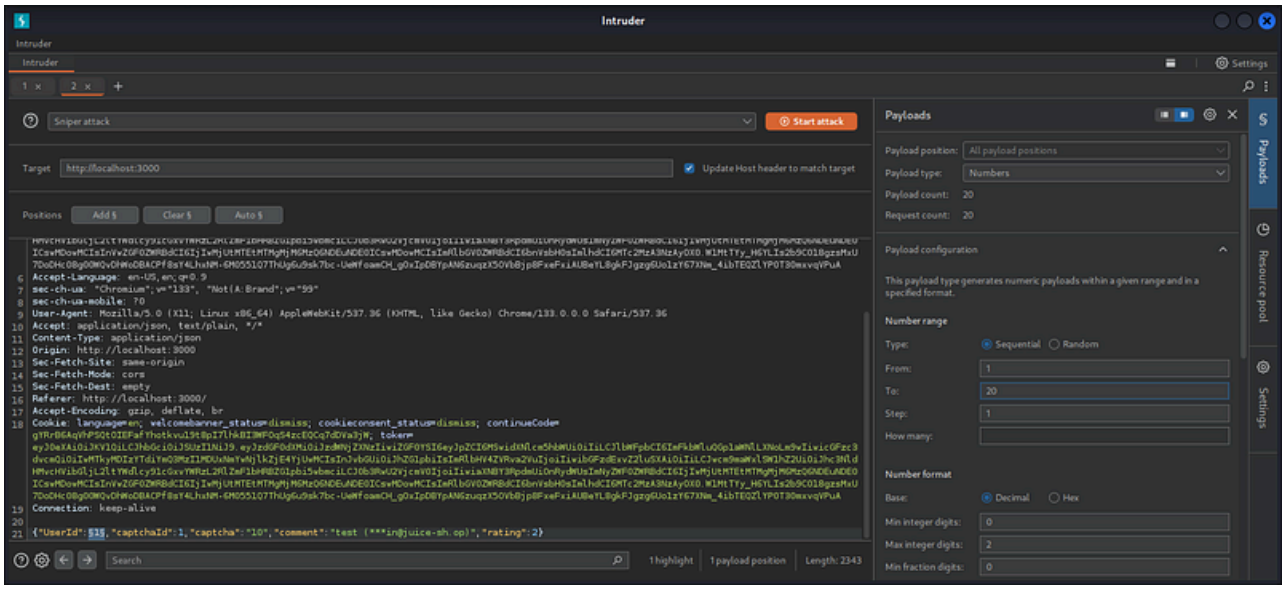
```
POST /api/Feedbacks/ HTTP/1.1

{
 "UserId":1,
 "captchaId":1,
 "captcha":"10",
 "comment":"test (***in@juice-sh.op)",
 "rating":2
}
```

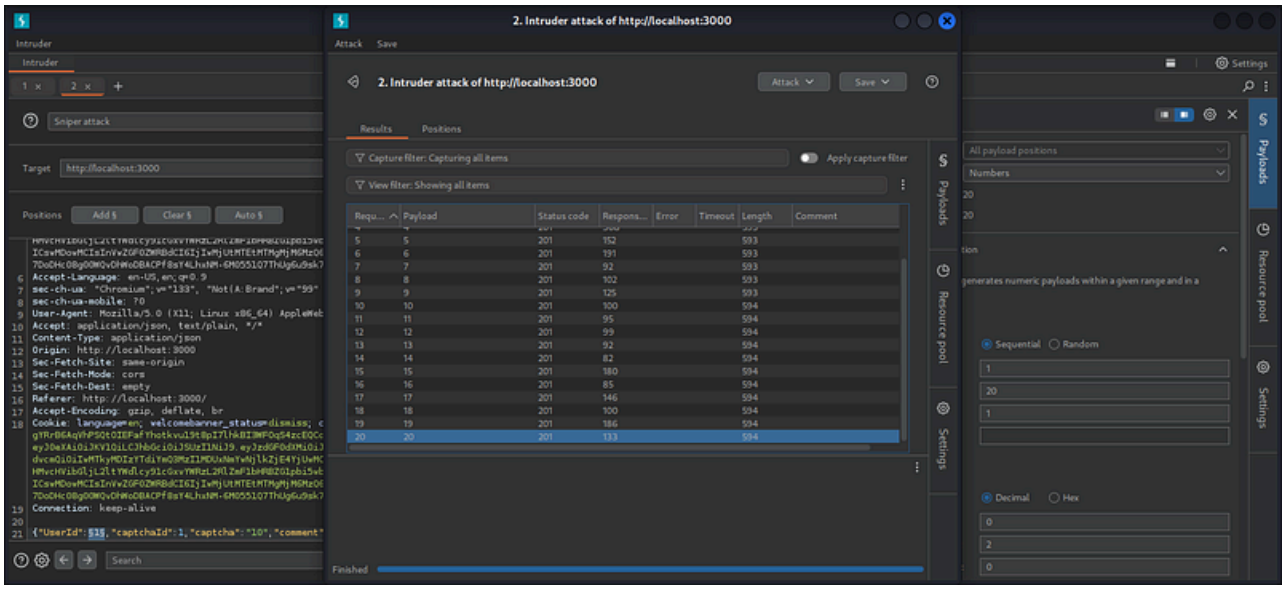


I sent the request to Burp Suite Intruder for automated testing. Burp Suite Intruder allows rapid repeated HTTP requests.

Technically, no payload modification is needed but I still selected the UserId field as a payload position—this step is optional and unnecessary but still works.



In the Payloads tab, I set the Payload Type to **Numbers**. I configured it to iterate from **1 to 20** to ensure enough submissions.



I clicked **Start Attack**, and Intruder quickly sent all the requests.

With 20 feedbacks submitted in rapid succession, the CAPTCHA Bypass challenge was successfully completed.

By Alexander Sapo on November 13, 2025.

Canonical link

Exported from Medium on February 7, 2026.