**Chapter 6**



**Bypassing Client-Side Controls:**

Authentication appears simple but is a critical and often weak point in web application security. If compromised, attackers can gain full access to an application and its data, rendering other security measures ineffective. Despite its importance, secure authentication is difficult to implement and frequently flawed, ranging from basic weaknesses like password guessing to subtle issues in complex login processes.

**Design Flaws in Authentication Mechanisms:**

Some major design flaws in authentication mechanisms are:

* Bad Passwords
* Brute-Forcible Login
* Verbose Failure Messages
* Vulnerable Transmission of Credentials
* Password Change Functionality
* Forgotten Password Functionality
* “Remember Me” Functionality
* User Impersonation Functionality
* Incomplete Validation of Credentials
* Non-Unique Usernames
* Predictable Usernames
* Predictable Initial Passwords
* Insecure Distribution of Credentials

**Bad Passwords:**

Many web applications allow weak passwords, such as short, common, or default passwords. Since users often choose easy passwords, attackers can easily guess them and gain unauthorized access.

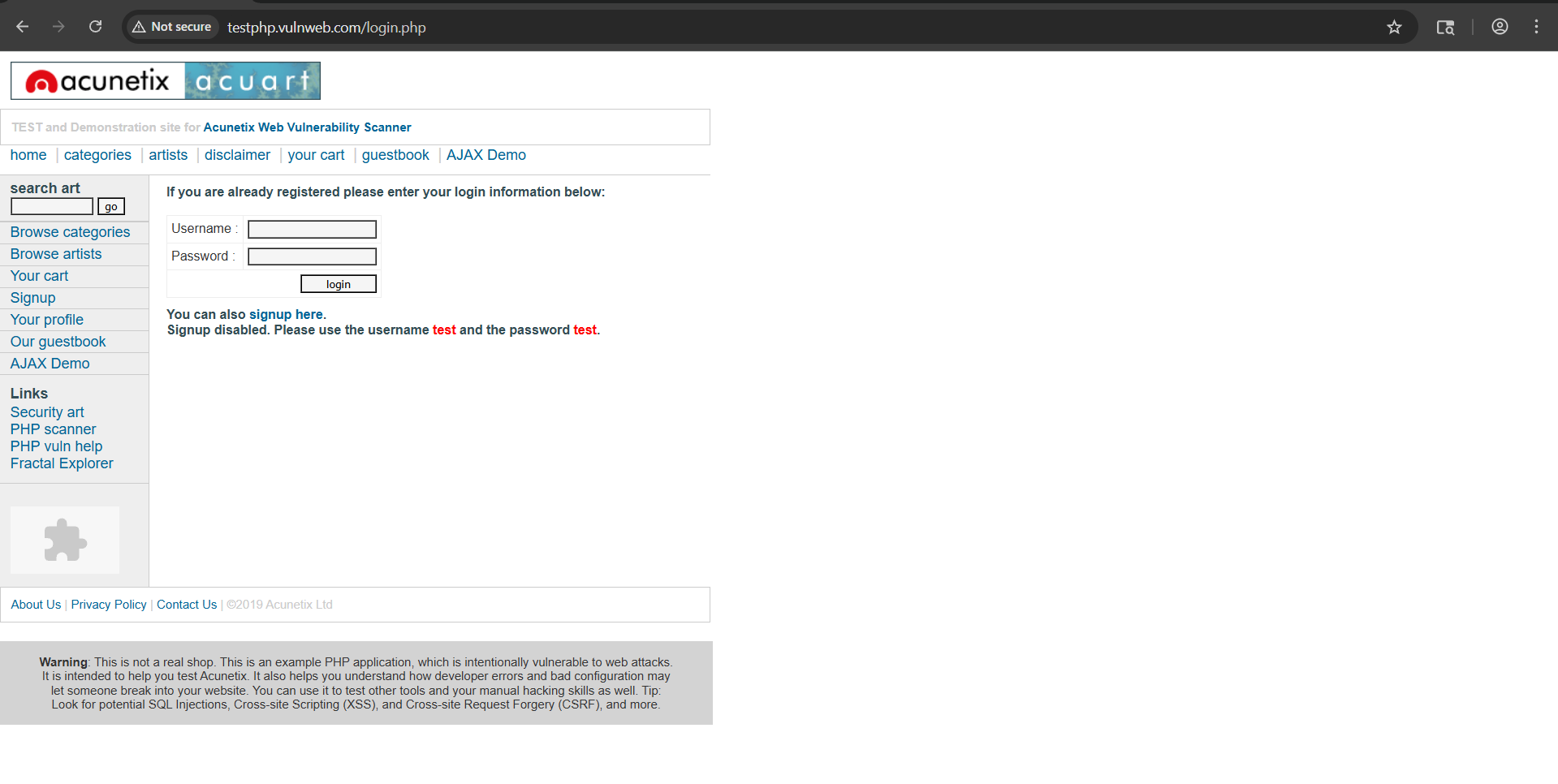
**Brute-Forcible login:**

If a login system allows unlimited attempts, attackers can repeatedly try common usernames and passwords until they succeed. Using automated tools, thousands of guesses can be made per minute, making even strong passwords vulnerable. Simple client-side protections can be easily bypassed and do not effectively stop these attacks.

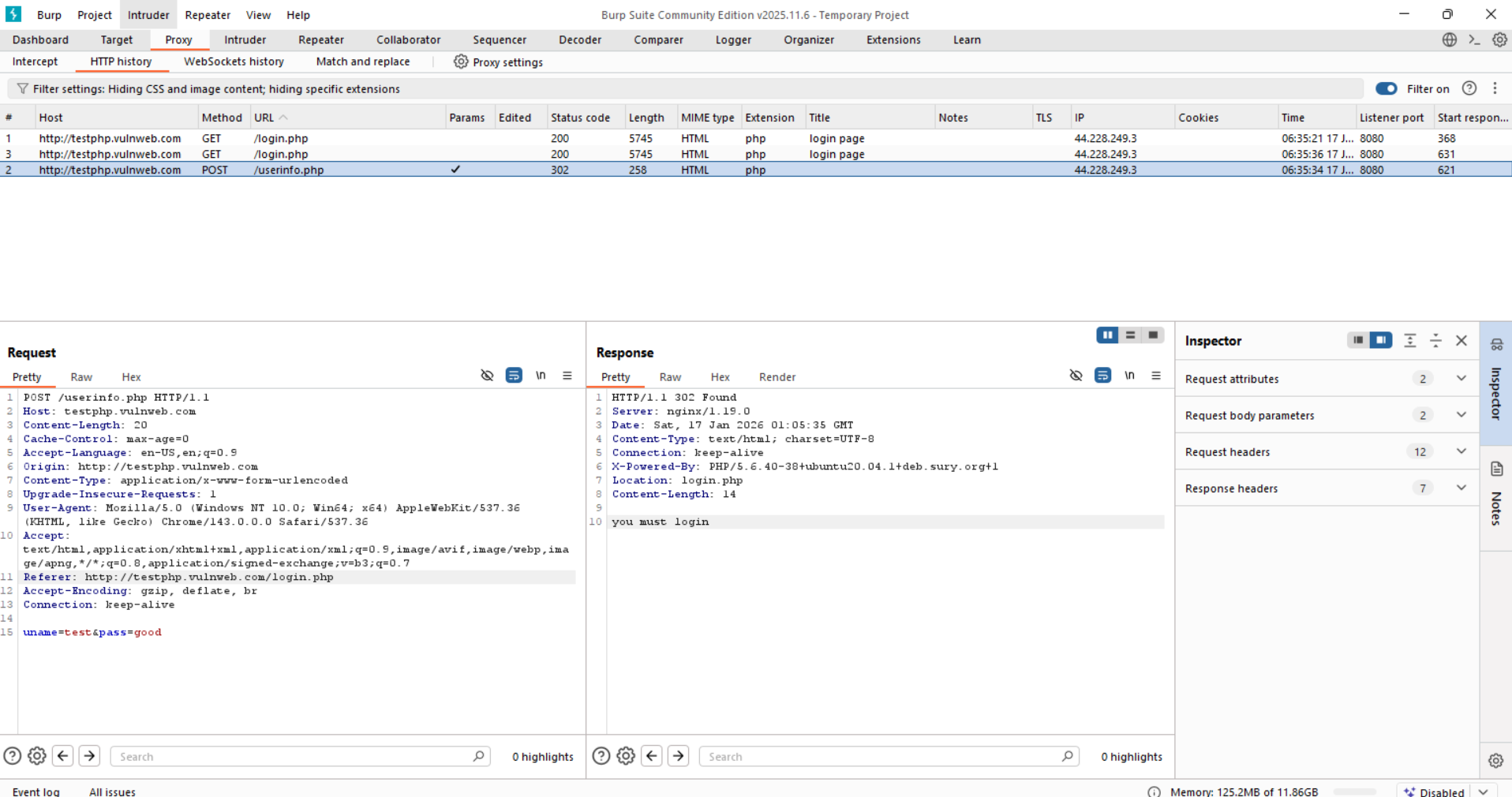
**Observing Brute-Forcible login attempts:**

Step1: Monitor several bad login attempts. Open the Burp Suite, and its inbuilt browser, visit the login page and try to submit any random set of id and password manually.

Browser:



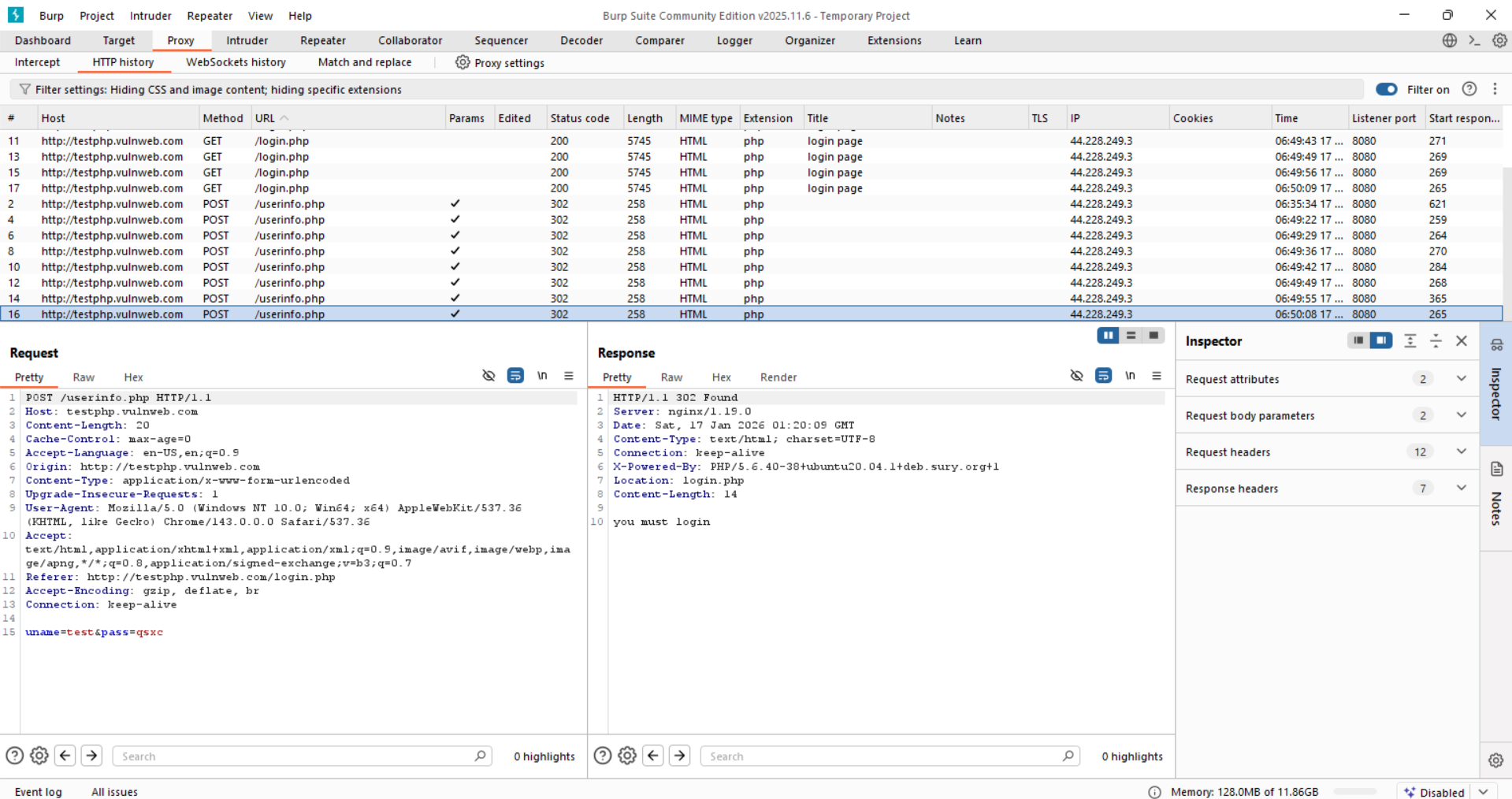
Burp:



Multiple failed login attempts were manually submitted using Burp Suite. For each invalid attempt, the server responded with an HTTP 302 redirect to login.php and a generic message stating “you must login.” The error message did not disclose whether the username or password was incorrect, indicating that the application does not leak authentication details through verbose error messages.

Step2: Approximately ten consecutive failed login attempts were submitted using invalid credentials.

Burp:



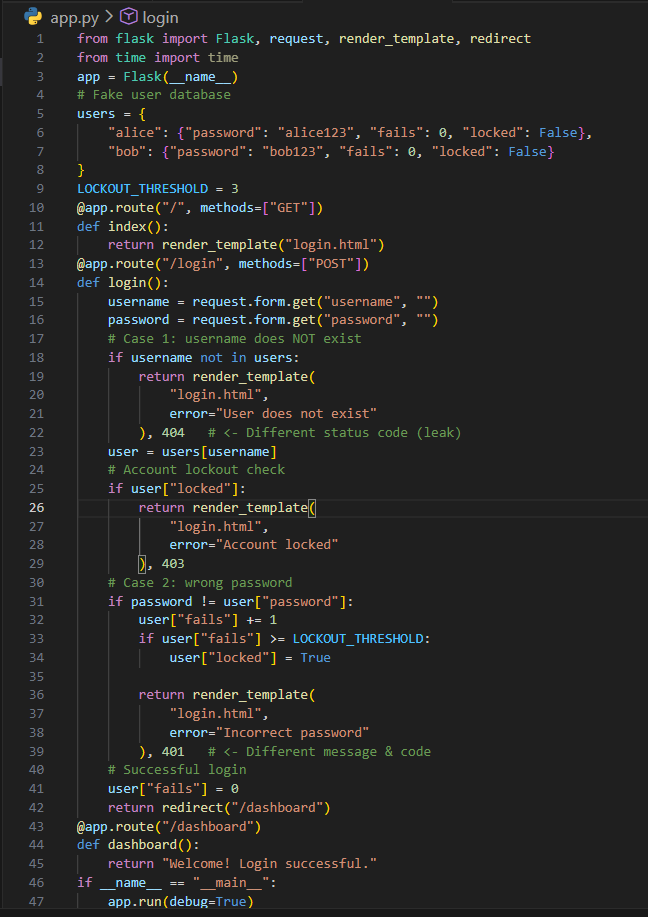
The application did not return any warning or account lockout message. A subsequent login attempt using valid credentials succeeded, indicating that no account lockout policy is implemented.

**Verbose Failure Messages:**

Verbose login failure messages weaken security by revealing too much information when a login attempt fails. If a system tells users whether a username or password is incorrect, attackers can use this to identify valid usernames through automated testing. Even subtle differences in error messages, hidden HTML, or response times can leak this information. Once attackers know valid usernames, it becomes much easier to carry out further attacks such as password guessing or social engineering.

**Observing Verbose Failure messages on a local website:**

App.py:

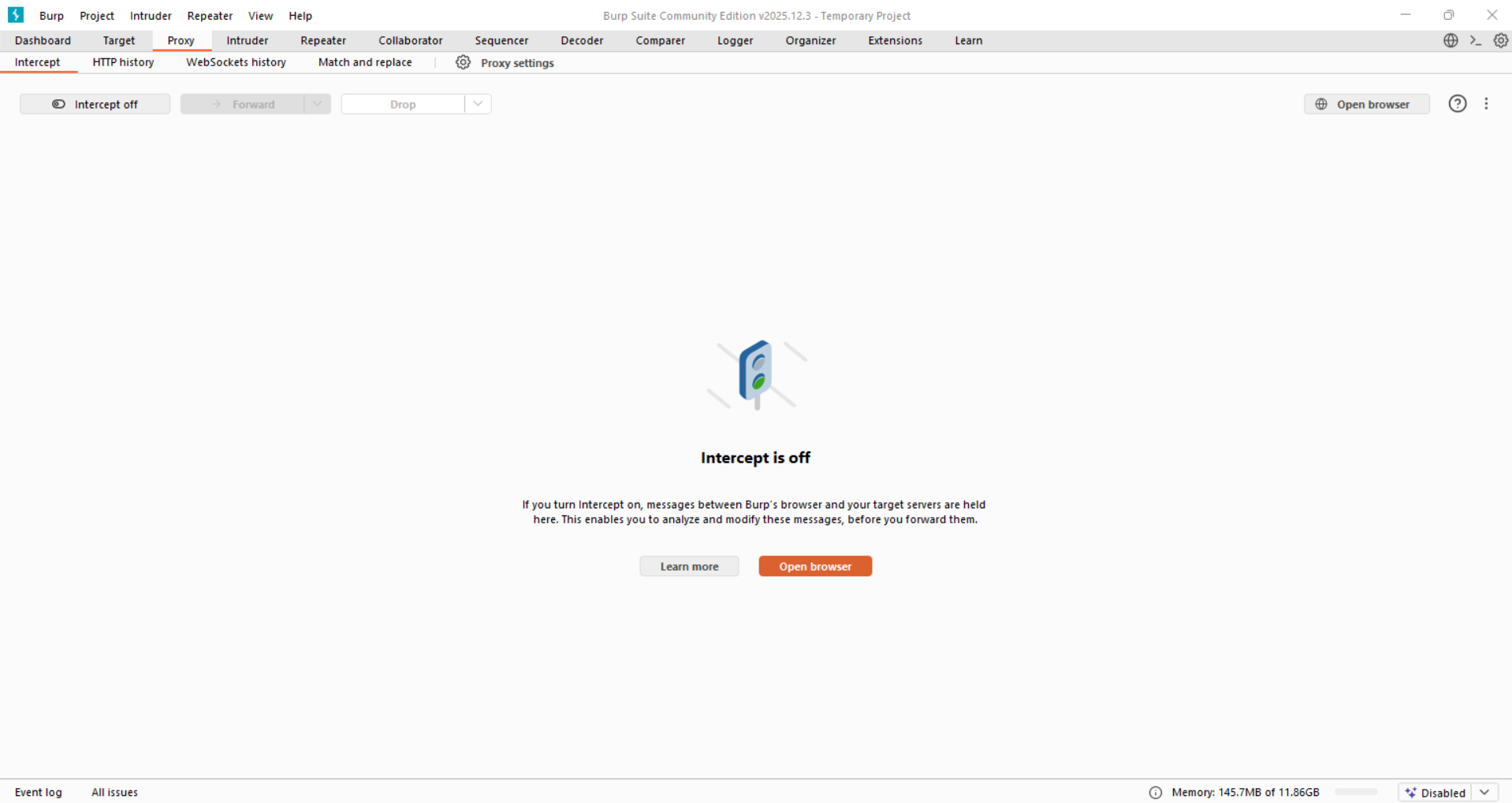


Templates/login.html:

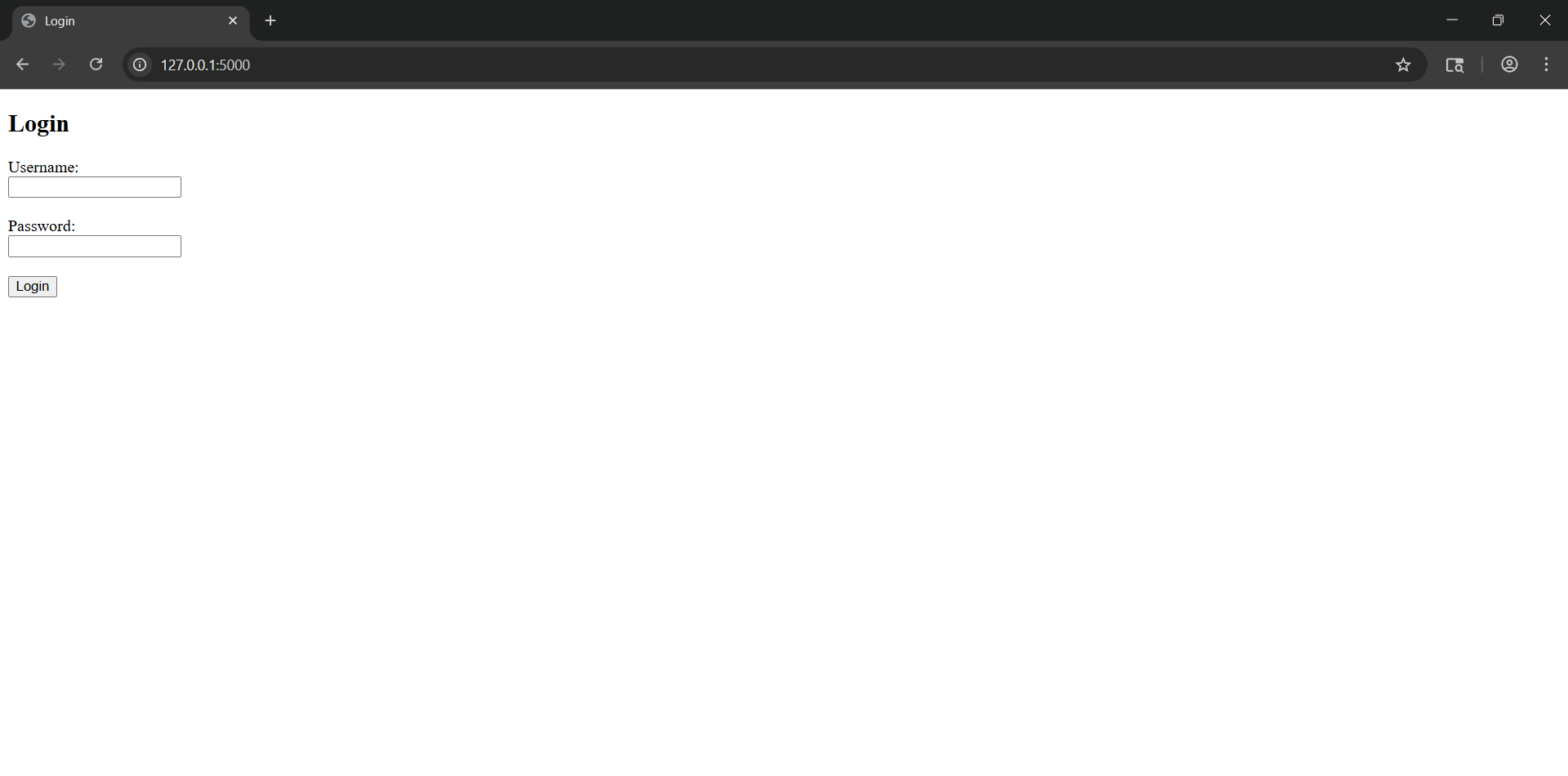


Step1: Open burp, go to <http://127.0.0.1:5000>. And in burp Turn Proxy → Intercept → ON (in step2)

Burp:

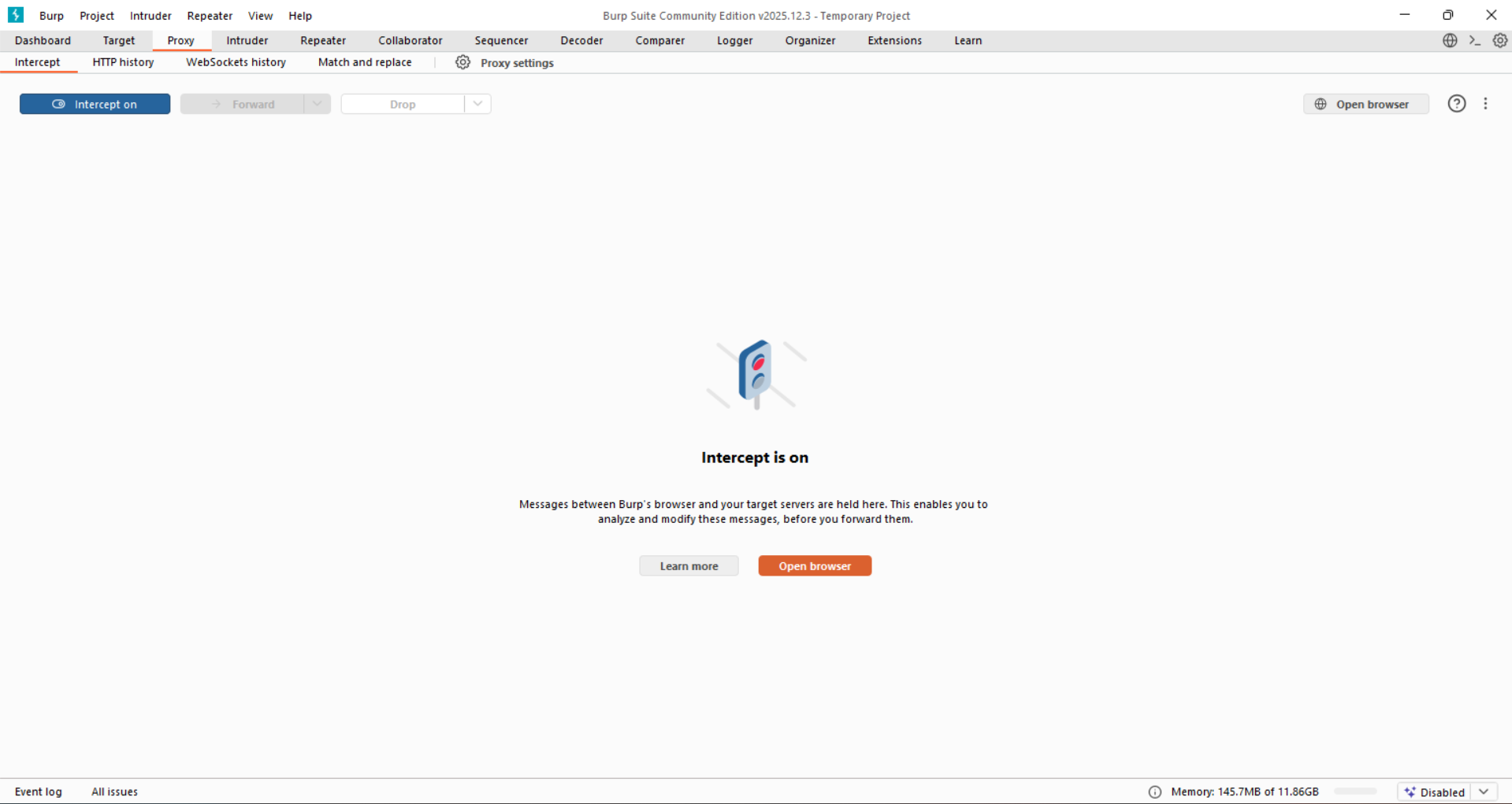


Browser:



Step2: Turn the intercept on. Fill details and click on “login” button.

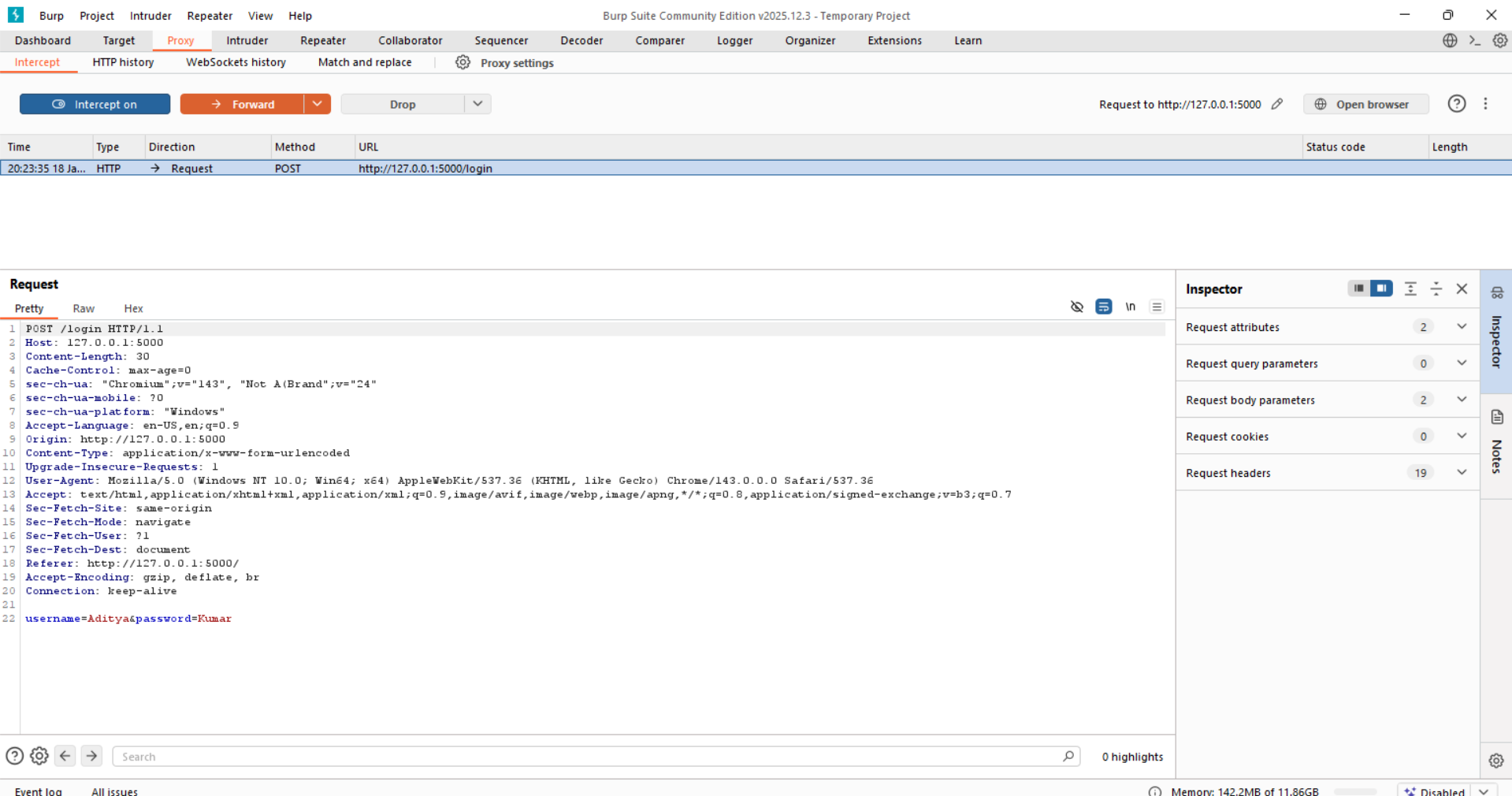
Burp:



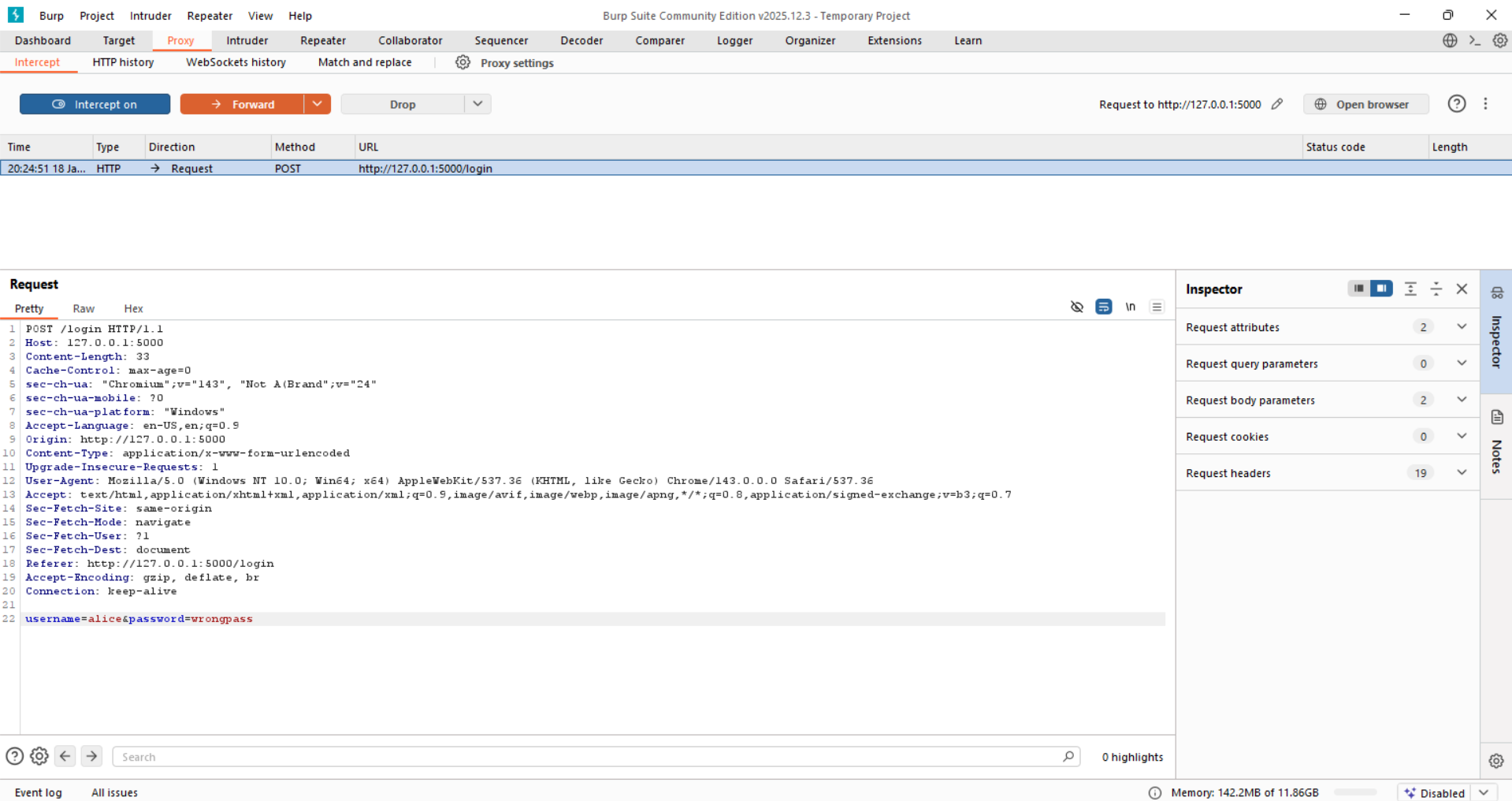
Browser: fill the details

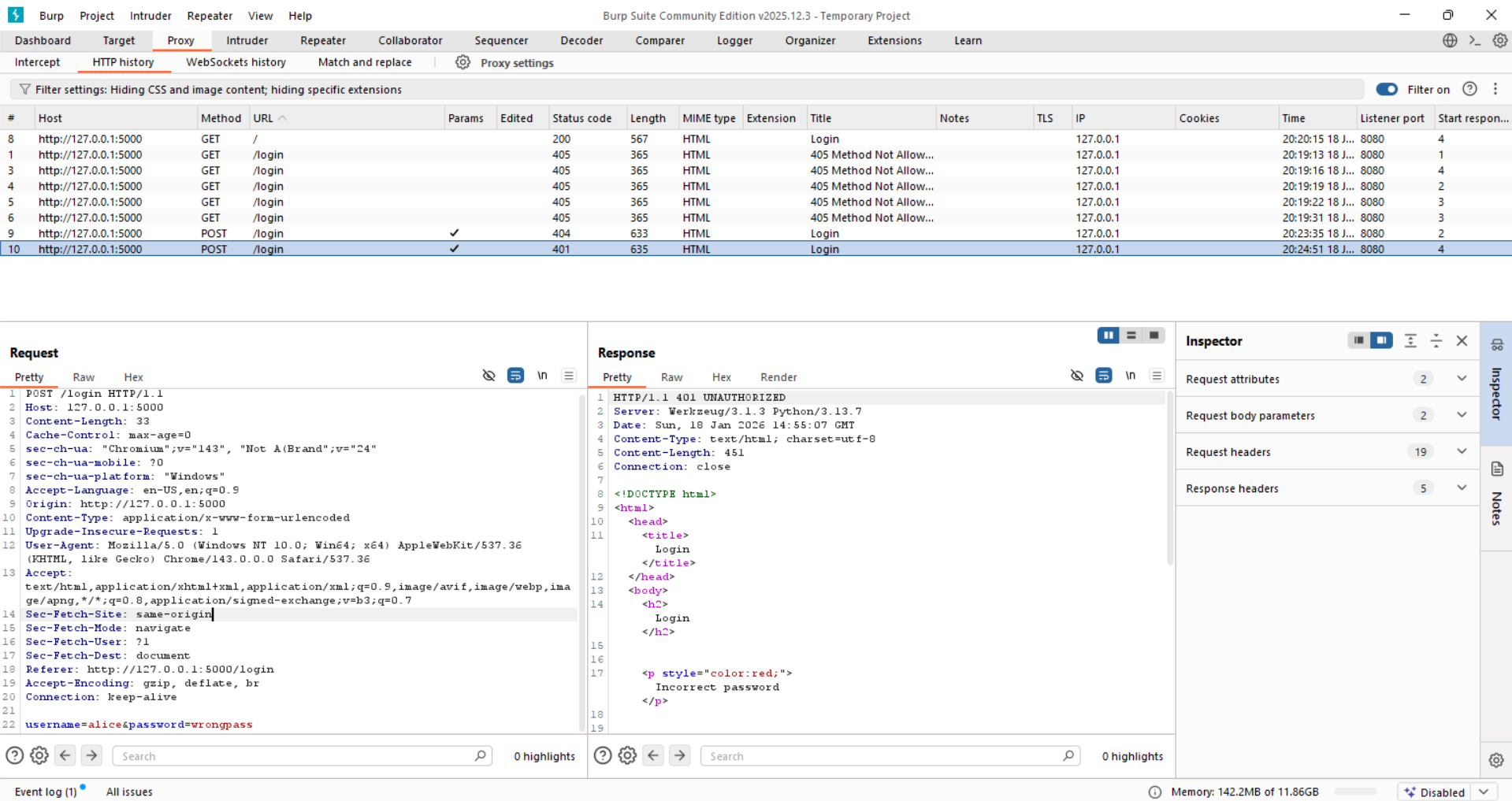


Step3: See the request in the burp.



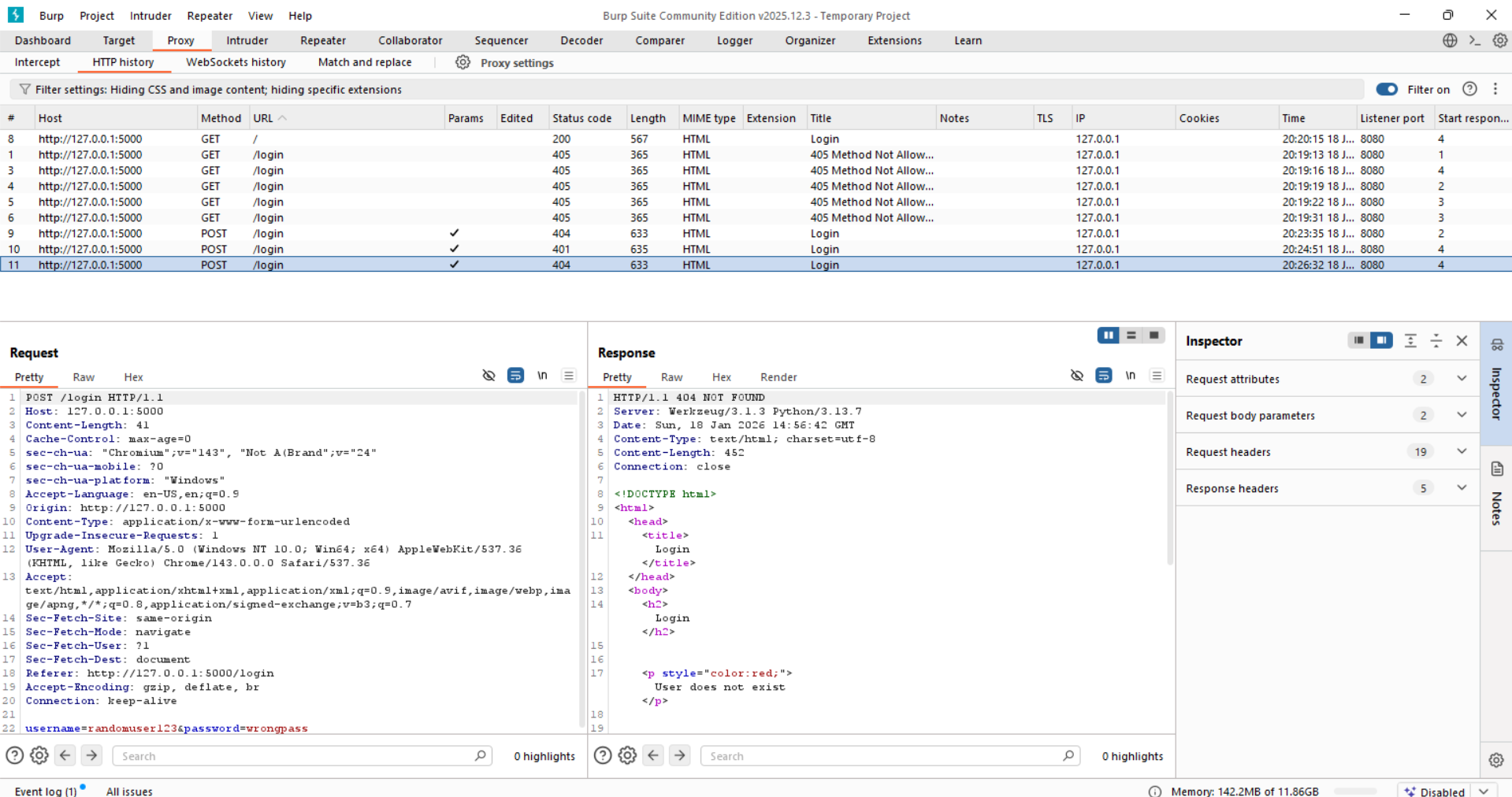
Now, test for valid username:





* Status: 401
* Message: Incorrect password

Now, test for invalid username:



* Status: 404
* Message: User does not exist

Enumeration vulnerability confirmed.