Unlocking Secrets - The Art of Cracking Passwords Write-up

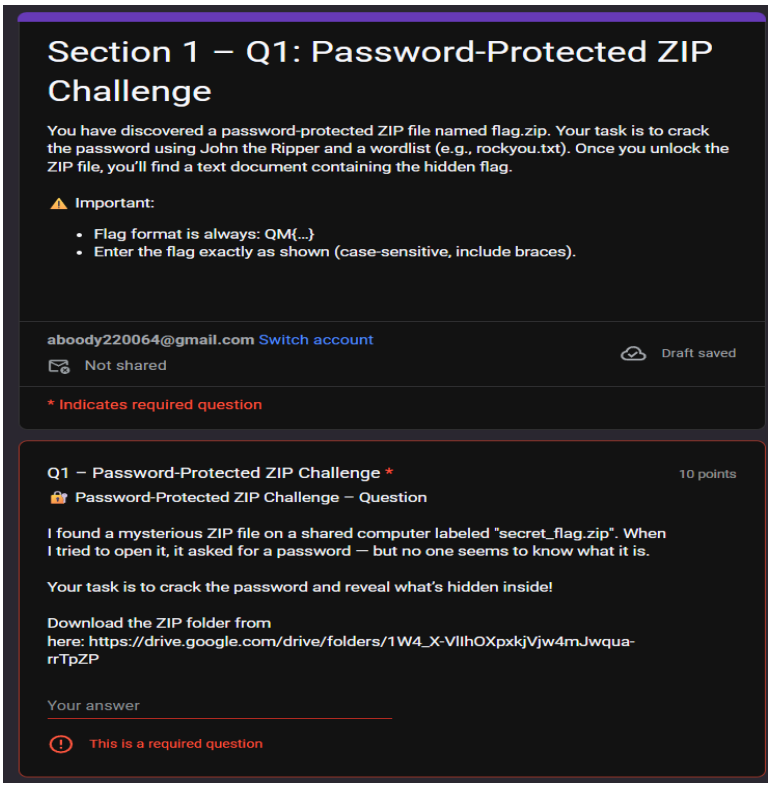
Introduction

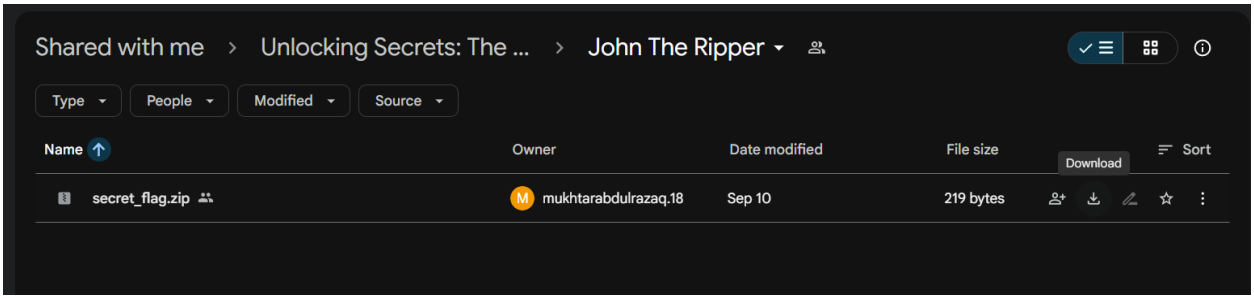
In this hands-on workshop, participants explored real-world techniques used in ethical hacking to recover hidden information—from cracking password-protected archives to uncovering secrets through OSINT (Open-Source Intelligence) and simulating phishing attacks.
All challenges emphasize ethical use, critical thinking, and defensive awareness.

Challenge 1: Password-Protected ZIP File

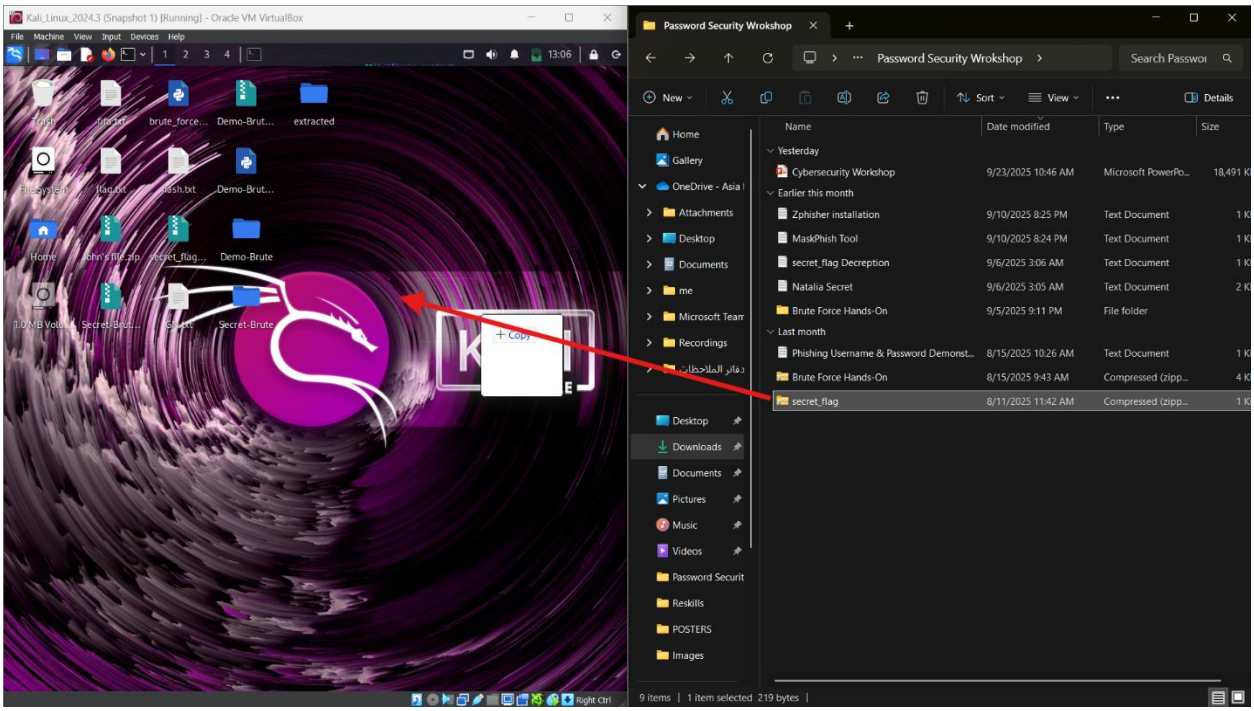
Explanation for Q1:

The first challenge, "Password-Protected ZIP Challenge," requires you to crack a password-protected ZIP file named "secret_flag.zip." Your goal is to find the hidden flag inside a text document within the ZIP file. The image provided shows the problem description and a link to download the ZIP file from Google Drive.

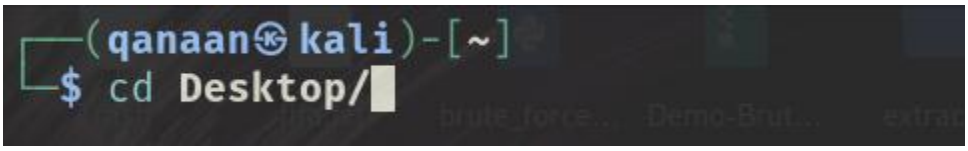




First step: We download "secret_flag.zip" from the Google Drive folder



Now that you've downloaded the challenge file, **secret_flag.zip**, from the provided link, the next step is to get it into your Kali Linux environment. As seen in the image, you can simply **drag and drop** the compressed ZIP file from your host machine's file explorer on the right directly onto the Kali Linux desktop on the left. This makes the file accessible within the virtual machine so you can use the necessary tools to begin the password cracking process.



To begin the challenge, we need to open the terminal in Kali Linux. We'll start by navigating to the **Desktop** directory where we placed our secret_flag.zip file. We do this using the cd command, which stands for "change directory."

```
┌──(qanaan㉿kali)-[~/Desktop]
└─$ ls
Demo-Brute              "John's file.zip"     extracted     secret_flag.zip
Demo-Brute-Script.py    Secret-Brute          fifa.txt
Demo-Brute.zip          Secret-Brute.zip      flag.txt
GN.txt                  brute_force_zip.py    hash.txt
```

Now we have the file in our desktop

```
┌──(qanaan㉿kali)-[~/Desktop]
└─$ zip2john secret_flag.zip > hero.txt
ver 1.0 efh 5455 efh 7875 secret_flag.zip/flag.txt PKZIP Encr: 2b chk, TS_chk
, cmplen=37, decmplen=25, crc=8B4E6516 ts=5BD2 cs=5bd2 type=0
```

The screenshot shows the command zip2john secret_flag.zip > hero.txt, which runs zip2john to extract a crackable hash from secret_flag.zip and redirects that hash output into the file hero.txt for use by password-cracking tools.

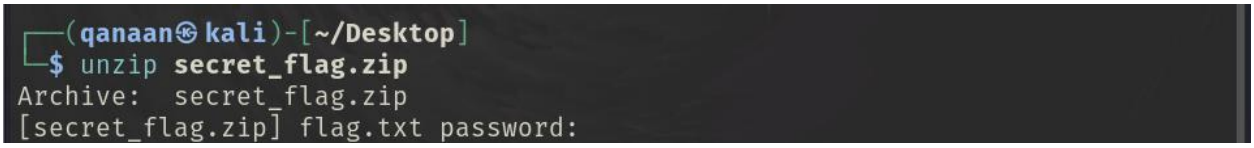
```
┌──(qanaan㉿kali)-[~/Desktop]
└─$ cat hero.txt
secret_flag.zip/flag.txt:$pkzip$1*2*2*0*25*19*8b4e6516*0*42*0*25*5bd2*9b43aff
0cdbb0e2da9f3f97696fef6642c02768aaa9007fdc673c014ebe3c362520fb6d1c6*$/pkzip$:
flag.txt:secret_flag.zip::secret_flag.zip
```

cat hero.txt is to display the contents of hero.txt — this prints the ZIP-derived hash (the output from zip2john) to the terminal so you can inspect and confirm the hash was created and saved correctly.

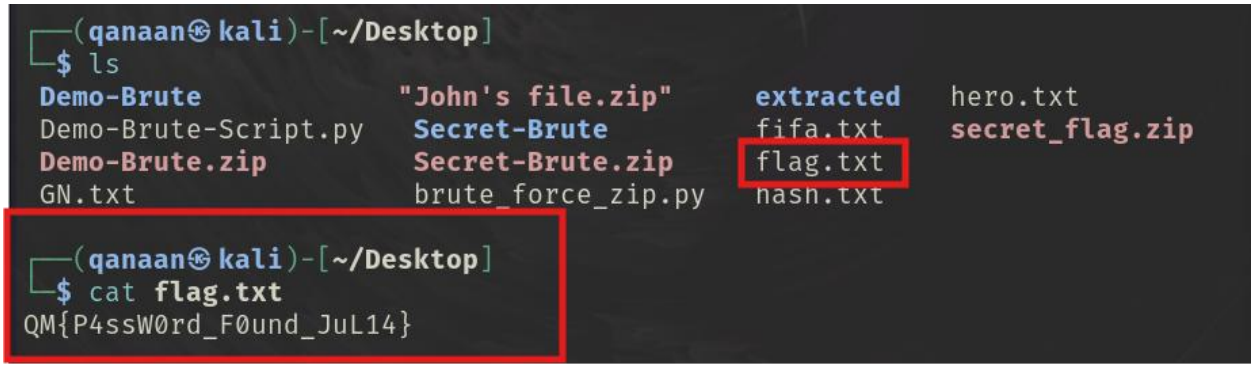
```
┌──(qanaan㉿kali)-[~/Desktop]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt hero.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
No password hashes left to crack (see FAQ)
```

I used the command john --wordlist=/usr/share/wordlists/rockyou.txt hero.txt to attempt cracking the ZIP password; John successfully loaded the hash from hero.txt

I used the command john --show hero.txt to display the cracked password, and the output revealed that the password for secret_flag.zip is **julia225**, confirming that one password hash was successfully cracked.



Now we unzip secret_flag.zip file to and put the correct password we got from John



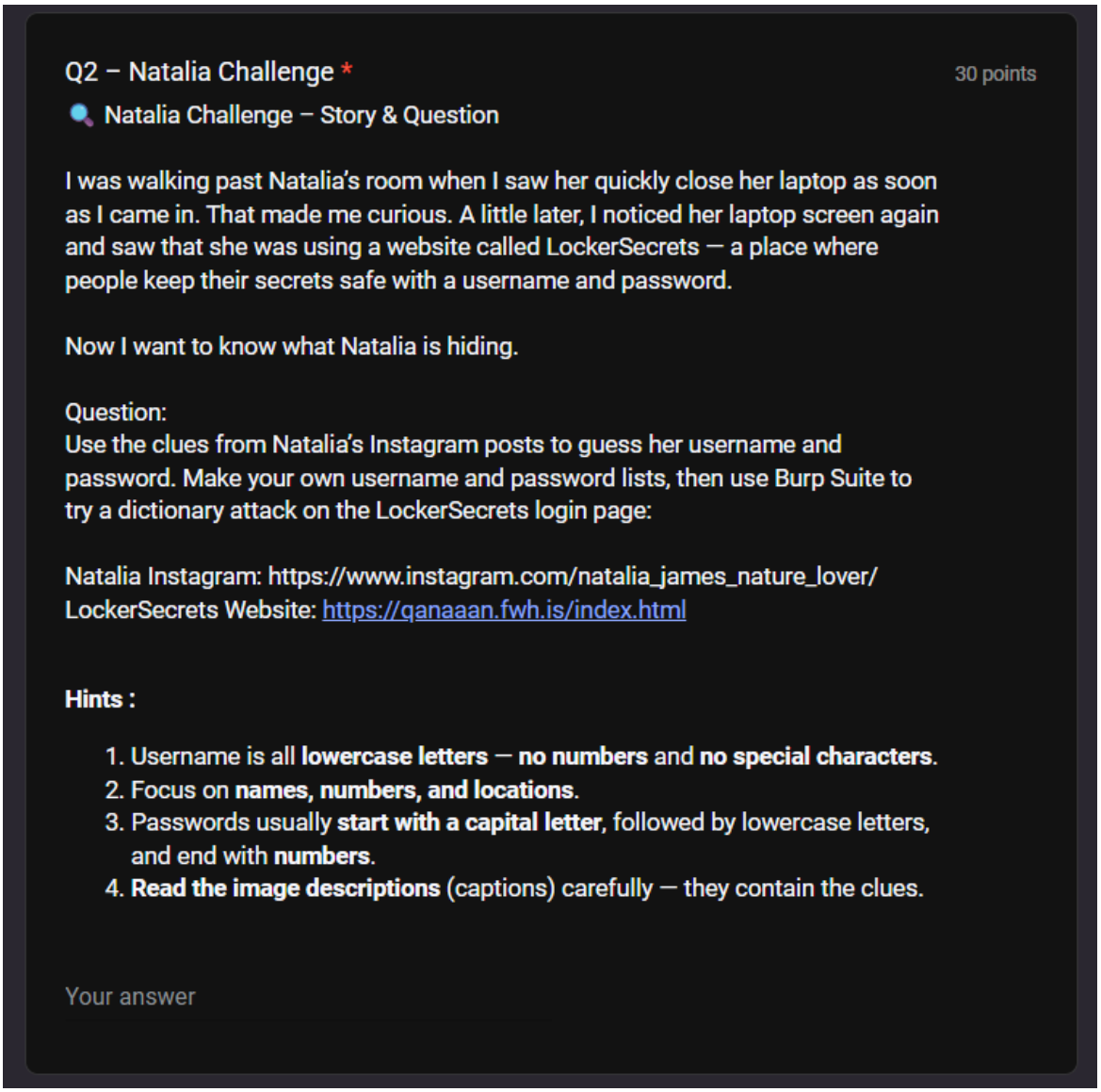
Now we extract the "secret_flag.zip" file on our Kali Linux desktop. After extraction, we find the "flag.txt" file. Next, we use the cat command to open "flag.txt" and reveal the flag "QM{P4ssW0rd_Found_Jul14}", which we submit with exact case and braces.

Challenge 2: Natalia's OSINT Secret

Explanation for Q2:

Dear students and lecturers,

Now we dive into the Natalia OSINT Challenge! Our goal is to uncover what Natalia is hiding on the LockerSecrets website using her Instagram account for clues.



Q2 – Natalia Challenge *

30 points

🔍 Natalia Challenge – Story & Question

I was walking past Natalia's room when I saw her quickly close her laptop as soon as I came in. That made me curious. A little later, I noticed her laptop screen again and saw that she was using a website called LockerSecrets — a place where people keep their secrets safe with a username and password.

Now I want to know what Natalia is hiding.

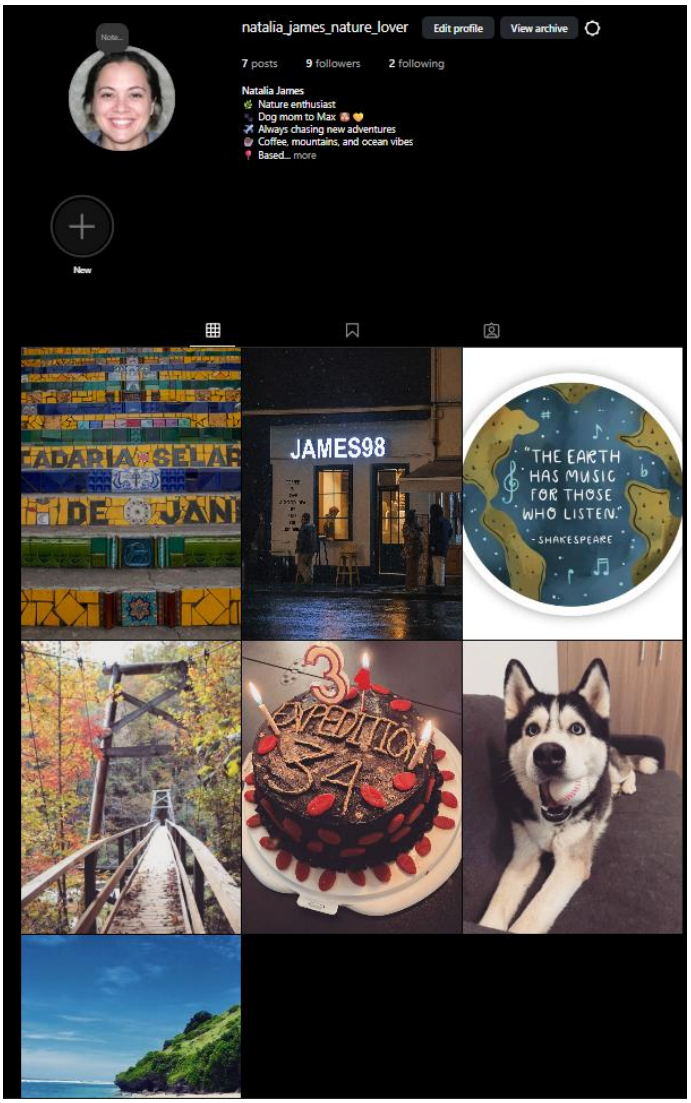
Question:
Use the clues from Natalia's Instagram posts to guess her username and password. Make your own username and password lists, then use Burp Suite to try a dictionary attack on the LockerSecrets login page:

Natalia Instagram: https://www.instagram.com/natalia_james_nature_lover/
LockerSecrets Website: https://qanaaan.fwh.is/index.html

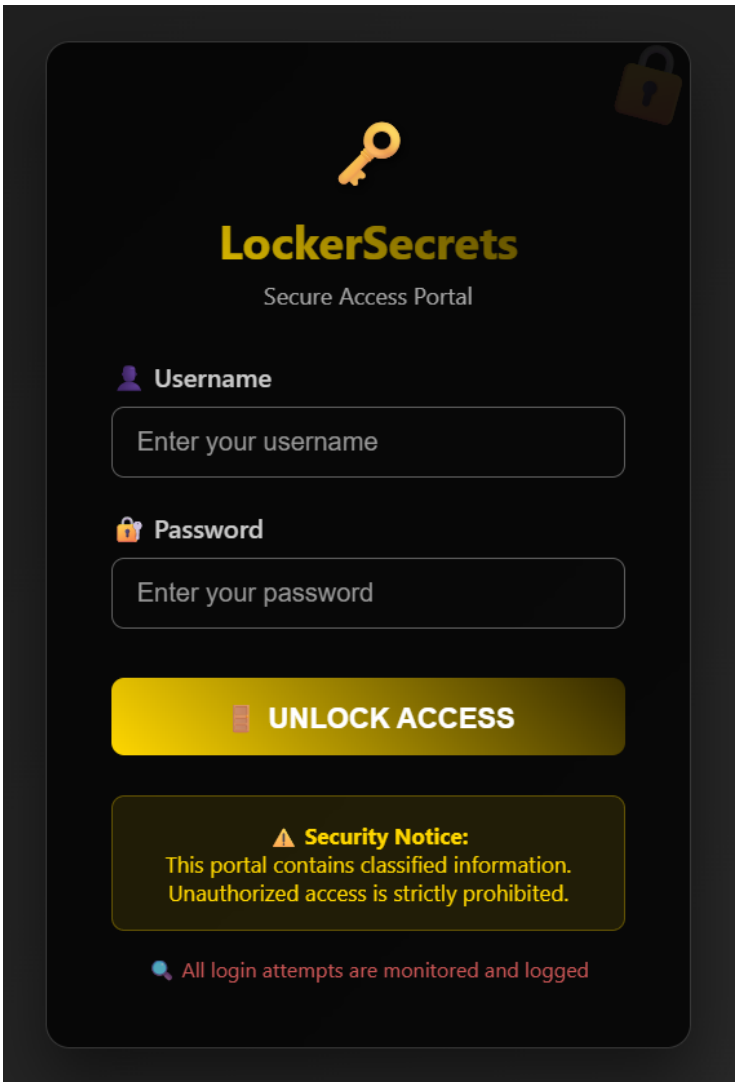
Hints :

1. Username is all **lowercase letters** — **no numbers** and **no special characters**.
2. Focus on **names, numbers, and locations**.
3. Passwords usually **start with a capital letter**, followed by lowercase letters, and end with **numbers**.
4. **Read the image descriptions** (captions) carefully — they contain the clues.

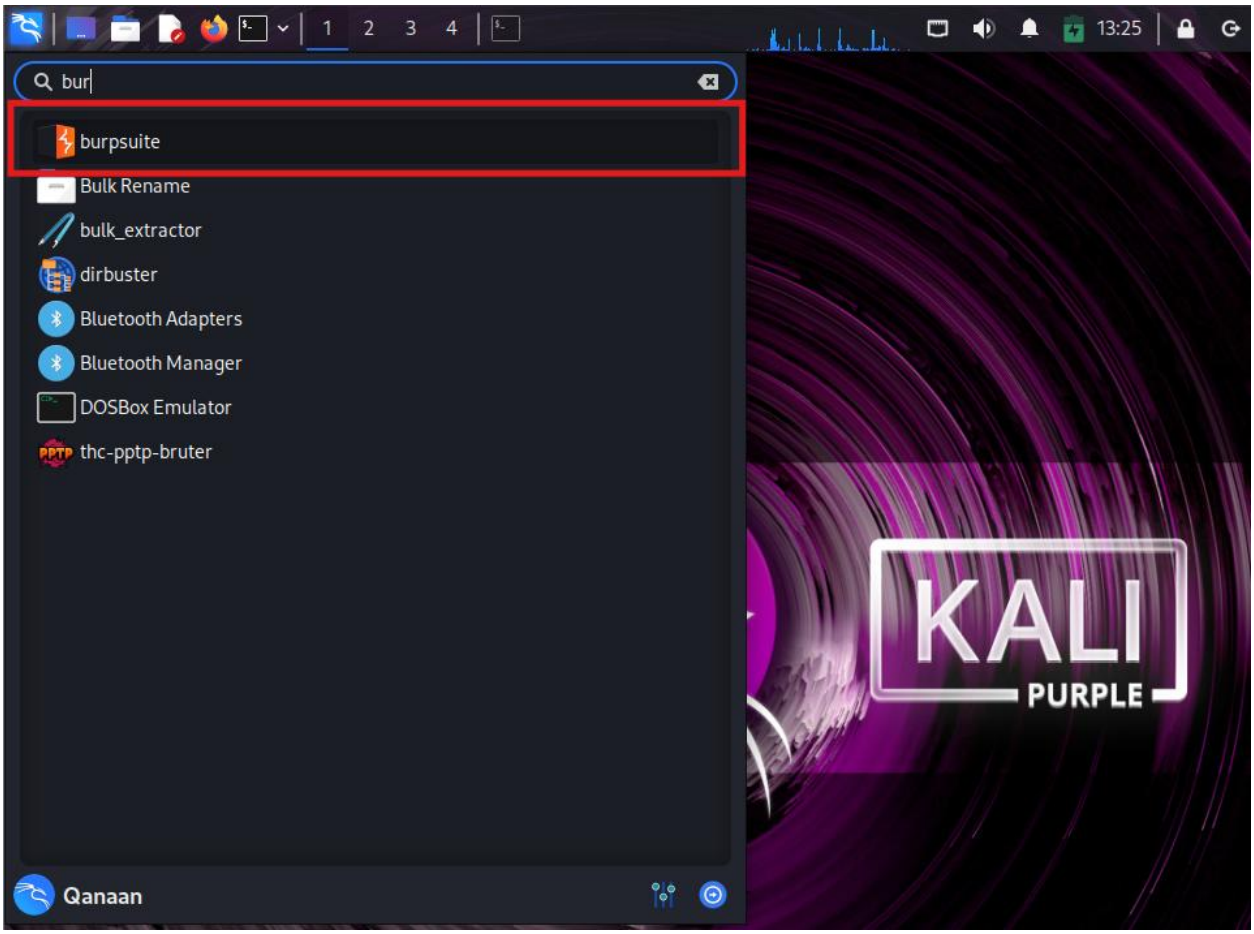
Your answer



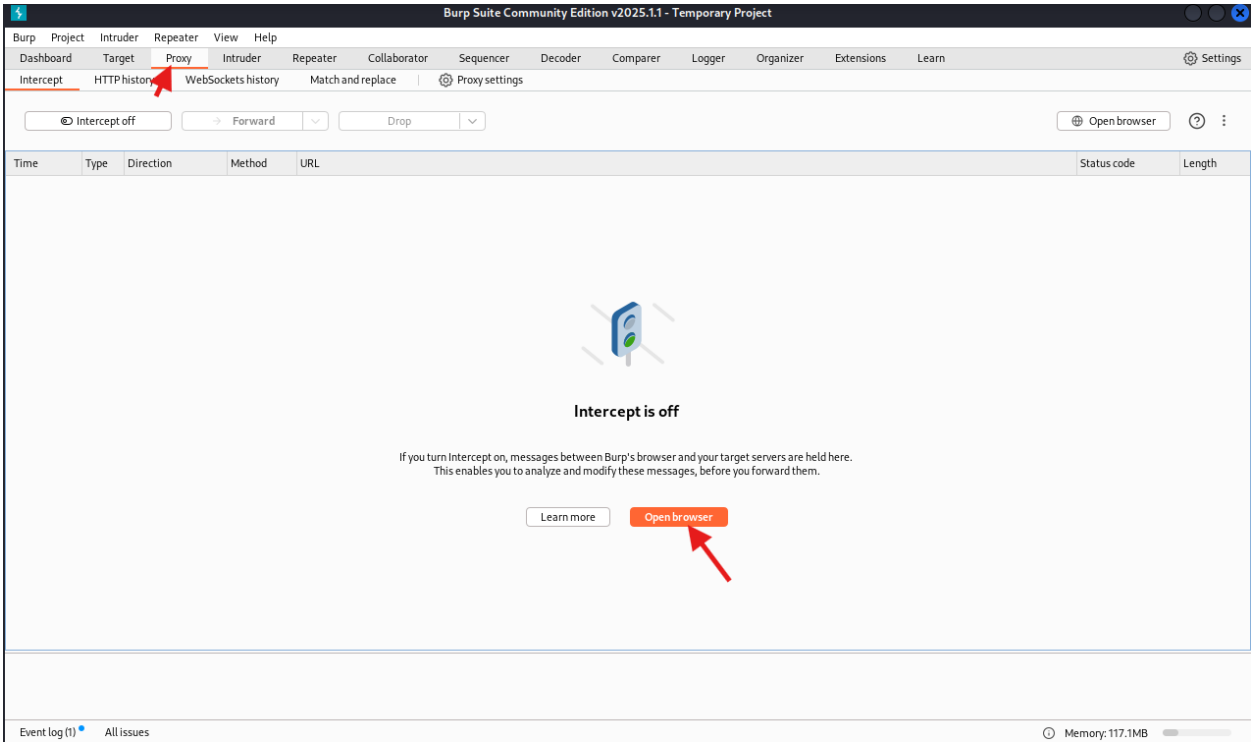
First, we visit Natalia's Instagram at https://www.instagram.com/natalia_james_nature_lover/ to gather hints about her username and password.



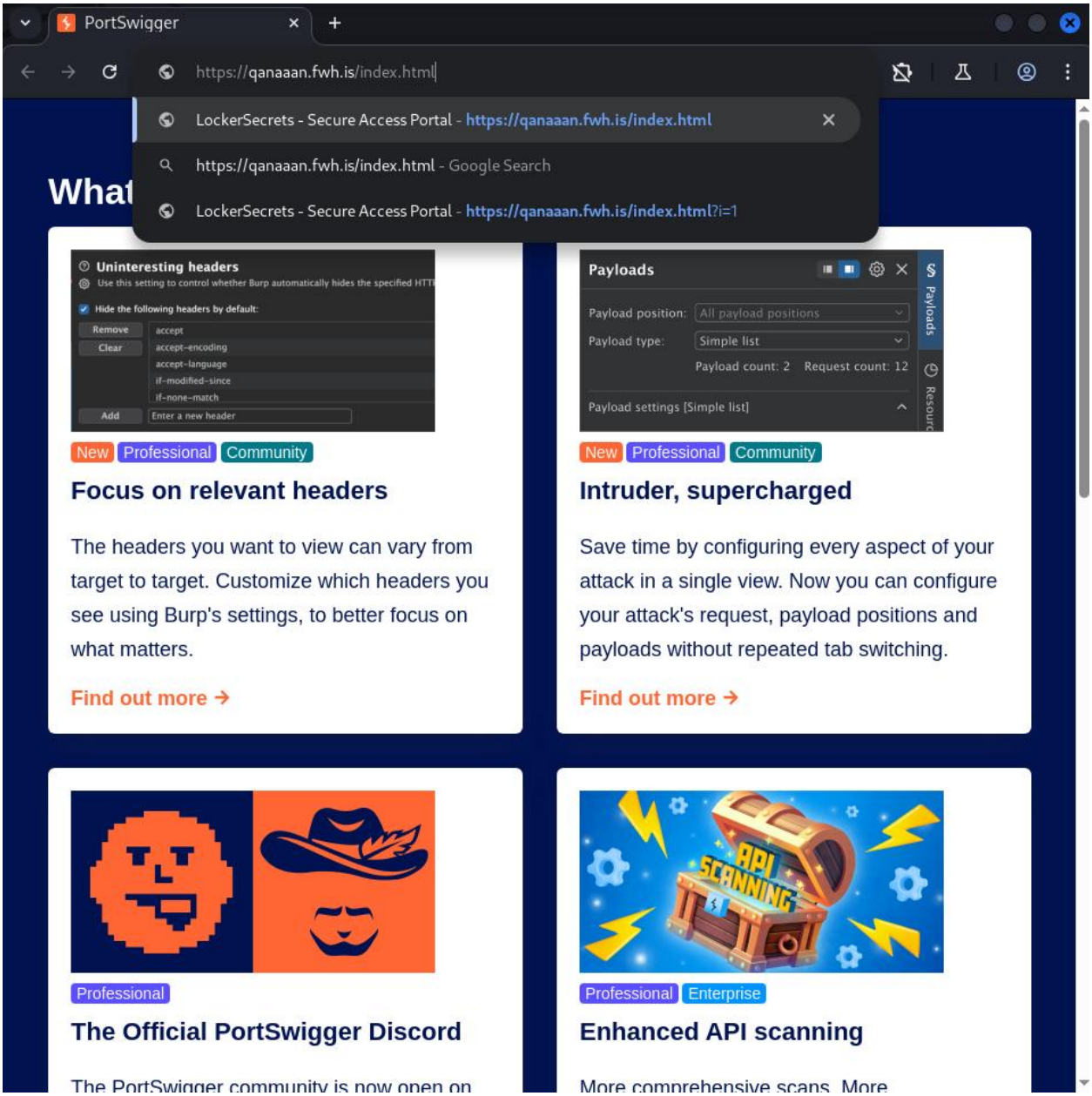
This is the login page for “LockerSecrets” — a fake secure portal where users must enter a username and password to gain access.



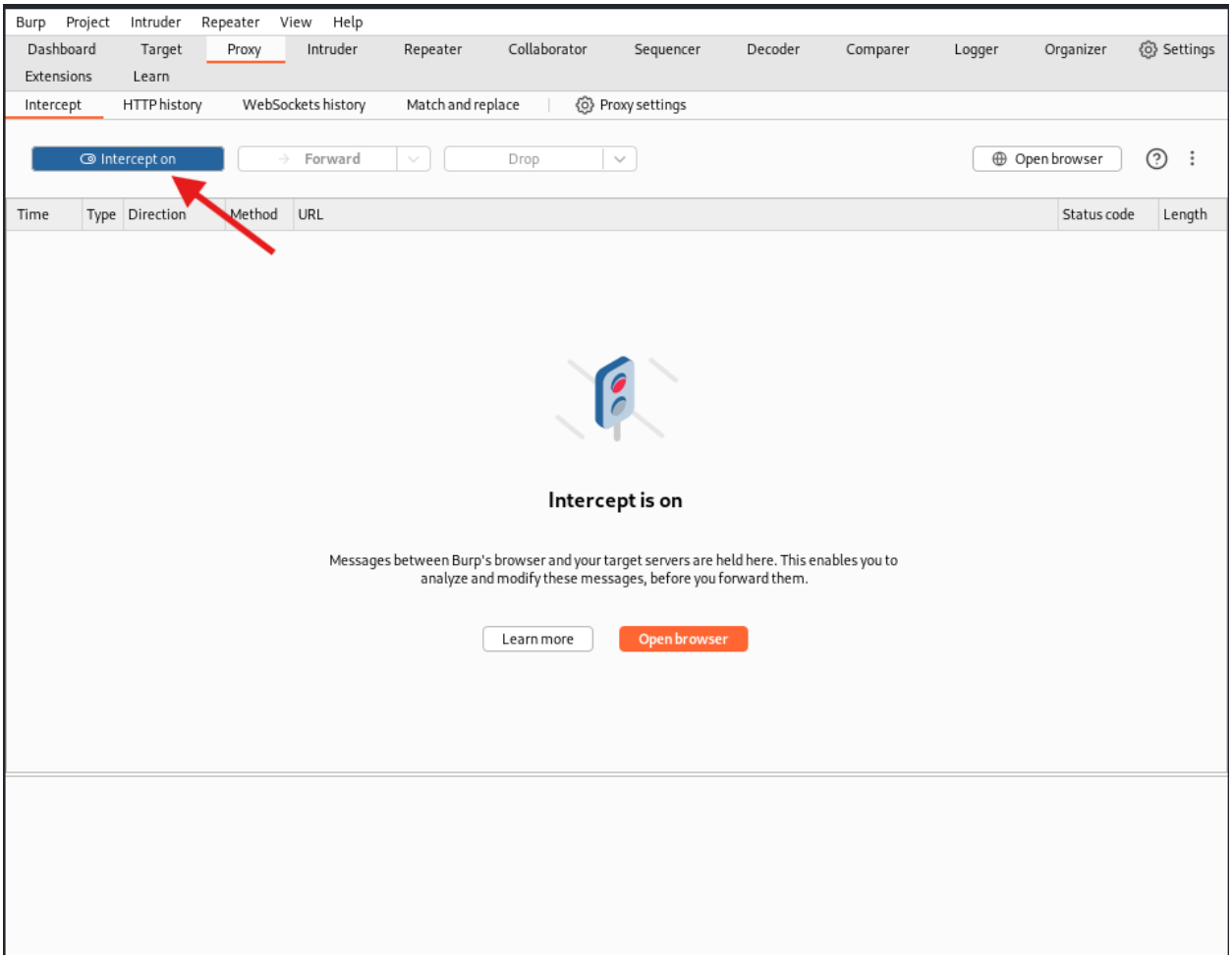
Now we need to enter Burp Suite — open it from the menu or terminal to start intercepting and attacking the LockerSecrets login page.



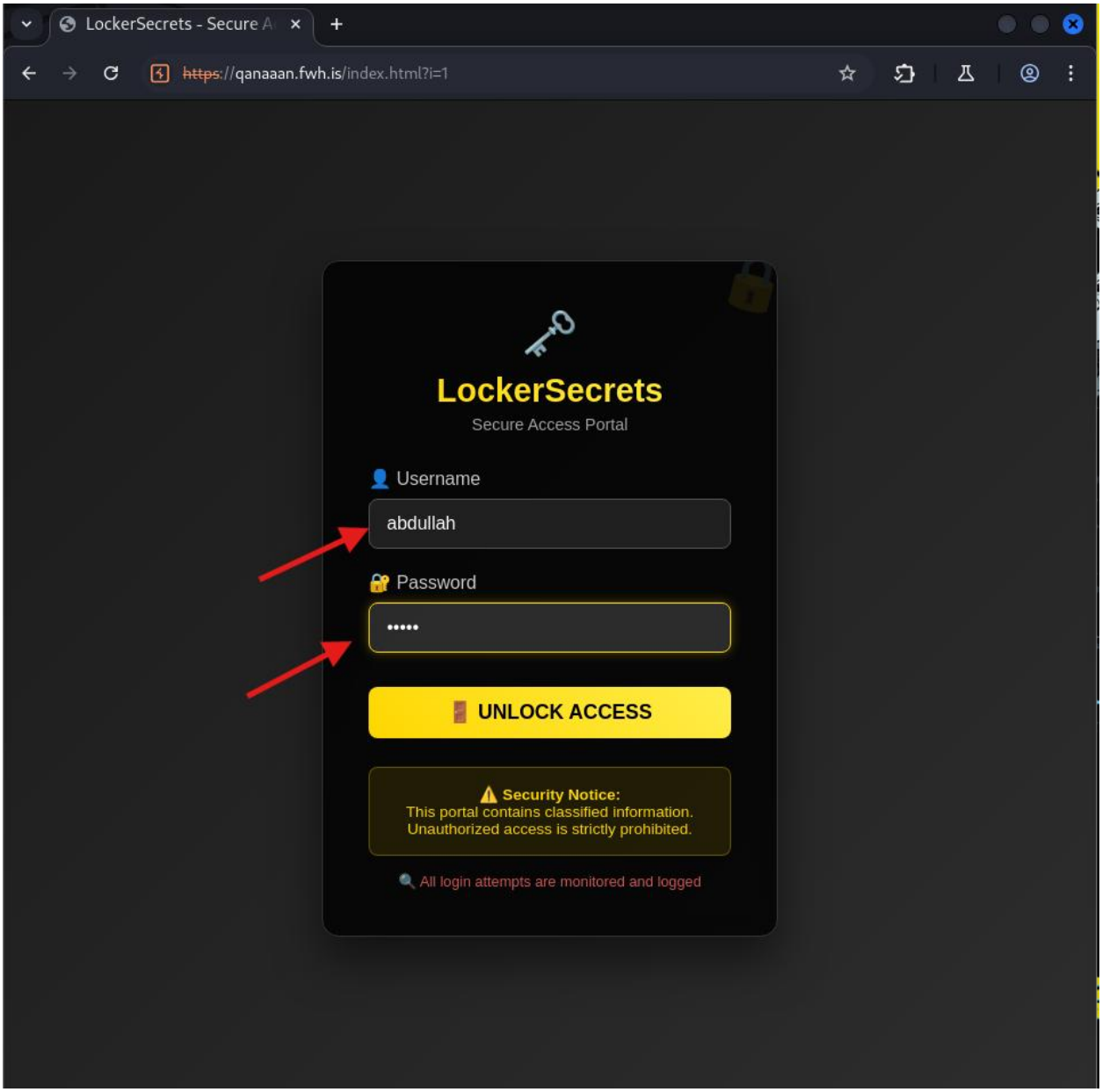
Now we're in Burp Suite — on the Proxy tab, with "Intercept is off."
Click "Open browser" (highlighted in red) to launch Burp's built-in browser — this lets you visit LockerSecrets while Burp captures your traffic for analysis.



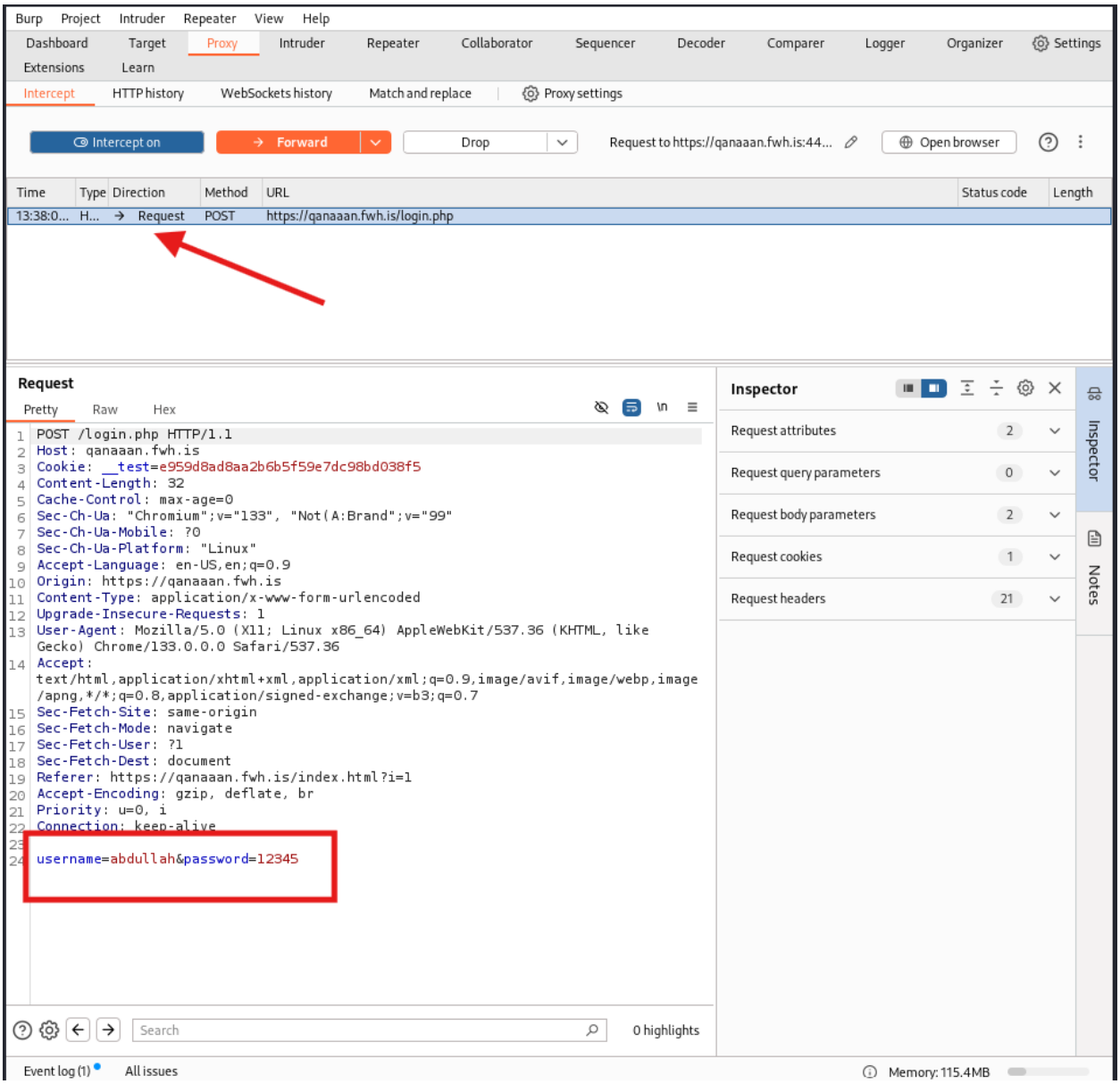
Now search for:
https://qanaaan.fwh.is/index.html
— paste it into the browser's address bar and hit Enter to load the LockerSecrets login page. Once loaded, you'll be ready to intercept the login request in Burp Suite.



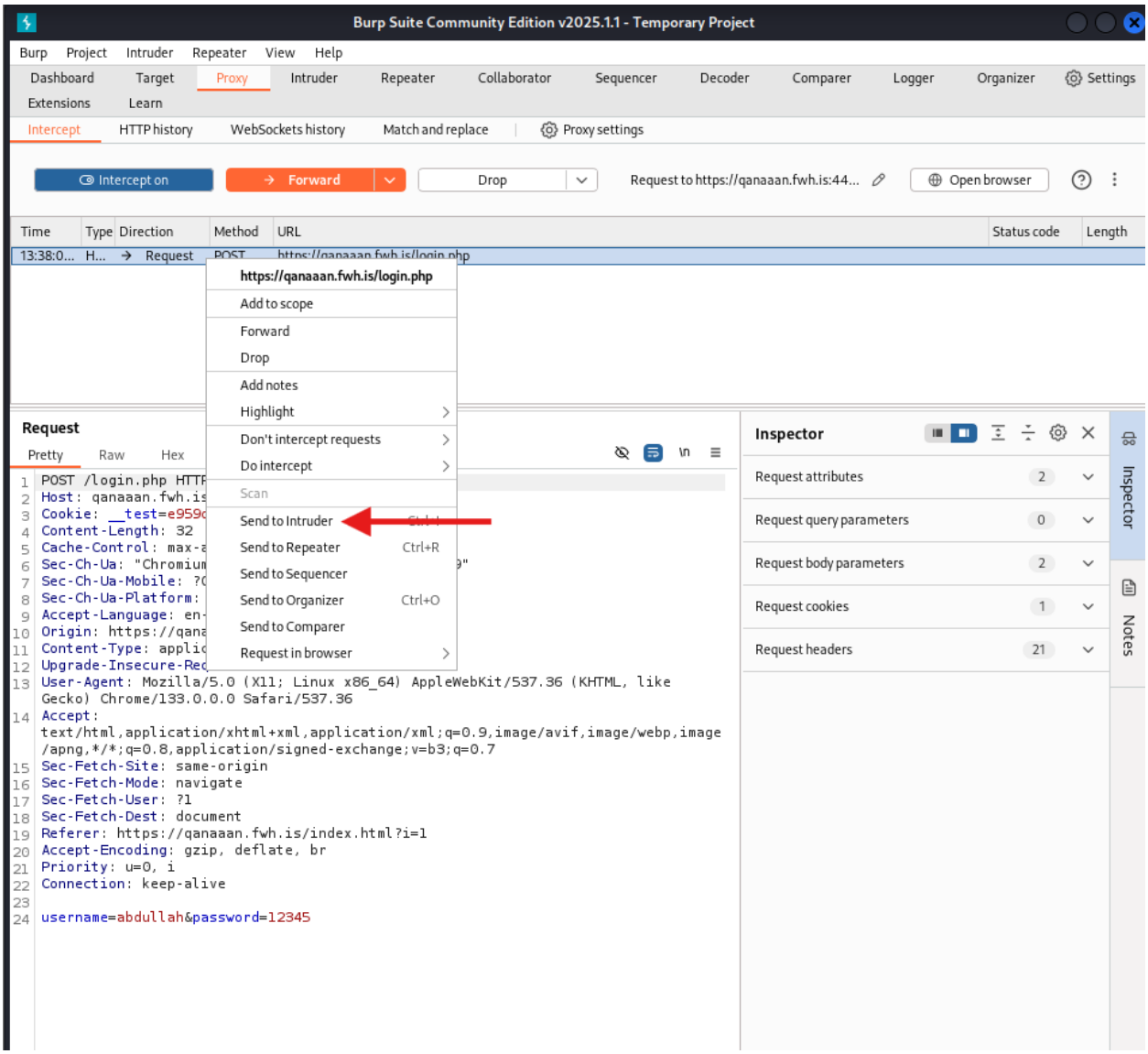
Now Intercept is ON — Burp Suite is ready to catch your login request.



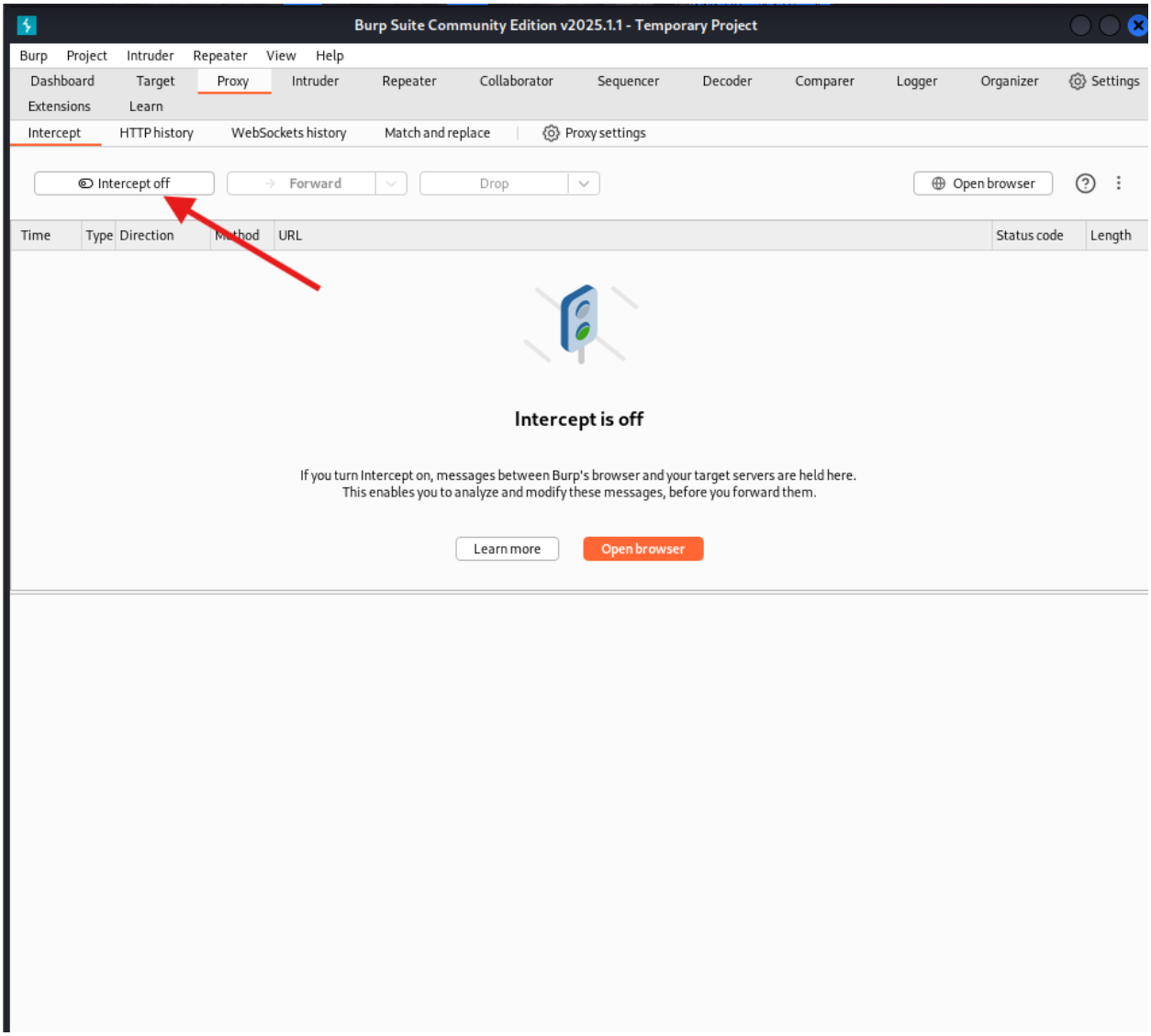
Now enter a test username (like “abdullah”) and password into the fields — this will trigger a login request that Burp Suite can intercept. And then click unlock access.



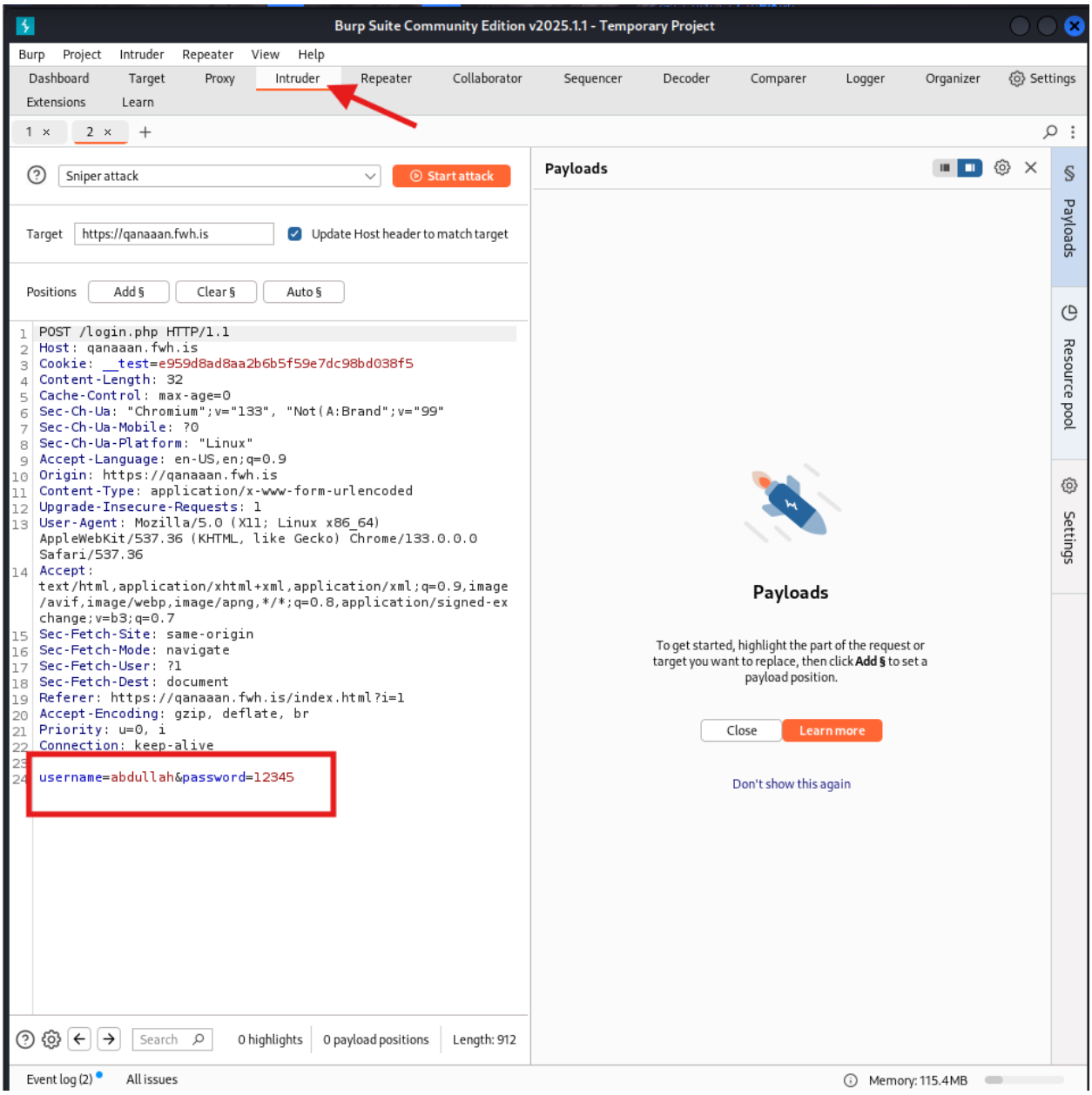
We've captured the login request — Burp Suite intercepted it while Intercept was ON. The request shows a POST to /login.php with username and password in the body: username=abdullah&password=12345
— this is the exact format we'll use in Intruder to test our wordlists.



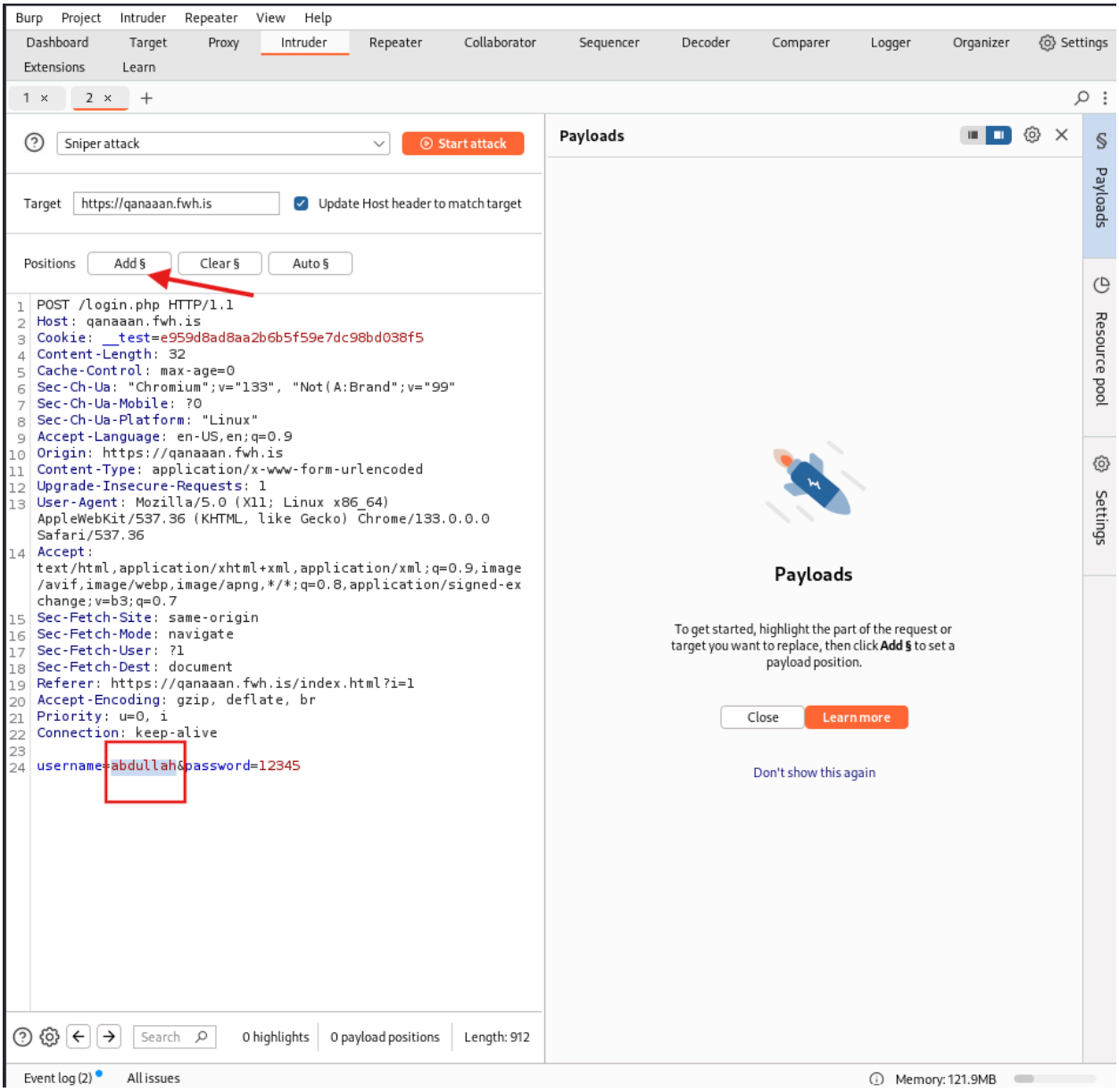
Now right-click the captured request → Send to Intruder (highlighted in red).
This sends the login request to Burp's Intruder tool — where we'll set up our username and password wordlists to automate guessing.



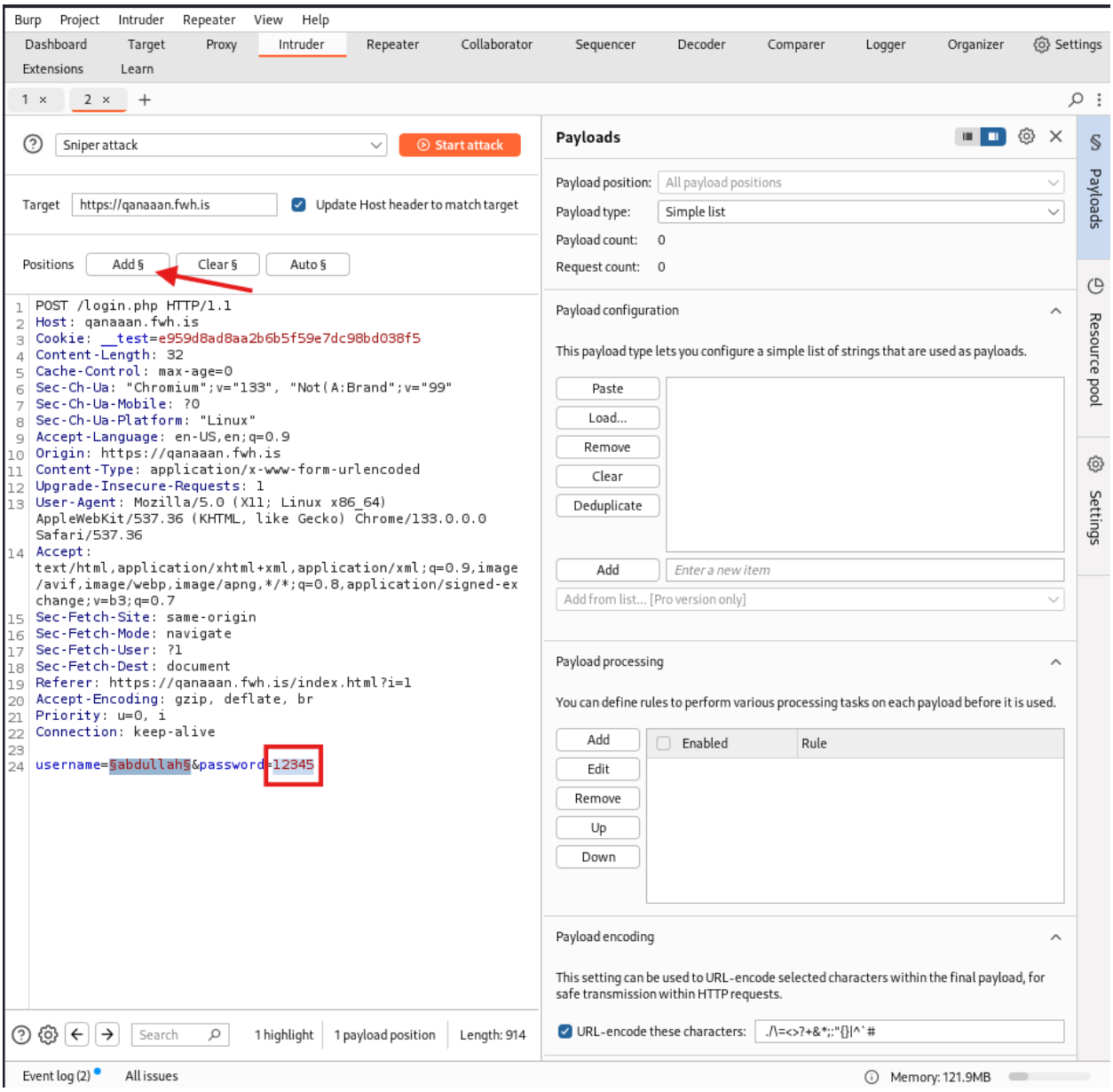
After sending the request to Intruder — turn Intercept OFF (as shown).
This lets Burp release any pending requests and prevents the browser from freezing.



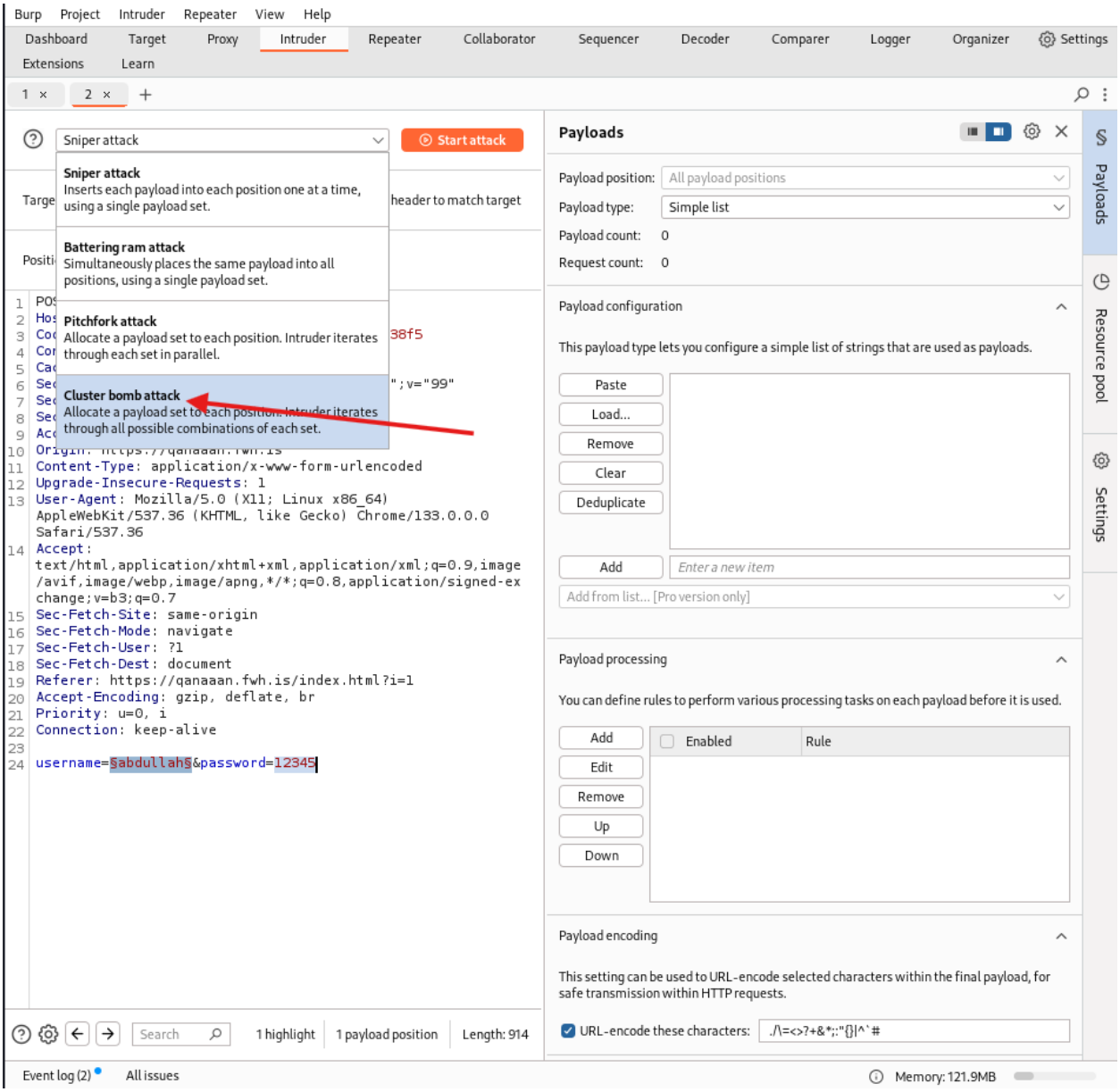
Now in the Intruder tab, we'll manipulate the request to automate guessing.



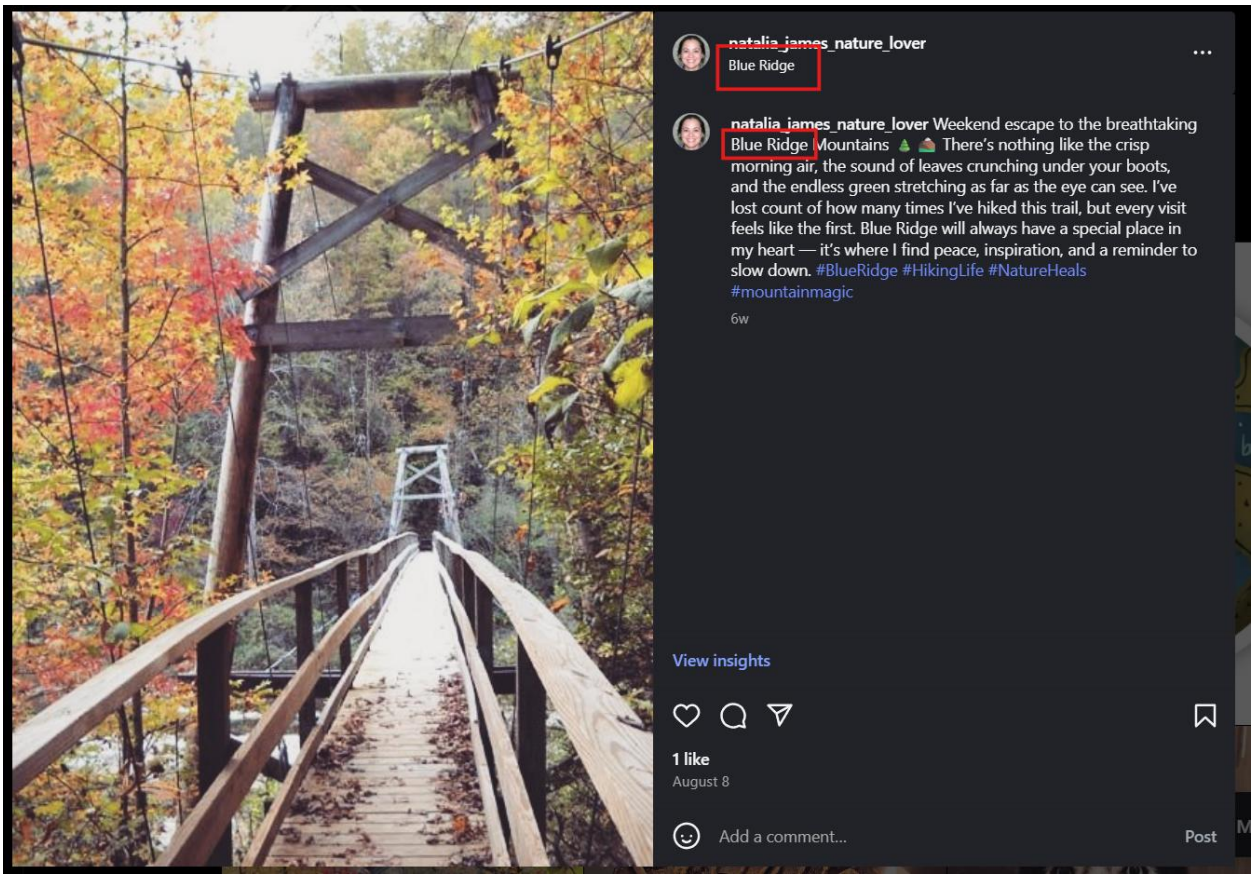
First, highlight the username value (abdullah) in the request body.
Then click “Add §” — this marks it as Payload Position 1, so Burp will replace it with usernames from your wordlist.



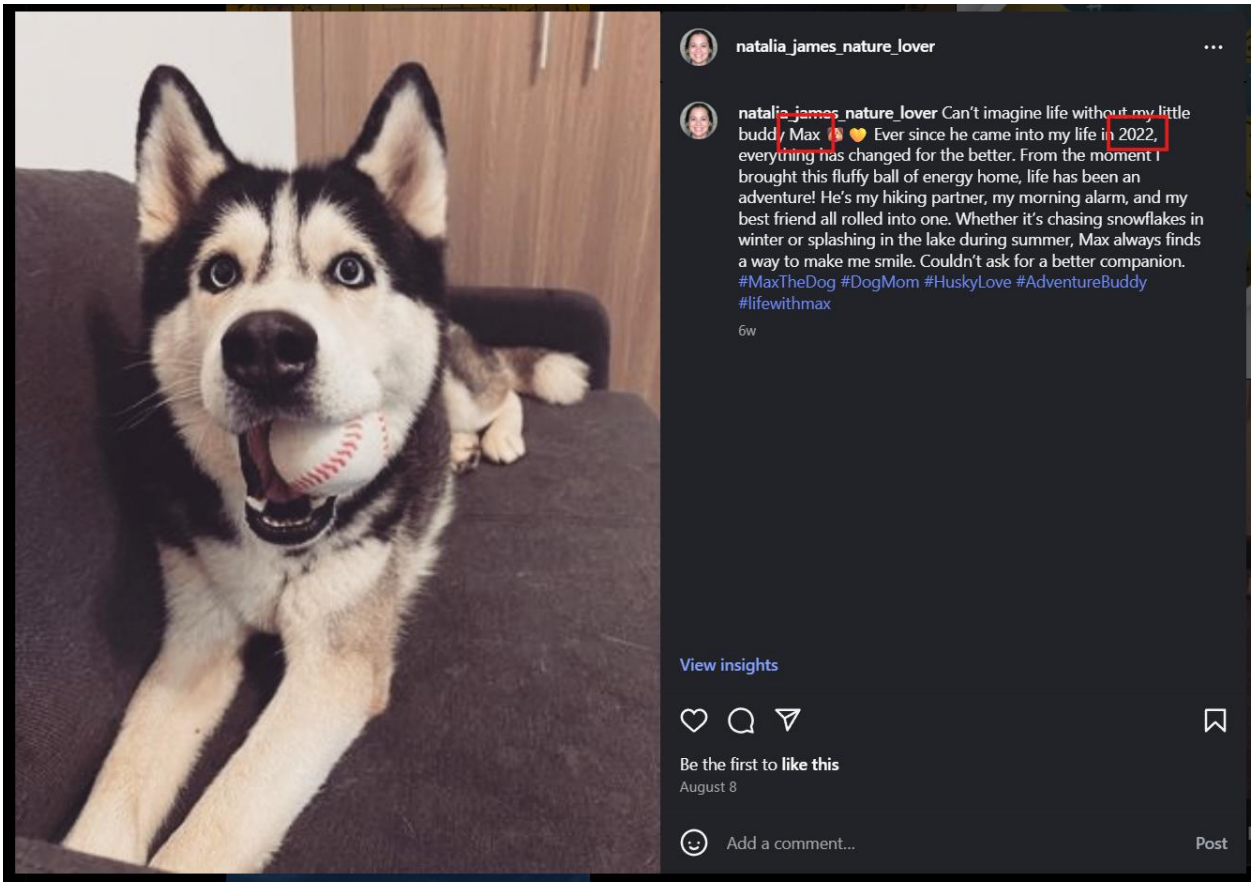
Second, highlight the password value (12345) in the request.
Then click “Add §” — this sets it as Payload Position 2, so Burp will replace it with passwords from your wordlist.



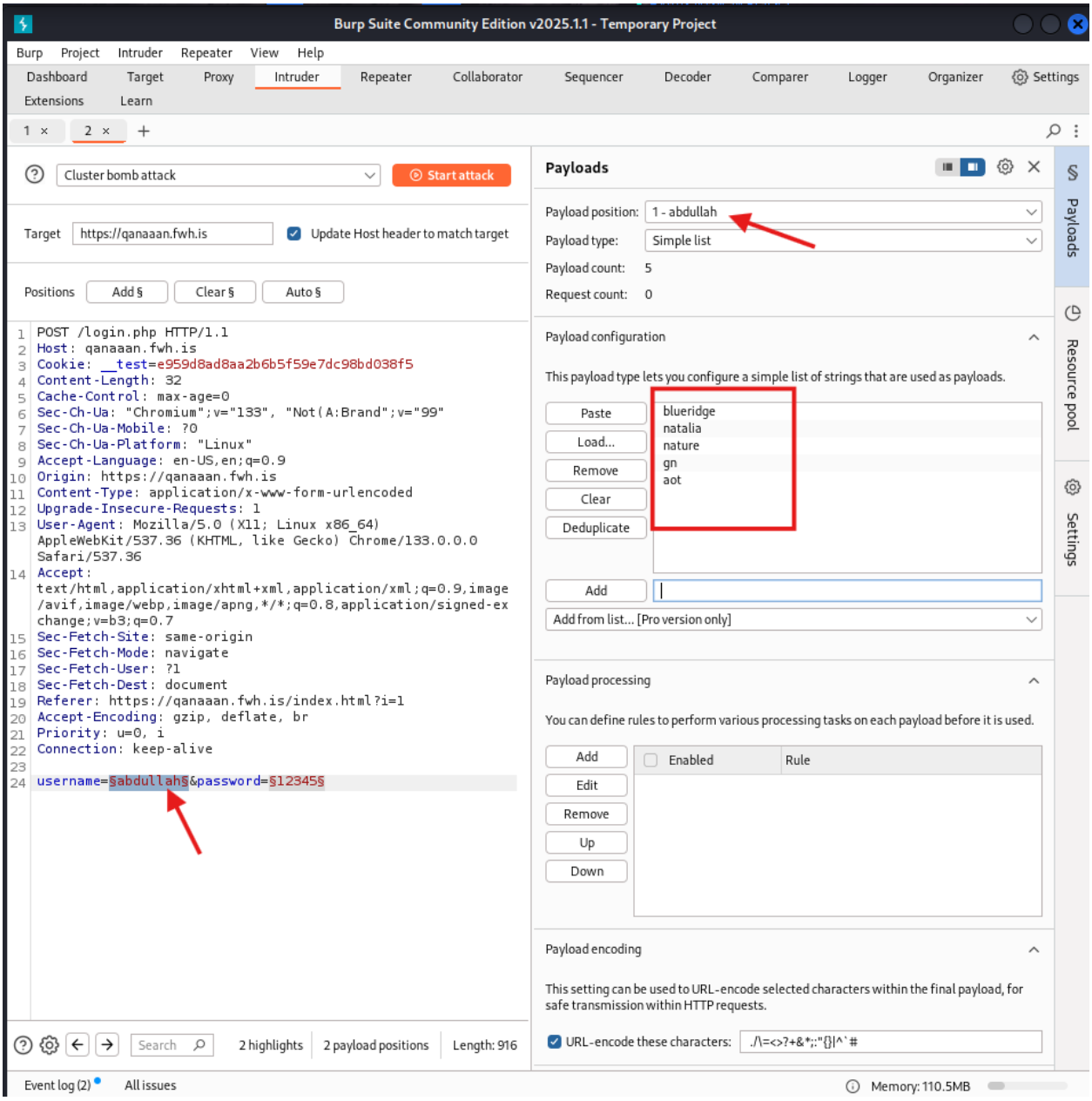
Now, change the Attack Type to “Cluster bomb” (highlighted in red).
This lets Burp try every username + every password combination — perfect for guessing login credentials.



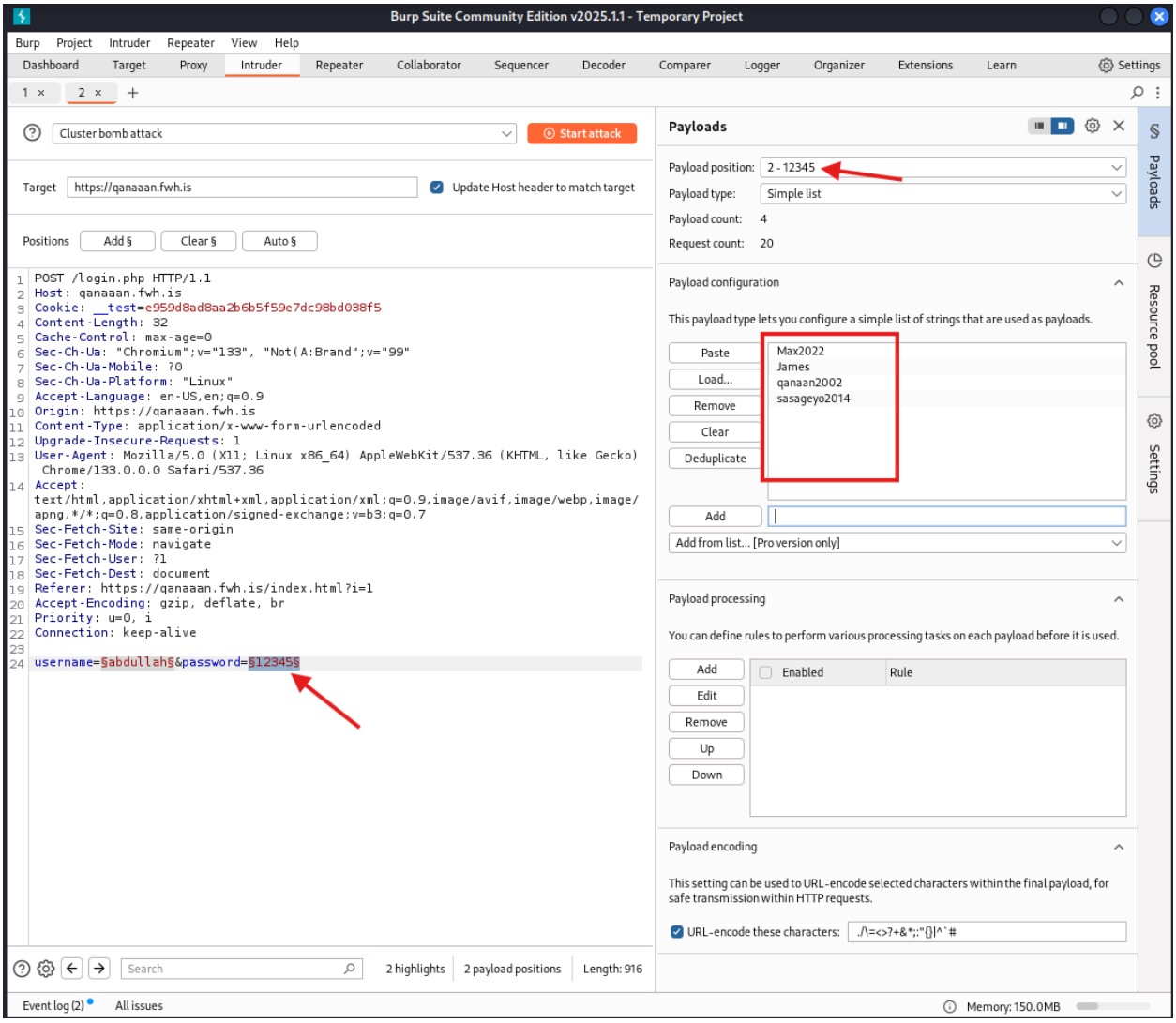
After doing OSINT on Natalia's Instagram, we found the correct username: blueridge — it's hidden in her post caption and profile info ("Blue Ridge Mountains" → lowercase = blueridge).



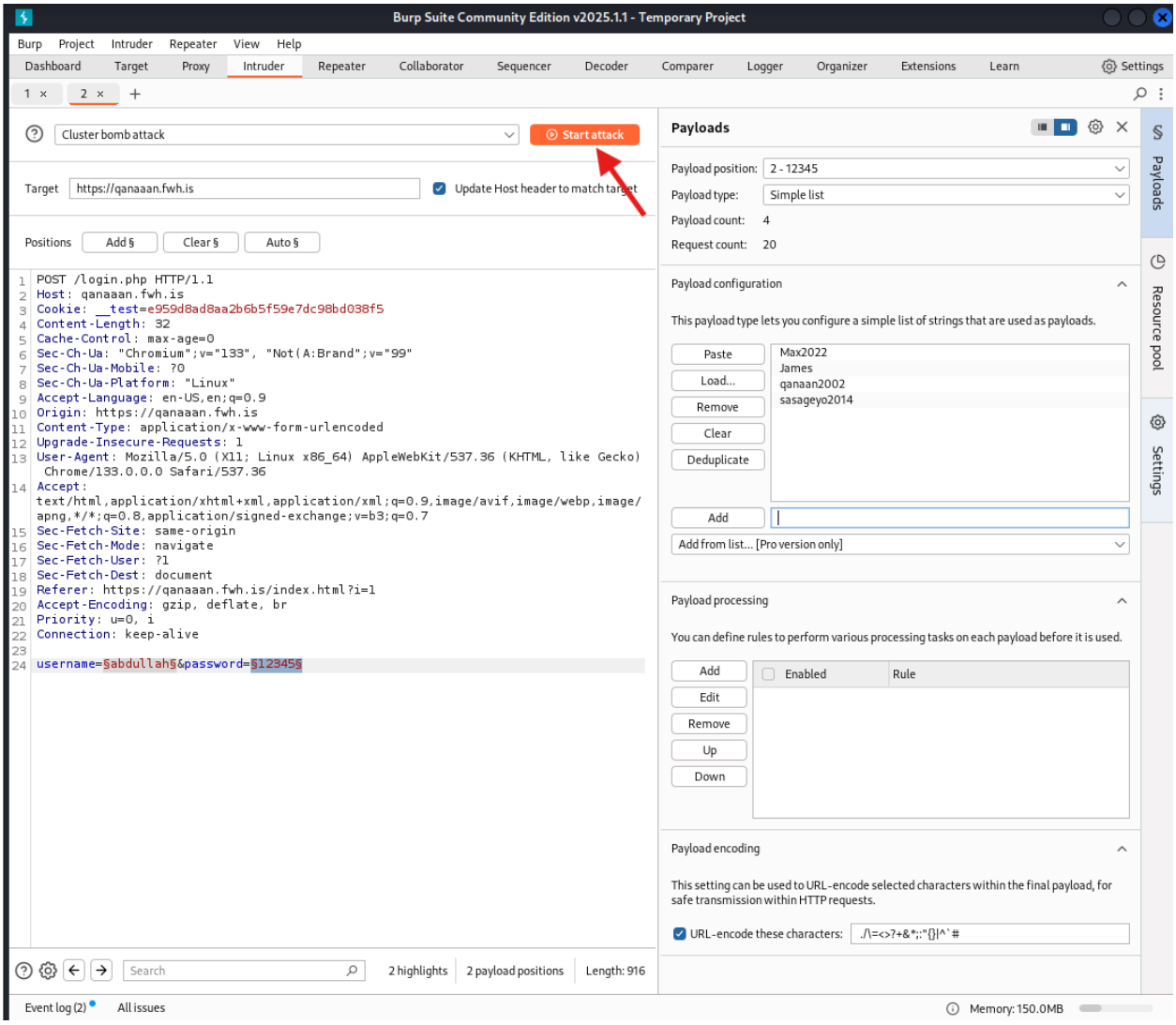
The correct password is: Max2022
Found from Natalia's Instagram post — she mentions her dog "Max" and the year "2022"



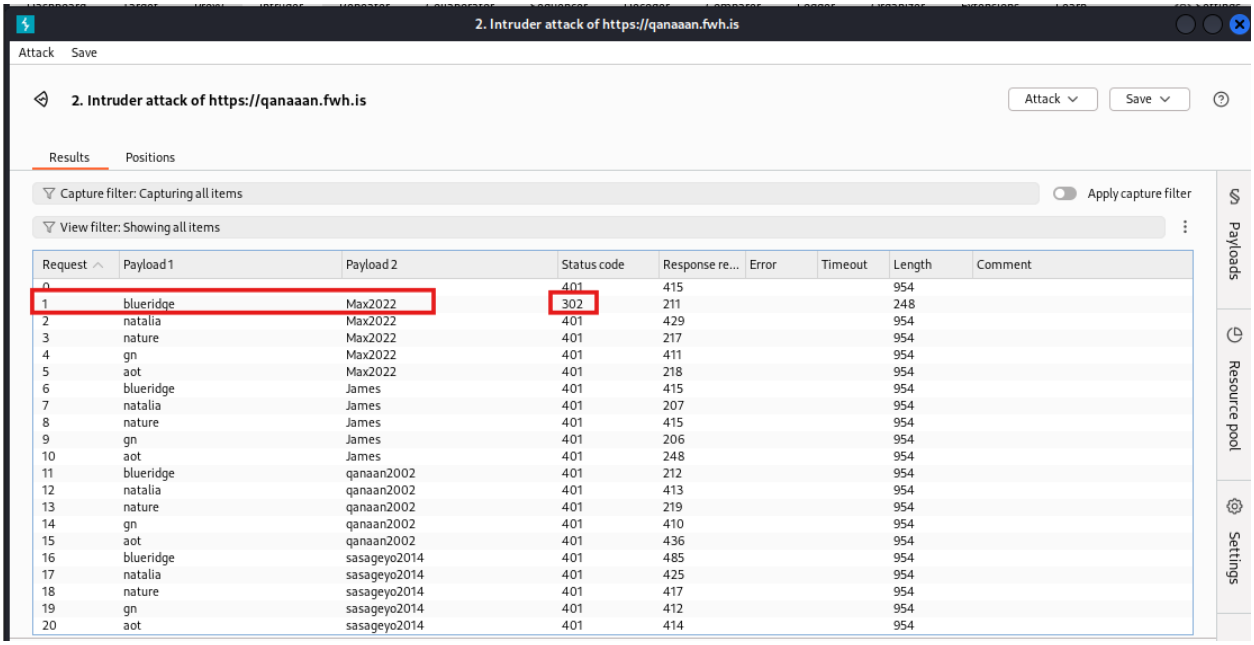
Now in Burp Suite → Intruder → Payloads tab →
We replace the test username with our correct one: blueridge. We add the correct guess within our own list we created.



For Payload Position 2 (password) — we add our custom password list, including the correct one: Max2022

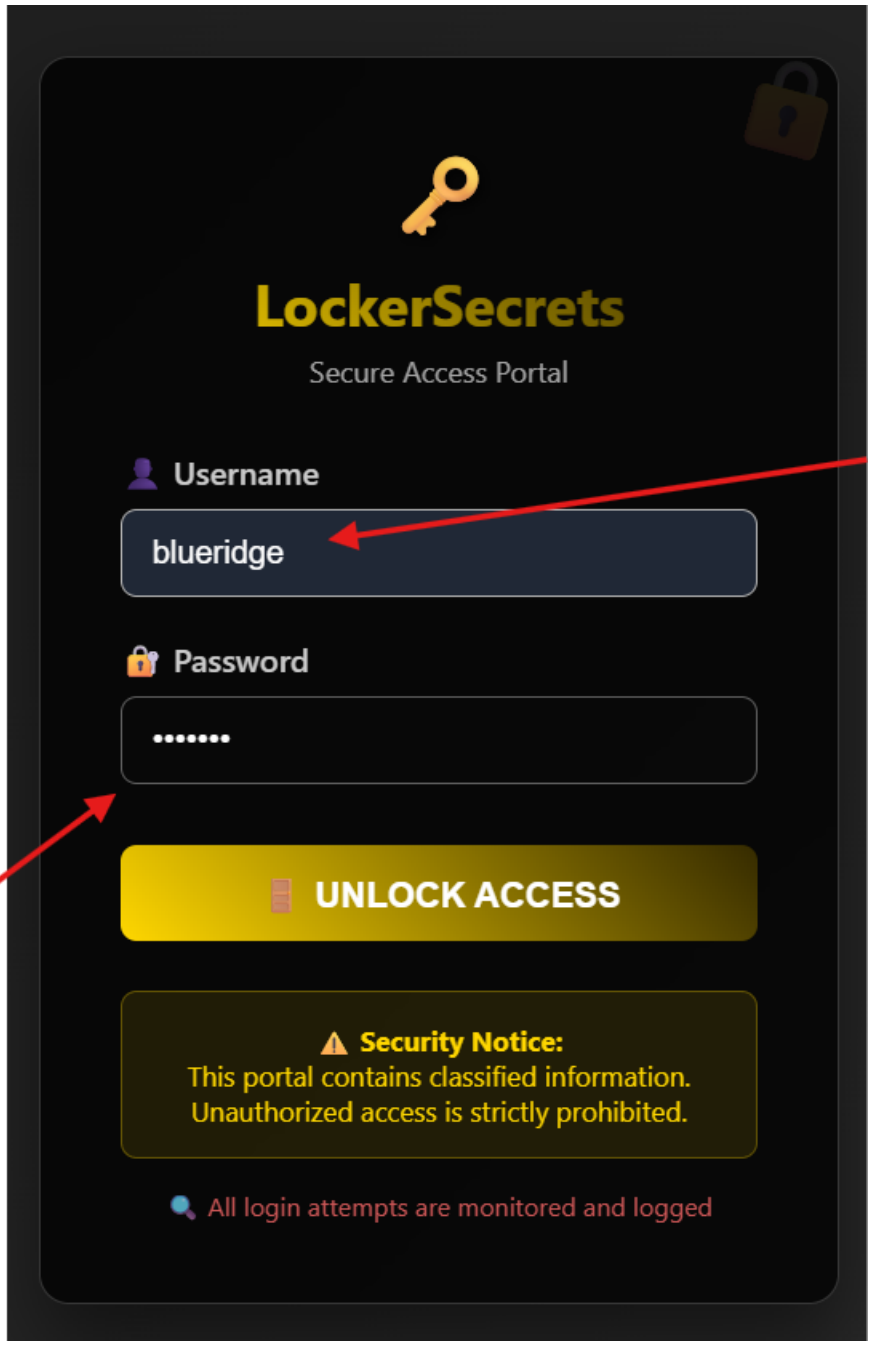


Now we start the attack

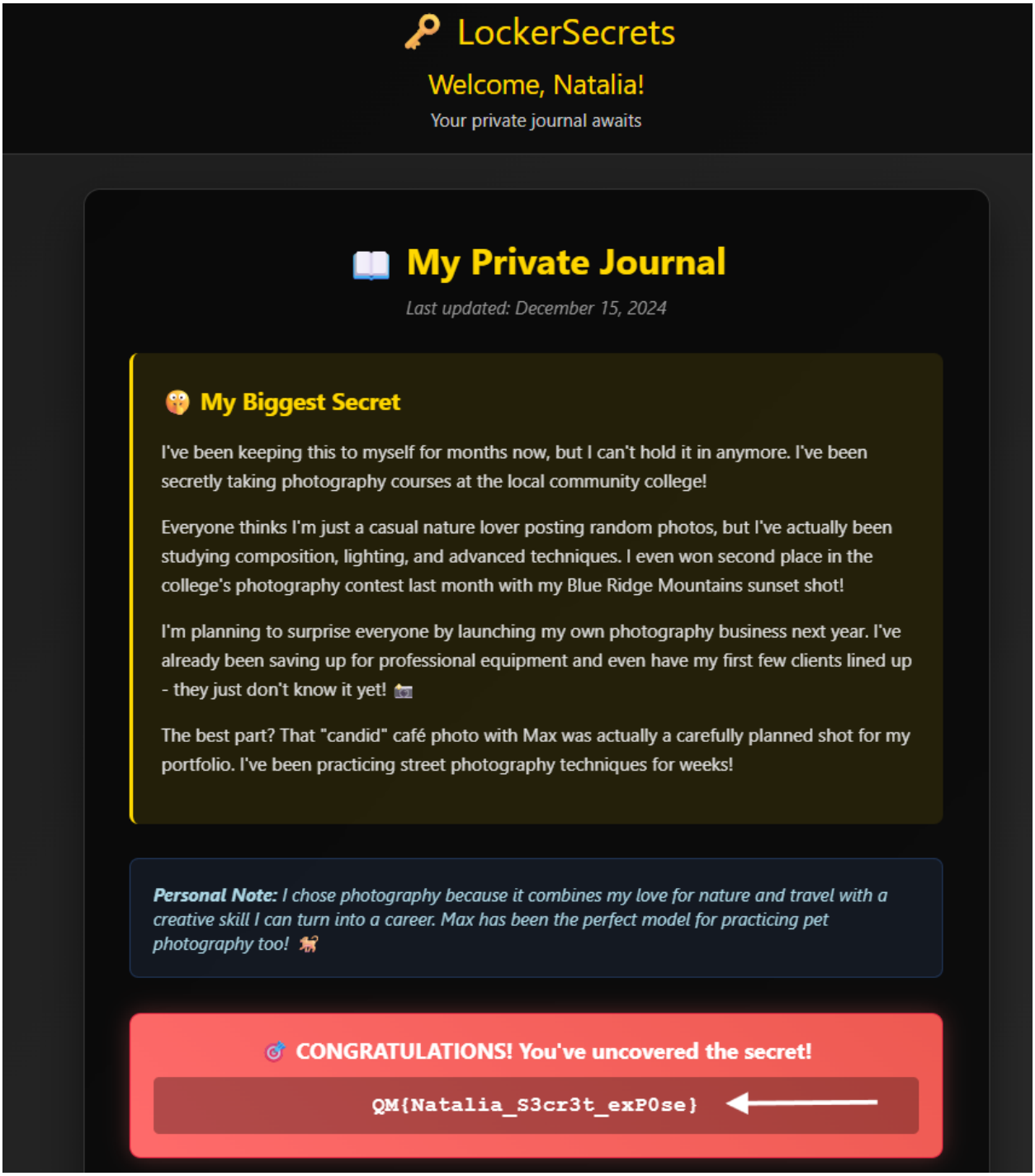


Request	Payload 1	Payload 2	Status code	Response re...	Error	Timeout	Length	Comment
0			401	415			954	
1	blueridge	Max2022	302	211			248	
2	natalia	Max2022	401	429			954	
3	nature	Max2022	401	217			954	
4	gn	Max2022	401	411			954	
5	aot	Max2022	401	218			954	
6	blueridge	James	401	415			954	
7	natalia	James	401	207			954	
8	nature	James	401	415			954	
9	gn	James	401	206			954	
10	aot	James	401	248			954	
11	blueridge	qanaan2002	401	212			954	
12	natalia	qanaan2002	401	413			954	
13	nature	qanaan2002	401	219			954	
14	gn	qanaan2002	401	410			954	
15	aot	qanaan2002	401	436			954	
16	blueridge	sasageyo2014	401	485			954	
17	natalia	sasageyo2014	401	425			954	
18	nature	sasageyo2014	401	417			954	
19	gn	sasageyo2014	401	412			954	
20	aot	sasageyo2014	401	414			954	

Now Burp Suite will test the username blueridge and the other usernames in our list with each password in your list one by one; when it tries the correct password Max2022, the server responds with a 302 Found status code—indicating a successful login and redirect to the secret page where the flag QM{...} is hidden.



Finally, go back to the LockerSecrets login page, enter the correct credentials:
Username: blueridge
Password: Max2022
Then click “UNLOCK ACCESS” — you’ll be logged in and see the hidden flag (QM{...}) on the next page. You’ve cracked it! 🏁

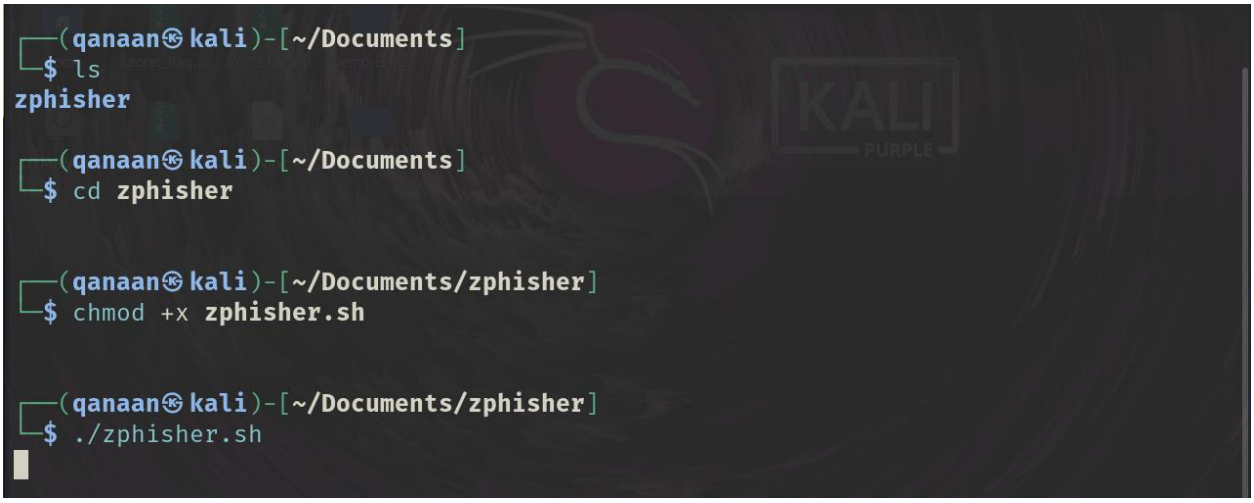


Congrats! You've successfully completed the challenge!
You logged into Natalia's secret journal using OSINT clues from her Instagram — username blueridge, password Max2022 — and uncovered her big secret: she's secretly a photographer planning to launch her own business. The flag is:
QM{Natalia_S3cr3t_exp0se}

Zphisher & MaskPhish

Zphisher is one of the most popular tools for phishing page generation (Instagram, Facebook, Gmail, etc.)

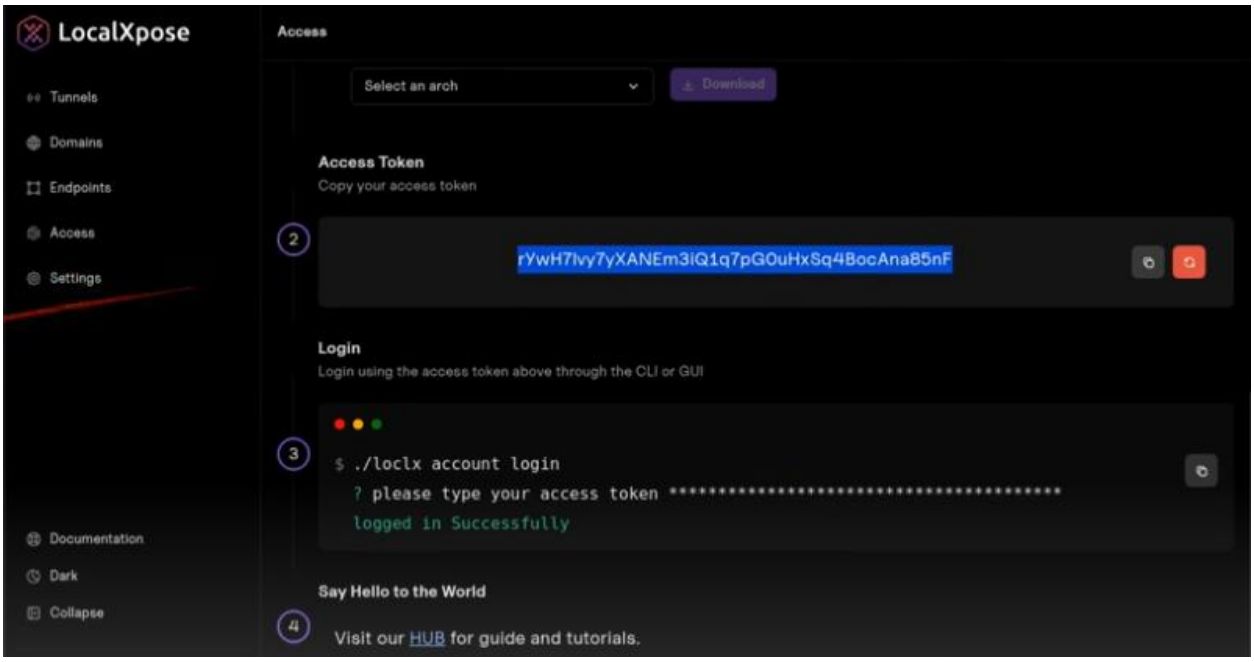
How to Download Zphisher:



Now we select option 2 — Instagram
→ Type 02 and press Enter to start building a fake Instagram login page.
Zphisher will clone the real Instagram login form so it looks 100% authentic

We select option 1 — Traditional Login Page
→ Type 01 and press Enter to create a fake Instagram login page that looks exactly like the real one.

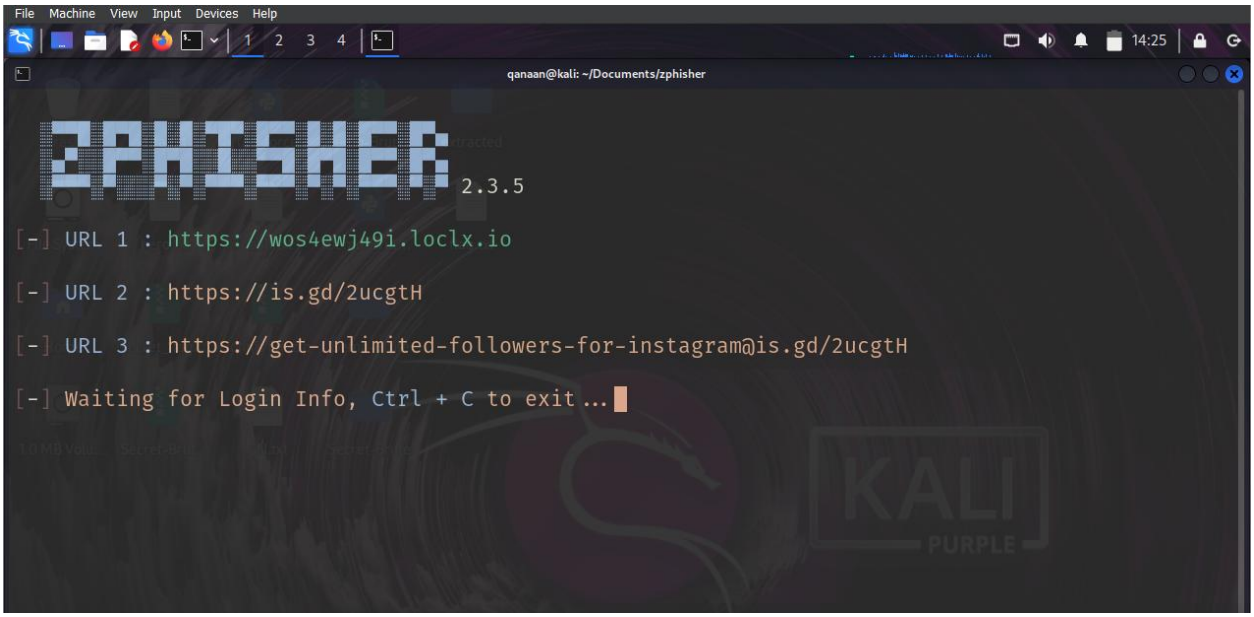
You have two choices: Option 2 is the quick and easy way to create the phishing website, while Option 3 is the longer, more manual method. I will demonstrate Option 3.



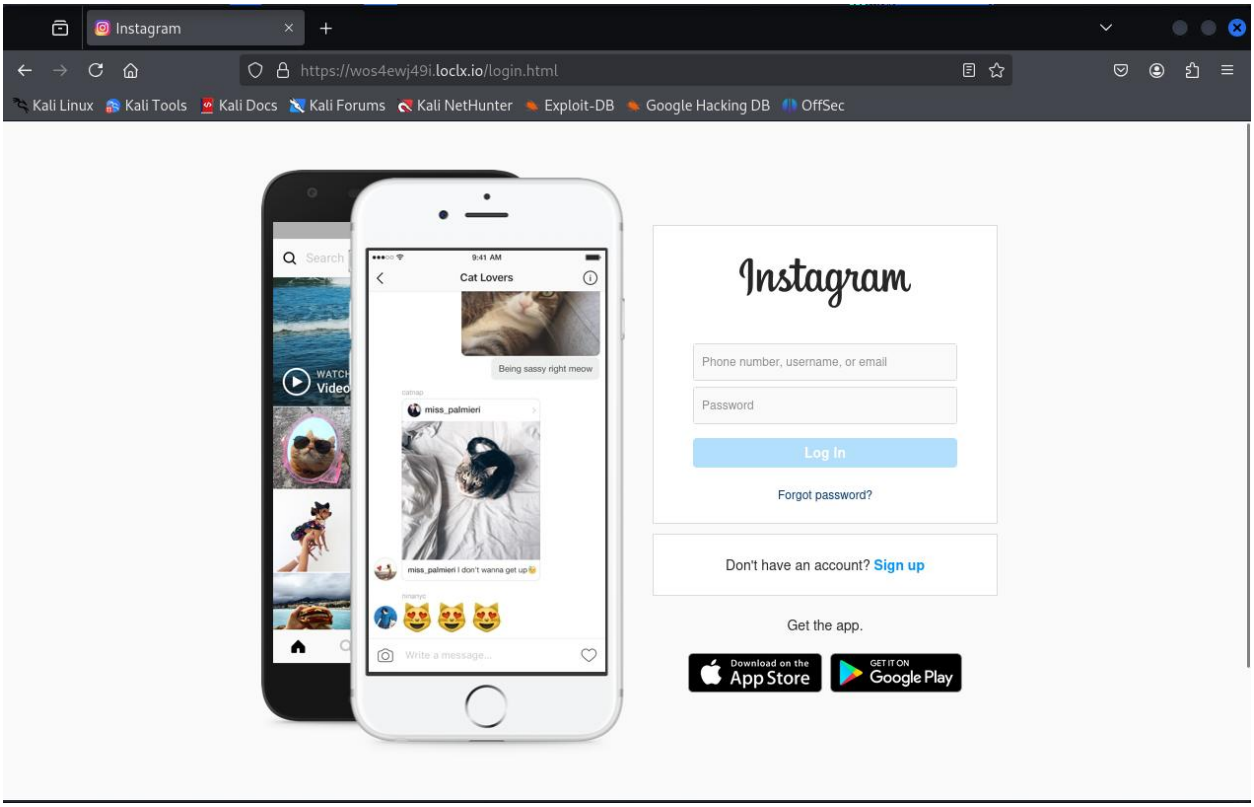
If Zphisher asks for a Token — Here's What to Do:

You need to create a free account on LocalXpose (https://localxpose.io) → then:

1. Go to Settings → copy your Access Token (highlighted in blue).
2. Back in Zphisher, when prompted, paste that token.



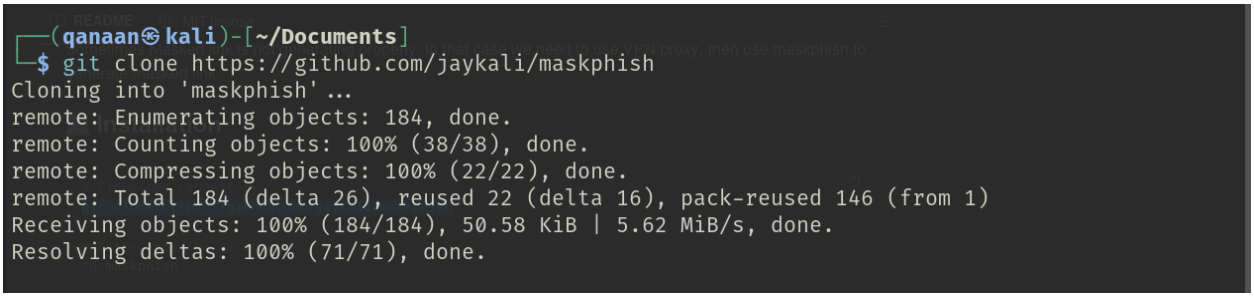
The fake Instagram phishing page is now LIVE!

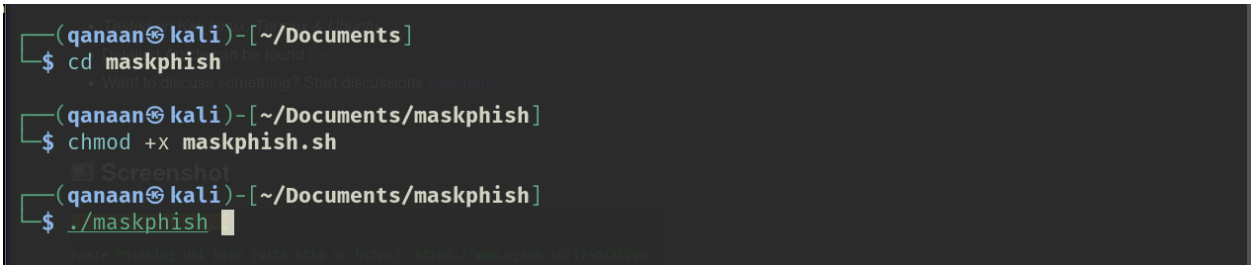


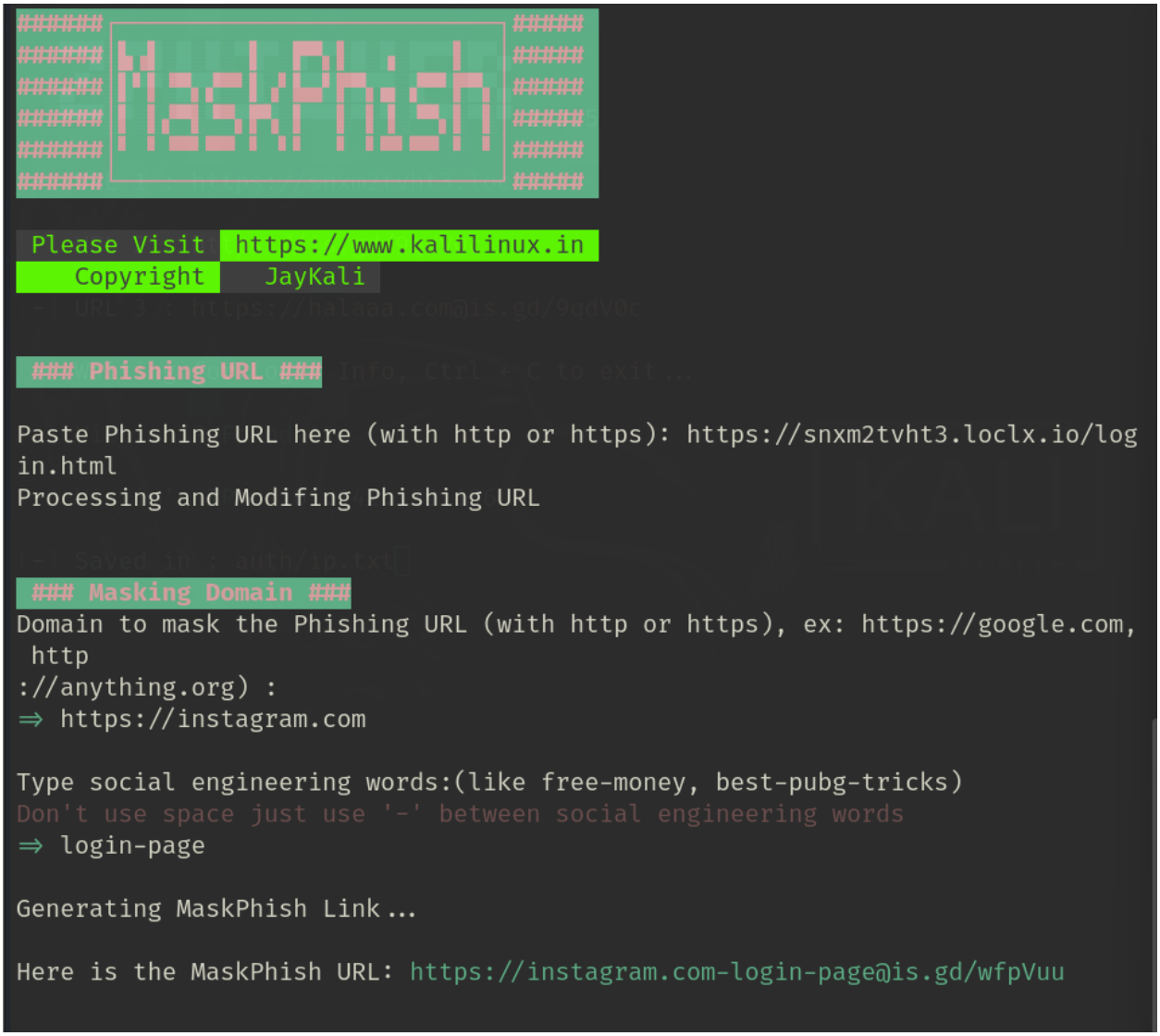
Here we go!!!

MaskPhish is a simple tool used to mask phishing URLs

How to Download MaskPhish:







How It Works:

1. Paste your real phishing URL (e.g., https://snxm2tvht3.locx.io/login.html)
2. Choose a trusted domain to mask it (e.g., https://instagram.com)
3. Add social engineering words like login-page, free-gift, or verify-account (no spaces — use -)
4. Press Enter → MaskPhish generates a sneaky link like:
   👉 https://instagram.com-login-page@is.gd/wfpVuu