# WRITEUP PICOCTF 2022

The Writeups by Coelend

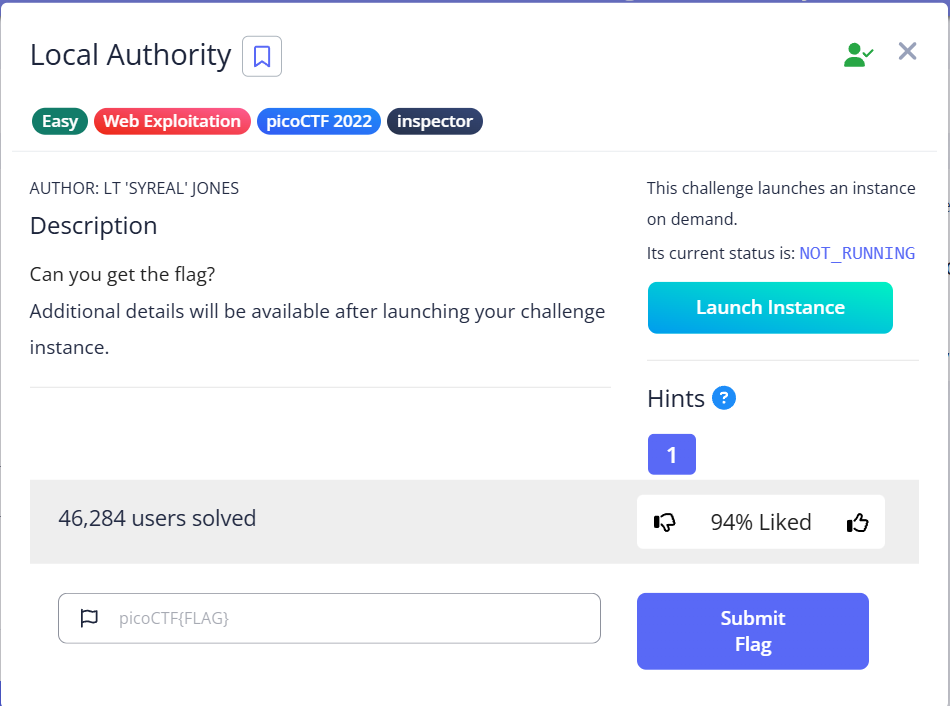


Presented by:

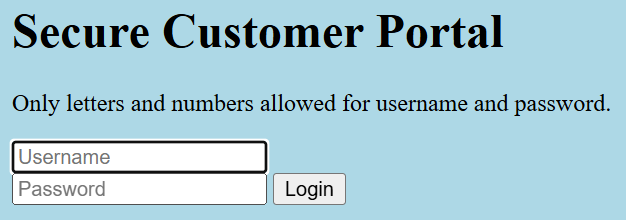
Akmal Zufa Fathurrahman (coelend)

## [WEB EXPLOITATION]

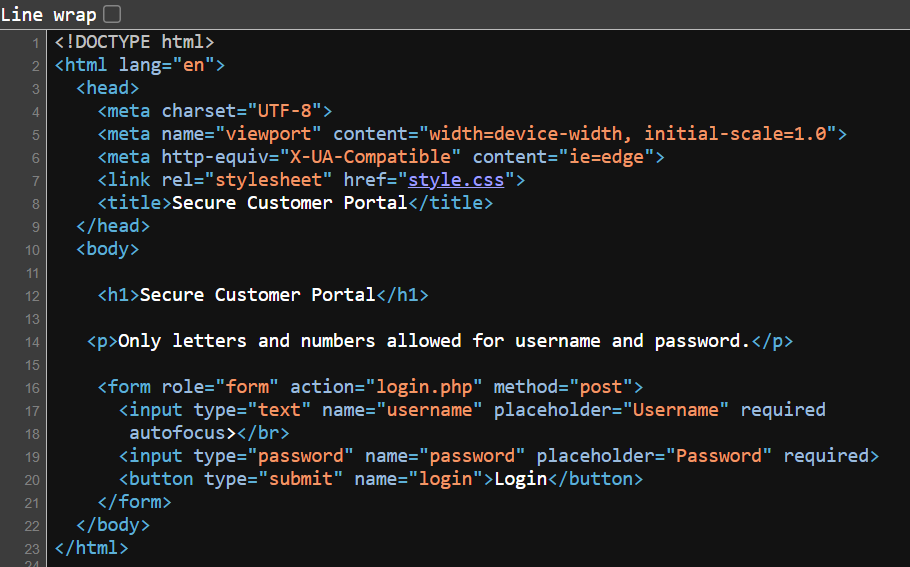
### Local Authority



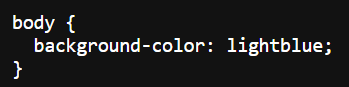
Ketika instance dijalankan, diberikan sebuah website login dengan tampilan sebagai berikut.



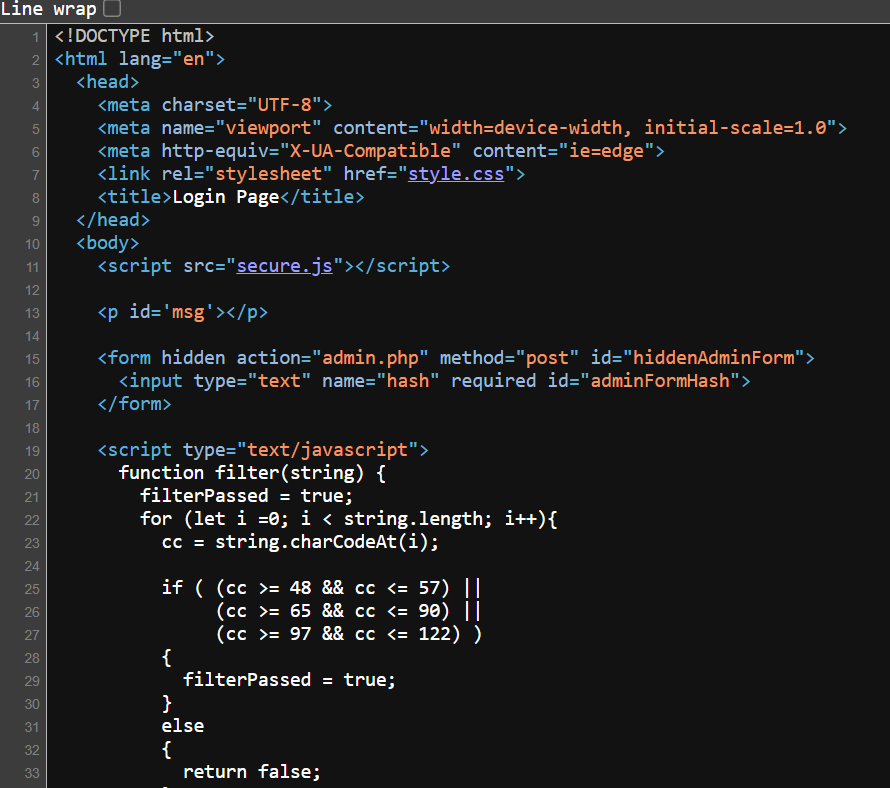
Sebagai awal analisa, kami mencoba berbagai fitur yang dimiliki oleh website ini. Website memiliki fitur form login dan halaman setelah dilakukan login, terdapat dua halaman untuk kredensial yang benar dan yang salah. Karena hanya fitur ini yang dimiliki, kami mulai melakukan analisa *source code* dari form login website ini.



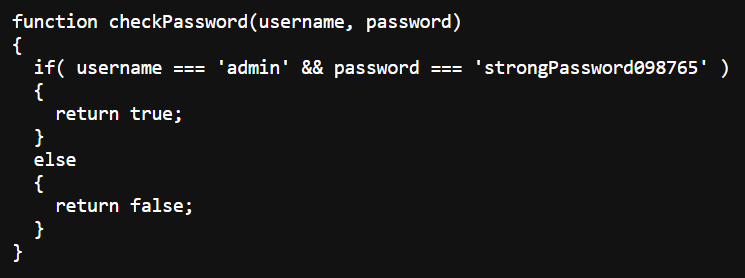
HTMl tidak memberikan informasi yang cukup berguna, namun melalui html ini kami dapat mengakses style.css. Selanjutnya dilakukan analisa file style.css.



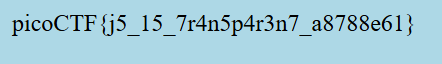
File CSS tidak memberikan informasi yang cukup berguna. Selanjutnya, dilakukan analisa *source code* halaman setelah login. Karena belum memiliki informasi kredensial yang benar, jadi untuk testing user & password bersifat random.



Pada *source code* tersebut terdapat file secure.js yang cukup mencurigakan.



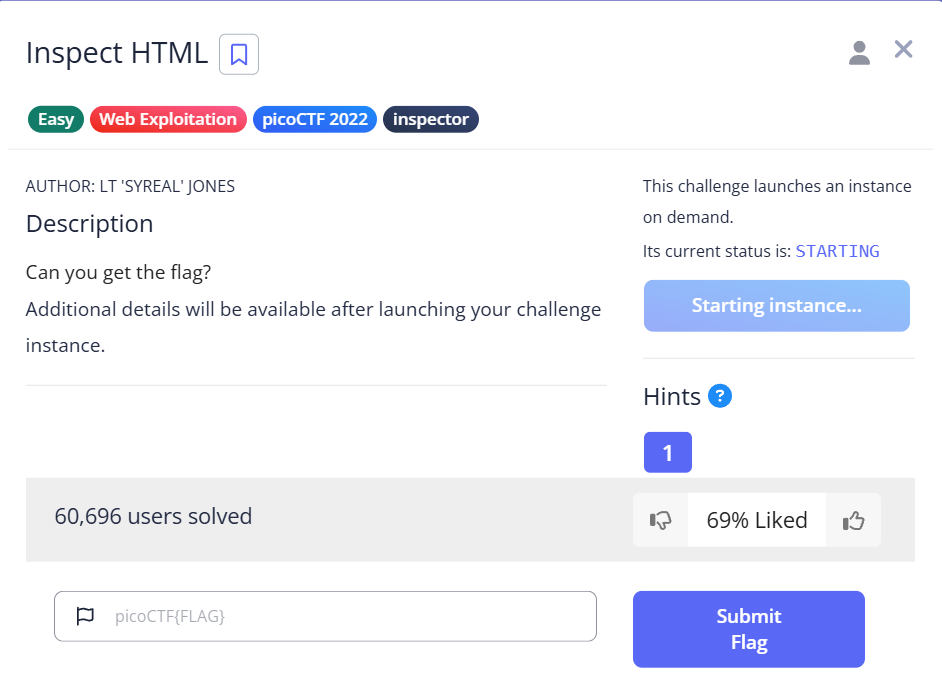
Dan benar saja, ditemukan username & password untuk user ‘admin’. Selanjutnya, lakukan login dengan kredensial tersebut.



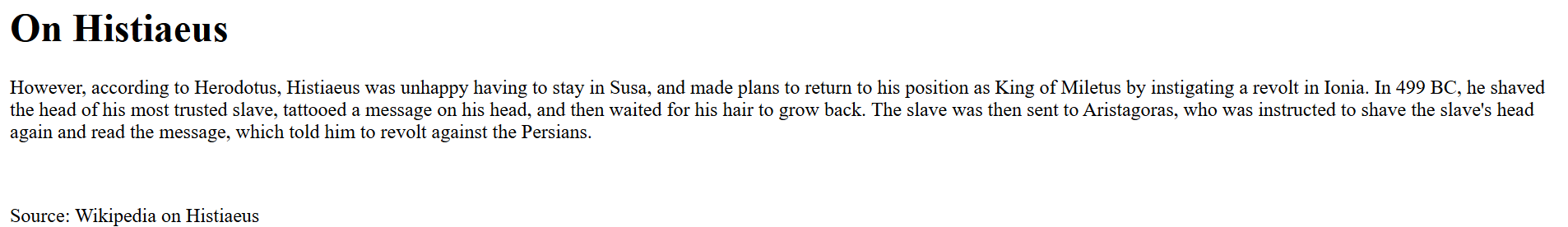
Flag berhasil didapatkan.

FLAG: picoCTF{j5\_15\_7r4n5p4r3n7\_a8788e61}

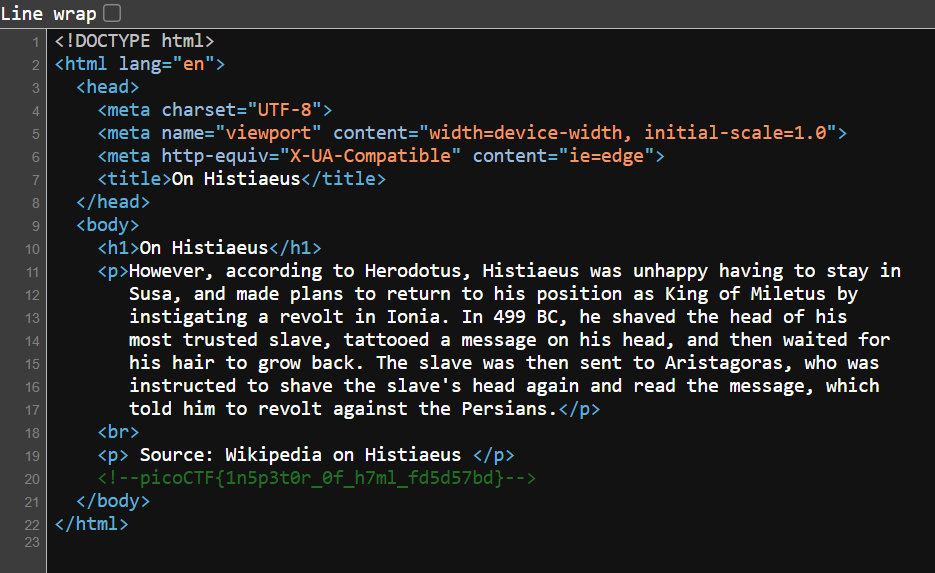
### Inspect HTML



Ketika instance dijalankan, diberikan sebuah website yang berisi sebuah paragraph yang menceritakan tentang Histiaeus dengan tampilan sebagai berikut.



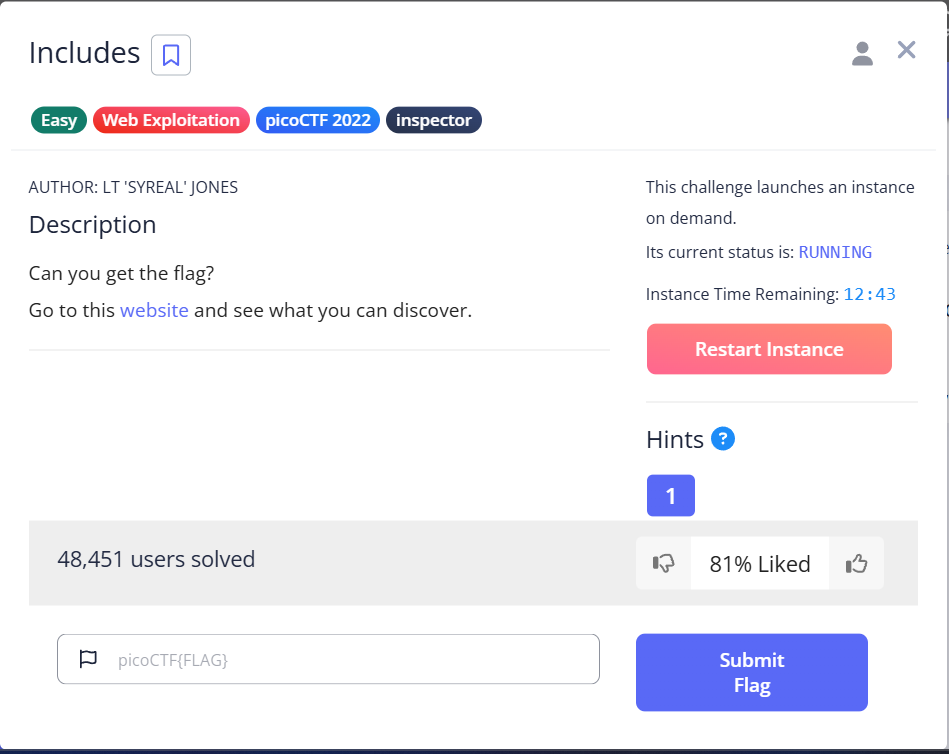
Karena tidak ditemukan fitur lain, selain informasi di atas, maka langsung saja kami lakukan pengecekan *source code* dengan cara klik kanan >> View page source.



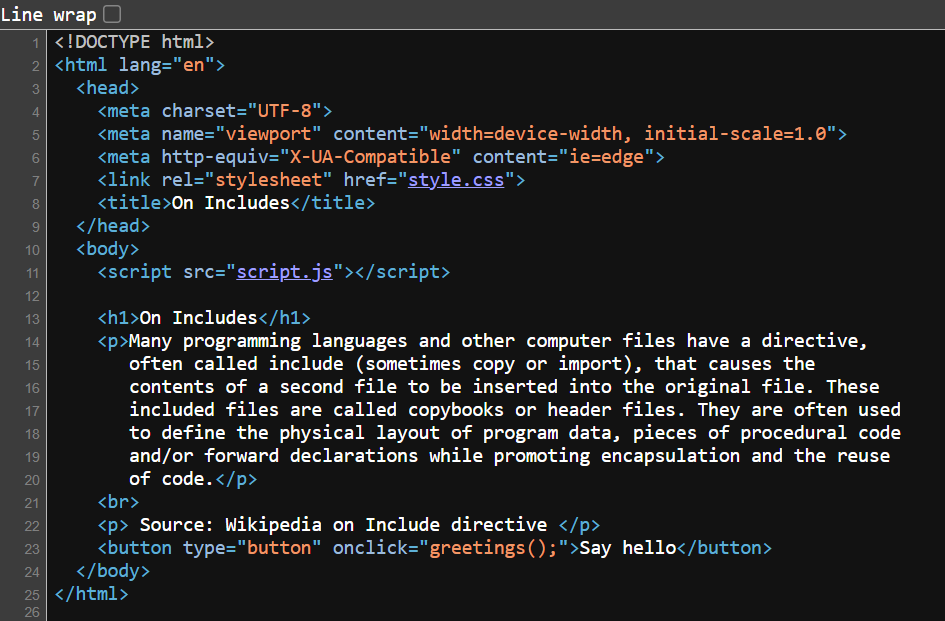
Terdapat html comment yang cukup mencurigakan setelah tag <p> terbawah. Ketika di-submit, flag valid.

FLAG: picoCTF{1n5p3t0r\_0f\_h7ml\_fd5d57bd}

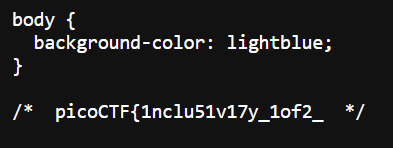
### Includes

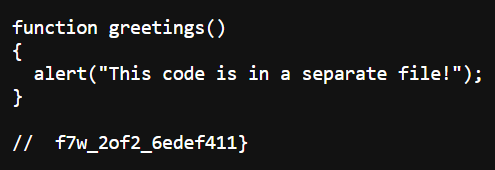


Mirip seperti pada challenge sebelumnya, untuk menemukan flagnya tinggal lakukan analisa *source code* pada tiap file yang digunakan pada website yang diberikan.



Flag dibagi menjadi dua bagian, bagian pertama berada pada file style.css dan yang kedua di file script.js

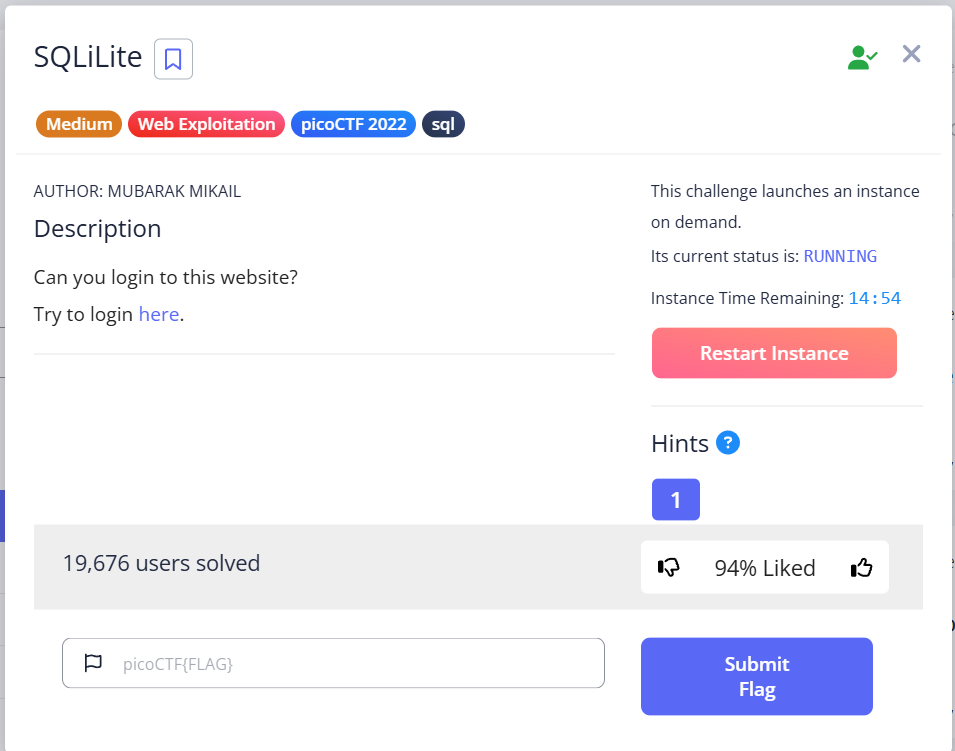




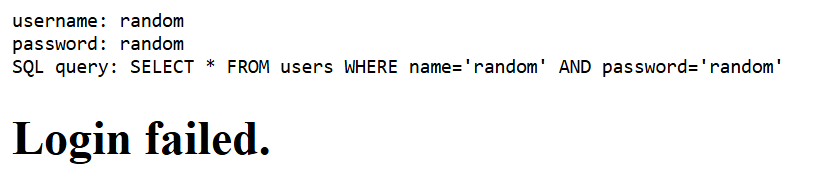
Setelah dilakukan validasi, flag berhasil di-submit.

FLAG: picoCTF{1nclu51v17y\_1of2\_f7w\_2of2\_6edef411}

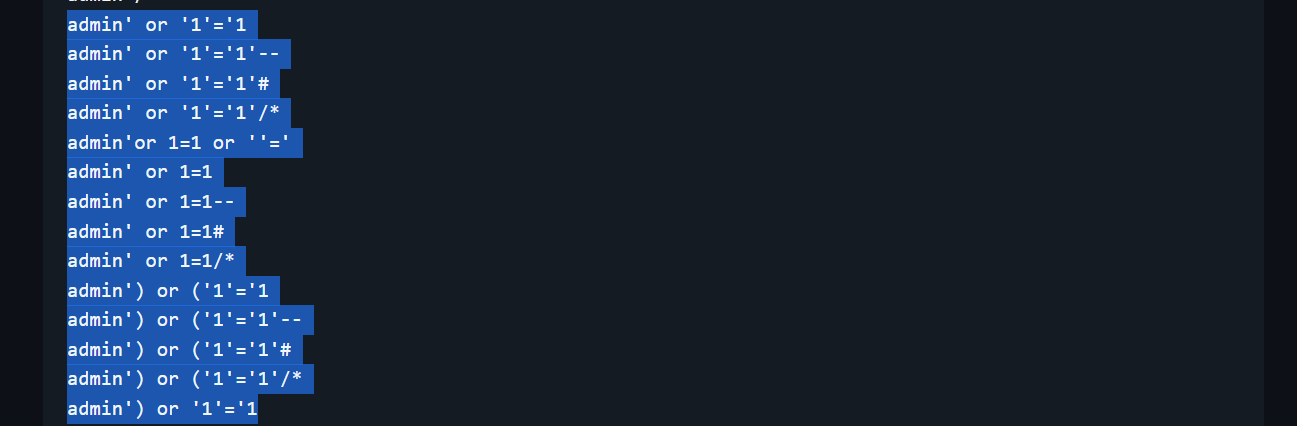
### SQLiLite [MEDIUM]



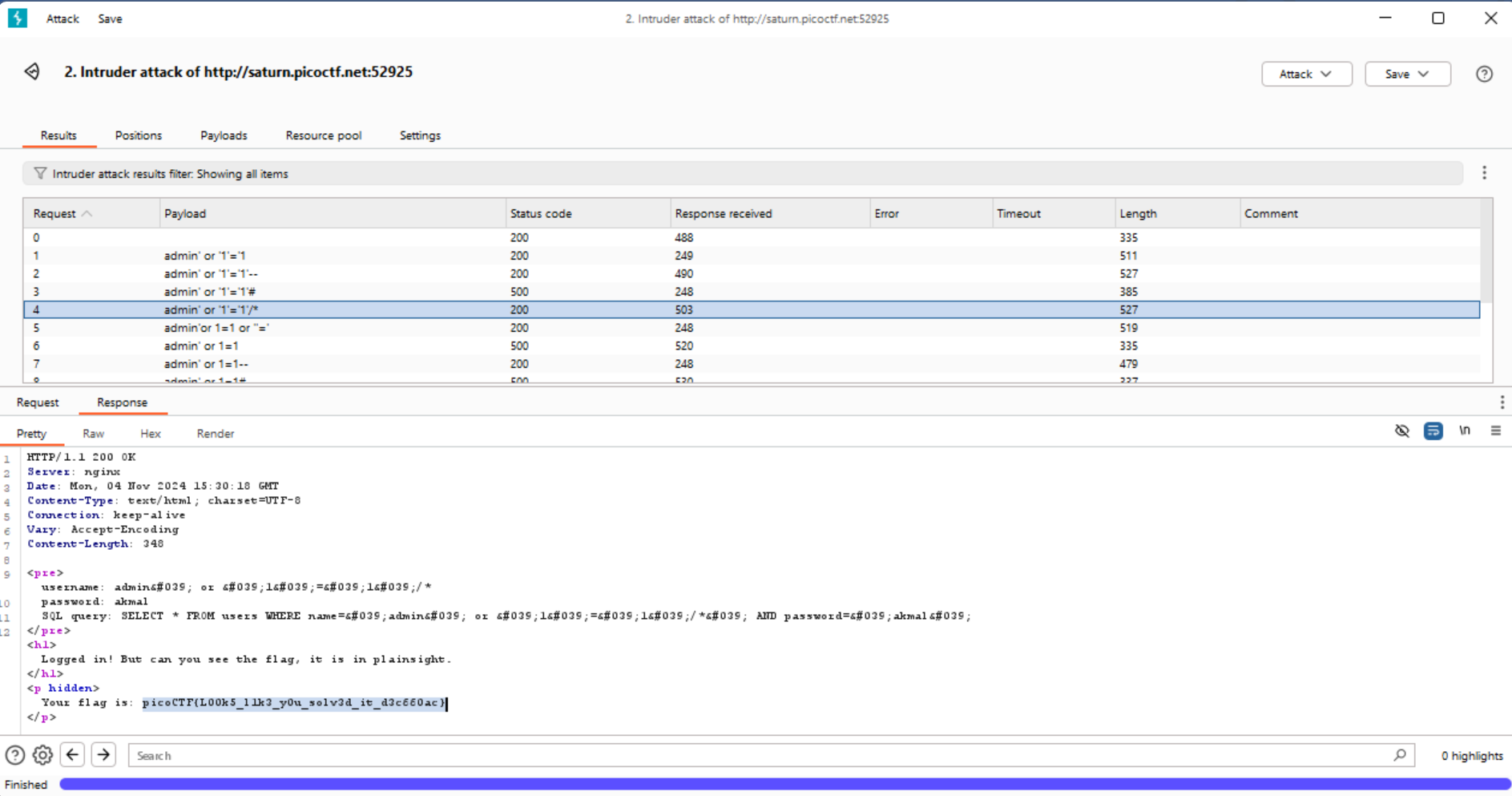
Pada challenge ini, akan diberikan sebuah website yang memungkinkan untuk melakukan login. Setelah dilakukan uji coba login dengan kredensial acak didapatkan hasil sebagai berikut.



Terdapat teks keluaran yang cukup aneh, teks merupakan SQL query. Hal ini mungkin mengindikasikan adanya kerentanan SQLi. Kemudian kami melakukan percobaan untuk menggunakan payload pada github <https://github.com/payloadbox/sql-injection-payload-list>.



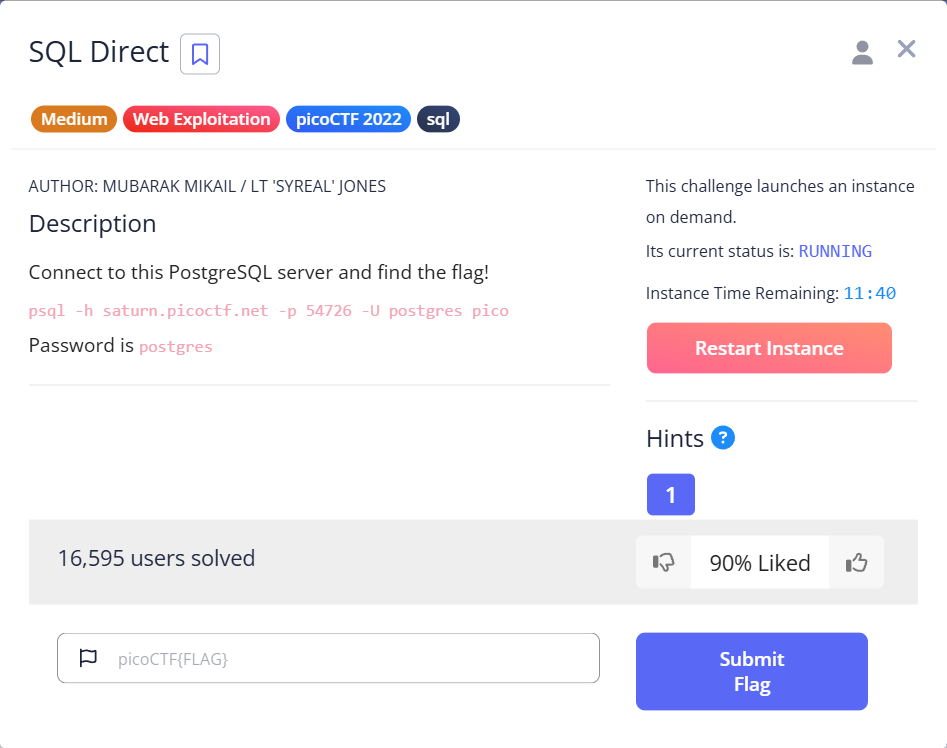
Untuk mempermudah proses ini, kami menggunakan burp suite intruder. Berikut hasil yang diberikan.



Seperti dapat dilihat pada gambar di atas, kita berhasil login dan berhasil mendapatkan flag.

FLAG: picoCTF{L00k5\_l1k3\_y0u\_solv3d\_it\_d3c660ac}

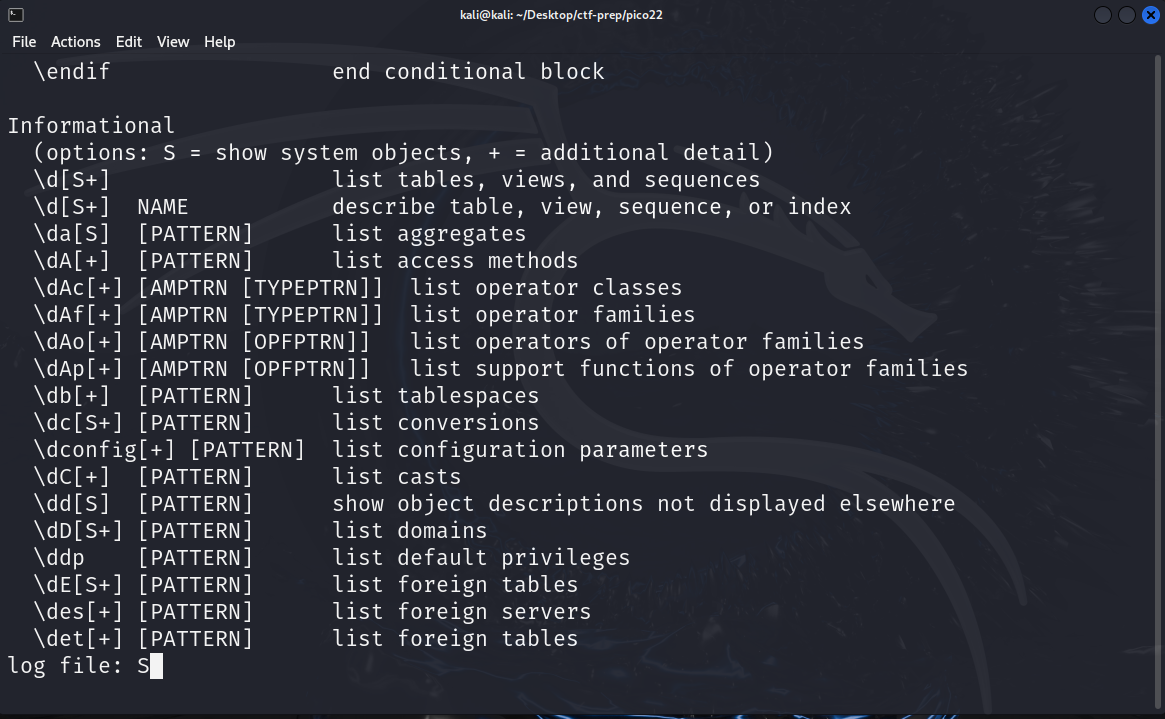
### SQL Direct



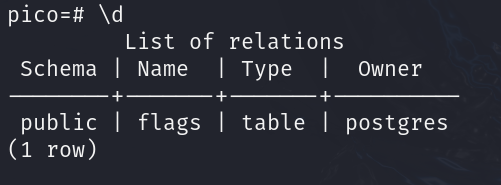
Pada challenge ini diberikan akses database PostgreSQL. Untuk mengakses database ini cukup masukkan command yang telah diberikan pada terminal.



Karena ini pertama kalinya penyusun meggunakan PostgreSQL, maka penyusun perlu mempelajari telebih dahulu beberapa perintah yang dapat digunakan. Untuk menampilkan dokumentasi (help) gunakan command ‘**\?**’.



Sebagai langkah awal, penyusun menggunakan perintah ‘**\d**’ untuk menampilkan list table, view dan sequences.



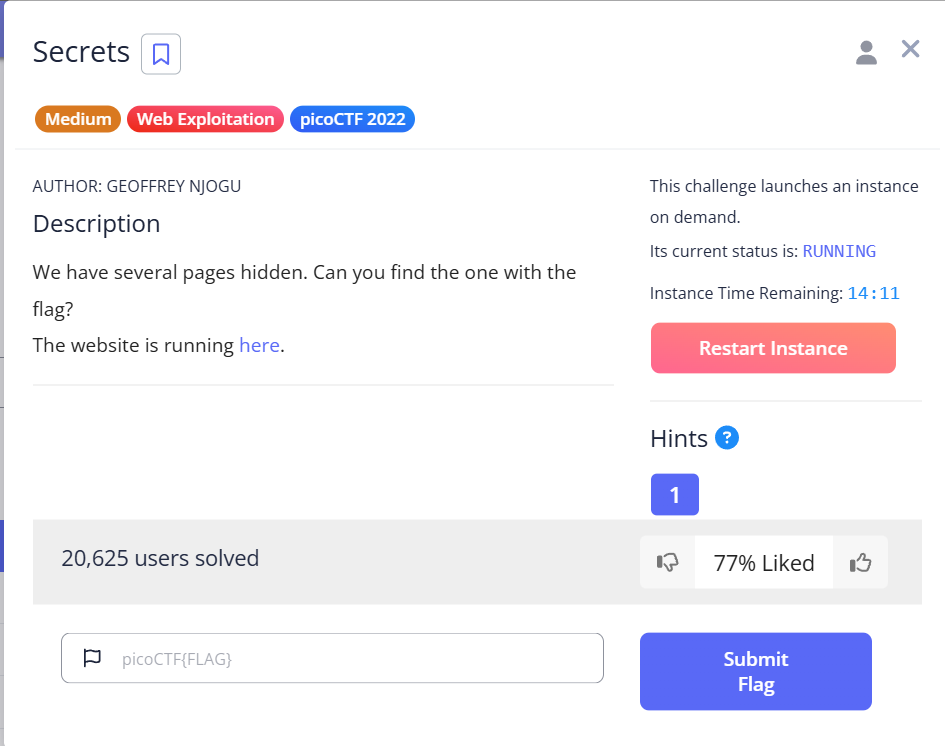
Selanjutnya, gunakan standar SQL query untuk mendapatkan data yang ada pada table flags. “SELECT \* FROM flags;”.



Flag ternyata tersimpang pada table ini, pada address milik user Luke Skywalker.

FLAG: picoCTF{L3arN\_S0m3\_5qL\_t0d4Y\_31fd14c0}

### Secrets



Challenge ini merupakan salah satu contoh dari path traversal vulnerability. Kerentanan ini memungkinkan user untuk pengakses direktori yang seharusnya tidak dapat diakses user. Untuk mengekploitasi kerentanan ini, penyusun menggunakan bantuan burp suite.

