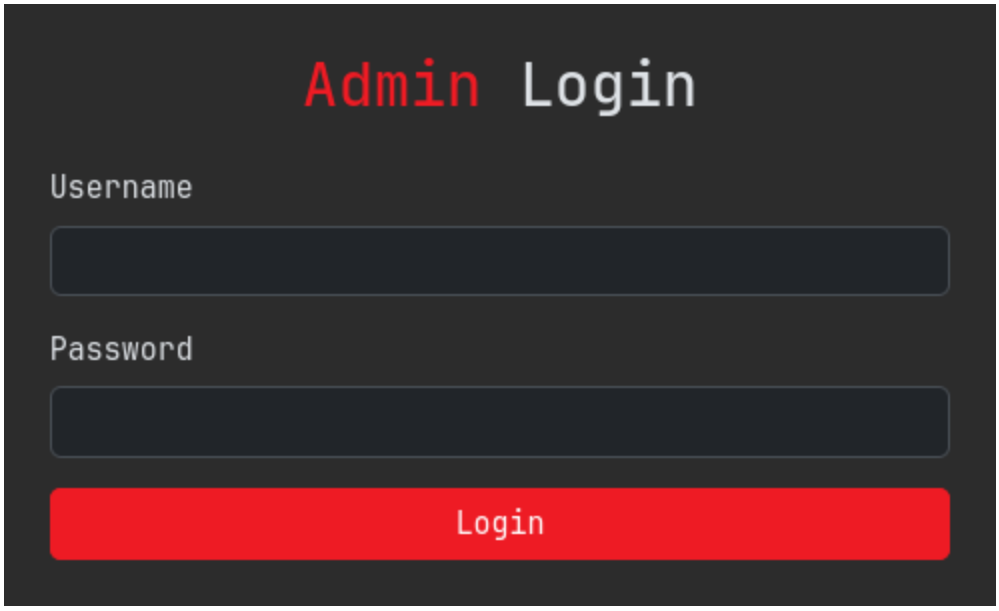
Injection

Player: constantine
Kategori: Web Exploitation

Phase 1: Initial Access (SQL Injection)

Kita langsung bertemu halaman login.



Login admin:admin tidak berhasil. Daripada menebak password, kita coba pendekatan *authentication bypass* dengan SQL Injection pada field username.

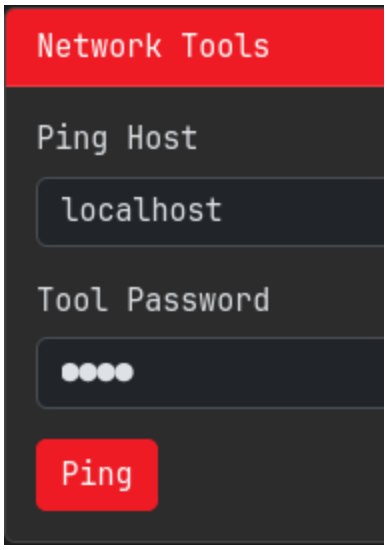
Payload yang kita gunakan:

```
' OR 1=1 -- -
```

Server tidak melakukan sanitasi input. Payload tersebut membuat query SQL selalu bernilai true → kita masuk sebagai admin.

Phase 2: Recon & Information Disclosure

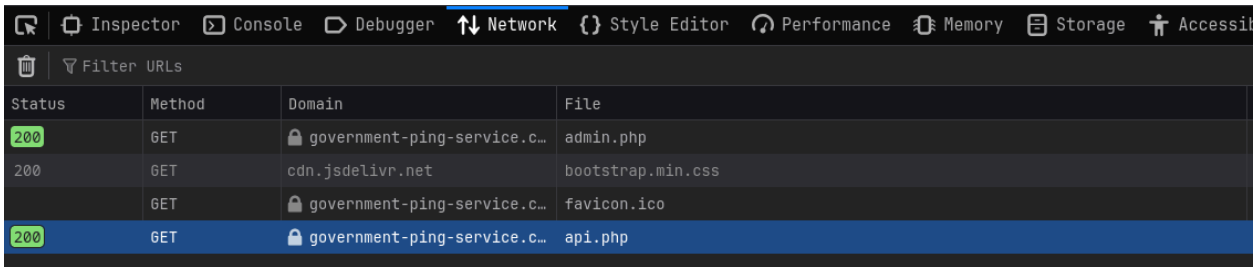
Setelah berhasil masuk, dashboard menampilkan menu *Network Tools*, tetapi fitur ini terkunci oleh sebuah **Tool Password**.



Password tidak diberikan di UI, jadi kita membuka *Developer Tools* → *Network* untuk melihat request internal.

Saat merefresh halaman, kita menemukan request menuju:

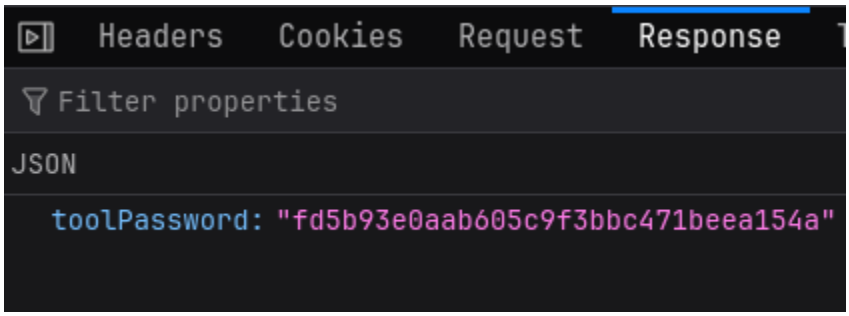
```
api.php
```



Ketika kita buka *Respons*enya, server **mengirimkan password secara terang-terangan dalam JSON**.

Begini responsenya:

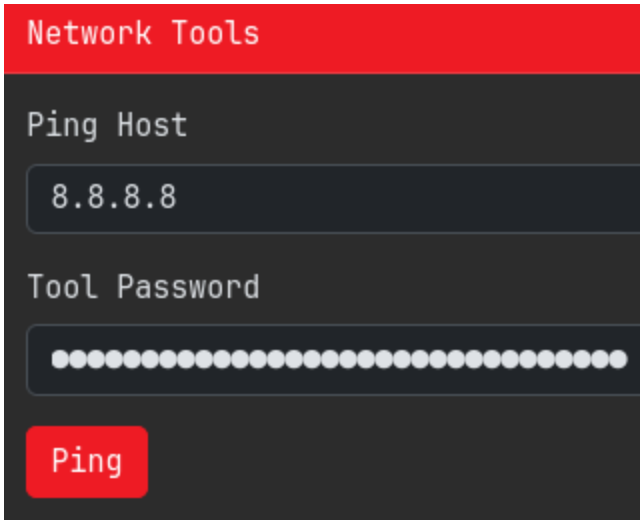
```
toolPassword: "fd5b93e0aab605c9f3bbc471beea154a"
```



Ini murni *Information Disclosure*. Password sudah diberikan langsung oleh server.

Phase 3: Execution (OS Command Injection)

Setelah memasukkan password, fitur Ping dapat digunakan.



Kita uji dengan command injection sederhana menggunakan:

```
localhost; cat /flag.txt
```

ping localhost dieksekusi lebih dulu, lalu server mengeksekusi cat /flag.txt.
File flag muncul langsung di output menandakan RCE (*Remote Command Execution*) sepenuhnya bekerja.

Final Flag

FORESTY{sqli_xss(just_see_net-traffic&localstorage)_command_injection_is_EASY}