

Web 5

CHALLENGE DESCRIPTION

<http://ideweb2.hh.se/~andalf20/ctf/injectme/>

Mitt admin-lösenord är 128 slumpmässiga tecken. Du kommer aldrig kunna knäcka det!

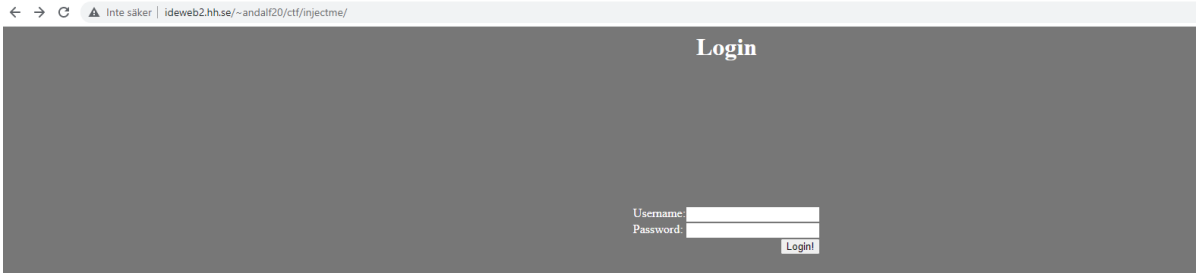
Flaggformat: CTF{<FLAGGA>}

TOOLS USED

- <https://cheatsheet.haax.fr/web-pentest/injections/server-side-injections/sql/>

SOLUTION

Går man till hemsidan som länkas, så ser den ut så här:

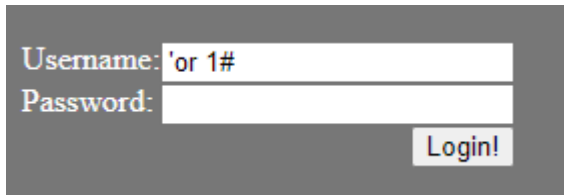


En inloggningsruta där man har möjlighet att skriva in både användarnamn och lösenord. En ledtråd ges redan i URLen. Där står det “injectme”. Det kan då betyda att man ska utföra någon typ av injektion för att förbigå inloggningen utan att veta varken användarnamn eller lösenord. Då användarnamn och lösenord kan lagras i en databas så kan man utnyttja ett eventuellt säkerhetsproblem i hanteringen av indata, om koden inte är säkrad mot detta dvs.

Målet är då att försöka manipulera SQL-förfrågan som görs mot databasen.

SQL-satsen kan t.ex. se ut så här: **select * from user_login where user=''** and **pwd=''**, där värdet inom apostroferna är det man skriver i

inloggningsfälten. Om man anger 'or 1# i fältet för username manipulerar vi SQL-satsen. SQL-satsen ser då ut så här istället: **select * from user_login where user=''or 1#' and pwd=''**. Denna SQL-sats returnerar aldrig ett tomt "result set" och returnerar alltid "true". På så sätt har vi således kommit förbi autentiseringen.



A screenshot of a login interface with a dark grey background. It features two white input fields. The first field is labeled 'Username:' and contains the text 'or 1#'. The second field is labeled 'Password:' and is empty. To the right of the password field is a white button with the text 'Login!' in black.

Klickar vi sedan på Login! ser vi följande:

CTF{41w4Y5_u53_Pr3P4r3D_qU3r135}

FLAG

CTF{41w4Y5_u53_Pr3P4r3D_qU3r135}