

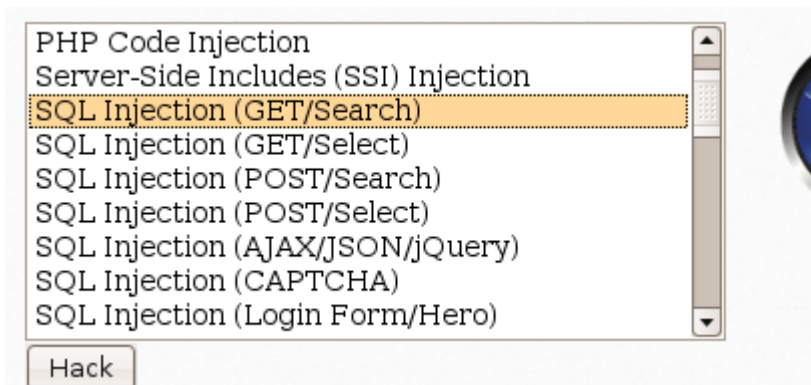
Fallos de criptografía - Hashing Débil de Contraseñas

Una implementación débil de hashing de contraseñas representa una vulnerabilidad crítica de **fallos criptográficos**, ya que en lugar de almacenar las contraseñas en texto plano, estas deben ser hasheadas (convertidas a un valor no reversible mediante un algoritmo criptográfico) antes de ser almacenadas en una base de datos. Sin embargo, si se utilizan algoritmos de hash débiles o inadecuados, como MD5 o SHA-1 (que ya no son seguros), los atacantes pueden crackearlos fácilmente usando técnicas como ataques de diccionario o fuerza bruta.

A través de este ejercicio identificaremos la vulnerabilidad con SQL Injection para obtener hashes de contraseñas desde la base de datos y explotaremos la vulnerabilidad de fallas criptográficas crackeando los hashes a través de la utilización de herramientas como **John the Ripper**. Posteriormente iniciaremos sesión con las contraseñas obtenidas para demostrar la vulnerabilidad.

Obtención de hashes de contraseñas

1. Selecciona la vulnerabilidad **SQL Injection (GET/Search)** y "Hack".

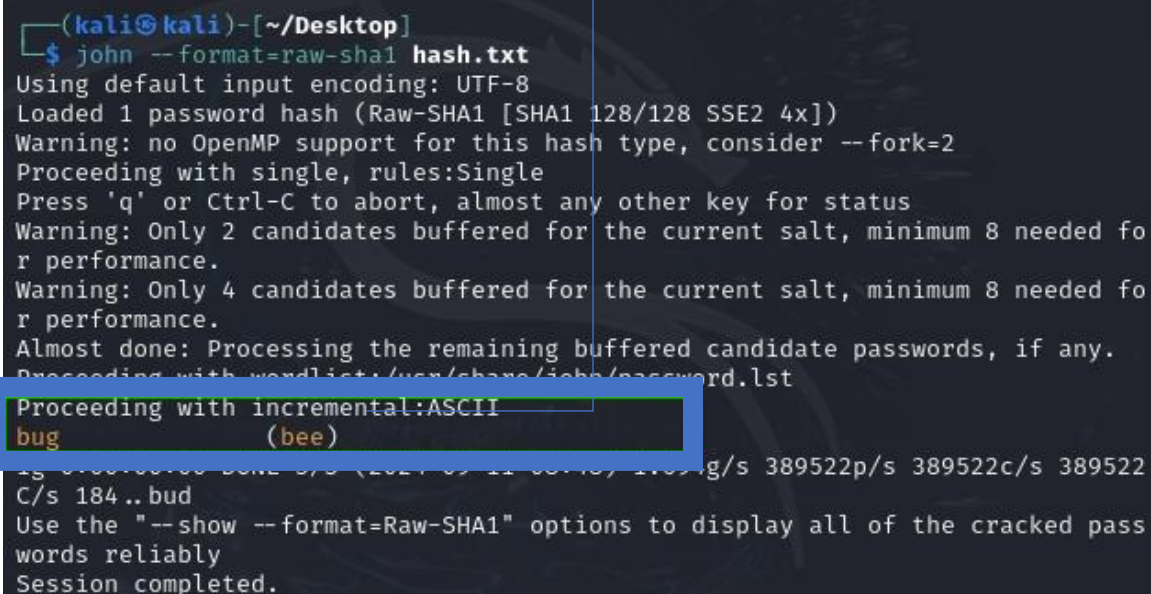


2. Tal como lo hicimos en el ejercicio de [sql injection](#) explora el formulario vulnerable. Verás un formulario con un campo para ingresar una búsqueda (generalmente llamado title o id).



3. Apoyandonos en lo aprendido en el ejercicio [sql injection](#) introduce un payload para obtener el nombre del usuario actual:

3. Revisa los resultados. Si John el Ripper crackea el hash correctamente, verás algo como:



```
(kali@kali)-[~/Desktop]
$ john --format=raw-sha1 hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 128/128 SSE2 4x])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 2 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist: /usr/share/john/password.lst
Proceeding with incremental:ASCII
bug (bee)
1g 0:00:00.00 DONE 0/0 (2021-07-11 08:48) 21071g/s 389522p/s 389522c/s 389522
C/s 184 .. bud
Use the "--show --format=Raw-SHA1" options to display all of the cracked passwords reliably
Session completed.
```