

M3,W4,D1

## INTRO & PROVA DI PING (da attaccante verso vittima)

Per lo svolgimento dell'esercizio è stato predisposto un ambiente virtuale composto da due macchine Kali Linux: una configurata come server "vittima" (ospitante la DVWA) e una come macchina "attaccante". Dopo aver configurato le interfacce di rete assegnando indirizzi IP statici per garantire la comunicazione interna, è stata verificata la connettività tra i due sistemi. Come evidenziato dallo screenshot, il comando ping lanciato dalla macchina attaccante verso l'indirizzo della vittima (192.168.50.100) ha dato esito positivo, confermando che la rete è operativa e che le due macchine possono vedersi.

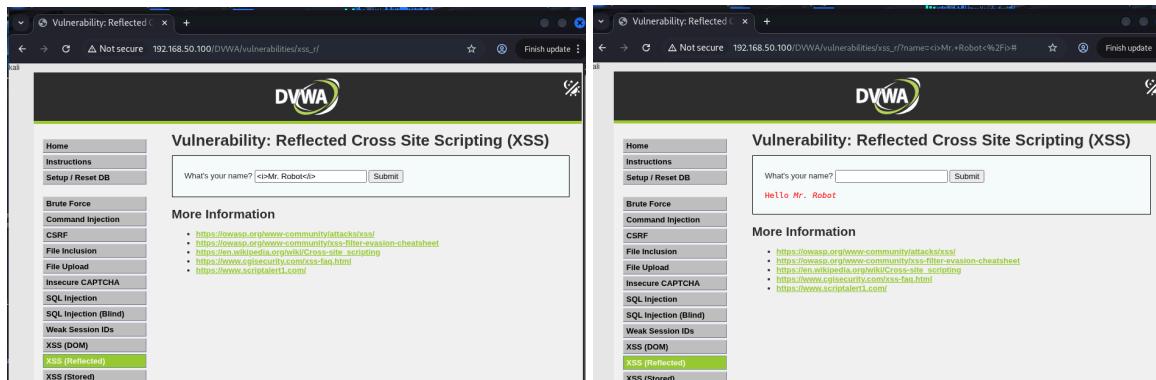


```
File Actions Edit View Help
File System
File Terminal
kali@kali: ~
valid_lft 86372sec preferred_lft 14372sec
inet6 fe80::2e6edab:5565:1407/164 scope link
    valid_lft forever preferred_lft forever
3: eth1: flags=4163<UP,BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel state UP
    group default qlen 1000
    link/ether 00:0B:27:AF:0B:06 brd FF:FF:FF:FF:FF:FF
        inet6 fe80::A00B:27ff:feaF:b0B6/64 scope link proto kernel_ll
            valid_lft forever preferred_lft forever
ping 192.168.50.100
PING 192.168.50.100(192.168.50.100) 56(84) bytes of data.
64 bytes from 192.168.50.100: icmp_seq=1 ttl=64 time=0.384 ms
64 bytes from 192.168.50.100: icmp_seq=2 ttl=64 time=0.768 ms
...
192.168.50.100 ping statistics
2 packets transmitted, 2 received, 0% packet loss, time 1034ms
rtt min/avg/max/mdev = 0.304/0.536/0.768/0.232 ms
kali@kali: ~
```

## 1 XSS REFLECTED

### - Primo test XSS: Inserimento di codice HTML

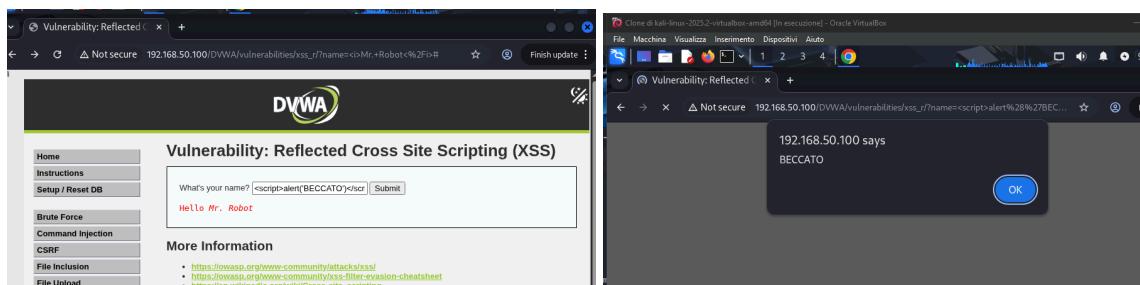
Entrati nel sito DVWA e messa la sicurezza al minimo (Low), si è iniziato a testare se il sito fosse vulnerabile agli attacchi XSS. Si è fatta una prova molto semplice inserendo il codice <i>Hacker</i> nel campo di testo. Il risultato è stato immediato: la parola è apparsa scritta in corsivo. Questo significa che il sito non controlla quello che l'utente scrive e lo esegue come se fosse codice valido.



The screenshots show the DVWA application interface. On the left, the user has entered "<i>Mr. Robot</i>" into the "What's your name?" input field. On the right, the application has reflected this input back to the user, displaying it in an italicized font as "Hello Mr. Robot". This demonstrates a reflected XSS vulnerability where user input is directly echoed back to the browser without being properly sanitized.

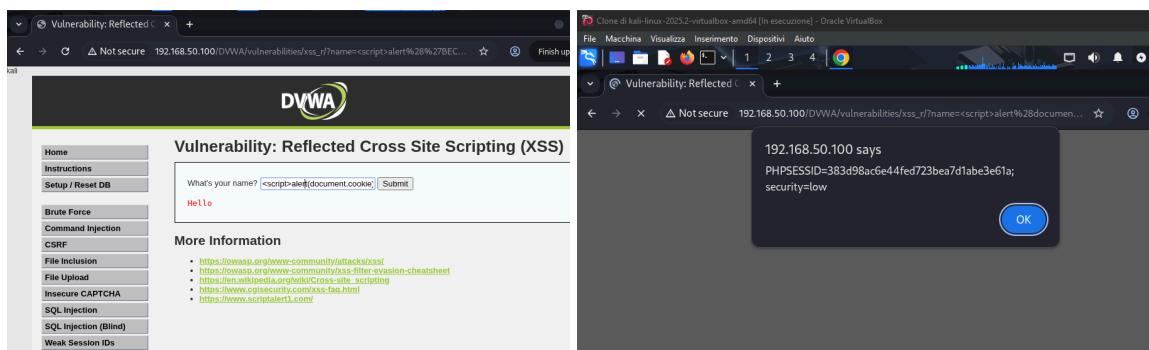
### - Secondo test XSS: Esecuzione di un avviso automatico

Visto che il codice HTML funzionava, si è provato a inserire un comando più avanzato usando il linguaggio JavaScript. Nel box è stato scritto: <script>alert('BECCATO')</script> per far sì che il messaggio venga letto come una stringa di codice. Appena inviato il comando, è comparso un pop-up di avviso al centro dello schermo con il messaggio inserito nel codice. Questo dimostra che un attaccante potrebbe far eseguire delle azioni automatiche al browser di chi visita la pagina.



### - L'attacco XSS: Furto dei cookies

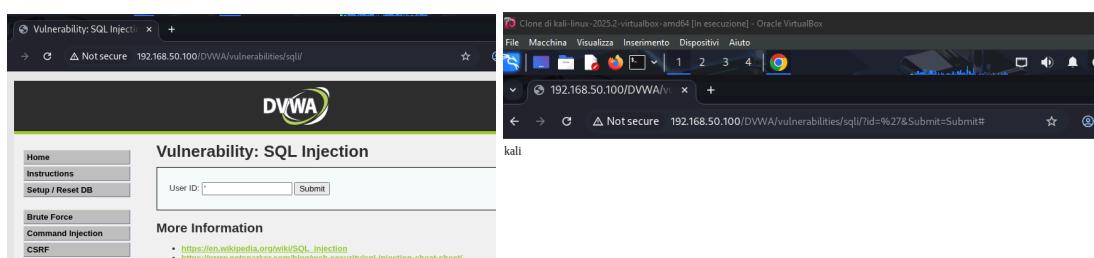
Per capire quanto può essere pericolosa questa falla, si è provato a leggere i dati riservati della sessione, i cookies. Cambiando il comando in <script>alert(document.cookie)</script>, la finestra di avviso ha mostrato un codice chiamato PHPSESSID: se un malintenzionato riuscisse a copiarlo, potrebbe entrare nell'account della vittima senza nemmeno sapere la password.



## 2 SQL INJECTION

### - SQL Injection: Verifica della vulnerabilità

Passando all'attacco al database (SQL Injection), si è voluto vedere se era possibile "rompere" la richiesta che il sito fa al server. È bastato inserire un semplice apice ' nel campo dell'ID utente. Il sito ha risposto con un errore o una pagina bianca. Questo è il segnale che il database non è protetto e va in confusione se riceve caratteri speciali non previsti.



## - Lettura della lista completa degli utenti

Capito che il database era vulnerabile, si è usato il trucco logico per farsi dare la lista di tutti gli iscritti al sito. Si è inserito il codice: '%' OR '1'='1. In pratica, si è detto al computer: "Fammi vedere l'utente OPPURE fammi vedere tutto se 1 è uguale a 1". Dato che 1 è sempre uguale a 1, il database ha obbedito e ha mostrato l'elenco completo di tutti gli utenti registrati.

The screenshot shows two browser tabs. The left tab is titled 'Vulnerability: SQL Injectio...' and the right tab is titled 'Not secure 192.168.50.100/DVWA/vulnerabilities/sql/?id=%25%27+OR+%271%27%3D...'. Both tabs show the DVWA logo. The left tab displays the 'Vulnerability: SQL Injection' page with a sidebar menu and a form where 'User ID' is set to '% OR '1'='1'. The right tab shows the results of the exploit, displaying a table with columns 'User ID', 'First name', and 'Surname'. The table contains five rows of data, each representing a user from the database.

User ID	First name	Surname
ID: % OR '1'='1	admin	admin
ID: % OR '1'='1	Gordon	Brown
ID: % OR '1'='1	Hack	Me
ID: % OR '1'='1	Pablo	Picasso
ID: % OR '1'='1	Bob	Smith

## - Furto delle password dal database

L'ultimo passaggio è stato il più critico: usare il comando UNION per leggere le password segrete. Scrivendo 1' UNION SELECT user, password FROM users#, si sono unite due richieste in una. Il risultato si vede nello screen: sotto al nome dell'amministratore, sono apparsi i nomi degli altri utenti con accanto una stringa lunga e incomprensibile. Quelle stringhe sono le password criptate (hash), che sono state estratte con successo dal database.

The screenshot shows the DVWA SQL Injection page with the 'SQL Injection' option selected in the sidebar. The main area has a form with 'User ID' set to '1' UNION SELECT user, password FROM users#. Below the form, several hash strings are listed, each corresponding to a user's password. These hashes are the result of the UNION query that combined the database's user table with its own definition.

User ID	Hashed Password
ID: 1' UNION SELECT user, password FROM users#	e99a18c428cb38d5f2e08853678922e03
ID: 1' UNION SELECT user, password FROM users#	8d3553d75ae2c396d7e6d4fcc69216b
ID: 1' UNION SELECT user, password FROM users#	5f4dcc3b5aa765d61d8327deb882cf99