

Vulnerability Name:	Weak Session IDs
Affected Vendor:	DVWA
Affected Product Name:	http://dvwa/vulnerabilities/weak_id/
Product Official Website URL:	http://dvwa/login.php
Affected Component:	Weak Session IDs

Description: - A weak session ID refers to a session identifier that is predictable, guessable, or easily brute-forced, making it vulnerable to session hijacking, fixation, or impersonation attacks. Session IDs are meant to uniquely identify a user's session in a web application, but if they are weak, attackers can exploit them to gain unauthorized access.

Root Cause: - Insufficient entropy, insecure random number generation, predictable patterns, lack of session id regeneration, Inadequate Session Management.

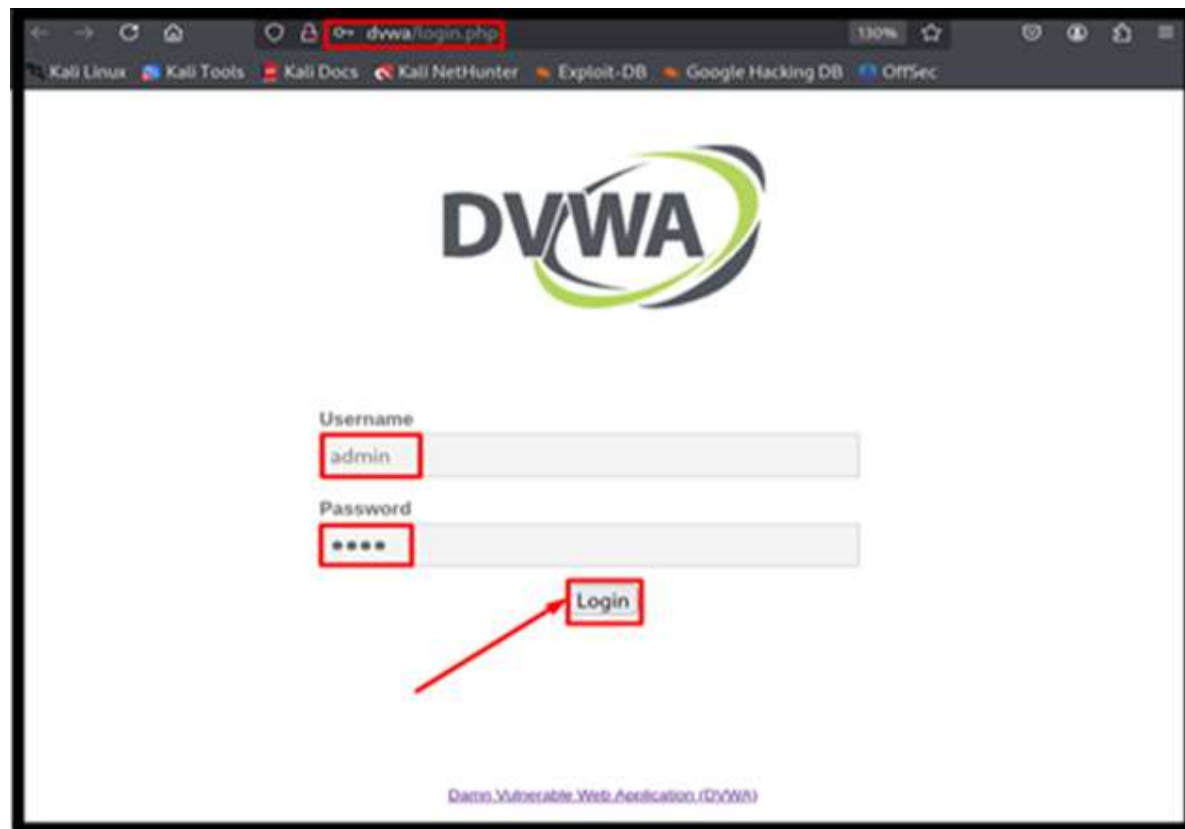
Impact: - Weak session IDs can have severe security consequences, as they allow attackers to hijack user sessions, impersonate users, and gain unauthorized access to sensitive information or functionalities within an application.

Mitigation: - Use cryptographically secure random session IDs (e.g., generated using SHA-256 or UUIDs). ensure session IDs are long and complex (128 bits or more). regenerate session IDs upon authentication and privilege escalation. Store session IDs in secure, HTTP-only cookies instead of URLs. Implement session expiration and inactivity timeouts.

Remediation: - **To Remediation of Weak Session id Use Secure Random Number Generators:** Generate session IDs using cryptographically secure random number generators. **Increase Entropy:** Increase the entropy of session IDs to make them less predictable. **Regenerate Session IDs:** Regenerate session IDs after a certain period or event, such as after logout. **Implement Secure Session Management:** Implement secure session management practices, such as invalidating sessions after logout. **use HTTPS:** Use HTTPS to encrypt session IDs in transit. **use Secure Cookie Flags:** Use secure cookie flags, such as "Secure" and "HttpOnly", to protect session IDs.

Proof Of Concept

Step: - First navigate to <http://dvwa/login.php> and login with username and Password.

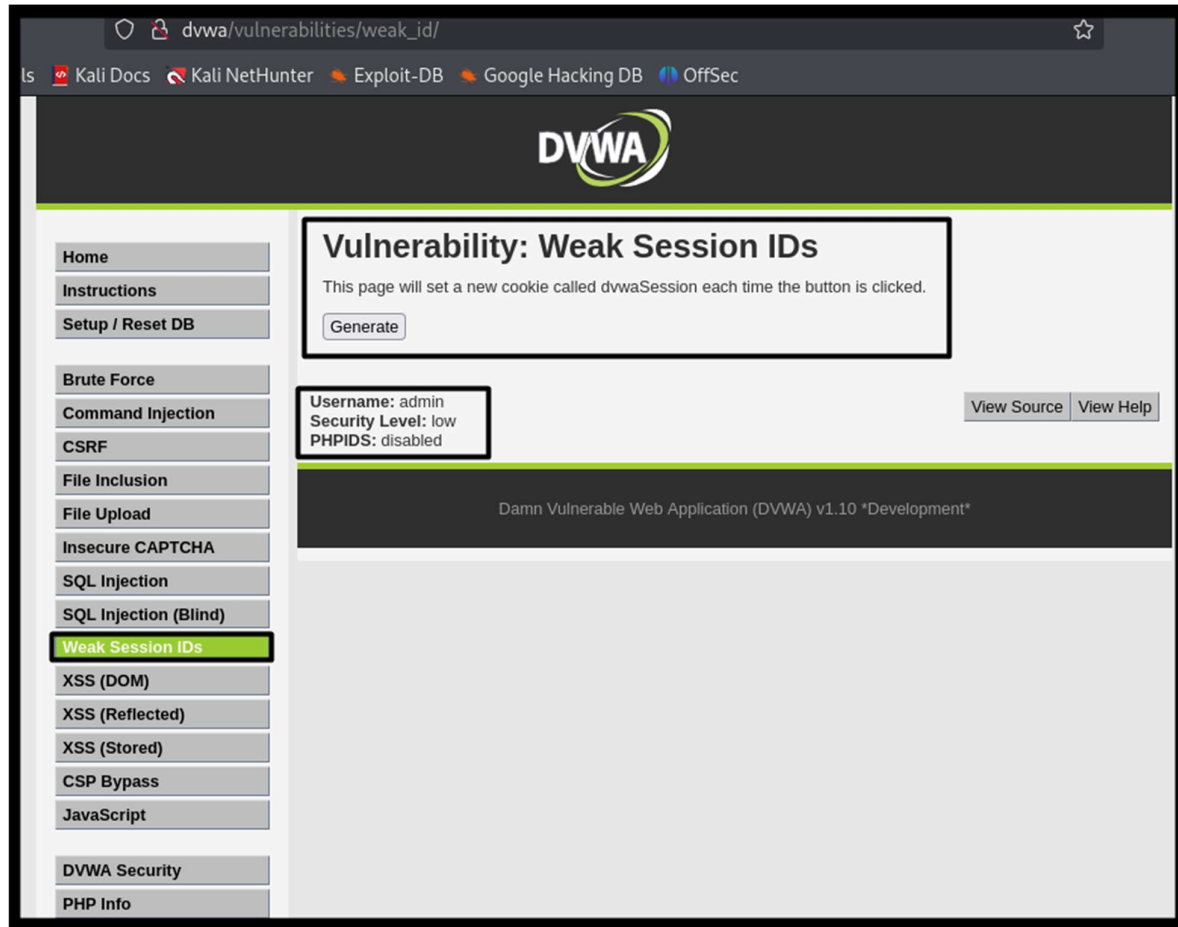


Security Level :- Low

As we Know, we will first view the source code.

```
Weak Session IDs Source  
vulnerabilities/weak_id/source/low.php  
  
<?php  
$html = "";  
  
if ($_SERVER['REQUEST_METHOD'] == "POST") {  
    if (!isset ($_SESSION['last_session_id'])) {  
        $_SESSION['last_session_id'] = 0;  
    }  
    $_SESSION['last_session_id']++;  
    $cookie_value = $_SESSION['last_session_id'];  
    setcookie("dvwaSession", $cookie_value);  
}  
>
```

Step: -2 log in the home page of DVWA then click to the Weak Session ID section.



Step: -3 If we click generate we can see the dvwaSession cookie's value via Inspect > Storage > Cookies:

The screenshot shows the DVWA web application interface. The left sidebar contains navigation links: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, and SQL Injection (Blind). The main content area is titled 'Vulnerability: Weak Session IDs' and includes a description: 'This page will set a new cookie called dvwaSession each time the button is clicked.' Below this is a 'Generate' button. Further down, it displays 'Username: admin', 'Security Level: low', and 'PHPIDS: disabled'. At the bottom, it says 'Damn Vulnerable Web Application (DVWA) v1.10 *Development*'. The browser's developer tools are open to the 'Cookies' tab, showing a table of cookies for 'http://dvwa'. The 'dvwaSession' cookie has a value of '7'.

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
dvwaSes...	7	dvwa	/vulnerabilit...	Session	12	false	false	None	Wed, 05 Mar 2025 1...
PHPSES...	a7sjcghfb1tvu253...	dvwa	/	Session	35	false	false	None	Wed, 05 Mar 2025 1...
security	low	dvwa	/	Session	11	false	false	None	Wed, 05 Mar 2025 1...

Step: -4 Each time we click Generate this value increments by 1, so after clicking 5 times.

This screenshot shows the same DVWA interface as the first image, but after clicking the 'Generate' button. The 'dvwaSession' cookie value in the browser's Cookies inspector has incremented to '8'. The rest of the page content remains the same.

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
dvwaSes...	8	dvwa	/vulnerabilit...	Session	12	false	false	None	Wed, 05 Mar 2025 1...
PHPSES...	a7sjcghfb1tvu253...	dvwa	/	Session	35	false	false	None	Wed, 05 Mar 2025 1...
security	low	dvwa	/	Session	11	false	false	None	Wed, 05 Mar 2025 1...

Step: -5 Each time we click Generate this value increments by 1, so after clicking 5 times.

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface in a web browser. The address bar shows the URL `dvwa/vulnerabilities/weak_id/`. The page title is "Vulnerability: Weak Session IDs". A text box explains: "This page will set a new cookie called dvwaSession each time the button is clicked." Below this is a "Generate" button. The page also displays the username "admin", security level "low", and "PHPIDS: disabled". A footer message reads "Damn Vulnerable Web Application (DVWA) v1.10 'Development'".

The browser's developer tools are open, showing the "Cookies" tab for the domain `http://dvwa`. The following table represents the cookies shown:

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
dvwaSession	9	dvwa	/vulnerabilities/weak_id	Session	12	false	false	None	Wed, 05 Mar 2025 17:20:10 GMT
PHPSESSID	a7sjcghfb1tvu253aeihq8dt7	dvwa	/	Session	35	false	false	None	Wed, 05 Mar 2025 17:20:10 GMT
security	low	dvwa	/	Session	11	false	false	None	Wed, 05 Mar 2025 17:20:10 GMT

SECURITY LEVEL (MEDIUM)

As We know we can view the source code.

```
Weak Session IDs Source
vulnerabilities/weak_id/source/medium.php

<?php
$html = "";

if ($_SERVER['REQUEST_METHOD'] == "POST") {
    $cookie_value = time();
    setcookie("dvwaSession", $cookie_value);
}
?>
```

Step: -1 If we click Generate now the cookie looks more complicated.

The screenshot shows the DVWA web application interface. The main heading is "Vulnerability: Weak Session IDs". Below it, a message states: "This page will set a new cookie called dvwaSession each time the button is clicked." A "Generate" button is highlighted with a yellow box. To the right of the button, the current session information is displayed: "Username: admin", "Security Level: medium", and "PHPIDS: disabled". There are "View Source" and "View Help" links. At the bottom, a footer reads "Damn Vulnerable Web Application (DVWA) v1.10 *Development*".

The browser's developer tools are open, showing the "Cookies" tab for the domain "http://dvwa". The table below lists the cookies:

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
dvwaSession	1741196052	dvwa	/vulnerabilities/weak_id	Session	21	false	false	None	Wed, 05 Mar 2025 17:34:12 GMT
PHPSESSID	a7sjcghfb1tvu253aeihq8dt7	dvwa	/	Session	35	false	false	None	Wed, 05 Mar 2025 17:34:01 GMT
security	medium	dvwa	/	Session	14	false	false	None	Wed, 05 Mar 2025 17:34:01 GMT

Step: - 2 If we click a second time:

The screenshot shows the DVWA interface with the 'Vulnerability: Weak Session IDs' page. The 'Generate' button has been clicked, and the cookie table in the browser's developer tools has been updated. The 'dvwaSession' cookie value is now 1741196221, which is a variation of the previous value 1741196052, with only the last three digits changed.

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
dvwaSession	1741196221	dvwa	/vulnerabilities/weak_id	Session	21	false	false	None	Wed, 05 Mar 2025 17:37:01 GMT
PHPSESSID	a7sjcghfb1tvu253aelihq8dt7	dvwa	/	Session	35	false	false	None	Wed, 05 Mar 2025 17:37:01 GMT
security	medium	dvwa	/	Session	14	false	false	None	Wed, 05 Mar 2025 17:37:01 GMT

Step: -3 So the first cookie has the value of 1741196052 and the second one the value of 1741196221; only the last 3 digits changed. If we click a third time similar changes occur:

The screenshot shows the DVWA interface with the 'Vulnerability: Weak Session IDs' page. The 'Generate' button has been clicked a third time, and the cookie table in the browser's developer tools has been updated again. The 'dvwaSession' cookie value is now 1741196345, which is another variation of the previous value 1741196221, with only the last three digits changed.

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
dvwaSession	1741196345	dvwa	/vulnerabilities/weak_id	Session	21	false	false	None	Wed, 05 Mar 2025 17:39:05 GMT
PHPSESSID	a7sjcghfb1tvu253aelihq8dt7	dvwa	/	Session	35	false	false	None	Wed, 05 Mar 2025 17:39:05 GMT
security	medium	dvwa	/	Session	14	false	false	None	Wed, 05 Mar 2025 17:39:05 GMT

SECURITY LEVEL (HIGH)

As we know we can see the source code.

```
Weak Session IDs Source
vulnerabilities/weak_id/source/high.php

<?php
$html = "";

if ($_SERVER['REQUEST_METHOD'] == "POST") {
    if (!isset($_SESSION['last_session_id_high'])) {
        $_SESSION['last_session_id_high'] = 0;
    }
    $_SESSION['last_session_id_high']++;
    $cookie_value = md5($_SESSION['last_session_id_high']);
    setcookie("dwwaSession", $cookie_value, time()+3600, "/vulnerabilities/weak_id/", $_SERVER['HTTP_HOST'], false, false);
}

?>
```

Step: -1 Let's start by generating 2 cookies in a row and see how they look using inspect this time.

The screenshot shows the DVWA v1.10 'Weak Session IDs' page. The 'Generate' button has been clicked, resulting in two cookies being set. The browser's developer tools are open to the 'Storage' tab, showing the following cookies:

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
dwwaSession	1741196345	dwwa	/vulnerabilities/weak_id	Session	21	false	false	None	Wed, 05 Mar 2025 17:52:09 GMT
dwwaSession	c81e728d9d4c2f636f067f89cc14862c	dwwa	/vulnerabilities/weak_id/	Wed, 05 Mar 2025 18:52:09 GMT	43	false	false	None	Wed, 05 Mar 2025 17:52:09 GMT
PHPSESSID	a75jcghfb7tvu253aeihq8dt7	dwwa	/	Session	35	false	false	None	Wed, 05 Mar 2025 17:52:09 GMT
security	high	dwwa	/	Session	12	false	false	None	Wed, 05 Mar 2025 17:52:09 GMT

Step: -2 Let's start by generating 2 cookies in a row and see how they look using Inspect this time.

The screenshot shows the DVWA application interface. The 'Vulnerability: Weak Session IDs' page is active, displaying a 'Generate' button. The browser's developer tools are open, showing the 'Cookies' tab for the domain 'dvwa'. The cookies table contains the following data:

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
dwvaSession	1741196345	dvwa	/vulnerabilities/weak_id	Session	21	false	false	None	Wed, 05 Mar 2025 17:57:15 GMT
PHPSESSID	ecbcb37e4b5c26e28308d972a7a9f3	dvwa	/vulnerabilities/weak_id	Wed, 05 Mar 2025 18:57:15 GMT	43	false	false	None	Wed, 05 Mar 2025 17:57:15 GMT

Step: -3 Let's start by generating 2 cookies in a row and see how they look using Inspect this time.

The screenshot shows the DVWA application interface. The 'Vulnerability: Weak Session IDs' page is active, displaying a 'Generate' button. The browser's developer tools are open, showing the 'Cookies' tab for the domain 'dvwa'. The cookies table contains the following data:

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
dwvaSession	1741196345	dvwa	/vulnerabilities/weak_id	Session	21	false	false	None	Wed, 05 Mar 2025 17:59:56 GMT
dwvaSession	a87f1679a273e76d9181a6797542122c	dvwa	/vulnerabilities/weak_id	Wed, 05 Mar 2025 17:59:56 GMT	43	false	false	None	Wed, 05 Mar 2025 17:59:56 GMT
PHPSESSID	a73c9f1b1072530e1q8d7	dvwa	/	Session	35	false	false	None	Wed, 05 Mar 2025 17:59:56 GMT
security	high	dvwa	/	Session	12	false	false	None	Wed, 05 Mar 2025 17:59:56 GMT

Step: -4 In this Step I am Cracking the all hashes value we can see.

CrackStation

Defuse.ca · Twitter

CrackStation ▾ Password Hashing Security ▾ Defuse Security ▾

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

c81e728d9d4c2f636f067f89cc14862c
eccbc87e4b5ce2fe28308fd9f2a7baf3
a87ff679a2f3e71d9181a67b7542122c

☐ I'm not a robot reCAPTCHA
Privacy · Terms

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1_bin), QubesV3.1BackupDefaults

Hash	Type	Result
c81e728d9d4c2f636f067f89cc14862c	md5	2
eccbc87e4b5ce2fe28308fd9f2a7baf3	md5	3
a87ff679a2f3e71d9181a67b7542122c	md5	4

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

[Download CrackStation's Wordlist](#)

[How CrackStation Works](#)

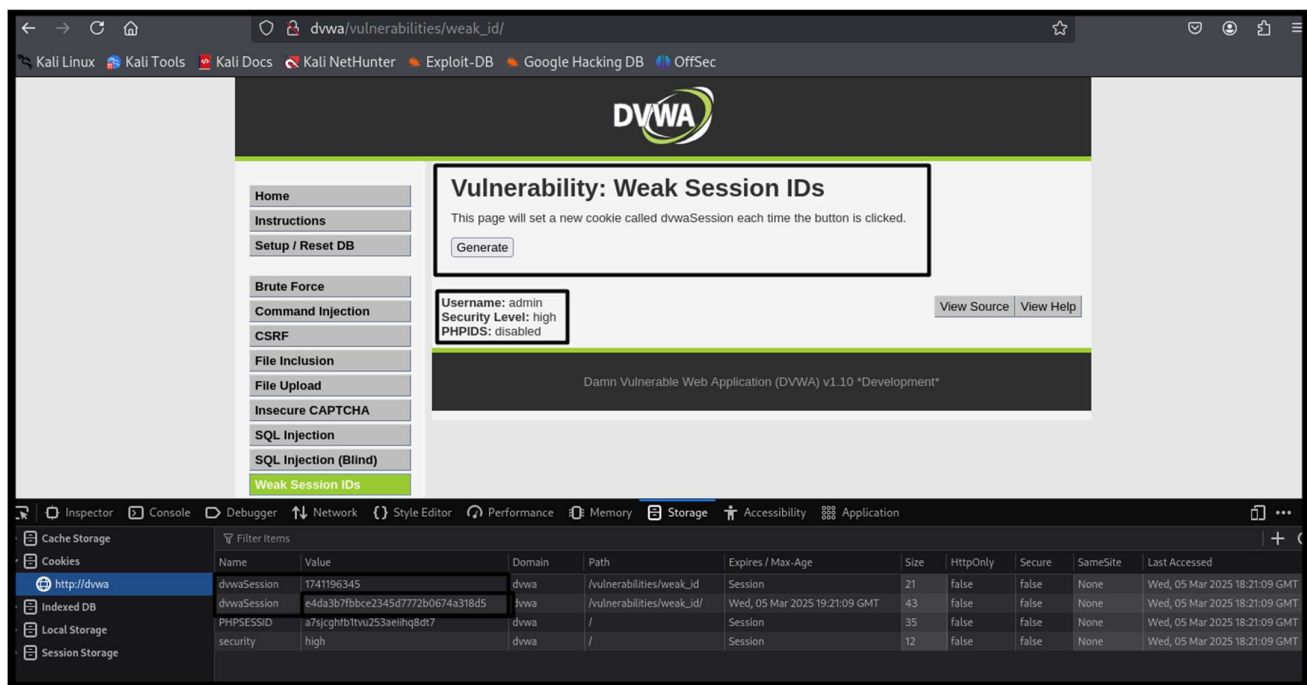
Step: -5 In this step, I am guessing a session ID of **5**, which corresponds to the hash value **e4da3b7fbbce2345d7772b0674a318d5**. Next, I check the following session ID to determine whether the hash values are similar or different.

5

Calculate Hashes Copy to clipboard (undo)

NTLM	94F23786FE827D0A3C0029DC5EB27A65
MD2	ed692e027c43c5f0f87039052e702a0b
MD4	01fa5ddd2e200f7fb50a0793f3133f60
MD5	e4da3b7fbbce2345d7772b0674a318d5
MD6-128	ccc274bde4ebb8a38b6f19a8e0c022c0
MD6-256	632a4ef13a940fec5c41cadd550dcd41d491024da
MD6-512	ad6fa2a0f8e35189d5d070559124bd6cbb1170965

Step: -6 In this step I am comparing both hashes value are same.



Step: -7 Comparison between hash value

Hash Value	Type	Result
c81e728d9d4c2f636f067f89cc14862c	md5	2
eccbc87e4b5ce2fe28308fd9f2a7baf3	md5	3
a87ff679a2f3e71d9181a67b7542122c	md5	4
e4da3b7fbce2345d7772b0674a318d5	md5	5

