

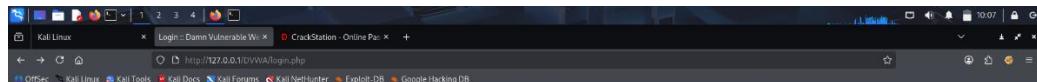
Task 3 — Web Application Security

Intern Name: Nitesh Sharma Date: 05/10/2025

Objective:-

In a controlled lab environment identify, validate and safely exploit OWASP Top 10 vulnerability to assess application security, all testing will be authorized, non-destructive, and aimed at improving security posture through documented findings and remediation recommendations.

1. SQL Injection:-



Username

Password

Damn Vulnerable Web Application (DVWA)

A screenshot of the DVWA homepage. The left sidebar contains a navigation menu with links like 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 (DOS), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript Attacks, Authorization Bypass, Open HTTP Redirect, Cryptography, API, DVWA Security, PHP Info, and About. The main content area features the DVWA logo and a welcome message: "Welcome to Damn Vulnerable Web Application! Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers learn about security issues in their applications, and to aid both students & teachers to learn about web application security in a controlled class room environment. The aim of DVWA is to practice some of the most common web vulnerabilities, with various levels of difficulty, with a simple straightforward interface." It also includes sections for General Instructions, a warning about not uploading to a hosting provider's public_html folder, and a disclaimer about responsibility for misuse.

Task 3 — Web Application Security

Intern Name: Nitesh Sharma Date: 05/10/2025

The screenshot shows the DVWA SQL Injection page. In the User ID field, the value '2 OR 1=1' is entered. The 'Submit' button is clicked, and the page displays the results of the exploit. The 'More Information' section lists several resources related to SQL injection.

```
<?php
if($_GET['id'] && $_GET['id'] == '1') {
    $id = 1;
} else {
    $id = 2;
}

if($_POST['id']) {
    $id = $_POST['id'];
}

if($id != 2) {
    $query = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysqli_query($link, $query);
    if(mysqli_num_rows($result) == 1) {
        $row = mysqli_fetch_assoc($result);
        $first_name = $row['first_name'];
        $last_name = $row['last_name'];
        echo "First name: $first_name  
Last name: $last_name";
    } else {
        echo "User not found";
    }
} else {
    $query = "SELECT first_name, last_name FROM users WHERE user_id = '2'";
    $result = mysqli_query($link, $query);
    if(mysqli_num_rows($result) == 1) {
        $row = mysqli_fetch_assoc($result);
        $first_name = $row['first_name'];
        $last_name = $row['last_name'];
        echo "First name: $first_name  
Last name: $last_name";
    } else {
        echo "User not found";
    }
}
?>
```

The screenshot shows the DVWA SQL Injection page with multiple entries in the User ID field. The entries are: '1 OR 1=1', '2 OR 1=1', '3 OR 1=1', '4 OR 1=1', '5 OR 1=1', '6 OR 1=1', '7 OR 1=1', '8 OR 1=1', and '9 OR 1=1'. The 'Submit' button is clicked, and the page displays the results of the multiple exploits. The 'More Information' section lists several resources related to SQL injection.

Task 3 — Web Application Security

Intern Name: Nitesh Sharma Date: 05/10/2025

The screenshot shows a browser window with the DVWA application open. The URL is `http://127.0.0.1/DVWA/vulnerabilities/sql_inj/?id=1#UNION+SELECT+user%2Cpassword+FROM+users%23&Submit#`. The page title is "Vulnerability: SQL Injection". On the left, there's a sidebar with various attack types: Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript Attacks, Authorization Bypass, Open HTTP Redirect, Cryptography, API, DVWA Security, PHP info, and About. The main content area shows a user input field with "User ID: 1#UNION SELECT user, password FROM users#". Below it, several database rows are listed, each with a different user name and password combination. A "More Information" section at the bottom provides links to external resources about SQL injection.

The screenshot shows a browser window with the CrackStation application open. The URL is `http://crackstation.net`. The page title is "CrackStation - Free Password Hash Cracker". It features a large text input field for pasting hashes, a reCAPTCHA verification box, and a "Crack Hashes" button. Below the input field, it says "Enter up to 20 non-salted hashes, one per line:" followed by a list of supported hash types: LM, NTLM, md2, md4, md5, md5crypt_hex, md5_half, sha1, sha224, sha256, sha384, sha512, ripemd160, whirlpool, MySQL 4.1+(blowfish_sha1), QuarkV3, BackupDefaults. A "Hash" column contains the hash value `5f46cc3b5aa76e61d8327de882cf99`, a "Type" column shows "md5", and a "Result" column shows "password". A note below the table says "Color Codes: Green Exact match, Yellow Partial match, Red Not found." At the bottom, there are links for "Download CrackStation's Wordlist" and "How CrackStation Works".

Task 3 — Web Application Security

Intern Name: **Nitesh Sharma** Date: **05/10/2025**

(Demonstrate prevention using Prepared Statement):-

```
<?php

if( isset( $_REQUEST['Submit'] ) ){
    // Get input
    $id = $_REQUEST['id'];

    switch ( $_DVWA['SQLI_DB'] ) {
        case MYSQL:
            // Check database connection
            $query = "SELECT first_name, last_name FROM users WHERE user_id = ?";

            // Prepare statement
            if ( $stmt = mysqli_prepare($GLOBALS["__mysqli_ston"], $query) ) {
                // Bind the input parameter
                mysqli_stmt_bind_param($stmt, "i", $id); // "i" means integer

                // Execute the statement
                mysqli_stmt_execute($stmt);

                // Get the result
                $result = mysqli_stmt_get_result($stmt);

                // Fetch results
                while ( $row = mysqli_fetch_assoc($result) ) {
                    $first = $row["first_name"];
                    $last = $row["last_name"];
                    echo "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {$last}</pre>";
                }

                // Close statement
                mysqli_stmt_close($stmt);
            } else {
                echo "Failed to prepare the query.";
            }

            // Close connection
            mysqli_close($GLOBALS["__mysqli_ston"]);
            break;

        case SQLITE:
            global $sqlite_db_connection;

            $query = "SELECT first_name, last_name FROM users WHERE user_id = :id";

            try{
                // Prepare statement
                $stmt = $sqlite_db_connection->prepare($query);
```

Task 3 — Web Application Security

Intern Name: **Nitesh Sharma** Date: **05/10/2025**

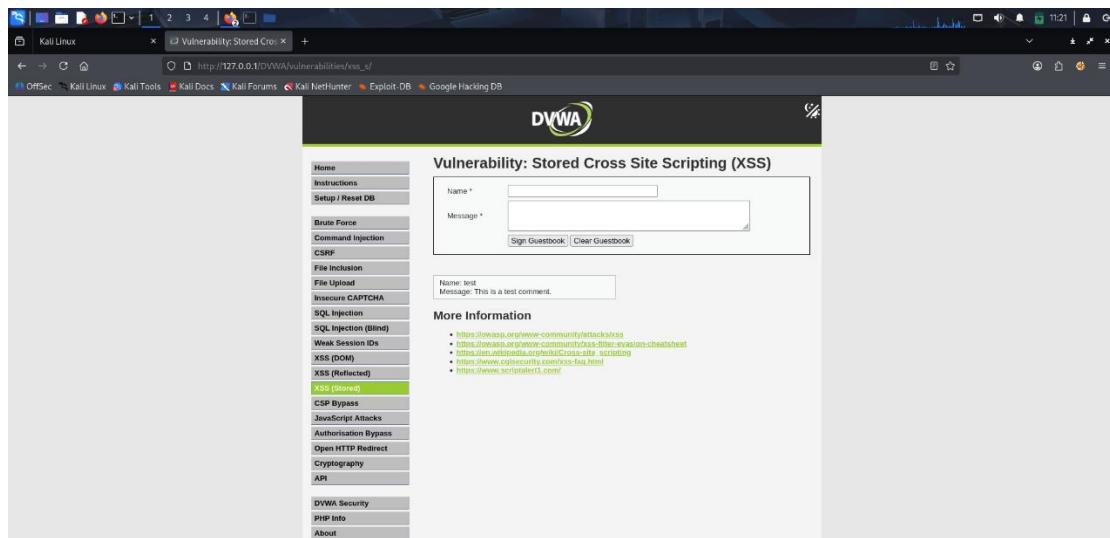
```
// Bind parameters
$stmt->bindValue(':id', $id, SQLITE3_INTEGER);

// Execute the query
$results = $stmt->execute();

// Fetch results
while ($row = $results->fetchArray()) {
    $first = $row["first_name"];
    $last = $row["last_name"];
    echo "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {$last}</pre>";
}

} catch (Exception $e) {
    echo 'Caught exception: ' . $e->getMessage();
}
break;
}
}
?>
```

2. Cross-Site Scripting (XSS):-



Task 3 — Web Application Security

Intern Name: Nitesh Sharma Date: 05/10/2025

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The main page title is "Vulnerability: Stored Cross Site Scripting (XSS)". On the left, there's a sidebar with various exploit categories like Brute Force, Command Injection, CSRF, etc. The "XSS (Stored)" option is selected. In the main content area, there are two input fields: "Name" and "Message". The "Name" field contains "test" and the "Message" field contains "Message: This is a test comment.". Below these fields are two buttons: "Sign Guestbook" and "Clear Guestbook". To the right of the form, there's a section titled "More Information" with several links related to XSS attacks.

This screenshot shows the same DVWA Stored XSS page after an exploit has been submitted. A modal dialog box appears in the center of the screen with the text "Yaaaaahhh" and an "OK" button. The rest of the page remains largely unchanged, with the exploit category still selected in the sidebar.

This screenshot shows the DVWA Stored XSS page again, but this time the exploit message is more complex, containing a script tag. A modal dialog box displays the message "Attacker is here" and includes a checkbox labeled "Don't allow 127.0.0.1 to prompt you again". The "OK" button is visible at the bottom of the dialog. The rest of the page, including the sidebar and other content, remains the same.

Task 3 — Web Application Security

Intern Name: Nitesh Sharma Date: 05/10/2025

The screenshot shows a browser window titled "Vulnerability: Stored Cross Site Scripting (XSS)". The URL is http://127.0.0.1/DVWA/vulnerabilities/xss_s/. The page displays a form with a "name" field containing the value "<script>alert('Welcome')</script>". A modal dialog box shows the output: "Welcome" with an "OK" button. Below the form, there is a "More Information" section with several links related to XSS attacks.

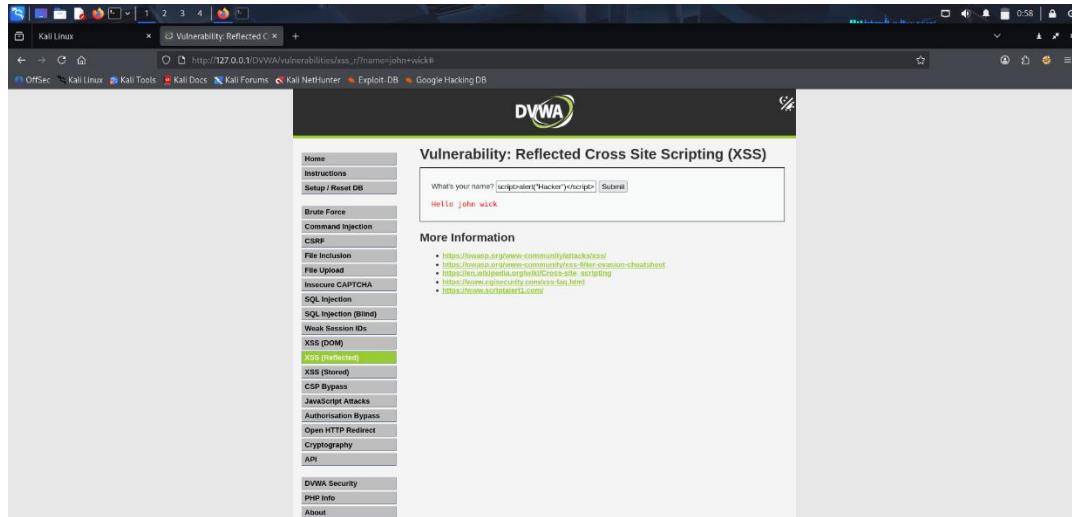
This screenshot is similar to the one above, showing the same stored XSS attack on the DVWA "Vulnerability: Stored Cross Site Scripting (XSS)" page. The modal dialog now displays the injected script as "PHPSSESSID=8fc5a3c08c024d937fd0320ad63; security=medium". The "More Information" section contains the same links about XSS attacks.

Reflected XSS using query parameters:-

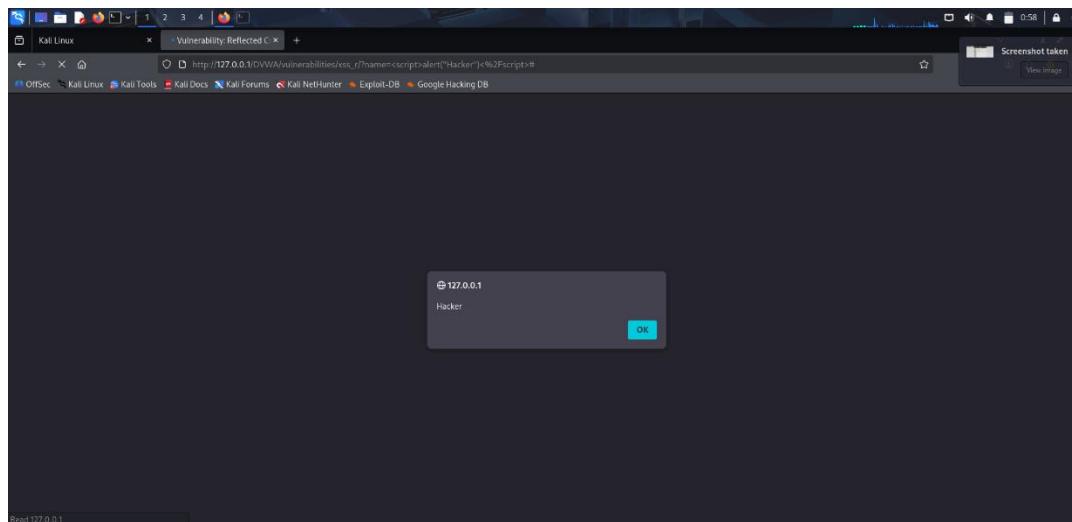
The screenshot shows a browser window titled "Vulnerability: Reflected Cross Site Scripting (XSS)". The URL is http://127.0.0.1/DVWA/vulnerabilities/xss_r/. The page has a form with a "What's your name?" input field containing "<script>alert('Welcome')</script>". A "Submit" button is next to the input field. Below the form, there is a "More Information" section with links related to reflected XSS attacks.

Task 3 — Web Application Security

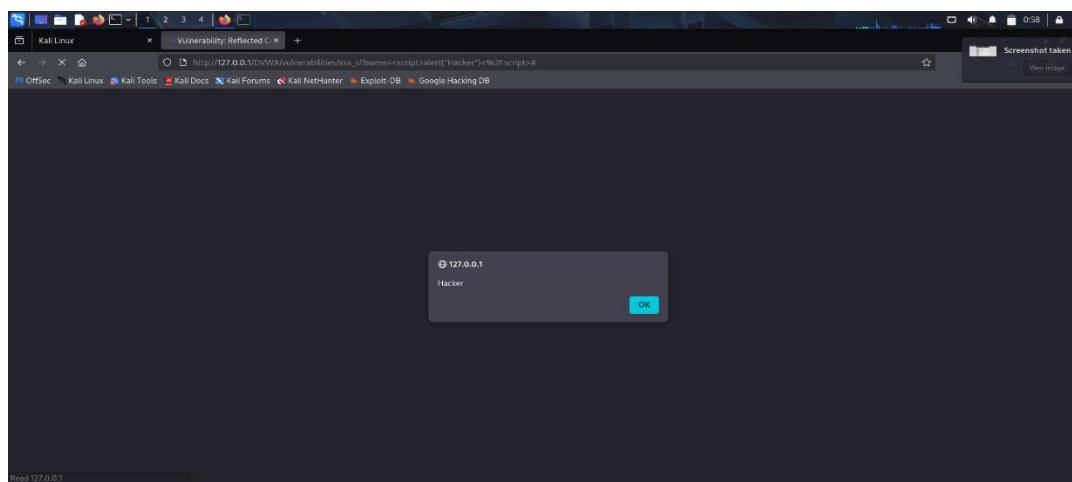
Intern Name: Nitesh Sharma Date: 05/10/2025



A screenshot of a Firefox browser window on a Kali Linux desktop. The address bar shows the URL:



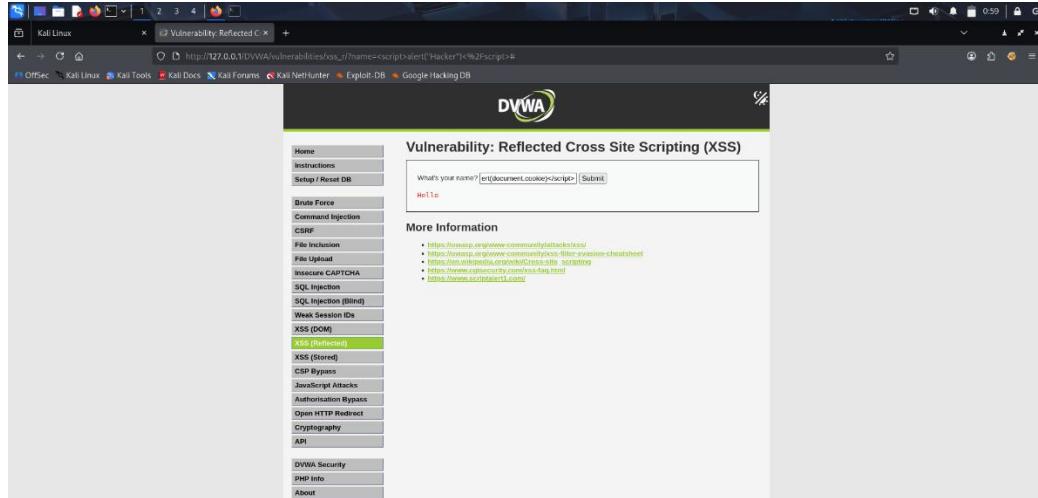
A screenshot of a Firefox browser window on a Kali Linux desktop. The address bar shows the URL:



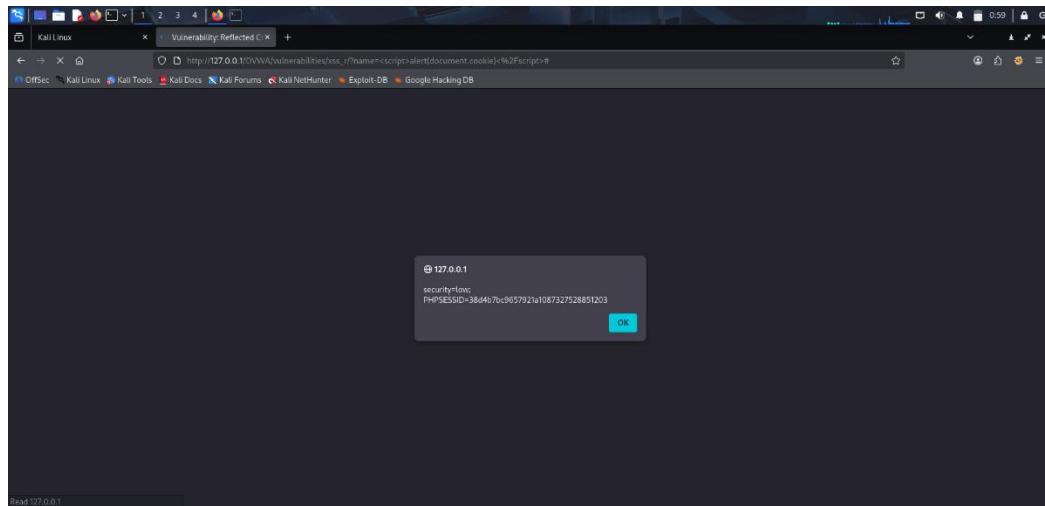
A screenshot of a Firefox browser window on a Kali Linux desktop. The address bar shows the URL:

Task 3 — Web Application Security

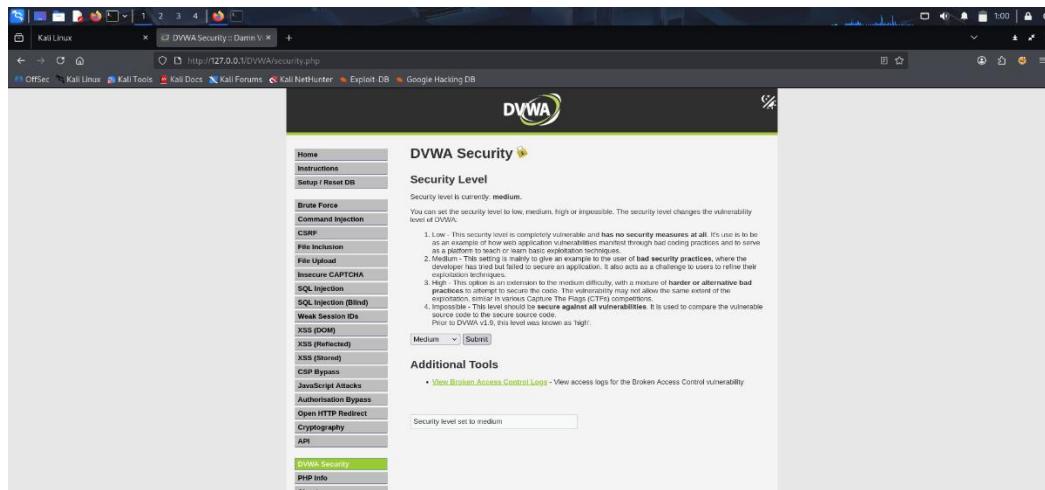
Intern Name: Nitesh Sharma Date: 05/10/2025



A screenshot of a Firefox browser window on Kali Linux. The URL is http://127.0.0.1/DVWA/vulnerabilities/xss_r/?name=csrf%3Cscript%3Ealert%28%27Hacker%27%2Fscript%29. The DVWA logo is at the top. The main content shows "Vulnerability: Reflected Cross Site Scripting (XSS)". A text input field contains "What's your name? Hellé". Below it is a "Submit" button. To the right, there is a "More Information" section with several links related to XSS attacks.



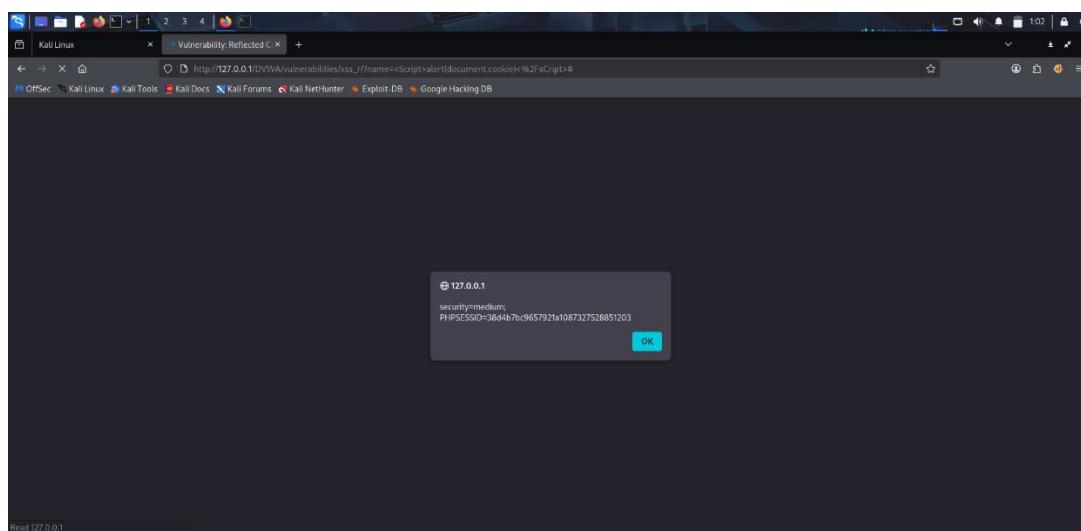
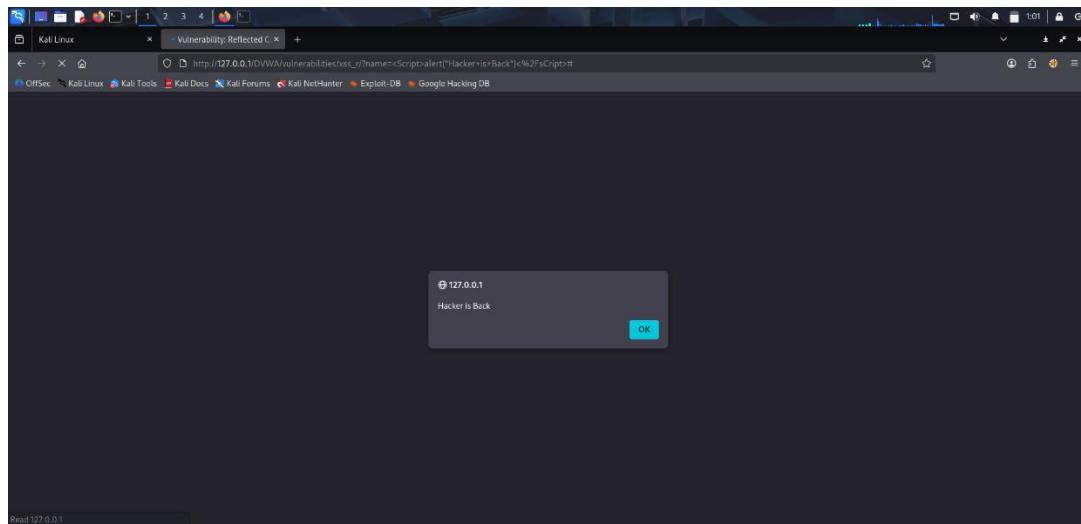
A screenshot of a Firefox browser window on Kali Linux. The URL is http://127.0.0.1/DVWA/vulnerabilities/xss_r/?name=csrf%3Cscript%3Ealert%28document.cookie%29%2Fscript%29. A JavaScript alert dialog box is displayed with the message "securityInfo: PnFSESSID=36d467bc9657921e1087327528851203". A "OK" button is visible at the bottom right of the dialog.



A screenshot of a Firefox browser window on Kali Linux. The URL is <http://127.0.0.1/DVWA/security.php>. The DVWA logo is at the top. The main content shows "DVWA Security". Under "Security Level", it says "Security level is currently: medium." There is a note about security levels: "I. Low - This security level is completely vulnerable and has no security measures at all. Its use is to be as close as possible to how web applications vulnerabilities manifest through bad coding practices and serve as a platform to teach or learn basic exploitation techniques." It also discusses "Medium" and "Impossible" levels. At the bottom, there is a "Submit" button and a note about the history of the security level: "Prior to DVWA v3.0, this level was known as 'tag'."

Task 3 — Web Application Security

Intern Name: **Nitesh Sharma** Date: **05/10/2025**



Task 3 — Web Application Security

Intern Name: Nitesh Sharma Date: 05/10/2025

The screenshot shows the DVWA Security interface. On the left, a sidebar lists various attack types: 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 Attacks, Authorization Bypass, Open HTTP Redirect, Cryptography, API, DVWA Security (highlighted in green), PHP Info, and About. The main content area displays the DVWA logo and the title "DVWA Security". It says "Security level is currently: high". Below this, there is a detailed description of the security level settings:

- Low - This security level is completely vulnerable and has no security measures at all. It's used to be an example of how web application vulnerabilities manifest through bad coding practices and to serve as a starting point for learning how to identify them.
- Medium - This setting is trying to give an example to the user of best security practices, where the developer has tried to implement some basic security measures, but still has room to improve further their exploitation techniques.
- High - This option is an extension to the medium difficulty, with a mixture of harder or alternative bad practices to challenge the user to secure the code. The vulnerability may not allow the same extent of manipulation, similar to various Capture The Flags (CTFs) competitions.

A note states: "Important: Security level against all vulnerabilities. It is used to compare the vulnerable source code to the secure source code." A dropdown menu shows "High" selected. A button labeled "Submit" is present. At the bottom, it says "Security level set to high".

The screenshot shows the DVWA Vulnerability: Reflected Cross Site Scripting (XSS) page. The sidebar and layout are identical to the previous screenshot. The main content area shows the DVWA logo and the title "Vulnerability: Reflected Cross Site Scripting (XSS)". There is a form with a text input field containing "What's your name? []" and a "Submit" button. Below the form, there is a "More Information" section with several links:

- <https://owasp.org/www-community/attacks/xss/>
- <https://owasp.org/www-community/kids-filter-evasion-cheatsheet>
- [https://www.owasp.org/index.php/XSS_\(Cross_Site_Scripting\)_Prevention_CheatSheet](https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_CheatSheet)
- https://www.owasp.org/www-community/vulnerabilities/Reflected_XSS

The screenshot shows the DVWA Vulnerability: Reflected Cross Site Scripting (XSS) page after an exploit has been submitted. The sidebar and layout are identical. The main content area shows the DVWA logo and the title "Vulnerability: Reflected Cross Site Scripting (XSS)". The exploit input field now contains "What's your name? []" with "Hello" typed into it. A modal dialog box appears in the center of the screen with the following content:

- IP: 127.0.0.1
- Security level: High
- PHPSESSID=38d4fb7b-9e57921a1087327528651203

A blue "OK" button is at the bottom right of the modal. The status bar at the bottom of the browser window shows "Viewed 137/0.0.1".

Task 3 — Web Application Security

Intern Name: Nitesh Sharma Date: 05/10/2025

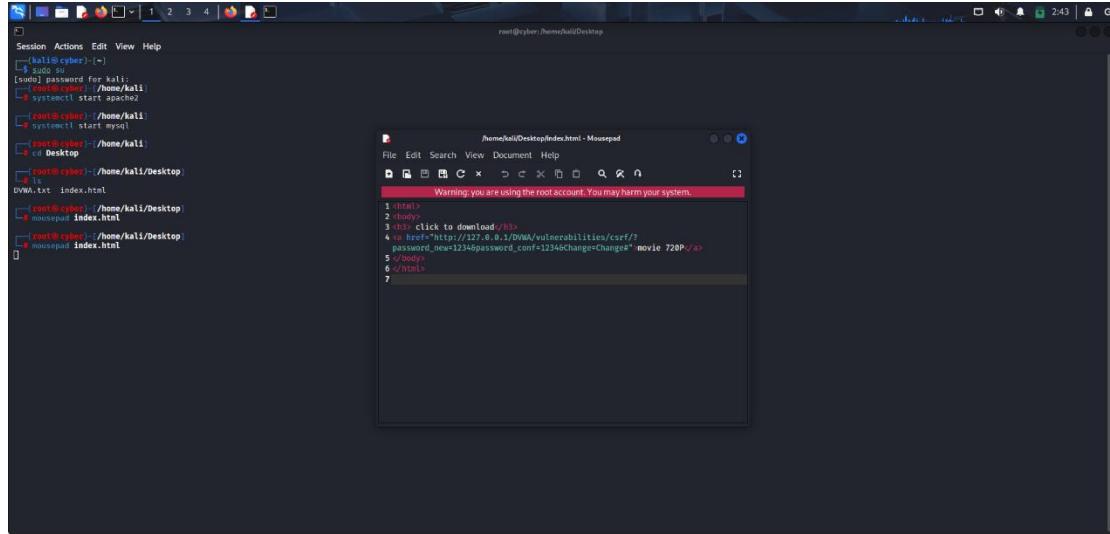
3. Cross-Site Request Forgery (CSRF):-

The screenshot shows the DVWA index page. The URL is <http://127.0.0.1/DVWA/index.php>. The left sidebar menu includes options like SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript Attacks, Authorization Bypass, Open HTTP Redirect, Cryptography, API, DVWA Security, PHP Info, About, and Logout. A warning message at the top states: "Please note, there are both documented and undocumented vulnerabilities with this software. This is intentional. You are encouraged to try and discover as many issues as possible. There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue." A "WARNING!" section follows, cautioning users about the nature of the application and its use. Below these are sections for "More Training Resources" and "More Information". A status bar at the bottom indicates "You have logged in as 'admin'". The footer says "Damn Vulnerable Web Application (DVWA)".

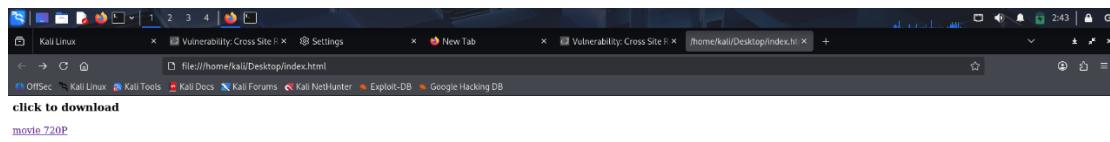
The screenshot shows the DVWA CSRF attack page. The URL is http://127.0.0.1/DVWA/vulnerabilities/csrf/?password_nevr1234&password_conf=1234&Change=Change. The left sidebar menu is identical to the previous screenshot. The main content area is titled "Vulnerability: Cross Site Request Forgery (CSRF)". It contains a form for changing the admin password, with fields for "New password" and "Confirm new password", both set to "1234". A "Change" button is present. A red error message "Password Changed." is displayed below the form. A note at the bottom left says: "Note: Browsers are starting to default to setting the [SameSite cookie](#) flag to Lax, and in doing so are killing off some types of CSRF attacks. When they have completed their mission, this lab will not work as originally expected." An "Announcements" section lists browser support: "• Chromium • Edge • Firefox". A "More Information" section provides links: "<https://owasp.org/www-community/attacks/carf>", "<https://www.cgisecurity.com/csrf-faq.html>", and "https://en.wikipedia.org/wiki/Cross-site_request_forgery".

Task 3 — Web Application Security

Intern Name: Nitesh Sharma Date: 05/10/2025



```
root@cyber:~# rootkit-check
[siude] password for kali:
[+] /var/www/html/index.html
[+] /etc/init.d/apache2
[+] /etc/init.d/mysql
[+] /etc/init.d/kali
[+] /etc/init.d/mysqld
[+] /etc/init.d/Desktop
[+] /etc/init.d/index.html
[+] /etc/init.d/cyber
[+] /etc/init.d/Desktop
[+] /etc/init.d/index.html
[+] /etc/init.d/cyber
[+] /etc/init.d/Desktop
[+] /etc/init.d/index.html
root@cyber:~# cd Desktop
root@cyber:~/Desktop# ls
DVWA.txt  index.html
root@cyber:~/Desktop# mousepad index.html
root@cyber:~/Desktop# mousepad index.html
```



Task 3 — Web Application Security

Intern Name: Nitesh Sharma Date: 05/10/2025

The screenshot shows a Firefox browser window on a Kali Linux desktop. The address bar shows the URL <http://127.0.0.1/DVWA/login.php>. The DVWA logo is at the top. A login form has 'Username' set to 'admin' and 'Password' set to '123'. A 'Login' button is present. Below the form, a message says 'You have logged out.'

The screenshot shows a Firefox browser window on a Kali Linux desktop. The address bar shows the URL <http://127.0.0.1/DVWA/index.php>. A modal dialog box titled 'Save password for http://127.0.0.17' is open, asking for a 'Username' (admin) and 'Password' (123), with a 'Save' button. The main DVWA page is visible in the background, showing the title 'Welcome :: Damn Vulnerable Web Application!' and a sidebar with various attack modules listed.

4. File inclusion attack:-

The screenshot shows a Firefox browser window on a Kali Linux desktop. The address bar shows the URL http://127.0.0.1/DVWA/vulnerabilities/file_inclusion/?page=include.php. The DVWA logo is at the top. The main content area is titled 'Vulnerability: File Inclusion'. It explains the PHP include function and its risks. A text input field contains the value 'file1.php'. Below it, a 'More Information' section lists several resources related to file inclusion vulnerabilities.

Task 3 — Web Application Security

Intern Name: Nitesh Sharma Date: 05/10/2025

The image consists of three vertically stacked screenshots of a web browser displaying the DVWA application on a Kali Linux desktop. The browser tabs are labeled 'Kali Linux', 'DVWA Security - Damn V. X', and 'Vulnerability: File Inclusion'. The DVWA interface has a sidebar with various attack types: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion (highlighted in green), File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript Attacks, Authorisation Bypass, Open HTTP Redirect, Cryptography, and API. The main content area shows the 'File Inclusion' lab details. In the top screenshot, the security level is set to 'Low'. In the middle screenshot, the 'File inclusion' section is expanded, showing PHP code examples for include, include_once, and require. In the bottom screenshot, the browser's address bar shows the URL 'http://127.0.0.1/DVWA/Vulnerabilities/vulnerabilities.php?page=http://google.com', indicating a successful exploit where the page content was injected.

Task 3 — Web Application Security

Intern Name: Nitesh Sharma Date: 05/10/2025

The image consists of three vertically stacked screenshots of the DVWA (Damn Vulnerable Web Application) interface, specifically demonstrating a File Inclusion vulnerability.

Screenshot 1: A browser window showing the DVWA homepage. The URL is `http://127.0.0.1/DVWA/vulnerabilities/fi?page=../../../../etc/passwd`. The page content displays the contents of the `/etc/passwd` file, which includes entries like `root:x:0:0:root:/root:/bin/bash` and `admin:x:100:100:admin:/var/www/html:/bin/bash`.

Screenshot 2: A browser window showing the DVWA Security level configuration page. The URL is `http://127.0.0.1/DVWA/security.php`. The security level is set to "Medium". The page contains a detailed explanation of security levels (Low, Medium, High, Impossible) and how they affect the application's behavior.

Screenshot 3: A browser window showing the source code of a PHP file. The URL is `http://127.0.0.1/DVWA/vulnerabilities/fi/source/medium.php`. The code includes a section for handling file inclusion:

```
<?php  
// The page we wish to display  
$file = $_GET['page'];  
  
// Input validation  
$file = str_replace(array( "http://", "https://", "\r", "\n", "\t", "\r\n" ), "", $file );  
$file = str_replace(array( "..", "./", "\\", "\\" ), "", $file );  
?>
```

Task 3 — Web Application Security

Intern Name: Nitesh Sharma Date: 05/10/2025

The image consists of three vertically stacked screenshots of a Kali Linux desktop environment. Each screenshot shows a Firefox browser window displaying the DVWA (Damn Vulnerable Web Application) interface.

- Screenshot 1:** Shows the DVWA main menu with "File Inclusion" selected. The URL is `http://127.0.0.1/DVWA/vulnerabilities/file_inclusion/?page=http://ip.google.com`. The DVWA logo is at the top right.
- Screenshot 2:** Shows the DVWA "Security Level" page. The "Security Level" dropdown is set to "High". Below it, there's a note about security levels and a list of additional tools: "View Broken Access Control Logs", "View access logs for the Broken Access Control vulnerability", "Authorization Bypass", "Open HTTP Redirect", "Cryptography", and "API".
- Screenshot 3:** Shows the DVWA "File Inclusion" page with the error message "ERROR: File not found!" displayed. The URL is `http://127.0.0.1/DVWA/vulnerabilities/file_inclusion/?page=http://ip.google.com`.

Task 3 — Web Application Security

Intern Name: Nitesh Sharma Date: 05/10/2025

The image consists of three vertically stacked screenshots of a web application interface. Each screenshot shows a browser window with multiple tabs open, all displaying the same URL: http://172.0.0.1/DVWA/vulnerabilities/file_inclusion/?id=1&file=1. The interface includes a navigation menu on the left with options like Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion (which is highlighted in green), File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), and CSP Bypass. The main content area displays a large amount of error and log text from the application's internal logs, indicating various security vulnerabilities and errors.

5. Burp-Suite Advanced:-

A screenshot of the Burp Suite Community Edition interface. On the left, there is a terminal window showing a Linux shell session with commands like 'systemctl start' and 'curl'. The main window is titled '1. Use passive crawl from Proxy(all traffic)' and shows a 'Summary' tab with a table titled 'Items added to site map'. The table has columns for Host, Method, URL, Status code, and MIME type. Below the table, it says 'No items to show' and 'Items found in the crawl will display here.' On the right side of the interface, there are sections for 'Task configuration', 'Task progress', and 'Task log'.

Task 3 — Web Application Security

Intern Name: Nitesh Sharma Date: 05/10/2025

The screenshot shows a dual-monitor setup. The top monitor displays a Firefox browser window for the DVWA login page at `http://127.0.0.1/DVWA/login.php`. The DVWA logo is at the top, followed by a form with 'Username' set to 'admin' and 'Password' set to '1'. Below the form, an error message says 'CSRF token is incorrect'. To the right of the browser is a terminal window showing the command `curl http://127.0.0.1/DVWA/login.php`. The bottom monitor displays the Burp Suite Community Edition interface. The 'Proxy' tab is selected, showing a captured POST request to `http://127.0.0.1/DVWA/login.php`. The 'Request' pane shows the raw HTTP request with various headers, including 'Content-Type: application/x-www-form-urlencoded'. The 'Inspector' pane on the right shows the request attributes, query parameters, body parameters, cookies, and headers.

Task 3 — Web Application Security

Intern Name: Nitesh Sharma Date: 05/10/2025

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A POST request is captured from the '127.0.0.1/DVWA/login.php' URL. The request payload includes a session cookie 'user_token' with the value '2bcce6df991d969388b0f02e7777d815'. The 'Inspector' tool is open, showing the decoded form data where the session cookie is visible.

Request

```
Pretty Raw Hex
5 sec-ch-ua: "Chromium";v="141", "NotA_Brand";v="8"
6 sec-ch-ua-mobile: ?0
7 sec-ch-ua-platform: "Linux"
8 Accept-Language: en;q=0.9
9 Origin: http://127.0.0.1/DVWA/
10 Content-Type: application/x-www-form-urlencoded
11 Upgrade-Insecure-Requests: 1
12 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36
13 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Prefetch-Document
18 Referer: http://127.0.0.1/DVWA/login.php
19 Accept-Encoding: gzip, deflate, br
20 Connection: keep-alive
21 Content-Type: application/x-www-form-urlencoded
22
23 username=admin&password=rانLogin=Login&user_token=2bcce6df991d969388b0f02e7777d815
```

Inspector

Selected text: `username=admin&password=rانLogin=Login&user_token=2bcce6df991d969388b0f02e7777d815`

Decoded from: URL Encoding

Decoded value: `username=admin&password=rانLogin=Login&user_token=2bcce6df991d969388b0f02e7777d815`

Request attributes: `Content-Type: application/x-www-form-urlencoded`

Request memory: 119.0MB

Welcome :: Damn Vulnerable Web Application

General Instructions

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and knowledge in a legal environment, help web developers better understand how to protect their web applications, and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to practice some of the most common web vulnerabilities, with various levels of difficulty, with a simple straightforward interface.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! Do not upload it to your hosting provider's public html folder or any Internet facing servers, as they will be compromised. It is recommended using a virtual machine (such as VirtualBox or VMWare), which is set to NAT networking mode. Inside a guest machine, you can download and install XAMPP for the web server and database.

Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA, it is not our responsibility it is the responsibility of the person who uploaded and installed it.

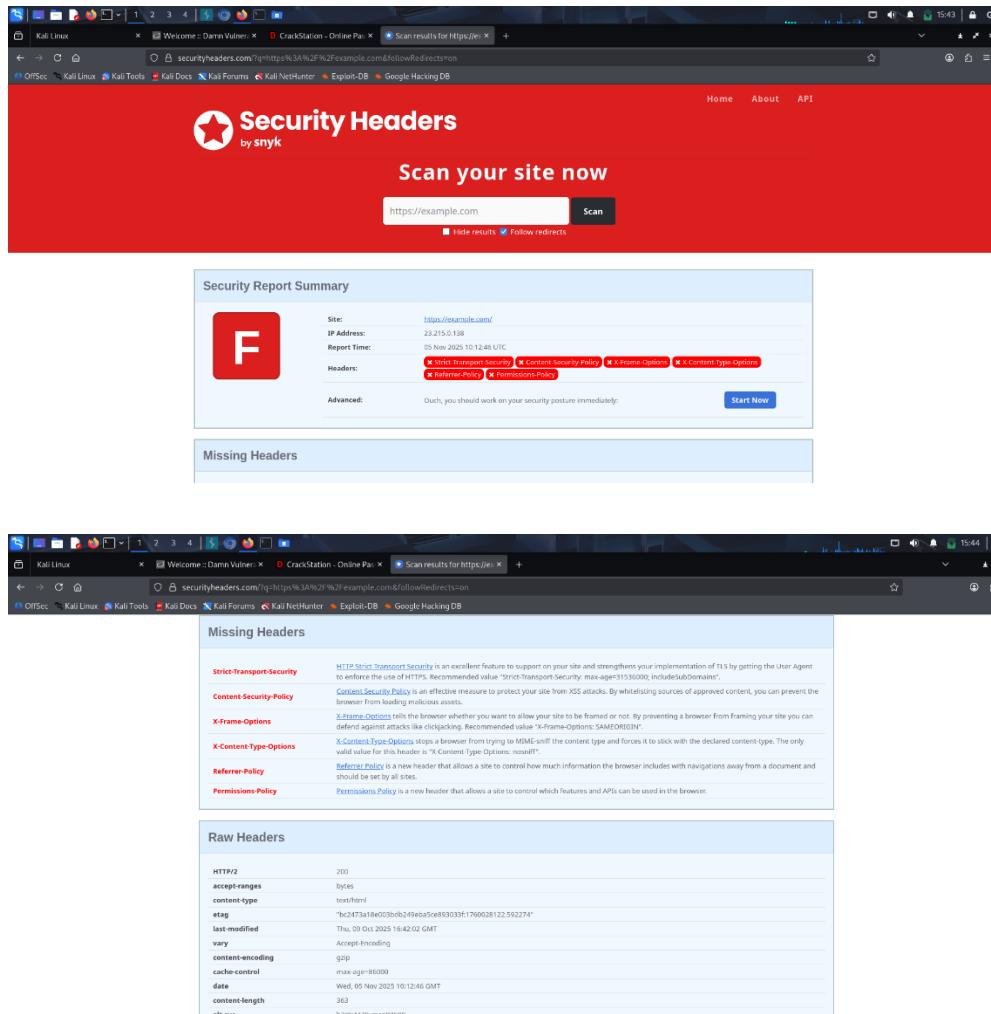
More Training Resources

DVWA aims to cover the most commonly seen vulnerabilities found in today's web applications. However there are plenty of other issues with web applications. Should you wish to explore any additional attack vectors, or want

Task 3 — Web Application Security

Intern Name: **Nitesh Sharma** Date: **05/10/2025**

6. Web-Security Headers:-



Task 3 — Web Application Security

Intern Name: **Nitesh Sharma** Date: **05/10/2025**