[红日安全]代码审计Day2 - filter_var函数缺陷

红日安全 / 2018-07-18 20:44:03 / 浏览数 6320 技术文章 技术文章 顶(0) 踩(0)

---

本文由红日安全成员： 七月火 编写，如有不当，还望斧正。

## 前言

大家好，我们是红日安全-代码审计小组。最近我们小组正在做一个PHP代码审计的项目，供大家学习交流，我们给这个项目起了一个名字叫 PHP-Audit-Labs 。现在大家所看到的系列文章，属于项目 第一阶段 的内容，本阶段的内容题目均来自 PHP SECURITY CALENDAR 2017 。对于每一道题目，我们均给出对应的分析，并结合实际CMS进行解说。在文章的最后，我们还会留一道CTF题目，供大家练习，希望大家喜欢。下面是 第2篇 代码审计文章：

## Day 2 - Twig

题目叫做Twig，代码如下：

```php
// composer require "twig/twig"
require 'vendor/autoload.php';
class Template {
    private $twig;

    public function __construct() {
        $indexTemplate = '<img ' .'src="https://loremflickr.com/320/240">' .
            '<a href="{{link|escape}}">Next slide »</a>';
        // Default twig setup, simulate loading
        // index.html file from disk
        $loader = new Twig\Loader\ArrayLoader(['index.html' => $indexTemplate]);
        $this->twig = new Twig\Environment($loader);
    }

    public function getNexSlideUrl() {
        $nextSlide = $_GET['nextSlide'];
        return filter_var($nextSlide, FILTER_VALIDATE_URL);
    }

    public function render() {
        echo $this->twig->render('index.html',['link' => $this->getNexSlideUrl()]);
    }
}

(new Template())->render();
```



漏洞解析 ：

这一关题目实际上用的是PHP的一个模板引擎 Twig ，本题考察XSS(跨站脚本攻击)漏洞。虽然题目代码分别用了 escape 和 filter_var 两个过滤方法，但是还是可以被攻击者绕过。在上图 第8行 中，程序使用 Twig 模板引擎定义的 escape 过滤器来过滤link，而实际上这里的 escape 过滤器，是用PHP内置函数 htmlspecialchars 来实现的，具体可以点击 这里 了解 escape 过滤器， htmlspecialchars 函数定义如下：

> htmlspecialchars ：(PHP 4, PHP 5, PHP 7)
>
> 功能 ：将特殊字符转换为 HTML 实体
>
> 定义 ：string htmlspecialchars ( string `$string` [, int `$flags` = ENT_COMPAT | ENT_HTML401 [, string`$encoding` = ini_get("default_charset") [, bool `$double_encode` = TRUE ]]] )

```
& (& ■■)  ===============  &amp;
" (■■■)  ===============  &quot;
' (■■■)  ===============  &apos;
< (■■■)  ===============  &lt;
> (■■■)  ===============  &gt;
```

第二处过滤在 第17行 ，这里用了 filter_var 函数来过滤 nextSlide 变量，且用了 FILTER_VALIDATE_URL 过滤器来判断是否是一个合法的url，具体的 filter_var 定义如下：

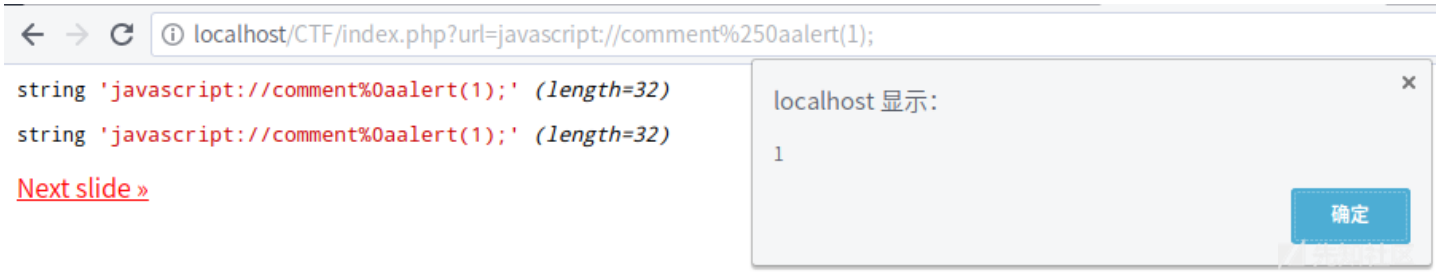
> filter_var ： (PHP 5 >= 5.2.0, PHP 7)
>
> 功能 ：使用特定的过滤器过滤一个变量
>
> 定义 ：mixed filter_var ( mixed `$variable` [, int `$filter` = FILTER_DEFAULT [, mixed `$options` ]] )

针对这两处的过滤，我们可以考虑使用 javascript伪协议 来绕过。为了让大家更好理解，请看下面的demo代码：

```php
<?php
    $url = filter_var($_GET['url'],FILTER_VALIDATE_URL);
    var_dump($url);
    $url = htmlspecialchars($url);
    var_dump($url);
    echo "<a href='$url'>Next slide »</a>";

?>
```

我们使用 payload ：`?url=javascript://comment■250aalert(1)` ，可以执行 alert 函数：




实际上，这里的 // 在JavaScript中表示单行注释，所以后面的内容均为注释，那为什么会执行 alert 函数呢？那是因为我们这里用了字符 %0a ，该字符为换行符，所以 alert 语句与注释符 // 就不在同一行，就能执行。当然，这里我们要对 % 百分号编码成 %25 ，因为程序将浏览器发来的payload：`javascript://comment■250aalert(1)` 先解码成： `javascript://comment%0aalert(1)` 存储在变量 $url 中（上图第二行代码），然后用户点击a标签链接就会触发 alert 函数。

## 实例分析

本次实例分析，我们选取的是 Anchor 0.9.2
版本，在该版本中，当用户访问一个不存在的URL链接时，程序会调用404模板，而这个模板则存在XSS漏洞，具体代码如下：

```php
<?php theme_include('header'); ?>
    <section class="content wrap">
        <h1>Page not found</h1>
        <p>Unfortunately, the page <code>/<?php echo current_url(); ?></code>
        could not be found. Your best bet is either to try the <a href=
        "<?php echo base_url(); ?>">homepage</a>, try <a href="#search">searching</a>
        ,or go and cry in a corner (although I don't recommend the latter).</p>
    </section>
<?php theme_include('footer'); ?>
```

该代码在 themes\default\404.php 中，看第4行 code 标签中的 current_url 函数，我们可在 anchor\functions\helpers.php 文件中，看到 current_url 函数是由 Uri 类的 current 方法实现的，具体代码如下：

```
function current_url() {
   return Uri::current();
}
```

我们跟进到 Uri 类，在 system\uri.php 文件中，我们发现这里调用了 static::detect 方法( statci:: 是在PHP5.3版本之后引入的延迟静态绑定写法)。

```php
public static function current() {
    if(is_null(static::$current)) static::$current = static::detect();
        return static::$current;
}
```



在 current 方法下面，我们就可以找到 detect 方法，该方法会获取 $_SERVER 数组中的 'REQUEST_URI' 、'PATH_INFO', 、'ORIG_PATH_INFO' 三个键的值(下图第3-4行代码)，如果存在其中的某一个键，并且符合 filter_var($uri, FILTER_SANITIZE_URL) 和 parse_url($uri, PHP_URL_PATH) ，则直接将 $uri 传入 static::format 方法(下图第10-14行代码)，具体代码如下：

```php
public static function detect() {
    // create a server object from global
    $server = new Server($_SERVER);
    $try = array('REQUEST_URI', 'PATH_INFO', 'ORIG_PATH_INFO');

    foreach($try as $method) {
        // make sure the server var exists and is not empty
        if($server->has($method) and $uri = $server->get($method)) {
            // apply a string filter and make sure we still have somthing left
            if($uri = filter_var($uri, FILTER_SANITIZE_URL)) {

                // make sure the uri is not malformed and return the pathname
                if($uri = parse_url($uri, PHP_URL_PATH)) {
                    return static::format($uri, $server);
                }
                // woah jackie, we found a bad'n
                throw new ErrorException('Malformed URI');
            }
        }
    }
    throw new OverflowException('Uri was not detected. Make sure the
    REQUEST_URI is set.');
}
```



我们跟进 static::format 方法，可以发现程序过滤了三次(下图第3-7行)，但是都没有针对XSS攻击进行过滤，只是为了获取用户访问的文件名，具体代码如下：

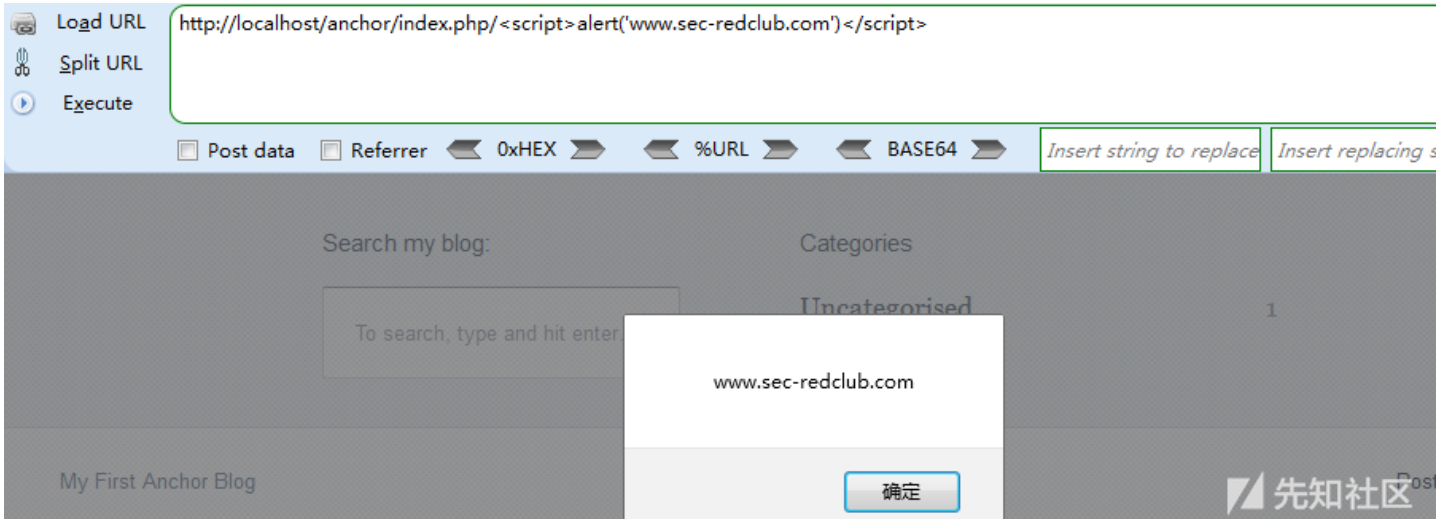
```php
public static function format($uri, $server) {
    // Remove all characters except letters,digits and $-_.+!*'(),{}|\\^~[]`<>#%";/?:@&=.
    $uri = filter_var(rawurldecode($uri), FILTER_SANITIZE_URL);
    // remove script path/name
    $uri = static::remove_script_name($uri, $server);
    // remove the relative uri
    $uri = static::remove_relative_uri($uri);
    // return argument if not empty or return a single slash
    return trim($uri, '/') ?: '/';
}
public static function remove_script_name($uri, $server) {
    return static::remove($server->get('SCRIPT_NAME'), $uri);
}
public static function remove_relative_uri($uri) {
    // remove base url
    if($base = Config::app('url')) {
        $uri = static::remove(rtrim($base, '/'), $uri);
    }
    // remove index
    if($index = Config::app('index')) {
        $uri = static::remove('/' . $index, $uri);
    }
    return $uri;
}
```



由于没有针对XSS攻击进行过滤，导致攻击十分容易，我们来看看XSS攻击具体是如何进行的。

## 漏洞利用

我们构造payload如下： `http://localhost/anchor/index.php/<script>alert('www.sec-redclub.com')</script>` 。根据上面的分析，当我们访问这个并不存在的链接时，程序会调用404模板页面，然后调用 current_url 函数来获取当前用户访问的文件名，也就是最后一个 / 符号后面的内容，所以最终payload里的 `<script>alert('www.sec-redclub.com')</script>` 部分会嵌入到 `<code>` 标签中，造成XSS攻击，效果图如下：




## 修复建议

这对XSS漏洞，我们最好就是过滤关键词，将特殊字符进行HTML实体编码替换，这里给出的修复代码为Dedecms中防御XSS的方法，大家可以在 uploads/include/helpers/filter.helper.php 路径下找到对应代码，具体防护代码如下：

```php
if (!function_exists('RemoveXSS'))
{
    function RemoveXSS($val) {
       $val = preg_replace('/([\x00-\x08,\x0b-\x0c,\x0e-\x19])/', '', $val);
       $search = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!@#$%^&*()';
       $search .= '~`";:?+/={}[]-_|\'\\';
       for ($i = 0; $i < strlen($search); $i++) {
          $val = preg_replace('/(&#[xX]0{0,8}'.dechex(ord($search[$i])).';?)/i', $search[$i], $val);
          $val = preg_replace('/(&#0{0,8}'.ord($search[$i]).';?)/', $search[$i], $val);
       }
       $ra1 = array('javascript', 'vbscript', 'expression', 'applet', 'meta', 'xml', 'blink', 'link',
                    'style', 'script', 'embed', 'object', 'iframe', 'frame', 'frameset', 'ilayer',
                    'layer', 'bgsound', 'title', 'base');
       $ra2 = array('onabort', 'onactivate', 'onafterprint', 'onafterupdate', 'onbeforeactivate',
                    'onbeforecopy', 'onbeforecut', 'onbeforedeactivate', 'onbeforeeditfocus',
                    'onbeforepaste', 'onbeforeprint', 'onbeforeunload', 'onbeforeupdate', 'onblur',
                    'onbounce', 'oncellchange', 'onchange', 'onclick', 'oncontextmenu', 'oncontrolselect',
                    'oncopy', 'oncut', 'ondataavailable', 'ondatasetchanged', 'ondatasetcomplete',
                    'ondblclick', 'ondeactivate', 'ondrag', 'ondragend', 'ondragenter', 'ondragleave',
                    'ondragover', 'ondragstart', 'ondrop', 'onerror', 'onerrorupdate', 'onfilterchange',
                    'onfinish', 'onfocus', 'onfocusin', 'onfocusout', 'onhelp', 'onkeydown', 'onkeypress',
                    'onkeyup', 'onlayoutcomplete', 'onload', 'onlosecapture', 'onmousedown', 'onmouseenter',
                    'onmouseleave', 'onmousemove', 'onmouseout', 'onmouseover', 'onmouseup', 'onmousewheel',
                    'onmove', 'onmoveend', 'onmovestart', 'onpaste', 'onpropertychange', 'onreadystatechange',
                    'onreset', 'onresize', 'onresizeend', 'onresizestart', 'onrowenter', 'onrowexit',
                    'onrowsdelete', 'onrowsinserted', 'onscroll', 'onselect', 'onselectionchange',
                    'onselectstart', 'onstart', 'onstop', 'onsubmit', 'onunload');
       $ra = array_merge($ra1, $ra2);

       $found = true;
       while ($found == true) {
          $val_before = $val;
          for ($i = 0; $i < sizeof($ra); $i++) {
             $pattern = '/';
             for ($j = 0; $j < strlen($ra[$i]); $j++) {
                if ($j > 0) {
                   $pattern .= '((&#[xX]0{0,8}([9ab]);)||(&#0{0,8}([9|10|13]);))*';
                }
                $pattern .= $ra[$i][$j];
             }
             $pattern .= '/i';
             $replacement = substr($ra[$i], 0, 2).'<x>'.substr($ra[$i], 2);
             $val = preg_replace($pattern, $replacement, $val);
             if ($val_before == $val) {
                $found = false;
             }
          }
       }
       return $val;
    }
}
```



## 结语

看完了上述分析，不知道大家是否对 filter_var 函数绕过有了更加深入的理解，文中用到的CMS可以从 这里
下载，当然文中若有不当之处，还望各位斧正。如果你对我们的项目感兴趣，欢迎发送邮件到 hongrisec@gmail.com 联系我们。 Day2
的分析文章就到这里，我们最后留了一道CTF题目给大家练手，题目如下：

```
// index.php
<?php
$url = $_GET['url'];
if(isset($url) && filter_var($url, FILTER_VALIDATE_URL)){
   $site_info = parse_url($url);
   if(preg_match('/sec-redclub.com$/',$site_info['host'])){
       exec('curl "'.$site_info['host'].'"', $result);
       echo "<center><h1>You have curl {$site_info['host']} successfully!</h1></center>
             <center><textarea rows='20' cols='90'>";
       echo implode(' ', $result);
   }
   else{
       die("<center><h1>Error: Host not allowed</h1></center>");
   }
```

```
}
else{
   echo "<center><h1>Just curl sec-redclub.com!</h1></center><br>
         <center><h3>For example:?url=http://sec-redclub.com</h3></center>";
}

?>

// f1agi3hEre.php
<?php
$flag = "HRCTF{f1lt3r_var_1s_s0_c00l}"
?>
```

题解我们会阶段性放出，如果大家有什么好的解法，可以在文章底下留言，祝大家玩的愉快！

## 相关文章

Anchor CMS 0.9.2: XSS

点击收藏 | 0 关注 | 1

上一篇：Meepwn2018：PyCalx... 下一篇：WebLogic任意文件上传漏洞复...

1. 9 条回复

御林铁卫 2018-07-19 18:26:49

url=test://"|cat<>f1agi3hEre.php;"sec-redclub.com

0 回复Ta

---

红日安全 2018-07-19 20:21:22

@御林铁卫 棒棒哒

0 回复Ta

---

weigr 2018-10-26 15:09:46

@红日安全 大佬，请问在windows下怎么过呢？？？

0 回复Ta

---

weigr 2018-10-26 23:26:07

@weigr windows下弄了半天总算成功啦，真是菜的扣脚。

```
test://"|type=flag.php;sec-redclub.com
```

0 回复Ta

---

白猫 2018-12-03 20:31:12

红日安全的师傅,有个问题想请教一下,就是您在文中说可以利用JavaScript伪协议来绕过filter_var 过滤,这里不是很明白是怎么绕过,麻烦您解释一下可以吗?

0 回复Ta

---

xubowen1566553** 2019-01-02 12:24:08

@白猫 类似http:// file:// test:// 都可以通过filter_var的检验，javascript:// 这里的//是单行注释，%0a是换行

0 回复Ta

---

xubowen1566553** 2019-01-02 12:40:27

@御林铁卫 大佬 为什么http://开头就不行呢

0 回复Ta

---

xubowen1566553** 2019-01-02 12:45:39

@御林铁卫 ok 明白了 过不了fileter_var

0 回复Ta

---

roothex 2019-07-29 10:59:29

看着修复建议是dedecms时总觉得心里不踏实，搜了一下果然这个过滤类是可以被绕过的，https://www.freebuf.com/vuls/181783.html

0 回复Ta

---

登录 后跟帖

先知社区

---

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板