【漏洞分析】METINFO 6.1.2 版本 XSS 漏洞分析

文/ icuke

0x00 漏洞概述

漏洞类型：XSS 漏洞
危险等级：中
CVE 编号：CVE-2018-18374
利用条件：该漏洞需要管理员在登录状态下访问攻击者构造的恶意链接。
漏洞位置：head.php 中的第 20 行 echo 输出语句
漏洞产生原因：echo 输出的 script 标签代码中，anyid 值由外部输入获得，且没有做很好过滤
受影响版本：全部版本
修复版本：无

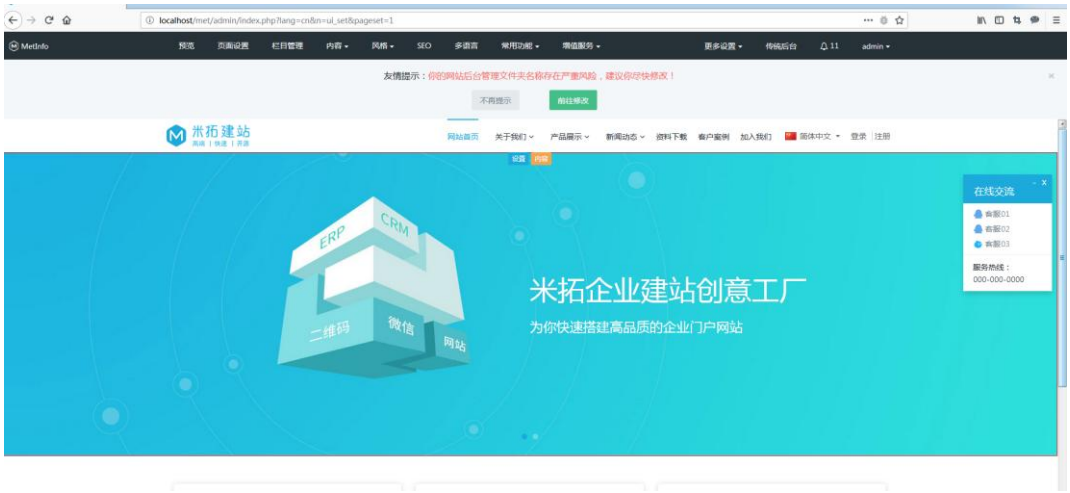
0x01 环境准备

IDE	PhpStorm
Web 环境	phpstudy 集成环境
PHP 版本	5.4.45
MetInfo 版本	6.1.2

0x02 漏洞复现

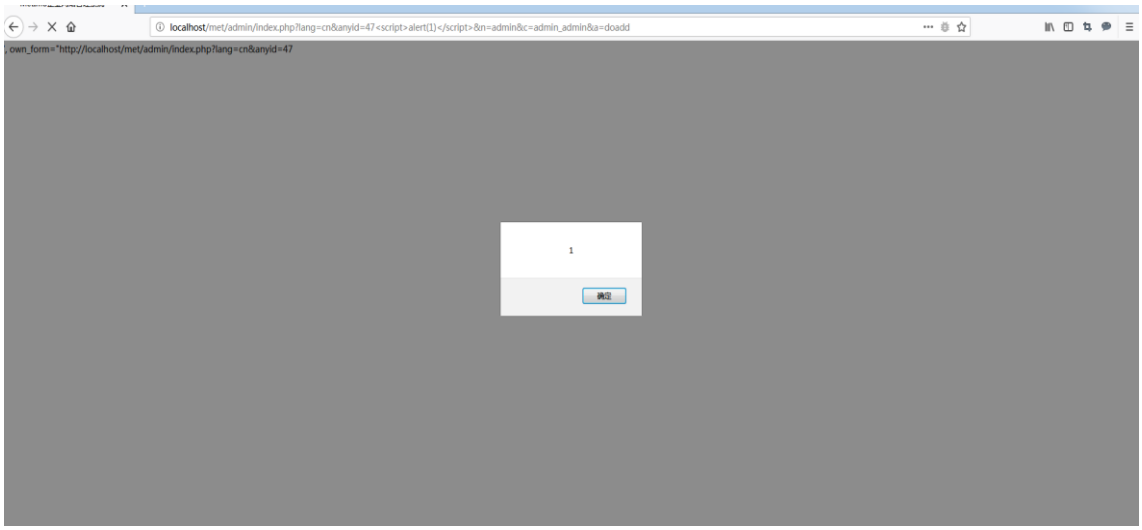
1. 管理员登录管理后台

使用 firefox 访问 localhost/met/admin/登录管理后台。



2. 管理员访问构造的恶意链接

使用 firefox 访问 http://localhost/met/admin/index.php?lang=cn&anyid=47%3Cscript%3Ealert(1)%3C/script%3E&n=admin&c=admin_admin&a=doadd，触发漏洞。

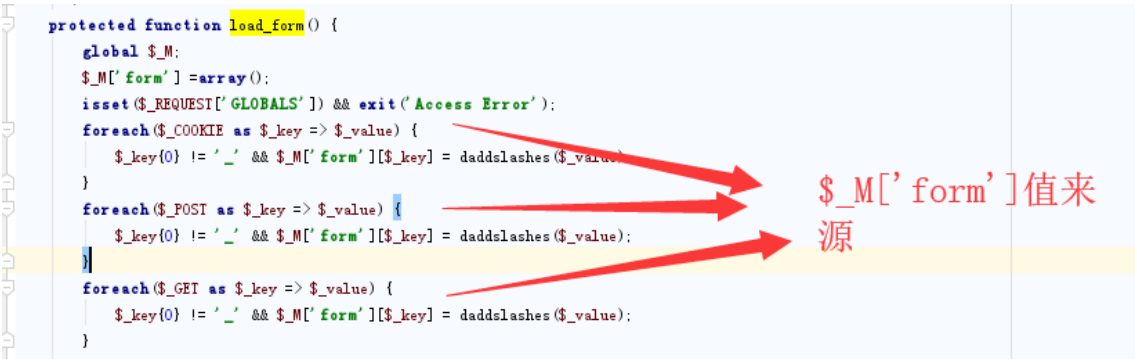


0x03 漏洞分析

1. 漏洞代码分析

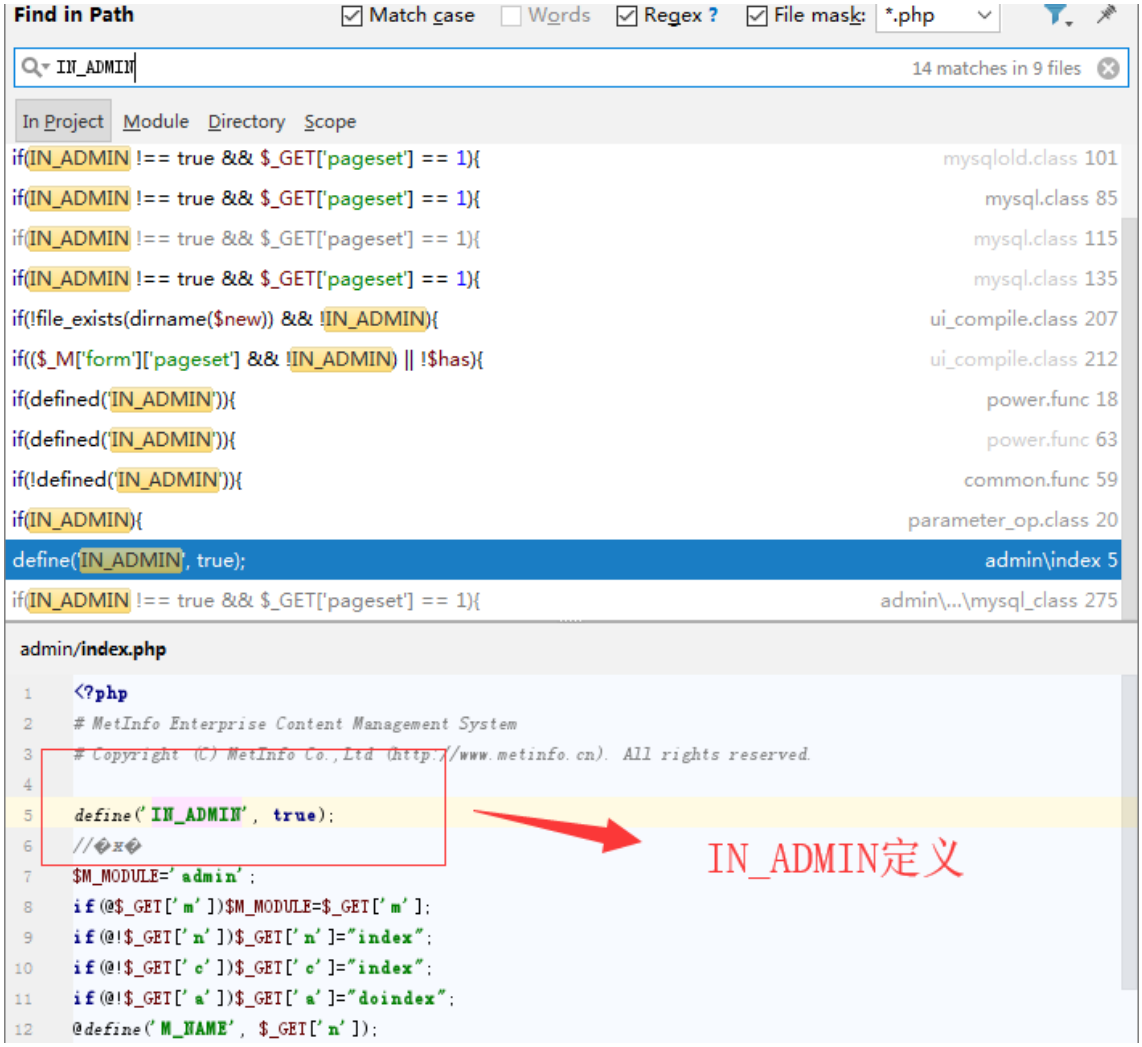
漏洞代码在 app/system/include/public/ui/admin/head.php 文件 20 行的 echo 输出语句中：

阅读代码可知，anyid 的内容将直接被输出到 script 标签中，变量值来自 $_M['form']['anyid']，看下 $_M['form']['anyid'] 变量值来源，在 app/system/include/class/common.class.php 中的 load_form() 方法里，可以看到 $_COOKIE、$_POST、$_GET 变量的 key 为 $_M['form'] 的 key，value 需要经过 daddslashes 函数处理。



daddslashes 过滤函数：

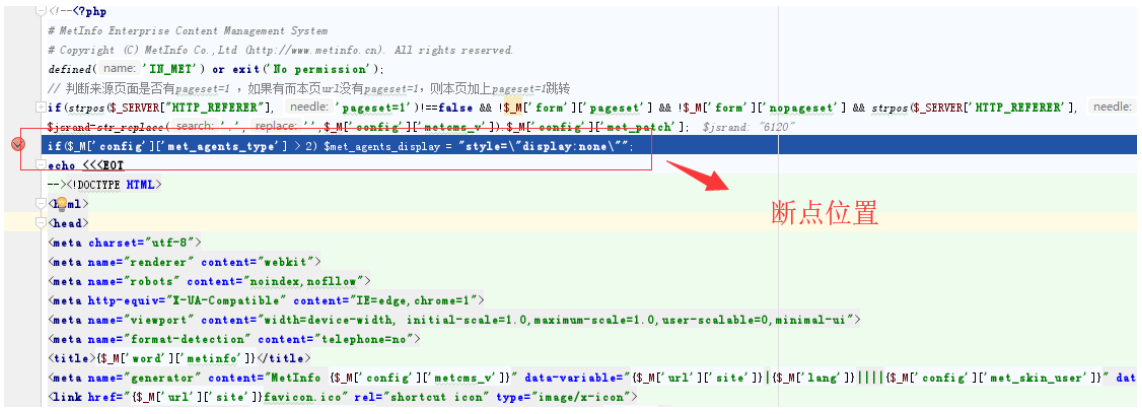
第 59 行代码中，如果未定义"IN_ADMIN"，会使用 addslashes、sqlinsert 函数对传入变量进行处理，如果定义"IN_ADMIN"，只使用 addslashes 进行处理。全局搜索下 IN_ADMIN 的定义位置，在 admin/index.php 文件中有定义。



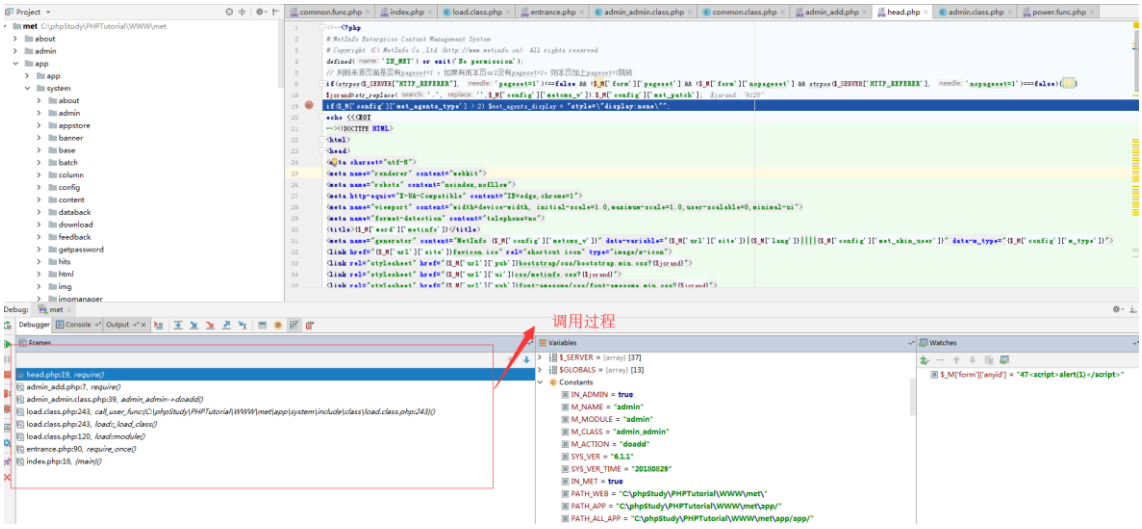
我们就是从 admin/index.php 文件入口进行利用的，所以对于 $_GET、$_POST、$_COOKIE 传入的变量值只会进行 addslashes 处理。

2. 回溯触发路径

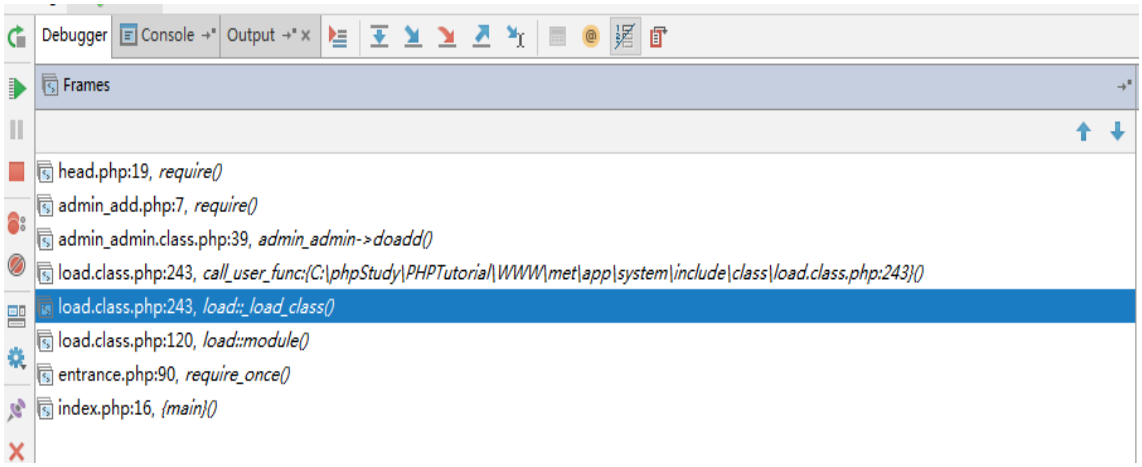
此处使用 PhpStorm xdebug 调试方式进行动态调试。在 app/system/include/public/ui/admin/head.php 的 19 行设置断点



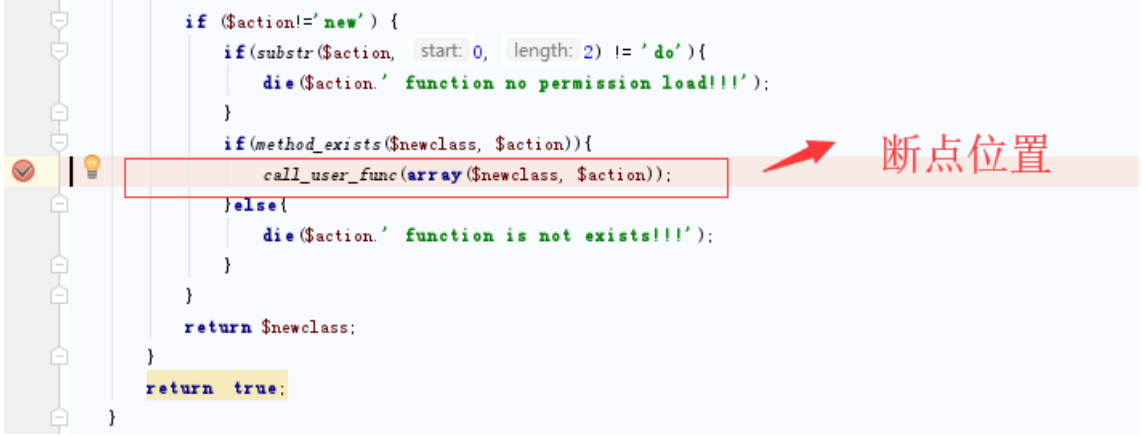
管理员登录后台后访问 http://localhost/met/admin/index.php?lang=cn&anyid=47%3Cscript%3Ealert(1)%3C/script%3E&n=admin&c=admin_admin&a=doadd，触发断点。



左下方为调用过程：



在 app/system/include/class/load.class.php 的 243 行代码下断点，重新调试。



$newclass 是一个类对象，我们 F7 跟踪 call_user_func 函数调用，进入 admin_admin 的 doadd 方法，类 admin_admin 存在构造函数，会调用父类的构造方法，父类为 admin。

类 admin 继承 common 类，查看 common 类，构造函数调用 load_form 方法，完成对 $_M['form'] 的赋值过程。

```php
*/
class common {

    /**
     * 初始化
     */
    public function __construct() {
        global $_M;//全局数组$_M
        ob_start();//开启缓存
        $this->load_mysql();//数据库连接
        $this->load_form();//表单过滤
        $this->load_lang();//加载语言配置
        $this->load_config_global();//加载全站配置数据
        $this->load_url_site();
        $this->load_config_lang();//加载当前语言配置数据
        $this->load_url();//加载url数据
    }

    /**
     * 链接数据库
     */
    protected function load_mysql() {
        global $_M;
        $db_settings = array();
        $db_settings = parse_ini_file( filename: PATH_CONFIG.'config_db.php');
        @extract($db_settings);
        DB::dbconn($con_db_host, $con_db_id, $con_db_pass, $con_db_name, $con_db_port);
        $_M['config']['tablepre'] = $tablepre;
        return true;
    }
```

类common构造函数调用load_form

再看下类 admin 的构造过程，其中有一步会调用 check() 方法，在此处下一断点，重新调试，并使用 F7 跟进该调用过程。

```php
protected function check() {
    global $_M;
    $http = isset($_SERVER['REQUEST_SCHEME']) ? $_SERVER['REQUEST_SCHEME'] : 'http';
    $current_url = $http.'://'.$_SERVER['HTTP_HOST'].$_SERVER['REQUEST_URI'];
    $login_url = $_M['url']['site_admin']."index.php?m=login&c=login&a=doindex";
    if (strstr($current_url, needle: $_M['url']['site_admin']."index.php")) {
        $admin_index = 1;
    } else {
        $admin_index = '';
    }
    $met_adminfile = $_M['config']['met_adminfile'];
    $met_admin_table = $_M['table']['admin_table'];
    $metinfo_admin_name = get_met_cookie('metinfo_admin_name');
    $metinfo_admin_pass = get_met_cookie('metinfo_admin_pass');

    if (!$metinfo_admin_name || !$metinfo_admin_pass) {
        if ($admin_index) {
            met_cookie_unset();
            met_setcookie("re_url", $re_url, time()-3600);
            Header( string: "Location: ".$login_url);
        } else {
            if (!$re_url) {
                $re_url = $_SERVER['HTTP_REFERER'];
                $HTTP_REFERERs = explode( delimiter: '?', $_SERVER['HTTP_REFERER']);
                $admin_file_len1 = strlen( string: "/{$met_adminfile}/");
                $admin_file_len2 = strlen( string: "/{$met_adminfile}/index.php");
                if(strrev(substr(strrev($HTTP_REFERERs[0]), start: 0, $admin_file_len1)) == "/{$met_adminfile}/" || strrev(substr(strrev($HTTP_REFERERs[0]), start: 0,$admin_file_len2)) == "/{$met_adminfile}/index.php"
                    $re_url = "{$http}://{$_SERVER['SERVER_NAME']}{$_SERVER['REQUEST_URI']}";
                }
            }
            if (!$_COOKIE[re_url]&&!strstr($re_url, needle: "return.php")) met_setcookie("re_url", $re_url, time()+3600);
            met_cookie_unset();
            Header( string: "Location: ".$login_url);
        }
        exit;
    } else {
        $query = "SELECT * FROM {$_M['table']['admin_table']} WHERE admin_id = '{$metinfo_admin_name}' AND admin_pass = '{$metinfo_admin_pass}' AND usertype = '3'";
```

登录判断

阅读代码可知，会从 cookie 中取 metinfo_admin_name ，met_info_admin_pass 进行判断，若判断未通过，则重定向到登录页面。

F8 回到 admin_admin.class.php 的 doadd 方法调用

```
26    public function doadd() {
27        global $_M;
28        $a = 'doaddsave';
29        $list['admin_group'] = '3';
30        $list['lang'] = load::mod_class( classname: 'language/language_op', action: 'new')->get_lang();
31        foreach ($list['lang'] as $key => $val) {
32            $list['lang_check'] .= $val['mark'].'|';
33        }
34        $list['lang_check'] .= '#metinfo#';
35        $list['admin_issueok'] = 1;
36        $list['op_check'] = "metinfo|add|editor|del";
37        $list['pop_check'] = 'all';
38        $metinfocolumn = $this->admin_list();
39        require $this->template( path: 'own/admin_add');
40    }
41
```

调用template

运行到 39 行 F7 步入：

```
protected function template($path){
    global $_M;
    // 前缀、路径转换优化（新模板框架v2）
    $dir = explode( delimiter: '/', $path);
    $postion = $dir[0];
    $file = substr(strstr($path, needle: '/'), start: 1);

    if ($postion == 'own') {
        return PATH_OWN_FILE."templates/{$file}.php";
    }
    if ($postion == 'ui') {
        if (M_MODULE == 'admin') {
            $ui = 'admin';
        } else {
            $ui = 'web';
        }
        return PATH_SYS."include/public/ui/{$ui}/{$file}.php";
    }
```

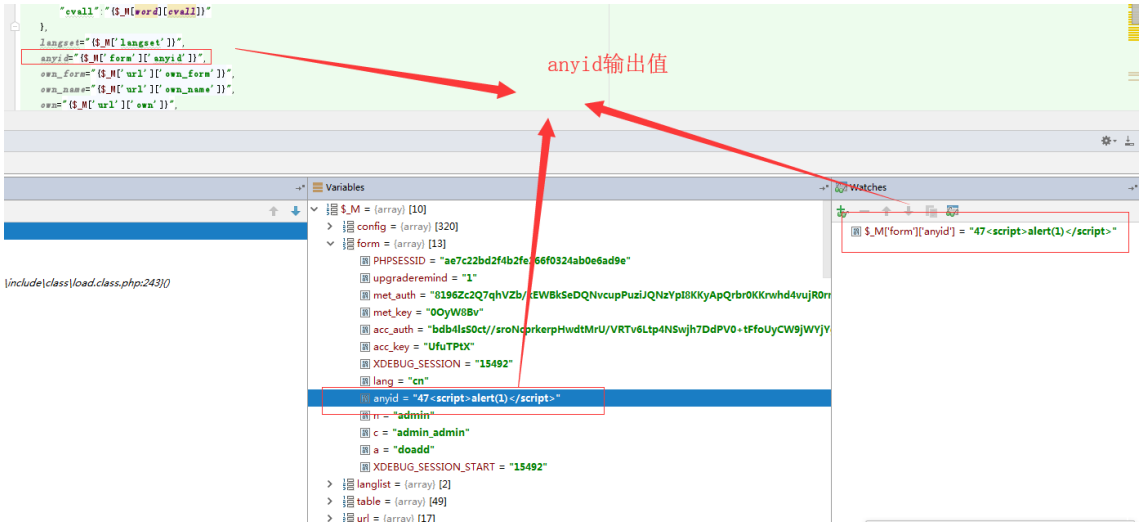
此处传入的$path值为own/admin_add

发现 template 方法是用于模板寻址，此处我们寻址地址为 template/admin_add.php。F8 跳出，进入 admin_add.php 文件。

```
5   defined( name: 'IN_MET') or exit('No permission');
6
7   require $this->template('ui/head');
8   echo <<<EOT
9   -->
10
11  <link rel="stylesheet" href="{$_M[url][own_tem]}css/metinfo.css?{$jsrand}" />
12  <form method="POST" class="ui-from" name="myform" action="{$_M[url][own_form]}a={$a}" target="_self">
13      <input type="hidden" name='id' value="{$_M['form']['id']}" />
14      <div class="v52fmbx">
15      <h3 class="v52fmbx_hr">{$_M['word']['admininfo']}</h3>
16      <dl>
17          <dt>{$_M['word']['adminusername']}</dt>
18          <dd class="ftype_input">
19              <div class="fbox">
20                  <input type="text" name="admin_id" value="{$list['admin_id']}" data-required="1">
21              </div>
22          </dd>
23      </dl>
24      <dl>
25        <dt>{$_M['word']['adminpassword']}</dt>
```

调用template传入/ui/head

第 7 行调用后进入 head.php 文件，触发漏洞执行。



0x04 总结

当管理员登录后台后，通过 url 传递的参数值 anyid 不会经过 sqlinsert 函数过滤，只经过 addslashes 处理后就返回到前台页面 script 标签中。

提供另一个触发链接：

http://localhost/met/admin/index.php?lang=cn&n=system&c=news&a=doindex&anyid=12%3Cscript%3Ealert(1)%3C/script%3E

0x05 修复方法

修改 app/system/include/function/common.func.php 的 daddslashes 方法，当 IN_ADMIN

为 true 时，对$string 进行标签过滤。