WordPress权限提升漏洞分析

大闸蟹清蒸最好吃 / 2018-12-22 08:46:00 / 浏览数 3176 技术文章 翻译文章 顶(0) 踩(0)

WordPress权限提升漏洞分析

原文：https://blog.ripstech.com/2018/wordpress-post-type-privilege-escalation/

0x00 前言

WordPress博客文章创建过程中存在一个逻辑缺陷，攻击者可以访问只有管理员能够访问的功能，这将导致WordPress core中存在存储型XSS（Stored XSS）和对象注入（Object Injection），也导致WordPress最受欢迎的插件Contact Form 7和Jetpack中存在更为严重的漏洞。

0x01 漏洞影响

WordPress本质上是一款博客软件，可以让用户创建和发布文章（post）。随着时间的推移，WordPress引入了不同的文章类型（post type），例如页面和媒体条目（图像、视频等）。WordPress插件可以注册新的文章类型，例如产品或联系表单（contact form）。根据插件已注册的文章类型，WordPress可以提供许多独特的新功能。例如，联系表单插件可能允许用户创建具有文件上传字段的联系表单（比如用来上传简历）core中出现存储型XSS和对象注入漏洞。根据具体安装的插件情况，攻击者还可以利用更为严重的漏洞。例如，当使用WordPress安装了Contact Form 7时（这款插件非常流行，超过500万个安装记录），攻击者就能够读取目标Wordpress站点的数据库凭据。大多数流行的WordPress插件都容易受到此权限提升漏洞的影响

0x02 技术背景

为了注册新的文章类型，插件需要调用`register_post_type()`，传入新文章类型的名称以及一些元信息。

```
// Example post type
register_post_type( 'example_post_type', array(
   'label' => 'Example Post Type',     // The name of the type in the front end
   'can_export' => true,               // Make it possible to export posts of this type,
   'description' => 'Just an example!' // A short description
));
```

0x03 自定义文章类型的安全防护机制

每个文章类型都有自己的编辑页面（比如`example.com/wordpress/wp-admin/?page=example_post_type_editor`）。

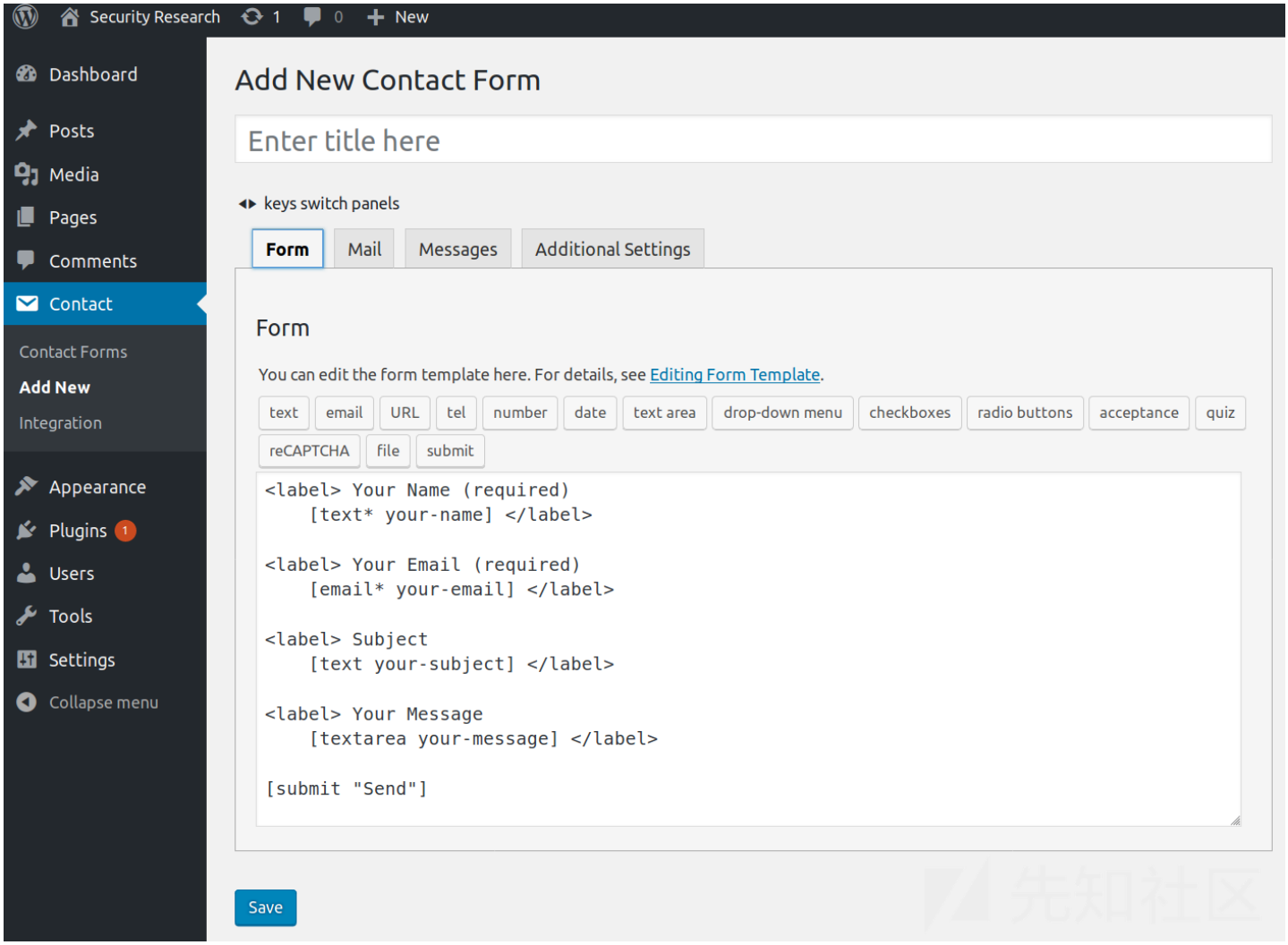

图1. Contact Form文章类型编辑页面，只有管理员能够访问该页面。

如果插件开发者决定只有管理员能够使用插件注册的文章类型，那么就应该在页面开头处检查用户是否为管理员，如果不满足条件则停止运行。

`/wp-content/plugins/example_plugin/example_post_type_editor.php`：

```
if(!current_user_is_administrator()) {
   die("You are not an administrator and not allowed to use this post type.");
}
```

0x04 WordPress文章提交逻辑

虽然所有已注册的文章类型都有自己的编辑器，但还是都可以使用WordPress的文章提交API，也能使用WordPress函数`wp_write_post()`来插入和更新文章。该函数接受

在WordPress文章提交的第一个步骤中，WordPress需要知道用户是想编辑已有的文章还是创建新的文章。为了做到这一点，WordPress会检查用户是否发送了文章的ID。

`/wp-admin/post.php`：

```
if ( isset( $_GET['post'] ) )
   $post_id = $post_ID = $_GET['post'];
elseif ( isset( $_POST['post_ID'] ) )
   $post_id = $post_ID = $_POST['post_ID'];

if($post_id)
   ■
```

在下个步骤中，WordPress需要判断用户尝试创建哪种文章类型。如果发送了文章ID，WordPress就会从数据库的`wp_posts`表中提取`post_type`列。
如果用户想要创建新文章，那么目标文章类型为`$_POST['post_type']`。

`/wp-admin/post.php`：

```
if ( isset( $_GET['post'] ) )
   $post_id = $post_ID = $_GET['post'];
elseif ( isset( $_POST['post_ID'] ) )
   $post_id = $post_ID = $_POST['post_ID'];
```

```
if($post_id)
   $post_type = get_post_type($post_id);
else
   $post_type = $_POST['post_type'];
```

一旦WordPress知道用户正在创建或编辑的文章类型，就会检查用户是否可以使用这种文章类型。
为了做到这一点，WordPress会验证只能从目标文章类型编辑页面获取的一个nonce值。

WordPress会执行如下代码验证nonce值。

/wp-admin/post.php：

```
if($post_id)
   $post_type = get_post_type($post_id);
else
   $post_type = $_POST['post_type'];

$nonce_name = "add-" . $post_type;
if(!wp_verify_nonce($_POST['nonce'], $nonce_name))
   die("You are not allowed to use this post type!");
```

如果$post_type为post，那么$nonce_name值就等于add-post。如果$post_type为example_post_type，则$nonce_name就等于add-example_post_type。只

0x05 存在的问题

虽然较低权限的攻击者（例如处于contributor（投稿者）权限的攻击者）无法访问示例文章类型的页面和nonce值，但总能获得普通文章的nonce值，这些文章的类型为po

/wp-admin/post.php：

```
// Send a post ID of a post of post type 'post'
if($post_id)
   // This would return 'post'
   $post_type = get_post_type($post_id);
else
   $post_type = $_POST['post_type'];

// All users can by default create 'posts' and get the nonce to pass this check
$nonce_name = "add-" . $post_type;
if(!wp_verify_nonce($nonce_name))
   die("You are not allowed to create posts of this type!");
```

然而，这种方法只能让攻击者更新现有文章，并且无法覆盖文章的post_type。如果设置了文章ID，WordPress将在更新文章之前从参数中删除post_type。

然而如果设置了$_POST[‘post_ID’]，那么WordPress只会删除$post_type参数。 攻击者可以通过$_POST['post_ID']或$_GET['post']发送文章ID。
如果攻击者通过$_GET['post']发送文章ID，就会出现以下情况：

1、WordPress发现用户设置了帖子ID，从数据库中提取对应的文章类型。

2、WordPress会检查攻击者是否为这个文章类型发送一个有效的nonce值（而攻击者可以从正常帖子中获取该值）。

3、一旦通过了nonce值检查，WordPress就会确定是否应该调用wp_update_post()或wp_insert_post()。WordPress会检查用户是否设置了$_POST[‘post_ID’]

由于WordPress在第三步中忘记检查$_GET['post']，因此攻击者可以通过nonce验证，并创建具有任意文章类型的一篇新文章。我们简化病抽象处理了如下代码片段，实

/wp-admin/post.php：

```
// An attacker sets $_GET['post'] to a post of a post type he can access
if ( isset( $_GET['post'] ) )
   $post_id = $post_ID = $_GET['post'];
elseif ( isset( $_POST['post_ID'] ) )
   $post_id = $post_ID = $_POST['post_ID'];

if($post_id)
   // The post type is now 'post'
   $post_type = get_post_type($post_id);
else
   $post_type = $_POST['post_type'];

// Since the attacker has access to that post type, he can get the nonce and
// pass the nonce verification check
```

```
$nonce_name = "add-" . $post_type;
if(!wp_verify_nonce($nonce_name))
   die("You are not allowed to create posts of this type!");

$post_details = array(
 'post_title' => $_POST['post_title'],
 'post_content' => $_POST['post_content'],
 'post_type' => $_POST['post_type']
);

// WordPress only unsets the post_type if $_POST['post_ID'] is set and forgets to
// check $_GET['post']
if(isset($_POST['post_ID'])) {

   unset($post_details['post_type']);
   $post_details['ID'] = $post_id;
   wp_update_post($post_details);
} else {
   // If we just set $_GET['post'] we will enter this branch and can set the
   // post type to anything we want it to be!
   wp_insert_post($post_details);
}
```

0x06 漏洞利用：通过Contact Forms 7读取wp-config.php

到目前为止，大家应该知道较低权限用户可以滥用这个错误来创建任何类型的文章，并且这个错误对目标网站的影响程度取决于网站已安装的插件以及插件所提供的文章类型

举个例子。处于contributor角色的攻击者有可能利用该漏洞，滥用Contact Form
7（WordPress最流行的一款插件）中的功能来读取目标网站的`wp-config.php`文件内容。该文件中包含数据库凭据信息以及加密秘钥信息。

在5.0.3版的Contact Forms
7中，用户可以设置本地文件附件。当管理员创建联系表单，并且页面的访问者通过该表单联系管理员时，就会向管理员发送一封电子邮件，其中包含用户输入的所有数据。

这意味着攻击者可以简单创建一个新的联系表单，将本地文件附件设置为`../wp-config.php`，并且设置需要将数据发送到攻击者自己的电子邮件，然后提交表单，就能读

0x07 插件开发者修复方案

插件开发者应该在调用`register_post_type()`时，显式设置`capability`和`capability_type`参数来进一步加强插件的安全性。

```
// Example post type
register_post_type( 'example_post_type', array(
   'label' => 'Example Post Type',

   'capability_type' => 'page'     // capability_type of page makes sure that
                                   // only editors and admins can create posts of
                                   // that type
));
```

大家可以阅读WordPress文档中关于使用`register_post_type`来保护文章类型的具体内容。

0x08 WordPress的XMLRPC及REST API

我们可以通过XMLRPC和WordPress的REST
API来创建文章，这些API不会对特定文章类型执行nonce验证。然而通过这些API创建文章时，用户无法设置任意文章的元（meta）字段。只有当用户可以设置这些文章的r

0x09 时间线

日期	进度
2018/08/31	通过官网将漏洞反馈给Contact Form 7
2018/09/02	通过Hackerone向WordPress反馈漏洞
2018/09/04	Contact Form 7修复漏洞
2018/09/27	WordPress安全团队对Hackerone上的漏洞进行了归类
2018/10/12	WordPress在Hackerone发布了补丁
2018/10/18	我们验证补丁有效性
2018/12/13	WordPress在5.0.1版中发布修复补丁

0x0A 总结

具有较低权限的攻击者（如contributor角色，WordPress中权限第二低的角色）可以通过该漏洞创建通常无法访问的某些类型的文章。这样攻击者就可以访问只有管理员能
5插件中发现了2个漏洞。我们估计有数千个插件可能存在这个漏洞。此外，我们还在WordPress的一个内部文章类型中找到出存储型XSS和对象注入漏洞。攻击者可以通过

上一篇：对微软Outlook的CVE-20... 下一篇：PHP反序列化由浅入深

1. 0 条回复
   - 动动手指，沙发就是你的了！