WordPress设计漏洞导致WooCommerce RCE

angel010 / 2018-11-09 09:01:00 / 浏览数 2746 安全技术 漏洞分析 顶(0) 踩(0)

研究人员发现WordPress处理权限的方式存在漏洞会导致WordPress插件提权。影响主流电子商务插件WooCommerce，安装量超过400万。该漏洞允许店铺管理人员删除

技术细节

WordPress处理权限的方式是通过分配特定权限给不同角色。如果定义了店铺管理员（shop manager），就分配给edit_users功能，这样就该角色就可以编辑店铺的客户账户了。这一过程在插件的安装过程就执行了。

```
woocommerce/includes/class-wc-install.php
   // Shop manager role.
add_role(
   'shop_manager',      // Internal name of the new role
   'Shop manager',      // The label for displaying
   array(               // Capabilities
            ■
       'read_private_posts'     => true,
       'edit_users'             => true,
       'edit_posts'             => true,
            ■
   )
);
```

然后该角色会以WordPress核心设置保持在数据库中。这就是说用户角色是不依赖插件的，即使插件不启用，用户角色也是存在的。

如果有非认证的用户尝试编辑另外一个用户，会调用current_user_can()来确保只有有权限的用户才能执行这一动作。
调用 current_user_can()示例：

```
$target_user_id = $_GET['target_user_id'];
if(current_user_can('edit_user', $target_user_id)) {
   edit_user($target_user_id);
}
```

调用的逻辑是"用户是否可以尝试用ID $target_user_id来执行编辑特定用户的动作"。

默认情况下编辑用户（edit_users）功能只允许有此权限的用户编辑包括管理员在内的任意用户来执行更新密码等操作，商铺管理员就有这样的权限。出于安全考虑，Wo
manager可以编辑用户，但只能编辑customer角色的用户。

为了完成这样的功能，WooCommerce这样的插件可以增加meta功能。meta功能可以以函数的方式应用，然后被current_user_can()调用。meta
privilege函数除了简单返回true或false外，返回值还会决定当前用户是否可以执行该动作。WooCommerce的meta privilege过滤器如下所示：

Meta功能示例：

```
function disallow_editing_of_admins( $capability, $target_user_id ) {

   // If the user is an admin return false and disallow the action
   if($capability == "edit_user" && user_is_admin($target_user_id)) {
       return false;
   } else {
       return true;
   }
}
add_filter( 'map_meta_cap', 'disallow_editing_of_admins');
```

当current_user_can('edit_user', 1)被调用时，过滤器就会执行来决定ID 1
($target_user_id)的用户是否是admin，如果是不允许编辑并返回false，否则允许用户继续。WooCommerce更复杂和真实的meta cap
hook保存在woocommerce/includes/wc-user-functions.php的第408行。

设计漏洞

过滤器工作时，只有插件启用时才执行。问题是用户角色保存在数据库中，即使插件禁用了用户角色也存在。也就是说如果WooCommerce被禁用了，限制shop
manager编辑管理员的meta权限检查就不会执行了，但默认允许有edit_users的用户来编辑任意用户的情况就发生了。因此，shop
manager可以更新管理员账户的密码然后获取站点的控制权。

禁用插件

默认情况下，只有管理员可以禁用插件。但RIPS检测到WooCommerce存在任意文件删除漏洞，漏洞允许shop manager删除可写服务器上的任意文件。通过删除WooCommerce的主文件woocommerce.php，WordPress就不能加载该插件了。

文件删除漏洞存在于WooCommerce的登录特征中。日志以.log文件形式保存在wp-content目录下。当shop manager想要删除日志文件，就以GET参数的方式提交文件名。

```
woocommerce/includes/admin/class-wc-admin-status.php
   class WC_Admin_Status
{
   public static function remove_log()
   {
   ■
       $log_handler = new WC_Log_Handler_File();
       $log_handler->remove(wp_unslash($_REQUEST['handle']));
   }
woocommerce/includes/log-handlers/class-wc-log-handler-file.php
   class WC_Log_Handler_File extends WC_Log_Handler
{
   public function remove($handle)
   {
   ■
       $file = trailingslashit(WC_LOG_DIR) . $handle;
   ■
   unlink($file);
```

filename ($handle)是加在日志目录wp-content/wc-logs/之后的，然后传递给unlink()。在设置$handle../../plugins/woocommerce-3.4.5/woocommerce.php时，

POC

POC视频：
https://blog.ripstech.com/videos/wordpress-design-flaw.mp4

影响

研究人员检测并报告了的文件删除漏洞，该漏洞在3.4.6版本中进行了修复。任意文件删除漏洞在大多数情况下并不任务是一个高危漏洞，因为攻击者删除网站的index.ph

总结

本文描述了如何在WordPress网站中删除特定插件文件可以禁用安全检查，并导致整个站点被接管。其根源在于WordPress网站权限系统的设计漏洞，受影响的用户超过40 manager用户角色。而shop manager是负责管理订单、商品和客户的雇员。可以通过XSS漏洞或钓鱼攻击的形式获取访问权限，如果漏洞被攻击者利用，shop manager就可以接管管理员账户并执行任意代码。

https://blog.ripstech.com/2018/wordpress-design-flaw-leads-to-woocommerce-rce/

上一篇：SKREAM（二）:内存地址的随机分配 下一篇：JDK解决反序列化的方法

1. 0 条回复
   - 动动手指，沙发就是你的了！