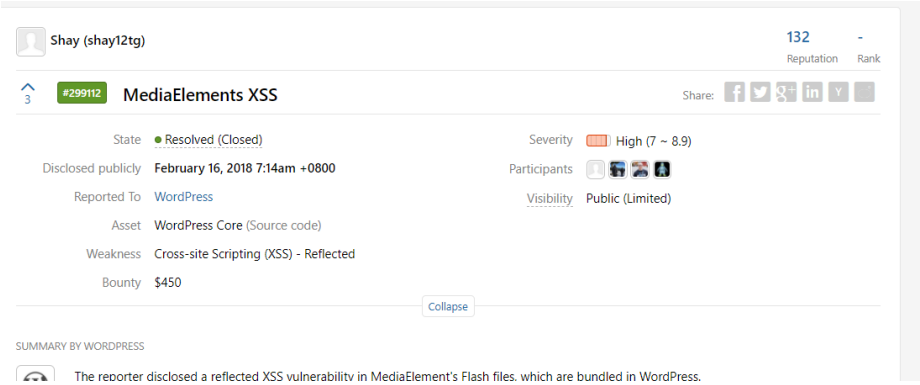
WordPress 4.9-4.9.1 Flash XSS

Yuyang Zhou

I. 背景

WordPress 数小时前在 Hackerone 平台上公开了一个等级为“高危”的漏洞报告：

https://hackerone.com/reports/299112



II. 漏洞分析

Wordpress 官方并没有公开此次漏洞报告的详情。不过通过报告标题和简短的描述可知

“

The reporter disclosed a reflected XSS vulnerability in MediaElement's Flash files, which are bundled in WordPress.

”

该漏洞是 Wordpress 使用的第三方组件 MediaElement 存在问题。因此判断该漏洞应该是一个 Flash XSS，查阅 MediaElement 项目在 Github 平台上近期的变动记录定位到涉及修复漏洞的更新记录如下：

https://github.com/mediaelement/mediaelement/commit/b9a6fde0cf856653cec968f6322bae8a89064a6c

更新涉及到的文件有两个 HlsMediaElement.as 和 DashMediaElement.as，更新的内容也比较简单就是添加了一个判断 URL 参数合法性的函数：

```
+		private function isIllegalQuerystring():Boolean {
+			var query:String = '';
+			var pos:Number = root.loaderInfo.url.indexOf('?') ;
+
+			if ( pos > -1 ) {
+			   query = root.loaderInfo.url.substring( pos );
+			   if ( ! /^\?\d+$/.test( query ) ) {
+			      return true;
+			   }
+			}
+
+			return false;
+		}
```

既然是在给参数做过滤，就更好定位了，查看这两个 as 文件的代码里哪里用到了参数应该就呼之欲出了，简单查找后发现获取参数的代码主要有三处：

```
//72 行开始
var flashVars: Object = LoaderInfo(this.root.loaderInfo).parameters;

// Use this for CDN
if (flashVars.allowScriptAccess == 'always') {
                    Security.allowDomain(['*']);
                    Security.allowInsecureDomain(['*']);
                }

                _id = flashVars.uid;
                _autoplay = (flashVars.autoplay == true);
```

allowScriptAccess 和 autoplay 内容固定，因此我们判定应该是 uid 这个参数出问题了，全局找一下_id 这个变量，调用这个变量的位置有两个：

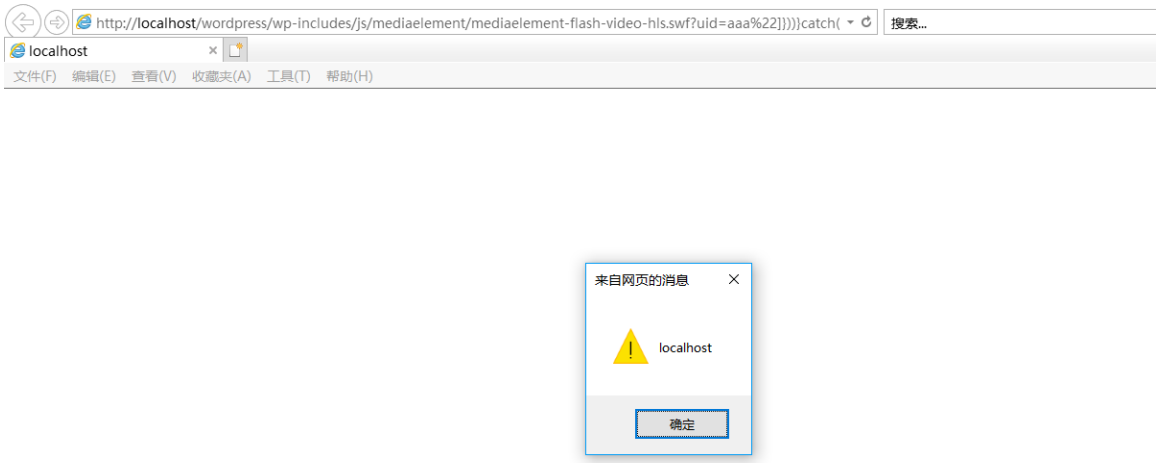
```
//139 行
ExternalInterface.call('(function(){window["__ready__' + _id + '"]()})()', null);

//433 行
        private function sendEvent(eventName: String): void {
```

```
                ExternalInterface.call('__event__' + _id, eventName);
        }
```

既然是参数没有做过滤，那么就应该是 ExternalInterface.call 造成 XSS 的第一种情况，我们很容易可以构造出 PoC，效果如下：

http://****.com/wp-includes/js/mediaelement/mediaelement-flash-video-hls.swf?uid=aaa%22]}))}catch(e){aa};alert(1);//&allowScriptAccess=always



HlsMediaElement.as 和 DashMediaElement.as 这两个 ActionScript 编译后涉及到的文件有两个：

mediaelement-flash-video-mdash.swf 和 mediaelement-flash-video-hls.swf

III. 影响范围

WordPress 4.9、4.9.1

**通过查阅 Wordpress 官网 https://cn.wordpress.org/releases/的历史版本，发现存在 mediaelement-flash-video-mdash.swf 和 mediaelement-flash-video-hls.swf 两个文件的版本目前仅有 4.9、4.9.1。因此认为，仅这两个版本受到此次 Flash XSS 的影响。*

IV. 风险等级

中危

V. 漏洞修复

1. 升级到 Wordpress 最新版本

2. 查找 Wordpress 目录下是否存在如下两个存在漏洞的文件：

[1] mediaelement-flash-video-hls.swf

MD5： B2DC69C327348B4774BCEFB6F8AA0408

http://****.com/wp-includes/js/mediaelement/mediaelement-flash-video-hls.swf?uid=aaa%22]}))}catch(e){aa};alert(document.domain);//&allowScriptAccess=always

[2] mediaelement-flash-video-mdash.swf

MD5：DC64796B1BB9F9A40F5F2F874E821266

http://****.com/wp-includes/js/mediaelement/mediaelement-flash-video-mdash.swf?uid=aaa%22]}))}catch(e){aa};alert(document.domain);//&allowScriptAccess=always

参考资料

[1] https://midzer0.github.io/2016/wordpress-4.5.1-xss/

[2] https://github.com/mediaelement/mediaelement/commits/master

[3] https://wizardforcel.gitbooks.io/xss-naxienian/content/14.html