

在IKEA.com中的本地文件包含

niexinming / 2019-01-18 09:42:00 / 浏览数 7965

翻译自: <https://medium.com/@jonathanbouman/local-file-inclusion-at-ikea-com-e695ed64d82f>

翻译: 聂心明

你想参加私有众测? 我很乐意邀请你, 请联系我Jonathan@Protozoan.nl

背景

通过本地文件包含攻击(LFI), 你可以拿到服务器中的敏感文件。比如, 配置文件, 日志文件和网站的源代码。有时, 你甚至会导致远程命令执行。本地文件包含一直被认为是危害性极高的漏洞。

大多数导致本地文件包含漏洞的原因是动态的加载图片或者其他文件。如果请求的文件或者路径没有被严格的验证, 那么攻击者就会获取服务器中的敏感数据。让我们学习一下关于它的知识。

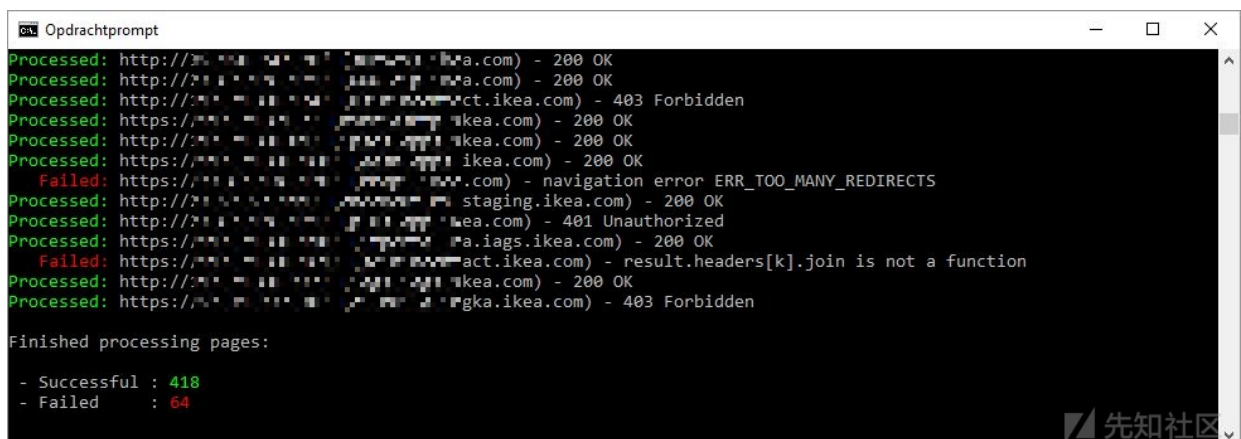
IKEA.com

宜家世界上最知名品牌之一, 它也是福布斯财富榜前50的企业之一。几乎每个人都会在家里有一件宜家的产品。我喜欢宜家的IKEA LATTJO brain 帽子。你们最喜欢宜家的什么产品呢? 请在下面的评论区里面评论。

对于我们来说, 它还有一个好处就是它有大量的各种类型的网站和app。这些网站和app都非常棒, 它们始终坚持服务着大量的客户。宜家有非常棒的[漏洞赏金平台](#), 宜家允许我们可以测试他们的网站, 并且我们可以在商家修复之后, 公开这些漏洞。只要我们遵守[漏洞披露规则](#), 就没有问题。来, 让我们渗透宜家吧。

寻找目标

在渗透开始之前, 我会先用Aquatone这个工具来探测目标的子域名。这个工具可以找到大量公共的子域名, 并且返回所有存活的网站, 下面的截图就是这个工具演示。关于Aquatone的使用, 你可以参考我的另外一篇文章《[Unrestricted File Upload at Apple.com](#)》



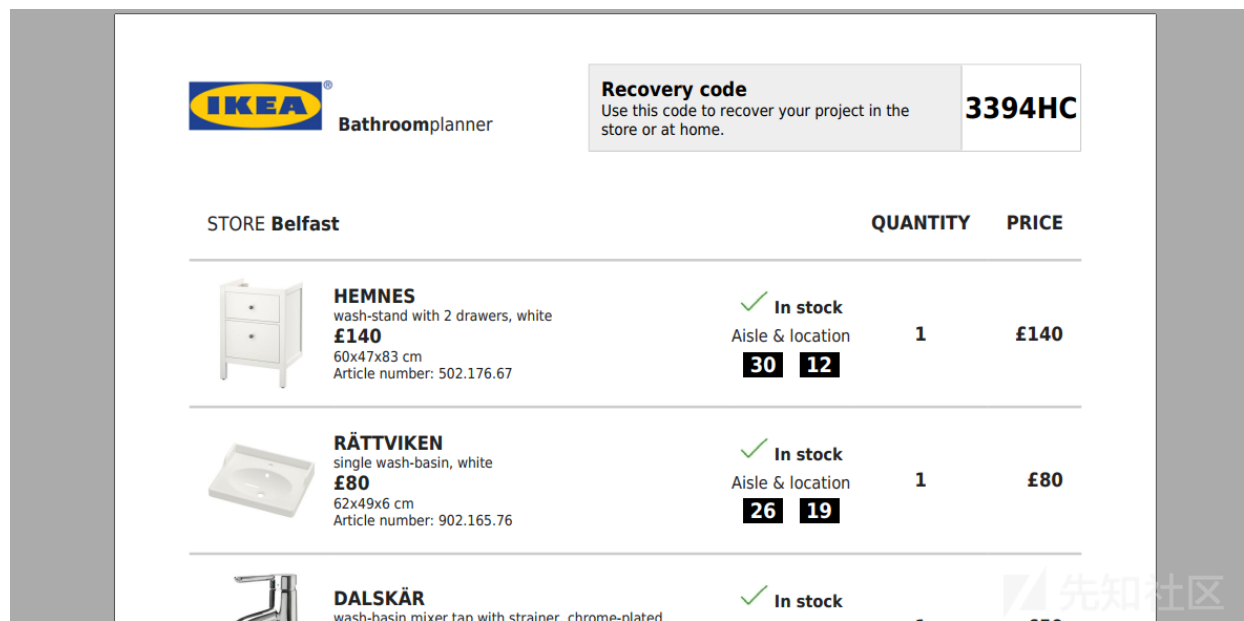
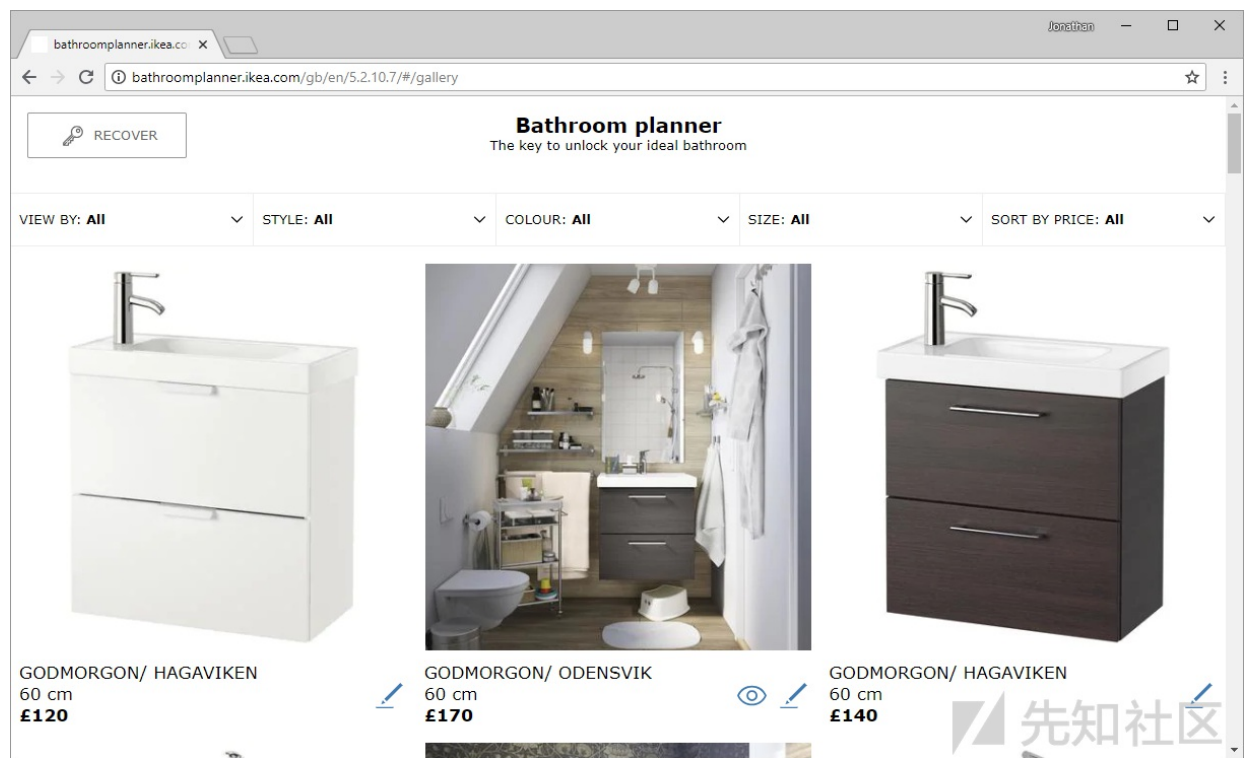
```
Opdrachtprompt
Processed: http://[redacted]ikea.com) - 200 OK
Processed: http://[redacted]ikea.com) - 200 OK
Processed: http://[redacted]ct.ikea.com) - 403 Forbidden
Processed: https://[redacted]ikea.com) - 200 OK
Processed: http://[redacted]ikea.com) - 200 OK
Processed: https://[redacted]ikea.com) - 200 OK
Failed: https://[redacted].com) - navigation error ERR_TOO_MANY_REDIRECTS
Processed: http://[redacted]staging.ikea.com) - 200 OK
Processed: https://[redacted]ikea.com) - 401 Unauthorized
Processed: https://[redacted]a.iags.ikea.com) - 200 OK
Failed: https://[redacted]act.ikea.com) - result.headers[k].join is not a function
Processed: http://[redacted]ikea.com) - 200 OK
Processed: https://[redacted]gka.ikea.com) - 403 Forbidden

Finished processing pages:
- Successful : 418
- Failed      : 64
```

宜家的Bathroom 子站

我们发现宜家的一个子站 [Bathroomplanner.IKEA.com](#),这是一个工具, 可以查找你想要的浴室家具产品, 并且把他们加入到购物列表里面。

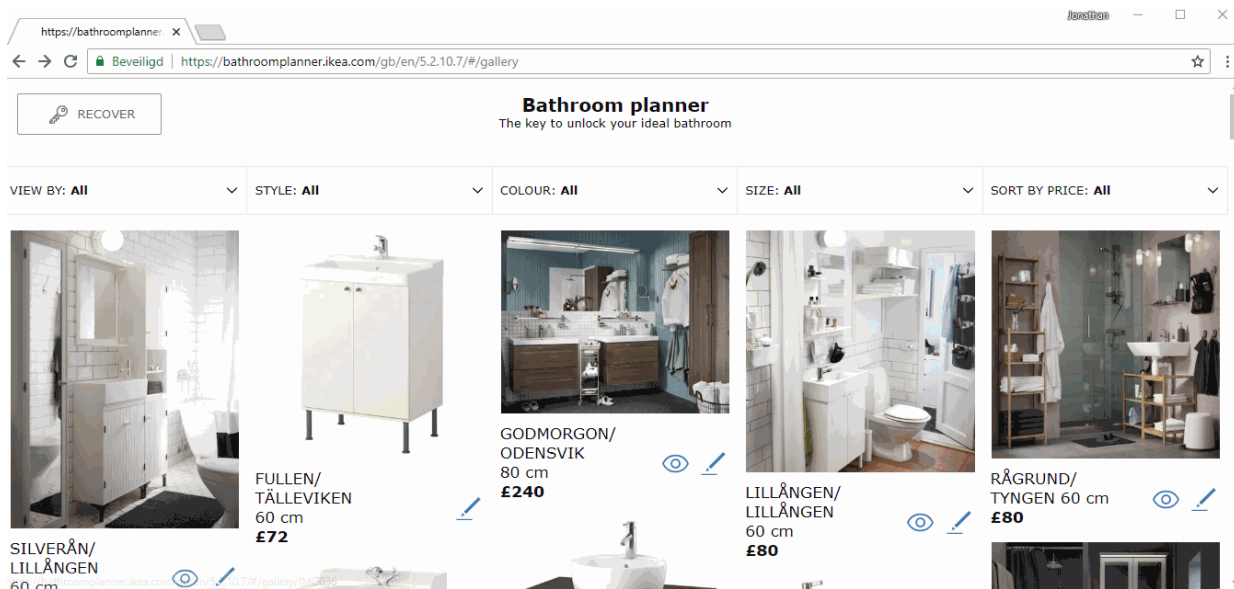
你可以让商家把产品列表以邮件的形式发给你，或者你可以把产品列表保存为pdf，这个pdf中包含文字和产品图片，其他的东西就没有了



那么这个pdf是怎样产生的呢？

抓个包

如果我说抓包的话，那么你肯定第一个想到的就是Burp Suite。现在我们打开Burp Suite，开始拦截发往宜家服务器的报文吧。我们首先打开一个产品页面，然后添加一个产品到我们的列表里面去。

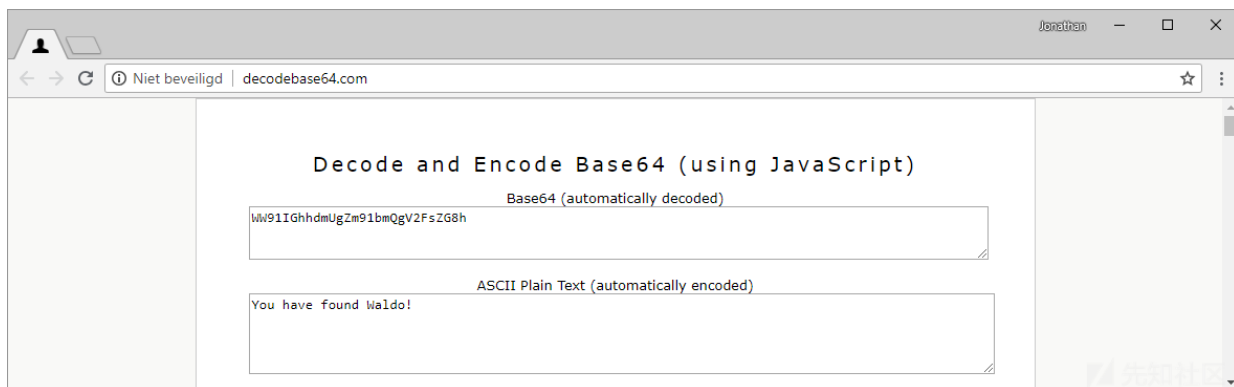


在上面的gif动画中，我们看到拦截的报文中有多参数：

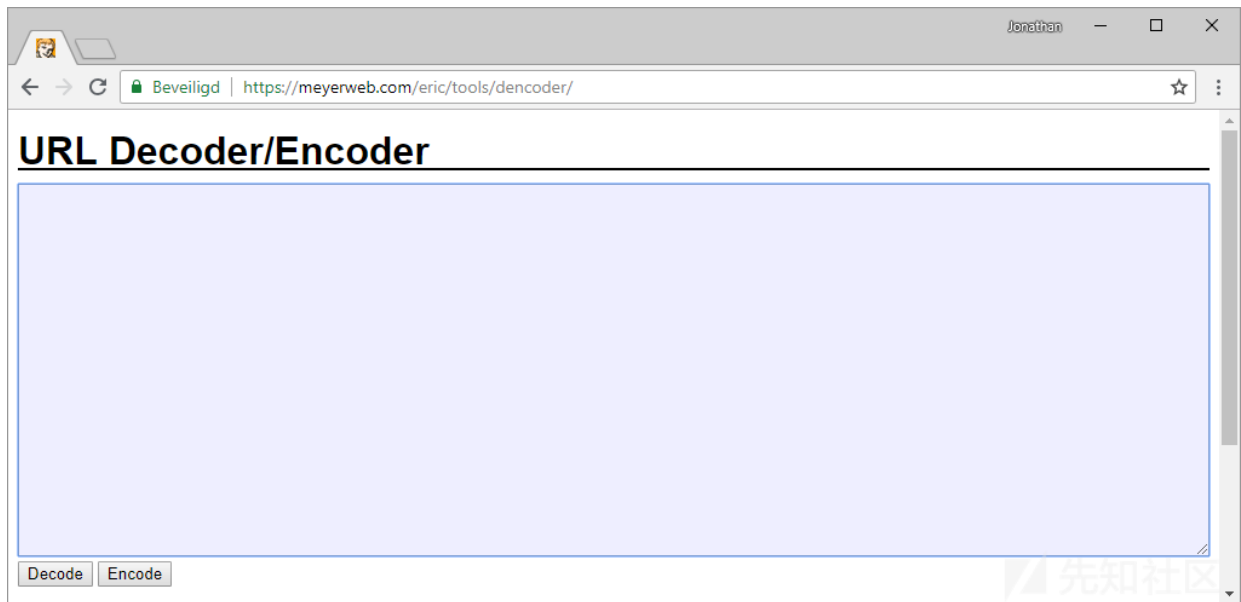
- data: 包含产品图片和产品的描述的json，里面没有路径信息。
- shopping: 包含我们产品列表的json，里面没有路径信息。
- pdf: 非常长的字符串，包含内容不清楚
- images: base64编码之后的图片

那一串长的字符串其实base64编码之后的数据

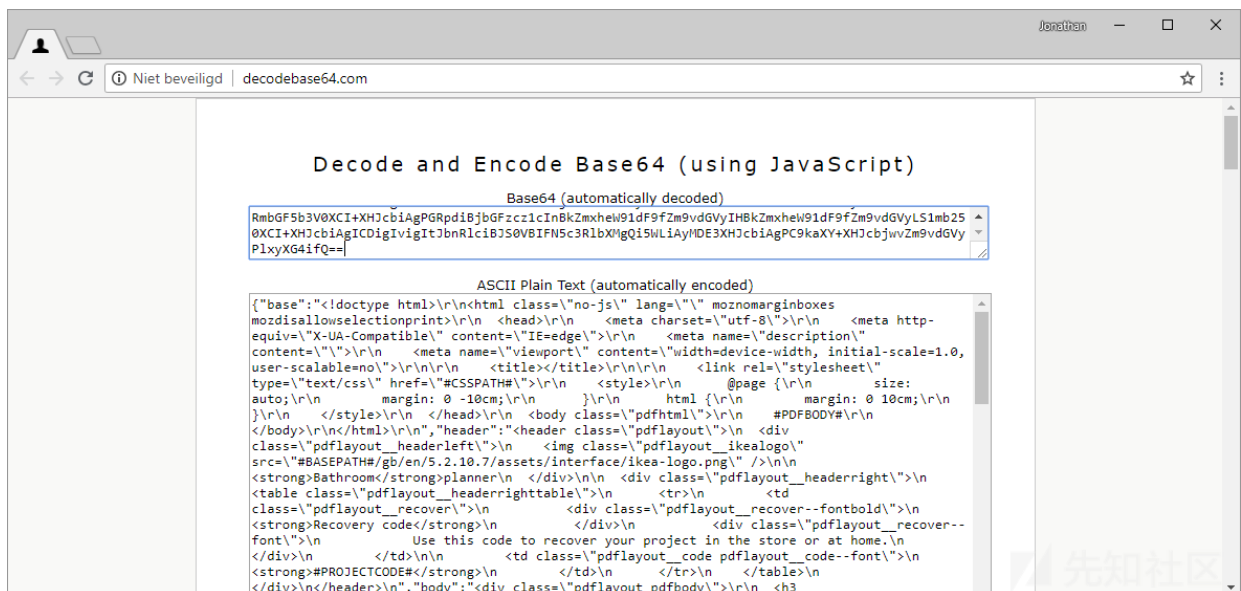
如果你遇到很长的数字和字母组成的字符串，那么你就看看这个是不是base64编码的字符串。Base64 经常用于文件传输。推荐给你们一个解base64的在线工具 <http://decodebase64.com/>



如果解码的页面报错，那么就是字符串中肯定包含错误的字符，比如：%，emmm，那么这一串字符串就有点像url编码的数据了，所以，我们首先应该url解码这些字符串，然后再把他们Base64解码。你可以使用 <https://meyerweb.com/eric/tools/dencoder/> 去解url编码。



我们首先将字符串进行url解码，然后再进行base64解码，就会得到下面的字符串。

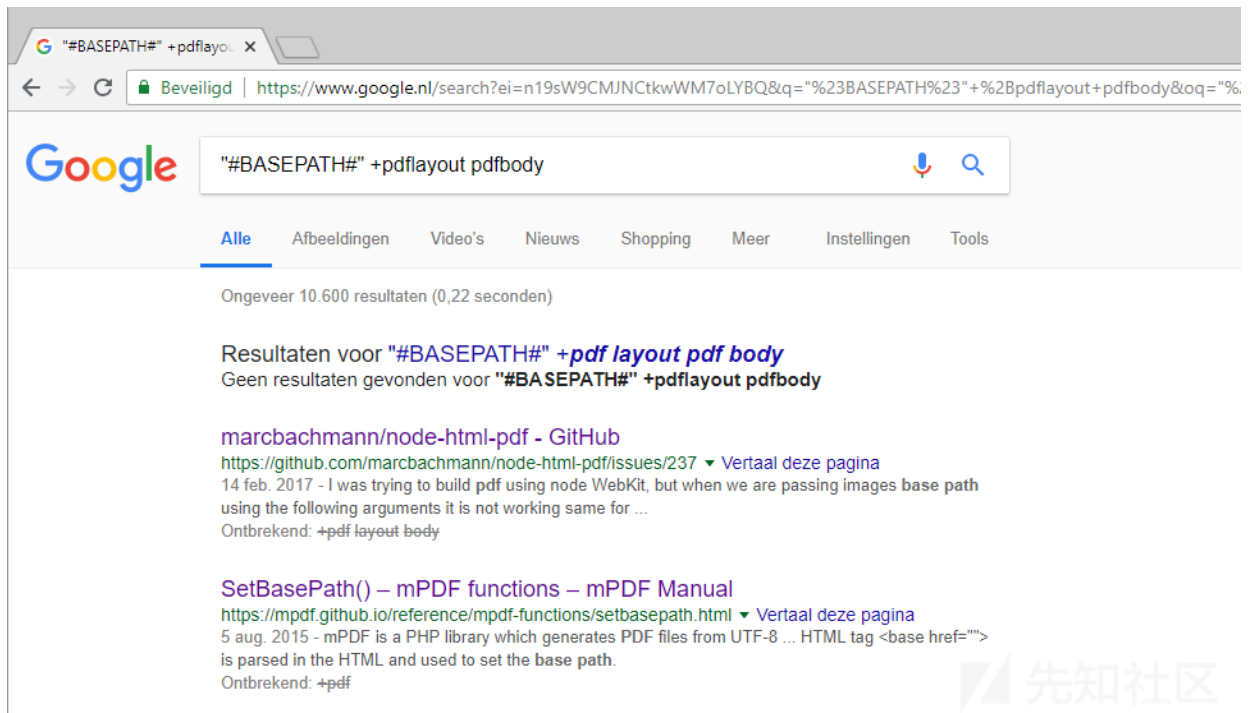


这看上去非常有趣，如果我们添加一些产品到我们的列表中时，宜家服务器就会用模板生成一个带产品清单的pdf文件。我们怎样才能包含服务器的本地文件呢？比如图片文件。好主意，让我们添加 `` 到模板文件中吧，然后把这个字符串先base64编码，然后再url编码，之后再放入数据报文的pdf参数中，最后再重放一下。

长话短说，这似乎不行。这个pdf解析器似乎不能识别这个文件。它也不能把这个字符串进行解析也不能输出内容。

B计划，识别解析pdf的库，用一些关键字来搜索这个库。

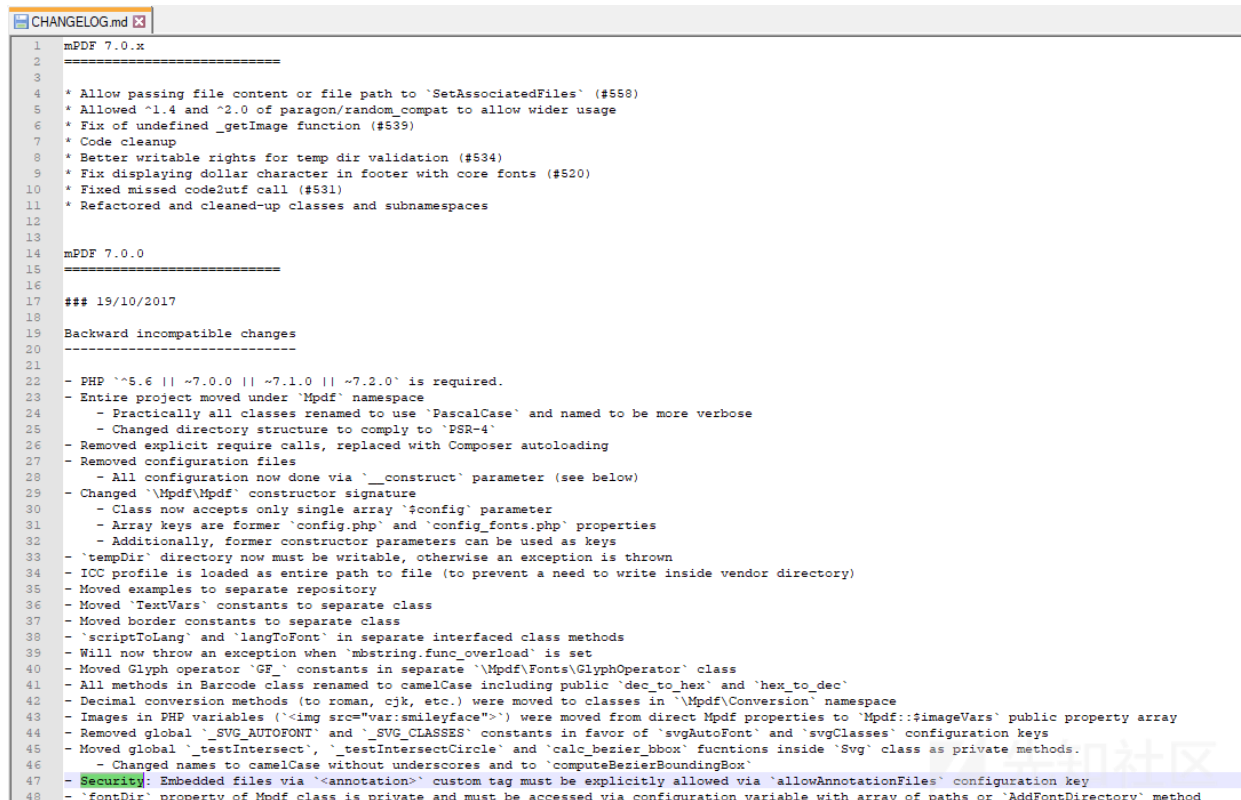
可能我们找到另一种方式在pdf中包含文件。我们需要一个强有力的工具去寻找这样的pdf解析库。谷歌是最好的选择。就用模板中出现的字符串进行搜索，看看谷歌会给我一个什么样的答案



我们得到两个选择，一个是node-html-pdf库另一个是 mPDF库。通过快速的阅读文档，我发现，这个项目中使用的pdf解析库是 mPDF

在mPDF中寻找安全相关的issues

我们马上就把 mPDF 下载到本地，目的寻找安全问题。其实最好的方式就是去看CHANGELOG，这个文件被程序员用来追踪产品之间的改变。



当我们读到h0ng10提交的issue的时候，发现在mPDF老的版本上有几个严重的安全漏洞，其中一个就是利用annotation标签进行本地文件包含。

如果我们仔细查看这个项目的提交记录就能发现这个commit。这个commit展示了漏洞所在的位置。

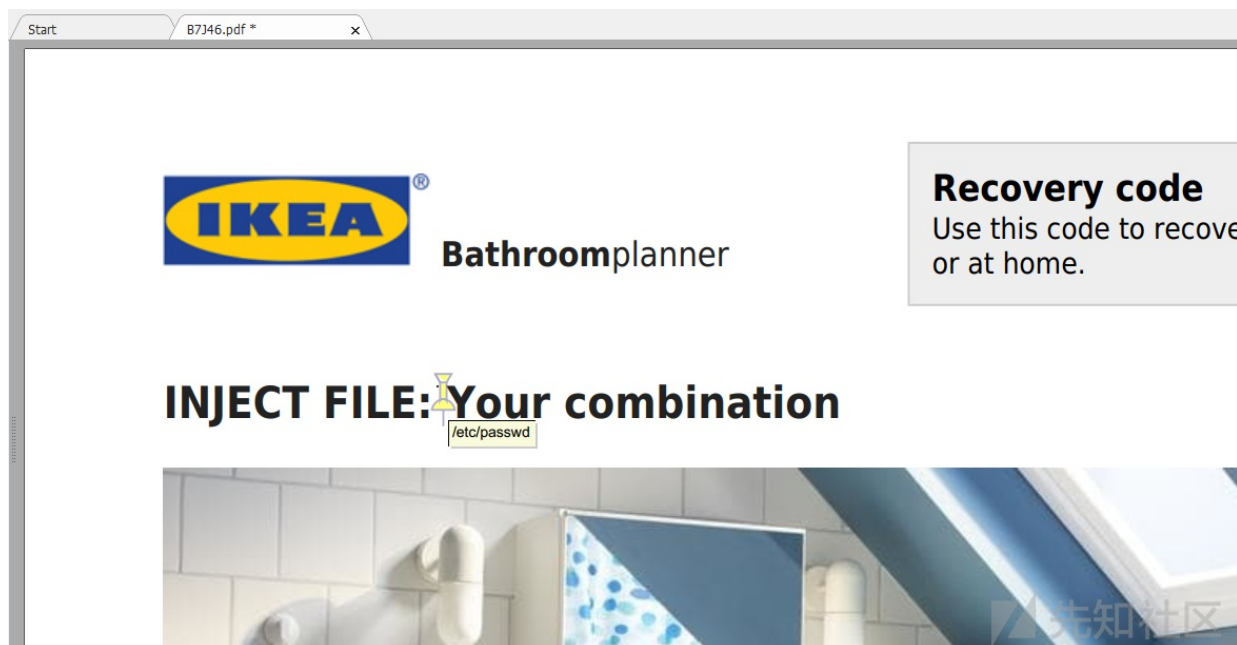
因为我们能够改变pdf的模板，所以我们就能在模板中添加一个能够包含文件的标签。完美，让我们看看宜家是否忘记把这个pdf解析库升级到了最新的版本。

发动攻击

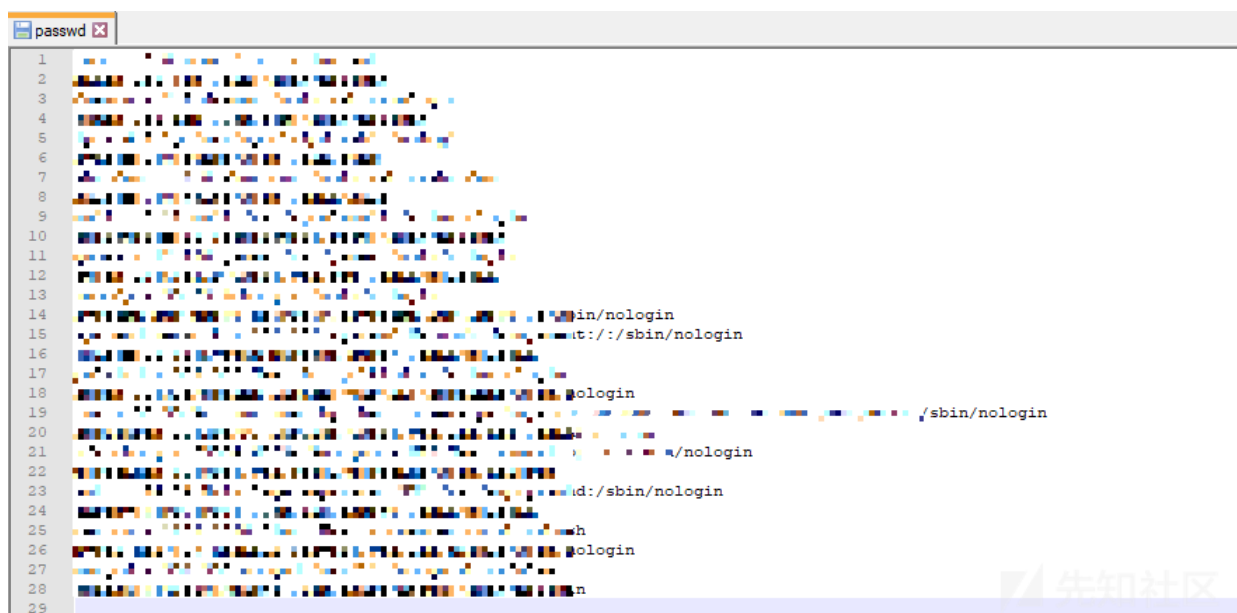
我们在模板中添加下面的标签:

```
<annotation file="/etc/passwd" content="/etc/passwd" icon="Graph" title="Attached File: /etc/passwd" pos-x="195" />
```

我们用Burp Suite中的Repeater 去发送一个新的模板，然后下载这个pdf文件。我们用 Foxit Reader去打开这个pdf文件。然后开始寻找那个黄色的小标签。



双击那个黄色的小标签就可以打开我们盗取的服务器文件了。是时候去提交这个漏洞并等待宜家的回应了。



结论

宜家允许用户操作pdf模板，然后用这个模板生成产品清单供人们下载。这个pdf库包含（抑或隐藏）了一个功能，这个功能就是在模板中添加一个指定标签去嵌入一个文件。在最新的pdf解析库里面，这个功能已经被禁用掉了，宜家没有及时的升级，所以导致

了这个漏洞。这个漏洞可以让攻击者在创建一个pdf的时候，包含服务器中/etc/passwd这个文件。

解决方案

- 不要让用户操作pdf的模板
- 在客户端生成包含产品列表的pdf，例如可以用 jsPDF
- 把mPDF升级到最新版

赏金

€250

讨论，什么是负责的披露

当你看到下面的关于漏洞披露的时间线时，整个流程确实花了很多的时间。我报告的大多数的漏洞没有任何争议。Bol.com 是一个非常好的例子。但是有时会花更多的时间。

这次我遇到了一个非常有趣的解决方案。宜家强制我用他们的赏金平台Zeroceptor来报告漏洞。但是Zeroceptor不允许我们在提交时拥有一个账户，所以我们不得不给Zeroceptor发邮件来让他们一起协同披露漏洞。我们在社区中没有直接联系到宜家。最开始宜家花了几周的时间才修复了这个漏洞，花了整整三个月和30封邮件才和他们合作披露了整个漏洞。



荷兰政府发布了一个政策（在2018年4月28日更新的：一个新版的手册），这个政策的其中几页解释了该如何负责的披露一个漏洞。其中安全研究员和原厂商的积极交流是其中的关键所在。

而且清晰的漏洞披露流程也是非常重要的。很少有厂商和研究员选择不去披露漏洞 (PDF, 7页, UPDATE 04-10-2018: PDF, 13页)

如果这个漏洞很难被解决，或者解决起来要花费很多的钱，那么披露者和厂商才选择不去公开这个漏洞。

这些天，大量的赏金平台被私有化，大量的披露者不去遵守这个规则。着是否符合公共利益呢？如果厂商没有及时的修复这个漏洞，消费者就会受到漏洞的影响（比如泄露数据）且消费者会毫不知情。难道是因为安全研究员透露了公司的名称而遭到律师的威胁吗？

任何国际标准建议中都没有关于非公开披露者的规则。并且，在2018年6月在欧洲举行的软件漏洞披露者大会上欧洲特别工作组就不建议不公开漏洞。相反的是，他们建议forbid政府签署非公开者协议。[见第84页](#)

如果赏金平台没有改变。那么安全研究员就会避开这些公司或者平台，他们会再一次公开这些漏洞。我们应该避免这些问题。

Bug Bounty是非常负责的平台，他们建议厂商要建立合理的披露流程。无论怎样他们也会保护公共利益和他们的安全研究员。向公众披露漏洞也是披露流程的一部分。

Hackerone是一个非常好的例子，他们的公开的漏洞是非常好的学习资源，与此同时hackerone会告知公众会有潜在的信息泄露的风险。

而其他的漏洞平台比如Bugcrowd 和 Zerocopter不会像hackerone那样公开披露漏洞。如果这些不公开漏洞的平台做出一些改变，向公众去公开这些漏洞而不是隐藏它们，那么整个互联网环境就会变的更好。

在更好的互联网世界中，私有赏金平台是不存在的。在某个时间点，所有的漏洞都必须被公开。如果报告的漏洞没有被修复，那么就必须要有很清晰的流程去让报告者知道下一步该怎么去做
并且如果安全研究者遵守这些规则，那么他们应该被保护。报告这些漏洞不应该有任何担忧。你怎么看这个问题，请在下面的评论区里面留言给我。

很高兴宜家能修复这个漏洞并且他们支持公开披露守则。非常感谢！

关注 | 1

点击收藏 | 1

上一篇： Mischief 靶机实战

下一篇： BurpSuite插件 - Hac...

1 条回复



d00ms

2019-01-18 14:46:54

之前有看过=><https://www.freebuf.com/articles/web/185321.html>

👍 0

回复Ta

登录 后跟帖