

SSRF in the Wild: A totally unscientific analysis of those SSRFs found in the wild

[mss**** / 2019-09-01 10:21:00 / 浏览数 3597 安全技术 WEB安全 顶\(0\) 踩\(0\)](#)

原文地址：<https://medium.com/swlh/ssrf-in-the-wild-e2c598900434>



先知社区

在本文中，我们将与读者一道，深入了解已公开披露的各种SSRF漏洞的详情，其中包括这些漏洞是在哪里找到的、漏洞的严重性以及供应商所采取的补救措施，等等。

Why I did this

=====

几个月前，我下定决心要深入学习一下SSRF漏洞方面的知识。

朋友们，你们是否体验过想要理解一个概念，却又无法完全掌握时的感受吗？好吧，我说的就是SSRF。后来，我想通了——搞定SSRF的最好方式，就是去野外寻找它，并亲

为了捕获这种类型的漏洞，我首先必须更深入地了解这些漏洞的运作机制、发生漏洞的原因以及出现漏洞的位置。

所以，我决定通读Hackerone网站上SSRF漏洞方面的所有安全报告，以便搞清楚：

- 1.黑客是如何找到SSRF漏洞的？
- 2.SSRF漏洞通常出现在哪些地方？
- 3.应用程序做错了什么事情才导致SSRF漏洞的？

Getting started

=====

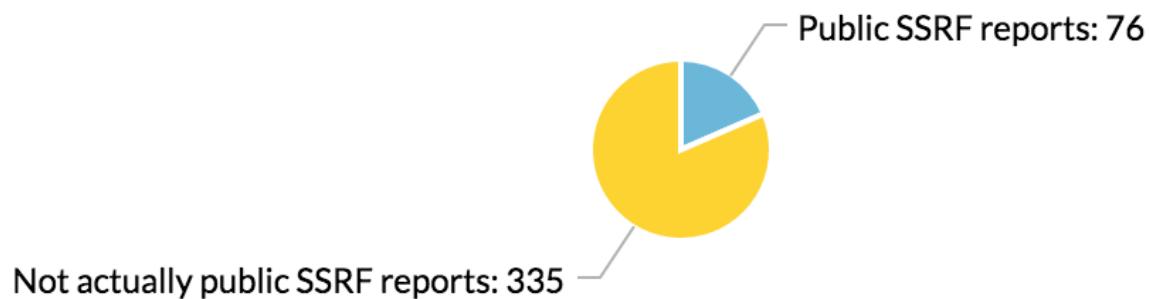
我的研究项目从两个简单的谷歌搜索开始——为了收集原始报告，可以使用下列方式：

```
site:hackerone.com inurl:/reports/ "ssrf"site:hackerone.com inurl:/reports/ "server-side request forgery"
```

在谷歌搜索页面中，返回了412个搜索结果。这是否表示有412份报告可供阅读？不幸的是，实际上并没有这么多。

首先，这两个谷歌搜索结果之间存在大量的重复结果。这是意料之中的。合并两个搜索结果集合的目的是确保获得最大的“有效”结果集合。这是因为，大多数包含术语“server-side request forgery”的报告中也包含术语“SSRF”。

此外，在过滤掉重复的结果之后，大量的结果链接指向访问受限的披露报告、与SSRF无关的报告或内部分享的报告。



谷歌返回的搜索结果

在过滤掉所有无用的报告后，现在只剩下76份公开披露的报告需要处理。那好，现在是时候通读这些报告了！

The research process

=====

阅读这些报告时，我主要关注下面几个问题：

- SSRF出现在哪些功能中？
- 报告漏洞之前，供应商是否采取了SSRF保护措施？
- SSRF的危害程度如何？借助这个特定的SSRF，攻击者可以发动哪些攻击？
- 收到漏洞报告后，供应商是如何补救的？

我通读了所有的报告，并根据上述标准对其进行分类。此外，我还考察了供应商实施的修复程序，以及修复之后是否出现了相应的绕过方法。此外，我还访问了所有曝出

[#6677](#) [#5912](#) [#6398](#) [#6145](#) [#6548](#) [#51011](#) [#5310](#) [#4190](#) [#4547](#) [#4881](#) [#5194](#) [#6657](#) [#5374](#) 先知社区

为期一周的时间内，我的浏览器标签页实际上一直处于这种状态

Results and analysis

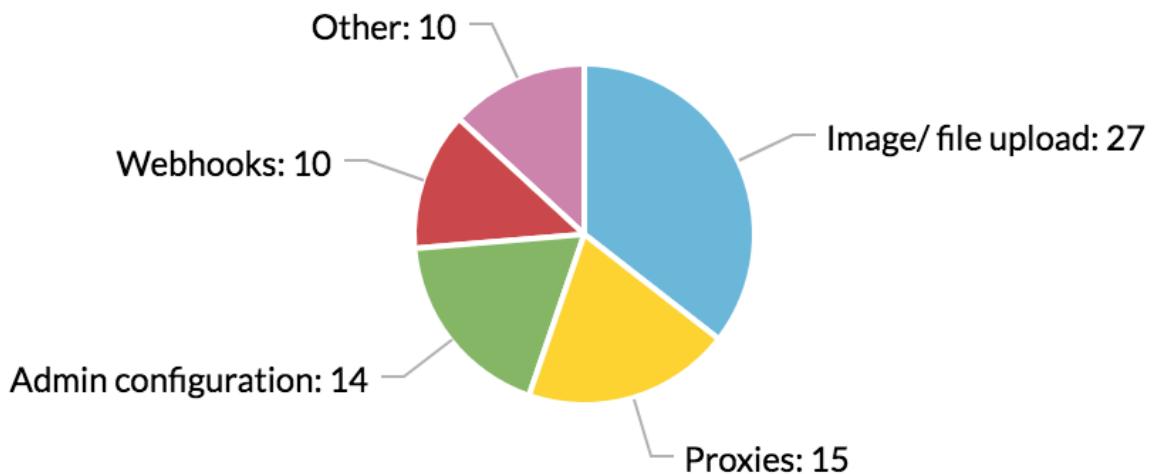
=====

废话少说，下面直接给出我从公开披露的报告中发现的结果。

Vulnerable feature

首先，这里根据SSRF漏洞所在的功能部件对76个报告中的SSRF漏洞进行了相应的细分。

其中，“管理配置”类型的SSRF漏洞主要是站点允许将设置导入为XML文件时由XXE所致。而“其他”类型的SSRF漏洞主要出现在包含接收非用于文件上传/代理/webhook用途的URL的功能部件中。



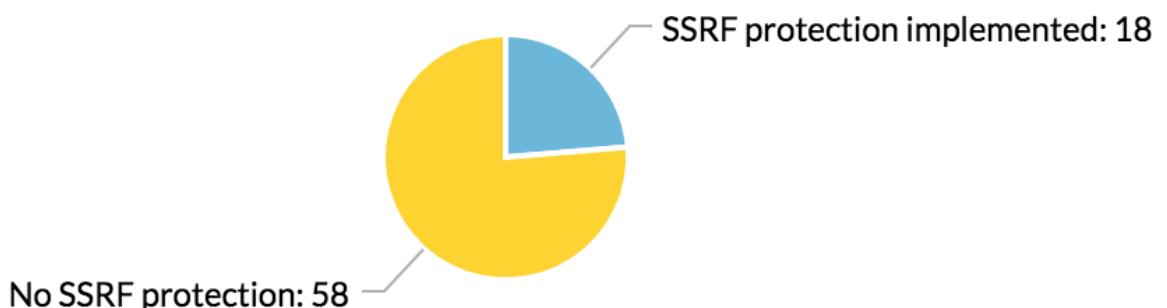
基于漏洞所在功能部件的分类。

如您所见，这些报告中的大多数SSRF漏洞都位于文件上传、代理或webhook服务中，所以，这些功能部件就是我们挖掘SSRF漏洞的首先位置。

值得注意的另一件事情是，各种可能导致SSRF漏洞的文件类型。任何可能包含由应用程序解析的URL的文件都可能触发该漏洞。在这些报告中，作为POC上传的文件的类型

SSRF protection before the report

然后，我查看了在提交漏洞报告之前，供应商实施SSRF保护措施的情况。例如，是SSRF保护措施被绕过了，还是根本没有部署SSRF保护措施？



漏洞曝出前，SSRF保护措施的实施情况。

令我非常惊讶的是，这些报告中，大多数的漏洞都是出现在没有采取SSRF保护措施的服务中。同时，这种类型的报告通常都是提交给那些安全措施比较到位的科技公司的，

据我猜测，这很可能是由于它们缺乏对SSRF漏洞原理以及出现该漏洞的常见位置方面的了解所致。例如，根据已经曝光的SSRF漏洞案例来看，它们通常并不是出现在应用程

Criticality of the SSRFs

根据已经报告的SSRF漏洞来看，大部分属于高危或中危漏洞。

然而，需要注意的是，大部分提交漏洞的研究人员并没有考虑如何提升该漏洞的危害性问题，相反，他们通常只是通过访问本地计算机上的已知端口来证明漏洞的存在性。这

Fix implemented by the vendor

在作者找到的漏洞报告中，有46份报告提交后，服务商提供的补救措施是实施黑名单（或者，如果该保护措施已经到位，则实施更多的黑名单）。另外，其中两份报告导致对于其他报告，供应商并没有披露自己所采取的补救措施。但最有可能的情况是，采用了某种黑名单或输入过滤措施来防御针对这种漏洞的黑客攻击。

Unexpected wins

=====

在研究这些报告的过程中，我对某些网站重新进行了相应的安全测试，并试图绕过提交漏洞后供应商所实施的保护措施。准确来说，重新测试的端点在25个左右。使用过去
至于我使用的绕过技术，大家可以参阅这篇[文章](#)。

此外，重新测试过程中，我还在某些网站上发现了一些其他类型的安全漏洞，主要是CSRF和信息泄漏漏洞。

Lessons learned

=====

Polish your SSRF-dar

在查找SSRF漏洞时，从文件上传URL、代理和webhook处下手是个不错的主意。此外，我们也要留意不太明显的SSRF入口点：嵌入在由应用程序处理的文件中的URL、接受

Escalate, escalate, escalate!

在阅读这些安全报告过程中，我一直感到惊讶的是，大多数研究人员根本没有尝试对这种漏洞进行改造升级，而是立即报告。在我看来，这种类型的漏洞的危害性还有待发

这里有一篇介绍以SSRF作为起点的漏洞链的[文章](#)，感兴趣的读者不妨阅读一下。

More public disclosure, please!

尽管这种漏洞方面的安全报告有很多，但我发现的大部分公开的报告实际上都是“有限披露”——这意味着只提供了报告标题和摘要。虽然许多报告的标题和摘要都很有趣，但

所以下次发现漏洞时，请大家发布完整的披露报告！

或者，如果供应商不同意完全披露，请努力撰写摘要，以便其他人能够从您的发现中获取知识。作为安全研究人员，我们将对此不胜感激！

Conclusion

=====

在这个研究过程中，我不仅玩得很开心，而且发现了一些安全漏洞。

完成这项研究后，我觉得自己对SSRF的漏洞有了更深入的理解，并在实际的漏洞挖掘过程中培养出了一定的直觉。最重要的是，希望本文能够对读者有所帮助！

点击收藏 | 0 关注 | 1

[上一篇：Google Bugbounty...](#) [下一篇：渗透测试：从XSLT注入到Gets...](#)

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](#)