IOS 的 WebView 自动拨号 bug

译者：赤(看雪 ID：呜呼哀哉)

原文链接：https://www.mulliner.org/blog/blosxom.cgi/security/ios_webview_auto_dialer.html

原文作者：未知

2016 年 十一月八日 星期二

摘要

IOS 的 WebView 可以被利用去控制手机自动的拨打一个被攻击者控制的电话号码，这种攻击可以在短时间内锁住手机的用户界面并阻止受害者取消呼叫行为。这个 bug 是一种 app 的 bug，貌似是由于严重的操作系统/框架错误导致。一个主要并且致命的问题是这个 bug 极易被利用。App 的开发者们必须尽快修复他们的代码，因为 Twitter 和 Linkedin 的 IOS 端 app 是极易受到攻击的(其他的 app 也可能受到攻击)。演示视频点击这里：Twitter 和 Linkedin（嵌入视频在这个页面的下方）。

正文

大概一周前（星期五）我读到一篇新闻报道【1,2】说一个人使用手机时打开了一个网页，结果出乎意料的，手机自动呼叫了 911，此人因为“调戏”911 而被逮捕。这最大的可能是因为一个手机的 URI（IMS 用户的身份标识）处理上的 bug【4,5】。我立刻想到了一个我在 2008 年十月报告给苹果的 bug。我不敢相信这个 bug 又重新出现了，所以我进行了调查研究。那篇文章说了一些有关在 Twitter 上发布的链接的事。

如果你认为在一个 app 中点击了一个链接然后手机自动拨打一个电话号码不是一个大新闻，再想想。且不说调戏 911 已经是件麻烦事了，还有其他的例子，像是如果拨打的是昂贵的 900 号码，攻击者可以因此而获得金钱；一个跟踪狂可以令他的受害者手机自动拨打他的号码，这样他就可以获得他的受害者的号码。以上所有的例子你都不会想看到它发生的。

无论如何…我去检查了 IOS 端的 Twitter 的 app。我在很短时间内就发现了一个非常简单的自动拨号工作器。我很开心同时也很崩溃因为这很容易。我一开始认为我准确的重现了那个的 bug，但在更加集中精神的重新阅读了那篇文章后，我确定了这可能是一个不同的 bug，或者至少是一个不同的触发器。那篇文章报道了负荷使用 JavaScript 和弹出窗口显示等问题，而我的**原始的**触发器是一行 HTML(一个指向 TEL URL 的中继刷新标记)，因此我决定在 HackerOne 上通过 Twitter 的 bug 赏金计划去上报这个 bug 给 Twitter。我之前从来没有上通过 bug 赏金计划上报过 bug，所以我对于能获得这些经验（我一般通过安全的@来上报 bug）感到十分开心。除此之外，现在是 2016 年，企业会为他们的 bugs 付出代价。所以我们会发现，Twitter 在短短几日内就承认了这个看起来是个问题。

在十一月六日，我给 Twitter 更新了那个 bug 报告，增加了一个锁定用户界面的问题（continue reading），并且上传了一个视频。今天 Twitter 就像是复制般仅仅是关闭了这个 bug，并没有任何的言论。尽管这可能只是一个复制行为，但是表现出了，他们对于做的更好应该是有兴趣的，也表现出了感谢给予那些上报了他们在业余时间所发现的 bugs 的人们。基于此，我决定今天公布这个问题的全部细节。

我在周末花了些时间去进一步研究这个问题。我认为这可能是一个使用 WebViews 来显示内容的 IOS apps 的普遍问题。我测试了一些我安装过的流行的 apps，那些易受攻击的 apps 需要一个方法来为使用者发布网页链接，且网页链接将被在 app 内部打开。而那些在手机的 Safari 或者 Chrome 浏览器中打开网页链接的 apps 则不易受到攻击（我测试了这一点）。Linkedin 是我测试的相当早的一个 app 因为 Linkedin 基本上是一个为商务环境提供的社交媒体软件。人们可以发送信息或者发布动态，动态通常是文本或者链接。我发布了一个链接然后点击它，然后没错，它拨打了我的另一个手机（演示视频在下方）。

我想给 Linkedin 提交这个 bug，然后发现他们有个 bug 赏金计划。不幸的是，那是个私人赏金，只有你在截止之前提交的 bugs 才会被添加进去。我试图绕过它，但是没用。经过一些思考后我决定不去以私人的方式上报给 Linkedin，而是将它公开（与这篇博客同时发布）。到现在，2016 年了，如果他们不想将我的建议加入他们的程序中，那是他们的自由。事实上如果私人 bug 赏金计划仍存在的话，我并不想通过博客公布 bugs 而非通过错误赏金计划。

又一个周末到来了，我有了些时间，于是开始再次与这个 bug 周旋。事实上当我试图去搞明白我是否将这个 bug 上报给了 Linkedin 时，我从我 2008 年的 PoC（Proof of Concept）开始浏览。在稍稍周旋了一会后，我或多或少的使我以前的 PoC 与 Twitter 和 Linkedin 的 apps 一起工作了。哇！

回到之前的事情上。我上报给 Twitter 的原始 bug 是通过访问网页来重新定向到一个 TEL URL，从而触发一个电话。一个人可以通过很多种技术来实现这个 bug，像：http 的元刷新，框架，设置文档，定位，获得当前界面，或者是一个 HTTP 的重定向(定位到首位)。这将轻松的使手机自动拨打一个号码。受害者会在屏幕上看到拨号器和目标号码，当然，还是可以通过按下那个大红按钮来取消掉呼叫。只是呼叫取消后会造成迷糊的人感到疑惑（为什么我的手机在拨打一些号码）。

我在 2008 年发现的 bug 的美妙之处在于我可以在短时间内锁定手机的用户界面，从而防止使用者取消呼叫。我试图去用我在 2008 年用过的相同的技巧来锁定用户界面。这个技

巧能造成操作系统在手机拨打指定号码的同时打开另一个 app 的情况。打开另一个 app 很简单,你打开一个 URL,导致操作系统运行另一个 app。这可能是任何来自信息 app（通过 SMS：URL）或者 iTunes（通过 itms-apps：URL）的 app。你几乎可以使任何有 URI 绑定的 app 启动。在 2008 年我试过用 SMS URL 和一串十分长的电话号码来阻塞用户界面线程。对于这个技巧是如何工作的,我最好的猜测是 IPC 子系统事实上很难将几个字节的 URL 数据通过各层进入 app,而目标 app 可能对于很大的 URLs 也不会感到很开心。我用下面的代码作为结束，这个代码使用组合元刷新标签和获得当前界面来完成攻击。这个代码会拖延设置获得当前界面 1.3 秒去保证拨号首先完成。这个拖延不能太长，否则这个 WebView 将不会执行这个 URL 去控制运行这个信息 app，所以基本上你不得不使时机刚刚好。

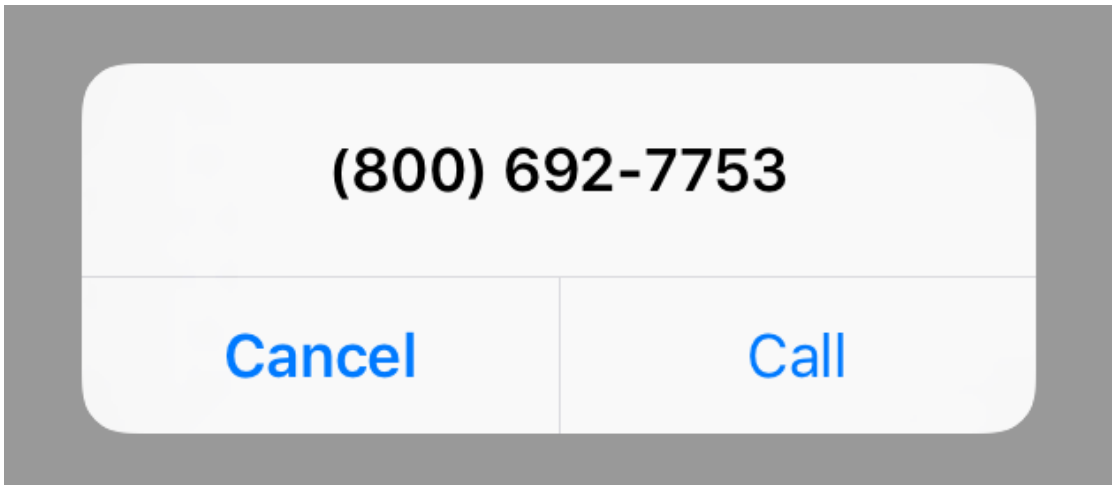
```
<html>
<head>
<title>iOS webview phone-auto-dial Exploit Demo by Collin Mulliner</title>
<!-- 800-692-7753 is the Apple support line I used my own phone number for testing! //-->
<meta http-equiv="refresh" content="1; URL=tel:8006927753">
</head>
<body>
<script lang=javascript>
l = "sms:";
for (i = 0; i < 10000; i++) {
    l = l + "3340948034298232";
}
function a() {
    window.location.href = l;
}
setTimeout("a()", 1300);
</script>
</body>
</html>
```

这个 PoC 是用来触发这个 bug 的。

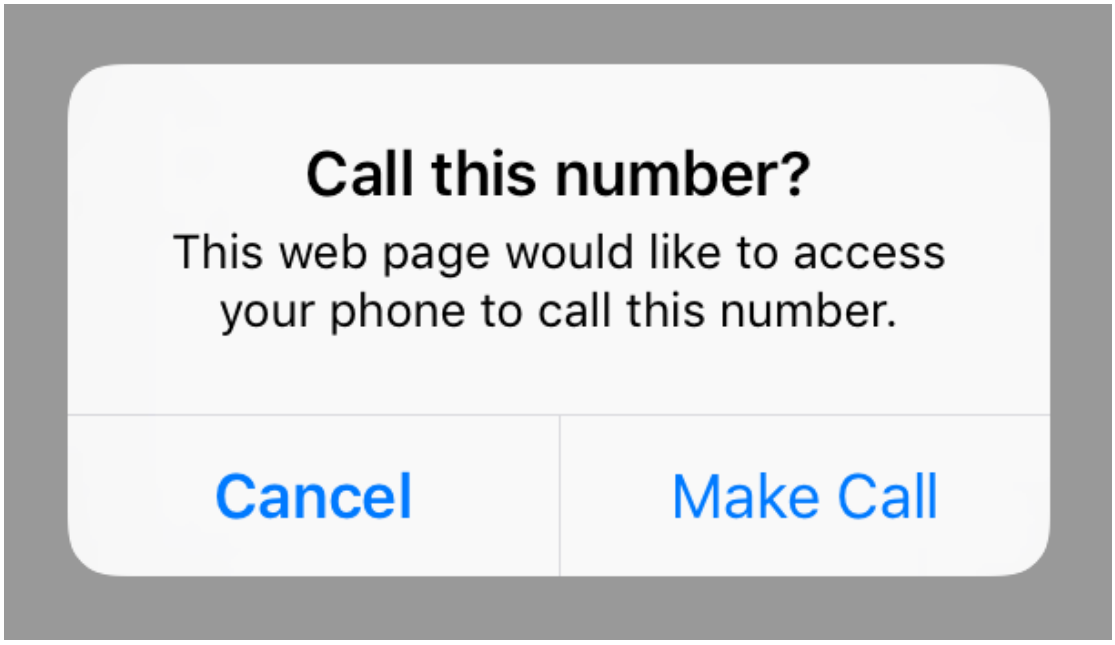
下面两个视频证明了这次攻击。你可以清楚地看到,用户界面在很短的时间内是不响应的。时间长到足够让另一边有人接起电话(尤其是服务热线自动接听)。

通常的好的 app 行为：

Apps 通常应该在执行前检查 URL 模式，在执行 app 在设备上前显示用户弹出对话框。一些例子如下所示:



手机上的 Safari 会在拨打苹果支持的号码前询问。这是好的 apps 应该有的表现。



Dropbox 显示一个警告而不是显示目标数，也不错但是可以变得更好。

Yelp app 通常表现的像 Safari 一样，但如果你用一个 HTTP 重定向它，它不显示目标数。我只是因为它有趣的地方才列入这个。

app 开发人员应该检查他们使用的 webview 去确定它们是否容易受到这种攻击。脆弱的 apps 需要被修复。像 Twitter 和 LinkedIn 一样的服务提供商可以检查发布到它们网站的链接是否包含恶意代码,防止这些链接被发布到它们的服务器。

在未来，苹果应该改变 WebView 的默认行为去排除 TEL URL 的执行，并使其显示明确的特性来避免这种问题。我已经把这个问题报告给了苹果。

引用：

[1] Bug Bounty Hunter Launches Accidental DDoS Attack on 911 Systems via iOS Bug (softpedia)

[2] iPhone hack that threatened emergency 911 system lands teen in jail (ars technica)

[3] 这是我在 2008 年十一月苹果修复了 IOS 3.0 中的这个 bug 后发布的充分的爆料: iPhone Safari phone-auto-dial vulnerability (original date: Nov. 2008)

[4] 原始的 TEL URI schema RFC2806 URLs for Telephone Calls

[5] 更新的 TEL URI schema RFC3966 The tel URI for Telephone Numbers