

iPhone的Wi-Fi芯片漏洞利用POC都公布了，赶紧更新系统吧

AngelaY 

2017-10-01 共397028人围观，发现 9 个不明物体

资讯

本周，谷歌 Project Zero 项目的研究员 Gal Beniamini 公布了 iPhone Wi-Fi 固件的漏洞利用 POC。这个漏洞（CVE-2017-11120）是个内存损坏（memory corruption）漏洞，存在于 iPhone 和其他苹果产品（Android 手机、Apple TV、Apple Watch 和其他智能 TV 等）所使用的 Broadcom 芯片中，影响 iOS 10 及更早的 iOS 版本。本周 iOS 11 版本发布后，漏洞才修复。



攻击详情

攻击者只需 iPhone 的 MAC 地址或网络端口 ID，就可以利用这个漏洞，在目标设备中执行恶意代码并建立后门，进而向固件发出远程读/写命令，轻松实现远程控制 Wi-Fi 芯片。入侵成功之后，攻击者“可以与后门进行交互，通过分别调用 ‘read_dword’ 和 ‘write_dword’ 功能获得对固件的读 / 写访问。”

Beniamini 发布的漏洞报告称：

该漏洞利用 iPhone 7 上的 Wi-Fi 固件进行代码执行，存档的密码是 “rrm_exploit”。这个漏洞利用已经针对 iOS 10.2（14C92）上的 Wi-Fi 固件进行了测试，但可以适用于包括 iOS 10.3.3 及以下的所有 iOS 版本，只是其中有些符号可能需要针对不同版本的 iOS 进行调整。此外，9.44.78.27.0.1.56 版本中的 BCM4355C0 芯片系统也存在这个漏洞。

此前其实也出现过类似的漏洞，都是通过本地 Wi-Fi 网络远程接管智能手机：

- 今年 4 月份，Gal Beniamini 在 Broadcom WiFi SoC (芯片软件) 中发现的漏洞；
- 今年夏天 Exodus Intelligence 研究员 Nitay Artenstein 披露的影响 Broadcom BCM43xx 系列 WiFi 芯片的 BroadPwn 关键远程代码执行漏洞 (CVE-2017-3544) ；

因为目前尚未有办法检测用户的设备是否在运行有漏洞的 BCM4355C0 版本固件，最好的办法还是将 iPhone 更新到没有漏洞的 iOS 11 版本。在最新的 tvOS 版本中，苹果也修复了这个漏洞。此外，本月初 Google 也在 Android 安全公告 2017-09-05 中修复 Nexus、Pixel 设备以及 Android 设备上解决了这个问题，不过安卓用户需要耐心等待手机厂商更新。



POC 重点及档案下载

6 月份 Beniamini 就已经发现并提交了这个漏洞，他在 Project Zero 网页中记录了这个问题：

Broadcom 固件中有一个典型的 RRM Neighbor 报告响应框架：

Category (5) Action (5) Dialog Token Neighbor Report Elements				
0	1	2	3	X

RRM Neighbor 报告响应框架

在固件版本为 9.44.78.27.0.1.56 的 BCM4355C0 SoC 上，RRM 相邻报告响应框架由 RAM 函数 0x1B0FE8 (代表 ROM 函数 0xABBBC) 处理。此函数主要可以验证对话令牌 (这是一个单字节字段，如果攻击者提前不知道也可以轻易暴力破解)。然后，该函数将 Neighbor

报告响应框架的内容复制到堆分配的缓冲区中，随后调用 0xAC0A8 的内部 ROM 功能，以存储每个给定的“操作类”（见9.4.2.37）的 Neighbor 数目。

以下是这个函数的近似高级逻辑：

```
int function_AC0A8(..., uint8_t* nrrep_buffer, ...)
{
    ...

    //Find and increment neighbor in given channel
    for given OP-Class

    int res = function_AC07C(..., nrrep_buffer, ...);

    //If there's no entry for the given OP-Class,
    create and populate it

    if (!res) {

        uint8_t* buffer = malloc(456);

        if ( !buffer ) {

            ...

        }

        else {

            buffer[4] = nrrep_buffer[16];
//Operational Class

            uint8_t channel_number = nrrep_buffer[17];
//Channel Number

            uint16_t* chan_neighbor_count_arr =
            (uint16_t*)(buffer + 6);

            chan_neighbor_count_arr[channel_number]++;

            ...

        }

    }

    ...
}
```

如上所述，该固件保存了缓冲区的链接列表，每个“操作类别”都有一个列表。每个缓冲区长为 456 字节，用于保存含有每个通道 Neighbor 数目的数组。输入条目的结构如下：

Next Pointer	Operational Channel	Padding	Neighbor Count Array
0	4	5	6 456

然而，由于“通道数目”字段未被验证，所以攻击者可以任意地提供较大的值。当最大允许通道数目为 0xE0 时，通过提供较大的值（如 0xFF），上述函数会将 16 位 word 增加到超出堆分配缓冲区的边界，从而执行 OOB 写入操作。请注意，内部函数 0xAC07C 中也存在相同的未验证问题。

在漏洞报告中，Beniamini 还分享了重要档案和漏洞利用步骤：

所附档案包含以下目录：

- hostapd-2.6：在exploit中使用的hostapd的修改版本。此版本的hostapd为配置可以支持802.11k RRM，尤其支持Neighbor报告。而且，这个版本的hostapd可用于添加各种命令，同时可实现整个漏洞利用过程中使用的动作框架的注入和接收；
- exploit：即exploit本身。

要实现漏洞利用，必须执行以下步骤：

- 将 SoftMAC 无线 dongle 连接到计算机并启用（如TL-WN722N）
- 编译提供的 hostapd 版本
- 修改“hostapd-2.6 / hostapd / hostapd.conf”下的“界面”设置，与你的界面名称相匹配；
- 在“exploit / conf.py”下方设置以下设置：
 - HOSTAPD_DIR：上述编译的 hostapd 二进制目录
 - TARGET_MAC：被入侵设备的 MAC 地址
 - AP_MAC：你的无线 dongle 的 MAC 地址
 - INTERFACE – 你的无线 dongle 界面的名称

- 通过运行 “exploit / assemble_backdoor.sh” 来组合后门 shellcode
- 运行 hostapd 以及上面提供的配置文件，广播 Wi-Fi 网络 (“test80211k”)
- 将目标设备连接到网络
- 运行 “exploit / attack.py”

按照上述步骤，可以安装简易后门，对固件进行读/写。还可以与后门进行交互，通过分别调用 “read_dword” 和 “write_dword” 功能来获得对固件的R / W访问。

感兴趣的读者可以点击[这里](#)查看 Gal Beniamini 发布的原文并下载相关文档：

链接: <https://pan.baidu.com/s/1cjOoLS> 密码: s482

*参考来源：[TheHackNews](#)，[Google Project Zero](#)，AngelaY 编译，转载请注明来自 FreeBuf.COM