Android OS 中通过 WiFi 广播泄漏敏感数据（CVE-2018-9489）

Stefano / 2018-09-17 00:32:00 / 浏览数 3272 安全技术 漏洞分析 顶(0) 踩(0)

原文地址：https://wwws.nightwatchcybersecurity.com/2018/08/29/sensitive-data-exposure-via-wifi-broadcasts-in-android-os-cve-2018-9489/

概要

Android OS的系统广播会将关于用户设备的信息向所有正在设备上运行的app公开。包括WIFI 网络名称，BSSID，本地 IP 地址，DNS 服务器信息，MAC 地址。其中一些信息（像 MAC 地址）在 Android 6及之后的版本不再通过
API提供，而其他的信息通常来说都访问权限。然而，通过监听广播，任何设备上的应用都可以获取到这些信息，直接绕过了权限检查和防范手段。

因为 MAC 地址不会变，而且是和硬件绑定的，因此即便使用 MAC 随机化，它还是可以用来唯一标识和追踪任何 Android设备。网络名称和 BSSID 可用于通过查找 BSSID 数据库（比如WiGLE和SkyHook）来定位用户的物理位置。其他的网络信息可以被恶意软件用于在本地 WIFI 网络中进行探索甚至进行攻击

所有设备上运行的所有Android版本都被认为会受到影响，包括一些分支（例如亚马逊的Kindle FireOS）。厂商（Google）修复了Android P / 9的 bug，但是并不打算修复旧版本的问题，GOOGLE鼓励用户更新到Android P / 9或更新的版本。GOOGLE
已指定CVE-2018-9489来跟踪此问题，并建议进一步研究以确定其是否会被在野利用。

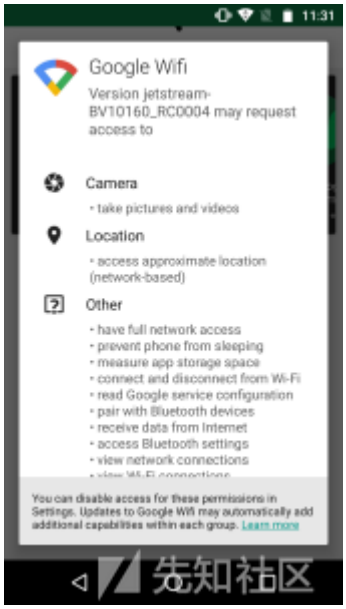
背景

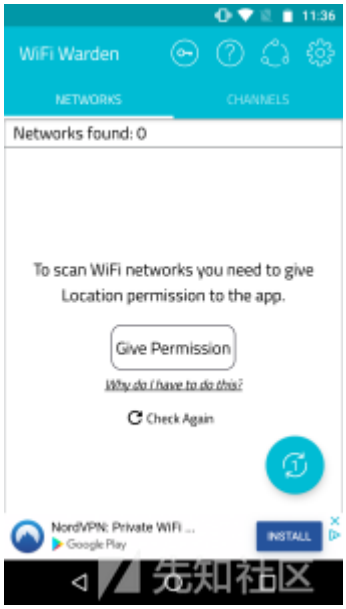
Android是一个由Google开发的、开源的、用于手机和平板的操作系统。据估计，全球大概有20亿设备运行 Android。Android 上的应用被系统分隔开，同时也和系统隔离。但是，通过某些机制仍然可以实现进程和/或OS之间的交互。

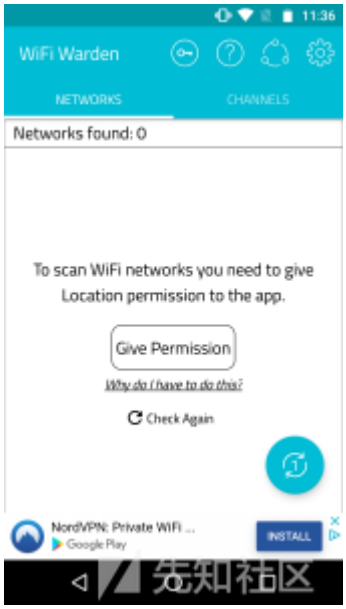
特别的，Android提供"Intents"作为进程间通信的的一种方式。使用"Intent"的广播允许应用或者操作系统在系统范围内发送可被其他应用接收到的消息。虽然存在限制使用 Android 应用的公共漏洞，恶意软件可以监测和截获其他应用广播的信息。

Android 的另一个安全机制是权限。这是为了保护用户隐私而设计的。应用程序必须通过应用程序清单（" AndroidManifest.xml "）中的特殊" uses-permission "标记明确请求访问某些信息或功能。根据权限的类型（"normal"，"dangerous"，
等等），系统可能会在用户安装程序时显示权限信息，也有可能在程序运行时再次提示。一些权限只能被系统使用，普通的开发者是不可使用的。

Google Play 和程序运行时的权限截图：







漏洞细节

Android OS使用两个 intent 定期广播关于 WIFI 连接和 WIFI
网络接口的信息：WifiManager的NETWORK_STATE_CHANGED_ACTION和WifiP2pManager的WIFI_P2P_THIS_DEVICE_CHANGED_ACTION。信息包括设备MAC地址，WIFI 接入点的BSSID和网络名称，以及各种网络信息，比如本地IP 地址范围网关 IP 和 DNS 服务器地址**。这些信息可以被任何运行在设备上的 app 使用。

虽然 app 也可以通过WifiManager获取这些信息，但通常需要在应用清单里添加"ACCESS_WIFI_STATE"权限。通过 WIFI
获取地理位置通常需要"ACCESS_FINE_LOCATION"或"ACCESS_COARSE_LOCATION"权限。在Android 6.0或更高版本中，不再通过API提供真实的 MAC地址，并且总是返回"02:00:00:00:00:00"。然而，一个监听系统广播的应用并不需要这些权限，因此可以在用户不知情的情况下获取这些信息，并且在Android 6或更高版本中获取到这些信息。

一个 APP 在 Android 7.0上尝试获取 MAC地址时的截图：

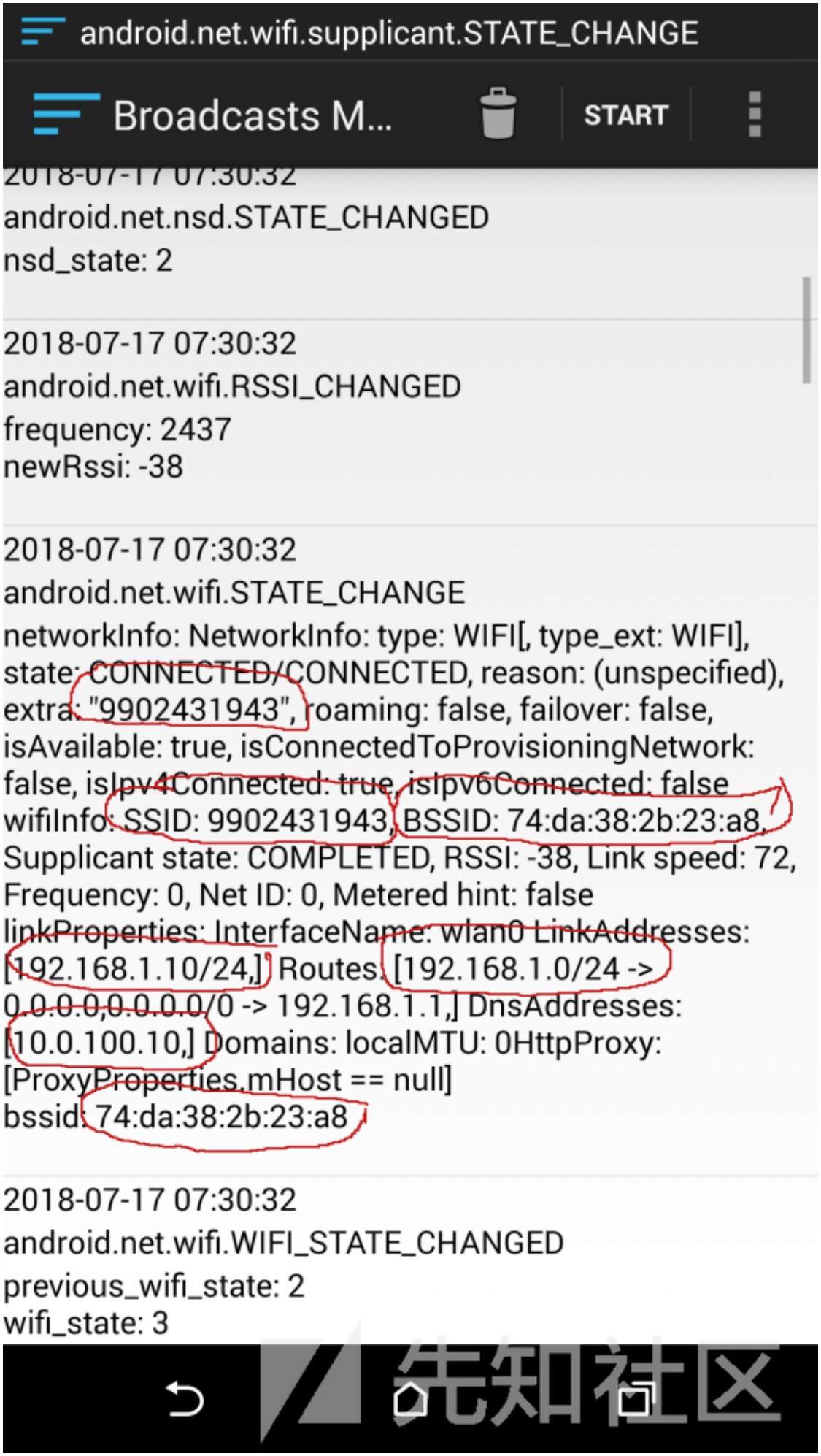
我们使用不同硬件和不同Android版本的设备在 test farm进行了实验。所有的硬件和所有的 Android 版本都确认了这个行为，即使有的设备不会在“NETWORK_STATE_CHANGED_ACTION”intent 中展示真实的 MAC 地址，但在“WIFI_P2P_THIS_DEVICE_CHANGED_ACTION”intent 中一样会展示出来。我们也测试了至少一个分支（Kindle 使用的Amazon FireOS），这些设备展示了同样的信息。

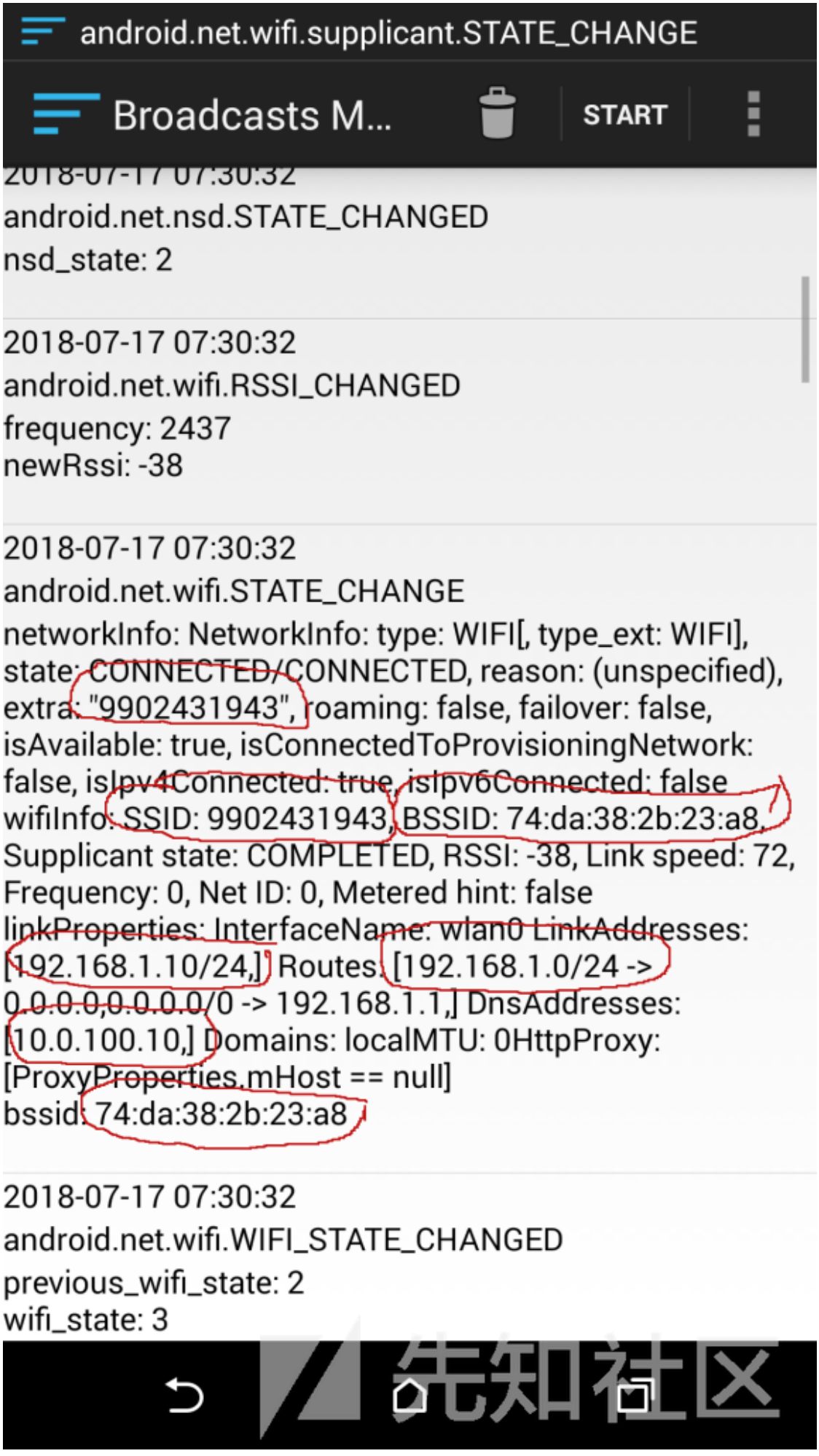
普通用户复现的步骤

对于 Android 设备用户，你可以通过以下步骤来复现：

1. 从 Google Play中安装由Vilius Kraujutis开发的Internal Broadcasts Monitor。
2. 打开app，点击“Start”来监控广播。
3. 注意系统广播，特别是“android.net.wifi.STATE_CHANGE” 和 “android.net.wifi.p2p.THIS_DEVICE_CHANGED”。

样例截图：


android.net.wifi.supplicant.STATE_CHANGE
Broadcasts M...
START
2018-07-17 07:30:32
android.net.nsd.STATE_CHANGED
nsd_state: 2
2018-07-17 07:30:32
android.net.wifi.RSSI_CHANGED
frequency: 2437
newRssi: -38
2018-07-17 07:30:32
android.net.wifi.STATE_CHANGE
networkInfo: NetworkInfo: type: WIFI[, type_ext: WIFI], state: CONNECTED/CONNECTED, reason: (unspecified), extra: "9902431943", roaming: false, failover: false, isAvailable: true, isConnectedToProvisioningNetwork: false, isIpv4Connected: true, isIpv6Connected: false
wifiInfo: SSID: 9902431943, BSSID: 74:da:38:2b:23:a8, Supplicant state: COMPLETED, RSSI: -38, Link speed: 72, Frequency: 0, Net ID: 0, Metered hint: false
linkProperties: InterfaceName: wlan0 LinkAddresses: [192.168.1.10/24,] Routes: [192.168.1.0/24 -> 0.0.0.0,0.0.0.0/0 -> 192.168.1.1,] DnsAddresses: [10.0.100.10,] Domains: localMTU: 0HttpProxy: [ProxyProperties.mHost == null]
bssid: 74:da:38:2b:23:a8
2018-07-17 07:30:32
android.net.wifi.WIFI_STATE_CHANGED
previous_wifi_state: 2
wifi_state: 3



开发者通过代码复现的步骤

为了使用代码复现，创建一个广播监听器，然后将其注册以监听这些活动：

1. “android.net.wifi.WifiManager.NETWORK_STATE_CHANGED_ACTION”
2. “android.net.wifi.WifiP2pManager.WIFI_P2P_THIS_DEVICE_CHANGED_ACTION”

样例代码如下：

```
public class MainActivity extends Activity {
@Override
public void onCreate(Bundle state) {
   IntentFilter filter = new IntentFilter();
   filter.addAction(
   android.net.wifi.WifiManager.NETWORK_STATE_CHANGED_ACTION);
   filter.addAction(
   android.net.wifi.WifiP2pManager.WIFI_P2P_THIS_DEVICE_CHANGED_ACTION);
   registerReceiver(receiver, filter);
}

BroadcastReceiver receiver = new BroadcastReceiver() {
@Override
public void onReceive(Context context, Intent intent) {
   Log.d(intent.toString());
   ….
}
};
```

厂商回应与缓解措施

厂商（Google）在 Android P/9 中修复了这些问题。因为修改 API 将会是一项重大的更新，所以厂商并不打算修补之前的 Android 版本。建议用户升级到 Android P/9 或更新版本。

亚马逊已就其Android分支（FireOS）做出如下回应：

> 我们计划在设备过渡到新版Fire OS时解决此问题

参考

Android ID # 77286245

CVE ID：CVE-2018-9489

Google Bug # 77236217

GitHub：内部广播监视器

致谢

我们要感谢Vilius Kraujutis开发内部广播监视器app并在GitHub中提供源代码。

该通报由Yakov Shafranovich撰写。

上一篇：TrendMicro CTF 20... 下一篇：利用不安全的跨源资源共享(CORS...

1. 0 条回复
   - 动动手指，沙发就是你的了！

目录