Security: WireGuard, Birds and Updates

By *Roy Schestowitz*
Created *18/10/2019 - 3:43pm*
Submitted by Roy Schestowitz on Friday 18th of October 2019 03:43:28 PM Filed under Security [1]

- **WireGuard Restored In Android's Google Play Store After Brief But Controversial Removal [2]**

  After Google dropped the open-source WireGuard app from their Play Store since it contained a donation link, the app has now been restored within Google's software store for Android users but without the donation option.

  The WireGuard app for Android makes it easy to setup the secure VPN tunnel software on mobile devices, similar to its port to iOS and other platforms. The WireGuard apps are free but have included a donation link to the WireGuard website should anyone wish to optionally make a donation to support the development of this very promising network tech.

- **Letting Birds scooters fly free [3]**

  At that point I had everything I need to write a simple app to unlock the scooters, and it worked! For about 2 minutes, at which point the network would notice that the scooter was unlocked when it should be locked and sent a lock command to force disable the scooter again. Ah well.

  So, what else could I do? The next thing I tried was just modifying some STM firmware and flashing it onto a board. It still booted, indicating that there was no sort of verified boot process. Remember what I mentioned about the throttle being hooked through the STM32's analogue to digital converters[3]? A bit of hacking later and I had a board that would appear to work normally, but about a minute after starting the ride would cut the throttle. Alternative options are left as an exercise for the reader.

  Finally, there was the component I hadn't really looked at yet. The Quectel modem actually contains its own application processor that runs Linux, making it significantly more powerful

than any of the chips actually running the scooter application[4]. The STM communicates with the modem over serial, sending it an AT command asking it to make an SSL connection to a remote endpoint. It then uses further AT commands to send data over this SSL connection, allowing it to talk to the internet without having any sort of IP stack. Figuring out just what was going over this connection was made slightly difficult by virtue of all the debug functionality having been ripped out of the STM's firmware, so in the end I took a more brute force approach - I identified the address of the function that sends data to the modem, hooked up OpenOCD to the SWD pins on the STM, ran OpenOCD's gdb stub, attached gdb, set a breakpoint for that function and then dumped the arguments being passed to that function. A couple of minutes later and I had a full transaction between the scooter and the remote.

The scooter authenticates against the remote endpoint by sending its serial number and IMEI. You need to send both, but the IMEI didn't seem to need to be associated with the serial number at all. New connections seemed to take precedence over existing connections, so it would be simple to just pretend to be every scooter and hijack all the connections, resulting in scooter unlock commands being sent to you rather than to the scooter or allowing someone to send fake GPS data and make it impossible for users to find scooters.

- **Security updates for Friday [4]**

  Security updates have been issued by Debian (poppler, sudo, and wordpress), Oracle (java-1.8.0-openjdk), Red Hat (java-1.8.0-openjdk), Scientific Linux (java-1.8.0-openjdk, java-11-openjdk, and kernel), and SUSE (kernel and postgresql10).

Security

Source URL: http://www.tuxmachines.org/node/129452

Links:
[1] http://www.tuxmachines.org/taxonomy/term/59
[2] https://www.phoronix.com/scan.php?page=news_item&px=WireGuard-Is-Back-Play-Store
[3] https://mjg59.dreamwidth.org/53258.html
[4] https://lwn.net/Articles/802622/rss