Agent-based simulation of Bluetooth LE Smart Privacy attack on scale

Python Mesa simulations to measure synthetic attack vectors in hardware-reliant large-scale exploits

Authors: Norman Luhrmann, V

daughter-of-v@protonmail.com / PGP key DF20D21B349C2A9A

Revision: 1.0 (2020-06-29) Initial draft

Foreword

In the course of research based on Bluetooth LE and global-scale mobile device infrastructure the authors have experienced problems in communicating a clear picture of a vulnerability impact. Agent-based simulation has proven to be a powerful artifact in the generation of data-driven reports. Real-world based models are able to capture the likelihood of different scenarios and provide plausible estimates for vulnerability spreads. A Jupyter notebook and Mesa based environment is developed to interactively explore a problem space this way.

Analysis

The vulnerability described in CVE-2020-13702 [1] enables an attacker to track the position of devices by abusing the high frequency of advertises sent in the unusual ExposureNotification Bluetooth LE protocol mechanism [2]. A specification frequency of around 300ms guarantees a high rate of discoverability. Even battery concerned real-world deployments of the API have shown a 3-5s advertise cycle, which do not evade discoverability the slightest.

The authors have identified the following approaches that assist in the avoidance of Bluetooth LE core privacy mechanisms (Smart Privacy [3]) or ExposureNotification encryption mechanisms [2]:

- Utilize dual identifier (randomized MAC, ExposureNotification RPI) to follow a one-sided rotation
- Temporally unique rotation ensures there is no potential collision of close rotation events

- Spatially unique rotation ensures there is no potential collision of nearby rotation events

Environment design

The Mesa [4] simulation environment is built according to the criteria provided in the Google/Apple specification [2] or real-world example traces taken on Samsung Galaxy S10 Android devices.

Google/Apple spec implementation

Google Android implementation

Google Android R implementation

Google Android R implementation fixed

Due to the temporal and spatial collision probability being related to the number and clustering of devices the environment is designed to assess a range of mobile device cluster sizes dynamically:

- The number of devices is chosen from the following list of Fibonacci numbers: 1, 2, 5, 8, 13, 21, 34
- Each device is allocated a simulation agent with it's unique timer instance
- Every simulation step all agents are triggered to move spatially and handle potential timer events
- Every simulation is run multiple times to identify trends

The timer implementations currently send exemplary data into a log container. A black box tracker retrofits the messages according to the rule set given above.

Spatial uniqueness of an event is simulated via agent movement. Agents move in a square grid, 1 box horizontally, vertically or diagonally each turn. If the target spot is already inhibited by an agent they switch locations.

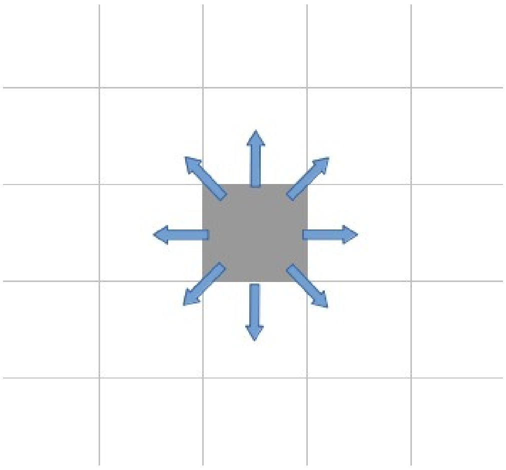

Fig.1: agent moves 1 box per turn

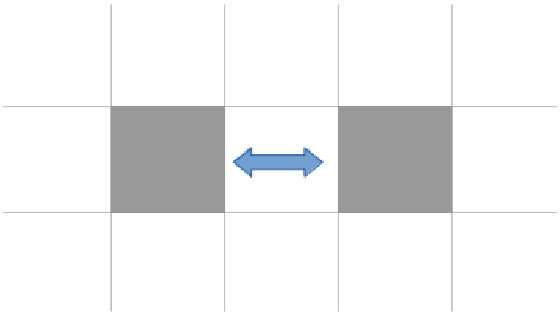

Fig.2: signal is spatially distinctive if there is one box in between

Simulation runs

The environment and device agents are simulated multiple times each with a growing cluster size of close by devices. The temporal resolution is 100ms so to capture the shortest timer specifications (~300 ms advertising interval).

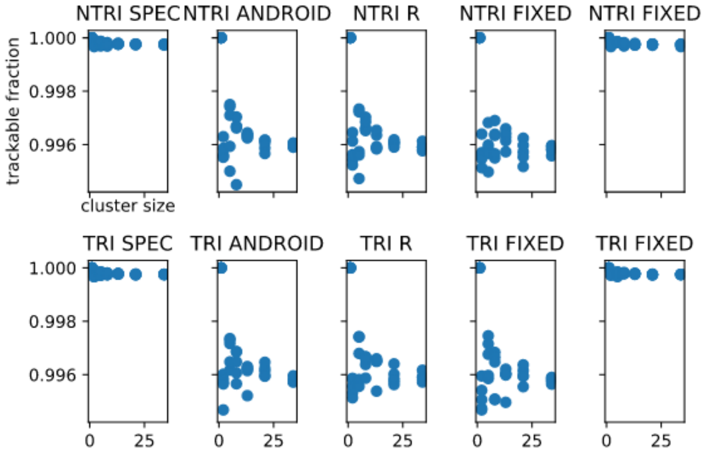

Fig.1: comparing different timer scenarios and triangulating vs non-triangulating beacon setups

Trackability is defined as the percentage of protocol protection elements that can be circumvented in the attack scenario. Very high 99% fractions are to be expected regardless of the concrete environment parameters. Clusters of devices are nearly unaffected by the tracking mechanisms up to usual social densities.

Conclusion

The picture drawn by the resulting numbers is bleak. Close to 100% tracking vulnerability inside core Bluetooth LE are to be expected.

The Bluetooth LE advertises contain enough information that through Bluetooth LE Smart Privacy and Google/Apple Exposure Notification encryption the protocol retains enough information that an attack is easily feasible.

Tracking is easily possible via multiple vectors with a very high likelihood of detection even in larger groups of people. So even indoor tracking densely populated areas seems feasible.

If ExposureNotification acceptance continues to grow this could lead to a future where a few key data holders can sell personal geo-location data on a market place. To the highest bidder, with up to GPS-like tracking accuracy.

Outlook

The black box tracker class implemented in this simulation could theoretically be directly fed with real-world beacon network log data to apply the concept to production purposes.

This simulation should be considered a proof-of-concept of what is possible with current means of technology and existing deployments.

The algorithms are efficient and scale easily, so potential damage is only limited by criminal/political motivation of an interested entity.

As the original CVE-2020-13702 [1] already highlighted it is not to be underestimated how the security context of Internet of Things/Botnets plays into this story, since beacon networks are not expected to follow stringent security means.

References

[1] https://nvd.nist.gov/vuln/detail/CVE-2020-13702

[2] https://covid19-static.cdn-apple.com/applications/covid19/current/static/contact-tracing/pdf/ExposureNotification-BluetoothSpecificationv1.2.pdf

[3] https://www.bluetooth.com/blog/bluetooth-technology-protecting-your-privacy/

[4] https://mesa.readthedocs.io/en/master/