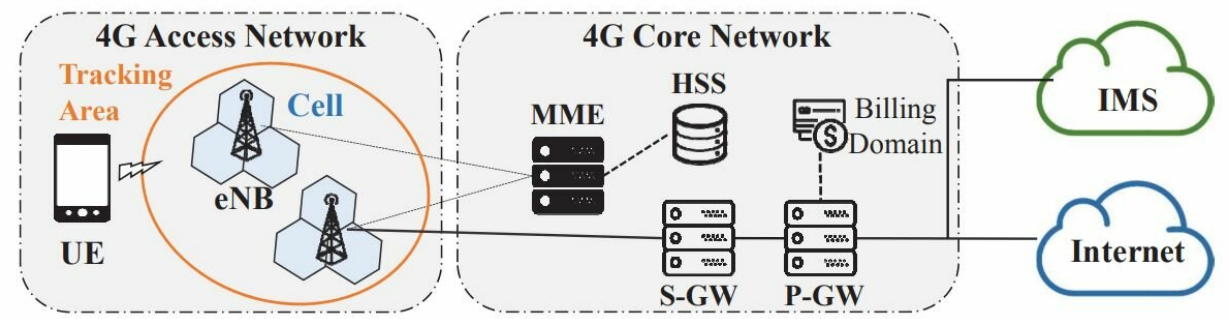
KAIST team used fuzzing to spot newer LTE protocol vulnerabilities

March 29 2019, by Nancy Cohen



LTE network architecture. Credit: Hongil Kim et al.

Researchers at the Korea Advanced Institute of Science and Technology (KAIST) discovered 36 vulnerabilities in 4G LTE wireless networks. Why the stir: Even though there is so much forward-looking talk about next-wave 5G, it is still 4G that is very much in use worldwide, by mobile networks and users.

LTE stands for Long Term Evolution, a standard for wireless broadband communication for mobile devices. A user-friendly expansion of what it is all about comes from T-Mobile, which tells its site visitors that LTE "allows you to download your favorite music, websites, and video really fast—much faster than you could with the previous technology."

The 4G wireless communications standard has picked up speeds of networks for devices such as phones, notebooks and tablets.

In the bigger picture, the KAIST team noted that mobile network operators are aggressively deploying LTE infrastructure; as of 2018, 600 carriers in 200 countries have deployed LTE networks, with over 3.2 billion subscribers worldwide.

As for North America, Caleb Chen in *Privacy News Online* let readers know that "LTE, or Long-Term Evolution, is the way that most smartphones are connected to the internet – and with 94% of mobile phones in North America connecting through LTE – the impacts of this new security finding are far reaching to say the least."

It's not even so much the words "flaws" or "vulnerabilities" that grabbed eyeballs over their findings but the numbers, considering there were 36 vulnerabilities found in the mobile networks explored. Actually, said Nicholas Fearn in *Computing*, they came upon 51 vulnerabilities but 15 had already been detailed, so the new ones totaled 36.

Two key attributes of this study are (1) the scale of the flaws identified and (2) the way in which the researchers found them, said Fearne.

Fearn said they used a technique called fuzzing. The authors wrote that they implemented "a semi-automated testing tool" dubbed LTEFuzz, "by using open-source LTE software over which the user has full control." LTEFuzz generates and sends test cases to a target network, and classifies problematic behavior by only monitoring the device-side logs.

The findings were categorized into five vulnerability types: Improper handling of (1) unprotected initial procedure, (2) crafted plain requests, (3) messages with invalid integrity protection, (4) replayed messages and (5) security procedure bypass.

So, what impact could an attack have as a result of vulnerabilities? Actually, the question should be what couldn't they do. The possibilities: "to either deny LTE services to legitimate users, spoof SMS messages, or eavesdrop/manipulate user data traffic," the researchers said.

The team alerted the relevant parties of the vulnerabilities that were newly discovered. The sleuths will not publicly release the LTEFuzz tool, as it can do damage in the wrong hands.

"After conducting the tests, we also responsibly disclosed our findings to the carriers and vendors to address any problems immediately. With regard to vulnerabilities attributed to specification defects, we are planning to contact the standard bodies soon."

In their study, the authors wrote that "We plan to privately release LTEFuzz to these carriers and vendors in the near future."

Pierluigi Paganini, security analyst, said that the flaws resided both "in design and implementation among the different carriers and device vendors."

The KAIST team's paper is titled "Touching the Untouchables: Dynamic Security Analysis of the LTE Control Plane." Reports said the paper would be presented in May at the IEEE Symposium on Security and Privacy.

Don't get it twisted, however; the team did not invent fuzzing; they rather applied it successfully to their research needs. Catalin Cimpanu in *ZDNet* provided a bit of history in looking at how they discovered the large number of flaws through fuzzing.

This, he said, is "a code testing method that inputs a large quantity of random data into an application and analyzes the output for

abnormalities, which, in turn, give developers a hint about the presence of possible bugs." While fuzzing has been used for years, the scenarios involved desktop and server software but "rarely for everything else."

More information: Touching the Untouchables: Dynamic Security Analysis of the LTE Control Plane, syssec.kaist.ac.kr/pub/2019/kim_sp_2019.pdf