Performing a Practical Paging Attack on the LTE Network

Nathan Yee

Advisor: Bruce DeBruhl

California Polytechnic State University, San Luis Obispo

I. INTRODUCTION

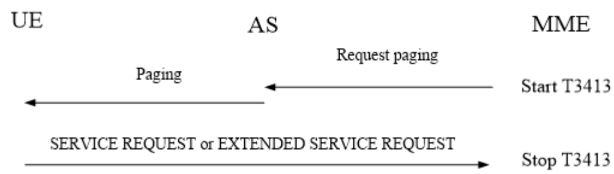

Fig. 1. Diagram showing how LTE performs a page

In this project I explore using a software defined radio to execute a paging attack on an LTE network. Paging is the process of a mobile network locating a device within a tracking area with broadcast messages or pages. Since these are broadcasted, all devices can see and decode these messages [1] [2]. The pages contain a devices globally unique temporary identifier (GUTI), which is used to locate a device within the tracking area.

Due to the lack of security around the GUTI, I show that an adversary with expertise and a software defined radio can capture these identifiers [1]. When these identifiers are captured, it is possible to identify an individual within a tracking area since the GUTI is unique. This is known as an LTE paging attack. The attack utilizes the GUTI value stored in an LTE page. An adversary would capture LTE pages over the air and link the GUTI values stored in the pages with a device that is in the same tracking area.

Tracking is another attack that can be executed using the GUTI values. Since GUTI values are not reassigned when the device changes mobile management entities (MME), an adversary can very easily track a moving individual [1]. This attack leverages the design that LTE only pages for a device within a tracking area. The MME knows that the device is in its tracking area and to locate the device it sends out a page [3] his design reveals that it is possible to execute a tracking attack if the device changes MMEs.

A practical implementation of this experiment has been performed before but only on the GSM network. Since LTE is generational iteration of the GSM network some of the techniques used in the GSM network still exist in LTE, and One of these techniques is paging. The GSM network also used a paging technique in order to locate devices but instead of using the GUTI identifier it would use the Temporary Mobile Subscriber Identity (TMSI) and in some case the International Mobile subscriber Identity (IMSI) [4]. I took the same implementation of the GSM experiment and applied it to the LTE network using a software defined radio.

This experiment allowed me to utilize a software defined radio (SDR). SDR devices are a new area of radio hardware technology where all the components of a hardware radio is implemented in software [5]. This makes this devices very versatile because it can be reprogrammed to perform any radio required tasks. In this case I had flashed a program on the SDR to use it as an LTE radio device.

The new SDR technology and software like srsLTE are important contributions that allowed for this practical attack. srsLTE turns the SDR into an LTE radio and has all the components to decode the paging messages. The evidence of previous practical implementations of this attack on the GSM network also contributed to the feasibility of performing this attack on LTE.

The following contributions where made as a result of the implementation:

- Implement a practical paging attack in LTE
- Utilize a software define radio to capture pages
- Tested 3 sources of LTE pages (phone calls, SMS messages, and WhatsApp)

I show that with the correct hardware and software a very simple identification and location attack can be performed using LTE pages. I also reveal that this attack is not limited to only phone calls and SMS messages, but messenger applications like WhatsApp can also be used in these attacks.

II. RELATED WORK

Related work within this subject has been limited to the GSM network. Since the LTE baseband implementation has just recently been released there has still been little research within this area. Although there has been limited LTE paging research the problems of mobile paging still exist going all the way back to the first generation of mobile communications.

Similar work to this project only centered around the GSM network on 2G [4]. The existing work focuses on the GPRS protocol that can also be known as 2.5G. When the GPRS protocol was implemented it also used a paging technique to locate devices within a tracking area, but instead of using temporary identifiers (TMSI), the GPRS protocol used the devices IMSI. The IMSI is unique to every device so it does not change. As a result, these identifiers can be collected and re-identification and tracking attacks can be executed on mobile subscribers.

Other forms of work in this area include the collection of GUTI identifiers in the LTE network. In the article, Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems, the authors show how the GUTI value can be a valuable source of information. The GUTI contains both a mobile subscriber's temporary identifier (m-TMSI) and the MME identifier that they are connected to. Given this information, an adversary has the ability to track an individual [1].

The majority of the work that has been done involves creating a rogue base station using an SDR. In this scenario, an adversary would create a fake cell tower that would operate on a common mobile frequency in its area. This is known as an IMSI catch or a rogue base station. This base station would act like a cell tower but provide no service, instead, it would collect IMSI's of devices nearby. With these IMSI's an adversary can carry out the same tracking attacks in the previous examples, but additionally, create a denial of service attack by sending a disconnect message to the device. In this scenario, the device will be in a disconnected state and will not be able to reconnect back into service until a restart is performed.

Some studies have shown that an adversary can implement a rogue base station and using the captured TMSIs or IMSI, impersonate the mobile subscriber [6]. The rogue base station would use the captured TMSIs and send messages out to subscribers using the TMSIs, impersonating the the subscriber's TMSI. Other work like the one implemented by Aragon et.al [7] show how a rogue GSM base station is able to eavesdrop on a GSM network.

III. Tools And Software

The project was to simulate a paging attack on LTE. The idea of the project is similar to ones that have been replicated on the GSM networks [4]. I wanted to be able to show that even though LTE uses temporary identifiers that it is still susceptible to identification attacks. Since most mobile networks do not rotate the GUTI values that are assigned to LTE devices [1], a tracking attack is also possible.

In this experiment, an Ettus B200 mini SDR [8] with a 900Mhz Omni-directional antenna was used with srsLTE [9]. srsLTE is an open source baseband LTE implementation. This means that it is able to mimic an LTE device and allows for users to sniff LTE pages and setup rogue base to test LTE networks. In this case, srsLTE was used for its Physical Downlink Shared Channel (PDSCH) sniffing capabilities.

The Physical Downlink Shared Channel is the channel that LTE uses to transmit pages. When the MME makes a paging request it is sent to all the LTE towers in its tracking area. The towers then broadcasts the page on the PDSCH.

srsLTE was modified so that it would capture and decode the pages, this makes it easier to search for a specific GUTI, in this case, the test device. Another modification that was made to srsLTE was combining it with and ASN1 library so that when the pages were captured they could be decoded to a human readable format. The ASN1 library that was used was ASN1C [10].

The test device is a rooted Nexus 5X. The device was rooted so that applications could be installed in the device to make sniffing identification easier. LTE Discovery [11] and TMSI catcher detector called AIMSICD [12]. With these two applications installed I was able to determine the devices EARFCN which is the radio-frequency channel number the device is connected to.

Fig. 2. Ettus B200 Mini and Nexus 5X

The test was performed in an open environment so that it was possible to see other GUTI identifiers. This allowed me to confirm that srsLTE and the SDR were in fact operating correctly and receiving LTE pages. Since srsLTE is required to listen to a specific cellular tower at a given frequency, performing this experiment in a Faraday cage would not be possible.

IV. Design

Due to the lack of page decoding in srsLTE, I combined it with an ASN.1 library in order to interpret the pages that srsLTE was capturing. In order to achieve this srsLTE was combined with asn1c, C implementation of the ASN.1 description language. Once these two libraries are combined the pages will now be displayed in a human readable format. This is pictured in Figure 3.

The first step was to determine what frequency the device was transmitting on. srsLTE has binaries for searching for a specific device on a specific LTE band. Figure 4 shows how srsLTE searches for devices on a given band. Once the frequency has been identified, srsLTE is then set to listen on the Physical Downlink Shared Channel (PDSCH) channel for paging messages.

In order to match the frequency of the scanned device to our Nexus 5X, I used the LTE Discovery app to match the cell

Fig. 3. srsLTE sniffing on PDSCH with asn1c implemented

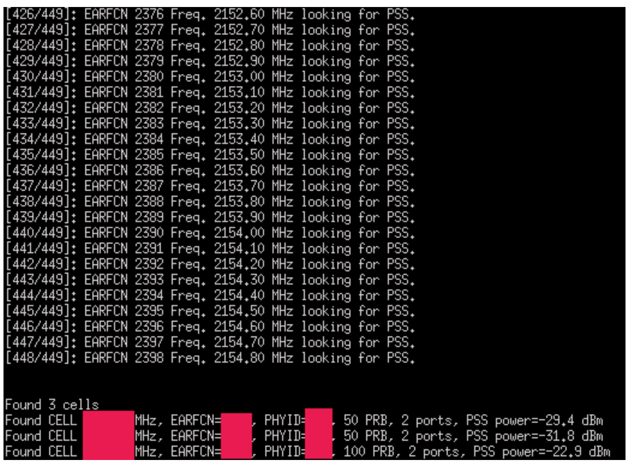

Fig. 4. Results of srsLTE searching for devices

ids. All LTE devices are given a cell id [13]. The cell search binary in srsLTE will show a devices cell id and the frequency it is transmitting on if it is found and using LTE Discovery I was able to determine the LTE frequency the Nexus 5X was on.

One of the difficult tasks of performing a paging attack in an experimental environment similar to the one that I have created is determining the GUTI value of the test device. The M-TMSI, the device identification that is part of the GUTI is stored in the SIM card [14]. But due to the different types of sim cards and the lack of documentation if the memory structure of them, it is almost impossible to retrieve these values without specialized hardware.

Due to the lack of information on how to extract the GUTI from a device, I performed multiple sniffing sessions. During the sniffing sessions, I would call test device on each sniffing session. Once multiple sniffing sessions were complete, they are compared to determine which GUTI value has appeared across all sessions. Once a list of GUTI identifiers has been determined, another sniffing session is performed, but this time I grep for each of the GUTI values while placing calls to the test device. If the GUTI belongs to the test device, then the GUTI identifier would appear in srsLTEs output.

This method of determining the GUTI of a device is potentially inaccurate, but since there is no known way of extracting this value from the device, this is the only way I was able to get a close approximation. At this point, if the correct GUTI value was found then srsLTE will display the GUTI value while it is sniffing for pages.

Other forms of communication that can trigger a page include WhatsApp and SMS messages. After identifying a possible GUTI for the test device, the same experiment was performed using WhatsApp and SMS messages. With SMS messages the results were very similar to the phone call. Once the device received the SMS message, the GUTI would appear in srsLTE.

WhatsApp has a very particular feature and that is during a conversation an individual is able to see when the sender is typing. This typing notification is also known for triggering pages [1]. In this experiment I was able to show that the typing indication does in fact trigger a page without having to send a message. Once WhatsApp sends the message, a page is sent to the receiver.

All experiments run using the phone call, SMS Message, and WhatsApp was under the assumption that the attacker knows the mobile subscriber. This assumption needs to made since in order for this attack to be executed, the attacker would need to know the phone number of the subscriber in order to trigger the page.

V. Results

In this section I will describe the results of the experiments that were performed on the test device. The 3 forms of communication that were tested was a call, SMS messages, and WhatsApp. One anomaly that was discovered for each of the tests was a noticeable delay between the communication and GUTI from srsLTE. The delay could be caused by srsLTE buffering the pages that it receives but that is outside of the scope of this experiment.

A. Phone Calls

The phone call was the first test that was performed. Since a phone call is known sending pages in previous generations of mobile communications, it was used to confirm whether a GUTI was associated with the test device. The results of this experiment did show that a phone call does trigger an LTE page when a call is placed to the test device. Once srsLTE had started listening for pages, multiple phone calls were made to the test device. The test devices GUTI would appear in srsLTE after said delay. All phone calls to the test device were accounted for.

B. SMS Message

The results from the SMS message was similar to the phone call. Similar to previous generations of mobile communications, the SMS messages are also expected to trigger a page when a device is receiving an SMS message. The same results also appeared in this test. srsLTE is able to capture the GUTI of the test device when an SMS Message is sent to it.

C. WhatsApp

WhatsApp is a mobile messenger that uses the internet in order to send and receive messages. A unique feature of these types of messengers is the ability to show when a user is typing a message. Some related research has proposed that the typing notification will also trigger an LTE page along with the sent message. The results defend the claim that the typing indicator also triggers a page. srsLTE indicates that it has captured the test device's GUTI when only typing a WhatsApp message without pressing send.

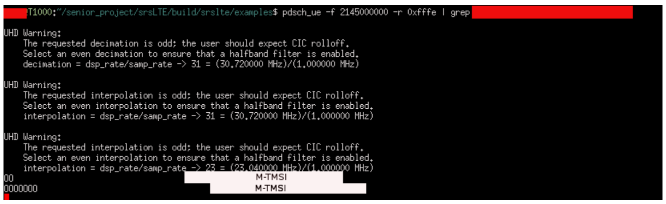

Fig. 5. srsLTE results from a phone call

VI. CONCLUSIONS

The results show that with proper hardware, an attacker can execute a simple identification attack. With a few more data points that the same attacker can track a user. Overall this leak in data can have large implications if it is not resolved by the mobile industry.

A. Cost

The cost of the demonstrated attack is calculated to be around $1000, but due to the versatility of software defined radios the cost can be much less. There have been articles that show individuals that convert television and radio receivers into SDRs. In this article by Vierinen shows how an SDR can be built using a Realtek RTL dongle of less then $25 [15]. With the cost of SDR hardware being so cheap, attacks like LTE paging attacks become more viable.

B. Risks

The risks of this attack is very minimal. The mobile subscriber is never aware that their GUTI is being collected and as a result the adversary has protected. The pages are also broadcasted so there is no easy way to identify individuals that are sniffing for it. The only possible way to identify the adversary was if they were seen with the SDR, but even then it will be difficult to prove that they were collecting/sniffing for GUTIs.

C. Future work

The next step to take in this project would be to determine when and how to get the GUTI value to change. Since these values are temporary, they should be able to change. Since the GUTI values do not change when the device switches MME, it would be interesting to find out when they do.

Another area to explore would be to determine how big of a radius an individual can be tracked using the GUTI and the SDR. Since this may be implementation specific, it would still be interesting to see how far a person can be tracked. The tracking capability will be dependent on what frequency the LTE signal is transmitted on, since some LTE frequencies are capable of traveling farther compared to other frequencies.

REFERENCES

[1] A. Shaik, R. Borgaonkar, N. Asokan, V. Niemi, and J.-P. Seifert, "Practical attacks against privacy and availability in 4g/lte mobile communication systems," *arXiv preprint arXiv:1510.07563*, 2015.

[2] 3gpp. (2016) Evolved universal terrestrial radio access (e-utra); radio resource control (rrc); protocol specification (release 14). [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2440

[3] ——. (2016) Non-access-stratum (nas) protocol for evolved packet system (eps); stage 3 (release 12). [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspxspecificationId=1072

[4] D. F. Kune, J. Koelndorfer, N. Hopper, and Y. Kim, "Location leaks on the gsm air interface," *ISOC NDSS (Feb 2012)*, 2012.

[5] "Software-defined radio," Jun 2017. [Online]. Available: https://en.wikipedia.org/wiki/Software-defined_radio

[6] A. Dubey, D. Vohra, K. Vachhani, and A. Rao, "Demonstration of vulnerabilities in gsm security with usrp b200 and open-source penetration tools," in *Communications (APCC), 2016 22nd Asia-Pacific Conference on*. IEEE, 2016, pp. 496–501.

[7] S. Aragon, F. Kuhlmann, and T. Villa, "Sdr-based network impersonation attack in gsm-compatible networks," in *Vehicular Technology Conference (VTC Spring), 2015 IEEE 81st*. IEEE, 2015, pp. 1–5.

[8] W. L. I. Agency, "Ettus research." [Online]. Available: https://www.ettus.com/product/details/USRP-B200mini-i

[9] srsLTE, "srslte/srslte," Apr 2017. [Online]. Available: https://github.com/srsLTE/srsLTE

[10] Vlm, "vlm/asn1c," May 2017. [Online]. Available: https://github.com/vlm/asn1c

[11] "Lte discovery android apps on google play." [Online]. Available: https://play.google.com/store/apps/details?id=net.simplyadvanced.ltediscovery&hl=en

[12] CellularPrivacy, "Cellularprivacy/android-imsi-catcher-detector." [Online]. Available: https://github.com/CellularPrivacy/Android-IMSI-Catcher-Detector/wiki

[13] "Cell id," Jun 2017. [Online]. Available: https://en.wikipedia.org/wiki/Cell_ID

[14] S. Kinney, "Guti - the lte id that replaces the p-tmsi," Apr 2016. [Online]. Available: http://www.rcrwireless.com/20140509/wireless/guti-explained-the-unique-id-in-lte

[15] "Building your own sdr-based passive radar on a shoestring," Jun 2015. [Online]. Available: https://hackaday.com/2015/06/05/building-your-own-sdr-based-passive-radar-on-a-shoestring/