A Platform Based on srsRAN for Security Research in LTE Network

Shanshan Shen
School of Electronic and Information Engineering
Beijing Jiaotong University
Beijing, China
21125070@bjtu.edu.cn

Hai Li
School of Information and Electronics
Beijing Institute of Technology
Beijing, China
haili@bit.edu.cn

***Abstract*—Because of the fully IP-based heterogeneous network architecture, the Long-Term Evolution (LTE) network faces the threats of malicious attacks. Generally, the open-source Software-Definition Radio (SDR) can be used for attack verification in LTE network. In order to improve research efficiency, we develop a platform based on OAI for attack verification in LTE network, which is called LEAP. However, due to the limitations and flaws of LEAP, we further improve it and develop a platform based on srsRAN for security research in LTE network, which is called LSCAP. LSCAP provides the functions of strategies control, front-end interaction and security data collection. In addition, it uses multi-threading technology to better support the verification of concurrent attacks. Therefore, LSCAP is better at scalability and flexibility than LEAP. In this article, we will introduce the system design of LSCAP and our improvement on LEAP.**

Keywords—LTE, network security, attack verification, srsRAN

I. Introduction

LTE is a cellular communication standard developed by the 3rd Generation Partnership Project (3GPP), which enables people living around the world to communicate in real-time and at low cost, and is also a major factor in promoting modernization of infrastructure and application sectors in various countries [1]. Currently, there are still vulnerabilities in the protection measures for security and privacy in LTE networks [2]. Thus, there is a critical need to carry out the security research for LTE network.

However, because of the inaccessibility to commercial networks and terminals, we are unable to modify their internal implementation for research purposes. Therefore, researchers often use open-source SDR to simulate LTE network [3]-[5]. Common open-source SDR software packages include OpenAirInterface (OAI) [6], openLTE [7], and srsRAN [8]. Because of the complexity of the LTE protocol and SDR code, understanding and modifying the code requires a significant amount of time and effort, which demands researchers to have a high level of programming skills. Furthermore, different research often requires different modifications to the SDR. Therefore, we can only verify one type of attack at the same time.

To address the above issues, we have developed the LTE Evaluation Assistant Platform (LEAP) [9], which is a platform based on OAI for attack verification in LTE network. LEAP can not only track the execution of OAI but also intercept or inject messages at different positions in the signaling process based on user's client scripts. However, there are still some defects of the design and function in LEAP, which pose some difficulties for our security research, including: (1) The function of OAI is not yet complete (such as lack of support for data services). Since the code structure of OAI is complex and the naming is unclear, the cost for learning it can be really high. In addition, OAI is unstable and easily prone to crashing during runtime. (2) when verifying different attacks, LEAP needs to write corresponding Python scripts for each attack, which requires users to do a lot of repetitive programming work, and the Python scripts are fixed before the platform runs. Once the experiment starts, users cannot manipulate the platform in real-time according to the experiment situation, which is not flexible enough. (3) LEAP does not isolate the code for different researches in OAI, which may result in conflicts between these codes. When verifying different attacks, it is necessary to modify the code in OAI again and recompile it.

Based on the above observations, we develop the LTE Signal Control and Analyze Platform (LSCAP) based on srsRAN for security research in LTE network. LSCAP optimized the platform's design and provided more functions, including: (1) LSCAP is developed based on srsRAN, which uses C++ language with object-oriented characteristics. Our design highlights code modularity and reuse, following the principles of high cohesion and low coupling, resulting in clearer and more readable code. (2) LSCAP optimizes the platform's code structure by isolating the code between different researches, fully decoupling them, allowing easy expansion of research content without affecting existing researches. (3) LSCAP encapsulates the code of client and provides external interaction interfaces for researchers, enabling the platform to dynamically read user's input and execute different operations, which makes it more flexible and user-friendly. (4) LSCAP provides the capability to collect and analyze security data. In srsRAN, this security data can be used for signaling assembly and transmission, or can be provided to users for analysis as required.

The rest of this article is organized as follows. Section II describes the system architecture and workflow of our platform. Section III introduces our improvements on LEAP. Section IV explains how to use our platform for security research. Finally, section V provides a summary of this article.

II. System Design

A. Architecture

Our platform is built based on srsRAN and consists of two parts: the server and the client. Its architecture is shown in Figure 1.

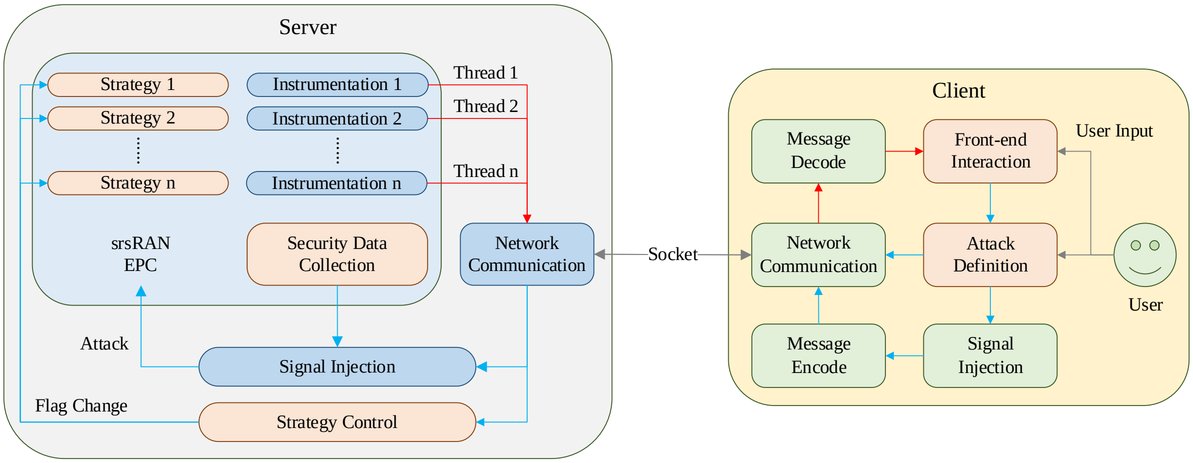

Fig. 1. Architecture of LSCAP. Modules of the server are marked in blue. Modules of the client are marked in green. The new modules we added based on LEAP are marked in orange. The downlink data flow is marked with a red arrow. The uplink data flow is marked with a blue arrow.

The server of LSCAP is written in C++ language and includes the following modules: (1) Instrumentation module that intercepts messages in srsRAN and creates threads to communicate with clients. (2) Network communication module that completes data transmission and reception between server and client. (3) Security data collection module that collects security data from srsRAN. (4) Signaling injection module that injects fake illegal signals into the network according to user's input. (5) Strategy control module that controls the message processing strategy of srsRAN based on user's input.

The client of LSCAP is written in Python language and includes the following modules: (1) Network communication module that completes data transmission and reception between server and client. (2) Message encoding and decoding module that encodes and decodes messages. (3) Front-end interaction module that reads and processes user's input in real time. (4) attack definition module that allows users to define attack processes flexibly. (5) Signaling injection module that commands the server to inject fake illegal signaling into the network according to the attack definition.

B. *Workflow*

Taking the attach process of UE as an example, the workflow of our platform will be introduced as shown in Figure 2.

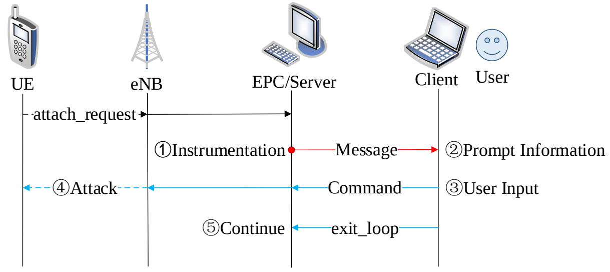

Fig. 2. Workflow of LSCAP. The downlink data flow is marked with a red arrow. The uplink data flow is marked with a blue arrow.

1) The code of instrumentation in the server successfully intercepts the attach_request message from the UE and creates a thread to loop and receive commands from the client. Then, the server packs the intercepted message and the information of instrumentation into a message and sends it to the client while blocking the subsequent protocol process.

2) The client receives messages from the server and parses them. Then it gives prompts to users about the current hooking position of srsRAN and waits for reading user's input.

3) The client packs and sends the corresponding commands to the server based on user's input.

4) The server launches attacks to the LTE network based on the commands received from the client.

5) The client sends an exit_loop command to the server, notifying it to stop the current loop reception at the code of instrumentation and continue with the subsequent protocol process.

III. IMPROVEMENT

In this section, we will introduce the advantages brought by the design of LSCAP and our specific improvement on LEAP.

A. *Scalability*

LSCAP provides more strategies for functions in srsRAN and controls them with the server. This design isolates the codes between different researches and encapsulates them into different strategies, making it easier to expand research content as required. Specifically, different strategies are differentiated by some flags. When receiving commands from the client, the server sets the corresponding flags according to the command. Then, the functions in srsRAN will automatically switch to different strategies according to the flag.

However, when verifying the authentication synchronization failure (ASF) attack with LEAP, the algorithm for SQN in HSS was modified by us to observe the results of the attack more effectively. When verifying other attacks, we need to restore the code in HSS and compile the code of our platform again, which can be very time-consuming. In contrast, when verifying ASF attack with LSCAP, we encapsulate different algorithms in HSS into different strategies and switch them according to the command from the client, reducing a lot of repetitive work for users and allowing for the expansion of more strategies as required, making it highly scalable.

B. *Flexibility*

LSCAP encapsulates the definitions of different attacks in the client into different commands. Each of them corresponds to a user-defined attack process. LSCAP also provides an interface for interaction with the user. This design allows users to send commands to the server in real-time based on the current execution status of srsRAN. We only need to enter the command that we want to execute. Then, the client will read the input and send the command to the server.

However, when verifying different attacks with LEAP, Python scripts of different attacks are required to be written. If the types of attacks are numerous, there will be a significant increase in the number of scripts. In addition, the Python scripts of LEAP are written in advance before the experiment begins and cannot be modified during the experiment, resulting in poor flexibility of LEAP. In contrast, LSCAP only needs to call different functions in one script according to the input of users and has a good flexibility.

C. *Concurrency*

LSCAP uses multi-threading technology to better support the verification of concurrent attacks. When executing the code of instrumentation, LSCAP uses the function of pthread_create() to create a new thread to communicate with the client. Thus, we can inject fake signaling at multiple locations in the network simultaneously, allowing users to verify attacks that their patterns are more complex. However, LEAP communicates with the client using the thread at the current protocol process. Thus, the signaling can only be injected by LEAP at one location in srsRAN at the same time, which has certain limitations.

In addition, LSCAP can block the current protocol process by using the function of pthread_join(), which ensures that the server successfully receives and executes the commands from the client before the protocol process continues downward. This design avoids the situations that the commands of users have not taken effect yet but the protocol has already entered the next process and caused the failure of attack verification. However, when verifying numb attack, LEAP does not block the subsequent process of authentication. Thus, the message of authentication_request may be sent before the message of authentication_reject, which will cause the failure of attack verification. Therefore, LSCAP has stronger ability to control the signaling process in LTE network than LEAP.

D. *Data Collection*

LSCAP provides the ability to collect security data in srsRAN. For example, LSCAP can dynamically collect and update security data just like the context of UE, socket information of base station, information of interface in EPC and so on, which can be used by the server to pack and send fake signaling. In addition, these data can also be extracted according to the need of users and provided to researchers for analysis.

IV. Experiment

A. *Environment*

To test the function of LSCAP, we use two computers with Ubuntu 16.04.7 operating system to run the programs of srsenb and srsepc respectively. We build a simple LTE mobile communication network with the hardware devices such as USRP B210. In order to analysis the effects of different attacks better, we use the commercial phone of Huawei as the terminal device. Our experimental environment is shown in Figure 3.

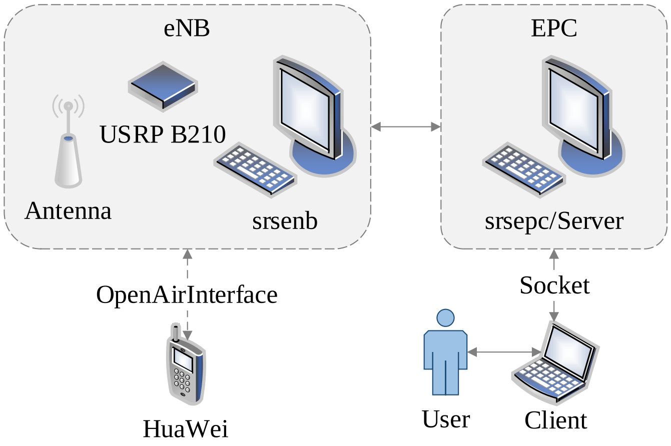

Fig. 3. Experimental environment of LSCAP.

B. *Attack Verification*

In this section, we will use LSCAP to verify several common Denial of Service (DoS) attacks.

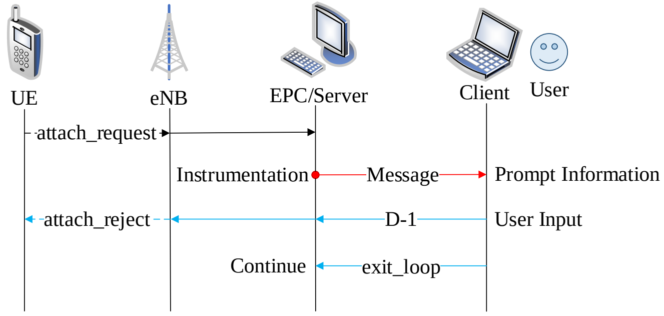

Fig. 4. Process of verifying D-1 using LSCAP.

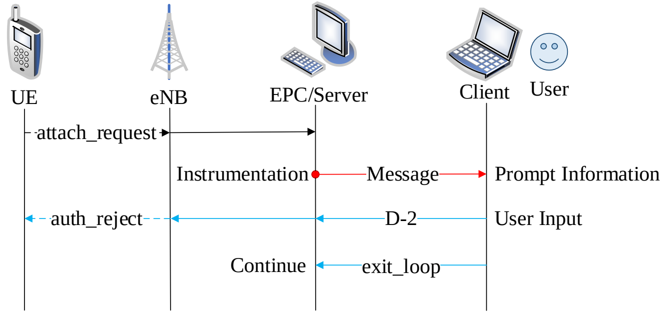

Fig. 5. Process of verifying D-2 using LSCAP.

*1) **Dos attack with attach_reject:*** When the UE initiates an attach_request message to our platform, LSCAP will intercept this message and communicate with the client. At this time, we can enter the command id of pack_and_send_attach_reject on the client to send a fake attach_reject message to the UE, indicating that the network refuse access of this UE. Since this attack occurs before the process of authentication, the victim UE cannot recognize that it is connected to an illegal network.

In the following text, we will abbreviate this attack as D-1. The process of verifying D-1 using LSCAP is shown in Figure 4. The result of our experiment indicates that the attach_reject message will cause the UE to disconnect from the network. Then, the UE will be prevented from accessing the current network again.

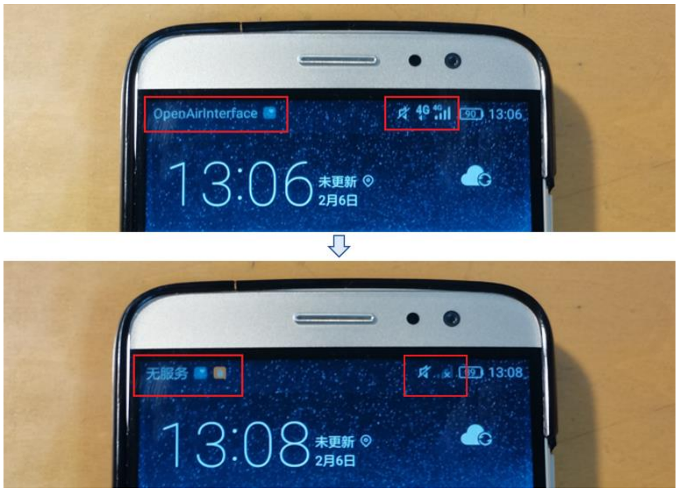

Fig. 6. The changes that occur in the UE under D-2.

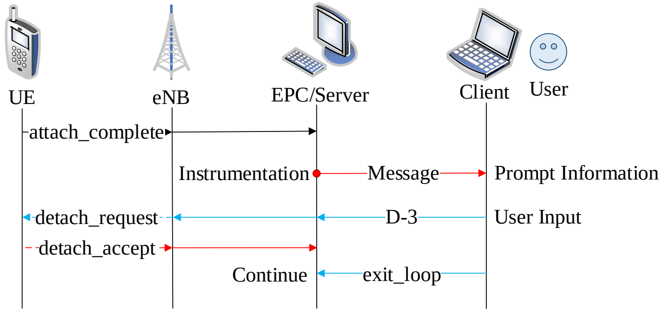

Fig. 7. Process of verifying D-3 using LSCAP.

*2) **Dos attack with authentication_reject:*** When the UE starts the process of authentication, we can send a fake authentication_reject message to the UE by entering the command id of pack_and_send_authentication_reject on the client, indicating that the network refuses to authenticate with this UE. Since any context of the UE is not required, this authentication_reject message may be forged and sent by an malicious attacker in the real LTE network. In the following text, we will abbreviate this attack as D-2. The process of verifying D-2 using LSCAP is shown in Figure 5. The result of our experiment indicates that the authentication_reject message will cause a state of user-inactivity in the UE. The UE will be unable to receive SMS and calls from the network and display “no service” until the phone is restarted or the SIM card is reinserted. The changes that occur in the UE under this attack are shown in Figure 6.

*3) **Dos attack with detach_request:*** When UE completes the process of attach, we can enter the command id of pack_and_send_detach_request on the client and send a fake detach_request message to the UE, indicating that the UE should datach from the current network. In the following text, we abbreviate this attack as D-3. The process of verifying D-3 using LSCAP is shown in Figure 7. The result of our experiment indicates that the detach_request message will cause the UE to disconnect from the network. But after a period of time, the UE will initiate an attach_request message to the network again and successfully access the network.

V. Conclusion

In this paper, we have developed a platform based on srsRAN for security research in LTE network, which is called LSCAP. Compared with LEAP, LSCAP has better scalability and flexibility. It is easier to use than LEAP and has provided more practical functions. First, we introduced the architecture and workflow of LSCAP. Then, we elaborated on our design and improvement in detail. Finally, we verified three DoS attacks using LSCAP. In the future, we hope to continue to develop the module of security data collection to automatically analyze the traffic in LTE network and complete the work of attack detection. In addition, with the large-scale deployment of the 5G network in China, we also hope to continue to develop our platform to support security research in 5G network.

References

[1] S. R. Hussain, “A Systematic Framework For Analyzing the Security and Privacy of Cellular Networks”. Purdue University Graduate School, 16-Jan-2020, doi: 10.25394/PGS.7347371.v1.

[2] Hussain, Syed and Chowdhury, Omar and Mehnaz, Shagufta and Bertino, Elisa. (2018). LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE. 10.14722/ndss.2018.23319.

[3] Gomez-Miguelez, Ismael & Garcia-Saavedra, Andres & Sutton, Paul & Serrano, Pablo & Cano, Cristina & Leith, Douglas. (2016). srsLTE: An Open-Source Platform for LTE Evolution and Experimentation.

[4] Fardan, Fardan & Istikmal, Istikmal & Mawaldi, Ikbal & Anugraha, Tides & Ginting, Ishak & Karna, Nyoman. (2020). Experimental Security Analysis for Fake eNodeB Attack on LTE Network. 140-145. 10.1109/ISRITI51436.2020.9315427.

[5] H. Kim, J. Lee, E. Lee, and Y. Kim, “Touching the untouchables: Dynamic security analysis of the LTE control plane,” in Proc. IEEE Symp. Security Privacy (SP), 2019, pp. 1153–1168.

[6] OpenAirInterface. Accessed: Mar. 21, 2023. [Online]. Available: http://www.openairinterface.org/

[7] OpenLTE. Accessed: Mar. 21, 2023. [Online]. Available: http://openlte.sourceforge.net/

[8] SrsRAN. Accessed: Mar. 21, 2023. [Online]. Available: https://github.com/srsran/srsRAN_4G/

[9] W. Wang and H. Li, "Light-Weight Platform for Attack Validation in LTE Network," in IEEE Networking Letters, vol. 2, no. 4, pp. 212-215, Dec. 2020, doi: 10.1109/LNET.2020.3025019.