The diverse and variegated reactions of different cellular devices to IMSI catching attacks

Ivan Palamà
CNIT/University of Rome - Tor Vergata
ivan.palama@cnit.it

Francesco Gringoli
CNIT/University of Brescia
francesco.gringoli@unibs.it

Giuseppe Bianchi
CNIT/University of Rome - Tor Vergata
giuseppe.bianchi@uniroma2.it

Nicola Blefari Melazzi
CNIT/University of Rome - Tor Vergata
blefari@uniroma2.it

ABSTRACT

The goal of this paper is to assess how different User Terminals react to IMSI-catching attacks, namely location privacy attacks aiming at gathering the user's International Mobile Subscriber Identity (IMSI). After having implemented two different attack techniques over two different Software-Defined-Radio (SDR) platforms (OpenAirInterface and srsLTE), we have tested these attacks over different versions of the mobile phone brands, for a total of 19 different radio modems tested. We show that while the majority of devices surrender almost immediately, iPhones seem to implement some cleverness that resembles proper countermeasures. We also bring about evidence that the two chosen SDR platforms implement different signaling procedures that differentiate their ability as IMSI-catchers. We finally analyse IMSI-catchers' behaviors against subscribers of different operators, showing that successfulness of the attack depends only on the chipset and the SDR tool. We believe that our analysis may be useful either to practitioners that need to experiment with mobile security, as well as engineers for improving the design of mobile modems.

CCS CONCEPTS

• **Security and privacy** → **Network security**; **Mobile and wireless security**;

KEYWORDS

IMSI catcher, Mobile Systems, users' privacy

ACM Reference Format:
Ivan Palamà, Francesco Gringoli, Giuseppe Bianchi, and Nicola Blefari Melazzi. 2020. The diverse and variegated reactions of different cellular devices to IMSI catching attacks. In *14th International Workshop on Wireless Network Testbeds, Experimental evaluation & Characterization (WiNTECH'20), September 21, 2020, London, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3411276.3412191

WiNTECH'20, September 21, 2020, London, United Kingdom
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8082-9/20/09…$15.00
https://doi.org/10.1145/3411276.3412191

1 INTRODUCTION

Location privacy has been a crucial native requirement of cellular networks. Indeed, since the old times of early GSM networks (2G systems), a systematic effort has been made in the protocol design so as prevent as much as possible disclosure of the "true" user identifier, namely the IMSI. For this reason, whenever possible, the users are identified using frequently changing temporary subscriber identities.

Still, there are some cases (for instance when a user registers in a network for the first time) where a temporary identifier is not yet assigned, and hence the user must explicitly expose her IMSI. This weakness may be exploited by IMSI catching attacks [1], usually carried out by devices which intentionally recreate scenarios devised to force a neighboring device to reveal its long term identity. Often commercially called *StingRays*, IMSI catchers are mainly used by public authorities and law enforcement agents for legitimate purposes, but can also be easily implemented over low cost SDRs, as discussed later on.

Therefore, several incremental mechanisms have been introduced in UMTS and LTE to protect subscribers' identities. And more recently, 5G systems have finally standardized a brand new approach which guarantees that the user identifier (the IMSI, renamed in 5G as SUPI - Subscription Permanent Identifier) shall be *never* transmitted in clear, but it is transmitted as a Subscription Concealed Identifier (SUCI) duly employing public key encryption means [2]. Still, despite such improvements, IMSI catchers remain an unsolved problem (and certainly it is not goal to solve them here!). A clever attacker may, in fact, combine a number of techniques (Jamming, man-in-the-middle, downgrade techniques) to coerce the mobile terminal in connecting to a rogue base station claiming to be a legitimate base station, but also declaring an older version of the protocol so as to defeat advanced protection techniques developed in later standard releases (including circumventing the 5G cryptographic concealment [3]) and force the user to reveal the IMSI.

Many papers [4–8] have described the protocol vulnerabilities that permit an attacker to deceive the user equipment, and how to implement IMSI catching techniques over inexpensive SDR platforms. Also, other papers [9–11] have developed solutions - such as IMSI catcher "catchers" - to detect and/or thwart an IMSI catching attack in progress. Still, to the best of our knowledge, no paper has so far performed a relatively large scale analysis of what is the effect of IMSI catching attacks on off-the-shelf devices, i.e. how "easy" is for an attacker to mislead commercial phones and convince them to disclose the IMSI.

As a matter of fact, this work was mainly driven by the curiosity to understand if some mobile phone brands had some native "resilience" to such attacks. Indeed manufacturers well know the vulnerabilities and techniques that make the attack possible; as such they might design devices which react to IMSI-catching "stimula" in different manners, and make the job of a layman attacker a bit harder. To shed some light on this matter, other than implementing IMSI-catching strategies over low cost SDRs, the main contribution of this paper is an experimental analysis devised to understand how different devices react to different forms of IMSI catching attacks. Moreover, in the case of emerging differences, our goal is also to understand whether these different reactions are due to the different radio modem, or brand, or operating system, as well as to understand whether the network operator to which the device is connected (and hence the provider of the device' SIM) plays some role.

We ran our analysis using two different SDR platforms (OpenAirInterface and srsLTE), two different IMSI catching strategies (see section 3), and with/without jamming of the legitimate network to which the terminal is initially registered. All the attacks are performed over publicly operated 4G/LTE networks, as it would not make sense to test them over past generation technologies, and all the four operators of our country (Italy) having a physical network deployed have been tested. In terms of devices, our analysis encompasses a quite large number of terminals, specifically 19 devices with the corresponding radio modems and operating system versions. Our experiments reveal that virtually all the mobile phones based on the Android Operating System surrender almost immediately to an attack - in most cases jamming is not even necessary! Rather, iOS-based devices seem to be a slightly harder target for an IMSI catching attack.

The remainder of the paper is organized as follows. After some necessary background in section 2, we describe our IMSI catching techniques in section 3, and our SDR implementation in section 4. Experimental setup and results are described in section 5 and 6, respectively. Finally, section 7 draws conclusions and outlines further research directions.

2 BACKGROUND

We provide in this Section a quick overview of Long Term Evolution (LTE), namely the 4th Generation (4G) standard for mobile networking. We focus on the aspects that are necessary for understanding the vulnerabilities that make IMSI catching possible. We refer to the standards [12, 13] for further details.

2.1 LTE main architecture components

Figure 1 highlights the three main components of the architecture: the user equipment (UE), the Evolved Universal Terrestrial Radio Access Network (E-UTRAN), and the Evolved Packet Core network (EPC), which we describe next.

UE It is the mobile device and is equipped with two main components: the LTE radio interface, and the Universal Subscriber Identity Module (USIM). The USIM holds the unique IMSI, and the cryptographic material and procedures used by the USIM for generating ephemeral quantities during the authentication procedure. The USIM also stores temporary identifiers that can be used during authentication instead of the IMSI for preserving the user's privacy: in this work we focus on the UE and on how to force it to expose its IMSI.

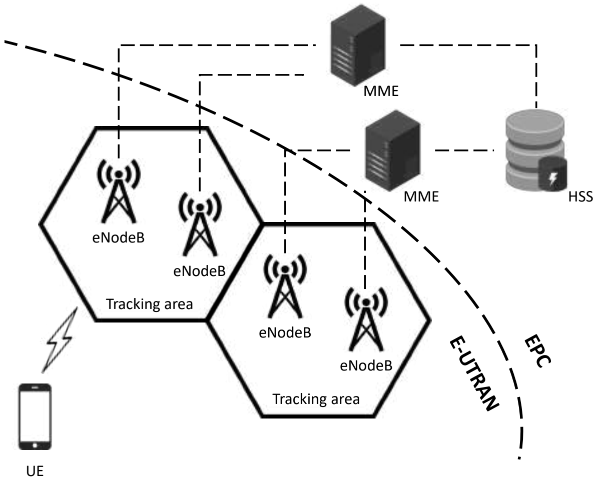

Figure 1: LTE architecture: main components.

E-UTRAN The radio access network comprises radio cells served by a transceiver, called Evolved Node B (eNodeB), that terminates the wireless channel, relays messages to the core network and schedules the access order of all connected UEs. Setting up a rogue eNodeB is straightforward and this is essentially the enabling factor behind the IMSI-catching attack.

EPC The core network is composed of many entities; we only mention the Mobility Management Entity (MME) and the Home Subscriber Server (HSS). Each MME connects several eNodeBs and the attached UEs to the core network. An MME allocates resources to the UEs and runs the authentication procedure when a UE attaches to the network: as we will see, the MME cannot authenticate a UE alone and it requires additional information from the HSS. The MME groups a few eNodeBs into the same Tracking Area Code (TAC): such codes are used for tracking the movement of UEs that are in standby mode and they play a fundamental role during the IMSI-catching attack.

HSS: it is essentially a database which contains user-related information, including cryptographic keys, Quality of Service (QoS) profiles and roaming policies. It also stores the identifier of the last MME that authenticated the UE, the Globally Unique MME Identifier (GUMMEI); and the Temporal Mobile Subscriber Identifier (TMSI). These two ids together form the Globally Unique Temporary ID (GUTI) that is also shared with the USIM after a successful authentication. The HSS generates the quantities that are used by the MME during the authentication to decide whether or not a UE can access the network.

2.2 Protocol Architecture

Several protocols cooperate to maintain the connectivity of UEs at two different levels: the control-plane that transports signaling messages and the user-plane, used to route user packets between

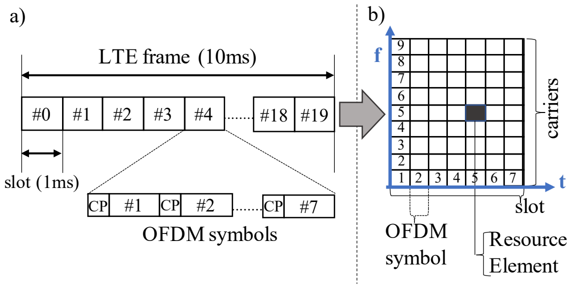

Figure 2: OFDM modulation in the downlink channel.

the UE, the MME and the Internet gateway. We will not discuss the user plane as it is not relevant to our work.

Control Plane: The control plane implements functionalities such as broadcasting system information and authentication. It is composed of two network layers: Non Access Stratum (NAS), which is the network layer communication between the UE and the MME, and the Access Stratum (AS), which is used for communication between the UE and the eNodeB. The control plane protocol stack is composed of the Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC), Medium Access Control (MAC) and the Radio Resource Control (RRC).

RRC: The Radio Resource Control protocol includes many features such as broadcast of system information (both AS and NAS), management of UE temporary identifiers, intra-frequency and inter-frequency handover, cell selection and reselection and context transferring.

2.3 Physical layer

We quickly recall here the modulation adopted by LTE for the downlink channel as it will later help understanding the jamming subsystem. An eNodeB schedules transmissions to connected UEs within a continuous sequence of 10 ms long *LTE frames*. Figure 2-a shows the typical LTE frame structure: a frame is composed of 20 equally sized *slots* and each slot, in turn, contains seven OFDM symbols[1]. Symbols are created in the frequency domain by directly assigning the values of $N \cdot 128$ sub-carriers: after IDFT, a cyclic prefix is inserted at symbol head, which adds either $N \cdot 10$ (only for the first OFDM symbol) or $N \cdot 9$ time samples. As sub-carrier spacing is fixed to 15 KHz, the total spectral width depends on the choice of the N factor: setting it to increasing values from the set $\{1, 2, 4, 8, 12, 16\}$ boosts the sample rate from $1.92MS/s$ up to $30.72MS/s$ that corresponds to spectral occupations respectively of $1.25MHz$ and $20MHz$. Figure 2-b shows the same elements in the typical *fabric*-like structure used for representing LTE signals: there we report a Resource Element (RE), i.e., the smallest defined unit which consists of one OFDM sub-carrier during one OFDM symbol interval. A group of 12 (over frequency) per 7 (over time) REs forms a Resource Block (RB), which is the smallest unit of resources that can be allocated to a user. In each RB, four RE are dedicated to carry synchronisation signals: as we will see, repeatedly destroying such elements could harm correct reception at the UE side.

2.4 Authentication in 3GPP

We report in Figure 3 the mutual authentication procedure that is executed when the UE connects to a serving network for the first time. The goals of the Evolved Packet System - Authentication and Key Agreement (EPS-AKA) protocol are: i) derive a set of ephemeral keys that will be used afterwards for encrypting and integrity protecting the exchanged traffic; and ii) create the GUTI, i.e., the temporary UE's identifier.

After the UE completes the RRC procedure with the eNodeB, the controlling MME transmits an Authentication Request to the HSS in the home network, including its own network identifier and the UE's IMSI. Starting from the UE's long term key K_i, the HSS creates many authentication vectors (AVs) that it sends back to the MME in an Authentication Response message. The MME extracts the AUTH and XRES fields from one AV and transmits the AUTH token to the UE. Here the USIM uses its copy of the key K_i to verify the token. If this proves the authenticity of the network, the USIM generates a response RES that the UE forwards to the MME. If the RES received by the MME matches the expected response XRES in the AV, then the authentication is mutually successful and all the involved entities generate and cache the UE's GUTI.

When the UE moves under a new MME, the new MME uses the UE's GUTI for identifying the previous MME. The two MME then share the IMSI of the UE that is hence not disclosed over the air. In exceptional cases, e.g., if the previous MME has purged its internal database, the new MME must obtain the IMSI from the UE itself by sending an Identity Request: when this happens the IMSI is transmitted in the clear over the air.

3 VULNERABILITY AND IMSI CATCHING

Despite EPS-AKA mutual authentication and strong encryption algorithms, attackers can steal IMSIs by capturing signaling messages that are broadcast as plaintext to all surrounding base stations (or IMSI Catchers). The following NAS messages can be exploited before a secure NAS signaling connection is established: Identity Request, Authentication Request, Authentication Reject, Attach Reject, Detach Accept, Tracking Area Update Reject, Service Reject. In our work we exploit two of these messages to perform an IMSI catching attack [12]:

- Cell Reselection Tracking Area Update Reject
- Service Reject

Cell Reselection Tracking Area Update: Thanks to the cell reselection procedure, the UE always camps on or connects to the best cell in terms of radio condition: to this end it keeps measuring the signal qualities of the serving and neighboring cells. There are two types of Cell reselection: intra-frequency cell reselection, which is based on cell ranking; and inter-frequency cell reselection, where the UE exploits absolute frequencies priorities to camp on the highest priority frequency available. During reselection, the UE examines the tracking area code from the cell's system information SIB1. If the UE has moved into a tracking area in which it was not previously registered, then it performs the tracking area update procedure. A legitimate MME receiving a Tracking Area Update Request message would retrieve the UE information from the network. Conversely, a malicious MME may claim that this is not possible, i.e., it will reply to the UE with a Tracking Area Update Reject with

[1] We only report here details of the *normal cyclic prefix* modulation scheme: readers can refer to the 3GPP documentation for the alternative *extended cyclic prefix* format.

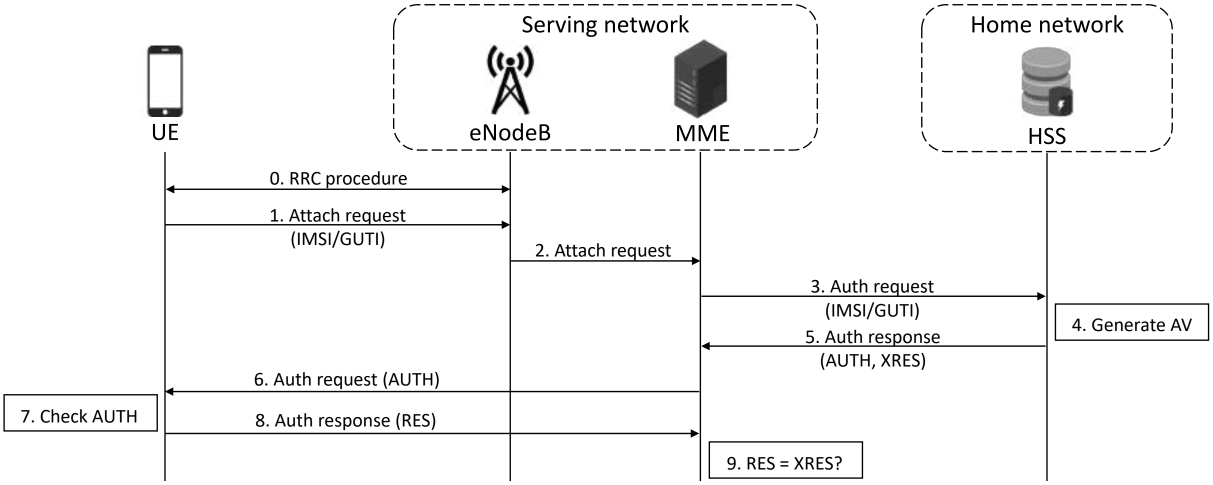

Figure 3: LTE authentication procedure.

Cause 9 *"UE identity cannot be derived by the network"*. When the UE receive the Tracking Area Update Reject message, it will try to perform an Attach procedure sending an Attach Request message containing its IMSI.

Service Request: When the UE needs to send new traffic or learn network's intent to send new traffic, it sends to the MME a Service Request message. Then the UE, by using the allocated radio and network resources, can receive or send traffic. Service requests can be triggered by a UE or by a network, and can be categorized depending on where the new traffic is generated:

- Service Request case 1: When there is uplink data to be sent from UE to the network;
- Service Request case 2: When there is downlink data to be sent from the network to UE.

When a malicious MME receives an UE Service Request message, it will reply with a Service Reject message with Cause 9 *"UE identity cannot be derived by the network"*, then the UE will try to perform an Attach procedure sending an Attach Request message containing its IMSI.

4 IMSI CATCHER IMPLEMENTATION

In this section, we describe how we implemented the LTE IMSI Catcher using low cost SDR. We carried out all the experiments in our wireless network security lab to avoid disturbing other UEs. We kept the victim UE close to the IMSI Catcher system in each experiment in order to meet the radio signal power requirement of cell reselection procedure.

To perform the IMSI catching attack we need to deploy a malicious LTE network and then we need to force the UE to connect to it. The IMSI Catcher system is therefore composed of 2 main components: the malicious network and the LTE jammer. The overall attack scenario is reported in Figure 4.

4.1 Malicious network

To deploy the rogue eNodeB we use either srsLTE developed by Software Radio Systems (SRS) [14], or OpenAirInterface developed by the OpenAirInterface Software Alliance (OSA) [15]. Both are complete solutions for setting up LTE networks and include the eNodeB, for managing the air interface, and the MME and HSS for creating the core network.

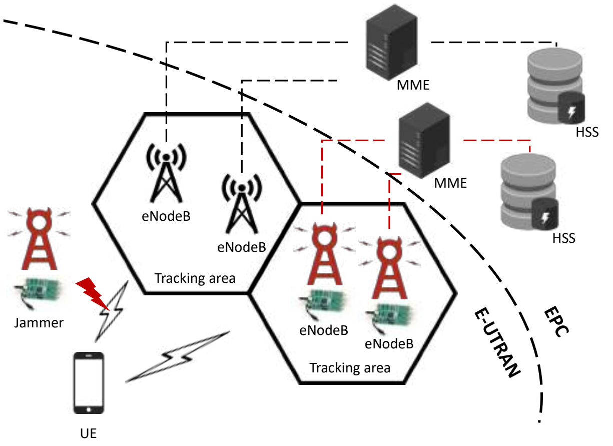

Figure 4: Attack scenario.

To set up a malicious LTE network we need to replicate the same parameters used by the real operator for the cell under attack, which include: the Mobile Country Code (MCC), the Mobile Network Code (MNC), the Physical Cell Identifier (PCI), and the Tracking Area Code (TAC). While the first two parameters identify the operator, PCI and TAC are specific of the eNodeB that connects the UEs under attack and depend on the geographical area. Another important set of parameters is the list of alternative eNodeBs of the same operator that are available in the same area, list that contains their PCI, TAC and frequencies. We should jam all such frequencies in order to avoid UEs connect to another eNodeB when the rogue cell is switched on.

To retrieve these parameters we have two options: one is to run a network-monitoring tool on a UE that is attached to the network under attack. On Android there are apps like *NetMonster* or *Netmonitor* that can collect and show such parameters to the user/attacker. On iPhone we can access the same parameters by dialing number “*3001#12345#*” on the phone app that will pop up the *field test menu*. The second option requires a SDR for sniffing all bands allocated for LTE and automatically discovering all parameters of all operators. This approach is hence much more oriented to an attack than the previous one and it does not even rely on the attacker owning any registered USIM. This approach runs in three phases: i) **automatic network discovery**, by using the srsLTE module called *cell_search* we discover all networks available in the area we are; ii) **collecting the scheduling information**, by using the srsLTE module called *cell_measurement* we sniff the signal transmitted by each operator, no matter from which eNodeB, and obtain such information from the System Information Block 1 (SIB 1) that is broadcast in the clear. SIB1 contains the Scheduling info list, the si-WindowLength and the si-Periodicity: these parameters are fundamental for running the third phase; iii) **obtain the Inter-frequency Cell Reselection priority list**, with a slightly modified version of *cell_measurement*, we extract the SIB 5 using the parameters collected during the previous phase, and we finally retrieve the list of frequencies and related priorities that are used by the operator under attack in this area.

4.2 LTE jammer

The jammer is the second main component of the IMSI Catcher system. In areas where multiple eNodeBs cover different bands, there is a non negligible chance that the UE moves to an eNodeB operating on a different frequency when we activate the rogue eNodeB. We can avoid this situation by jamming all other eNodeBs, and we can use a single SDR as jammer by hopping over the corresponding frequencies. By taking inspiration from existing solutions [16, 17], which suggest to use for jamming a signal with a LTE structure, we developed our own dual chain hopping jammer: this made the deployment of the jamming system easier and allowed a better control of the jamming frequency over time.

We implemented the jammer to mimic the *fabric*-like LTE signal structure: it can operate with all the possible spectral widths by configuring the N parameter as in Section 2. Instead of jamming the entire spectrum of the selected LTE channel, the signal generated by our system covers a small window of N_{sub} contiguous sub-carriers that quickly moves inside the LTE channel over time with configurable period. The software creates the signal in the frequency domain and applies a IDFT to get samples in the time domain. The jammer uses both transmission chains to maximize the effectiveness.

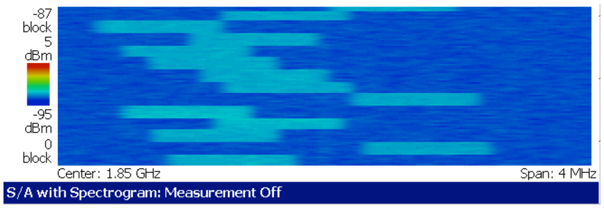

Figure 5: Jamming signal over time (vertical).

We generated Figure 5 with a Tektronix Real-Time Spectrum Analyzer RSA3408A [18]: we captured the behavior of our system inside the LTE channel centered at 1.85 GHz, for $N = 2$, $N_{sub} = 60$ (a window corresponding to 450 KHz of jammed spectrum), and period set to 38 ms. After approximately 14 small hops inside the channel, or equivalently after 500 ms, the jammer changed the LTE channel by re-tuning the SDR device.

As a rule of thumb we suggest to configure the window to cover at least $N_{sub} = 30$ sub-carriers so that it would always disturb at least two RBs and the associated reference signals. During the experiments we also noticed that the B210 SDR takes a bit of time for settling on a new frequency after hopping: for this reason we suggest to avoid hopping times smaller than 500 ms.

5 EXPERIMENTAL SETUP

In this section, we present the experimental setup of our IMSI Catcher. To overcome the excessive cost and size of traditional telecommunications equipment, we use a SDR approach that includes devices that can be easily accessed from the commercial market and that we show in Figure 6, more specifically:

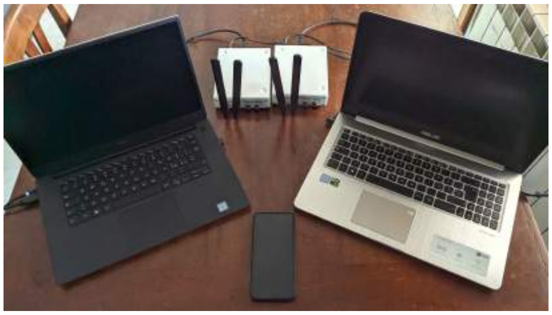

Figure 6: LTE IMSI Catcher experimental setup.

Computers We used two laptops: one Dell XPS 15 7590 with a 6-core Intel Core i7-9750H CPU clocked at 4.50 GHz; and one Asus VivoBook Pro N580G with a 6-core Intel Core i7-8750H CPU clocked at 4.10 GHz. Both computers were running Ubuntu 18.04 LTS with kernel version 5.3.0-53-lowlatency.

Radio Transceiver As radio front-end we used two USRP B210 devices [19]. As they can be tuned over a wide radio frequency range, from 70 MHz to 6 GHz, they cover all the LTE frequency bands. We used LTE specific antennas attached to the SMA connectors of the SDR devices.

Software In our experiments we used two different opensource frameworks that implement LTE RAN and LTE CN: srsLTE and OpenAirInterface. Both are fully compliant with LTE Release 10 and provide us an excellent LTE experimentation platform. For both we used the most recent commit available at the time of this writing (May/June 2020).

Software radio systems LTE (srsLTE) is an LTE open source software suite for SDR applications that includes: srsUE, a complete SDR LTE UE application; srsENB, a complete SDR LTE eNodeB application; srsEPC, a light-weight LTE EPC implementation with MME, HSS and S/P-GW.

OpenAirInterface (OAI) is an open source project that includes: OAI Radio Access Network (OAI-RAN), which implements 4G and 5G Radio Access Network (eNB, gNB and 4G, 5G UE), and OAI Core Network (OAI-CN), which implements 4G Evolved Packet Core (EPC) and 5G Core Network.

We also used our own jammer software that we described in Section 4.2. Moreover, for obtaining the list of cells available in a given area, we modified module cell_measurement from srsLTE. We release the code of the jammer, the patch for cell_measurement, and a detailed HOWTO for helping practitioners and researchers setting up the IMSI catchers described in this paper. Everything is available through github at

https://github.com/ansresearch

6 RESULTS

In this section, we present the obtained results highlighting the different behavioral analyses carried out, with which we started the reverse engineering process to understand how the devices approach the IMSI catching attack.

6.1 Dependence on the operating system

We have tested many heterogeneous devices in order to verify how the operating system could influence the behavior of the device to an IMSI catching. We have tested both Android and iOS devices, the detailed list of tested devices and related results are reported in Table 1, the informations reported in the table are the most reliable that we have been able to find.

The interesting result that we obtained is that, differently from Android devices, iPhone 7 and newer versions display different behaviors, i.e., they are more robust to IMSI catching and they provide IMSI only occasionally. We repeated multiple tests in order to better understand why iPhones behave differently. If we do not run LTE jammer over non-priority frequencies, when we start the malicious eNodeB the iPhone switches to another LTE cell. If we run LTE jammers over non-priority frequencies, often when we start the malicious eNodeB the iPhone automatically downgrades to 3G or even GPRS without providing IMSI to our IMSI Catcher.

6.2 Impact of the Operator

A natural question that emerged during the experiments was whether the subscriber's Operator plays some role. In fact, on one side, all the procedures and relevant parameters involved in IMSI catching scenarios are actually managed by the actual Operator to which the UE is registered. But on the other side the vulnerabilities exploited by our IMSI catchers revolve around a level of detail which appears more related to the radio modem and device firmware, hence might not be targeted by the operators' configuration of the UE USIM.

To shed some light on this question, we bought commercial off-the-shelf USIMs/contracts from all four Italian Operators (Wind3, Vodafone, TIM, Iliad), and we ran experiments using one of the Android phones, specifically the Realme X2 Pro (Android 10). Results appear to suggest that the IMSI catching behavior of the victim does not depend on the operator. Indeed, as summarized in Table 2, the attack had success with all the possible combinations - highlighted in the table as the "cartesian" product among the four operators, the two rogue BS technologies, the two different attack techniques, and the usage or not usage of jamming.

6.3 Further aspects and lessons learned

Different USIMs. To better understand iPhones' behavior, we ran some experiments for studying how they perform network attachment. To this end, we set up our own LTE networks with both OAI and srsLTE, and we used programmable USIMs provided by Osmocom [20] and Open-Cells [21]. We noticed two weird facts with recent phones from Apple: i) they seem to ignore networks generated using the two software, no matter what the configured MCC/MNC parameters are. For instance, they do not even list them when toggling the "Network Selection" switch from "Automatic" to manual; ii) these phones seem also to ignore programmable USIMs: when configured with fake MCC/MNC taken from real operators they do not even try connecting, action that should anyways end with an authentication error as the long term key in the USIM is obviously wrong.

iPhones use Carrier Bundles to manage all parameters related to cell phone providers. It would be interesting to try adding new bundles but unfortunately, even if Jailbreak allows to create the corresponding files, carrier settings are signed. Bypassing this would require an OS level patch.

Different version of operating system Another experiment carried out to better understand iPhones' behavior was to test how different versions of iOS impact network management. We found that while iPhone 5S ignores programmable USIMs with iOS 11, it accepts them when upgraded to iOS 12. This allows us to say that the operating system has (some) impact on the device's network behavior.

Different LTE software From the experimental tests we ran, we found a different behavior between OpenAirInterface and srsLTE. OpenAirInterface has a protocol implementation that allows the IMSI Catcher system to deceive the user and steal its IMSI almost immediately ($< 5s$), while srsLTE needs slightly longer time.

7 CONCLUSIONS

The main contribution of our work is a relatively large scale analysis of how commercial off-the-shelf devices react to IMSI catching attacks. Our results reveal that the sheer majority of devices immediately falls victim of our attacks, and often - to our surprise - with no need for targeted jamming! Only in one case (iPhone) we encountered a more resilient behavior, with jamming being necessary. In this paper we so far "limited" to bring about evidence of such different behavior. Indeed, with no technical information available from the manufacturers, a thorough understanding on *why* iPhones are more robust, and whether they specifically implement tailored defences, would require a way more careful and time consuming investigation and reverse engineering work, activities that we leave as our future next step.

As side contributions, the investigation that we carried out has unveiled interesting technical insights on the diverse behavior not only of the phones (and their radio modems and operating systems), but also of the SDR platforms employed. This non marginal difference between srsLTE and OpenAirInterface was not initially foreseen, and hence appears worth of a deeper investigation.

Model	OS	Modem	LTE Cat.	OpenAirInterface				srsLTE			
				w/o jammer		w/ jammer		w/o jammer		w/ jammer	
				Service Request	TAU Request	Service Request	TAU Request	Service Request	TAU Request	Service Request	TAU Request
Samsung Galaxy S9	Android 9	Exynos 9810	18	✓	✓	✓	✓	✓	✓	✓	✓
Samsung Galaxy A7 2018	Android 10	Exynos 7885	12	✓	✓	✓	✓	✓	✓	✓	✓
Samsung Galaxy Note Pro	Android 5	Snapdragon 800	4	✓	✓	✓	✓	✓	✓	✓	✓
Realme X2 Pro	Android 10	Snapdragon X24	20	✓	✓	✓	✓	✓	✓	✓	✓
Realme 6	Android 10	Helio G90T	13	✓	✓	✓	✓	✓	✓	✓	✓
Xiaomi Redmi Note 7	Android 9	Snapdragon X12	12	✓	✓	✓	✓	✓	✓	✓	✓
Xiaomi Mi A1	Android 9	Snapdragon X9	7	✓	✓	✓	✓	✓	✓	✓	✓
Huawei Mate 20 Pro	Android 9	HiSilicon Kirin 980	21	✓	✓	✓	✓	✓	✓	✓	✓
Huawei P30 Lite	Android 9	HiSilicon Kirin 710	12	✗	✓	✓	✓	✗	✓	✓	✓
Huawei P8 Lite	Android 7	HiSilicon Kirin 655	6	✓	✓	✓	✓	✓	✓	✓	✓
Asus Zenfone 2	Android 5	Intel XMM 7260	4	✓	✓	✓	✓	✓	✓	✓	✓
iPhone 11	iOS 13	Intel XMM 7660	18	✗	✗	✓	✓	✗	✗	✓	✓
iPhone XS	iOS 13	Intel XMM 7560	16	✗	✗	✓	✓	✗	✗	✓	✓
iPhone 8	iOS 13	Intel XMM 7480	16	✗	✗	✓	✓	✗	✗	✓	✓
iPhone 7	iOS 13	Intel XMM 7360	9	✗	✓	✓	✓	✗	✓	✓	✓
iPhone SE	iOS 12	Qualcomm MDM9625M	4	✓	✓	✓	✓	✓	✓	✓	✓
iPhone 5S	iOS 12	Qualcomm MDM9615M	3	✓	✓	✓	✓	✓	✓	✓	✓
Huawei E3272 USB Stick	-	HiSilicon Balong 710	4	✓	✓	✓	✓	✓	✓	✓	✓
Huawei E392 USB Stick	-	Qualcomm MDM9200	3	✓	✓	✓	✓	✓	✓	✓	✓

✓ : IMSI catched

✗ : IMSI not catched

Table 1: Analysis of modem impact on IMSI catching

Tim	×	OAI	×	Service Request	×	w/o jammer	=	Positive
Wind3		srsLTE		Tracking Area Update Request		w/ jammer		
Vodafone								
Iliad								

Table 2: Analysis of operator impact on IMSI catching

Finally, our work was limited to 4G deployments, as the commercial roll out of the 5G technology has not yet systematically started to date (and due to COVID19 mobility limitations, we could not access experimental 5G sites). Our obvious next step is to assess whether 5G deployments are more resilient than 4G ones with respect to IMSI catching attacks - even if tailored solutions such as SUCI concealment have been standardized, we strongly suspect that location privacy will remain a widely open issue also in 5G.

ACKNOWLEDGMENTS

This work was partially supported by the European Union's Horizon 2020 programme under Grants No. 871249, project LOCUS and No. 830892, project SPARTA.

REFERENCES

[1] Andy Lilly. Imsi catchers: hacking mobile communications. *Network Security*, 2017(2):5 – 7, 2017.

[2] Haibat Khan, Benjamin Dowling, and Keith M Martin. Identity confidentiality in 5g mobile telephony systems. In *International Conference on Research in Security Standardisation*, pages 120–142. Springer, 2018.

[3] Mohsin Khan, Philip Ginzboorg, Kimmo Järvinen, and Valtteri Niemi. Defeating the downgrade attack on identity privacy in 5g. In *International Conference on Research in Security Standardisation*, pages 95–119. Springer, 2018.

[4] Chuan Yu, Shuhui Chen, and Zhiping Cai. Lte phone number catcher: A practical attack against mobile privacy. *Security and Communication Networks*, 2019:7425235:1–7425235:10, 2019.

[5] S. Hussain, O. Chowdhury, S. Mehnaz, and E. Bertino. Lteinspector: A systematic approach for adversarial testing of 4g lte. *Network and Distributed Systems Security (NDSS) Symposium 2018*, Feb 2018.

[6] M. Labib, V. Marojevic, J. H. Reed, and A. I. Zaghloul. Enhancing the robustness of lte systems: Analysis and evolution of the cell selection process. *IEEE Communications Magazine*, 55(2):208–215, 2017.

[7] Roger Piqueras Jover. Lte security, protocol exploits and location tracking experimentation with low-cost software radio, 2016.

[8] Roger Piqueras Jover. Security attacks against the availability of lte mobility networks: Overview and research directions. *2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pages 1–9, 2013.

[9] Adrian Dabrowski, Nicola Pianta, Thomas Klepp, Martin Mulazzani, and Edgar Weippl. Imsi-catch me if you can: Imsi-catcher-catchers. In *Proceedings of the 30th Annual Computer Security Applications Conference*, ACSAC '14, page 246–255, New York, NY, USA, 2014. Association for Computing Machinery.

[10] Altaf Shaik, Ravishankar Borgaonkar, N. Asokan, Valtteri Niemi, and Jean-Pierre Seifert. Practical attacks against privacy and availability in 4g/lte mobile communication systems, 2015.

[11] S. F. Mjølsnes and R. F. Olimid. *Easy 4G/LTE IMSI catchers for non-programmers*, volume 10446 LNCS of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2017.

[12] Third Generation Partnership Project (3GPP). In *TS 36.300 Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2*, 2020.

[13] Third Generation Partnership Project (3GPP). Service request procedures. In *TS 23.401 General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access*, 2020.

[14] Software Radio Systems. srslte, your own mobile network. https://www.srslte.com, Last accessed on 2020-07-29.

[15] OpenAirInterface Software Alliance (OSA). Openairinterface, 5g software alliance for democratising wireless innovation. https://www.openairinterface.org, Last accessed on 2020-07-29.

[16] YaYa Brown, Cynthia Teng, and Alexander Wyglinski. Lte frequency hopping jammer, Dec 2019.

[17] V. Marojevic, R. M. Rao, S. Ha, and J. H. Reed. Performance analysis of a mission-critical portable lte system in targeted rf interference. In *IEEE Vehicular Technology Conference*, pages 1–6, 2018.

[18] Tektronix. Rsa3408a real-time spectrum analyzers. https://www.tek.com/datasheet/rsa3408a-real-time-spectrum-analyzers-datasheet, Last accessed on 2020-07-29.

[19] Ettus Research, National Instruments. Usrp b210. https://www.ettus.com/all-products/ub210-kit/, Last accessed on 2020-07-29.

[20] Open source mobile communications, Osmocom. Open source mobile communications. https://osmocom.org/, Last accessed on 2020-07-29.

[21] Laurent Thomas. Open cells project. https://open-cells.com/, Last accessed on 2020-07-29.