# IMSI Probing: Possibilities and Limitations

Daniel Fraunholz[1(✉)], Dominik Brunke[1], Simon Beidenhauser[1], Sebastian Berger[1], Hartmut Koenig[1], and Daniel Reti[2]

[1] ZITiS, Communications Department, Munich, Germany
Daniel.Fraunholz@zitis.bund.de
[2] German Research Center for Artificial Intelligence, Kaiserslautern, Germany

**Abstract.** Mobile networks are vital for modern societies. Recent generations of mobile communication systems have introduced increased security and privacy features to enhance their trust and reliability capabilities. Several well-known vulnerabilities, however, have not been mitigated due to design choices so far. An example is the IMSI probing attack considered in this paper which exploits vulnerabilities of the paging mechanism in mobile networks, whereas the reference to the International Mobile Subscriber Identifier (IMSI) is arbitrary and misleading. The IMSI probing attack can be used to locate and track mobile phones to infer the behavior their users. Although first published already ten years ago, it can be applied to all cellular network generation up to the upcoming 5G Stand Alone. The attack requires a certain effort to be successful. It is therefore considered less practicable. In this paper, we present an in-depth analysis of the IMSI probing attack and discuss the likelihood for its success including the required presumptions. We show that under certain conditions the attack may be successful and that the success rate can significantly be improved. Finally, we present a novel attack variant that doubles the success rate and enables location determination at the cell granularity level.

**Keywords:** IMSI probing attack · Privacy violation · 4G · 5G SA · Paging · RRC protocol

## 1 Introduction

Mobile devices have become ubiquitous in recent decades. The number of services offered on mobile devices is increasing with an astonishing speed. Sensitive services, such as online banking and health data capturing, have become standard. The growing impact of mobile technologies on human beings improves their life quality in various ways, but they also provide a lucrative target for misuse by exploiting their vulnerabilities.

With each generation, mobile telecommunications networks have become more and more secure. At the very beginning of the development the lack of mutual authentication between base station and the mobile phone (UE, user


H. P. Reiser and M. Kyas (Eds.): NordSec 2022, LNCS 13700, pp. 80–97, 2022.
https://doi.org/10.1007/978-3-031-22295-5_5

equipment) in 2G was the main reason for the well-known privacy attacks based on false base stations (FBSs) [9]. For this attack type, solutions were discussed through all mobile network generations, e.g., the authentication of System Information Block (SIB) broadcast messages to prevent the use of false base stations. With 5G Stand Alone (5G SA), the use of the permanent subscription identifier has been restricted over the air interface which significantly limits the possibility of privacy attacks. Another example of a vulnerability existing since 3G is the so-called *Linkability of Failure Messages* (LFM) attack [2] which exploits a weakness in the authentication and key agreement (AKA) procedure. With paging, used to wake up disconnected mobile devices, the situation is similar. The IMSI probing attack exploits the fact that paging messages include the M-TMSI (Temporary Mobile Subscriber Identity) of the phone to wake up, which is typically not changed frequently [6]. The attack was first published in 2012 [8], but it was not considered a severe attack at that time, since the same result could readily be achieved with other attacks. The reference to the International Mobile Subscriber Identifier (IMSI) was arbitrarily chosen for this attack because the initial attack scenario published did not use the IMSI as target identifier but the phone number. In principle, the attack can use any identifier to initiate interactions with a target device.

In 4G and 5G, the attack can be applied for degrading privacy guarantees. So in 5G SA, for instance, an attacker should not be capable to verify the presence or absence of a user because no permanent identity is transmitted over the air, but the paging vulnerability allows one to link the reuse of temporary identities, e.g., the GUTI (Globally Unique Temporary Identifier), to verify the presence and absence of a certain user. The 3rd Generation Partnership Project (3GPP) - the relevant body for the specification of cellular networks - has been already aware of this problem and proposed to change the 5G GUTI after certain events (3GPP TS33.501, 6.12.3), e.g., after receiving paging messages (GUTI re-allocation). However, 3GPP has not specified exactly how the GUTI reallocation should be implemented. Recent research has found out that several operators did so even before it was required by the standard [6]. In many cases, however, the new GUTI was based on the previous one with only slight and predictable modifications. The 3GPP standard (3GPP TS33.501, 6.12.3) explicitly states that the change of the GUTI is up to the operator. So, the re-allocation of GUTIs was not implemented as required in the three investigated 5G SA networks in China, rendering IMSI probing attacks still feasible [10].

In this paper, we present an in-depth analysis of IMSI probing attacks and examine the effectiveness as well as the limitations of the attack. This is necessary because no such investigation has been performed since the introduction of the attack. We discuss the conditions when the attack may be successful and present an attack variant that doubles the success rate and allows for a location accuracy at cell granularity. Currently, the use of paging messages is the only available probing method in public. Therefore, we refer to paging-based probing here. Paging messages can only be triggered when the target phone has no radio connectivity, i.e., when it is in the so-called *idle* mode. We explore the idle mode

with and without user interactions, called *active* and *passive* phone, respectively. The remainder of the paper is organized as follows. In the following Sect. 2, we introduce the IMSI probing attack and discuss various aspects of its use. Section 3 analyzes the behavior with passive phones in a testbed consisting of 4G and 5G mobile network lab environments and commercial off-the-shelf (COTS) mobile phones. Section 4 then considers the behavior with active phones. For this, a user behavior model has been developed to enable simulation-based analysis of the IMSI probing attack. The impact of applications installed on mobile phones is analyzed in depth also in this section. In Sect. 5, we present measurements from public land mobile networks (PLMNs) to augment the results of the lab network and the simulation. Based on these analyses, we quantify the overall success probability of IMSI probing attacks in Sect. 6 and analyze the impact of the most relevant parameters for the success probability is analyzed. In Sect. 8 we present a novel, more efficient attack vector. Some final remarks conclude the paper.

## 2 IMSI Probing Attack

The goal of the IMSI probing attack is to verify whether a given device is currently in a certain cell or tracking area. This can be used to track a person in certain area (e.g. city), to observe its movement in a shopping area, or to prepare further activities, e.g., to check whether a person is at home to prepare a burglary. The presence can be determined with cell (200 m–20 km) or tracking area (multiple cells) granularity, respectively.

The software for this attack is freely available. The hardware cost for commercial-of-the-shelf software-defined radios, which are sufficient for launching the attack, is below $2000. Only minimal programming knowledge is required to adapt open-source software like srsRAN [14], OpenAirInterface [11], or other open-source monitoring tools, such as Falcon [5], LTEEye [7], OWL [3], and C3ACE [4] for the attack. Qualcomm baseband chips offer access to layer-2/3 messages via the DIAG protocol. Open-source frameworks, such as QCsuper [12], built upon this DIAG protocol can also be used.

### 2.1 Attack Presumptions

To launch the attack the attacker has to trigger a paging message. This can be done by sending a message to the target device. For this, the attacker needs to know at least one identity of the target device, e.g., the social media account or others, as discussed below. The attacker must further be connected to the mobile network of the target device, either directly or via a proxy network, e.g., the Internet. Moreover, the attacker must be able to passively monitor and analyze the paging channel of the given cell or of at least one cell within the tracking area in which the target device's presence should be checked. The analysis itself can be either performed in real time or offline afterwards. Finally, it requires that the attacker is close to the target device. In practice, this can be several

hundred meters in urban cells and some kilometers in rural environments. It is difficult to determine the proximity in advance as beam propagation is subject to different factors, e.g., weather or position. The distance to the target device can be increased using high-end antennas and other monitoring equipment.

**Target Device Identities.** Attacks on mobile devices often use different identities for their purpose, such as GUTI, IMSI, IMEI (International Mobile Equipment Identity), or MSISDN (Mobile Subscriber Integrated Services Digital Network Number). IMSI probing, in contrast, can use any identity that trigger paging messages. From the attacker's perspective, this is a substantial advantage because social media or e-mail accounts can be exploited as well. Additionally, the attack can also be performed without knowledge of any mobile network-related identity, if other identities are known as shown by Shaik et al. [13]. Thus, the attack can be used in versatile scenarios.

### 2.2 Triggering Paging Messages

Assuming that all presumptions of the attacker model are met the attack can be launched triggering a paging message. For this, the attacker has to select an appropriate triggering method based on the available identities. Almost all instant messaging applications trigger a paging message at the base station. Since an attacker usually does not know the instant message apps installed, default applications, such as telephony or SMS, can also be used for triggering. There is also the option to secretly trigger paging messages. Shaik et al. [13] proposed the use of the typing notification of Whatsapp for this because it is not shown on the target device, even if the application is used by the device owner during the reception of the paging message. There is one exception when the attacker's Whatsapp account is already known to the target Whatsapp's one. In this case, the attacker's account is indicated as "*typing*", i.e., the account is currently writing a message. A similar behavior is also expected in other instant messaging apps. Another less concealed option is the use of Facebook [13]. This proposal relies on the fact that the Facebook application does not prominently show messages of unknown users, but instead stores them in a folder, called "*Other messages*", not be seen by the user. In addition, an attacker must know in advance whether various additional conditions are fulfilled: does the target use the application, also the mobile version of it, and whether it is active. Therefore, it is most likely that attacks prefer to use phone or vendor default applications which require less assumptions on installed applications. We did not examine any other means in the context of this investigation to trigger paging messages in an open or concealed manner.

### 2.3 Connection Mode

The connection mode is of crucial importance for the success of the attack. It is idle or connected. A phone is in the *connected* mode when it has established

a layer-2 (Access Stratum, AS) and a layer-3 (Non Access Stratum, NAS) connection. It is in the *idle* mode when only a layer-3 (NAS) connection has been established but no layer-2 (AS) connection. Only if the phone is in the idle mode the attack is successful. Therefore, the switching between these two modes has to be detected. This has to be done by simultaneously monitoring the paging channel of the target cell or tracking area, respectively.

### 2.4 Monitoring the Paging Channel

Regarding the monitoring two kinds are distinguished: regular and smart paging. *Regular paging* sends paging messages in each cell of a tracking area, whereas *smart paging* only in the cell in which the mobile device is located, i.e., with smart paging, the presence of the target device can be verified with cell-granularity. An attacker can easily figure out whether smart paging is used by triggering a paging message and monitoring a neighbor cell of the same tracking area. If the paging message is received only tracking area-granularity can be achieved.

In order to recognize the paging messages triggered by the attack the paging messages must be sent with defined sending frequency. This frequency can arbitrarily be chosen by the attacker, e.g., every 10 s, and can also be non-equidistant. In this case, the paging channel analysis tries to recognize whether there are messages to a certain device that follow this pattern.

### 2.5 Result Verification

Regarding the sending two types of errors can be distinguished. If the frequency pattern is detected although the target device is currently not located in the cell or tracking area we have a *false positive* (FPs). This type of error occurs if a short pattern has been chosen. Kune et al. [8] triggered a paging message with a probe $j$ ($j \in 1 \leq j \leq n$) and stored any temporary identifier in a set $I_j$ that were addressed in paging messages within a specified time interval $t_{min} \leq t \leq t_{max}$ after triggering the paging message to the target device.

$$I_j = \begin{cases} TMSI_t, & t_{min} \leq t \leq t_{max} \\ \emptyset, & \text{otherwise} \end{cases} \tag{1}$$

This step was repeated $n$ times until only one identifier occurred in each of the stored set $I$, i.e. the intersection of temporary identifiers send after each probe is the temporary identifier of the target device ($I_1 \cap I_2 \cap ... \cap I_n$). While using a distinct pattern, the likelihood of a false positive is lower than in the approach of Kune et al. Simple repetitions of the pattern further decreases the likelihood. *False negatives* (FNs), in contrast, occur when the device is located in the monitored cell or tracking area, but the frequency pattern is not found on the paging channel. This happens if the target device is not in idle mode when the attack is carried out or the monitoring system misses to capture the message. It is enough to fail the attack, when one paging message to the target device is missing in at least one set because there is no identifier in this case that occurs

in each list ($I_1 \cap I_2 \cap ... \cap I_n = \emptyset$). Pattern matching algorithms that employ thresholds for list comparison achieve more reliable results. After sending 10 probes, for example, it is sufficient to have a single identity in 80% of the lists, i.e., eight lists to conclude that the target is present, while the algorithm of Kune et al. would fail in this example. We apply the results of this examination to develop optimized attack success strategies which we present in Sect. 6.

## 3 Analyzing Idle Behavior with Passive Phones

The most significant limitation of the IMSI probing attack is that an attacker does not know whether the target device is in idle mode. When it is not in idle mode the attack results in a false negative, i.e., the target presence cannot be verified even if the target device is in the cell or tracking area. We consider two cases for the further analysis: (1) the device is not in use at all during the attack. Even then, there is a probability that the device is not in idle mode. This case is referred to as *passive mode* and is considered in this section. (2) The device is in use during the attack, e.g., for a telephone call or for browsing the Internet. This is referred to as *active mode* and is analyzed in a subsequent section.

### 3.1 Test Setup

To assess the passive mode we set up a testbed to measure idle times using an Amarisoft Callbox Classic [1]. The Callbox consists of a base station with 5G SA support, the respective core, and an IP Multimedia Subsystem (IMS). Base station and core also support 4G and multiple cell scenarios, such as 5G NSA. In our experiments, we used 5G SA for phones that support it, otherwise 4G. The phones were Android ones from various vendors, e.g., Samsung, Google, Huawei, Oppo, and Sony. The Callbox also possesses other basic monitoring capabilities, e.g., the possibility to verify whether a device has a layer-2 identifier (Cell-Radio Network Temporary Identifier, C-RNTI) associated with. If no such identifier is associated, the device is in idle mode and possesses only the layer-3 identity (GUTI). A script has been written to monitor the association of a layer-2 identifier in 1 s periods. Thus, the idle times of any device connected to the test setup could be determined. All phones had a factory-reset prior to experimentation and no additional software was installed. Any software installed on the phones can potentially alter the idle time behavior because any software with network connection can initiate a communication and thus forcing the phone to switch from idle to connected. It can be assumed that the more software, i.e., apps, are installed on the target phone the more likely is that the device is not in the idle mode, so that the presence verification leads to a false negative. Because only default software was installed on the phones, the results can be interpreted as best case scenarios in which the attack can be launched on each phone model.

### 3.2 Idle Time Behavior of Passive Phones

We found out that the examined mobile phones (Phone A, Phone B, Phone C, Phone D, Phone E) were between 89% and 96% of the overall monitoring period


**Fig. 1.** Idle behavior of the investigated phones with default configuration

in the idle mode. An exception was Phone E* for which we used Phone E with disabled data connectivity and an idle ratio of 99.8%. In all cases other than E*, an increased connectivity period in the first minutes of the experiment was observed followed by a cyclic periodicity of idle time behavior after about 5 min. An exemplary visualization of the idle behavior of the investigated mobile phones is given in Fig. 1. The shortest observed idle time period was 1s for Phone A, the longest 4972 s for Phone E* with disabled mobile data connectivity. With enabled mobile data connectivity, Phone E had a maximal idle duration of 530 s, indicating the significant influence of the data connectivity. An overview of the results is given in Table 1. When the mobile data connection between the mobile phone and the network was disabled, the mobile phone connected only once during the monitoring. From this, it can be derived that the phones' idle phases are significantly correlated with the Internet connectivity, which is controlled in the upper layers of the phones' protocol stack. It can also be concluded that the idle behavior of passive phones in default configuration is characterized by short but frequent connection phases, which are temporally grouped in many cases.

### 3.3 Analysis of Process Behavior in Passive Mode

In order to assess which processes are relevant in the idle and connected mode, a mobile application has been developed to collect information about the network packets and the corresponding processes. The app uses the *proc* file system to gather this information. It requires a rooted phone for installation. Two 24h experiments were conducted. In the first experiment, the test phone was in the default configuration, i.e., no apps (besides our monitoring app) were installed. A total of 6287 packets was observed resulting in an average of 4.4

**Table 1.** Overview of the idle time behavior of selected mobile phones with default settings and default software installed, Phone E* is like Phone E but with disabled data connectivity

| | Phone A | Phone B | Phone C | Phone D | Phone E | Phone E* | Av. wo E* |
|---|---|---|---|---|---|---|---|
| Exp. duration | 4271 s | 3592 s | 3870 s | 3942 s | 4095 s | 6790 s | 3954 s |
| Number conn. | 34 | 32 | 17 | 12 | 27 | 2 | 24,40 |
| Idle conn. ratio | 91% | 89% | 91% | 96% | 90% | 99,8% | 91% |
| Av. conn. phase | 11 s | 13 s | 21 s | 15 s | 16 s | 6 s | 15.20 s |
| Min. conn. phase | 4 s | 10 s | 10 s | 10 s | 10 s | 1 s | 8.80 s |
| Max. conn. phase | 26 s | 27 s | 76 s | 40 s | 73 s | 10 s | 48.40 s |
| Av. idle phase | 118 s | 96 s | 207 s | 290 s | 136 s | 2260 s | 169.40 s |
| Min. idle phase | 1 s | 2 s | 3 s | 5 s | 2 s | 290 s | 2.60 s |
| Max. idle phase | 287 s | 328 s | 821 s | 821 s | 530 s | 4972 s | 557.40 s |

packets per minute. However for IMSI probing, the temporal distribution is significant. Therefore, we further analyzed the phases in which no packets were sent. We found out that most packets (6015) had a distance less than 1s to the preceding packet. There were several phases in which no packets were sent, up to 1680 s, which provide multiple options for successful IMSI probing attacks. Google services, such as *GMS persistent* and *quicksearchbox*, accounted for the majority of network traffic and dominated the connectivity phase. Since cellular phones are usually not in default configuration, the opportunities for IMSI probing may differ from the previously discussed scenarios. To take the impact of mobile applications on the IMSI probing success probability into account we installed and registered in the second experiment a number of popular applications on the phones. The complete list is given in Appendix A. Where necessary, user accounts were registered. However, no further interactions with apps were induced, e.g., no subscriptions, friend requests, likes etc. Interestingly, the behavior changed drastically. With these popular applications, the number of packets transferred in 24 h increased almost 35 times, resulting in a total of 217748 packets transferred and on average of 151 packets per minute. As expected, the number of idle phases reduced significantly to only 15 phases between 20 s and 30 s. The maximal idle duration observed was 25.4 s. The effective time is even smaller in reality, since we measured the time between two consecutive packets in the experiment. The actual idle phases are initiated by the network after a certain threshold (phone's inactivity timer) which further reduces the time slot available for the attack. On process-level, the most significant change is that Google services have no significant share in the number of packets sent over the network. Instead the process *com.zhiliaoapp.musically* (60.76%) and *com.zhiliaoapp.musically:push* (5.21%) account caused more than 65% of the observed packets. As the two processes can be attributed to the *TikTok* application, this observation indicates that an installed *TikTok* application significantly reduces the probability of successful IMSI probing attacks. With 1.32% of the

observed packets, the aforementioned *Google GMS* service is ranked third. This process also almost doubled the number of packets sent from 1589 to 2885, suggesting that at least one of the installed applications affects Google services as well.

## 4 Analyzing Idle Behavior with Active Phones

IMSI probing attacks can only be successful when the target phone besides being in idle mode is close to the target person. However most likely, the phones are actively used over time thus reducing the idle mode times. To enable an in-depth analysis of IMSI probing attacks under these conditions the idle behavior model of the mobile phones must include the user behavior. For this purpose, we have developed an additional user behavior model based on the following assumptions: The average screen time of a person varies by many factors, such as weekday, habits, and age. To evaluate the effectiveness of the IMSI probing attack several assumptions about the target must be made. We assume here that an average person spends about 3:15 h with its phone per day and that this time spreads over 58 sessions [15], 70% of them are shorter than 2 min, 25% are between 2–10 min, and 5% longer than 10 min. Moreover, the sessions are equally distributed between 8 a.m. and 7 p.m. The time spent on smartphones is significantly lower between 9 p.m. and 7 a.m., what increases the probability of a successful attack. It was also found that on average the time spent on mobile phones is less on weekdays than on weekends [15]. During the pandemic the screen time increased about 20% to 30% rendering probes less likely to be successful. A visualization of an exemplary idle behavior of the developed model of such a behavior for an 8h period is contained in Fig. 2 (lower graph, cyan and magenta solid lines).

## 5 Empirical Evaluation in Real World Mobile Networks

To further quantify the success probability of IMSI probing attacks other parameters were examined. An important parameter is the number of consecutive probes required for a successful attack. The capture rate of the available monitoring systems is another parameter that needs to be considered because there is a likelihood of probes being missed.

### 5.1 Experimental Environment

Figure 3 depicts the number of paging messages in 10 s time interval over a 24 h period in the network used for the experimental evaluation. The tracking area is located in the north-east metropolitan region of Munich, but it is no longer part of the Munich city. As it can be seen, up to almost 1400 paging messages per 10 s interval need to be captured and processed by the monitoring system.

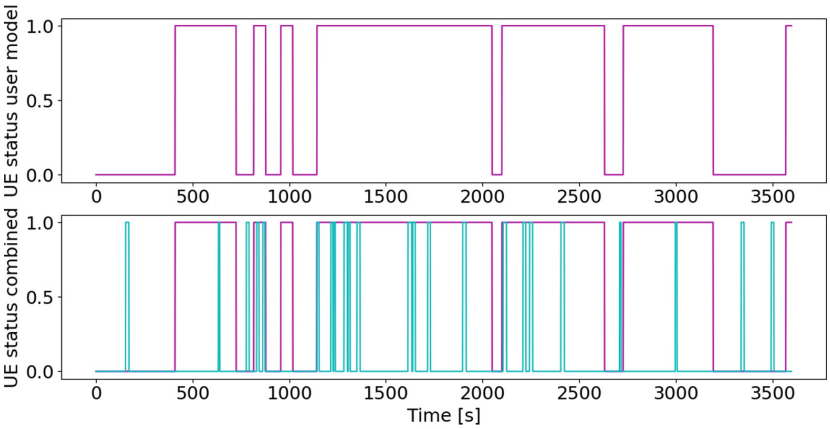



**Fig. 2.** Upper graph: Exemplary visualization of the idle and connected times of the assumed user behavior model. Lower graph: Exemplary visualization of the idle and connected phases resulting from the combined user and processes model

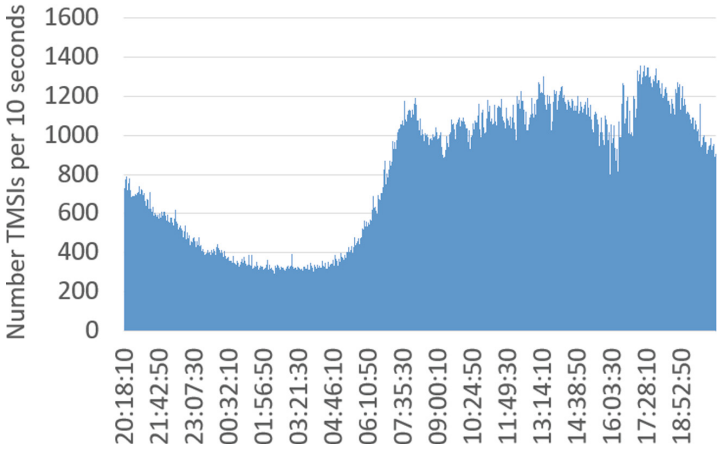



**Fig. 3.** Visualization of the number of received temporal user identities (TMSI) per 1680 s over the course of about 24 h in a cell in the northern metropolitan area of Munich.

### 5.2 Required Number of Probing Repetitions

The determination of the required probe number for a successful identification of a phone is crucial for further investigations. This is because the number of required probes in combination with the duration of a single probe determines the total idle period necessary for a successful attack. The required number of probes is not known in advance and can only be estimated. As previously described, an attacker needs to define a time window in which paging messages are collected. This must be repeated until only one or none GUTI is included in

each observed window. The number of paging messages per window depends on the window size and the network usage behavior which depends, as argued above, on daytime, habits, and so on. To quantify the IMSI probing success probability we performed several attacks (as described in [8]) against a known target phone on a working day at noon in an urban region in the network of a major German operator. In general, we needed about 6 probes at this time to receive only one GUTI included in each monitored time window. This corresponds to the results from the literature [8,13].

### 5.3 Capture Rates

An attacker cannot know whether all paging messages in a monitored time window are captured or not. The capture rate depends on multiple factors. In our experiments we deployed the Falcon monitoring system [5] to collect the paging messages and to determine the capture rate. We found that the value significantly depends on the signal quality and the network usage. The achieved capture rates were between 70% and 99%. Mobile phones do also not have a capture rate of 100% because the base station repeats the paging messages several times to reduce the likelihood of missing them. This effect increases the capture rate of the monitoring system.

## 6 Attack Success Probability and Impact Factors

To determine the probability of a successful IMSI probing attack two cases have to be distinguished. (1) The success probability is estimated using the developed model based on the experimental results of the idle behavior of the mobile phones that are not in use (passive) and have no applications installed (default configuration). (2) Based on this, user interactions with a mobile phone are included in the quantification. False positive probabilities are not considered here because this would require assumptions on third-party devices connected to the cell or tracking area in which the attack is launched. In the two cases, the success probability depends on the number of required probes, the time window to monitor the probes, the capture rates, and the idle times of the mobile phones. Given that an attacker cannot know the exact idle behavior of the mobile phones, any attack time is equally sufficient.

### 6.1 Passive Mobile Phone with Default Configuration

In our experiments, we used the Amarisoft Callbox as test network and the aforementioned script to monitor the phone's idle times. Additionally, a Python script was written to use the JSON API to periodically send SMSs every 20 s via the test network to the target phone. 20 s were chosen because the network typically releases the phone after 5–15 s of inactivity and 20 s ensure that the previous SMS does not influence the subsequent ones. The script used the JSON API of the Callbox. The paging channel of the test network was again monitored

**Table 2.** Overview of the idle time behavior of selected mobile phones with default settings and default software installed

| | Ph. A | Ph. B | Ph. D | Ph. E | Ph. E* |
|---|---|---|---|---|---|
| True-pos | 98 | 128 | 96 | 100 | 108 |
| False-pos | 7 | 19 | 27 | 14 | 1 |
| True-neg | 2823 | 2767 | 2295 | 2151 | 2053 |
| False-neg | 45 | 18 | 9 | 12 | 10 |
| Precision | 0.933 | 0.871 | 0.780 | 0.877 | 0.991 |
| Recall | 0.685 | 0.877 | 0.914 | 0.893 | 0.915 |
| F1-score | 0.790 | 0.874 | 0.842 | 0.885 | 0.952 |

with the Falcon tool. A statistical evaluation of the complete experiment is given in Table 2.

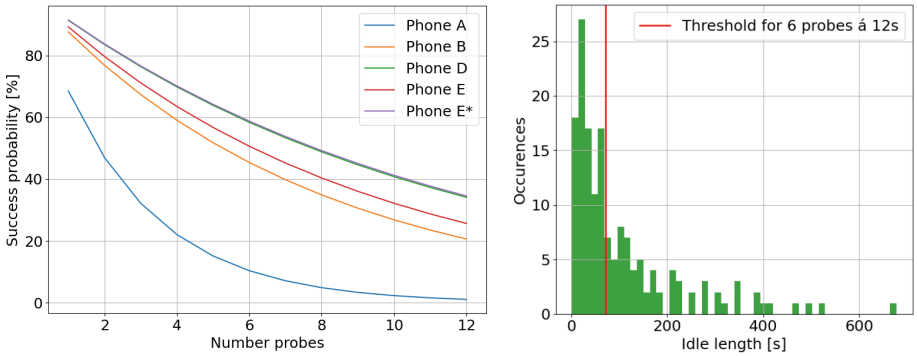
The success rate for IMSI probing attacks against passive phones in default configuration varies significantly depending on the number of probes required and the phone model. This is because the false negative classifications, i.e., the missed probes, were significantly higher than for other phones. The reason for this is the shorter idle time or the more frequent connection phases, respectively, since the same monitoring system was used in all experiments. The success probability for one probe is about 70% and reduces to about 1% for 12 consecutive probes for Phone A. All other phones had a success probability of about 90% for one and between 20% and 35% for the 12 consecutive probes. The success probability for 7 consecutive probes, as identified as typical number in the experiment, is about 7% for Phone A and between 40% and 53% for the other phones.

From this experiment, the time between a received SMS and the return to idle mode was estimated to be about 12 s. This is important to quantify the waiting period between two consecutive probes in the subsequent evaluation.

### 6.2 Active Mobile Phone with Default Configuration

To determine the success probability for active phones the idle behavior model as described in Subsect. 4 was investigated. An exemplary visualization of the distribution of the connections and idle phases is shown in Fig. 4b, whereby the threshold of 7 consecutive probes with a 12 s window between probes is marked with a vertical solid red line.

It shows that there are several idle phases that are long enough to perform a successful IMSI probing attack. An attacker, however, cannot know the begin or end of these phases, i.e., the likelihood of a successful attack is the same at any point in time from the attackers perspective. Therefore, the attacker's probability to launch the attack is equally distributed over the considered time. Consequently, the number of required probes and the capture rate of the monitoring system are the two parameters that cannot be known in advance because they depend on the cell or tracking area load during the attack. An exemplary




(a) (b)

**Fig. 4.** (a) Visualization of the relation of the attack success probability and the number of probes required for presence verification. Phone D (green) and Phone E* (purple) are almost similar. (b) Histogram of the resulting length of idle times in the user behavior simulation and the threshold for the minimal idle time required for a successful attack (6 consecutive probes á 12 s) (Color figure online)

visualization of the success rates for several required numbers of probes and capture rates is given in Fig. 5.

Several of the considered scenarios have a success probability close to zero. This is because the availability of idle phases is limited if the user behavior is taken into account. The required number of consecutive probes linearly increases the needed duration of the idle phase. The gradient depends on the minimal time between two probes which is the sum of the time between sending the probe, receiving by the phone, connecting to the network, receiving the data (i.e., the SMS in our experiment), and switching back to idle mode after a short inactivity period. In best-case scenario, the IMSI probing attack only achieves a success probability of 36.88%. This is the case with a 100% capture rate and only 4 probes being necessary to differentiate the target phone from all other phones in the cell or tracking area. In our experiments with different provider networks, four probes was the minimum number of probes that were required for a successful attack.

## 7 Optimizing the Attack Success Probability

The results obtained suggest that IMSI probing is not a reliable attack technique at all. If we consider the typical user behavior with eight required probes and a capture rate of 99%, the success probability is below 20%. Moreover, this only holds if no applications are installed. The in-depth process analysis revealed that the required idle phases are even less if applications are installed on the phone, which is the normal case in real world scenarios.

The success probability is determined based on the assumption that the pattern detection algorithm cannot handle errors, leading to an unsuccessful

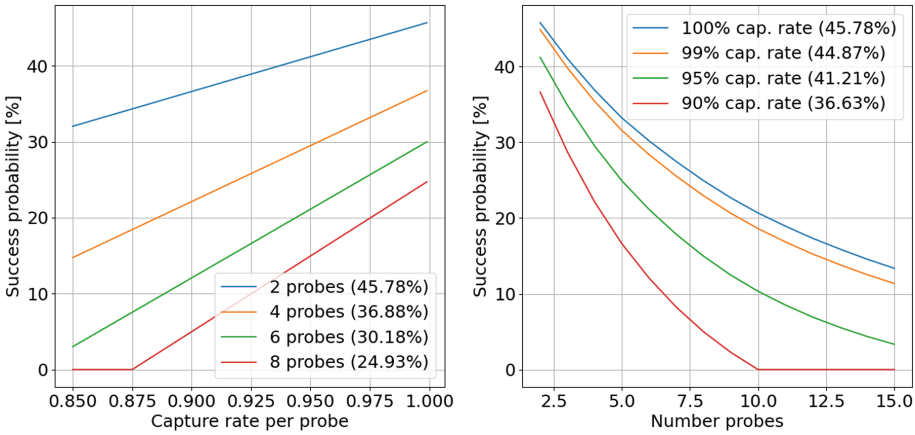



**Fig. 5.** Left: Impact of the capture rate on the success probability for different number of required probes. Right: Impact of the required number of probes on the success probability for different capture rates.

probe if one probe is missed (e.g., because of the capture rate or because the phone was not in idle mode). More advanced pattern detection algorithms can take these factors into account and increase the success probabilities as follows. There are two strategies. (1) An attacker can simply repeat the IMSI probing attack after completion. In this case, the overall success probability of the attack is generally determined by the discrete binominal distribution $P(X \geq 1) = P(X = 0) + ... + P(X = n)$, whereby $P(X) = \binom{n}{k} p^k (1-p)^{n-k}$ and $P$ is the overall success probability, $n$ the number of attacks, $k$ the number of successful attacks and $p$ the success rate of a single probe. For example, if the attack success rate is 40%, the probability of at least one successful attack after 4 attacks is about 87%. This strategy is limited by the overall number of idle phases that are long enough to perform a successful attack, i.e., if there are no idle phases long enough for a successful attack the attack fails regardless of the number of repetitions. In all other cases, this strategy will succeed given infinite time. (2) A more advanced strategy is to adapt the pattern detection during the attack. As presented in Table 2, the recall for a single probe is about 90% (excluding Phone A), i.e., there is only a 10% probability to miss a paging message. If one paging message is missed the intersection does not converge to one remaining identity but to none. In such a situation, additional probes can be sent to verify the result. The formula from above applies here as well. For example, if the likelihood of a single successful probe is 90%, the likelihood of seven consecutive successful probes is about 48%, i.e., when seven probes were sent the probability that at least one was missed is about 52%. However, by simply sending another probe, i.e., eight probes are sent and only seven are required to be successful, the overall success probability reaches 81%. Adding a ninth probe, i.e., nine probes are sent and only seven of them are required for success, would increase the

overall attack success probability to 95%. This strategy is more effective than the first one presented, since only a single probe has to be repeated instead of the complete probe sequence. The additional probes in the two strategies can be sent at any time. The only limitation is that the identity can change over time or that the target phone changes its location.

## 8 A Novel Attack Technique for IMSI Probing

Currently only paging messages are used as attack vector for IMSI probing attacks. We present here another attack vector - *the RRCConnectionSetup message* - which achieves a better success rate. RRCConnectionSetup is a layer-2 downlink message that is part of the *Radio Resource Control* (RRC) protocol. It is sent from the phone to the base station in the process of establishing Access Stratum connectivity.

Performing the IMSI probing attack using the RRCConnectionSetup message has two major advantages. (1) The location accuracy can be improved from tracking area granularity to cell one. RRCConnectionSetup messages are not sent in the tracking area but only on layer-2 of the cell the phone is in. For paging messages, this granularity has been only possible up to now when smart paging (i.e., paging messages are only sent in the last known cell instead of the last known tracking area) was applied in the target cell. (2) The number of probes reduces drastically because the number of RRCConnectionSetups is generally lower than the number of paging messages. This is because RRCConnectionSetup messages are only sent when the paging procedure is successful and the phone establishes a connection, or when the phone establishes a connection without prior paging.

For the evaluation of this attack technique, we performed an experiment comparing paging- and RRCConnectionSetup-based attacks. We performed twenty consecutive attacks and compared the efficiency of the two attack vectors regarding the required number of probes to verify the target presence which ultimately impacts the attack success probability, as argued above. It was found that paging-based IMSI probing attacks required between 4 and 10 probes with on average 5.6 probes and a standard deviation of 1.2 probes. For the RRCConnectionSetup-based attack, 2 or 3 probes were needed with on average of 2.2 probes and a standard deviation of 0.4 probes. These results show the greater efficiency of the new attack vector. The number of messages collected during the specified time window (4.5 s) after the probe was ten RRCConnectionSetup messages compared to on average ninety paging messages before. This is the reason why the new attack method converges faster leading to a better correlation between the number of required probes and the number of messages collected after each probe.

Based on these experimental results, we further assessed the attack success probability of the two vectors. We run hundred simulations with a given capture rate of 95% and a minimal waiting period of 12 s between the probes, and a minimal required number of two probes for RRCConnectionSetup and

six for paging. Our model showed an average attack success probability for the RRCConnectionSetup-based method of 39.72% with a standard deviation of 4.42% and for the paging-based one 20.71% with a standard deviation of 3.00%, i.e., the number of RRCConnectionSetup messages is about one tenth of the number of paging messages in the same time window (90 vs. 10 messages). This results in a reduction of about one third in the number of probes required (six vs. two probes) which itself doubles the success probability of the IMSI probing attack (20% vs 40%).

The RRCConnectionSetup attack vector, however, increases the uncertainty of the calibration of the time window after the probe. This is because measuring paging messages is subject to uncertainties of the phone, the API that initiates the probe, and the network that receives the probe and subsequently sends the paging message to the target phone. For monitoring RRCConnectionSetup messages, the uncertainty is extended by the target phone receiving the paging message and subsequently sending the RRCConnectionRequest message back to the base station which then in turn responds with the RRCConnectionSetup to the phone. We found that the delay of this process is on average 0.4 s, while the standard deviation increases from 0.77 s to 1.06 s. This effect must be compensated by adjusted time windows when performing the attack. For paging-based IMSI probing, the calibration of the time window is also necessary.

We identified a limitation in our experimental setup because the Falcon tool [5] captured paging messages with an accuracy of about 98.8% and RRCConnectionSetup messages with an accuracy of about 76%. This may be caused by the increased complexity in decoding for the RRCConnectionSetup messages. We made the measurements from the phone's baseband chip which we accessed via the Qualcomm DIAG protocol with QCsuper [12] as baseline. The ratio of messages received at the baseband chip and SMS triggered was 92.8% for both paging and RRCConnectionSetup messages. The remaining messages (7.2%) were not being sent, since the phone was not in idle mode. We mitigated this limitation by considering positive results only. This is valid because the target phone was present in the monitored network and only the number of required probes for verification was evaluated.

## 9 Conclusions

In this paper, we have studied on the effectiveness of the IMSI probing attack that allows for invading user privacy. With the advance of cellular network generations, this relative old attack is becoming more attractive for attackers in the context of modern telecommunication networks like 5G. The effectiveness of the attack significantly depends on the manner how the target phone is used. Modern devices with many installed apps and a frequent device usage through phone calls, video streaming, messaging etc. render IMSI probing attacks more complex with a relative low success rate. Nevertheless, 3GPP has proposed mitigation mechanisms (3GPP TS33.501, 6.12.3) which are ineffective though [6]. The attack leaves traces on the device, i.e., in the baseband processor, and hence

can be detected. The results of our empirical study have shown that the IMSI probing attack based on the algorithm of Kune et al. using paging messages is rather ineffective and only successful under optimal conditions. We have investigated the success rate for phones in passive and active mode. Especially the passive mode contradicts the current trends of phone and data service usage. These use cases are pretty unlikely. We have shown, however, that the use of additional probes and more robust detection algorithms can improve the success rate significantly. Nevertheless the execution of the attack remains complex. We have proposed a novel attack vector based on the monitoring of the RRCConnectionSetup messages of the RRC protocol. It significantly reduces the number of probes thus increasing the attack success likelihood. This makes the deployment of the attack less complicated. Using the RRCConnectionSetup message instead of paging messages doubles the attack success rate and also improves the localization accuracy to cell granularity level. The impact of network and phone delays, e.g., from network utilization, on the IMSI probing time window needs to be investigated further. In particular, the trade-off between the number of messages (i.e., paging or RRCConnectionSetup) per time interval and the necessary window size for capturing in the context of messages captured after a single probe are of special interest here. Moreover, only smartphones were considered. Other device types, such as IoT devices or vehicles, may have a different susceptibility to IMSI probing attacks. In this domain, the influence of the Discontinuous Reception (DRX) or Extended Discontinuous Reception (eDRX) cycles may play a significant role for the attack success rate, as paging occasions might be reduced drastically.

## A Installed Applications in Section 4

Facebook, Whatapp, Facebook Messenger, Instagram, TikTok, Subway Surfers, Facebook Lite, Microsoft Word, Microsoft PowerPoint, Snapchat, SHAREit, Netflix, Twitter, Flipboard, Candy Crush Saga, Skype, Spotify, Dropbox, Viber, LINE.

## References

1. Amarisoft: Amari callbox (2022). https://www.amarisoft.com
2. Arapinis, M., et al.: New Privacy Issues in Mobile Telephony: Fix and Verification. CCS, North Carolina, USA (2012)
3. Bui, N., Widmer, J.: Owl: a reliable online watcher for LTE control channel measurements. In: Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges, pp. 25–30. ATC 2016, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2980055.2980057
4. Falkenberg, R., Ide, C., Wietfeld, C.: Client-based control channel analysis for connectivity estimation in LTE networks. In: 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), pp. 1–6 (2016). https://doi.org/10.1109/VTCFall.2016.7880932

5. Falkenberg, R., Wietfeld, C.: FALCON: an accurate real-time monitor for client-based mobile network data analytics. In: 2019 IEEE Global Communications Conference (GLOBECOM). IEEE, Waikoloa, Hawaii, USA (2019). https://doi.org/10.1109/GLOBECOM38437.2019.9014096, https://arxiv.org/abs/1907.10110
6. Hong, B., Bae, S., Kim, Y.: GUTI reallocation demystified: cellular location tracking with changing temporary identifier. In: NDSS (2018)
7. Kumar, S., Hamed, E., Katabi, D., Erran Li, L.: LTE radio analytics made easy and accessible. SIGCOMM Comput. Commun. Rev. **44**(4), 211–222 (2014). https://doi.org/10.1145/2740070.2626320
8. Kune, D.F., Koelndorfer, J., Hopper, N., Kim, Y.: Location leaks on the GSM air interface. ISOC NDSS (2012)
9. Mjølsnes, S.F., Olimid, R.F.: Easy 4G/LTE IMSI catchers for non-programmers. In: Rak, J., Bay, J., Kotenko, I., Popyack, L., Skormin, V., Szczypiorski, K. (eds.) MMM-ACNS 2017. LNCS, vol. 10446, pp. 235–246. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65127-9_19
10. Nie, S., Zhang, Y., Wan, T., Duan, H., Li, S.: Measuring the deployment of 5g security enhancement. In: Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks, pp. 169–174. WiSec 2022, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3507657.3528559
11. OpenAirInterface software alliance: openairinterface (2022). https://openairinterface.org
12. P1sec: QCSuper (2022). https://github.com/P1sec/QCSuper
13. Shaik, A., Borgaonkar, R., Asokan, N., Niemi, V., Seifert, J.P.: Practical attacks against privacy and availability in 4g/LTE mobile communication systems. NDSS16 (2016)
14. Software Radio Systems: srsRAN (2022). https://github.com/srsran/srsRAN
15. Zalani, R.: Screen time statistics 2021: your smartphone is hurting you (2021). https://elitecontentmarketer.com/screen-time-statistics/