Detecting IMSI-Catchers by Characterizing Identity Exposing Messages in Cellular Traffic

Tyler Tucker,* Nathaniel Bennett,* Martin Kotuliak,† Simon Erni,† Srdjan Capkun,† Kevin Butler,* and Patrick Traynor*
*University of Florida, {tylertucker1, bennett.n, butler, traynor}@ufl.edu
†ETH Zurich, {martin.kotuliak, simon.erni, srdjan.capkun}@inf.ethz.ch

***Abstract*—IMSI-Catchers allow parties other than cellular network providers to covertly track mobile device users. While the research community has developed many tools to combat this problem, current solutions focus on correlated behavior and are therefore subject to substantial false classifications. In this paper, we present a standards-driven methodology that focuses on the messages an IMSI-Catcher *must* use to cause mobile devices to provide their permanent identifiers. That is, our approach focuses on causal attributes rather than correlated ones. We systematically analyze message flows that would lead to IMSI exposure (most of which have not been previously considered in the research community), and identify 53 messages an IMSI-Catcher can use for its attack. We then perform a measurement study on two continents to characterize the ratio in which connections use these messages in normal operations. We use these benchmarks to compare against open-source IMSI-Catcher implementations and then observe anomalous behavior at a large-scale event with significant media attention. Our analysis strongly implies the presence of an IMSI-Catcher at said public event ($p << 0.005$), thus representing the first publication to provide evidence of the statistical significance of its findings.**

I. Introduction

Cellular networks attempt to protect their users from unauthorized tracking through the use of temporary identifiers. Unfortunately, a combination of limited authentication, downgrade attacks, and pre-authentication messages enable adversaries to force mobile phones to transmit their permanent identifiers, also known as their International Mobile Subscriber Identity (IMSI). So-called IMSI-Catchers remain an elusive threat to nearly all widely deployed cellular networks [1]. These devices continue to operate regardless of security improvements offered by each successive generation and are now attainable by nation-states as official products [2] and citizens as publicly available open-source projects [3], [4], [5].

The problem of IMSI-Catchers has been well-known for over two decades [2], [6], [7], [8], [9]. Unsurprisingly, the research community has developed many tools to attempt to detect such devices. For instance, techniques including detecting weak cipher use [10], [11], [12], anomalous broadcast messages [13], [14], [15], [16], [17], [18], [19], [20], the presence of ephemeral base stations [21], [12], [13], [14], [15], [16], [17], [18], [19], [20], and many more [22], [23], [24], [25], [26], [27], [28] have all been proposed as effective means of detecting such devices. The problem with these solutions is that they focus on characteristics that are not *causal*; rather, they instead look for behaviors that are *correlated* with changes that occur in both malicious and benign scenarios. For instance, a cellular provider may set up a temporary base station[1] at a large public event or may change the configuration of their towers, both of which may lead to false positive classifications. As such, no published academic detector has been able to present strong evidence (i.e., statistical significance) of an IMSI-Catcher performing mass surveillance in the wild.

In this paper, we focus on the messages an adversary must send as opposed to behaviors their IMSI-Catcher might exhibit. We develop a detection methodology that searches cellular downlink traffic for all possible messages an adversary can use to force a mobile phone to reveal its IMSI. While some app-based solutions look for the presence of one particular message (i.e., Identity Requests), no solutions consider the much wider set of IMSI-exposing messages or their prevalence across all downlink channels. In so doing, we make the following contributions:

- **Comprehensive Analysis of IMSI-Exposing Messages:** Prior work has used either heuristics or the presence of Identity Request messages to potentially identify IMSI-Catchers; however, we determine that many more messages that expose IMSIs exist. We perform a detailed analysis of 3GPP standards documents to create the first comprehensive list of 53 pre-authentication messages for 2G, 3G, 4G, and 5G-NSA networks that can be used to elicit a mobile device to reveal its IMSI.
- **Multi-Continent Network Measurement:** We perform over 400 hours of network measurement to benchmark the regularity with which downlink connections contain IMSI-exposing messages. This study allows us to create baseline profiles that include measurements across multiple provider networks, across population densities, and both in the presence and absence of large public events. Ultimately, we show that connections forcing IMSI exposures represent the

[1]Known as a Cellular on Wheels, or CoW [29].

Network and Distributed System Security (NDSS) Symposium 2025
24-28 February 2025, San Diego, CA, USA
ISBN 979-8-9894372-8-3
https://dx.doi.org/10.14722/ndss.2025.241115
www.ndss-symposium.org

minority of total connections (e.g., a median of less than 3% across total LTE connections) between base stations and User Equipment (UE).

- **Detection of IMSI-Catchers:** We use our baseline profiles to detect the presence of IMSI-Catchers in two scenarios: open-source IMSI-Catchers we deploy ethically in the lab, and suspected IMSI-Catchers present at an event with significant media and public attention in which a prominent public figure was required to appear in court. In the latter case, we demonstrate that the difference in ratios of IMSI-exposing connections against the benchmarks are statistically significant ($p << 0.005$), suggesting IMSI-Catcher presence. We then perform an experiment to demonstrate detection of large-scale overshadow attacks [30] using our methodology. No prior work has been able to substantiate their observations with statistical significance.

In spite of the wealth of papers in this space, IMSI-Catchers remain a persistent mass surveillance threat to cellular users. As such, calls for more effective detection mechanisms have recently come from both academia [1] and government [31]. Characterizing the effectiveness of detection mechanisms remains challenging–legal and contractual roadblocks to obtaining commercial IMSI-Catchers have impeded any hands-on analysis or operation of such devices by the research community. By focusing on messages that an IMSI-Catcher *must* use and their prevalence in downlink traffic, our approach is both effective and resilient against an adaptive adversary, and we demonstrate its efficacy in the wild.

The remainder of this paper is organized as follows: Section II provides necessary background information on cellular networks; Section III motivates our work in the IMSI-Catcher detection field; Section IV outlines all ethical considerations we take when conducting our experiments; Section V briefly discusses our threat model; Section VI describes our system implementation; Section VII shares the results of our experiments; Section VIII discusses these results and the broader impact of the work; Section IX recalls previous work in the space; Section X concludes our work.

II. Background

Cellular networks have experienced numerous improvements in security through the release of successive generations. One enduring problem despite these improvements, however, is the "catch-and-release"-style fake base station known as an *IMSI-Catcher* [15]. These devices take advantage of unauthenticated messages to extract a permanent identifier from a UE[2] (e.g., smartphones), representing a major threat to cellular subscriber anonymity from the perspective of a third party. Although 5G-SA networks introduce security mechanisms to conceal IMSIs from an attacker, the existence of pre-authentication downgrade attacks to LTE [35] and replay attacks [36] make IMSI-Catchers an enduring threat in multi-generation cellular networks.

The term "IMSI-Catcher" is overloaded in the broader security community. In this work, IMSI-Catcher refers to standalone base stations that broadcast themselves to local UEs with the explicit purpose of exposing the IMSI of each victim UE. Other forms include the passive IMSI-Catcher, a downlink channel sniffer that recovers any IMSI that is sent by a legitimate base station [37], and Man-in-the-Middle (MitM) fake base stations that can intercept all data between a UE and a base station [38] and/or send SMS spam to local cellular subscribers [39], [40].

[2]We refer to cellular network components by their respective names in the Long Term Evolution (LTE) standards [32], [33], [34] throughout this work.

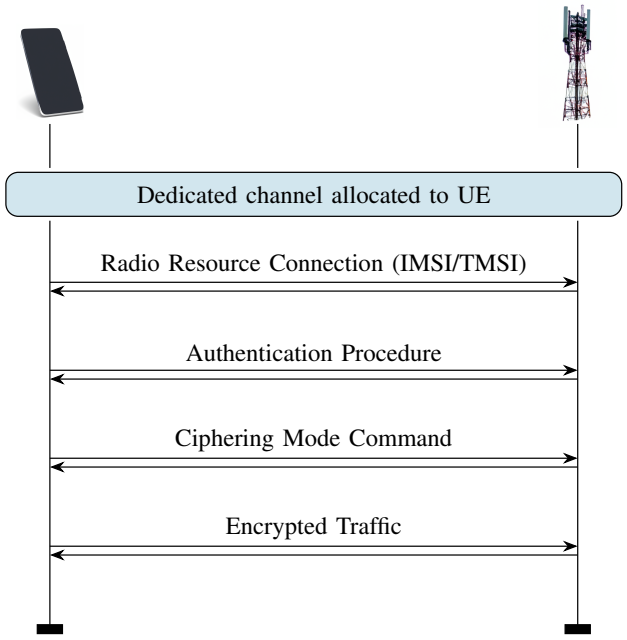


Figure 1: The attachment process between a UE and a base station, agnostic to any cellular generation. The UE will request a radio resource connection with the base station, and then begin the authentication procedure with the network through the base station. Finally, the two parties will agree on a cipher and generate encrypted traffic.

A. Cellular Authentication

A UE will initially identify itself to a base station using its IMSI. To support device anonymity over the air, the network will allocate a temporary version of the IMSI called the Temporary Mobile Subscriber Identity (TMSI). This value persists for a time (e.g., eight hours [41]) determined by the network operator, after which the network replaces the UE's old TMSI with a new one.

A UE using a TMSI attaches to the network using the process outlined in Figure 1. The base station first allocates bandwidth for the UE using a dedicated channel, and then the UE requests a radio resource connection with the base station. After the base station responds to this request, the UE can begin the authentication procedure with core components of the network. Once authentication is complete, the UE and base station decide on a cipher before finally exchanging encrypted traffic.

If a base station does not recognize the TMSI sent by the UE, it may ask the UE to transmit its IMSI. This request can be

performed while the two parties are unauthenticated, meaning that even a base station isolated from core network components can instruct a UE to reveal its IMSI. This detail enables an IMSI-Catcher attack and has been exploited across multiple cellular generations [42], [43], [44], [45], [46], [3], [47], [48], [49].

B. IMSI-Catchers

An IMSI-Catcher, visualized in Figure 2, follows a two-step process: (1) encourage a local UE to attach to it, and (2) force the UE to transmit its IMSI. To attract a local UE, an IMSI-Catcher may be positioned nearer to crowds than the base station, transmit at a higher power than nearby base stations, or even select frequencies based on broadcast frequency priority lists. A UE generally prioritizes stronger broadcasting base stations and frequency channels listed in priority lists, as it assumes they will provide the most reliable connection to the network. Furthermore, an IMSI-Catcher can adopt a more aggressive strategy by jamming popular cellular frequencies to force multiple UE to immediately seek a new connection.

Once an IMSI-Catcher begins receiving UE associations, it can force each UE that connects to it to reveal its IMSI. Following this, the IMSI-Catcher no longer needs to communicate with the victim UEs, allowing them to ignore the remainder of the authentication process. An IMSI-Catcher may even record the TMSI associated with each IMSI it captures to continue tracking UEs in a passive manner [41], [30].

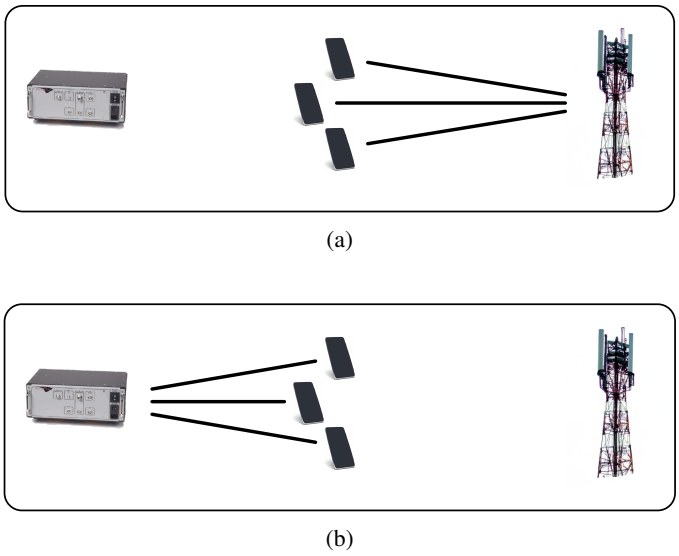
For years, the public believed IMSI-Catchers were only available to law enforcement [2], so the threat of such devices operated by civilians was largely dismissed [50]. With the development of open-source base station software [51], [5] and inexpensive Software-Defined Radios (SDRs) [52], [53], however, IMSI-Catchers targeting modern cellular generations can now be deployed for as little as $1,000 [3], [4].

III. Motivation

Numerous publications in cellular security focus on IMSI-Catcher detection [15], [22], [23], [11], [24], [25], [26], [27], [20], [12], [28], [54], [55], [56], [16], with theoretical and practical studies spanning over a decade. Despite this effort, the solutions offered by the community still lack reliability and too often rely on heuristics based on oversimplified assumptions of how a normal base station should appear. Recent work supports this observation and calls for further effort to tackle the problem [1].

A. Limitations of Current Detection Techniques

Multiple proposals for IMSI-Catcher detection focus on physical base station attributes such as transmitter location and radio fingerprinting [15], [20], [28]. Others focus detection techniques on base station configuration collected by monitoring broadcast channels, supported cipher lists, and cellular generations in use [57], [58]. These proposals rely on the assumption that an IMSI-Catcher will not adequately mimic surrounding base stations and therefore exhibit abnormal behavior or configuration during operation; from this, we draw our first observation on the limitation of current detection methods:



(a)

(b)

Figure 2: Visualization of an IMSI-Catcher attack. To begin, (a) a group of UEs will be connected to a commercial base station. A hidden IMSI-Catcher will begin transmitting near a target group of UEs. Victim UEs will disconnect from a valid base station and (b) connect to the now favorable IMSI-Catcher. The IMSI-Catcher instructs UEs to transmit their IMSIs after attaching.

Observation 1. Current detection mechanisms rely on behaviors *possibly correlated* with an IMSI-Catcher, rather than network effects that an IMSI-Catcher *must cause* to operate successfully.

While prior IMSI-Catcher detection heuristics successfully flag the presence of new base stations, they cannot distinguish between a new commercial base station and a properly configured IMSI-Catcher. Cellular providers temporarily set up new base stations to improve coverage or provide additional bandwidth during large events [29], which can trigger false positives for these techniques. Conversely, misconfigurations in legitimate base stations lead to false positives, and the assumptions detection mechanisms make about expected base station operation (such as avoiding A5/0 "Null" cipher use) do not reliably hold across network operators or even across all regions of a given operator. Lastly, heuristics that rely on RF strength or similar physical-layer characteristics are additionally impacted by environmental and landscape factors, such as the occasional but normal propagation of distant base station broadcasts observed in SeaGlass [20]. We assess the impact of these various conditions on existing detection methods in Table I, and draw our second observation from this assessment:

Observation 2. Current detection mechanisms suffer from a high incidence of false positives when faced with non-standard RF conditions, temporary-use base stations, or misconfigured/weakly-configured base stations.

In addition to the incidence of false positives for such detection techniques, prior work assumes a naïve IMSI-Catcher

	IC-SC [10]	TMSB [11]	NB-ICD [21]	ICD-CN [12]	SITCH [13]	Apple Patent [14]	mICC [15]	sICC [15]	App-Based [16]	Crocodile Hunter [17]	GSMK Overwatch [18], [19]	SeaGlass [20]	Our Approach
Noisy Background RF	◐	◐	●	●	◐	◐	●	○	●	◐	●	●	●
Weak Cipher Use	○	○	●	●	●	●	●	●	◐	●	●	◐	●
Freq. Reassociation	◐	◐	●	●	○	●	●	●	○	●	◐	○	●
Freq. Downgrade	○	○	●	●	●	◐	●	●	◐	●	●	●	●
Anomalous Propagation	◐	●	○	○	○	○	○	○	○	○	○	○	●
Temporary Tower	●	◐	○	○	○	○	○	○	○	○	○	○	●
Misconfigured Tower	○	○	○	○	○	○	○	○	○	○	○	○	●

Table I: Efficacy of various detection methods when exposed to suboptimal cellular network conditions. ○ indicates the condition will directly lead to increased false positives in the detection mechanism, ◐ indicates the condition may lead to false positives in the detection mechanism under certain circumstances, and ● indicates the condition has no effect on the detection rate of the mechanism.

implementation that makes little attempt to conceal itself within its environment. However, an IMSI-Catcher is privy to the same network information that a detector is, and could readily be designed to adapt its configuration to mimic nearby base stations. For instance, patent applications for IMSI-Catchers describe mimicking location areas and similar parameters [59], [60]. Some detection mechanisms additionally assume an IMSI-Catcher will always exhibit certain behaviors, such as downgrading communications to 2G or short-lived operation. Furthermore, most detection techniques require a ground-truth measurement of benign base for stations and are unable to reliably detect IMSI-Catchers when first activated in a new area. Taken together, these factors result in false negatives for prior detection techniques. We categorize and compare these factors in Table II.

Beyond the concealment of an imposter base station, techniques exist to inject malicious payloads into traffic between a cellular device and a legitimate base station. These are referred to as *overshadow* attacks, and recent research has shown such an attack to be practical and scalable on both downlink and uplink cellular traffic [61], [62]. Such attacks can readily be used to carry out IMSI-catching: a downlink overshadow attack may inject an *Identity Request* appearing to be from the base station, while an uplink overshadow attack may insert an *Attach Request* containing an invalid TMSI, thereby leading the base station to query the handset for its IMSI. Rather than having to mimic the base station location, radio frequency, and configuration expected within a given area, an IMSI-Catcher using overshadow conceals itself within actual legitimate base station communications. This leads to our third observation:

Observation 3. Advances in IMSI-Catcher stealth (including the demonstration of practical overshadow attacks) defeat all current detection mechanisms.

	IC-SC [10]	TMSB [11]	NB-ICD [21]	ICD-CN [12]	SITCH [13]	Apple Patent [14]	mICC [15]	sICC [15]	App-Based [16]	Crocodile Hunter [17]	GSMK Overwatch [18], [19]	Seaglass [20]	Our Approach
No Jamming	●	●	●	●	●	●	●	○	●	●	●	●	●
Fixed Location	●	●	●	●	●	○	●	●	●	○	●	●	●
Long-Term Campaign	●	●	●	●	●	●	○	●	●	●	○	○	●
Benign Radio Freq.	●	◐	●	●	○	●	●	○	●	●	●	○	●
No Unusual Paging	●	●	●	●	●	●	○	○	●	●	○	○	●
Benign Ciphers	○	○	●	○	●	●	●	●	●	●	●	◐	●
Parroted Cell-ID	●	●	●	●	○	○	○	○	○	●	●	●	●
Plausible RF Power	●	●	○	●	○	○	●	●	○	○	●	○	●
Benign Broadcast Params	●	●	●	●	●	○	○	○	○	○	○	○	●
Parroted LAC	●	○	○	○	●	●	○	●	○	○	●	○	●
LTE-Only	○	○	○	○	○	◐	○	○	○	●	◐	○	●
New Location	○	○	○	○	○	●	○	○	◐	○	○	○	●
DL Overshadow	○	○	○	○	○	○	○	○	○	○	○	○	●
UL Overshadow	○	○	○	○	○	○	○	○	○	○	○	○	●

Table II: Efficacy of various detection methods against an IMSI-Catcher exhibiting the specified behavior. ○ indicates the behavior will lead to increased false negatives in the detection mechanism, ◐ indicates the behavior may lead to false negatives in the detection mechanism under certain conditions, and ● indicates the behavior has no effect on the detection rate of the mechanism.

Of the prior IMSI-Catcher detection works we explore, only two perform controlled experiments in real-world cellular networks: *SeaGlass* [20] and *IMSI-Catch Me if You Can* [15]. Each of these works carries out wide-scale detection campaigns spanning several months of cellular traffic capture. In both cases, researchers observe frequent abnormalities in individual heuristic measurements, but multiple factors are never observed together. During analysis, the authors identified these abnormalities as resulting from environmental factors, misconfigured towers, or operator reorganization of cellular infrastructure. To add to this, prior IMSI-Catcher detection work has never measured a large number of IMSIs being requested by a base station. As IMSI discovery is the fundamental goal of an IMSI-Catcher, this represents a key missing step in determining whether an anomaly stems from an actual attack or not. From this, we draw our final observation:

Observation 4. No published detection mechanism to date has observed an IMSI-Catcher actively harvesting a large number of IMSIs in the wild.

We note that several organizations and governmental agencies have reported conclusive evidence of IMSI-Catchers in various contexts [45], [63], therefore we do not claim to be the first to detect an IMSI-Catcher. Whereas other reports have kept details of detection methodology private, our work represents a solution that contributes to an open, scientifically grounded understanding of IMSI-Catchers.

Generation	Downlink Message	Reference
2G GSM	Identity Request	GSM TS 04.08 Sec. 4.3.3.1
	Authentication Reject	GSM TS 04.08 Sec. 4.3.2.5
	Abort, Cause #6	GSM TS 04.08 Sec. 4.3.5.2
	Location Updating Reject, #2-3, 6, 11-13	GSM TS 04.08 Sec. 4.4.4.7
	CM Service Reject, Cause #4 or 6	GSM TS 04.08 Sec. 4.5.1.1
3G UMTS	Identity Request	3GPP TS 124.008 Sec. 4.3.3
	Authentication Reject	3GPP TS 124.008 Sec. 4.1.1.2
	Abort, Cause #6	3GPP TS 124.008 Sec. 4.3.5.2
	Location Updating Reject, Cause #2-3, 6, 11-12	3GPP TS 124.008 Sec. 4.4.4.7
	CM Service Reject, Cause #4, 6	3GPP TS 124.008 Sec. 4.5.1.1
	Attach Reject, Cause #3, 6-8, 11-15	3GPP TS 124.008 Sec. 4.7.3.1.3
	Detach Request, Type "re-attach not required", Cause #2-3, 6-8, 11-15	3GPP TS 124.008 Sec. 4.7.4.2.2
	Routing Area Update Reject, Cause #3, 6-7, 9, 11-12, 14	3GPP TS 124.008 Sec. 4.7.5.1.4
	Authentication and Ciphering Reject	3GPP TS 124.008 Sec. 4.7.7.5
	Service Reject, Cause #3, 6-7, 9, 11-12	3GPP TS 124.008 Sec. 4.7.13.4
4G LTE	Identity Request	3GPP TS 124.301 Sec. 4.3.3
	Attach Reject, Cause #3, 6-8, 11-15, 35	3GPP TS 124.301 Sec. 5.5.1.2.5
	Detach Request, Type "re-attach not required", Cause #3, 6-8, 11-15	3GPP TS 124.301 Sec. 5.5.2.3.2
	Tracking Area Update Reject, Cause #3, 6-7, 9, 11-12, 14	3GPP TS 124.301 Sec. 5.5.3.2.5
	Service Reject, Cause #3, 6-7, 9, 11-12	3GPP TS 124.301 Sec. 5.6.1.5
5G NR (NSA)	Same as 4G LTE	3GPP TS 137.340 Sec. 7.1

Table III: The first comprehensive enumeration of IMSI-exposing messages in 3GPP standards. This includes downlink messages that either (a) explicitly request that the UE transmit its IMSI over the air (i.e., Identity Request), or (b) implicitly require the UE to delete its TMSI, leading to IMSI exposure in a subsequent connection. Certain messages require a specific 'Type' and/or 'Cause' value for the UE to be instructed to delete its TMSI.

B. Research Questions

Taking our four observations on prior research into account, we seek to improve on past work in IMSI-Catcher detection using a standards-driven approach. We outline multiple research questions to guide our approach:

RQ1 What are the messages that IMSI-Catcher designers can use to harvest IMSIs based on cellular standards?

RQ2 How often do commercial networks gather IMSIs in relation to total network connections?

RQ3 Can we detect an open-source IMSI-Catcher in an ethically controlled setting based on the rate of IMSI-exposing connections?

RQ4 Can we detect an IMSI-Catcher in the wild that we do not control?

We begin our study by analyzing the 3GPP cellular standards (i.e., 2G GSM [64], 3G UMTS [65], 4G LTE [33], [34], 5G NSA[3]). Due to the structure of 3GPP standards, this information is contained in only a small number of specific documents, significantly limiting our search space. To identify all IMSI-exposing messages, we manually look at all pre-authentication flows and their associated message types, as well as post-authentication messages that can be sent without integrity or ciphering enabled. We additionally review text surrounding certain keywords (e.g., IMSI, TMSI, GUTI) that are necessary to describe IMSI-exposing events. From these, we categorize IMSI-exposing messages as messages that either: (a) request that the UE provide its IMSI (e.g., *Identity Request*), or (b) direct the UE to delete its stored TMSI, thereby leading it to reveal its IMSI in subsequent communications.

While our approach is not formally complete, we believe it is a best-efforts approach to handling natural language specifications. We list all such IMSI-exposing messages in Table III. To our knowledge, no prior work has identified the full extent of IMSI-exposing messages (***RQ1***).

By observing the incidence of IMSI-exposing messages in connections, we take advantage of the opposing goals that commercial networks and IMSI-Catchers have with uplink IMSI transmissions. Our detection mechanism is therefore based on what IMSI-Catchers *must* transmit to function, a set of messages that commercial networks are designed to minimize. These messages should therefore be present in only a small amount of total dedicated connections. A sudden influx of connections containing these messages, based on this observation, would be highly indicative of an IMSI-Catcher attack. We formulate a null hypothesis based on this observation:

> ***Null Hypothesis*** *(H_0)* — The incidence of connections with IMSI-exposing messages from commercial base stations and IMSI-Catchers come from the same population.

Establishing a null hypothesis in this format allows us

[3]5G NSA uses control messages from the LTE standard before directing the user to the 5G interface.

to perform standardized statistical tests when evaluating the efficacy of our approach, a technique that has not previously been used in IMSI-Catcher detection. To complement H_0, we provide an alternative hypothesis:

> ***Alternative Hypothesis (H_a)*** **—** The incidence of connections with IMSI-exposing messages from commercial base stations and IMSI-Catchers come from different populations.

C. Novelty of Approach

Identity Requests have long been recognized in prior works as the message an IMSI-Catcher would use to obtain IMSIs from victims. As such, several IMSI-Catcher detectors flag the presence of Identity Requests as suspicious. To determine the extent to which existing detectors track the wider set of IMSI-exposing messages (identified in Table III), we systematically analyze existing detection techniques and note our findings in Table IV. From this, we draw two observations:

- **Only app-based approaches track IMSI-exposure:** Works that do track IMSI-exposing messages have always done so on an individual basis, usually by monitoring baseband logs on a single cellphone. As we will show in Section VII, the presence of a single IMSI-exposing message is not a reliable indicator of IMSI-catching. Our approach expands measurement to cover all cellphones attaching to a base station.
- **No approaches consider TMSI-deleting messages:** In nearly every case, app-based detection mechanisms assume the Identity Request is the sole means by which an IMSI-Catcher can harvest cellphone identities. While one exception–Snoopsnitch–additionally flags LAU Rejects, it only does so when accompanied by an Identity Request (and therefore does not consider it as an independant IMSI-exposing message).

Detection Technique	Monitored Identity Messages
App-Based (Single UE)	
Darshak [66]	Identity Request
Snoopsnitch [58]	Identity Request, LAU Reject[a]
CellSpyCatcher [67]	N/A
AIMSICD [57]	N/A
GSMSpyFinder [68]	N/A
SDR-Based	
Crocodile Hunter [17]	N/A
mICC/sICC [15]	N/A
Apple Patent [14]	N/A
IC-SC [10]	N/A
SITCH [13]	N/A
TMSB [11]	N/A
ICD-CN [12]	N/A
NB-ICD [21]	N/A
Seaglass [20]	Identity Request[b]
GSMK Overwatch [18]	N/A[c]

[a] only flags LAU rejects if accompanied with an Identity Request
[b] uses an app-based detector in conjunction with other heuristics
[c] based on published technical reports and prior researcher experiments

Table IV: IMSI-exposing messages that have been used by prior works for IMSI-Catcher detection. No other work to date has explored monitoring such messages at scale–only app-based approaches have tracked a restricted subset of IMSI-exposing messages (namely Identity Requests).

IV. Ethical Considerations

When designing our methodology, we followed best practices on minimizing unnecessary data collection [69], [70]. Accordingly, we only passively listen to downlink traffic sent on control channels while evaluating our research questions through measurements on commercial networks. This approach ensures that we *cannot* collect any personally identifying information (e.g., IMSIs) from users or attempt to analyze encrypted data. As such, we maintain the privacy and confidentiality of cellular network subscribers when operating our tool.

When operating open-source IMSI-Catchers, we isolate them using a Faraday cage so that no external device can detect or connect to them. By doing so, we ensure that our experiments do not interfere with phones attempting to connect to emergency 911 services or other important cellular operations. We additionally set the open-source base stations (some of which inherently lack mutual authentication) to only allow connections from our specific test Subscriber Identity Module (SIM) cards. This ensures that a foreign device cannot fully connect to our base station even if it detects it. We review all logs taken by our base station software to confirm that no IMSIs are collected from devices we do not own. When performing overshadow attacks on commercial base stations, we obtain explicit permission from cellular providers. Additionally, we ensure that the TMSI of each tracked connection is that of the test device *before* beginning each attack.

During our experiments, we also travel to a large event to scan for the presence of IMSI-Catchers similar to efforts in previous work [20], [15]. When attending the event, we maintain distance from dense crowds and ensure that our system in no way interferes with communications (i.e., we never transmit). Furthermore, we did not bring personal UEs to ensure protect their personally identifying information. Due to the passive nature of our system and the absence of personally identifying information in our logs, we were not required to seek an IRB for these efforts.

V. Security Model

We assume that an adversary can execute a subset of the 2G-5G NSA protocols at frequencies set within those protocol specifications. Additionally, the adversary can jam arbitrary frequencies of local base stations to aggressively force victim UEs to connect to their IMSI-Catcher. We do not assume that any IMSI-Catcher operated by the adversary is connected to the core network of any cellular provider, such that the adversary does not need to have access to network keys necessary to complete the Authentication and Key Agreement (AKA) protocol. We also assume that the adversary does not use malformed packets when transmitting. Finally, while 5G

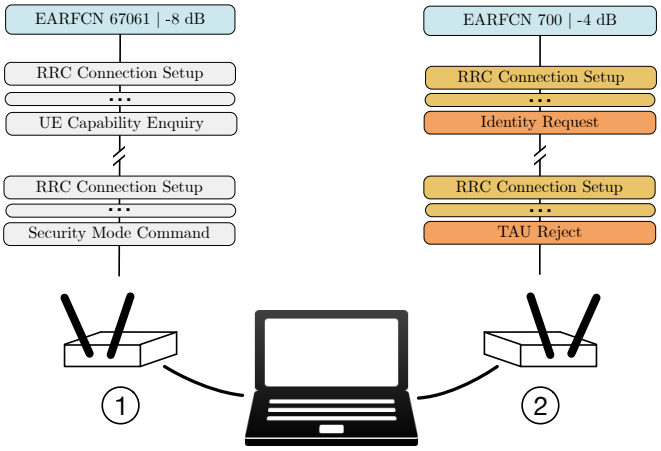


Figure 3: Overview of our system, showcasing two simultaneous captures using two SDRs set to different frequencies. The first capture (1) contains zero IMSI-exposing connections while the second capture (2) includes two such connections, each with an IMSI-exposing message highlighted in orange. This latter case produces a high IMSI exposure ratio, causing our system to identify an IMSI-Catcher operating on EARFCN 700.

SA networks are purported to solve these issues, recent news releases suggest that 5G-centric IMSI Catchers do exist [71]; however, their specific mechanisms are not yet discussed. As such, we must wait until further public disclosure before developing detection techniques.

We consider an adversary whose goal is to gather IMSIs from all mobile devices present in an area, as such attacks have been widely reported in the wild [56]. The adversary may employ advanced capabilities (such as a large-scale overshadow attack) to carry out the mass collection of IMSIs. We do not consider individual targeting of devices by an IMSI-Catcher, as such an attack would not even provide the adversary sufficient information to determine the presence or absence of a chosen subscriber IMSI in an area. We also assume that victim UEs are connected to commercial base stations before the attack begins, meaning that each UE will initially attempt to connect using its TMSI.

VI. Methodology

While previous work in IMSI-Catcher detection relies on base station misconfiguration or the sudden appearance of a new transmitter, we aim to detect what an IMSI-Catcher must do to achieve its goal: send IMSI-exposing messages to local UEs. To evaluate this approach, we implement a mobile system running consumer-grade hardware and open-source software.

A. Capturing Traffic

Before we can filter for IMSI-exposing messages, we first need to recover network captures of traffic originating from commercial base stations. Note that our approach requires that we obtain dedicated traffic sent between a base station and several UEs, meaning that we cannot use a rooted UE as a sniffer. For our implementation, we instead employ SDRs running open-source cellular sniffing software. The research community provides sniffer software for 2G networks [72] and LTE/5G-NSA networks [30], [73], [74], [75], [76]. Of these, we choose *gr-gsm* [72] running on a bladeRF x40 [77] to capture GSM downlink traffic and to capture LTE downlink traffic we use *LTESniffer* [73] or *DLProbe* [30] running on an Ettus B210 [53]. *LTESniffer* became available during this project and we have changed over from *DLProbe* to *LTESniffer* for its reliability and ease of use. In Appendix B, we show that for our use case, these two tools can be used interchangeably. To our knowledge, no code supporting a 3G sniffer is available. Additionally, 3G networks are being phased out [78] at the time of writing, limiting the availability of these networks for benchmarking. We therefore do not benchmark commercial 3G networks during our experiments.

We choose to limit our analysis to downlink channels for three reasons: (1) scalably monitoring uplink traffic requires additional hardware and is often prone to loss in packet recovery [73], [30] and the hidden terminal problem, (2) IMSIs are only sent by a UE on uplink traffic, so monitoring only downlink traffic inherently preserves the privacy of subscribers, and (3) downlink traffic reveals the *intent* to gather IMSIs and is therefore sufficient for our analysis.

Both of these sniffers output cellular messages as Wireshark packet captures (**.pcap*). In this format, we can use display filters to analyze captures for every IMSI-exposing message that we identify in Table III. To recover individual connections from a general packet capture, we scan for messages indicating the beginning of a connection with an associated identifier (i.e., searching for RAR messages which include a C-RNTI temporary identifier in LTE). This process allows us to timestamp the beginning of a connection and associate later packets with those connections.

B. Discovering Base Stations

Cellular network frequencies constitute a significant amount of wireless spectrum in developed countries [79]. These frequencies are referenced by a number representing a pair of uplink and downlink frequencies, such as EARFCNs in LTE (e.g., EARFCN 700 represents a downlink frequency of 1940 MHz and an uplink frequency of 1860 MHz). Base stations therefore have the option of operating on one of hundreds of frequencies within the spectrum allocated to their providers. However, baseband processors implementing these cellular standards often favor certain frequencies depending on their cellular provider to reduce the need to search all possible options. Consequently, an IMSI-Catcher is heavily incentivized to operate on one of these popular frequencies to attract UEs quickly. Our approach benefits from this, as it can narrow our search space significantly and reduce the amount of hardware necessary by identifying these frequencies.

We determine these popular frequencies by querying the public database cellmapper [80] for all base station information in the top ten most populous cities in one of our countries. Then, we sort our frequency list by the number of base stations operating on each frequency. To augment this list with real-time data, we log baseband messages from a UE that connects to local base stations. These messages include not only the currently connected base station but also neighbor

frequencies of nearby base stations. Results from these two sources comprise our final input list of frequencies.

C. Detection Algorithm

With the ability to capture cellular traffic and generate a list of frequencies to analyze, we implement an algorithm that maximizes our chance of detecting sudden upticks in IMSI-exposing connections. Our algorithm first chooses a frequency by looping through a list of potential options, prioritizing cells based on received signal strength. Then, the system captures traffic from that frequency for one minute before moving to the next to ensure that we do not ignore any frequency for a significant amount of time; Figure 3 shares this process.

After the allotted time, we turn off the radios and save the packet capture(s). We then scan each packet capture for both total connections and IMSI-exposing connections (i.e., connections containing at least one IMSI-exposing message). If the sniffer fails to connect to a base station on a given frequency after ten seconds, the algorithm will move on to the next frequency in the list. Finally, we formulate our IMSI Exposure metric as follows:

$$\text{IMSI Exposure Ratio} = \frac{\text{Total IMSI Exposing Connections}}{\text{Total Connections}}$$

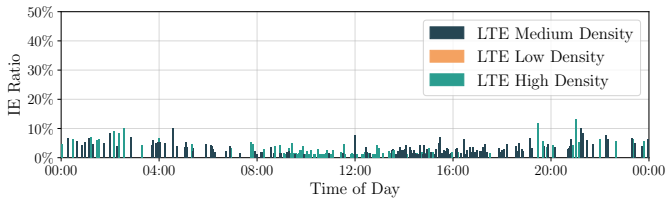
Our system can monitor frequencies simultaneously by incorporating multiple SDRs, providing scalability to this implementation. We provide an overhead analysis of our system in Appendix C.

VII. Results

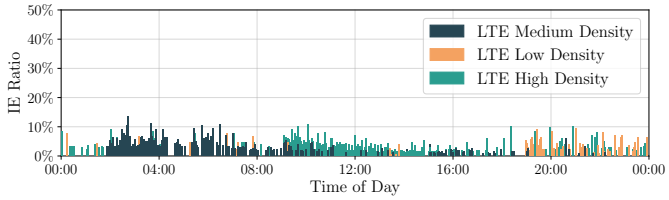
To evaluate our remaining research questions, we select several settings and locations for base stations in our experiments. First, we investigate the IMSI Exposure Ratio of commercial base stations in normal conditions, periods of high population density, and periods of low population density. These experiments provide us with a benchmark range of values from which we can compare to IMSI-Catcher traffic in our lab. Then, we set up open-source IMSI-Catcher implementations to provide data on attainable products. Finally, we bring our system to a large event to search for IMSI-Catchers we do not control.

A. Benchmarking Cellular Traffic

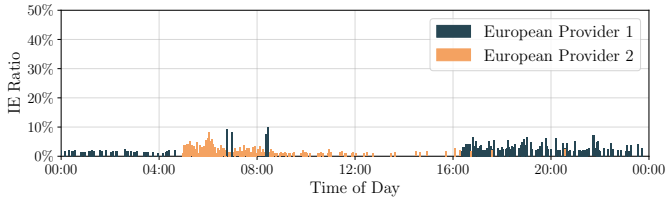
We investigate ***RQ2*** by collecting long captures of commercial base stations in several different settings and locations. To begin, we choose two network providers *Provider 1* and *Provider 2* which represent two large cellular providers in the United States. We select common frequencies in our local area based on cellmapper [80] data. Our cellular sniffers run on each frequency for 24 hours. We collect 192 hours (4 days) of 4G LTE traffic from each provider. We also collect 24 hours of data from the only active 2G GSM tower in our local area, which belongs to Provider 2. After collecting all of this data, we separate the results into bins of one-minute intervals and calculate the incidence of IMSI-exposing connections for each time bin. The results of this analysis are shown throughout Figure 4, labeled *LTE Medium Density* and *GSM Medium Density*. We observe that both networks exhibit consistent



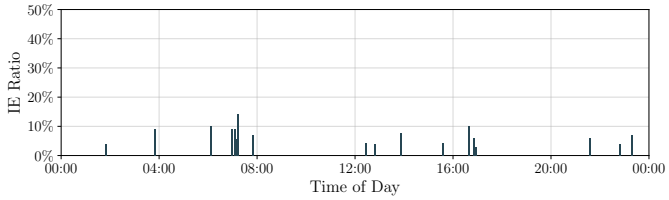
(a) Provider 1 LTE Benchmark Results (median=2.08%).



(b) Provider 2 LTE Benchmark Results (median=2.78%).



(c) European LTE Benchmark Results (median=2.00%).



(d) Provider 2 GSM Benchmark Results (median=5.88%).

Figure 4: Results of our various benchmark tests including normal conditions, periods of high population density, and periods of low population density. Throughout our experiments, the median IMSI-exposing ratio is less than 3% for LTE networks and less than 6% for GSM networks.

results in IMSI exposure ratio during normal conditions when implementing either cellular generation.

To further explore this area, we collect additional LTE captures during periods of high and low population density. For the former case, we target two sporting events: the first, a college football rivalry game with over 90,000 reported attendees, and the second, a college basketball game with over 10,000 attendees. Our system runs for 24 hours to capture network behavior before, during, and after these events. Despite increased traffic on local cell towers in these two cases, we see in Figure 4 that neither commercial network needs to transmit a high density of IMSI-exposing connections to handle dense crowds. For the latter case, we set up our equipment in a rural area over 400 miles from our lab for 24 hours. Our results for this experiment, also shown in Figure 4, contain exclusively low IMSI exposure ratios similar to the

results of our high-density tests. We observe that even with a modest number of connections, commercial base stations still maintain a low IMSI exposure ratio. These results are consistent with our expectations given that TMSIs are known to last many hours.

We additionally take 48-hour captures of base stations operated by two different providers in Europe to gather data on a separate continent (shown in Figure 4). Similar to our captures taken in the United States, we see that IMSI-exposure ratios remain consistently low throughout the day. After capturing over 400 hours of commercial network traffic in varying conditions and locations, we observe that all LTE IMSI-exposure ratios are below 3% (***RQ2***).

As a final test, we investigate cellular traffic on towers located near airport runways during scheduled landings. We choose these settings as they represent situations in which large groups of UEs will suddenly reconnect to the cellular network after a long period of disconnection (i.e., airplane mode). We take our measurements from within an airport building at an international airport and in a parking lot for a regional airport, both within the immediate vicinity of runways; each test runs for at least one hour. Our results for both cases remain correlated with those of our steady state captures and our high/low population density captures.

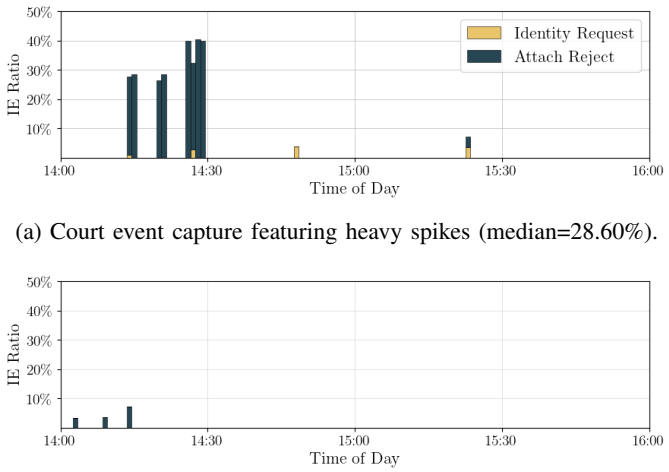
> ***Takeaway*** **—** Commercial base stations maintain low IMSI Exposure Ratios during normal conditions and periods of high and low population density.

B. *Open-Source IMSI-Catcher Experiments*

After evaluating commercial networks, we analyze the behavior of open-source IMSI-Catchers in a controlled lab setting. As researchers, we cannot legally obtain official IMSI-Catcher products [46] and therefore must source our devices from open-source guides [81], [44], [82] and consumer-grade hardware. Additionally, we operate these implementations within a Faraday cage to prevent loss of service for nearby people, as stated in our ethics section. This approach serves as the community standard in IMSI-Catcher academic research [83], [84], [85], [86], [87], [30], [88], [89], [90], [91], [92], [93].

Setup **—** We measure IMSI-Catchers created using open-source base station software (e.g., YateBTS [51] for 2G GSM, OpenBTS-UMTS [94] for 3G UMTS, srsRAN [5] for 4G LTE) in an isolated environment (i.e., Faraday cage). These base stations do not have a direct connection to a cellular provider's core network, meaning that they cannot associate a given TMSI with an identity and thus must ask the UE to provide its IMSI. When operating these IMSI-Catchers, we choose a frequency matching that of a nearby base station to ensure our UEs will monitor for base stations on that frequency. We do not employ more aggressive techniques (e.g., jamming) to force UEs to connect to these IMSI-Catchers as our methodology does not rely on how IMSI-Catchers implement this step of the attack.

For target UEs, we use smartphones containing a valid SIM card as well as simulated UEs running srsUE [5] with pre-programmed TMSIs. We test both Android and iPhone smartphones containing a variety of basebands. Both the smartphones and simulated UEs represent devices that fulfill the assumptions in our security model (i.e., devices that have a current TMSI). We use both methods simultaneously to scale up the number of UEs attempting to connect to our IMSI-Catcher in a short time window. Similar to our approach with base stations, we do not edit any source code with our UEs to ensure that they operate normally.



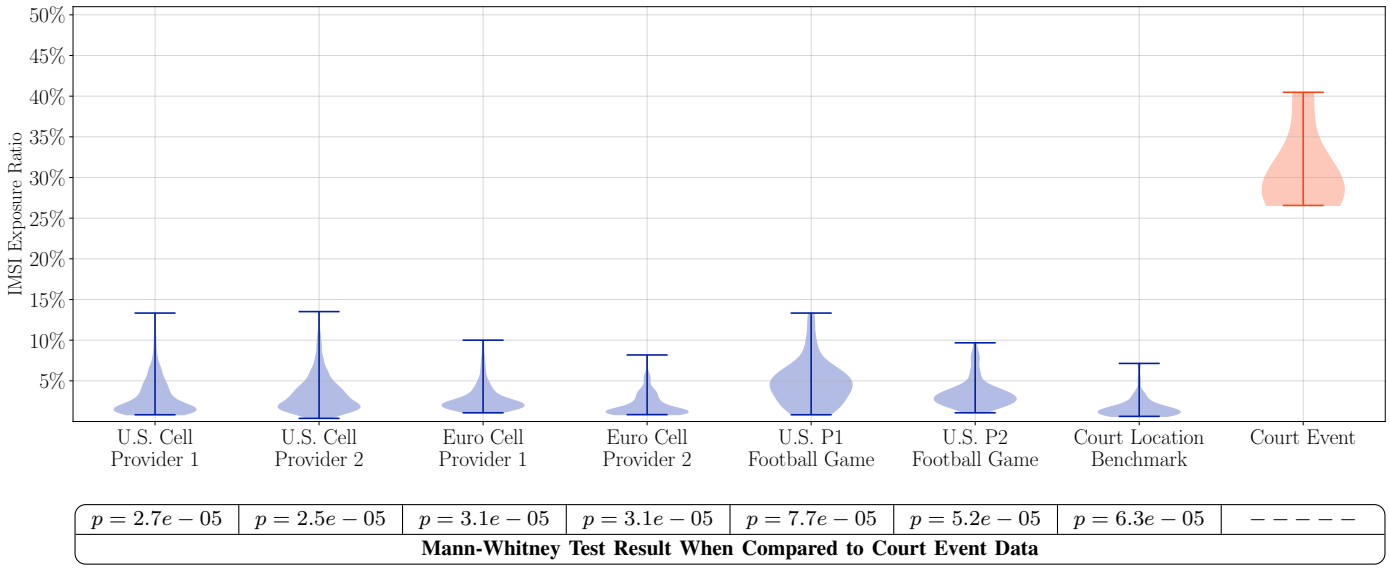
(a) Court event capture featuring heavy spikes (median=28.60%).

(b) Benchmark capture at identical location (median=1.30%).

Figure 5: Results of our cellular captures on LTE base stations (a) during an event compared to (b) the same location during normal conditions. We observe that our capture during the event contains several spikes in IMSI Exposure Ratio similar to the behavior of open-source IMSI-Catchers. Conversely, our later benchmark captures feature relatively small IMSI-Exposing Ratios.

Experiment **—** We place all components necessary for our attack in a Faraday cage along with the hardware running our detector, then seal the enclosure to prevent transmissions from entering or leaving the controlled environment. Because of the naïve design of open-source IMSI-Catchers, IMSI-exposure ratios were all measured at 100% across tests. However, we observe that each implementation of an IMSI-Catcher relies on different IMSI exposing messages to operate. The 2G implementation relies completely on *Location Updating Reject* messages while the 3G IMSI-Catcher uses a combination of *Identity Requests* and *RAU Rejects*; the 4G IMSI-Catcher uses exclusively *Identity Request* messages. These experiments highlight the need to monitor multiple different types of messages, as even naïve open-source code exhibits varying approaches to IMSI harvesting. Throughout these experiments, we review logs taken by each respective IMSI-Catcher to ensure that our target devices transmit their IMSIs in response to these messages.

> ***Takeaway*** **—** IMSI-Catchers constructed using open-source projects produce high IMSI Exposure Ratios (100%) during operation.



$p = 2.7e-05$	$p = 2.5e-05$	$p = 3.1e-05$	$p = 3.1e-05$	$p = 7.7e-05$	$p = 5.2e-05$	$p = 6.3e-05$	$-----$
Mann-Whitney Test Result When Compared to Court Event Data							

Figure 6: Violin plots for the distributions of LTE data from seven benchmark captures and our captures during the court event. We observe a steady distribution of data throughout the benchmark captures with a significant change during our event capture.

C. Field Experiments

***Event Selection* —** Having a positive result from our lab experiments, we mobilize our hardware and attend an event that we suspect would attract IMSI-Catcher presence. We attended an event at a major city away from our lab during which a prominent public figure appeared in court, drawing in a significant crowd of people and media attention. Public statements from law enforcement for this event state they took significant precautions when preparing for these crowds as well, without citing specific measures. We choose to keep the specific event anonymous as a matter of public safety.

***Results* —** During the event, we run our detector for two hours while seated at a coffee shop across the street from the courthouse. Our code follows the methodology outlined in Section VI. The top plot of Figure 5 shares the results of this process. We observe several spikes in IMSI-exposing connections, bringing the IMSI-exposing ratio to around 40%. Though not as high as the results of our naïve implementation experiments, they remain exceptional numbers when compared to those of our steady-state experiments.

These spikes additionally feature a significantly different makeup of messages when compared to our open-source IMSI-Catchers. In this case, the device we monitor heavily favors the use of *Attach Rejects*, only seldomly using *Identity Requests*. While *Identity Requests* directly ask a UE to respond with its IMSI, an *Attach Reject* instructs a UE to attempt reconnection using its IMSI. Observing yet another distribution of messages again highlights the importance of considering all possible message options that IMSI-Catcher operators can use. Previous work only considers the former messages, and would therefore miss the overall large ratio of IMSI exposures (i.e., *Identity Requests* contribute under 10% of the overall IMSI Exposure Ratio in all cases, meaning that they alone would not produce anomalous results compared to our benchmark captures). Based on news articles collected following the event, we confirm that the time of arrival of the public figure correlates with the timing of our detected spikes.

Post-Event Benchmarking* —** To discover if the results we collect during the event are anomalous for the area, we return to the same coffee shop sometime later and perform a two-hour capture while no major event occurs nearby. We ensure that one of our radios is set to the same frequency that we detected suspicious traffic on during the event. The bottom plot of Figure 5 compares the results of this test to the data from the event. We see that our results correlate well with those of our extended benchmarking experiments in Section VII-A, namely that all time bins in the capture contain a low density of IMSI-exposing messages (i.e., all samples are under 10%). We believe that these results strongly suggest that the event we chose had IMSI-Catcher presence (RQ4***).

D. Statistical Analysis

We use a violin plot to visualize the distributions of our datasets in Figure 6. Results from our observations on commercial LTE networks all have a median of less than 3%, while results from the suspected IMSI-Catcher event comprise a much higher median of 29%. Observing this distinct distribution of our event data, we characterize the difference between populations of samples taken on commercial networks and IMSI-Catchers by performing statistical tests on our data. We adopt new, stricter requirements for p-values [95] (i.e., $p << 0.005$ required to reject the null hypothesis).

First, we test if each dataset is normally distributed using the Shapiro-Wilk test [96] (results in Appendix A). The results of these tests suggest that not all datasets fit the requirements for a normal distribution. As such, we compare benchmark data and IMSI-Catcher data using the Mann-Whitney U Test [97], a nonparametric alternative to the t-test.

We perform seven Mann-Whitney U tests, each comparing a set of benchmark captures with the results of our court event capture. The results of all of these tests, shown in Figure 6,

produce $p << 0.005$, suggesting that results from commercial captures and the court event are drawn from different populations. This result, combined with the results of our open-source IMSI-Catcher experiments, allows us to reject our null hypothesis (H_0 ***Rejected***). These results therefore support our alternative hypothesis H_a that states IMSI exposing traffic generated by base stations and IMSI-Catchers come from different populations.

> ***Takeaway —*** Our captures from commercial networks and the Court Event are distinct and statistically significant.

E. Overshadow Attacks

Based on recent advances in packet injection over cellular downlink channels known as *overshadow attacks* [61], [62], the base station itself can be optimized out of the adversary's requirements. This solution offers adversaries an alternative to an IMSI-Catcher by allowing them to instead inject an IMSI-exposing message (such as an Identity Request) on a frequency of a commercial base station during a valid time slot [30]. This approach renders all prior IMSI-Catcher detection methods useless as the adversary can perform an attack without the existence of a separate base station. However, our approach can detect this form of attack because it still produces downlink traffic that contains an elevated incidence of IMSI-exposing connections.

To test this, we contacted the authors of an overshadow attack [30] and monitored traffic during their experiments on a commercial-grade base station [98]. In this experiment, the adversary overshadows uplink messages, which triggers an Identity Request from the base station. Our methodology accurately identified the Identity Requests triggered by this attack, showing the potential our approach has in detecting this family of attacks when applied at scale. To further verify the claim, we conducted the same experiment against a real-world base station and observed the Identity Requests triggered by the overshadowing attack. We note that present downlink overshadow techniques can be extended to make use of any of the messages we enumerate in Table III to collect IMSIs, and that our approach would likewise detect such attacks. Similar to our case with IMSI-Catchers, our detector must remain along the signal path of the injected message to detect it.

VIII. Discussion

We discuss the takeaways from our results, including the effectiveness of prior work against our real-world event capture and potential further optimizations IMSI-Catchers could take to limit detection. Additionally, we provide further discussion on a companion website[4].

A. Efficacy of Prior Work on Event Data

We analyze data gathered from our field experiments to compare the efficacy of our approach against past detection methods. Specifically, we try to determine what other detection methods (if any) would have correctly flagged suspicious base stations during the event, given the characteristics of recorded base stations.

Detector	Detection Metric	Incidence
Prior Work	Unusual Frequencies	0
	New Cell-IDs Detected	0
	New Location Areas	0
	Suspicious BCCH Params	0
	Weak Ciphers Advertised	0
	Unusual RF Power/Position	N/A*

Table V: Number of times each heuristic would have observed suspicious activity during the selected field experiment event. '*' indicates that insufficient information was gathered to make any determination on physical-layer heuristics.

First, we compare the configuration of the tower we flagged as suspicious to the configuration of towers we analyze in the same area later to check for any apparent signs of misconfiguration. This allows us to check if past approaches could have also flagged it given a similar opportunity. We determine that the flagged base station advertised identical country codes, network codes, tracking area codes, and cell identities as a legitimate base station observed in the later capture. From an analysis of downlink data sent by the flagged base station during the event, we further determine that no suspicious cipher suite options (e.g., null cipher) were advertised, and broadcast parameters were within range of those observed in other towers. We note that as we do not record granular physical layer information, we were unable to draw any conclusions on the reliability of detection mechanisms that rely on recorded RF power to catch anomalous stations.

Table V summarizes these heuristics. Our observations suggest that previous IMSI-Catcher detection techniques relying on base station misconfigurations [13], [14], [15], [16] and techniques that do not consider LTE networks [10], [11], [21], [12], [13], [15], [16], [18], [20] *would have missed this notable leak of IMSIs*. This is not unexpected, given that prior IMSI-Catcher patents have described mimicking local base stations [59], [60] and commercial catchers advertise support for LTE-specific attacks [19]. Since our approach detects causal attack traffic rather than potentially correlational variables (e.g., implausible location area) while also supporting multiple generations, it remains resilient to these more sophisticated adversaries.

B. Variance in Message Types

We note that the incidence of message types varied significantly between lab experiments and real-world event captures (seen in Figure 5). The open-source LTE IMSI-Catcher software we tested solely used *Identity Requests* to expose IMSIs, while captured event data revealed only a small number of *Identity Requests* combined with a large number of *Attach Rejects*.

Across all captures of open-source IMSI-Catchers and events, we observe only a small subset of IMSI-exposing

[4] https://www.cellularsecurity.org/marlin

message types previously identified in Table III. Despite the relative infrequency of other messages, however, it is essential that we follow a standards-driven approach to measure all IMSI-exposing connections, not just those we have observed in use. For example, a behavior-based approach relying on the messages observed in our open-source IMSI-Catcher experiments would have failed to correctly identify anomalies during the event capture. Beyond this, we ensure that an adaptive IMSI-Catcher adversary cannot cloak its behavior by switching to a different IMSI-exposing message.

C. Other Event Observations

While near the event, we observed several spikes of IMSI-exposing connections occurring sporadically throughout the two hours we spent observing network traffic. We believe that three possible factors contribute to this result including operator stealth, frequency selection, and IMSI-Catcher mobility.

To avoid detection, an IMSI-Catcher operator can choose to operate their base station in short bursts. This both minimizes the chance that victims notice that their device is having trouble connecting to the network and the chance that any IMSI-Catcher detection techniques notice their transmissions. These results can also be sporadic due to our methodology of cycling through frequencies while running our system. The exceptional behavior we observe occurred on a single frequency, meaning that we did not monitor it continuously during our time window. Finally, we may have observed sporadic spikes due to the mobility of an IMSI-Catcher. To increase the target area, an IMSI-Catcher operator can move their base station using a vehicle (e.g., DRT boxes [99] use IMSI-Catchers equipped to aircraft to track users). If this occurred during our capture, we would only see a spike when the vehicle passes by our location.

The reject messages we observed at the event could have been used as a denial of service attack, though we would have expected to see similar behavior on other frequencies during the event if that was the case. Furthermore, all reject messages used "cause" values that led to an IMSI exposure; a denial of service attack would not have needed to use this cause value as the LTE standards provide 12 "cause" options, many of which are not IMSI-exposing. We would also like to note that the cause value for "congestion" is not IMSI-exposing, suggesting that what we observed was not a response to a lack of availability.

After the event, we disclosed our findings to national-level law enforcement. They thanked us for sharing our research but did not validate our findings. Therefore, we do not have the opportunity to confirm the existence of IMSI-Catcher use at the public event we visited. This limitation is similar to that of censorship detection or other anomaly detection problems where verifying results is an infeasible task. Recognizing this, we offer statistical analysis to strengthen our findings in this setting.

D. Effect on Surveillance

Our method can be used by any party interested in detecting attacks on privacy in cellular networks. IMSI-Catchers can be run by individuals as well as law enforcement agencies [15] and foreign nation-states [6]. Parties concerned about surveillance, especially in sensitive areas (e.g., conflict zones, embassies, military bases), can adopt our approach to monitor frequencies in their area for spikes in IMSI Exposure Ratio.

E. Limitations & Adaptive Adversary

To develop our detection approach, we lean on standards-driven and experimentally-observed behavior on the ratio of IMSI-exposing connections to total connections. While we believe our manual review of these messages to be thorough, we acknowledge the possible existence of additional message flows that lead to IMSI exposure.

Our approach allows the user to scale-up their analysis using several SDRs monitoring different frequencies simultaneously. However, monitoring all frequencies simultaneously would be cost-prohibitive (i.e., dozens of SDRs totaling thousands of dollars), computationally expensive, and potentially difficult to move. Thus, we choose to cycle between common frequencies to maximize our chances of detecting an IMSI-Catcher given these constraints. An adversary using a quick burst of IMSI-exposing messages could therefore evade detection by transmitting on a frequency that we are not currently monitoring. However, we note that an IMSI-Catcher needs to be found by UEs for the adversary to achieve their goals. Therefore, an adversary is highly incentivized to transmit on common frequencies of different network operators to detect users of various UEs. Deviating from this behavior ultimately works against the goals of the adversary.

Adaptive adversaries in this setting can employ several optimizations to minimize the repetition of IMSI-exposing connections. For example, an IMSI-Catcher operator could attempt to conceal their system by collaborating with controlled UEs to produce "filler" connections to force a lower ratio of IMSI-exposing connections. Such an approach may prove challenging, however, as we found during our experiments that phones will often downgrade to previous cellular generations after failing to connect to a base station five times in a row. Therefore, a system implementing this adaption will likely need to orchestrate a collection of trusted devices to force a low IMSI-exposing ratio; such a countermeasure would suffer from severe scalability issues if the IMSI-Catcher is targeting hundreds of devices in a small timeframe.

One such optimization is detecting the continued use of a TMSI after the UE receives a message instructing it to *"delete any stored TMSI"*. We observe this behavior experimentally while operating the open-source LTE IMSI-Catcher; namely, a test smartphone re-attached to a commercial base station after our IMSI-Catcher attack using the same TMSI it used when initially connecting to the IMSI-Catcher. An adversary can take advantage of this TMSI persistence by automatically ignoring a connection request from a UE claiming a TMSI the IMSI-Catcher has seen before. The operator can additionally choose to use specific reject messages with timeout values to instruct the UE not to attempt reconnection for a set time (e.g., *RRC Connection Reject* disallows UE reconnection to that cell for up to 16 seconds [34]). Alternatively, the IMSI-Catcher adversary can generate paging messages containing the TMSIs of previously seen UEs [100]. Any UE that responds to the paging message can then be sent a similar reject message to prevent it from re-attaching to the base station. Such

optimizations offer two incentives for the adversary: increased stealth and a smaller chance that the owner of each UE notices a loss of service.

Despite these adaptations, IMSI-Catchers under our security model will fundamentally produce at least one downlink IMSI-exposing connection per UE in an area to obtain reasonable certainty as to what devices are present. Even an adversary that is only interested in determining the presence or absence of one particular IMSI must query the IMSI of most or all UEs within an area. Therefore, each of these optimizations faces a scalability issue in producing enough "benign" connections to force a low IMSI-exposing ratio in the presence of a large pool of local UEs, thus our approach of observing IMSI-exposing connections thus remains a reliable indicator in a variety of threat scenarios.

F. Applicability to 5G

In 5G SA networks, the IMSI can be protected over the air using public key-based encryption [101]; the resulting encrypted identifier is called a Subscription Concealed Identifier (SUCI). Therefore, an adversary cannot operate an IMSI-Catcher using the same approach used in older cellular generations. However, recent work shows that linkability attacks are still possible in 5G SA networks [36]. In this setting, an adversary can probe for a user in an area if they have linked a previous SUCI or an IMSI to that user.

The SUCI-Catcher attack relies on responses to Authentication Requests to indirectly confirm the identity of a UE. An adversary therefore needs to transmit a large flood of requests *containing the same SUCI* to many nearby UEs to confirm the presence or absence of a known user. To detect a SUCI-Catcher, our approach can be extended to detect a ratio of connections containing identical Authentication Requests, rather than IMSI-exposing messages, to total connections. Furthermore, the approach could use uplink sniffing to check for the specific messages a SUCI-Catcher expects from UEs (e.g., Authentication Failure with cause *Synch Failure*, Authentication Failure with cause *MAC Failure*). Unlike in older cellular networks, a detector would not be able to log personally identifying information from the uplink channel because there is no IMSI sent in the clear.

Finally, an adversary can attempt to downgrade UEs from 5G SA to older protocols, enabling an IMSI-Catcher attack. The 5G standards [101] include measures to prevent such downgrade attacks using the Anti-Bidding down Between Architectures (ABBA) parameter. If the adversary succeeds despite these measures, our methodology can be used to detect the IMSI-Catcher operating using an older cellular technology.

G. Reproducibility

Conducting security research on wireless protocols is a historically difficult task [102], [103], [82], demanding often outdated environments to provide the necessary dependencies. Additionally, reproducibility is a recently highlighted concern in the security community [104]. To streamline the process of reproducibility for researchers in the community, we provide all cellular code in the form of Docker containers that can operate within multiple Linux distributions even with the added difficulty of passing through USB devices (e.g., SDRs).[5] We hope to inspire other researchers to consider reproducibility when developing similar systems in this space.

IX. Related Work

Academic work in cellular security contains many theoretical and implemented solutions to IMSI-Catcher detection. A recent survey [19] in this space states that approaches fall into one of three categories: app-based [15], [26], [57], [105], [58], [16], sensor-based [15], [20], [106], or network-based [25], [11], [22], [23], [24], [26], [54], [27]. We review this body of work, including the advantages and challenges of each approach.

Mobile application solutions are desirable because they can interface directly to a UE and therefore take immediate action upon detection of an IMSI-Catcher. Detection schemes use changing geographical network topology [15], cell tower information consistency [57], [58], and encryption downgrades [58]. Many of these base station configurations, however, are trivial to change for an adversary running an IMSI-Catcher, so these applications naturally produce a considerable number of false negatives. Furthermore, they cannot see network behavior past the interaction between the specific host UE and the base station. This is crucial because an IMSI-Catcher uses protocol-compliant messages to extract identifiers, and the detection of a small number of individual messages sent to a given UE is not indicative of an attack.

Network-based solutions offer a broader view of the network and rely on measurement reports from legitimate base stations [25] or implausible location updates from UEs to base stations [11]. These approaches require cooperation with existing network operators and are therefore unlikely to be implemented. A third possible solution is the use of a Public Key Infrastructure (PKI) to encrypt the IMSI over the air [22], [54], [24], [26], [27], which we group into network-based solutions as it requires a change in the core network. Such a solution is finally being deployed in the form of a SUCI in 5G Standalone networks [36], but cannot protect against IMSI-Catchers that directly target other cellular generations or perform step-down attacks to force UEs to seek base stations in on of these generations.

Sensor-based solutions provide a broader perspective of network traffic without operator cooperation using inexpensive SDR hardware. Similar to the other approaches, previous sensor-based work has relied on geographical network topology [15], [20] and odd base station configurations [20] to detect IMSI-Catchers. Additionally, RF fingerprinting techniques have been combined with Machine Learning to create profiles of base stations on the physical layer [28]. All of these approaches, however, can produce false positives if a legitimate mobile base station is suddenly deployed within their range (e.g., cellular providers providing "cellular-on-wheels" services during sporting events [29]). In contrast, our novel approach looks at the message flow for a given base station for the specific messages necessary to extract an IMSI from a victim device. Any IMSI-Catcher, *regardless of configuration*, must send these messages en masse to achieve their goal.

[5] https://doi.org/10.5281/zenodo.14262356

X. Conclusion

In this work, we propose a detection method for attacks against user privacy in multiple generations of cellular networks. Unlike previous detection methods, we review cellular standards to determine the exact messages IMSI-Catchers in each cellular generation need to send to achieve their goal and look for a high ratio of connections including these messages as evidence of IMSI-Catcher transmissions. We first confirm a low ratio of such connections in commercial networks using hundreds of hours of network captures, then show opposing behavior when analyzing open-source IMSI-Catcher implementations. Finally, we use our detection methodology to reveal evidence of IMSI-Catcher behavior during a large event in the United States ($p << 0.005$). These results suggest that commercial networks and IMSI-Catchers have opposing goals when treating the cellular IMSI over the air. Namely, a commercial network will minimize IMSI transmissions while an IMSI-Catcher will maximize them. Our methodology therefore offers a causal approach to detection compared to the correlational approach of past work.

Acknowledgment

This work was supported in part by the National Science Foundation grants CNS-1933208 and CNS-2055014. Any findings and opinions expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies. Finally, the authors would like to thank the members of the Artifact Evaluation Committee for their efforts in improving our open-source artifact and William Enck for his feedback.

References

[1] S. Park, "Why We Cannot Win: On Fake Base Stations and Their Detection Methods," Ph.D. dissertation, Technische Universität Berlin, 2023.

[2] S. Biddle, "Long-Secret Stingray Manuals Detail How Police Can Spy on Phones," https://theintercept.com/2016/09/12/long-secret-stingray-manuals-detail-how-police-can-spy-on-phones/, 2016.

[3] S. Margaritelli, "How to Build Your Own Rogue GSM BTS for Fun and Profit," https://www.evilsocket.net/2016/03/31/How-To-Build-Your-Own-Rogue-GSM-BTS-For-Fun-And-Profit/index.html, 2016.

[4] R. Chirgwin, "Now You, Too, Can Snoop on Mobile Users from 3G to 5G with a Raspberry Pi and €1,100 of Gizmos," https://www.theregister.com/2018/12/05/mobile_users_can_be_tracked_with_cheap_kit_aka_protocol/, 2018.

[5] S. R. S. (SRS), "srsRAN - Your Own Mobile Network," https://www.srslte.com/, 2022.

[6] P. A. Johansen, "Secret Surveillance of Norway's Leaders Detected," https://www.aftenposten.no/norge/i/q36m/secret-surveillance-of-norways-leaders-detected, 2014.

[7] C. H. Romine, "Bolstering Data Privacy and Mobile Security: An Assessment of IMSI Catcher Threats," https://www.nist.gov/speech-testimony/bolstering-data-privacy-and-mobile-security-assessment-imsi-catcher-threats, 2018.

[8] D. Strobel, "Imsi catcher," *Chair for Communication Security, Ruhr-Universität Bochum*, vol. 14, 2007.

[9] D. Fox, "Der IMSI Catcher," *Datenschutz und Datensicherheit*, vol. 26, no. 4, pp. 212–215, 2002.

[10] T. Van Do, H. T. Nguyen, N. Momchil, and V. T. Do, "Detecting IMSI-catcher using Soft Computing," in *Soft Computing in Data Science: First International Conference*. Springer, 2015, pp. 129–140.

[11] A. Dabrowski, G. Petzl, and E. R. Weippl, "The Messenger Shoots Back: Network Operator Based IMSI Catcher Detection," in *Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses (RAID)*, 2016.

[12] H. Alrashede and R. A. Shaikh, "IMSI Catcher Detection Method for Cellular Networks," in *Proceedings of the International Conference on Computer Applications & Information Security (ICCAIS)*, 2019.

[13] A. Wilson, "SITCH: Situational Information from Telemetry and Correlated Heuristics," in *DEF CON 24*, 2016.

[14] E. Briggs and Z. Ji, "Detection of a rogue base station," US Patent 10,129,283, 2018.

[15] A. Dabrowski, N. Pianta, T. Klepp, M. Mulazzani, and E. Weippl, "IMSI-Catch Me If You Can: IMSI-Catcher-Catchers," in *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2014.

[16] S. Park, A. Shaik, R. Borgaonkar, A. Martin, and J.-P. Seifert, "White-Stingray: Evaluating IMSI Catchers Detection Applications," in *Proceedings of the USENIX Workshop on Offensive Technologies (WOOT)*, 2017.

[17] C. Quintin, "Detecting Fake 4G Base Stations in Real Time," in *Blackhat USA*, 2020.

[18] GSMK, "GSMK Overwatch: IMSI Catcher Detection," https://www.gsmk.de/products/network-security/, 2024.

[19] S. Park, A. Shaik, R. Borgaonkar, and J.-P. Seifert, "Anatomy of Commercial IMSI Catchers and Detectors," in *Proceedings of the ACM Workshop on Privacy in the Electronic Society (WPES)*, 2019.

[20] P. Ney, I. Smith, G. Cadamuro, and T. Kohno, "SeaGlass: Enabling City-Wide IMSI-Catcher Detection," in *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2017.

[21] S. Steig, A. Aarnes, T. Van Do, and H. T. Nguyen, "A Network Based IMSI Catcher Detection," in *Proceedings of the International Conference on IT Convergence and Security (ICITCS)*. IEEE, 2016, pp. 1–6.

[22] F. van den Broek, R. Verdult, and J. de Ruiter, "Defeating IMSI Catchers," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2015.

[23] T. V. Do, H. T. Nguyen, N. Momchil, and V. T. Do, "Detecting IMSI-Catcher Using Soft Computing," in *Proceedings of the International Conference on Soft Computing in Data Science (SCDS)*, 2015.

[24] K. Norrman, M. Näslund, and E. Dubrova, "Protecting IMSI and User Privacy in 5G Networks," in *Proceedings of the International Conference on Mobile Multimedia Communications (MobiMedia)*, 2016.

[25] S. Steig, A. Årnes, T. van Do, and H. T. Nguyen, "A Network Based IMSI Catcher Detection," in *Proceedings of the International Conference on IT Convergence and Security (ICITCS)*, 2016.

[26] E. C. Jimenez, P. K. Nakarmi, M. Näslund, and K. Norrman, "Subscription Identifier Privacy in 5G Systems," in *Proceedings of the International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT)*, 2017.

[27] M. S. A. Khan and C. J. Mitchell, "Trashing IMSI Catchers in Mobile Networks," in *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2017.

[28] Z. Zhuang, X. Ji, T. Zhang, J. Zhang, W. Xu, Z. Li, and Y. Liu, "FBSleuth: Fake Base Station Forensics via Radio Frequency Fingerprinting," in *Proceedings of the ACM Asia Conference on Computer and Communications Security (ASIACCS)*, 2018.

[29] D. Parenti, "Everything You Need to Know About Cell Towers on Wheels," https://www.mccr.info/blog/everything-you-need-to-know-about-cell-towers-on-wheels, 2021.

[30] M. Kotuliak, S. Erni, P. Leu, M. Roeschlin, and S. Capkun, "LTrack: Stealthy Tracking of Mobile Phones in LTE," in *Proceedings of the USENIX Security Symposium*, 2022.

[31] M. C. Burgess, "What is a Stingray? Is it surveilling you?" https://burgess.house.gov/news/documentsingle.aspx?DocumentID=402684, 2020.

[32] 3GPP, "TS 36.321 - Medium Access Control (MAC) Protocol Specification," 2012.

[33] ——, "TS 24.301 - Non-Access-Stratum (NAS) Protocol for Evolved Packet System (EPS)," 2013.

[34] ——, "TS 36.331 - Radio Resource Control (RRC) Protocol Specification," 2010.

[35] B. Karakoc, N. Fürste, D. Rupprecht, and K. Kohls, "Never Let Me Down Again: Bidding-Down Attacks and Mitigations in 5G and 4G," in *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2023.

[36] M. Chlosta, D. Rupprecht, C. Pöpper, and T. Holz, "5G SUCI-Catchers: Still Catching Them All?" in *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2021.

[37] Oros42, "Oros42/IMSI-catcher: This Program Shows You IMSI Numbers of Cellphones Around You." https://github.com/Oros42/IMSI-catcher, 2022.

[38] C. Peeters, T. Tucker, A. Jain, K. Butler, and P. Traynor, "Leopard-Seal: Detecting Call Interception via Audio Rogue Base Stations," in *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2023.

[39] Z. Li, W. Wang, C. Wilson, J. J. Chen, C. Qian, T. Jung, L. Zhang, K. Liu, X. Li, and Y. Liu, "FBS-Radar: Uncovering Fake Base Stations at Scale in the Wild," in *Proceedings of the ISOC Network and Distributed System Security (NDSS) Symposium*, 2017.

[40] Y. Zhang, B. Liu, C. Lu, Z. Li, H. Duan, S. Hao, M. Liu, Y. Liu, D. Wang, and Q. Li, "Lies in the Air: Characterizing Fake-Base-Station Spam Ecosystem in China," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2020.

[41] B. Hong, S. Bae, and Y. Kim, "GUTI Reallocation Demystified: Cellular Location Tracking with Changing Temporary Identifier," in *Proceedings of the ISOC Network and Distributed System Security (NDSS) Symposium*, 2018.

[42] E. Kelly, "Bipartisan Bill Seeks Warrants for Police Use of 'Stingray' Cell Trackers," https://www.usatoday.com/story/news/politics/onpolitics/2017/02/15/bipartisan-bill-seeks-warrants-police-use-stingray-cell-trackers/97954214/, 2017.

[43] S. Musgrave, "Boston Police Argue That Releasing StingRay Docs Makes Devices "Essentially Useless"," https://www.muckrock.com/news/archives/2015/jun/08/boston-police-argue-releasing-stingray-docs-makes-/, 2015.

[44] S. Margaritelli, "Building Your Own Rogue GSM Basestation with a BladeRF," https://www.rtl-sdr.com/building-your-own-rogue-gsm-basestation-with-a-bladerf/, 2016.

[45] L. H. Newman, "DC's Stingray Mess Won't Get Cleaned Up," https://www.wired.com/story/dcs-stingray-dhs-surveillance/, 2018.

[46] K. Zetter, "How Cops Can Secretly Track Your Phone," https://theintercept.com/2020/07/31/protests-surveillance-stingrays-dirtboxes-phone-tracking/, 2020.

[47] J. Fenton, "Key Evidence in City Murder Case Tossed Due to Stingray Use," https://www.baltimoresun.com/news/crime/bs-md-ci-stingray-murder-evidence-suppressed-20160425-story.html, 2016.

[48] E. Priezkalns, "New German Law Forces Telcos to Assist State Surveillance with IMSI-Catchers," https://commsrisk.com/new-german-law-forces-telcos-to-assist-state-surveillance-with-imsi-catchers/, 2021.

[49] C. Cullen and B. Bureau, "Someone is Spying on Cellphones in the Nation's Capital," https://www.cbc.ca/news/politics/imsi-cellphones-spying-ottawa-1.4050049, 2017.

[50] D. O'Dea, "The StingRay You've Never Heard Of: How One of the Most Effective Tools in Law Enforcement Operates Behind a Veil of Secrecy," https://mjlst.lib.umn.edu/2021/11/30/the-stingray-youve-never-heard-of-how-one-of-the-most-effective-tools-in-law-enforcement-operates-behind-a-veil-of-secrecy/, 2021.

[51] SS7ware, "YateBTS," https://yatebts.com/, 2018.

[52] Nuand, "bladeRF 2.0 micro xA4," https://www.nuand.com/product/bladerf-xa4/, 2023.

[53] Ettus, "USRP B210 USB Software Defined Radio (SDR)," https://www.ettus.com/all-products/ub210-kit/, 2020.

[54] S. R. Hussain, M. Echeverria, A. Singla, O. Chowdhury, and E. Bertino, "Insecure Connection Bootstrapping in Cellular Networks: The Root of All Evil," in *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2019.

[55] B. Brenninkmeijer, "Catching IMSI-Catcher-Catchers: An Effectiveness Review of IMSI-Catcher-Catcher Applications," Ph.D. dissertation, Radboud University, 2016.

[56] Y. Nasser, "Gotta Catch 'Em All: Understanding How IMSI-Catchers Exploit Cell Networks," Electronic Frontier Foundation (EFF), Tech. Rep., 2019.

[57] C. Privacy, "Android IMSI-Catcher Detector by CellularPrivacy," https://cellularprivacy.github.io/Android-IMSI-Catcher-Detector/, 2015.

[58] SRLabs, "SnoopSnitch," https://opensource.srlabs.de/projects/snoopsnitch, 2014.

[59] P. Martin and B. Dolby, "Acquiring identity parameter," US Patent App. 20090023424A1, 2009.

[60] A. Pridmore, P. Martin, and R. Dolby, "Acquiring identity parameters by emulating base stations," US Patent 9215585B2, 2015.

[61] H. Yang, S. Bae, M. Son, H. Kim, S. M. Kim, and Y. Kim, "Hiding in Plain Signal: Physical Signal Overshadowing Attack on LTE," in *Proceedings of the USENIX Security Symposium*, 2019.

[62] S. Erni, M. Kotuliak, P. Leu, M. Roeschlin, and S. Capkun, "AdaptOver: Adaptive Overshadowing Attacks in Cellular Networks," in *Proceedings of the International Conference On Mobile Computing And Networking (MobiCom)*, 2022.

[63] A. Kredo, "Malicious Entity Could Be Tracking Phones of Domestic, Foreign Officials," https://freebeacon.com/national-security/hack-attack-cell-phone-data-dc/, 2017.

[64] ETSI, "TS/SMG-030408QR2 - Mobile Radio Interface Layer 3 Specification," 1996.

[65] 3GPP, "TS 24.008 - Core Network Protocols," 2016.

[66] S. Udar and R. Borgaonkar, "Darshak," https://github.com/darshakframework/darshak, 2024.

[67] skibapps, "Cell Spy Catcher," https://cell-spy-catcher-anti-spy.en.softonic.com/android, 2024.

[68] GALAN, "GSM Spy Finder," https://gsm-spy-finder.fileplanet.com/apk, 2024.

[69] M. Bailey, D. Dittrich, E. Kenneally, and D. Maughan, "The Menlo Report," *IEEE Security & Privacy*, vol. 10, no. 2, pp. 71–75, 2012.

[70] L. Vargas, L. Blue, V. Frost, C. Patton, N. Scaife, K. R. Butler, and P. Traynor, "Digital Healthcare-Associated Infection: A Case Study on the Security of a Major Multi-Campus Hospital System," in *Proceedings of the ISOC Network and Distributed System Security (NDSS) Symposium*, 2019.

[71] B. Lipton and C. Quintin, "The Next Generation of Cell-Site Simulators is Here. Here's What We Know." https://www.eff.org/deeplinks/2024/06/next-generation-cell-site-simulators-here-heres-what-we-know, 2024.

[72] P. Krysik, "The Gr-Gsm Project," https://github.com/ptrkrysik/gr-gsm, 2021.

[73] T. D. Hoang, C. Park, M. Son, T. Oh, S. Bae, J. Ahn, B. Oh, and Y. Kim, "LTESniffer: An Open-source LTE Downlink/Uplink Eavesdropper," in *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2023.

[74] T. D. Byrd, "CSAI," https://github.com/tkb140/CSAI, 2021.

[75] T. Luijkman, "E-UTRAN LTE Visualiser (ELVis)," https://github.com/thomasluijkman/4Gvisualiser, 2022.

[76] N. Ludant, P. Robyns, and G. Noubir, "From 5G Sniffing to Harvesting Leakages of Privacy-Preserving Messengers," in *Proceedings of the IEEE Symposium on Security and Privacy*, 2023.

[77] Nuand, "bladeRF x40," https://www.nuand.com/product/bladerf-x40/, 2022.

[78] Consumer and Governmental Affairs, "Plan Ahead for Phase Out of 3G Cellular Networks and Service | Federal Communications Commission," https://www.fcc.gov/consumers/guides/plan-ahead-phase-out-3g-cellular-networks-and-service, 2022.

[79] A. Jasso, "Cellular Frequency Bands: A Simple Breakdown," https://www.signalboosters.com/blog/cellular-frequency-bands-a-simple-breakdown/, 2023.

[80] C. S. Limited, "Cellular Coverage and Tower Map," https://www.cellmapper.net/, 2023.

[81] D. Klyuykov, "How to Get 3G Working on the UmTRX," https://fairwaves.co/blog/openbts-umts-3g-umtrx/, 2016.

[82] J. Ross, "How To Run Your Own Cell Tower," https://freezion.com/?p=214, 2018.

[83] K. van Rijsbergen, "The Effectiveness of a Homemade IMSI Catcher Build with YateBTS and a BladeRF," Semantic Scholar, Tech. Rep., 2016.

[84] D. Rupprecht, K. Kohls, T. Holz, and C. Pöpper, "Breaking LTE on Layer Two," in *Proceedings of the IEEE Symposium on Security and Privacy*, 2019.

[85] T. D. Byrd, V. Marojevic, and R. P. Jover, "CSAI: Open-Source Cellular Radio Access Network Security Analysis Instrument," in *Proceedings of the IEEE Vehicular Technology Conference (VTC2020-Spring)*, 2020.

[86] M. Kotuliak, "LTE Monitoring," Ph.D. dissertation, ETH Zurich, 2020.

[87] M. Chlosta, D. Rupprecht, T. Holz, and C. Pöpper, "LTE Security Disabled: Misconfiguration in Commercial Networks," in *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2019.

[88] R. Borgaonkar, L. Hirschi, S. Park, and A. Shaik, "New Privacy Threat on 3G, 4G, and Upcoming 5G AKA Protocols," *Proceedings on Privacy Enhancing Technologies (PoPETs)*, vol. 2019, no. 3, pp. 108–127, 2019.

[89] A. Shaik, R. Borgaonkar, S. Park, and J.-P. Seifert, "New Vulnerabilities in 4G and 5G Cellular Access Network Protocols: Exposing Device Capabilities," in *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2019.

[90] ——, "On the Impact of Rogue Base Stations in 4G/LTE Self Organizing Networks," in *Proceedings of the ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*, 2018.

[91] V. Adarsh, M. Nekrasov, E. Zegura, and E. Belding, "Packet-level Overload Estimation in LTE Networks using Passive Measurements," in *Proceedings of the ACM Internet Measurement Conference (IMC)*, 2019.

[92] M. Echeverria, Z. Ahmed, B. Wang, M. F. Arif, S. R. Hussain, and O. Chowdhury, "PHOENIX: Device-Centric Cellular Network Protocol Monitoring using Runtime Verification," in *Proceedings of the ISOC Network and Distributed System Security (NDSS) Symposium*, 2021.

[93] A. Shaik, R. Borgaonkar, N. Asokan, V. Niemi, and J.-P. Seifert, "Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems," in *Proceedings of the ISOC Network and Distributed System Security (NDSS) Symposium*, 2016.

[94] M. Iedema, "OpenBTS-UMTS," https://github.com/RangeNetworks/OpenBTS-UMTS, 2021.

[95] D. J. Benjamin and J. O. Berger, "Three Recommendations for Improving the Use of p-Values," *The American Statistician*, vol. 73, no. sup1, pp. 186–191, 2019.

[96] S. S. Shapiro and M. B. Wilk, "An Analysis of Variance Test for Normality (Complete Samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965.

[97] H. B. Mann and D. R. Whitney, "On a Test of Whether One of Two Random Variables is Stochastically Larger than the Other," *The Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 50–60, 1947.

[98] Amarisoft, "AMARI Callbox Series," https://www.amarisoft.com/products/test-measurements/amari-lte-callbox/, 2019.

[99] S. Khandelwal, "Police Using Planes Equipped with Dirtbox to Spy on Your Cell Phones," https://thehackernews.com/2016/01/drtbox-cellphone-interception.html, 2016.

[100] E. Goldfarb, "Systems and Methods for Identifying Rogue Base Stations," United States Patent 0344844, Verint Systems Ltd, 2013.

[101] 3GPP, "Security Architecture and Procedures for 5G System," 2018.

[102] G. Hernandez and T. Tucker, "Creating a Cellular Testbed with YateBTS and srsLTE," https://hernan.de/blog/creating-a-cellular-testbed-with-yatebts-and-srslte/, 2020.

[103] M. Patel, "How to Build an IMSI Catcher to Intercept GSM Traffic," https://www.paladion.net/blogs/how-to-build-an-imsi-catcher-to-intercept-gsm-traffic, 2020.

[104] D. Olszewski, A. Lu, C. Stillman, K. Warren, C. Kitroser, A. Pascual, D. Ukirde, K. Butler, and P. Traynor, ""Get in Researchers; We're Measuring Reproducibility": A Reproducibility Study of Machine Learning Papers in Tier 1 Security Conferences," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2023.

[105] K. Nohl, "CatcherCatcher," https://opensource.srlabs.de/projects/mobile-network-assessment-tools/wiki/CatcherCatcher, 2013.

[106] E. F. F. (EFF), "Crocodile Hunter," https://www.eff.org/pages/crocodile-hunter, 2021.

APPENDIX A
STATISTICAL TESTS

Type	Target Data	p-value
Shapiro-Wilk Test (normal distribution)	Provider 1 Captures	$5.8e-19$
	Provider 2 Captures	$8.7e-21$
	Euro Provider 1 Captures	$9.0e-16$
	Euro Provider 2 Captures	$3.3e-15$
	Provider 1 Football Captures	$3.0e-03$
	Provider 2 Football Captures	$6.3e-08$
	Event Benchmark	$3.6e-09$
	Event	$4.4e-02$
Result: Nonparametric statistical test required.		

Table VI: Summary of results from our Shapiro-Wilk statistical tests. Not all populations pass this test (i.e., tests on our provider 1 football capture and event capture produce $p > 0.005$), meaning that we cannot treat all populations as normally distributed and therefore must use a nonparametric statistical test to compare them.

APPENDIX B
COMPARISON OF *LTESniffer* AND *DLProbe*

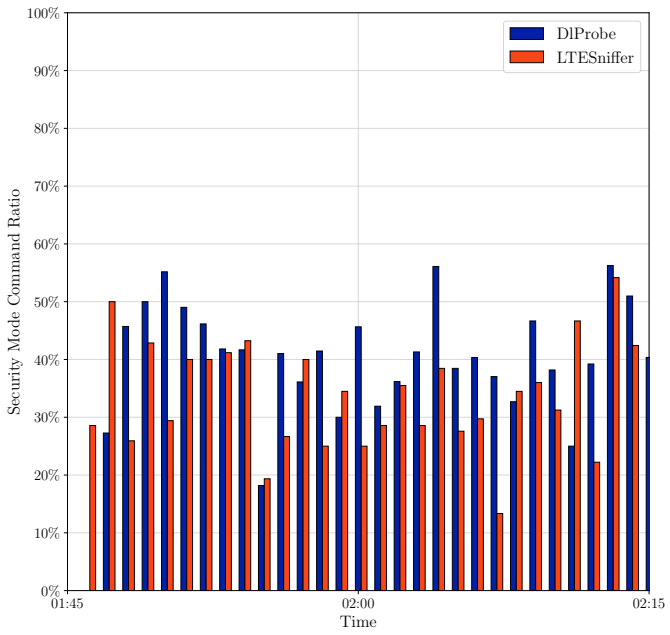


Figure 7: Comparison of *LTESniffer* and *DLProbe* using the ratio of RRC Security Mode Command messages to UE connections.

To compare *LTESniffer* and *DLProbe* we run the two tools in parallel camping on the same cell. In our IMSI-Catching detection use case, we are analyzing the IMSI Exposure Ratios observed over different captures. To compare the two tools, we should compare if the IMSI Exposure Ratios differ for the two captures made in parallel. However, IMSI-exposing message tends to occur rarely on operators' cells, therefore, the comparison would not be precise. Instead, in Figure 7 we compare ratios of the RRC Security Mode Command message, which similarly to Identity Request or Attach Reject, is transmitted during an attach procedure. We conclude that even though the ratios observed by the two tools do not match precisely in every time bin, on average, they observed similar ratios of RRC Security Mode Command messages to UE connections. Given that the IMSI Exposure Ratios for IMSI-Catchers and honest base stations differ significantly, we conclude that these two tools can be used interchangeably for our use case.

Radios	CPU Usage (%)	Memory Usage (MB)
1	64.98	510.09
2	133.64	968.29
3	175.15	1415.64

Table VII: System resources reported by *docker stats* scaled with the number of simultaneously running USRP software-defined radios. CPU and memory usage each appear to scale linearly with an increase in the number of radios.

APPENDIX C
RESOURCE ANALYSIS

We implement our system as a Docker container to support reproducibility for other researchers. Additionally, we provide performance metrics using *docker stats* to investigate how increasing the number of software-defined radios affects the resources necessary to run our container; *docker stats* enables live monitoring of the resources used by a specified docker container during runtime. To gather these metrics, we perform three experiments in which we monitor resource usage of our docker container when using one, two, and three software-defined radios for five minutes each. During each experiment, we collect performance metrics in five second intervals and take the average value for each metric. We use USRP B210s [53] as our cellular sniffers for these experiments. For reference, our host computer runs an i5-12500H CPU and 16GB of RAM.

Table VII exhibits how memory usage and CPU usage change with additional USRP radios. We observe that both metrics increase linearly with the number of radios; these results are expected as our program must create a new thread for each available radio. Additionally, these results indicate that while running a moderate number of radios does not require excessive resources, operating dozens of radios simultaneously would incur a heavy load on the host computer. Monitoring a large number of cellular frequencies simultaneously therefore would require a very performant host machine and a large financial investment. To avoid these issues, users can employ our frequency cycling approach with a modest number of radios.

APPENDIX D
ARTIFACT APPENDIX

A. Description & Requirements

1) How to access: Our code is hosted using a permanent DOI with Zenodo[6] and additionally as a publicly-available repository on GitHub.[7]

2) Hardware dependencies: Our detector requires one or more USRP B210 software-defined radios. However, we provide our existing captures that can be analyzed and plotted without the need for this specialized hardware.

3) Software dependencies: To recreate our plots, either a Python 3 environment or docker are required. To run the full detector, a Unix host machine with docker is required.

4) Benchmarks: N/A

B. Artifact Installation & Configuration

Begin by cloning our GitHub repository, which supports usage through a Python virtual environment or a docker container. This code can also be downloaded from our Zenodo repository.

```
git clone https://github.com/MarlinDetection/Marlin
cd Marlin
```

1) Python Virtual Environment Option: Create a Python virtual environment with a provided `requirements.txt` resource to install necessary Python package dependencies.

```
# From the "Marlin" repository
sudo apt-get install tshark python3-dev python3-pip
     python3.10-venv
# Create virtual environment
cd code
python3 -m venv ./venv
source ./venv/bin/activate
python3 -m pip install -r requirements.txt
```

2) Analysis Docker Container Option: Use our provided *Dockerfile* to create a Docker image containing all environments and scripts.

```
# From the "Marlin" repository
cd docker/marlin-data
docker build -t marlin-data .
# Start Docker container from image
docker run -it marlin-data
```

3) Marlin Tool Docker Container Option: We provide a separate docker container to run our detector (requiring specialized hardware) that collects and analyzes new data. This container only works with Unix host systems, as the USRP must be detectable when passing through the `/dev/bus/usb` directory. Please note that large dependency libraries cause long build times.

```
# From the "Marlin" repository
cd docker/marlin
docker build -t marlin .
# Start Docker container from image
docker run -it --privileged --device /dev/bus/usb
    /:/dev/bus/usb/ marlin
```

[6]https://doi.org/10.5281/zenodo.14262356
[7]https://github.com/MarlinDetection/Marlin/tree/v0.0

C. Experiment Workflow

Our repository is organized as follows:

- `(Marlin/data)`: data collected from hours of cellular network traffic. Our analysis scripts process this data.
- `(Marlin/code/analysis)`: scripts that process cellular network captures located in the *data* directory and reproduce the plots and tables in our paper.
- `(Marlin/code/marlin)`: code necessary to run our *Marlin* detector that monitors for IMSI-Catchers and collects new data.
- `(Marlin/docker)`: Dockerfiles that enable containerized reproducibility of our work.

D. Major Claims

- **(C1) Commercial cell towers exhibit low IMSI-exposure ratios:** Our evaluation of our system captured hundreds of hours of cellular traffic operating on commerical cell towers. We provide the resulting data in our repository for our Python scripts to analyze. Our benchmark experiments show that the IE ratio remains low for all test cases, captured in the file `benchmark.py`.
- **(C2) Open-source IMSI-Catchers exhibit high IE ratios:** We operate lab base stations as IMSI-Catchers using open-source software and commodity hardware in a Faraday cage. During these tests, we run our detection system within the isolated environment and analyze traffic between our IMSI-Catchers and test phones. We observe a 100% IE ratio for all three IMSI-Catchers from each cellular generation tested (e.g., 2G GSM, 3G UMTS, 4G LTE). These results are captured in the script `lab.py`.
- **(C3) We detect an anomalous base station at the courthouse event we attended**: Unlike our benchmark tests, we provide a capture for a base station transmitting during a notable event. The results of this experiment are captured in the script `event.py`. We observe large spikes in IE ratio during the event, providing strong evidence of IMSI-Catcher presence.

E. Evaluation

1) Experiment (E1): [Verify Experimental Results] [15 human-minutes + 5 compute-minutes]: after setting up our environment, generate each plot and numerical result presented in our work from source data provided in our repository.

[How to] Our scripts process the captures in the `data` directory. Each script outputs a combination of `*.pdf` files and console output.

[Preparation] After installing our repository and building a virtual environment, run the following lines of code to activate the virtual environment if necessary. Note that our docker containers perform this step automatically.

```
# From the "Marlin" repository
cd code
source ./venv/bin/activate
cd analysis
```

[Execution] Run each script using Python, all of which will output one or more pdf documents including the plots found in our paper.

[Results] Our scripts regenerate the plots and statistics found in our paper. Run any reproducibilty script using the command `python3 <script-name.py>`.

- `benchmark.py` (Figure 4) plots the data from our benchmark experiments for GSM and LTE network traffic. Generated plots include U.S. experiments on two LTE network providers (`benchmark-provider-1.pdf`, `benchmark-provider-2.pdf`), European experiments on two LTE network providers (`benchmark-euro.pdf`), and U.S. experiments on one GSM network provider (`benchmark-gsm.pdf`). We observe relatively low IMSI-exposing ratios during these benchmark experiments.
- `comparison.py` (Figure 7) compares the performance of two different LTE network analyzers that we use when collecting data. We perform this comparison by running both detectors simultaneously, then looking for what percentage of connections include the common Security Mode Command LTE message. The resulting plot (`comparison.pdf`) shows that each detector produces similar results during a two hour test.
- `event.py` (Figure 5) plots the data for from our event captures in pdf format. Generated plots include data from the day of the event (`event.py`) and a benchmark capture from the same location on a different day (`event-benchmark.pdf`). We observe significantly different IMSI-exposing ratios between these two days, with notably large spikes appearing during the event; these spikes provide strong evidence of IMSI-Catcher presence.
- `lab.py` (Section VII.B) analyzes the data from our lab experiments and outputs the average IMSI-exposing ratio during each experiment. During these experiments, we operated GSM, UMTS, and LTE IMSI-Catchers in a controlled environment and found that every test produced IMSI-exposing ratios of 100%.
- `statistics.py` (Figure 6, Table VI) performs Shapiro-Wilk and Mann-Whitney statistical tests on our data. Results of these tests are printed to console while all data is plotted in a single violin plot (`violin-plot.pdf`). For the Mann-Whitney tests, each capture is compared to the event capture to test for statistical significance of the event.

2) Experiment (E2): [Run the Detector] [15 human-minutes + 15 compute-minutes]: run the `marlin` detector to monitor for IMSI-Catchers.

[How to] Start by opening the marlin docker container with USB device passthrough using the command:

```
docker run -it --device /dev/bus/usb/:/dev/bus/usb/
     marlin
```

Our `marlin.py` script has multiple prerequisites to run properly, including:

- USRP B210 software-defined radio attached to the host computer using a USB 3.0 connection. The host machine also needs to have the USRP library installed and be able to recognize the software-defined radio.
- Configured Python environment.
- List of frequencies to analyze, provided by default.
- Configuration file, provided by default.

[Preparation] Our docker container will automatically activate a Python virtual environment and create a default `marlin.ini` configuration file. This file can be edited directly if needed. Additionally, the `frequencies.txt` file can be edited to change the list of target frequencies. This file includes ten popular frequencies by default.

[Execution] Run the `marlin` script, which will output results directly to the console.

```
python3 marlin.py -c marlin.ini
```

[Results] The script will output IMSI-exposing ratio results to the console for each base station it analyzes. The results on this experiment depend on the cellular activity of the surrounding area and will resemble the following example:

```
EARFCN <#>: Searching for cell using radio <#>.
EARFCN <#>: Found cell using radio <#>.
EARFCN <#>: IMSI-exposing ratio = <#>%.
```

These results will also be saved to a log file located at:

```
./locations/<location>/<date>/marlin.log
```

3) Experiment (E3): [Visualize New Data] [15 human-minutes + 15 compute-minutes]: analyze existing package captures.

[How to] Provide a packet capture to one of the `parse-[gsm,umts,lte].py` scripts available in the `code/analysis` directory. The script will separate unique connections from the capture and note if any connections contain an IMSI-exposing message. We provide a local `example.pcap` file for testing.

[Preparation] Our docker container will automatically activate a Python virtual environment and create a default `marlin.ini` configuration file.

[Execution] Run one of the parse scripts depending on the type of traffic collected. Then, run the `plot.py` script on the output to visualize the data. Please note that the parse scripts can take a long time to execute if the input packet capture is large.

```
python3 parse-[gsm,umts,lte].py <capture>.pcap
python3 plot.py output.pkl
```

When using the example file, use the following commands:

```
python3 parse-lte.py example.pcap
python3 plot.py output.pkl
```

[Results] The parse scripts will output a pickle file called `output.pkl` of your capture in the current directory, which can then be supplied to the `plot.py` script for visualization. The final plot will be saved as `plot.pdf`.