Gotta Catch ‘Em All

UNDERSTANDING HOW IMSI-CATCHERS EXPLOIT CELL NETWORKS (PROBABLY)

ELECTRONIC FRONTIER FOUNDATION

Author: Threat Lab

A publication of the Electronic Frontier Foundation, 2018.

View this report online:

Gotta Catch ’Em All

UNDERSTANDING HOW IMSI-CATCHERS EXPLOIT CELL NETWORKS (PROBABLY)

AUTHOR
Threat Lab

July 1st 2019

Section 1: Introduction

You’ve probably heard of Stingrays or IMSI-catchers, which belong to the broader category of “Cell Site Simulators” (CSSs). These devices let their operators “snoop” on the phone usage of people nearby. There’s a lot of confusion about what CSSs are actually capable of, and different groups—from activists to policy makers to technologists—understand them differently.

In the research community, there has been a tendency to dismiss the prevalence of CSS and the threat they pose to the public. Congress recently asked the Department of Homeland Security for more information about their use by federal law enforcement, as well as state and local partners. It's unclear how much oversight the Department has been exercising, and when it comes to state and local law enforcement, only a few cities have any protections at all. Many activists aren’t aware that CSSs could be in use around them without their knowledge, particularly during protests. The truth is that CSSs are significantly more widespread than most policy makers, researchers, and activists are aware, and their danger to privacy is more significant than most realize. Of course, it’s hard to acknowledge the prevalence of CSSs when law enforcement goes to great lengths to keep information about them from the public.

There is a plethora of low-level academic research in the area of cell network security, and many high-level posts that don’t really explain in any meaningful detail what’s going on with “IMSI-catcher” type cell network attacks. Our goal is to bridge that gap, and with this post we hope **to make accessible the technical inner workings of CSSs, or rather, the details of the kind of attacks they might rely on**. For example, what are the different kinds of location tracking attacks and how do they actually work? Another example: it’s also widely believed that CSSs are capable of communication interception, but what are the known limits around cell network communication interception and how does that actually work?

We won’t be updating this post with new kinds of attacks as they come out, and we can’t cover every potentially relevant detail of every attack we explain, but this post should form a basis for non-experts to better understand new attacks.

Section 2: Necessary background info

There’s a lot of confusion about what CSSs actually do and how they do it. This confusion comes from the fact that the term “cell site simulator” actually encapsulates quite a variety of different cell network attacks that have evolved significantly over the last 25 years or so. Adding to the confusion is the fact that the term “IMSI-catcher” is both used interchangeably with “cell site simulator” and also refers to specific capabilities that some CSSs have.

A very important distinction when talking about CSSs is which cell network generations they use when operating. The term "cell network generation" refers to the complete set of operating protocols covering everything from how cell towers are laid out geographically to how a mobile phone establishes a connection with a cell tower.

Here's a high-level overview of the most relevant cell network generations:

- 2G (e.g. GSM): the oldest type of cell network still in use and still very widely used. 2G only supports calling/texting, but in 2.5G the capability to support data transmission (e.g. email and Internet access) was introduced.

- 3G (e.g. UMTS or CDMA2000): improved upon 2G by having much faster data rates (which could support video calls, for example) and adding better security (more on this later).

- 4G (e.g. LTE or WiMax): significantly faster speeds and better security.

The specifications for these networks are developed by working groups organized by the 3GPP,[1] an international organization that any group can apply to join (though it has a high membership fee). Members typically include mobile carriers, university research labs, and wireless gear manufacturers (including surveillance tech manufacturers).

It's important to note that in practice there's often a lot of variance between what the specifications say and what actually ends up being implemented. This is usually due to (1) implementers needing to differ from the specifications for practical reasons (many parts of the specifications get marked as optional), and (2) mistakes.

There's a bit more vocabulary and background that needs to be introduced:

- IMSI (International Mobile Subscriber Identity): the unique identifier linked to your SIM card that is one of the pieces of data used to authenticate you to the mobile network. It's meant to be kept private (because, as we'll see later, it can be linked to your physical location and your phone calls/messages/data).

- TMSI: upon first connecting to a network, the network will ask for your IMSI to identify you, and then will assign you a TMSI (Temporary Mobile Subscriber Identifier) to use while on their network. The purpose of the pseudonymous TMSI is to try and make it difficult for anyone eavesdropping on the network to associate data sent over the network with your phone.

[1] The name "3GPP" is confusing since it contains "3G". While they didn't exist when GSM (a 2G technology) was originally being developed, they did later absorb some of the organizations that were responsible for developing GSM. It is still one of the main organizations that develops and maintains existing and future protocols.

- IMEI (International Mobile Equipment Identity): the unique identifier linked to your physical mobile device.

- K_i: a secret cryptographic key also stored on the SIM card used to authenticate your phone to the network (and prove you are who you say you are).

- MCC (Mobile Country Code): your mobile country code, but not to be confused with a country's mobile telephone prefix. For example, Canada's MCC is 302, but its telephone prefix is +001.

- MNC (Mobile Network Code): the code that represents which carrier you're using. For example, 410 is one of AT&T's MNCs.

- Cell ID: each cell tower is responsible for serving a small geographic area called a cell, which has a cell ID attached it.

- LAC/TAC ("Location Area Code"): in GSM, groups of nearby cells are organized by ID into "Location Areas" ("LA" for short), with each LA's identifier being referred to as a "Location Area Code". In 4G these are respectively referred to as Tracking Area (TA) and Tracking Area Code (TAC).

- BTS ("base station"): a more general term for devices like cell towers (and CSSs pretending to be cell towers).[2]

It's important to note that some of this terminology varies by network generation. For example, in LTE a base station is referred to as an eNodeB, and in 3G/UMTS the LAC and Cell ID are replaced by PSC (primary scrambling code) and CPI (Cell Parameter ID). For simplicity, we will be sticking to the above terminology.

Section 3: Overview of attacks

To be clear, as far as we know no one (outside of government or surveillance tech vendors) has ever gotten their hands on a commercial CSS (e.g. a Harris Corp Stingray) and published publicly available details of its inner workings, so this information all comes from academic literature and the work of open source hackers attempting to reproduce how commercial CSSs might work.

There are three main categories of attacks that will be covered:

1. Communication interception

[2] Unfortunately, most phones usually don't have an ability to specify connection settings. Recently some phones have begun to implement features like "use LTE only" though.

2. Denial of service and service downgrading

3. Location tracking

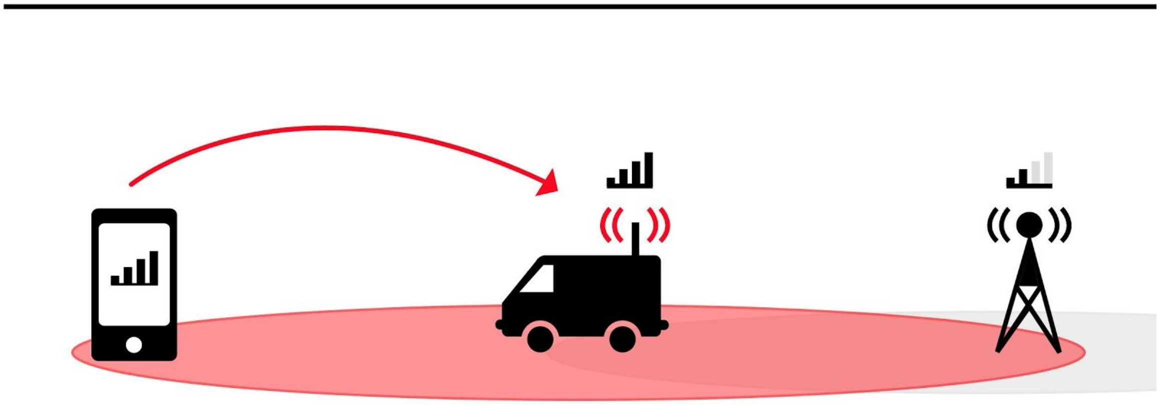
Practical implementation details are left out of the following explanations for the sake of brevity.

Section 3.1: Basic IMSI-catcher

Classic "IMSI-catchers" simply record nearby IMSIs, and then don't interact with their target phones in a significant way beyond that. They quite literally "catch" (i.e. record) IMSIs by pretending to be real base stations and then release the target phones (Paget, 2010). Let's go over how they work in more detail.

In GSM networks, phones will try to connect to whatever base station is broadcasting at the highest signal strength.

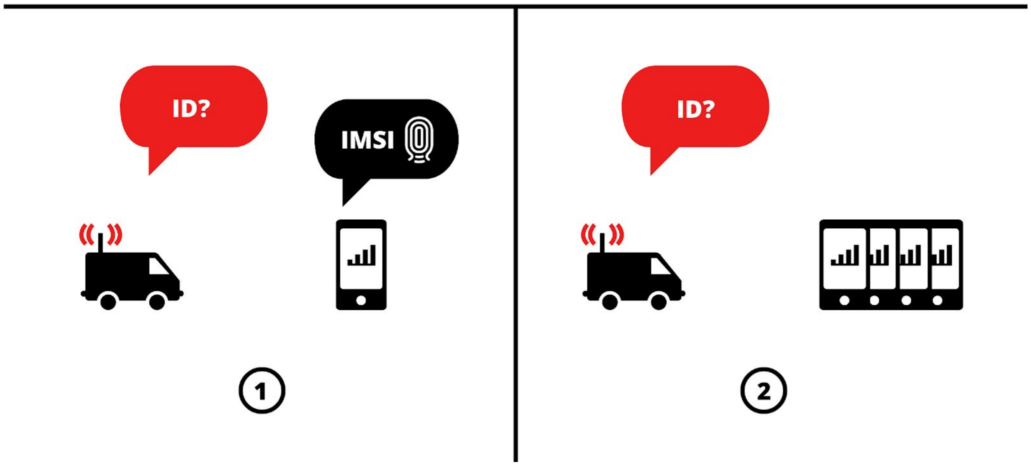
BASIC CSS: PHONE CONNECTS TO THE STRONGEST SIGNAL STRENGTH



Once a phone has identified a base station as having the best signal strength, it can begin negotiating a connection to it. The base station first asks the phone to send its encryption capabilities to it. If the base station is a CSS rather than a cell tower, it can then either ignore the response or set it to have no encryption.[3]

After this, the base station sends an Identity Request, which the phone responds to with its IMSI. The phone does this because the IMSI is stored on your SIM card, which was issued by your mobile carrier, and the phone network needs to identify that you are in fact a paying customer associated with a mobile carrier. After receiving your IMSI, the CSS then releases your phone back to the real network and moves on to try and capture another phone's IMSI. That's all it takes to collect an IMSI from a nearby phone!

BASIC CSS SENDS IDENTITY REQUEST, COLLECTS IMSI, PROCEEDS TO NEXT PHONE



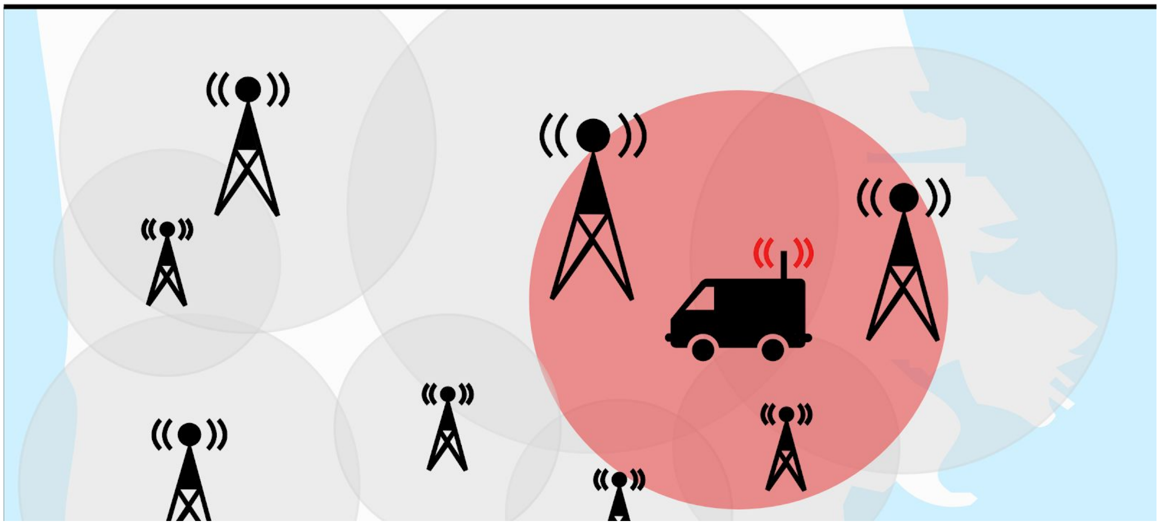
Caption: the CSS sends an Identity Request to collect the target mobile phone's IMSI. Afterwards, it proceeds to repeat this same action with other phones.

If law enforcement is operating such a CSS in a geographic area, once they've obtained the relevant IMSIs, they can then use legal process to get more data on all the users who were present.[4]

[3] Unfortunately, most phones usually don't have an ability to specify connection settings. Recently some phones have begun to implement features like "use LTE only" though.

[4] According to the Department of Justice, some CSSs can directly collect a subscriber's phone number, meaning LE can skip the step of subpoenaing a service provider to obtain the subscriber's phone number. See page 6 of https://www.eff.org/files/2015/11/30/illinois.dist__.ct__.stingrays.pdf.

A CSS IN A GEOGRAPHIC AREA



From here, many more sophisticated attacks can be launched, but that’s how the most basic kind of IMSI-catchers work: they simply collect IMSIs during the connection procedure, then abort the connection procedure and move on to their next target.

In later protocols (e.g. 4G/LTE), phones are a bit smarter about not connecting to any random base station with high signal strength, so an attacker needs more involved techniques to convince a phone to connect to their CSS. See section 3.3 for details.

Section 3.2: Communication interception

As far as we know, communication interception between a mobile phone and a legitimate cell tower is **only possible in GSM** (as opposed to later 3G or 4G protocols). There are two reasons for this:

1. Communicating over GSM doesn’t always require encryption.

2. Even when encryption is enabled, several of the cryptographic algorithms used in GSM can be broken (and in real time).

Imagine that the CSS is trying to launch an active attack where it intercepts a phone’s communications. The CSS must be able to situate itself between the phone and the tower to be able to do so, which is what’s usually referred to as a “machine in the middle” (MitM) attack.

There are two main steps to completing the MitM:

1. Spoofing authentication: the CSS needs to convince the network that it's actually the targeted mobile phone. (Section 3.2.1)

2. Deal with any encryption the network tries to set (i.e. disable it or try to break it). (Section 3.2.2)

Section 3.2.1: Spoofing authentication

Picking up from Section 3.1 where the CSS has already obtained a phone's IMSI via an Identity Request:

1. The CSS reaches out to a legitimate cell tower with a Location Update Request. This type of request is used to update the cell network about a phone's location (specifically, its LAC), which the phone needs to do periodically in order for the network to be able to route calls and messages to it quickly.

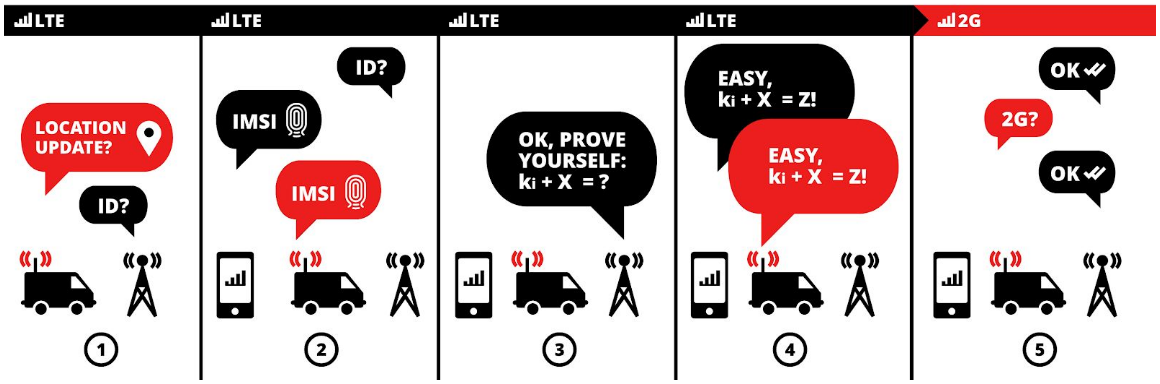
2. In response to the Location Update Request, the cell network asks the CSS to identify itself using an Identity Request. The CSS responds using the stolen IMSI.

3. At this point the tower responds with a cryptographic challenge that requires the secret key K_i (stored on the SIM card) to solve. Since the CSS doesn't have access to K_i, it passes it onto the phone to solve. The phone solves the challenge, passes it to the CSS, who then passes it back to the network.

4. After this, the network accepts the connection between it and the CSS as being authenticated.

Reminder: this is only applicable to 2G.

SPOOFING AUTHENTICATION: CSS TRICKING THE NETWORK



Caption: an illustration of steps 1-4 from above on how the CSS is able to complete the authentication MitM.

Section 3.2.2: Dealing with encryption

There are several encryption algorithms used in GSM, and at a high level, they have names like: A5/1, A5/2, etc ... with A5/0 being used to indicate that no encryption is being used.

If the network tries to specify that it wants to communicate using encryption, the CSS can just respond by saying it doesn't have encryption capabilities and defaults to A5/0. The CSS has now completed the MitM attack and can read the plaintext messages being sent between the phone and the real network.

Alternatively, if the network decides to use the A5/1 algorithm to communicate, this type of encryption can be broken in real time. The details of this attack are beyond the scope of this post, but you can read about it in the Barkan et al 2006 paper. Additionally, the A5/2 algorithm is so weak that its use has been banned since 2006. While there are known attacks against A5/3, there are no known real-time attacks.

Section 3.2.3: Why aren't users alerted that encryption is off?

At this point, many people ask: why doesn't their phone tell them something's up? According to the GSM specifications, cell phone users are supposed to be notified when encryption is disabled, and in some markets they used to be. However, this caused a lot of confusion because:

1. People would travel with their phones to places where cell towers were configured very differently (e.g. in some countries cell network encryption is banned) and it would cause a "Warning: encryption disabled" pop-up to come up a lot.

2. Cell towers everywhere were misconfigured, also causing this pop-up to appear a lot.

These issues led to many confused consumers and support calls to mobile carriers, resulting in the warning ultimately being disabled.

Section 3.2.4: Service downgrading

Even though, as far as we know, communication interception is only possible in GSM, it's trivial to downgrade a target cell phone's connection from 3G or 4G to GSM (see Section 3.5 for more information). This is because in general the base station gets to pick whatever configuration settings it wants, which includes the ability to request a protocol downgrade. Alternatively, someone could jam the 3G or 4G bands by pumping lots of white noise into them, making it too noisy to establish a connection, and phones will downgrade in search of a usable signal. LTE service downgrading is covered in detail at the end of Section 3.5.

Section 3.3: LTE CSS connection techniques

It's also important to understand how it's possible for a CSS to get around the safeguards in LTE and other modern protocols that are meant to stop phones from connecting to any base station with a high enough power.

In GSM, phones are always scanning looking for a tower with a higher signal strength to connect to. However, in LTE if the signal strength is above a certain sufficient threshold, the phone will not scan for other towers to connect to in order to save power.

Additionally, in LTE phones keep track of a "nearest neighbors" list that is broadcast from the tower that they are connected to. If for any reason they lose the connection with the tower they're connected to (or the ability to connect to it), they'll try to connect to ones that were advertised in the nearest neighbors list first, before doing a full scan of the available LTE bands for other eligible cell towers.

So, how can an attacker force a phone using LTE into connecting to their CSS? One technique would be to masquerade as a tower in the nearest neighbor's list (e.g. same frequency, same cell id, etc ...) and transmit at a higher power, so the phone will eventually switch over.

But there is a faster technique! It relies on the fact that LTE frequencies are assigned various priorities (this is referred to as "absolute priority based cell reselection"), and if a phone sees that there is a base station operating on a higher priority frequency than the one it's on, it must switch to it, regardless of its signal strength. To discover the higher priority frequencies used in a given area, all that's required is to extract them from the unencrypted configuration messages from base stations, which anyone can monitor (Shaik et al, 2017).

Using these techniques, attackers can probably force even an LTE phone to connect to their CSS, which reveals the phone's IMSI and allows followup attacks.

Section 3.4: Location tracking attacks

Often when the dangers of CSSs are being discussed, the focus is on their communication interception ability. However, in practice the consequences of real time location tracking are often much more severe. The potential for location tracking by your cell provider is unavoidable, so the specific threat model being used here is a 3rd party (such as a law enforcement agency) trying to get your location without cooperation from your cell provider.

There are generally two types of location tracking that CSSs are capable of:

1. **Presence testing:** check if a phone is present in or absent from a geographic area (where geographic area usually means a "Location Area" from before, i.e. a group of cells)

2. **Fine-grained location:** figure out the exact or rough GPS coordinates of a phone either through trilateration or by getting the phone to tell the attacker its exact GPS coordinates

Section 3.4.1: Presence Testing in LTE

Passive Presence Testing

The simplest way to do presence testing in LTE doesn't actually require someone to have what we usually consider a CSS (e.g. a device that pretends to be a legitimate cell tower). Instead, all that's required is simple radio equipment to scan the LTE frequencies, e.g. an antenna, an SDR (Software Defined Radio), and a laptop. Passive presence testing gets its name because the attacker doesn't actually need to do anything other than scan for readily available signals (Shaik et al, 2017).

A fundamental aspect of wireless technology is the paging model. When the network has a message it wants to route to a phone, it sends an "RRC paging message" which is received by every phone listening to their carrier's paging frequency in that area (which is basically every phone),[5] asking for that particular phone to contact the base station to negotiate completing a connection to receive a call or message. Thus, phones are constantly listening for RRC paging messages and receiving and discarding ones not addressed to them.

[5] Generally, the network will first direct the message to the last known cell tower the phone was connected to, and that tower will send out a paging message to everyone listening on its paging frequency. If it doesn't get a response, then it will spread out and try all the towers in a given Location Area, and so on. The exact details of how this works varies by type of data being routed (e.g. SMS vs phone call vs LTE data message) and by carrier.

RRC is short for Radio Resource Control, which is the protocol used to communicate between a cell phone and a base station. The RRC takes care of connection establishment and paging notifications that you're getting a message or phone call, among other things.

The exact way paging works varies based on several factors, including the type of message the network is trying to route to you. For example, say the network is trying to route a phone call to you. Phone calls are considered high priority (since there's someone on the other side waiting for you to connect), so the network notifies every cell tower in the last Location Area your phone was in to send out the RRC paging message addressed to your phone (as opposed to only the last cell tower the phone was using). More on this later!

RRC paging messages are usually addressed to a TMSI, but sometimes IMSI and IMEI are also used. By monitoring these unencrypted paging channels, anyone can record the IMSIs and TMSIs the network believes is in a given area. In the next section, we'll see how an attacker can correlate a TMSI to a specific target phone, as right now collecting TMSIs simply means recording pseudonyms.

Additionally, phones periodically transmit unencrypted messages about their location and measurements of cell service quality that anyone with the right equipment can easily intercept. Sometimes these messages contain the phone's exact GPS location, but usually the information about the signal strength of nearby cells is enough to calculate the phone's location. We'll look at these measurement reports in detail in the Exact GPS Coordinates section below.

Semi-Passive Presence Testing

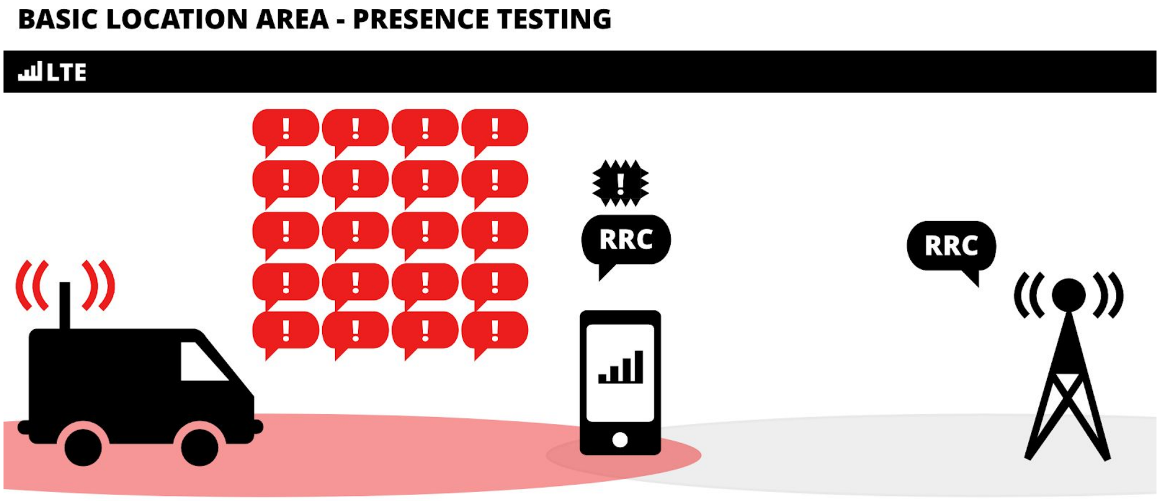
Semi-passive means that the attacker only uses network functions in ways in which they are meant to be used. An example of what it means for an adversary to be "semi-passive": the attacker can text the person they're trying to track (assuming they know their phone number) in order to generate a paging message being sent to their phone, but they can't go and send malicious or malformed data to phones or towers in the area (Shaik et al, 2017).

In this section, we are going to cover two location attacks: one which checks for a phone in a given Location Area ("Basic Location Area Test"), and one which checks for a phone connected to a specific cell tower (the "Smart Paging Test" method, which has a much smaller radius of use).

Basic Location Area Test

The first step of a basic Location Area test is to trigger about 10-20 notifications to the target's phone via phone calls while also monitoring the RRC paging messages that are sent out. To not alert the user, the attacker can almost immediately hang up after initiating the call so that the paging message makes it to the phone, but the user doesn't actually get an incoming call notification.

Because there's someone waiting on the other line to connect to you, phone calls are considered higher priority, so the network notifies every cell tower in the last Location Area the phone was in to send out the RRC paging message (as opposed to only the last cell tower the phone was using). The attacker can then use set intersection analysis (explained in [6]) with their well-timed calls to figure out the target's TMSI from the RRC messages.



Caption: CSS triggering many RRC paging requests to determine if a phone is in a given LA.

Smart Paging Test

Usually the radius of a Location Area is quite large, so from here the attacker can use something referred to as "smart paging" (explained below) to figure out the exact cell tower the target is using (which translates to knowing the user's location within a ~2 km radius) (Shaik et al, 2017).

Because general data messages (e.g. WhatsApp and FB Messenger messages) are not high priority, the network initially only broadcasts paging messages for them from the last tower the phone was known to be connected to (this is referred to as "smart paging"). Thus, once the attacker has confirmed the target's location in a TA ("Tracking Area"), they can test various cells to find the target's cell. (Note: we're switching briefly from the "Location Area" terminology to "Tracking Area" here for the sake of a concept covered below.) Similar to before, they send timed WhatsApp or FB Messenger messages

[6] Basically, you compare the paging identities in the RRC messages sent out after each short call you initiate, and extract the value(s) that are repeated the number of times you placed calls. You can read a much more here in the R*evealing Identities* section here: https://www-users.cs.umn.edu/~hoppernj/celluloc.pdf

and use set intersection analysis to verify the TMSIs being sent in RRC messages in that cell.[7]

Note that in order for this to work, the attacker needs to either have equipment in every cell (which is expensive), or move about through cells repeating this procedure until they get a match.

Section 3.4.2: Active location tracking and exact GPS coordinates

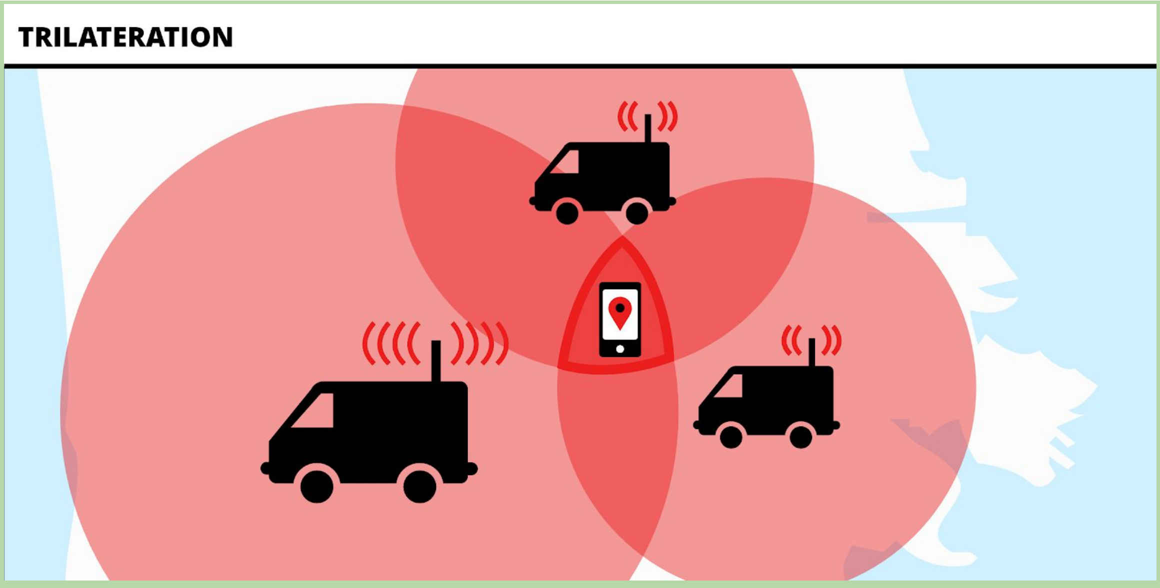
In this section, the attacker's assumed goal is to find the target's exact or rough GPS coordinates. In this section, we'll be describing active attacks, meaning ones in which the attacker can use any means available to them to figure out their target's information, including operating a CSS and sending malicious or false information to the phone or other cell towers.

In this scenario, suppose the attacker has a CSS and they've managed to lure their target into trying to connect using techniques described in Section 3.3. After completing the initial connection procedure steps, the phone enters into a CONNECTED state.

Now the attacker creates a "RRC Connection Reconfiguration" command, which contains the cell IDs of at least 3 neighbouring cell towers and their connection frequencies and sends this command to their target's phone.

Usually, the "RRC Connection Reconfiguration" command is used to modify an existing connection to a base station, but the attacker is only interested in the target phone's initial response to its message. This response contains the signal strengths of the previously specified cell towers, which can then be used to find the phone's location via trilateration:

[7] Note that Facebook messages have the advantage of not needing to know your target's phone number to be able to trigger a notification being sent to their phone! (The attacker doesn't need to be Facebook friends with their target either, as Facebook Messenger messages sent to strangers end up in the 'Other' folder, but still trigger LTE push notifications that aren't displayed to the user.)



Caption: In short, trilateration involves calculating the intersection of circles drawn around the previously specified cell towers, where the radius of each circle is a function of the reported signal strength. Note: trilateration is different than triangulation.

For newer phones and networks which support the "locationInfo-r10" feature, this report will also contain the phone's exact GPS coordinates, meaning no trilateration calculations are required. The exact GPS coordinates are just a field in the response (Shaik et al, 2017).

In addition to the technique described above, there is another way to get similar trilateration and GPS data by using RLF ("Radio Link Failure") reports, but we will not cover it in any detail as it's similar to the techniques just covered.

Section 3.5: Denial of Service and Downgrading

Cell network denial of service and protocol downgrade attacks are possible (and can have quite similar implementation details, as we'll see below). Additionally, downgrade attacks make it such that a target phone can be forced down to a less secure protocol, where more severe privacy invasive attacks can be launched.

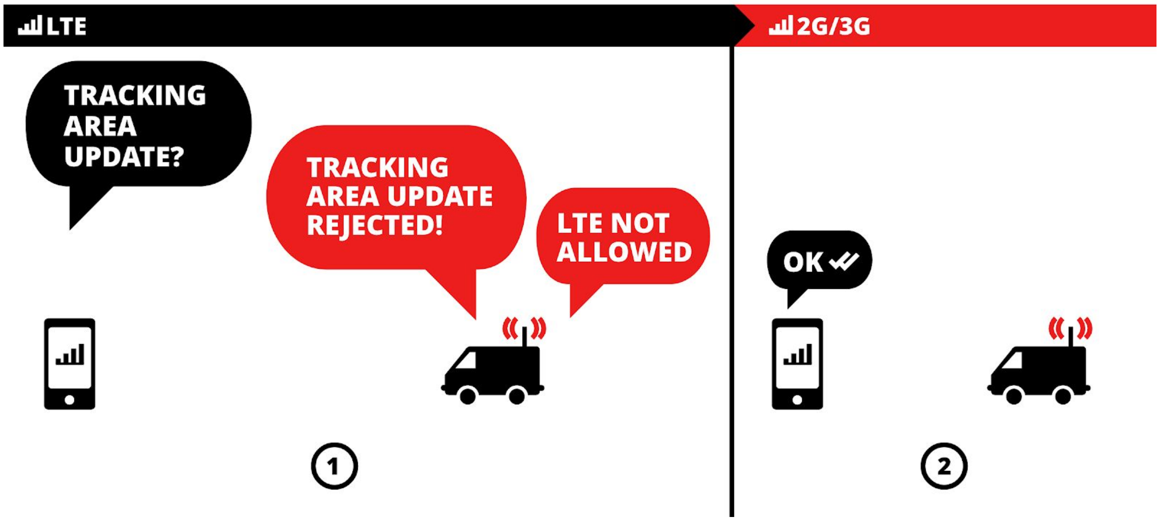
Section 3.5.1: Protocol downgrade attacks

Suppose that the attacker has set up their CSS and tricked the target into trying to connect (which was covered in Section 3.3). After the initial connection procedure, the phone will send a "Tracking Area Update Request" ("TAU" for short). This kind of message is used by the phone to keep the cell network updated about the phone's most recent location, so that the network can route calls to it faster. TAU Requests are usually sent by phones whenever they're connecting to a new base station.

The CSS responds with a "TAU Reject" message. Within the Reject message is something referred to as the "EMM cause numbers", which indicates why the message was rejected. In this case, the attacker sets it to 7 ("LTE services not allowed").

Upon receiving this EMM value, the phone deletes all information it had about the previous real network it was connected to, and then puts itself in a state where it considers its SIM card to be invalid for LTE. It then searches for 3G and GSM networks to connect to, and will not again try to negotiate an LTE connection until it is rebooted (Shaik et al, 2017).

DOWNGRADING: CSS REJECTS TRACKING AREA UPDATE REQUEST



The key reason why protocol downgrade attacks are so bad is that it renders LTE-capable phones vulnerable to attacks that normally only work on earlier protocols (e.g. the communication interception from Section 3.2).

Section 3.5.2: Denial of Service (DoS)

If the attacker is looking to launch a large scale DoS attack, the simplest thing is to jam the LTE frequencies by pumping them full of white noise. However, there are also techniques for DoS attacks that only target individual phones.

Launching a denial of service attack against an individual phone is exactly the same as the protocol downgrade attack described above, except the CSS responds with EMM cause number 8 ("LTE and non-LTE services not allowed"). The phone then puts itself in a state where it does not try to negotiate any network connections until it's been rebooted.

Additionally, there has been some research done into denying select network services (e.g. only allowing SMS, and disallowing calls and data), but for the sake of space we will not be covering this. Please see Shaik et al, 2017 below for details.

Section 4: Detection methods & apps

At this point you're probably wondering:

- Are there ways to detect CSSs?
- How to defend oneself from a CSS?
- What led to these vulnerabilities in the cell networks and what do we do about them?

These are three questions we're going to explore in this section, and unfortunately they don't have simple answers.

Section 4.1: Detection methods

To reiterate an important truth from before: a fundamental problem when researching detection methods is that **we don't know how commercial CSSs work**. Instead we rely on how we think they might work based on research findings. It's important to keep this in mind when going over some of the known detection methods below. This following list is not exhaustive, and instead is meant to be an introduction to this topic.

Unusual base station parameters or fingerprints

- There's been some speculation that commercial CSSs mask themselves as cell towers that are normally in the area, but with some configuration parameters or characteristics being subtly off (e.g. broadcast power is suddenly much higher), enough so that the "fingerprint" of the tower is different. While configuration parameters and other characteristics differ across network operators, they're usually uniform across a specific operator (Dabrowski et al, 2014).

Missing normal base station capabilities

- It's unlikely that a CSS manufacturer will have implemented the full set of capabilities of a normal base station. Missing capabilities, such as not broadcasting certain standard System Information Broadcast (SIB) messages, being unable to respond to certain standard requests, or there being very little to no paging traffic coming from the base station might be indicators of a CSS (Dabrowski et al, 2014).

Ephemerality

- It’s generally believed that CSSs don’t stay in a single place for a significant period of time, and so a base station appearing for only a short period of time could be worth investigating. However, there are also many completely normal reasons why something would only appear for a short period of time. For example, it could simply be testing equipment, or if there’s a large event happening, it could be there to help facilitate the increased traffic load.

The cell landscape is ever changing. Large scale and long term data collection is the best way to survey an area to be able to determine what’s normal versus what’s unusual. The University of Washington’s Sea Glass project is a great example of this.

You can read much more about this topic in Dabrowski et al’s IMSI-Catch Me If You Can: IMSI-Catcher-Catchers. To reiterate, while these could be indicators that something’s amiss, there are also many completely normal reasons (that have nothing to do with surveillance) as to why we’d be seeing unusual behaviour. E.g. testing equipment, temporary equipment brought in for a large event (e.g. at a sporting event), a cell tower crashed and upon restarting broadcasts temporarily incorrect values until it’s completely finished restarting, and so on.

Section 4.2: Detection apps

Many apps have been released that claim to alert users when it seems likely they’re connected to a CSS. The most popular ones include: Android IMSI-Catcher Detector (AIMSICD), SnoopSnitch, Sitch, GSM Spy Finder, Cell Spy Catcher. The quality of these apps varies, and some are still popular despite no longer being maintained.

Most of these apps implement at least some of the detection methods listed above and in Dabrowski et al. Even though sometimes multiple apps will have implemented the same detection methods, they won’t necessarily produce the same result when evaluating if a particular base station is suspicious or not (Borgaonkar et al, 2017). Let’s look at some examples of how detection apps have failed to include basic detection heuristics, as well as how there could be discrepancies in the evaluations they produce.

Varying power levels

One of the previously described detection methods is to track if a tower you’ve seen before suddenly broadcasts at much higher power. In Borgaonkar et al’s *White-Stingray: Evaluating IMSI Catchers Detection Applications*, researchers analyzed four of the previously mentioned apps and found that while most of them stored regular measurements of BTS power levels, none of them compared new values to historical values. This means that none of the apps could detect when towers had an unusually high broadcast power.

LAC change

As we saw in Section 3.2.1, when phones move to a new Location Area (or when they're in the process of connecting to a base station that's advertising as having a different LAC), they'll need to update their information. As a result, they'll eventually respond to an *Identity Request* (the command that reveals a phone's IMSI). It's generally believed that CSSs advertise as having a different LAC than the one that corresponds to the area they're in, allowing them to exploit this mechanism to force phones to hand over their IMSIs or connect to them.

All previously mentioned detection apps monitor for LAC changes. As Borgaonkar et al point out, one of them checks to see if the LAC matches that of neighbouring base stations, and displays a warning to the user when it's close to the edge of an LA. Since LAC changes are common when the user is near the edge of a LA, these warnings are often false positives. Another app stores all LACs the phone has seen before, and sends out warnings whenever a new one appears, meaning false positive warnings are constantly sent out when the user travels to new places. Another app defaults to marking anything broadcasting a LAC value between 0-9 as suspicious. This is an example of how even though all the detection apps have heuristics for detecting if a base station is suspicious based on a determination that a required value (the LAC) is unusual, their interpretations of how to do this and their implementations vary so much that they produce different results.

Because we don't have global standards for what's normal, and because things vary so wildly by country, carrier, etc, it's difficult to come up with heuristics that could universally work for detecting CSSs. As a result, the apps that have attempted to tackle this problem so far have ended up having dramatically different thresholds for alerts.

Section 4.3: Defending against CSSs

CSSs have such a wide range of capabilities (based on what we know about possible cell network attacks they could be based on) that there is no feasible way to defend against all of the things they can do. Defense should begin by considering what someone's specific threat model is and coming up with ways to defend after that.

Examples

At the time of writing, there are no publicly known confirmed examples of CSSs being used by law enforcement for communication interception or service denial. However, there are quite a few examples of CSSs being used for location tracking.

Since the main threat CSSs pose is that of real time location tracking, and there are no adjustable user settings one can change to affect this, there are currently no immediate steps one can take to defend themselves against these devices, other than either not having a cell phone, (which isn't a reasonable option for many of us) or turning off and/or leaving behind your phone when doing something important.

Despite that, there are many steps you can take to defend against online surveillance, many of which we've outlined in EFF's Surveillance Self Defense Guide.

Conclusion: the past & future of cell network security

The intersection of cell networks, security, and user privacy has historically not been an accessible field, but that's slowly changing. Each year there is more research in this field being published and open source projects (such as srsLTE) that enable this research are improving dramatically—and more people are starting to question why more work isn't being done to fix these issues.

Cell network security is broken in some pretty fundamental ways. It's up to all of us over the next few years to demand lawmakers pay closer attention to the issue, and to put pressure on standards groups, carriers, network operators, and vendors to make necessary improvements. Together, we can protect and defend user's privacy.

References

IMSI-Catch Me If You Can: IMSI-Catcher-Catchers. Adrian Dabrowski, Nicola Pianta, Thomas Klepp, Martin Mulazzani, Edgar Weippl. https://www.sba-research.org/wp-content/uploads/publications/DabrowskiEtAl-IMSI-Catcher-Catcher-ACSAC2014.pdf
(Dabrowski et al, 2014)

IMSI Catcher Detection Apps Might Not Be All That Good, Research Suggests. Joseph Cox. https://www.vice.com/en_us/article/neeb5g/stingray-detection-apps-might-not-be-all-that-good-research-suggests

Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication. Elad Barkan, Eli Biham, Nathan Keller. http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2006/CS/CS-2006-07.pdf
(Barkan et al, 2006)

Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems. Altaf Shaik, Ravishankar Borgaonkar, N. Asokan, Valtteri Niemi§and Jean-Pierre Seifert. https://arxiv.org/pdf/1510.07563.pdf
(Shaik et al, 2017)

Practical Cellphone Spying. Kristen Paget. Defcon 18. https://www.youtube.com/watch?v=fQSu9cBaojc
(Paget, 2010)

White-Stingray: Evaluating IMSI Catchers Detection Applications. Ravishankar Borgaonkar, Andrew Martin, Altaf Shaik, Jean-Pierre Seifert. https://ora.ox.ac.uk/objects/uuid:15738ed0-c144-49e9-a4fa-466362cf7754
(Borgaonkar et al, 2017)