"don't bring your phone"

I'm sure you've heard this before but it bears repeating. If you can, leave your phone at home, in a Faraday bag, or off in your car.

But we both know that somebody at every protest has a phone. The medic who might need to call 911. The legal observer. The organizer coordinating across groups. The person live-streaming because accountability requires a camera.

Phones are at protests because people need them there.

So if a phone is going to be present, the person carrying it should understand what's being done to it. And ideally, someone in the crowd should be watching.

glossary

You're going to see these terms. Here's what they mean.

CSS : Cell Site Simulator. The technically accurate umbrella term for any device that impersonates a cell tower. Also called a "fake base station."

Stingray : A brand name from L3Harris that became the generic term, like Kleenex. The actual product line includes the StingRay, StingRay II, KingFish, Hailstorm, Harpoon, and Crossbow.

IMSI Catcher : Another name for a CSS. Describes the function: it catches your IMSI.

IMSI : International Mobile Subscriber Identity. A unique 15-digit number on your SIM card that identifies your account with your carrier. This is the prize. If someone has your IMSI, they know *you* were *there*.

IMEI : International Mobile Equipment Identity. Identifies your phone *hardware*, the physical device, not the account. Stays the same even if you swap SIM cards.

SIM : Subscriber Identity Module. The card that connects your phone to your carrier. Carries the IMSI.

Dirtbox : An aircraft-mounted CSS, named after the manufacturer Digital Receiver Technology (now Boeing). Flies in Cessnas. Can surveil thousands of phones per flight. The US Marshals have operated these since at least 2007.

the problem

Your phone can be tracked. The apps, the OS, the location services, all of that sure... but even in its functionality below its use as a smartphone. The way a phone interacts with the underlying infrastructure that supports it is exploitable.

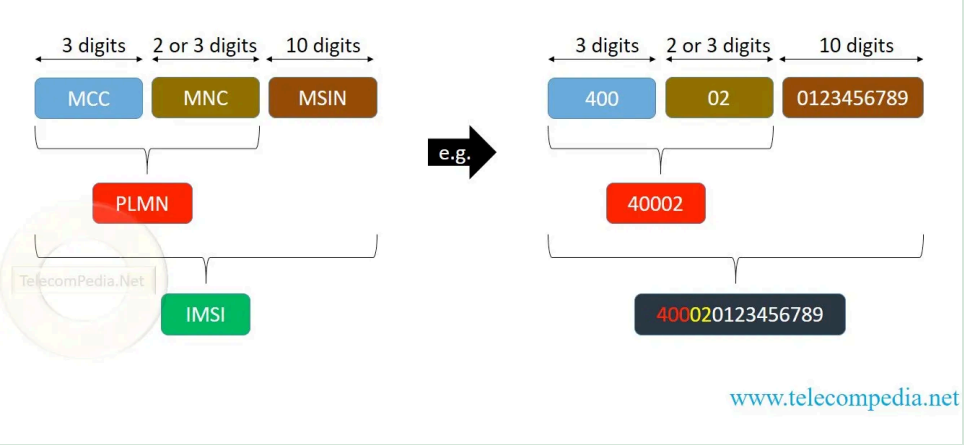
Law enforcement, lately specifically ICE, will use tools called Cell Site Simulators or CSS.

Sometimes called Stingrays, or when connected to an aircraft a "dirt box", these devices mimic a cell tower so your phone attempts to connect to it.

Before your phone makes a call or loads a web page, there's a conversation that happens first. The tower says "who are you?" and your phone answers. A legitimate tower usually already has context from previous connection. It uses a temporary ID and doesn't ask too many questions. A fake tower doesn't have that history. So it has to ask for the real thing: "what's your IMSI?"

That number can be used to identify you. For my fellow gluttons for technical suffering, here's how (not necessary to understand the rest of the doc):

This diagram denotes the IMSI number schema.



MCC (3 digits) : Mobile Country Code. Identifies the country. US is 310, 311, 312, etc.

MNC (2–3 digits) : Mobile Network Code. Identifies the carrier *within* that country. T-Mobile is 260, AT&T is 410, Verizon is 311-480. The MCC + MNC together form the **PLMN** (Public Land Mobile Network). that's the carrier's unique global ID.

MSIN (10 digits) : Mobile Subscriber Identification Number. That's *you*. Your unique account within that carrier's network.

So reading left to right: country → carrier → individual. The first 5–6 digits tell anyone who intercepts it exactly which carrier to query, and the remaining 10 digits are enough to identify the specific subscriber on that carrier's systems.

The request process: when a CSS sends an Identity Request (type = IMSI), your phone responds with all 15 digits in a single NAS message. The operator of the CSS now has your PLMN (so they know your carrier) and your MSIN (so they know your account). With a "lawful" (unfortunately it is *legal*) request, or just a database, they can map that to a name, billing address, and call records. Even without that step, the IMSI alone is a persistent tracking handle: same number every time, same device, same person, confirmed present at that location at that time.

This exchange happens *before* encryption is negotiated. The identity response travels in plaintext. That's not a bug in the protocol, it's how LTE was designed. The phone has to say who it is before the network agrees to protect the conversation.

the solution (part of it anyway)

The Electronic Frontier Foundation or EFF, otherwise known as our best friends, is a nonprofit who has been fighting for digital civil liberties since 1990. They've sued the government over surveillance, published research, built tools, and puts them in your hands. Consistently on the right side of this. **We love the EFF**.

They've developed a user friendly and very cost-accessible bit of kit to help activists and free speech advocates to better detect the use of a CSS.

Meet Rayhunter.

From the EFF:

> Rayhunter works by intercepting, storing, and analyzing the control traffic (but not user traffic, such as web requests) between the mobile hotspot Rayhunter runs on and the cell tower to which it's connected. Rayhunter analyzes the traffic in real-time and looks for suspicious events, which could include unusual requests like the base station (cell tower) trying to downgrade your connection to 2G which is vulnerable to further attacks, or the base station requesting your IMSI under suspicious circumstances.

Rayhunter checks for the following:

IMSI Requested : The big one. Flags when a tower asks for your identity (IMSI/IMEI) and then *doesn't authenticate* and tells you to disconnect. That's the classic commercial IMSI catcher pattern: grab the ID, kick you off, move to the next phone. Also flags identity requests that happen out of order or without proper follow-through.

2G Downgrade (Connection Release) : Tower drops your LTE connection and redirects you to a 2G tower. 2G is where the crypto is breakable and man-in-the-middle attacks are trivial.

2G Downgrade (SIB6/7) : Tower broadcasts fake priority lists telling your phone that nearby 2G/3G cells are preferable to LTE. Same goal as above, different mechanism, exploits the fact that these broadcast messages aren't authenticated.

Null Cipher (RRC) : Tower tells your phone to use no encryption. Should basically never happen on a real commercial network.

Null Cipher (NAS) : Same thing but at the core network signaling layer. Could indicate a CSS that's actually plugged into the carrier's infrastructure with government cooperation, or has obtained key material via SS7.

Incomplete SIB : Tower's broadcast metadata is suspiciously sparse. Real towers advertise a full chain of system information blocks; lazy IMSI catchers often only bother with the minimum. Not conclusive alone but strong when combined with other flags.

These are NOT all the methods used by cell site simulators, these are just the ones that Rayhunter checks for. Later I will shortly discuss deeper analysis but for now let's get practical.

shopping list

1. Orbic RC400L mobile hotspot : ~$10–30

eBay: search "Orbic RC400L." Factory refurbs go for $10–15 regularly. Also sold as "Kajeet RC400L" (same hardware, different branding.) Don't pay more than $30. (There was a guy on eBay selling refurbs for $10.45 free shipping last I checked.)

2. A SIM card with data service

Rayhunter needs the modem to actually connect to a real tower so it can monitor the handshake. Cheapest path: a **Tello** SIM (~$3 on Amazon), activated on their $5/month 1GB data plan. No voice needed. You just need the modem talking to a tower. However for robustness, I have historically used a mint mobile sim with data, so that I can connect a SIM-less phone to the hot spot.

Check current prices before you buy, this shifts around.

3. A USB-C cable

For connecting the Orbic to your computer during setup. You probably already have one, or it came with the device.

4. A computer

Mac, Linux, or Windows. One-time use for the install.

Total: ~$15–35. Even your broke housemate Sock can afford one.

setup

I'm not going to reproduce the EFF's install docs here because they update them and I don't want this zine to go stale. Go here:

> https://efforg.github.io/rayhunter/installing-from-release.html

The short version:

1. Download the latest release from GitHub.
2. Put SIM in Orbic. Turn it on.
3. Connect to your computer via USB or WiFi.
4. Run the install script: `./installer orbic --admin-password 'yourpassword'`
5. Wait ~3 minutes.
6. Done. Rayhunter runs automatically every time you power on the Orbic from now on.

If you've never touched a command line, that's fine. The EFF wrote this for you specifically. The script does the work. If you get stuck, the Rayhunter community Mattermost channel (linked from the GitHub repo) is helpful and not condescending about it.

using it

Turn it on. Put it in your pocket.

The light on top tells you what the devices status is:

- **Green line along the top** : running, nothing suspicious.
- **Red line** : something happened. Pay attention.

To see details: connect your phone to the Orbic's WiFi (it's a hotspot : that's what it was built for) and go to `http://192.168.1.1:8080/` in a browser. You'll see the web interface with current status and downloadable logs.

If it goes red:

Note the time and your location. Download the logs (PCAP and QMDL files) from the web interface. Report to the EFF on Signal at **ElectronicFrontierFoundation.90**. include the date, location, device model, and Rayhunter version. They are actively collecting this data.

where it falls short

It only sees what happens to the device it's on. If a stingray targets your phone but ignores the Orbic, Rayhunter sees nothing. CSS devices can be selective.

It doesn't catch everything. Rayhunter checks for a specific set of suspicious behaviors. A well-configured modern CSS might not trip any of them. The best detection tool currently available, **Marlin**, built by the University of Florida and ETH Zürich, identifies more indicators than Rayhunter currently checks for. Marlin is what SAN used at the Portland ICE facility.

You can read the paper here: https://www.cise.ufl.edu/~butler/pubs/ndss25-tucker-marlin.pdf

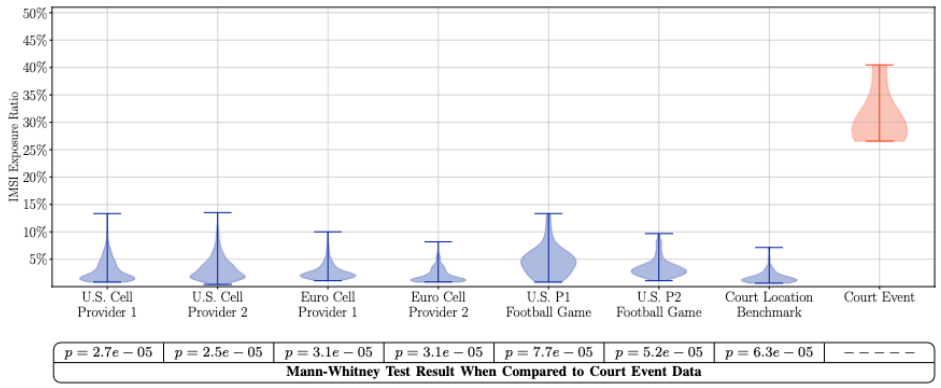
The difference is that Marlin uses a ~$1,500 software-defined radio to scan *all* nearby towers from the network side. Rayhunter runs on a $15 hotspot and only sees its own connection. So you trade accessibility for completeness.

(nerd shit for nerds): In the paper they outline a method by which they catalog ALL relevant messages a cell tower can send that would cause your mobile device to identify itself.

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.

then compares that to the baseline for how often those kinds of requests should be occurring in a given environment.



$p = 2.7e-05$	$p = 2.5e-05$	$p = 3.1e-05$	$p = 3.1e-05$	$p = 7.7e-05$	$p = 5.2e-05$	$p = 6.3e-05$	– – – – –

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.

This provides a far more robust detection metric.

Thus, Rayhunter is not a yes/no answer. It is ONE Indicator of Compromise. Red means "something unusual happened in the control plane." Context matters: where are you, what's happening around you, is there law enforcement activity nearby. Treat it like a smoke detector, not a verdict.

Note: Our very own University of Washington had their own project for detecting Cell Site Simulators.

Meet Seaglass: https://techpolicylab.uw.edu/wp-content/uploads/2018/07/SeaGlass-Enabling-City-Wide-IMSI-Catcher-Detection.pdf?utm_source=chatgpt.com

From their abstract:

> SeaGlass consists of sensors that measure and upload data on the cellular environment to find the signatures of portable cell-site simulators. SeaGlass sensors are designed to be robust, low-maintenance, and deployable in vehicles for long durations. The data they generate is used to learn a city's network properties to find anomalies consistent with cell-site simulators.

Even though the Marlin method seems to be the gold standard, I Highly recommend checking out their documentation.

why you should carry one anyway

I went to the Portland ICE protests after SAN published their findings carrying an orbic. It didn't detect anything.

I don't know what that means. Maybe the CSS was gone by the time I got there. Maybe it was there but didn't interact with my device. Maybe the reporting changed someone's calculus about deployment. I don't have a definitive answer.

But here's what I think about: the fact that someone showed up with Marlin and published what they found in a national outlet changed the situation. ICE and DHS didn't respond to SAN's requests for comment, but they didn't have to respond for the risk calculation to shift. It's harder to deploy a surveillance tool quietly when people are *looking for it*.

Somebody at the protest has a phone. Maybe it's you. If it is, you deserve to know what's happening to it. Not because the knowledge protects you (it mostly doesn't), but because understanding your threat model is the first step toward making real decisions about risk.

Every report sent to the EFF helps them map where CSS devices are being deployed, how they behave, and what detection gaps remain. Your single data point joins a dataset. That dataset informs the next version of Rayhunter, the next policy challenge, the next FOIA request.

This is relevant to WA as well

When I was at B-Sides, a security conference held last year in Redmond WA, I was at the lockpicking table and spotted one of these Orbic devices. I started a conversation with the guy and learned he runs WA Surveillance Watch. Below I'll store the QR code. To share his abstract:

> Between 2008 and 2014, the Tacoma Police Department secretly operated cell-site simulator surveillance technology—commonly known as a “StingRay”—funded by federal homeland security grants and shielded from public disclosure by a non-disclosure agreement with the FBI. The program was justified to city officials as counter-terrorism equipment but was used almost exclusively for drug investigations. When the program became public in August 2014, it triggered public records lawsuits, over $361,000 in court-ordered penalties against the City of Tacoma, and directly prompted Washington State to pass one of the nation’s strongest cell-site simulator warrant laws.
>
> This report compiles the complete public record of Tacoma’s cell-site simulator program from primary sources.

2014 is over a decade ago, but generally these deals for hardware renew, and while this info is available now, it's possible this program didn't end entirely.

resources

Rayhunter

- Install guide: `https://efforg.github.io/rayhunter/`
- GitHub: `https://github.com/EFForg/rayhunter`
- Report detections on Signal: **ElectronicFrontierFoundation.90**

The EFF

- `https://eff.org`
- Atlas of Surveillance: `https://atlasofsurveillance.org`

SAN reporting

- Portland: https://san.com/cc/exclusive-fake-cellphone-tower-likely-surveilled-protesters-at-portland-ice-facility/
- Washington: https://san.com/cc/exclusive-evidence-of-cell-phone-surveillance-detected-at-anti-ice-protest/

Further reading

- The Intercept, "What Are Stingrays and Dirtboxes?": https://theintercept.com/2020/07/31/protests-surveillance-stingrays-dirtboxes-phone-tracking/
- Marlin paper (technical): Tucker et al., NDSS 2025: https://www.cise.ufl.edu/~butler/pubs/ndss25-tucker-marlin.pdf
- SeaGlass paper: Peter Ney et al, 2017: https://techpolicylab.uw.edu/wp-content/uploads/2018/07/SeaGlass-Enabling-City-Wide-IMSI-Catcher-Detection.pdf?utm_source=chatgpt.com
- EFF Atlas of Surveillance, a tool to look up records of purchases of surveillance tech by city, county, state or agency: https://wasurveillancewatch.org/tacoma-stingray-report.html

protest basics

- Don't bring your phone if you can avoid it.
- If you must: airplane mode, WiFi off unless actively needed.
- Faraday bags work. Test yours before you rely on it.
- Use Signal. Always.