SPACEX

STARLINK WELCOMES SECURITY RESEARCHERS (BRING ON THE BUGS)

Starlink's mission is to provide high-speed, low-latency connectivity across the globe. We operate the world's largest satellite constellation, with a rapidly-growing user base in 37 countries and counting.

Different parts of this system contain different security challenges – from embedded Linux running on hundreds of thousands of computers in space and more than a million on the ground, to distributed services, phone apps, and even starlink.com. We are ultimately responsible for the security of our satellites, gateways, internet exchange points, and the Starlink kits that our customers use at home.

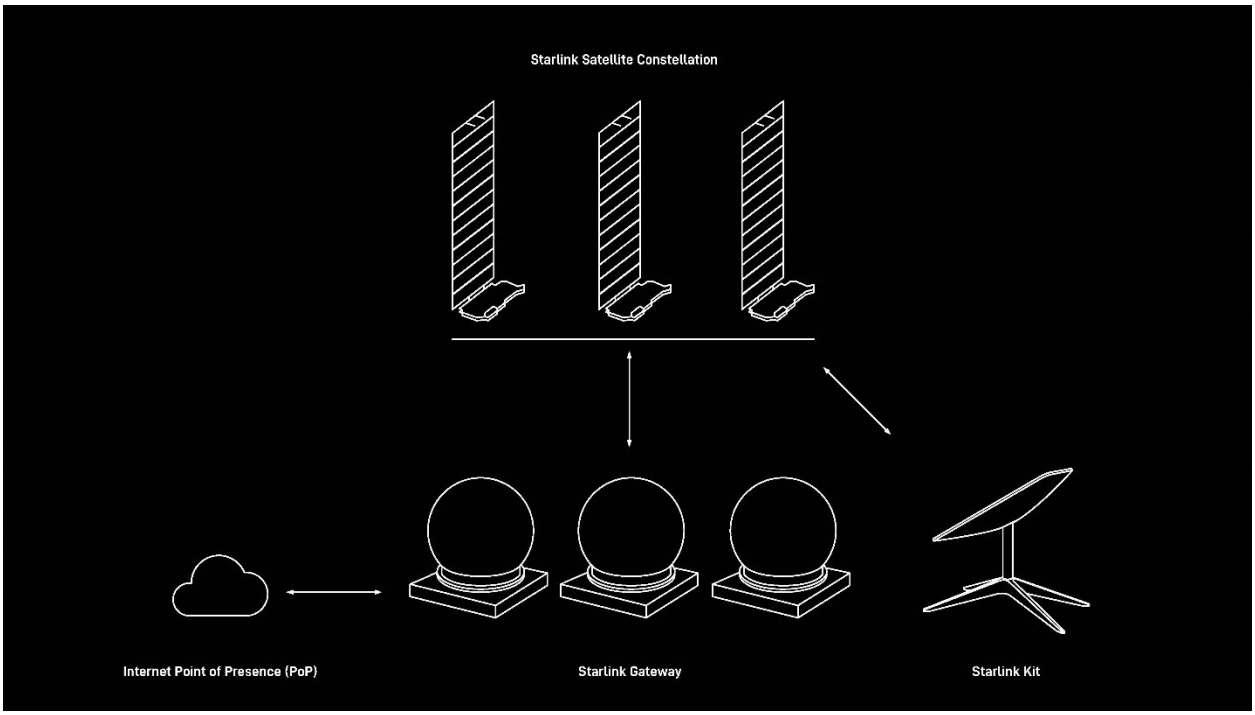
This is a huge system that has a meaningful impact on the world. If you want to help us secure it, please consider contributing as a researcher via our bug bounty program, or come join our product security team.

How We Protect Starlink

With any large and complex system, especially one that provides network connectivity on a global scale, it's essential to have multiple layers of defense to limit the overall impact of individual vulnerabilities. This defense-in-depth means that if a single layer is compromised, there are additional layers behind it that can mitigate and constrain the problem.

The Starlink kit is the user's entry point into the broader network – it converts normal ethernet traffic into the Starlink-specific radio waves that our satellites use. We are going to sell a lot of Starlink kits (that's our business!), so we have to assume some of those kits will go to people who want to attack the system. As the bridge between a user's home network and the rest of the world, the Starlink kit is both the last line of defense protecting the user from the outside world, and also the first line of defense protecting the network from malicious users.

Starlink At-A-Glance



We will happily ship a Starlink kit to any customer that purchases one. With Starlink kits all over the world, we don't have much control over what users do to them. History shows us that it's hard (and maybe impossible) to make devices completely resilient to persistent attackers with unrestricted physical access – the attacker just has too much power when they have infinite time to modify the hardware. In the limit they could always just build their own user device from scratch, though we know from experience that it's pretty hard to do so. Ultimately, the only way for us to build a secure system is to assume that attackers will eventually get into the Starlink kit, and add additional layers of defense-in-depth to protect our network and the other users within it. Other parts of the Starlink network, like satellites, might be more difficult for a consumer to get their hands on, but similarly are built with layers of defense.

To provide these additional layers of protection, there are a number of security properties that we believe are important both in the Starlink kit and in the rest of the system:

Secure Boot

First, we want our devices to only run software that we wrote. This isn't like a personal computer where the user can install apps or save files – the only software we want to run on our devices is software that we've explicitly built, tested, and signed off on. While personal

computers have to use complicated virus scanning software to guess at what software is good vs. bad, we can confidently say that if it's not from us, it's bad.

When one of our devices boots up, we cryptographically validate every layer of software before we start running it. This technique is called "secure boot", and it's designed to make it hard for an attacker to achieve a persistent compromise of a device – when the device reboots we can have confidence that it's running trusted software, even if an attacker had previously gotten in. Additionally, secure boot also protects against accidental corruption of the software – it's why you'll never see a warning that it's not safe to power down your device; we just ensure it's always safe.

Secure boot is primarily important to us on things like satellites, which are both expensive and difficult to access physically. Maintaining control of the software on our satellites is absolutely critical to the safe operation of the constellation.

The same concepts that go into secure boot on our satellites are also useful on the Starlink user terminals. Even though we know that an attacker with persistent and invasive physical access will eventually be able to defeat secure boot on their own device, the protections of secure boot are still valuable for protecting against remote attacks over the Internet (or over wifi). There is a big difference between being able to take your own device off your roof and attack it, vs. someone else being able to compromise your device without you noticing. Remote attacks, especially over the internet, can scale very quickly to affect many users, and secure boot is one important tool we use to protect our customers.

Software Updates

Secondly, we want to be able to keep all of our devices up-to-date as we add new features and make security improvements. We designed the Starlink system to be able to iterate quickly – we deploy new software roughly weekly to well more than a million Linux devices across our fleet – so it has to be extremely simple and ultra-reliable. Our Starlink user terminals are constantly updated against the latest remote exploits – as the gateway to home networks and customer Internet traffic, we think our customers deserve the best possible protection. Being able to quickly and safely deploy security fixes to the fleet is a critical piece of being able to respond to vulnerabilities.

Identity Management

Next, every device in our network must be able to uniquely identify itself in a way that we can trust. Similar to a SIM card for cell phone networks, a device's identity is a foundational piece of information that is essential for network management. It needs to have some important properties:

- Impossible for attackers to create – if an attacker can create new identities all by themselves, we can't trust that each identity is tied to a device that we manufactured
- Difficult for an attacker to copy – we don't want an attacker to be able to steal a copy of your user terminal's identity, and then do bad things with it
- Easy for us to revoke – if we notice malicious usage from a device, we can explicitly stop trusting it, restricting its ability to continue attacking the rest of the system

We use standard public key cryptography, digitally signed certificates, and hardware-based secure key storage within our devices to provide these properties.

Least Privilege

We aim to give each part of the system the minimal set of privileges required to get its job done, so that if any piece is compromised it can only impact the smallest possible set of other things. With this in mind we design network interactions between parts of the system to limit what participants can do:

- Satellites only see encrypted user traffic passing through them – an attacker who compromises a satellite can't snoop on your traffic
- Satellites have to download software updates for themselves, but they have no business ever downloading a user terminal software update package, and vice-versa
- A Starlink user terminal knows its own location precisely, but the satellite only needs to know what cell the user is in – the user terminal doesn't share its precise location information with the satellite
- A Starlink user terminal can request a change in its network path (to avoid an obstruction or to route around a network problem), but it can't change the network path of another user terminal
- One Starlink user terminal talking to our centralized services has no reason to talk to other user terminals via that control network – compromising one user terminal doesn't immediately give you a privileged position from which to attack other user terminals

We expect attackers with invasive physical access to be able to take malicious actions on behalf of a single Starlink kit using its identity, so we rely on the design principle of "least privilege" to constrain the effects in the broader system. We treat Starlink user terminals as inherently untrusted and only expose the minimal necessary information and capabilities to each specific client.

SPACEX

Starlink Bug Bounty

Our engineers are constantly trying to hack our own systems, but we're always excited to accept help! We allow responsible security researchers to do their own testing, and we provide monetary rewards when they find and report vulnerabilities. We recognize and appreciate the support of the broader security community in making Starlink better and more secure.

We encourage researchers to test Starlink for security issues in a non-disruptive way, and to report their findings through our bug bounty program. Please review that page for a full list of responsible research guidelines and in-scope targets.

When evaluating the impact of issues on our system, we consider the following factors:

- Target: Does this vulnerability just impact Starlink user terminals and routers, or does it affect shared infrastructure like satellites and our centralized services? What does compromise of the target mean in the larger Starlink system?
- Access Required: Is the vulnerability exploitable over the Internet? Is it exploitable over the local network? Is it only exploitable with physical access? Does it require other authentication or preconditions to be viable?
- Access Gained: What access does the exploit grant to the target? Can it to be used to affect other users or the system overall? Would it allow an attacker to gather information about Starlink customers?
- Scale: How hard would it be to exploit the vulnerability to affect many devices in the fleet?
- Persistence: Does this exploit allow an attacker to gain access to the target and maintain it across reboots?

Recent Black Hat/Def Con Talks

First of all, we want to congratulate Lennert Wouters on his security research into the Starlink user terminal – his findings are likely why you're reading this, and help us create the best product possible. They describe an attack where invasive physical access (taking apart the Starlink user terminal and attaching wires and additional components to it) can be used to bypass the secure boot protections within the user terminal by messing with the electrical power rails at just the right time during boot. We find the attack to be technically impressive, and is the first attack of its kind that we are aware of in our system.

We believe that our defense-in-depth approach to security limits the overall impact of this issue to our network and users:

- This vulnerability allows for persistent and arbitrary code execution on a Starlink user terminal at all privilege levels, but **only** if you have invasive physical access to the user terminal
- This vulnerability does **not** allow Starlink user terminals to be exploited remotely over their Starlink connection or over the local network
- This vulnerability does **not** directly affect Starlink satellites, since it requires physical access to the hardware
- This vulnerability does **not** allow a compromised user terminal to exploit other components of the Starlink network, such as other user terminals or satellites
- This vulnerability did **not** expose other user's information

Normal Starlink users do **not** need to be worried about this attack affecting them, or take any action in response.

If this technology sounds exciting to you, come join the team! In addition to various security roles, we have all kinds of exciting opportunities across the company on SpaceX's career page.