# POSTER: Dishing out DoS: How to Disable and Secure the Starlink User Terminal

Edd Salkield†
edd.salkield@cs.ox.ac.uk
University of Oxford
United Kingdom

Joshua Smailes†
joshua.smailes@cs.ox.ac.uk
University of Oxford
United Kingdom

Sebastian Köhler
sebastian.kohler@cs.ox.ac.uk
University of Oxford
United Kingdom

Simon Birnbach
simon.birnbach@cs.ox.ac.uk
University of Oxford
United Kingdom

Ivan Martinovic
ivan.martinovic@cs.ox.ac.uk
University of Oxford
United Kingdom

† Both authors contributed equally to this paper.

## ABSTRACT

Satellite user terminals are a promising target for adversaries seeking to target satellite communication networks. Despite this, many protections commonly found in terrestrial routers are not present in some user terminals.

As a case study we audit the attack surface presented by the Starlink router's admin interface, using fuzzing to uncover a denial of service attack on the Starlink user terminal. We explore the attack's impact, particularly in the cases of drive-by attackers, and attackers that are able to maintain a continuous presence on the network. Finally, we discuss wider implications, looking at lessons learned in terrestrial router security, and how to properly implement them in this new context.

**ACM Reference Format:**
Edd Salkield, Joshua Smailes, Sebastian Köhler, Simon Birnbach, and Ivan Martinovic. 2022. POSTER: Dishing out DoS: How to Disable and Secure the Starlink User Terminal. In *WiSec '23: 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks, May 29–June 01, 2023, Guildford, Surrey, United Kingdom.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/xxxxxxx.xxxxxxx

## 1 MOTIVATION

It is well known that router administrative interfaces present an attack surface, allowing attackers to scan for vulnerabilities and reconfigure the router through malicious requests [1, 2]. Unlike in a terrestrial setting, a satellite router is often part of a physical system including a motorized dish which can be affected. Since these networks are in remote locations and contain many untrusted users, one bad actor can deny service to many users.

Despite these issues, new routers are being implemented without the institutional memory of historic vulnerabilities and their mitigations. Alongside traditional router interface attacks, this opens up new denial of service through the physical system: rotating the dish and overusing the motor.


*WiSec '23, May 29–June 01, 2023, Guildford, Surrey, United Kingdom*

ACM ISBN 978-1-4503-XXXX-X/21/06…$15.00
https://doi.org/10.1145/xxxxxxx.xxxxxxx

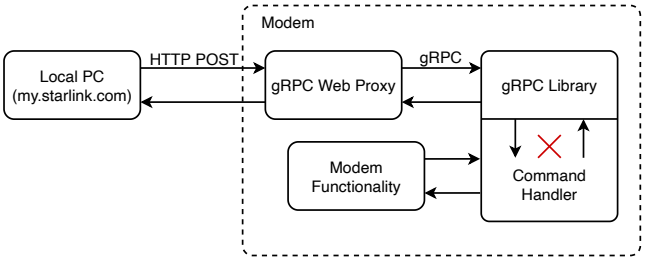



**Figure 1: Overview of the Starlink modem functionality. gRPC calls, encapsulated within HTTP POST requests, are decoded and processed. Malformed requests cause the command handler to crash, resulting in the modem no longer being able to respond to commands.**

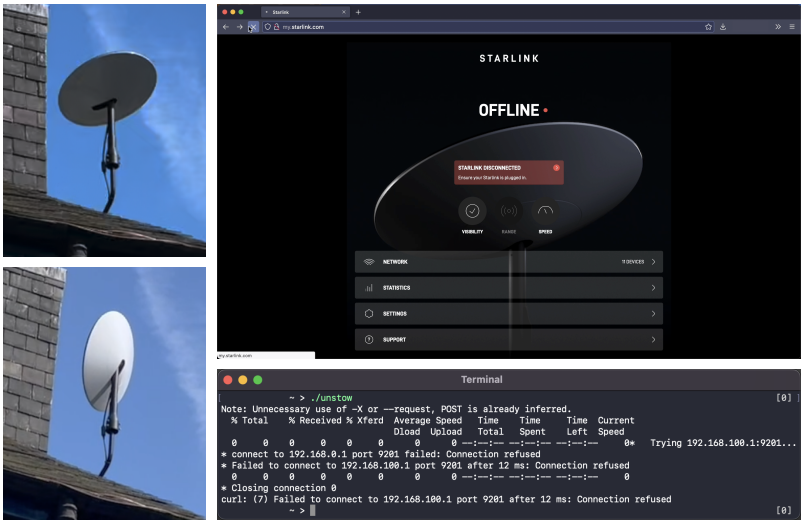
In this poster we summarize the key findings of most recent work which assesses the security of the Starlink user terminal [3]. We pay particular attention to the attack surface exposed by its web admin interface. We explore both how configuration requests are made and the effects of sending undocumented commands, using a fuzzer to iterate through the unauthenticated command space. We find an exploit in the command decoding and execution logic which, when combined with commands affecting the physical state of the dish, result in denial of service persisting until the router is physically power-cycled. This can be widely exploited due to poor security practices such as a lack of password authentication on the admin interface, and default passwords on the WiFi network itself.

After disclosure, Starlink mitigated this issue in December 2022 in patch *8c03f1b9-de75-404b-87fd-7986892cdacb.uterm.release*.

## 2 ATTACK

The Starlink user terminal is typically administered via the "http://my.starlink.com" web interface, which sends gRPC commands to the modem over the local network. As shown in Figure 1, these requests are decoded by a gRPC web proxy and forwarded to a command handler.

The vast majority of gRPC commands require no authentication, including commands affecting the physical state of the dish. We




**(a) The dish in "active" and "stowed" modes.** **(b) The web control panel error screen following the attack, and the result of sending commands to an inoperative dish[1].**

**Figure 2: The outcome of a successful attack on the Starlink dish, and the resulting web control panel and response to commands.**

note, in particular, that the "dish stow" command, which stops communication with the satellites, requires no authentication.

Since the gRPC payload is usually between 2 and 5 bytes, the command space can be fuzzed with random contents to discover unexpected behavior in the command handler[1]. Through this we discovered invalid command `00 00 00 00 03 EA 3E 00` which causes the command handler of the user terminal to crash and no longer respond to commands. This allows the attacker to lock the physical dish into a stowed state as seen in Figure 2, persistently denying service even after the attacker is no longer present on the network. Restoring internet access requires a physical power-cycle.

## 3 ATTACK VECTORS

This attack requires that a device is connected to the local network for just a few seconds to send HTTP POST requests to the modem. This can be accomplished through a new malicious device on the network, or exploiting an existing device through a browser "drive-by" attack. Whilst the Cross-Origin Resource Sharing (CORS) policies of modern browsers prevent unauthorized requests to external domains, legacy browsers are vulnerable [4]. Although the lack of encryption to "http://my.starlink.com" increases convenience in not configuring local certificates, local attackers can hijack the DNS requests or respond with a malicious website to make the request instead. The attacker could alternatively trick a user into executing a malicious executable or script to make the requests.

## 4 IMPACT AND MITIGATIONS

Since this attack can cause outages on the order of minutes or hours from just a few requests, large networks containing many untrusted users are at the greatest risk. Examples may include maritime and aviation traffic, internet cafés, or large organizations. Large networks are more vulnerable to the "drive-by" attack by users running outdated browsers. Since the Starlink routers do not password protect the network by default, malicious devices joining the network is also a serious concern.

Securing these physical systems will involve adopting measures such as password protection, local TLS certificates, and HTTP Strict Transport Security. A dedicated administrative wireless network would prevent drive-by attacks, since no device will be connected to both the administrative interface and the internet.

## 5 CONCLUSION

We have explored the security challenges faced by the Starlink router in light of existing work on router security, and uncovered a denial of service vulnerability that abuses the physical properties of the attached dish. The simplicity of the attack has highlighted a lack of due consideration to the threat posed by users inside the network, which could be mitigated through proactive security testing such as fuzzing. Required technical improvements include the use of administrative passwords, local TLS certificaties, and restricted network interfaces. We believe that, through these mitigations, it is possible to have a polished satellite internet user experience without sacrificing security.

## ACKNOWLEDGEMENT

We are grateful for hardware access through armasuisse S+T.

## REFERENCES

[1] Marcus Niemietz and Jörg Schwenk. 2015. Owning your home network: router security revisited. *arXiv preprint arXiv:1506.04112.*

[2] Philipp Jeitner, Haya Shulman, Lucas Teichmann, and Michael Waidner. 2022. XDRI Attacks – and – How to Enhance Resilience of Residential Routers. In *31st USENIX Security Symposium (USENIX Security 22)*, 4473–4490.

[3] Joshua Smailes, Edd Salkield, Simon Birnbach, Martin Strohmeier, and Ivan Martinovic. 2023. Dishing out dos: how to disable and secure the starlink user terminal. *arXiv preprint arXiv:2303.00582.*

[4] Web Hypertext Application Technology Working Group. 2023. Fetch Living Standard – CORS Protocol. Retrieved Jan. 3, 2023 from https://fetch.spec.whatwg.org/#http-cors-protocol.

[1] Source code available at *https://github.com/ssloxford/dishing-out-dos*