CVE-2026-0993: Pre-Auth Denial of Service in Sodola SL902-SWTGW124AS Firmware

AUTHOR
Emre Bastuz
Experiments, practical guides, reverse engineering, and security research.

The Sodola SL902-SWTGW124AS managed switch crashes and drops all traffic forwarding for roughly 11–13 seconds when sent a single unauthenticated HTTP POST with an oversized `boundary=` value in the `Content-Type` header. No credentials, no session, no user interaction — one request. The bug can be triggered from at least three endpoints (`/fwug.cgi` , `/login.cgi` , `/login_ft.cgi`). It's fixed in firmware V200.2.12; as of this writing that build has not appeared on the vendor's public download page. CVE-2026-0993 (CVSS 3.1: 7.5 High / CVSS 4.0: 8.7 High) has been assigned by VulnCheck.

Background

According to the sources referenced below (a community-maintained OEM cross-reference gist, corroborated by the vendor lists in the terraform-provider-hrui project — see References), the original manufacturer here is **Shenzhen Hongyavision**

Technology Co., Ltd. (brand name “HRUI”; the firmware carries the internal `hengrui` codename in its strings and source paths). Hongyavision builds a family of RTL8367N/RTL8372-based managed switches that are resold under a long list of OEM brands — Sodola, Horaco, XikeStor, AmpCom, HRUI, OptFocus, KeepLink, and others. The unit I own and tested is a **Sodola SL902-SWTGW124AS** (4×2.5G + 2×10G SFP+), running firmware **V200.2.4** at the start of this research, however the bug might also be in other OEM brands’ products.

The vulnerability

CWE-120 — Buffer Copy without Checking Size of Input.

The embedded HTTP server parses the `Content-Type: multipart/form-data` header — including its `boundary=` parameter — before it validates any session. That parsing happens on every request that declares a multipart content type, authenticated or not.

The `boundary=` value gets copied into a fixed-size internal buffer seemingly with no length check. Supply a boundary string of **1440 bytes or more** and the device crashes: the management interface stops responding, ping to the switch fails, and — as far as I could observe from the outside — **all port-to-port traffic forwarding halts**. It self-recovers after roughly **11–13 seconds**.

Nothing about this requires a login. No cookie header, no prior request, nothing. A single crafted POST from anywhere that can route a packet to the switch’s management IP is sufficient.

Scope beyond `/fwug.cgi`: I originally found and reported this against the firmware-upgrade endpoint, `/fwug.cgi`. While retesting during fix verification, I confirmed the same crash at the same ~1440-byte threshold on `/login.cgi` and `/login_ft.cgi` — both of which are, by design, reachable before any authentication happens. That tells me the vulnerable code is in HTTP request parsing shared across multiple CGI handlers, not something specific to the upgrade flow.

A firmware string, `HTTPD_MULTIPART_BOUNDARY_MAX is too small!`, sits right next to the vulnerable parsing code — it reads like a compile-time or runtime assertion the

vendor's own code emits. Which is to say: the buffer size constraint was something the original developers were aware existed as a limit, even if the code doesn't enforce it against attacker input.

Proof of concept

```
POST /fwug.cgi HTTP/1.1
Content-Type: multipart/form-data; boundary=<1440 × 'A'>
Content-Length: 0
```

That's the entire PoC — no cookie, no body. I confirmed the exact threshold using a crash-reset-then-probe methodology (each measurement preceded by a full crash/recovery cycle, to rule out cumulative state corruption skewing the result).

Send the ≥1440-byte request in a loop and the device never gets a chance to catch up — a sustained, trivially repeatable DoS against anything reachable on the management network.

CVSS

- **CVSS 3.1:** `AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H` — **7.5 (High)**
- **CVSS 4.0:** `AV:N/AC:L/AT:N/PR:N/UI:N/VC:N/VI:N/VA:H/SC:N/SI:N/SA:N` — **8.7 (High)**

Availability-only impact, network vector, no privileges or user interaction required.

Affected products

Confirmed, personally tested: Sodola SL902-SWTGW124AS, firmware V200.2.4 and V200.2.10.

Fixed: firmware V200.2.12 (see below).

I also have a lower-confidence finding worth stating carefully, because I think it's important not to blur "I tested this" with "a tool told me this." The same vulnerable HTTP parsing code, at an identical file offset, shows up in firmware images I downloaded for other switches on Hongyavision's firmware branch — Horaco, XikeStor, AmpCom, and HRUI-branded units built on the same RTL8372/8373 chipset. That determination came from AI-assisted static analysis (automated offset and string matching across binaries), not from hands-on testing. **I have not personally examined the raw binary data for those images, and I have not tested the crash on physical hardware from any vendor besides Sodola.** If you run one of these OEM-derived firmware builds, I'd treat this as "worth checking," not "confirmed vulnerable."

One scope boundary worth naming: this applies to the **HRUI/Hongyavision** firmware branch specifically. A separate group of brands (MokerLink, NICGIGA, and others) is built by a *different* manufacturer, Hasivo, on a separate codebase — I did not analyze those images, and they are **out of scope** for this finding. A few Sodola SKUs appear to be dual-sourced across both manufacturers, so the brand on the box is not by itself a reliable indicator; the firmware lineage is what matters.

Brand	Known model(s)
Horaco	HPOE-series, 2.5GbE series
XikeStor	SKS-series
AmpCom	AMG-series
HRUI	HR-series

Fix status

Sodola's engineering team supplied two candidate builds during coordinated disclosure:

- **V200.2.10** (2026-08-14) — did **not** fix the issue. I reproduced the same crash at the same threshold.
- **V200.2.12** (2026-08-18) — **fixes it.** I tested all three affected endpoints (`/fwug.cgi` , `/login.cgi` , `/login_ft.cgi`) at boundary lengths up to 50,000

bytes — about 35x the original crash threshold — with no disruption, on two separate physical units. Legitimate multipart requests (i.e., an actual firmware upload) still work correctly, so this looks like a real bounds check rather than the server just refusing all multipart traffic outright.

As of publication, **V200.2.12 has not appeared on Sodola's public firmware download page**, which still lists V200.2.4 as current. If you own this switch, I'd recommend contacting Sodola support directly and referencing this advisory to request the fixed build until it's posted publicly.

A caveat about the original vendor, and why it matters if you don't own a Sodola. The fix I verified came entirely from Sodola — one reseller of this platform. I have no indication that the upstream manufacturer, Hongyavision (HRUI), has issued a corresponding platform-wide fix, nor that the corrected code has propagated to the other OEM brands that ship the same firmware branch. So while I can say "V200.2.12 fixes this on a Sodola unit," I can't say the same for a Horaco, XikeStor, AmpCom, HRUI, or OptFocus device — those may still be running the vulnerable parser with no fixed build available. If you own one, the mitigation below is your practical protection until the manufacturer or your specific reseller ships a corrected image.

Mitigation in the meantime: restrict management-interface access to trusted hosts (ACL or a dedicated management VLAN). Changing the default admin password does **not** help here — this bug is reachable before any authentication check runs.

Disclosure timeline

Date	Event
2026-06-17	Vulnerability confirmed on live hardware
2026-07-15	Vendor (Sodola) notified
2026-08-01	Follow-up sent — no response yet to initial notification

Date	Event
2026-08-05	Sodola acknowledged and requested reproduction detail
2026-08-14	Sodola supplied candidate fix V200.2.10 — tested, does not fix the issue
2026-08-18	Sodola supplied candidate fix V200.2.12 — tested, confirmed fixed across all three endpoints
2026-08-22	CVE-2026-0993 assigned by VulnCheck
2026-10-15	Public disclosure (90 days from vendor notification), regardless of public patch availability

Once engaged, Sodola moved fast: four days from “this build doesn’t fix it” to a build that does. My one open ask to them, still outstanding as of publication, is getting V200.2.12 onto the actual public download page rather than distributed only by email.

Credit

Discovered and reported by **Emre Bastuz**, independent security researcher. Contact: info [at] emre.de

References

- https://www.cve.org/CVERecord?id=CVE-2026-0993 (CNA VulnCheck)
- OEM manufacturer / reseller cross-reference: https://gist.github.com/nijave/311aebf28cc1fb94dcf8661c7f32c6e9
- terraform-provider-hrui (corroborating vendor list): https://github.com/brennoo/terraform-provider-hrui
- This post: https://hml.io/posts/blog-post-cve-2026-0993/