ospfd: harden TE/SR TLV iteration against malformed lengths #21002

Merged **mjstapp** merged 1 commit into `FRRouting:master` from `Jafaral:ospf-fix` on Mar 5

Conversation 7 Commits 1 Checks 19 Files changed 2

Jafaral commented on Mar 4 • edited — Member

Use 32-bit counters and per-iteration TLV size bounds checks in OSPF TE/SR TLV parsers so malformed opaque LSAs cannot wrap loop accounting and advance pointers beyond the LSA buffer.

- Change loop accumulators from 16-bit to 32-bit (uint32_t) to prevent wraparound
- Rework TLV iteration so pointer advancement is controlled in-loop
- Add per-iteration guard before advancing:
  - `tlv_size <= (len - sum)` (or `length - sum`)
- On malformed size, parser now logs and exits/breaks the loop safely instead of stepping past buffer limits

👍 1

frrbot Bot added the ospf label on Mar 4

github-actions Bot added master size/L labels on Mar 4

Jafaral force-pushed the `ospf-fix` branch from **`c633e41`** to **`3ca8edf`** last month — Compare

mjstapp self-requested a review last month

greptile-apps Bot commented on Mar 4

Greptile Summary

This PR hardens the OSPF TE/SR TLV parsers in `ospfd/ospf_sr.c` and `ospfd/ospf_te.c` against malformed opaque LSAs by (1) widening loop accumulators from `uint16_t` to `uint32_t` to prevent wraparound, and (2) adding an explicit upper-bound check (`tlv_size > remaining`) before advancing each TLV pointer. The changes are correct and the loop-termination logic is sound.

- **uint32_t promotion** — `length` / `sum` / `len` variables changed from `uint16_t` to `uint32_t` in all seven affected loops, eliminating the wraparound path where an adversary could craft a sequence of TLVs whose cumulative size wraps a 16-bit counter and causes the parser to re-enter already-consumed data.
- **Upper-bound guard** — Each loop now checks `tlv_size > length - sum` before processing and advancing, preventing out-of-bounds pointer advancement when a single malformed TLV field advertises a length larger than the remaining buffer.
- **Safe early exit** — On a malformed size, every loop now emits a `zlog_warn` and `break`s instead of advancing blindly, which is the correct hardening posture.
- **Dead lower-bound check** — Every loop also checks `tlv_size < TLV_HDR_SIZE`. Because `TLV_SIZE()` is defined as `TLV_HDR_SIZE + ROUNDUP(ntohs(tlvh->length), 4)` and `ROUNDUP` always returns >= 0, `TLV_SIZE` can never be less than `TLV_HDR_SIZE`. This half of the condition is unreachable across all 7 modified loops and creates misleading assumptions about TLV_SIZE invariants.

Confidence Score: 4/5

- This PR is safe to merge; it strictly strengthens existing parsing guards without changing protocol behavior on well-formed LSAs.
- The core changes (uint32_t widening and upper-bound TLV size checks) are mechanically straightforward and consistently applied across all seven modified loops. Loop termination is correct: the `if (sum < length)` guard before `TLV_HDR_NEXT` prevents advancing the pointer on the final iteration, and the `sum < length` outer condition ensures the loop exits properly. The identified issue is a dead `tlv_size < TLV_HDR_SIZE` check across all modified loops (unreachable since TLV_SIZE always >= TLV_HDR_SIZE), which is a clarity nit—not a functional or security concern, but worth addressing in a hardening PR for code maintainability.
- No files require special attention; both changed files follow the same consistent hardening pattern, though all occurrences of the dead lower-bound check could be simplified for clarity.

Flowchart

Last reviewed commit: `3ca8edf`

greptile-apps Bot reviewed on Mar 4

View reviewed changes

ospfd/ospf_sr.c Outdated

```
1008 +     for (; sum < length && sub_tlvh;) {
1009 +         uint32_t tlv_size = TLV_SIZE(sub_tlvh);
1010 +
1011 +         if (tlv_size < TLV_HDR_SIZE || tlv_size > length - sum) {
```

greptile-apps Bot on Mar 4

Dead lower-bound guard `tlv_size < TLV_HDR_SIZE`

`TLV_SIZE(tlvh)` expands to `(uint32_t)(TLV_HDR_SIZE + TLV_BODY_SIZE(tlvh))`, and `TLV_BODY_SIZE` is `ROUNDUP(ntohs(tlvh->length), sizeof(uint32_t))` which is always `>= 0`. Therefore `TLV_SIZE >= TLV_HDR_SIZE` (4 bytes) unconditionally — the `tlv_size < TLV_HDR_SIZE` branch can never be taken.

This same dead check appears in every modified loop across both files:

- `ospfd/ospf_sr.c:1011` — `get_ext_link_sid`
- `ospfd/ospf_sr.c:1098` — `get_ext_prefix_sid`
- `ospfd/ospf_sr.c:1429` — `ospf_sr_ri_lsa_update`
- `ospfd/ospf_te.c:2159` — `ospf_te_parse_te`
- `ospfd/ospf_te.c:2377` — `ospf_te_delete_te`
- `ospfd/ospf_te.c:2484` — `ospf_te_parse_ri`
- `ospfd/ospf_te.c:2829` — `ospf_te_parse_ext_link`

The condition is harmless today, but it implies to readers that `TLV_SIZE` can return a value smaller than the header, which is misleading. The effective guard is solely `tlv_size > length - sum`. Consider simplifying by dropping the dead half of the condition:

Suggested change

```
-                 if (tlv_size < TLV_HDR_SIZE || tlv_size > length - sum)
  {
+                 if (tlv_size > length - sum) {
```

▶ Prompt To Fix With AI

Jafaral on Mar 4 Member Author

Fixed!

mjstapp reviewed on Mar 4

View reviewed changes

mjstapp left a comment — Contributor

I think the greptile suggestion is valid: it would be better to be as clear as possible in this parsing code.

👍 1

`ospfd: harden TE/SR TLV iteration against malformed lengths` ··· ✓ `d3e8aed`

Jafaral force-pushed the `ospf-fix` branch from **`3ca8edf`** to **`d3e8aed`** last month — Compare

Jafaral commented on Mar 4 • edited — Member — Author

@**Mergifyio** backport stable/10.6 stable/10.5 stable/10.4 stable/10.3 stable/10.2 stable/10.1 stable/10.0

👍 1

mergify Bot commented on Mar 4 • edited

> backport stable/10.6 stable/10.5 stable/10.4 stable/10.3 stable/10.2 stable/10.1 stable/10.0

✅ **Backports have been created**

▶ Details

github-actions Bot added the backport label on Mar 4

mjstapp approved these changes on Mar 4

View reviewed changes

mjstapp left a comment — Contributor

Thanks, looks good

mjstapp merged commit `f098dec` into `FRRouting:master` on Mar 5

23 of 42 checks passed

View details

This was referenced on Mar 5

ospfd: harden TE/SR TLV iteration against malformed lengths (backport #21002) #21012

Merged

ospfd: harden TE/SR TLV iteration against malformed lengths (backport #21002) #21013

Merged

ospfd: harden TE/SR TLV iteration against malformed lengths (backport #21002) #21014

Merged

ospfd: harden TE/SR TLV iteration against malformed lengths (backport #21002) #21015

Merged

ospfd: harden TE/SR TLV iteration against malformed lengths (backport #21002) #21016

Merged

ospfd: harden TE/SR TLV iteration against malformed lengths (backport #21002) #21017

Merged

ospfd: harden TE/SR TLV iteration against malformed lengths (backport #21002) #21018

Merged

donaldsharp added a commit that referenced this pull request on Mar 5

`Merge pull request #21012 from FRRouting/mergify/bp/stable/10.6/pr-21002` ··· ✗ `65edc70`

donaldsharp added a commit that referenced this pull request on Mar 5

`Merge pull request #21013 from FRRouting/mergify/bp/stable/10.5/pr-21002` ··· ✓ `c06b61a`

donaldsharp added a commit that referenced this pull request on Mar 5

`Merge pull request #21014 from FRRouting/mergify/bp/stable/10.4/pr-21002` ··· ✓ `67119c4`

donaldsharp added a commit that referenced this pull request on Mar 5

`Merge pull request #21015 from FRRouting/mergify/bp/stable/10.3/pr-21002` ··· ✕ `172e945`

donaldsharp added a commit that referenced this pull request on Mar 5

`Merge pull request #21016 from FRRouting/mergify/bp/stable/10.2/pr-21002` ··· ✕ `524965c`

donaldsharp added a commit that referenced this pull request on Mar 5

`Merge pull request #21017 from FRRouting/mergify/bp/stable/10.1/pr-21002` ··· ✕ `67dbdbd`

donaldsharp added a commit that referenced this pull request on Mar 5

`Merge pull request #21018 from FRRouting/mergify/bp/stable/10.0/pr-21002` ··· ✕ `221814b`

mattiaswal mentioned this pull request on Mar 18

Bump FRR to 10.5.3 kernelkit/infix#1451

Merged

17 tasks

Reviewers

greptile-apps[bot]

mjstapp

Assignees

No one assigned

Labels

backport master ospf size/L

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

2 participants