

FRRouting / frr Public

<> Code Issues 372 Pull requests 356 Discussions Actions Projects

Commit f098dec



mjstapp authored on Mar 5 · ✖ 8 / 18 · Verified

Merge pull request #21002 from Jafaral/ospf-fix
ospfd: harden TE/SR TLV iteration against malformed lengths

master (#21002)
2 parents [3299015](#) + [d3e8aed](#) commit f098dec

2 files changed +91 -23

[↑ Top](#)

- ▼ ospfd
 - ospf_sr.c
 - ospf_te.c

```

  ▼ ospfd/ospf_sr.c
  @@ -989,7 +989,8 @@ static struct sr_link *get_ext_link_sid(struct
  tlv_header *tlvh, size_t size)
  989  989      struct ext_subtlv_rmt_itf_addr *rmt_itf;
  990  990
  991  991      struct tlv_header *sub_tlvh;
  992  992      - uint16_t length = 0, sum = 0, i = 0;
  992  992      + uint32_t length = 0, sum = 0;
  993  993      + uint16_t i = 0;
  993  994
  994  995      /* Check TLV size */

```

```

995     996         if ((ntohs(tlvh->length) > size)
@@ -1004,7 +1005,15 @@ static struct sr_link *get_ext_link_sid(struct
tlv_header *tlvh, size_t size)
1004     1005         length = ntohs(tlvh->length) - EXT_TLV_LINK_SIZE;
1005     1006         sub_tlvh = (struct tlv_header *)((char *)(tlvh) + TLV_HDR_SIZE
1006     1007                 + EXT_TLV_LINK_SIZE);
1007     - for (; sum < length && sub_tlvh; sub_tlvh = TLV_HDR_NEXT(sub_tlvh)) {
1008     + for (; sum < length && sub_tlvh;) {
1009     +     uint32_t tlv_size = TLV_SIZE(sub_tlvh);
1010     +
1011     +     if (tlv_size > length - sum) {
1012     +         zlog_warn("Malformed Extended Link sub-TLV size %u (remaining
%u)",
1013     +                 tlv_size, length - sum);
1014     +         break;
1015     +     }
1016     +
1008     1017         switch (ntohs(sub_tlvh->type)) {
1009     1018         case EXT_SUBTLV_ADJ_SID:
1010     1019             adj_sid = (struct ext_subtlv_adj_sid *)sub_tlvh;
@@ -1045,7 +1054,9 @@ static struct sr_link *get_ext_link_sid(struct
tlv_header *tlvh, size_t size)
1045     1054             default:
1046     1055                 break;
1047     1056         }
1048     - sum += TLV_SIZE(sub_tlvh);
1057     + sum += tlv_size;
1058     + if (sum < length)
1059     +     sub_tlvh = TLV_HDR_NEXT(sub_tlvh);
1049     1060     }
1050     1061
1051     1062     IPV4_ADDR_COPY(&sr1->itf_addr, &link->link_data);
@@ -1066,7 +1077,7 @@ static struct sr_prefix *get_ext_prefix_sid(struct
tlv_header *tlvh,
1066     1077     struct ext_subtlv_prefix_sid *psid;
1067     1078
1068     1079     struct tlv_header *sub_tlvh;
1069     - uint16_t length = 0, sum = 0;
1080     + uint32_t length = 0, sum = 0;
1070     1081

```

```

1071 1082      /* Check TLV size */
1072 1083      if ((ntohs(tlvh->length) > size)
@@ -1081,7 +1092,15 @@ static struct sr_prefix *get_ext_prefix_sid(struct
    tlv_header *tlvh,
1081 1092      length = ntohs(tlvh->length) - EXT_TLV_PREFIX_SIZE;
1082 1093      sub_tlvh = (struct tlv_header *)((char *)tlvh) + TLV_HDR_SIZE
1083 1094                  + EXT_TLV_PREFIX_SIZE);
1084 - for (; sum < length && sub_tlvh; sub_tlvh = TLV_HDR_NEXT(sub_tlvh)) {
1095 + for (; sum < length && sub_tlvh;) {
1096 +     uint32_t tlv_size = TLV_SIZE(sub_tlvh);
1097 +
1098 +     if (tlv_size > length - sum) {
1099 +         zlog_warn("Malformed Extended Prefix sub-TLV size %u (remaining
    %u)",
1100 +                 tlv_size, length - sum);
1101 +         break;
1102 +     }
1103 +
1085 1104      switch (ntohs(sub_tlvh->type)) {
1086 1105      case EXT_SUBTLV_PREFIX_SID:
1087 1106          psid = (struct ext_subtlv_prefix_sid *)sub_tlvh;
@@ -1106,7 +1125,9 @@ static struct sr_prefix *get_ext_prefix_sid(struct
    tlv_header *tlvh,
1106 1125      default:
1107 1126          break;
1108 1127      }
1109 - sum += TLV_SIZE(sub_tlvh);
1128 + sum += tlv_size;
1129 + if (sum < length)
1130 +     sub_tlvh = TLV_HDR_NEXT(sub_tlvh);
1110 1131      }
1111 1132
1112 1133      osr_debug("  |- Found SID %u for prefix %pFX", srp->sid,
@@ -1374,7 +1395,7 @@ void ospf_sr_ri_lsa_update(struct ospf_lsa *lsa)
1374 1395      struct ri_sr_tlv_sid_label_range *ri_srlb = NULL;
1375 1396      struct ri_sr_tlv_sr_algorithm *algo = NULL;
1376 1397      struct sr_block srgb;
1377 - uint16_t length = 0, sum = 0;
1398 + uint32_t length = 0, sum = 0;

```

```

1378 1399      uint8_t msd = 0;
1379 1400
1380 1401      ospf_debug("SR (%s): Process Router Information LSA 4.0.0.%u from %pI4",
    ↓
    ↑
@@ -1402,8 +1423,15 @@ void ospf_sr_ri_lsa_update(struct ospf_lsa *lsa)
1402 1423      srgb.range_size = 0;
1403 1424      srgb.lower_bound = 0;
1404 1425
1405 -      for (tlvh = TLV_HDR_TOP(lsa); (sum < length) && (tlvh != NULL);
1406 -          tlvh = TLV_HDR_NEXT(tlvh)) {
1426 +      for (tlvh = TLV_HDR_TOP(lsa); (sum < length) && (tlvh != NULL);) {
1427 +          uint32_t tlv_size = TLV_SIZE(tlvh);
1428 +
1429 +          if (tlv_size > length - sum) {
1430 +              zlog_warn("Malformed RI TLV size %u (remaining %u)", tlv_size,
1431 +                  length - sum);
1432 +              break;
1433 +          }
1434 +
1407 1435      switch (ntohs(tlvh->type)) {
1408 1436      case RI_SR_TLV_SR_ALGORITHM:
1409 1437          algo = (struct ri_sr_tlv_sr_algorithm *)tlvh;
    ↕
@@ -1420,7 +1448,9 @@ void ospf_sr_ri_lsa_update(struct ospf_lsa *lsa)
1420 1448      default:
1421 1449          break;
1422 1450      }
1423 -      sum += TLV_SIZE(tlvh);
1451 +      sum += tlv_size;
1452 +      if (sum < length)
1453 +          tlvh = TLV_HDR_NEXT(tlvh);
1424 1454      }
1425 1455
1426 1456      /* Check if Segment Routing Capabilities has been found */
    ↓

```

ospfd/ospf_te.c



```

    ↑
@@ -2119,7 +2119,7 @@ static int ospf_te_parse_te(struct ls_ted *ted,
struct ospf_lsa *lsa)
2119 2119      struct ls_attributes *old, attr = {};
2120 2120      struct tlv_header *tlvh;

```

```

2121 2121 void *value;
2122 - uint16_t len, sum;
2122 + uint32_t len, sum;
2123 2123 uint8_t lsa_id;
2124 2124
2125 2125 /* Initialize Attribute */
@@ -2149,11 +2149,19 @@ static int ospf_te_parse_te(struct ls_ted *ted,
struct ospf_lsa *lsa)
2149 2149
2150 2150 sum = sizeof(struct tlv_header);
2151 2151 /* Browse sub-TLV and fulfill Link State Attributes */
2152 - for (tlvh = TLV_DATA(tlvh); sum < len; tlvh = TLV_HDR_NEXT(tlvh)) {
2152 + tlvh = TLV_DATA(tlvh);
2153 + while (sum < len) {
2154 + uint32_t tlv_size = TLV_SIZE(tlvh);
2153 2155 uint32_t val32, tab32[2];
2154 2156 float valf, tabf[8];
2155 2157 struct in_addr addr;
2156 2158
2159 + if (tlv_size > len - sum) {
2160 + zlog_warn("Malformed TE sub-TLV size %u (remaining %u)",
tlv_size,
2161 + len - sum);
2162 + break;
2163 + }
2164 +
2157 2165 value = TLV_DATA(tlvh);
2158 2166 switch (ntohs(tlvh->type)) {
2159 2167 case TE_LINK_SUBTLV_LCLIF_IPADDR:
@@ -2248,7 +2256,9 @@ static int ospf_te_parse_te(struct ls_ted *ted,
struct ospf_lsa *lsa)
2248 2256 default:
2249 2257 break;
2250 2258 }
2251 - sum += TLV_SIZE(tlvh);
2259 + sum += tlv_size;
2260 + if (sum < len)
2261 + tlvh = TLV_HDR_NEXT(tlvh);
2252 2262 }
2253 2263

```

```

2254 2264      /* Get corresponding Edge from Link State Data Base */
↕
@@ -2348,7 +2358,7 @@ static int ospf_te_delete_te(struct ls_ted *ted,
↑
struct ospf_lsa *lsa)
2348 2358      struct tlv_header *tlvh;
2349 2359      struct in_addr addr;
2350 2360      struct ls_edge_key key = {.family = AF_UNSPEC};
2351 -      uint16_t len, sum;
2361 +      uint32_t len, sum;
2352 2362      uint8_t lsa_id;
2353 2363
2354 2364      /* Initialize TLV browsing */
↕
@@ -2360,14 +2370,25 @@ static int ospf_te_delete_te(struct ls_ted *ted,
struct ospf_lsa *lsa)
2360 2370      sum = sizeof(struct tlv_header);
2361 2371
2362 2372      /* Browse sub-TLV to find Link ID */
2363 -      for (tlvh = TLV_DATA(tlvh); sum < len; tlvh = TLV_HDR_NEXT(tlvh)) {
2373 +      tlvh = TLV_DATA(tlvh);
2374 +      while (sum < len) {
2375 +          uint32_t tlv_size = TLV_SIZE(tlvh);
2376 +
2377 +          if (tlv_size > len - sum) {
2378 +              zlog_warn("Malformed TE sub-TLV size %u (remaining %u)",
2379 +                  tlv_size,
2380 +                  len - sum);
2381 +              break;
2382 +          }
2364 2383      if (ntohs(tlvh->type) == TE_LINK_SUBTLV_LCLIF_IPADDR) {
2365 2384          memcpy(&addr, TLV_DATA(tlvh), TE_LINK_SUBTLV_DEF_SIZE);
2366 2385          key.family = AF_INET;
2367 2386          IPV4_ADDR_COPY(&key.k.addr, &addr);
2368 2387          break;
2369 2388      }
2370 -      sum += TLV_SIZE(tlvh);
2389 +      sum += tlv_size;
2390 +      if (sum < len)
2391 +          tlvh = TLV_HDR_NEXT(tlvh);
2371 2392      }
2372 2393      if (key.family == AF_UNSPEC)

```

2373	2394	<code>return 0;</code>
↕		@@ -2442,7 +2463,7 @@ static int ospf_te_parse_ri(struct ls_ted *ted, struct ospf_lsa *lsa)
2442	2463	<code>struct ls_node *node;</code>
2443	2464	<code>struct lsa_header *lsah = lsa->data;</code>
2444	2465	<code>struct tlv_header *tlvh;</code>
2445	-	<code>uint16_t len = 0, sum = 0;</code>
	2466	<code>uint32_t len = 0, sum = 0;</code>
2446	2467	
2447	2468	<code>/* Get vertex / Node from LSA Advertised Router ID */</code>
2448	2469	<code>vertex = get_vertex(ted, lsa);</code>
↕		@@ -2453,13 +2474,18 @@ static int ospf_te_parse_ri(struct ls_ted *ted, struct ospf_lsa *lsa)
2453	2474	
2454	2475	<code>/* Initialize TLV browsing */</code>
2455	2476	<code>len = lsa->size - OSPF_LSA_HEADER_SIZE;</code>
2456	-	<code>for (tlvh = TLV_HDR_TOP(lsah); sum < len && tlvh;</code>
2457	-	<code>tlvh = TLV_HDR_NEXT(tlvh)) {</code>
	2477	<code>for (tlvh = TLV_HDR_TOP(lsah); sum < len && tlvh;) {</code>
	2478	<code>uint32_t tlv_size = TLV_SIZE(tlvh);</code>
2458	2479	<code>struct ri_sr_tlv_sr_algorithm *algo;</code>
2459	2480	<code>struct ri_sr_tlv_sid_label_range *range;</code>
2460	2481	<code>struct ri_sr_tlv_node_msd *msd;</code>
2461	2482	<code>uint32_t size, lower;</code>
2462	2483	
	2484	<code>+ if (tlv_size > len - sum) {</code>
	2485	<code>+ zlog_warn("Malformed RI TLV size %u (remaining %u)", tlv_size,</code>
		<code>len - sum);</code>
	2486	<code>+ break;</code>
	2487	<code>+ }</code>
	2488	<code>+ }</code>
2463	2489	<code>switch (ntohs(tlvh->type)) {</code>
2464	2490	<code>case RI_SR_TLV_SR_ALGORITHM:</code>
2465	2491	<code>if (TLV_BODY_SIZE(tlvh) < 1 </code>
↕		@@ -2544,7 +2570,9 @@ static int ospf_te_parse_ri(struct ls_ted *ted, struct ospf_lsa *lsa)
2544	2570	<code>default:</code>
2545	2571	<code>break;</code>
2546	2572	<code>}</code>
2547	-	<code>sum += TLV_SIZE(tlvh);</code>

```

2573 +         sum += tlv_size;
2574 +         if (sum < len)
2575 +             tlvh = TLV_HDR_NEXT(tlvh);
2548 2576     }
2549 2577
2550 2578     /* Vertex has been created or updated: export it */
@@ -2756,7 +2784,8 @@ static int ospf_te_parse_ext_link(struct ls_ted
*ted, struct ospf_lsa *lsa)
2756 2784     struct ext_tlv_link *ext;
2757 2785     struct ls_edge *edge;
2758 2786     struct ls_attributes *atr;
2759 -     uint16_t len = 0, sum = 0, i;
2787 +     uint32_t len = 0, sum = 0;
2788 +     uint16_t i;
2760 2789     uint32_t label;
2761 2790
2762 2791     /* Get corresponding Edge from Link State Data Base */
@@ -2790,11 +2819,18 @@ static int ospf_te_parse_ext_link(struct ls_ted
*ted, struct ospf_lsa *lsa)
2790 2819     len -= EXT_TLV_LINK_SIZE;
2791 2820     tlvh = (struct tlv_header *)((char *) (ext) + TLV_HDR_SIZE
2792 2821         + EXT_TLV_LINK_SIZE);
2793 -     for (; sum < len; tlvh = TLV_HDR_NEXT(tlvh)) {
2822 +     for (; sum < len;) {
2823 +         uint32_t tlv_size = TLV_SIZE(tlvh);
2794 2824         struct ext_subtlv_adj_sid *adj;
2795 2825         struct ext_subtlv_lan_adj_sid *ladj;
2796 2826         struct ext_subtlv_rmt_itf_addr *rmt;
2797 2827
2828 +         if (tlv_size > len - sum) {
2829 +             zlog_warn("Malformed Extended Link sub-TLV size %u (remaining
2830 +                 %u)",
2831 +                 tlv_size, len - sum);
2831 +             break;
2832 +         }
2833 +
2798 2834         switch (ntohs(tlvh->type)) {
2799 2835         case EXT_SUBTLV_ADJ_SID:
2800 2836             if (TLV_BODY_SIZE(tlvh) != EXT_SUBTLV_ADJ_SID_SIZE)

```

		⌵	@@ -2873,7 +2909,9 @@ static int ospf_te_parse_ext_link(struct ls_ted
		⌶	*ted, struct ospf_lsa *lsa)
2873	2909		default:
2874	2910		break;
2875	2911		}
2876	-		sum += TLV_SIZE(tlvh);
	2912	+	sum += tlv_size;
	2913	+	if (sum < len)
	2914	+	tlvh = TLV_HDR_NEXT(tlvh);
2877	2915		}
2878	2916		
2879	2917		/* Export Link State Edge if needed */
		⌵	

Comments 0



Please [sign in](#) to comment.