cesanta / **mongoose** Public

Code Issues 4 Pull requests 5 Discussions Actions Projects

Commit `0d882f1`

scaprile committed 2 days ago

```
patch
```

`master` · 7.21

1 parent 1bb8579 commit 0d882f1

13 files changed +528 -252 lines changed

Top

Filter files…

- mongoose.c
- mongoose.h
- src
  - dns.c
  - http.c
  - json.c
  - net_builtin.c
  - str.c
  - tls_aes128.c
  - tls_aes128.h
  - tls_builtin.c
  - tls_chacha20.c
  - tls_chacha20.h
- test
  - unit_test.c

13 files changed +528 -252 lines changed

Search within code

mongoose.c

Load Diff

Large diffs are not rendered by default.

mongoose.h

```
@@ -1961,7 +1961,9 @@ int mg_aes_gcm_encrypt(unsigned char *output, const unsigned char *input,
1961 1961  int mg_aes_gcm_decrypt(unsigned char *output, const unsigned char *input,
1962 1962                        size_t input_length, const unsigned char *key,
1963 1963                        const size_t key_len, const unsigned char *iv,
1964      -                     const size_t iv_len);
     1964 +                     const size_t iv_len, unsigned char *aead,
     1965 +                     size_t aead_len, const unsigned char *tag,
     1966 +                     const size_t tag_len);
1965 1967
1966 1968  #endif /* TLS_AES128_H */
1967 1969
@@ -2727,6 +2729,7 @@ PORTABLE_8439_DECL size_t mg_chacha20_poly1305_encrypt(
2727 2729  PORTABLE_8439_DECL size_t mg_chacha20_poly1305_decrypt(
2728 2730      uint8_t *restrict plain_text, const uint8_t key[RFC_8439_KEY_SIZE],
2729 2731      const uint8_t nonce[RFC_8439_NONCE_SIZE],
     2732 +    const uint8_t *restrict ad, size_t ad_size,
2730 2733      const uint8_t *restrict cipher_text, size_t cipher_text_size);
2731 2734  #if defined(__cplusplus)
2732 2735  }
```

src/dns.c

```
@@ -287,13 +287,15 @@ void mg_resolve(struct mg_connection *c, const char *url) {
287 287    }
```

```
288 288   }
289 289
    290 + // Response header length is 10 bytes
290 291   static const uint8_t mdns_answer[] = {
291 292       0, 1,          // 2 bytes - record type, A
292 293       0, 1,          // 2 bytes - address class, INET
293 294       0, 0, 0, 120,  // 4 bytes - TTL
294 295       0, 4           // 2 bytes - address length
295 296   };
296 297
    298 + // A name length is name->len + '.local' + 2 = name->len + 8
297 299   static uint8_t *build_name(struct mg_str *name, uint8_t *p) {
298 300     *p++ = (uint8_t) name->len;  // label 1
299 301     memcpy(p, name->buf, name->len), p += name->len;
        @@ -305,6 +307,7 @@ static uint8_t *build_name(struct mg_str *name, uint8_t
        *p) {
305 307
306 308   void mg_getlocaddr(struct mg_connection *, struct mg_addr *, struct mg_addr *);
307 309
    310 + // An A record length is 10 + 4 = 14 bytes
308 311   static uint8_t *build_a_record(struct mg_connection *c, uint8_t *p,
309 312                                  struct mg_addr *addr) {
310 313     memcpy(p, mdns_answer, sizeof(mdns_answer)), p += sizeof(mdns_answer);
        @@ -326,6 +329,7 @@ static uint8_t *build_a_record(struct mg_connection *c,
        uint8_t *p,
326 329     return p;
327 330   }
328 331
    332 + // A srv name length is r->srvcproto.len + '.local' + 2 = r->srvcproto.len + 8
329 333   static uint8_t *build_srv_name(uint8_t *p, struct mg_dnssd_record *r) {
330 334     *p++ = (uint8_t) r->srvcproto.len - 5;  // label 1, up to '._tcp'
331 335     memcpy(p, r->srvcproto.buf, r->srvcproto.len), p += r->srvcproto.len;
        @@ -346,6 +350,7 @@ static uint8_t *build_mysrv_name(struct mg_str *name,
        uint8_t *p,
346 350   }
347 351   #endif
348 352
    353 + // A PTR record length is 10 + name->len + 3 = name->len + 13
349 354   static uint8_t *build_ptr_record(struct mg_str *name, uint8_t *p, uint16_t o) {
350 355     uint16_t offset = mg_htons(o);
```

```
351  356      memcpy(p, mdns_answer, sizeof(mdns_answer));
          @@ -360,6 +365,7 @@ static uint8_t *build_ptr_record(struct mg_str *name,
          uint8_t *p, uint16_t o) {
360  365      return p;
361  366    }
362  367
     368  + // An SRV record length is 10 + name->len + 9 = name->len + 19
363  369    static uint8_t *build_srv_record(struct mg_str *name, uint8_t *p,
364  370                                     struct mg_dnssd_record *r, uint16_t o) {
365  371      uint16_t port = mg_htons(r->port);
          @@ -380,6 +386,7 @@ static uint8_t *build_srv_record(struct mg_str *name,
          uint8_t *p,
380  386      return p;
381  387    }
382  388
     389  + // A TXT record length is r->txt.len (txt contents) + 10
383  390    static uint8_t *build_txt_record(uint8_t *p, struct mg_dnssd_record *r) {
384  391      uint16_t len = mg_htons((uint16_t) r->txt.len);
385  392      memcpy(p, mdns_answer, sizeof(mdns_answer));
          @@ -390,6 +397,8 @@ static uint8_t *build_txt_record(uint8_t *p, struct
          mg_dnssd_record *r) {
390  397      return p;
391  398    }
392  399
     400  + // Each additional record has a 2-byte field pointing to the name label
     401  +
393  402    // RFC-6762 16: case-insensitivity --> RFC-1034, 1035
394  403
395  404    static void handle_mdns_query(struct mg_connection *c) {
          @@ -478,6 +487,10 @@ static void handle_mdns_query(struct mg_connection *c)
          {
478  487        uint8_t *o = p, *aux;
479  488        uint16_t offset;
480  489        if (respname->buf == NULL || respname->len == 0) return;
     490  +     if ((sizeof(*h) + req.r->srvcproto.len + 8 + respname->len + 13 + 2 +
     491  +          respname->len + 19 + 2 + req.r->txt.len + 10 + 2 + 14) >
     492  +         sizeof(buf))  // srv name + PTR + 2 + SRV + 2 + TXT + 2 + A
     493  +       return;
481  494        h->num_other_prs = mg_htons(3);  // 3 additional records
482  495        p = build_srv_name(p, req.r);
```

```
483  496          aux = build_ptr_record(respname, p, (uint16_t) (o - buf));
     @@ -498,12 +511,18 @@ static void handle_mdns_query(struct mg_connection *c)
     {
498  511          *p |= 0xC0, p += 2;
499  512          p = build_a_record(c, p, req.addr);
500  513        } else if (rr.atype == MG_DNS_RTYPE_TXT) {
     514  +       if ((sizeof(*h) + req.r->srvcproto.len + 8 + req.r->txt.len + 10) >
     515  +           sizeof(buf))  // srv name + TXT
     516  +         return;
501  517          p = build_srv_name(p, req.r);
502  518          p = build_txt_record(p, req.r);
503  519        } else if (rr.atype == MG_DNS_RTYPE_SRV) {  // serve SRV + A
504  520          uint8_t *o, *aux;
505  521          uint16_t offset;
506  522          if (respname->buf == NULL || respname->len == 0) return;
     523  +       if ((sizeof(*h) + req.r->srvcproto.len + 8 + respname->len + 19 + 2 +
     524  +            14) > sizeof(buf))  // srv name + SRV + 2 + A
     525  +         return;
507  526          h->num_other_prs = mg_htons(1);  // 1 additional record
508  527          p = build_srv_name(p, req.r);
509  528          o = p - 7;  // point to '.local' label (\x05local\x00)
     @@ -517,6 +536,8 @@ static void handle_mdns_query(struct mg_connection *c) {
517  536        } else {  // A requested
518  537          // RFC-6762 6: 0 Auth, 0 Additional RRs
519  538          if (respname->buf == NULL || respname->len == 0) return;
     539  +       if ((sizeof(*h) + respname->len + 8 + 14) > sizeof(buf))  // name + A
     540  +         return;
520  541          p = build_name(respname, p);
521  542          p = build_a_record(c, p, req.addr);
522  543        }
```

src/http.c

```
     @@ -237,6 +237,8 @@ static const char *skiptorn(const char *s, const char
     *end, struct mg_str *v) {
237  237    static bool mg_http_parse_headers(const char *s, const char *end,
238  238                                      struct mg_http_header *h, size_t max_hdrs)
           {
239  239      size_t i, n;
     240  +   int cl_count = 0, te_count = 0, auth_count = 0;
```

```
                241 +   int conn_count = 0, cookie_count = 0;
240             242     for (i = 0; i < max_hdrs; i++) {
241             243       struct mg_str k = {NULL, 0}, v = {NULL, 0};
242             244       if (s >= end) return false;
                    @@ -252,6 +254,13 @@ static bool mg_http_parse_headers(const char *s,
                    const char *end,
252             254         while (v.len > 0 && (v.buf[v.len - 1] == ' ' || v.buf[v.len - 1] ==
                    '\t')) {
253             255           v.len--;  // Trim spaces
254             256         }
                257 +     // detect duplicated headers -> discard
                258 +     if (((mg_strcasecmp(k, mg_str("Content-Length")) == 0) && (++cl_count >
                    1)) ||
                259 +       ((mg_strcasecmp(k, mg_str("Transfer-Encoding")) == 0) && (++te_count >
                    1)) ||
                260 +       ((mg_strcasecmp(k, mg_str("Authorization")) == 0) && (++auth_count >
                    1)) ||
                261 +       ((mg_strcasecmp(k, mg_str("Cookie")) == 0) && (++cookie_count > 1)) ||
                262 +       ((mg_strcasecmp(k, mg_str("Connection")) == 0) && (++conn_count > 1)))
                263 +       return false;
255             264       // MG_INFO(("--HH [%.*s] [%.*s]", (int) k.len, k.buf, (int) v.len,
                    v.buf));
256             265       h[i].name = k, h[i].value = v;  // Success. Assign values
257             266     }
                    @@ -286,6 +295,8 @@ int mg_http_parse(const char *s, size_t len, struct
                    mg_http_message *hm) {
286             295     // If we're given a version, check that it is HTTP/x.x
287             296     version_prefix_valid =
288             297         hm->proto.len > 5 && (mg_ncasecmp(hm->proto.buf, "HTTP/", 5) == 0);
                298 +   if (!is_response && !version_prefix_valid)
                299 +     return -1; // no version detected in request
289             300     if (!is_response && hm->proto.len > 0 &&
290             301         (!version_prefix_valid || hm->proto.len != 8 ||
291             302          (hm->proto.buf[5] < '0' || hm->proto.buf[5] > '9') ||
                    @@ -1035,7 +1046,11 @@ static void http_cb(struct mg_connection *c, int
                    ev, void *ev_data) {
1035            1046          hm.message.len = c->recv.len - ofs;  // and closes now, deliver MSG
1036            1047          hm.body.len = hm.message.len - (size_t) (hm.body.buf -
                    hm.message.buf);
1037            1048        }
```

```
1038      -       if ((te = mg_http_get_header(&hm, "Transfer-Encoding")) != NULL) {
      1049  +       bool is_http_1_0 =
      1050  +           hm.proto.len > 8 && mg_ncasecmp(hm.proto.buf, "HTTP/1.0", 8) == 0;
      1051  +       // HTTP/1.0 does not use "Transfer-Encoding: chunked"
      1052  +       if (!is_http_1_0 &&
      1053  +           (te = mg_http_get_header(&hm, "Transfer-Encoding")) != NULL) {
1039  1054            if (mg_strcasecmp(*te, mg_str("chunked")) == 0) {
1040  1055              is_chunked = true;
1041  1056            } else {
```

src/json.c

```
@@ -60,15 +60,28 @@ static double mg_atod(const char *p, int len, int
*numlen) {
60  60
61  61      // Exponential
62  62      if (i < len && (p[i] == 'e' || p[i] == 'E')) {
63        -     int j, exp = 0, minus = 0;
      63  +     int exp = 0, minus = 0;
64  64        i++;
65  65        if (i < len && p[i] == '-') minus = 1, i++;
66  66        if (i < len && p[i] == '+') i++;
67  67        while (i < len && p[i] >= '0' && p[i] <= '9' && exp < 308)
68  68          exp = exp * 10 + (p[i++] - '0');
69        -     if (minus) exp = -exp;
70        -     for (j = 0; j < exp; j++) d *= 10.0;
71        -     for (j = 0; j < -exp; j++) d /= 10.0;
      69  +     // use fast exponentiation
      70  +     // https://en.wikipedia.org/wiki/Exponentiation_by_squaring
      71  +     if (exp != 0) {
      72  +       double x = 10, y = 1;
      73  +       if (exp > 308) exp = 308;
      74  +       if (minus) x = 0.1;
      75  +       while (exp > 1) {
      76  +         if (exp & 1) {
      77  +           y *= x;
      78  +           --exp;
      79  +         }
      80  +         x *= x;
      81  +         exp >>= 1;
```

```
      82  +       }
      83  +       d *= x * y;
      84  +     }
72    85        }
73    86
74    87        if (numlen != NULL) *numlen = i;
```

src/net_builtin.c

```
      @@ -551,9 +551,13 @@ static struct mg_connection *getpeer(struct mg_mgr
      *mgr, struct pkt *pkt,
551   551           !(c->loc.is_ip6 ^ (pkt->ip6 != NULL)))  // IP or IPv6 to same dest
552   552         break;
553   553       if (!c->is_udp && pkt->tcp && c->loc.port == pkt->tcp->dport &&
554         -         !(c->loc.is_ip6 ^ (pkt->ip6 != NULL)) &&
555         -         lsn == (bool) c->is_listening &&
556         -         (lsn || c->rem.port == pkt->tcp->sport))
      554 +         ((lsn && c->is_listening && !(c->loc.is_ip6 ^ (pkt->ip6 != NULL))) ||
      555 +          (!lsn && !c->is_listening && c->rem.port == pkt->tcp->sport &&
      556 +           ((!c->loc.is_ip6 && c->rem.addr.ip4 == pkt->ip->src)
      557 + #if MG_ENABLE_IPV6
      558 +            || (c->loc.is_ip6 && MG_IP6MATCH(c->rem.addr.ip6, pkt->ip6->src))
      559 + #endif
      560 +                ))) // validate addr for established (not listening) conns
557   561         break;
558   562     }
559   563     return c;
      @@ -1497,14 +1501,15 @@ static void backlog_poll(struct mg_mgr *mgr) {
1497  1501   }
1498  1502
1499  1503   // process options (MSS)
1500        - static void handle_opt(struct connstate *s, struct tcp *tcp, bool ip6) {
      1504 + static bool handle_opt(struct connstate *s, struct tcp *tcp, bool ip6) {
1501  1505     uint8_t *opts = (uint8_t *) (tcp + 1);
1502  1506     int len = 4 * ((int) (tcp->off >> 4) - ((int) sizeof(*tcp) / 4));
1503  1507     s->dmss = ip6 ? 1220 : 536;  // assume default, RFC-9293 3.7.1
1504  1508     while (len > 0) {            // RFC-9293 3.1 3.2
1505  1509       uint8_t kind = opts[0], optlen = 1;
1506  1510       if (kind != 1) {         // No-Operation
```

```
1507  1511          if (kind == 0) break;  // End of Option List
      1512  +       if (len < 2 || opts[1] == 0) return false; // Malformed options
1508  1513          optlen = opts[1];
1509  1514          if (kind == 2 && optlen == 4)  // set received MSS
1510  1515            s->dmss = (uint16_t) (((uint16_t) opts[2] << 8) + opts[3]);
      @@ -1513,6 +1518,7 @@ static void handle_opt(struct connstate *s, struct
      tcp *tcp, bool ip6) {
1513  1518        opts += optlen;
1514  1519        len -= optlen;
1515  1520      }
      1521  +   return true;
1516  1522    }
1517  1523
1518  1524    static void rx_tcp(struct mg_tcpip_if *ifp, struct pkt *pkt) {
      @@ -1527,7 +1533,7 @@ static void rx_tcp(struct mg_tcpip_if *ifp, struct
      pkt *pkt) {
1527  1533      // - check clients (Group 1) and established connections (Group 3)
1528  1534      if (c != NULL && c->is_connecting && pkt->tcp->flags == (TH_SYN | TH_ACK))
      {
1529  1535        // client got a server connection accept
1530        -     handle_opt(s, pkt->tcp, pkt->ip6 != NULL);  // process options (MSS)
      1536  +     if (!handle_opt(s, pkt->tcp, pkt->ip6 != NULL)) return;  // process
      options (MSS)
1531  1537        s->seq = mg_ntohl(pkt->tcp->ack), s->ack = mg_ntohl(pkt->tcp->seq) + 1;
1532  1538        tx_tcp_ctrlresp(ifp, pkt, TH_ACK, pkt->tcp->ack);
1533  1539        c->is_connecting = 0;  // Client connected
      @@ -1538,8 +1544,9 @@ static void rx_tcp(struct mg_tcpip_if *ifp, struct
      pkt *pkt) {
1538  1544      } else if (c != NULL && c->is_connecting && pkt->tcp->flags != TH_ACK) {
1539  1545        mg_error(c, "connection refused");
1540  1546      } else if (c != NULL && pkt->tcp->flags & TH_RST) {
1541        -     // TODO(): validate RST is within window (and optional with proper ACK)
1542        -     mg_error(c, "peer RST");  // RFC-1122 4.2.2.13
      1547  +     uint32_t seqno = mg_ntohl(pkt->tcp->seq);
      1548  +     if (seqno >= s->ack && seqno < (s->ack + MG_TCPIP_WIN))  // RFC-9293
      3.5.3
      1549  +       mg_error(c, "peer RST");  // RFC-1122 4.2.2.13
1543  1550      } else if (c != NULL) {
1544  1551        // process segment
1545  1552        s->tmiss = 0;                         // Reset missed keep-alive counter
```

```
@@ -1561,7 +1568,7 @@ static void rx_tcp(struct mg_tcpip_if *ifp, struct pkt *pkt) {
1561 1568         int key;
1562 1569         uint32_t isn;
1563 1570         if (pkt->tcp->sport != 0) {
1564      -         handle_opt(&cs, pkt->tcp, pkt->ip6 != NULL);  // process options (MSS)
     1571 +         if (!handle_opt(&cs, pkt->tcp, pkt->ip6 != NULL)) return;  // process options (MSS)
1565 1572           key = backlog_insert(c, pkt->tcp->sport,
1566 1573                                cs.dmss);  // backlog options (MSS)
1567 1574           if (key < 0) return;  // no room in backlog, discard SYN, client retries
```

src/str.c

```
@@ -87,7 +87,9 @@ bool mg_match(struct mg_str s, struct mg_str p, struct mg_str *caps) {
87  87       } else if (i < p.len && (p.buf[i] == '*' || p.buf[i] == '#')) {
88  88         if (caps && !caps->buf) caps->len = 0, caps->buf = &s.buf[j];  // Init cap
89  89         ni = i++, nj = j + 1;
90      -     } else if (nj > 0 && nj <= s.len && ((ni < p.len && p.buf[ni] == '#') || s.buf[j] != '/')) {
    90 +     } else if (nj > 0 && nj <= s.len &&
    91 +                ((ni < p.len && p.buf[ni] == '#') ||
    92 +                 (j < s.len && s.buf[j] != '/'))) {
91  93         i = ni, j = nj;
92  94         if (caps && caps->buf == NULL && caps->len == 0) {
93  95           caps--, caps->len = 0;  // Restart previous cap
@@ -168,8 +170,7 @@ bool mg_str_to_num(struct mg_str str, int base, void *val, size_t val_len) {
168 170         i++, ndigits++;
169 171       }
170 172       break;
171      -   default:
172      -     return false;
    173 +   default: return false;
173 174     }
174 175     if (ndigits == 0) return false;
```

```
175  176      if (i != str.len) return false;
```

Comments 0

Please sign in to comment.