Red Hat
Enterprise Linux

TLS 1.3 Handshake In Linux Kernel

Xin Long

Aug, 2022

Red Hat

Overview

APIs & Usage:

TLS_HS (msg)

|-- TLS_HS_GEN (sk)

| |

| |---- TCP (TLS_HS_TCP)

| |---- NFS

|

|-- QUIC

Implementation:

repo: https://github.com/lxin/tls_hs

- include/crypto/tls_hs.h

- crypto/tls_hs.c

(CONFIG_CRYPTO_TLS_HS=m/y)

APIs: TLS_HS Core (msg)

tls = tls_handshake_create(is_serv)

tls_handshake_get/set(tls, opt, vec)

- opt:

- TLS_T_PKEY: private key.
- TLS_T_PSK:
- TLS_T_CA:
- TLS_T_CRT_REQ: client certificate request.
- TLS_T_CRT: certificate chain.
- TLS_T_EXT: unknown extension.
- TLS_T_EARLY: flag for early data send/recv.

state = tls_handshake(tls, msg)

- msg:

- **input: a tls 1.3 msg to process.**
- **output: a tls msg to send for reply.**

Note: when input is NULL, output is tls 1.3 CH/EE msg.

- state:

- TLS_ST_START
- TLS_ST_RCVD
- TLS_ST_WAIT
- TLS_ST_CONNECTED
- < 0: Error

APIs: TLS_HS Extra (msg)

- tls_secret_get(tls, level, srt)
- tls_hkdf_expand(tls, srt, h, l, k)
- tls_hkdf_extract(tls, srt, h, k)

- level:

  - TLS_SE_RMS: resumption secret.
  - TLS_SE_EA: early secret.
  - TLS_SE_HS: handshake secret.
  - TLS_SE_AP: master secret.

- tls_handshake_destroy(tls)
- tls_handshake_post(tls, type, msg)

- type:

  - TLS_P_NONE: process post the handshake msg.
  - TLS_P_TICKET: create session ticket msg.
  - TLS_P_KEY_UPDATE: create new keys and key update msg.

APIs: TLS_HS_GEN Core (sk)

tls = tls_sk_handshake(sk, data, keyring, flag)

- **sk**:

- **input: a TCP established socket.**
- **output: a kTLS socket with keys set.**

- data:

- early data to send and early data received.

- keyring:

- set for reading keys/crts from userspace, and left none if the keys/crts are set by kernel users via tls_handshake_set(...).

- flag:

- TLS_F_SERV: works as a server.
- TLS_F_PSK/CRT/CRT_REQ:
- TLS_F_NO_KTLS:

  w/o kTLS users can use tls_ap_de/encrypt()

  to send/recv app data.

- tls (return obj):

- users can either save it for future post handshake msg processing or destroy it.

APIs: TLS_HS_GEN Extra (sk)

- tls_sk_handshake_post(sk, tls, type, msg)
- tls_ap_encrypt(tls, data, seq)
- tls_ap_decrypt(tls, data, seq)

Usage: QUIC

Module in Kernel:

- repo: https://github.com/lxin/tls_hs/tree/quic
  - net/quic/*
  - include/net/quic/quic.h
  - include/uapi/linux/quic.h

  (CONFIG_IP_QUIC=m/y)

Kernel Code:

- handshake code: https://github.com/lxin/tls_hs/blob/quic/net/quic/frame.c#L654

User Programs:

- https://github.com/lxin/tls_hs-tests#ii-quic

Usage: TCP/NFS

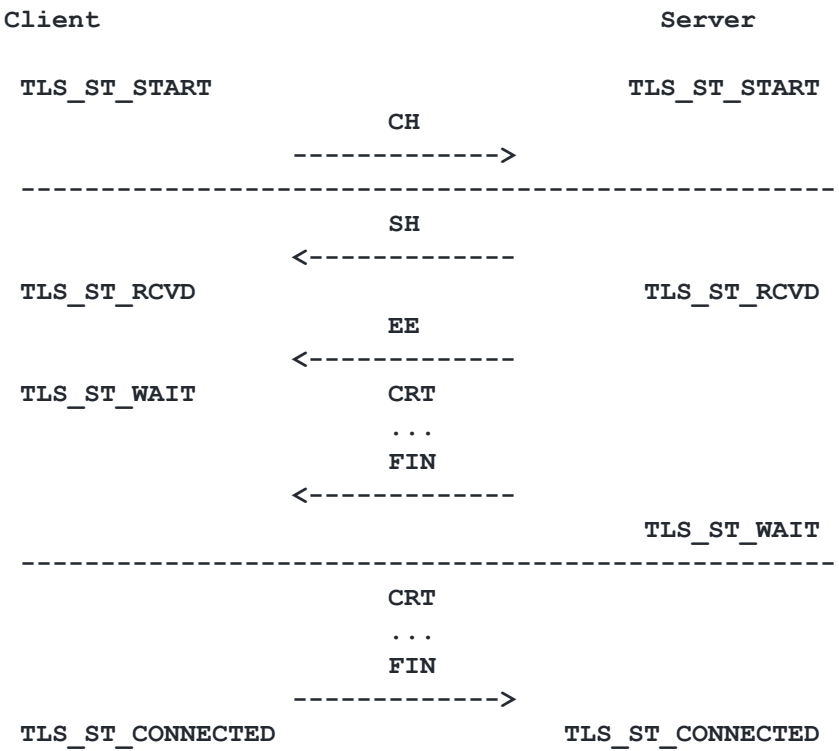
Kernel Code:

- TCP: https://github.com/lxin/tls_hs/blob/tcp/net/ipv4/tcp.c#L4363
- NFS: https://github.com/lxin/tls_hs/blob/sunrpc/net/sunrpc/xprtsock.c#L2433

User Programs:

- https://github.com/lxin/tls_hs-tests#i-tls_hs_tcp

Implementation: state transition



Red Hat

Implementation: ciphers & algorithms

- HKDF extract/expand: hmac(sha256)
- ECDH key exchange: secp256r1(0x0017)
- PSK exchange mode: psk_dhe_ke(1)
- Certificate: rsa_pkcs1_sha256(0x0401)/rsa_pss_rsae_sha256(0x0804)
- Signature Algorithm: rsa_pss_rsae_sha256(0x0804)
- AEAD: TLS_AES_128_GCM_SHA256(0x1301)

Implementation: functions

- Certificate Chain and CA on Both Sides
- PSK
- Session Resumption
- Early Data
- Hello Retry Request on Server

Implementation: messages & extensions

-	TLS_MT_HELLO_RETRY_REQUEST	0
-	TLS_MT_CLIENT_HELLO	1
-	TLS_MT_SERVER_HELLO	2
-	TLS_MT_NEWSESSION_TICKET	4
-	TLS_MT_END_OF_EARLY_DATA	5
-	TLS_MT_ENCRYPTED_EXTENSIONS	8
-	TLS_MT_CERTIFICATE	11
-	TLS_MT_CERTIFICATE_REQUEST	13
-	TLS_MT_CERTIFICATE_VERIFY	15
-	TLS_MT_FINISHED	20
-	TLS_MT_KEY_UPDATE	24

-	TLS_EXT_server_name	0
-	TLS_EXT_supported_groups	10
-	TLS_EXT_ec_point_formats	11
-	TLS_EXT_signature_algorithms	13
-	TLS_EXT_heartbeat	15
-	TLS_EXT_alpn	16
-	TLS_EXT_signed_cert_timestamp	18
-	TLS_EXT_padding	21
-	TLS_EXT_encrypt_then_mac	22
-	TLS_EXT_extended_master_srt	23
-	TLS_EXT_session_ticket	35
-	TLS_EXT_psk	41
-	TLS_EXT_early_data	42
-	TLS_EXT_supported_versions	43
-	TLS_EXT_cookie	44
-	TLS_EXT_psk_kex_modes	45
-	TLS_EXT_certificate_authorities	47
-	TLS_EXT_post_handshake_auth	49
-	TLS_EXT_signature_algs_cert	50
-	TLS_EXT_key_share	51

Implementation: memory management

struct tls_hs

Creating and Parsing msgs:

- struct tls_hello h;
- struct tls_vec ext; /* 1 page */
- struct tls_vec cmsg; /* 1 page */
- struct tls_vec omsg; /* 1 page */

Secrets and HS msg buffers:

- struct tls_vec buf[TLS_B_MAX];
- struct tls_vec srt[TLS_SE_MAX];

State and Flags:

- u8 state:2, early:1, is_serv:1, crt_req:1;

Crypto API algorithm objects:

- struct crypto_kpp *kpp_tfm;
- struct crypto_aead *aead_tfm;
- struct crypto_shash *srt_tfm;
- struct crypto_shash *hash_tfm;
- struct crypto_akcipher *akc_tfm;

Certs and Keys:

- struct tls_vec pkey;
- struct tls_crt *tcrts;
- struct tls_crt *rcrts;
- struct tls_crt *ca;
- struct tls_psk *psks;

END