draft-ietf-bootstrapped-tls-03

Dan Harkins & Owen Friel
EMU WG, IETF 117
What?

• Solves the onboarding *Catch-22*:
  “You need a credential to get on the network but you need to get on the network to get a credential”

• Reuse Wi-Fi alliance Easy Connect / Device Provisioning Profile (DPP) bootstrap approach for wired devices with limited or no user interface

• Reuse DPP EC bootstrap key pair and bootstrapping methods (e.g. QR code)

• Provides mutual authentication between bootstrapping client and server that knows client’s bootstrap public key

• No new TLS extensions, changes or new funky crypto required, uses other RFCs:
  • RFC 8773 Cert Based Auth with External PSK
  • RFC 7250 TLS with raw public key using bootstrapping key
  • RFC 9258 Importing External (PSKs) for TLS 1.3 to import derived PSK
  • RFC 5869 HKDF to derive External PSK ID from bootstrap key
Changes from IETF 116/draft-02 to make -03

• Removed unnecessary second HKDF when deriving EPSK from BSK
• Corrected bad reference
• Cleaned up language (Thank you Hannes)
New EPSK Derivation

• Given:
  • Base_Key = DER-encoded ASN.1 subjectPublicKeyInfo representation of the BSK public key

• Then RFC 9258 EPSK = (Base_Key, External_Identity, Hash)
  • External_Identity = HKDF-Expand(HKDF-Extract(<>, bskey), "tls13-bspsk-identity", L)
  • Hash is SHA-256 (as per rfc9258#section-5.1)

• RFC 9258 ImportedIdentity struct
  • external_identity = External_Identity
  • context = "tls13-bsk"
  • target_protocol = TLS1.3
  • target_kdf = HKDF_SHA256

• This removes one unnecessary HKDF calculation
  • draft-02 specifies Base_Key = HKDF-Expand(HKDF-Extract(<>, bskey), "tls13-imported-bsk", L)
TLS authentication w/DPP bootstrapping keys

Legend:
- present for bootstrapped-tls
- existing exchange

RFC 5869
RFC 8773
RFC 7250
Client
---------
ClientHello
+ cert_withExtern_psk
+ client_cert_type=RawPublicKey
+ key_share
+ pre_shared_key

ServerHello
+ cert_withExtern_psk
+ client_cert_type=RawPublicKey
+ key_share
+ pre_shared_key

{EncryptedExtensions}
{CertificateRequest}
{Certificate}
{CertificateVerify}

{Finished}

<?

Server uses RFC 9258 ImportedIdentity to lookup the bootstrap key

---
TEAP w/DPP bootstrapping keys

Authenticating Peer
--------------------
Authenticator

<--- EAP-Request/Identity

<--- EAP-Request/EAP-Type=TEAP(TLS Start)

---
EAP-Response/Identity
("tls-pok@eap-dpp.arpa")

authenticate TEAP with TLS-POK using bootstrapping key

---
PKCS#10 TLV

---
CSR Attrs TLV

---
PKCS#7 TLV

no initial realm, just say: “tls-pok@eap-dpp.arpa”

Supplicant’s subsequent connection uses provisioned certificate
Questions?

Next step... maybe WGLC?