Legacy RSASSA-PKCS1-v1_5 code points for TLS 1.3

David Benjamin, Google LLC

Andrei Popov, Microsoft Corp.

Commonly Used Client-side Cryptographic Devices Cannot Sign TLS 1.3 CertificateVerify

- TLS 1.3 [RFC8446] removed support for RSASSA-PKCS1-v1_5 [RFC8017] in CertificateVerify messages in favor of RSASSA-PSS.
- Widely-deployed hardware cryptographic devices protecting TLS client certificates cannot produce TLS-compatible RSASSA-PSS signatures:
  - TPM specifications prior to 2.0 did not define RSASSA-PSS support.
  - TPM 2.0 includes RSASSA-PSS, but only those TPM 2.0 devices compatible with FIPS 186-4 can be relied upon to use the salt length matching the digest length.
  - Similar issues observed with certain smart cards.
- Windows telemetry shows only ~15% of the deployed TPM devices are TPM 2.0 operating in FIPS 186-4 mode.

This Blocks TLS 1.3 Deployments

- TLS version is often negotiated before the client certificate is selected, therefore affected TLS clients, generally, cannot detect the issue in advance and disable TLS 1.3/RSASSA-PSS.
- Insecure TLS version fallbacks implemented in applications break TLS security analysis and may introduce vulnerabilities [POODLE].
- This means that the TLS server cannot enable TLS 1.3 and negotiate TLS 1.2 with PSS-incapable clients only.
- Entire deployments are stuck with TLS 1.2:
    - Open to retroactive decryption by attackers using quantum computers.
    - Possibly, leaking client certificate details to network observers.

Allowing RSASSA-PSS in the Client CertificateVerify

```
enum {   rsa_pkcs1_sha256_legacy(0x0420),
         rsa_pkcs1_sha384_legacy(0x0520),
         rsa_pkcs1_sha512_legacy(0x0620),
} SignatureScheme;
```

- TLS implementations SHOULD disable these code points by default.
- Only defined for signatures in the client CertificateVerify message.
- Servers advertising support for RSASSA-PKCS-v1_5 signatures in certificates should use the rsa_pkcs1_* definitions in [RFC8446].
- Clients MUST NOT advertise these values in the signature_algorithms extension of the ClientHello.
- Clients MUST NOT accept these values in the server CertificateVerify message.

Allowing RSASSA-PSS in the Client CertificateVerify

```
enum {    rsa_pkcs1_sha256_legacy(0x0420),
          rsa_pkcs1_sha384_legacy(0x0520),
          rsa_pkcs1_sha512_legacy(0x0620),
} SignatureScheme;
```

- Servers that wish to support clients authenticating with legacy RSASSA-PKCS1-v1_5 keys MAY send these values in the signature_algorithms extension of the CertificateRequest message and accept them in the client CertificateVerify message.
- Servers MUST NOT accept these code points if not offered in the CertificateRequest message.
- Clients with such legacy keys MAY negotiate the use of these signature algorithms if offered by the server. Clients SHOULD NOT negotiate them with keys that support RSASSA-PSS.

Requesting Adoption on the Standards Track

https://datatracker.ietf.org/doc/draft-davidben-tls13-pkcs1/

- David Benjamin
  Google LLC
  Email: davidben@google.com

- Andrei Popov
  Microsoft Corp.
  Email: andreipo@microsoft.com