ERICSSON

Date: May 16, 2023

Ericsson AB
Group Function Technology
SE-164 80 Stockholm
SWEDEN

Comments on 1800-37A, Addressing Visibility Challenges with TLS 1.3

Dear NIST,

Thanks for your continuous efforts to produce well-written open-access security documents.

Please find below our comments on 1800-37A:

- The Transport Layer Security (TLS) protocol is indeed an essential building block for enterprise security. It would be good if NIST clarified if the project is only considering the TLS 1.3 protocol itself or if the scope is all security protocols using the TLS 1.3 handshake (TLS 1.3, DTLS 1.3, QUIC, EAP-TLS, EAP-TTLS, EAP-FAST 1.3, PEAP 1.3, TEAP 1.3, DTLS/SCTP, WebRTC data channels, DTLS-SRTP, etc.) or a subset like HTTPS. It would also be good to clarify if any specific use cases of these protocols are more in focus than others. The security and privacy implications of visibility solutions might be worse in some protocols and use cases than in others.

- *"TLS 1.3, has been strengthened so that even if a TLS-enabled server is compromised, the contents of its previous TLS communications are still protected—better known as forward secrecy. In TLS 1.2 forward secrecy is optional, while in TLS 1.3 it is required."*

  Good that NIST mentions the important security benefits of ephemeral key exchange. In addition to protecting previous TLS communications, the ephemeral key exchange in TLS 1.3 also protects future TLS communication against passive attackers. The differences in security impact between key exchange without forward secrecy (psk_ke), key exchange with only forward secrecy (key_update), and ephemeral key exchange (ecdhe) are illustrated in Figure 1 borrowed from [1]. There are very strong reasons why ephemeral key exchange is required in TLS 1.3. The soon to be released revision of the TLS 1.3 specification [2] contains a new paragraph describing the importance of frequently rerunning ephemeral Diffie-Hellman as well as the security implications of failing to do so.

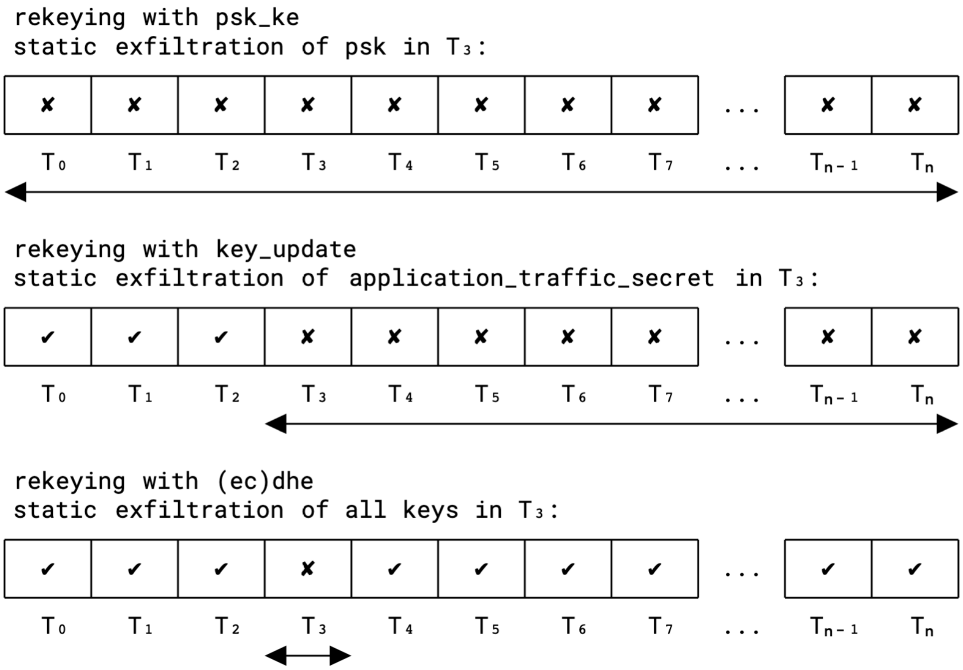
rekeying with psk_ke
static exfiltration of psk in T₃:
✘ ✘ ✘ ✘ ✘ ✘ ✘ ✘ ... ✘ ✘
T₀ T₁ T₂ T₃ T₄ T₅ T₆ T₇ ... Tₙ₋₁ Tₙ
rekeying with key_update
static exfiltration of application_traffic_secret in T₃:
✓ ✓ ✓ ✘ ✘ ✘ ✘ ✘ ... ✘ ✘
T₀ T₁ T₂ T₃ T₄ T₅ T₆ T₇ ... Tₙ₋₁ Tₙ
rekeying with (ec)dhe
static exfiltration of all keys in T₃:
✓ ✓ ✓ ✘ ✓ ✓ ✓ ✓ ... ✓ ✓
T₀ T₁ T₂ T₃ T₄ T₅ T₆ T₇ ... Tₙ₋₁ Tₙ

Figure 1: Impact of static key exfiltration in time period T_3

– *"Forward secrecy conflicts with passive decryption techniques that are widely used by enterprises"*

  The TLS 1.3 key update mechanism provides forward secrecy but does not conflict with passive decryption techniques used by enterprises. It is the use of ephemeral key exchange that conflicts with these techniques. Passive decryption techniques are also widely used by attackers such as hostile nation-state actors. This should be mentioned in the document.

– *"enterprises to choose between using the old TLS 1.2 protocol or adopting TLS 1.3 with an alternative method for internal traffic visibility."*
  "If an enterprise chooses the old TLS 1.2 protocol"
  "Enterprises using the old TLS 1.2 protocol without forward secrecy"

  Using the old obsolete TLS 1.2 protocol beyond January 2024 violates NIST SP 800-52 [3]. NIST SP 800-52 requires support of TLS 1.3 by January 2024 without exceptions. This means that two compliant nodes will never negotiate TLS 1.2. IETF is planning to deprecate and discourage use of TLS 1.2 [4]. NIST should make it clear that choosing the old obsolete TLS 1.2 protocol or using TLS without forward secrecy are not acceptable.

– *"However, TLS 1.2 visibility solutions provide more privilege than is needed to just view the traffic."*

We strongly agree, this is not following zero trust principles and these kinds of solutions should therefore be phased out.

- *"In the first option, the enterprise would provision bounded-lifetime Diffie-Hellman key pairs for TLS 1.3 servers as a substitute for the standard ephemeral key pairs. In the second case, the server would use ephemeral Diffie-Hellman key pairs as specified in TLS 1.3 and the enterprise would retain the symmetric key used to encrypt the connection."*

  The first option clearly violates the TLS 1.3 standard and also completely breaks several of the TLS 1.3 security properties, namely "Forward secret with respect to long-term keys" and "Protection of endpoint identities". That reuse violates forward secrecy with respect to long-term keys is obvious. By comparing key shares in different handshakes an attacker can track an endpoint or reveal the identity of the TLS server that a user connected to. This is a significant violation of user privacy. The revised TLS 1.3 specification [2] contains a new normative requirement stating that to prevent tracking and identification, client and servers SHOULD NOT reuse a key share for multiple connections. Informing the user after the handshake has already taken place does not help at all.

  Reuse of key shares also violates US government's zero trust principles. Two essential zero trust principles defined by US government are to assume that breach is inevitable or has likely already occurred [5], and to minimize impact when breach occur [6]. Reusing key shares is the opposite of this as it expands the impact of breach. Reuse of key shares is in violation of the zero trust principles. To always use ephemeral key exchange follows directly from the two above stated zero trust principles and is therefore required for any zero trust network. Several governments are recommending to always use ephemeral key exchange. Instead of discussing how to achieve less zero trust by violating established security standards such as TLS 1.3, NIST should strengthen its requirements and recommendations regarding always using ephemeral key exchange and frequently rerunning ephemeral key exchange in all protocols.

Conclusion

Visibility is an important problem, and we think it is excellent NIST have started this project. It is very good that NIST is not even discussing NULL encryption as an option. We completely agree with NIST that encryption of all traffic without exceptions, also in enterprise networks, is a requirement [7]:

> *"The entire enterprise private network is not considered an implicit trust zone. Assets should always act as if an attacker is present on the enterprise network, and communication should be done in the most secure manner available. This entails actions such as authenticating all connections and encrypting all traffic."*

We are strongly against the reuse of key shares. By discussing reuse of key shares, NIST is doing both companies that sell visibility products and their customers a disservice. Reuse of key shares is not an acceptable solution anymore and any company believing so has already lost the 5 years since TLS 1.3 was published to develop and deploy new acceptable solutions. We strongly suggest that NIST completely remove the solution with reuse of key shares and clearly states that reuse of key shares is forbidden as is violates not only the TLS 1.3 standard, but also the TLS security properties, the TLS privacy properties, and zero trust principles. We find it very surprising that NIST is discussing solutions that clearly violates the TLS 1.3 standard and its security and privacy properties. Sharing of symmetric traffic keys is a much better solution. The amount of data needed to securely transport a symmetric traffic key is negligible compared to the total data in a TLS connection.

We think that most or all of the other discussed solutions (endpoint mechanisms, analyze network traffic without decryption, sharing of symmetric traffic keys) are acceptable solutions that should be studied further in the project. Visibility solutions providing limited privilege should be preferred. Optimally the privilege should only be to view the required subset of the traffic.

Best Regards,
John Preuß Mattsson,
Expert Cryptographic Algorithms and Security Protocols

[1] IETF, “NULL Encryption and Key Exchange Without Forward Secrecy are Discouraged”
https://datatracker.ietf.org/doc/html/draft-mattsson-tls-psk-ke-dont-dont-dont

[2] IETF RFC8446bis, “The Transport Layer Security (TLS) Protocol Version 1.3“
https://datatracker.ietf.org/doc/html/draft-ietf-tls-rfc8446bis

[3] NIST SP 800-52, “Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations”
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-52r2.pdf

[4] IETF, “TLS 1.2 Deprecation”
https://datatracker.ietf.org/meeting/116/materials/slides-116-tls-tls-12-deprecation-discussion-00

[5] NSA, "Embracing a Zero Trust Security Model"
https://media.defense.gov/2021/Feb/25/2002588479/-1/-1/0/CSI_EMBRACING_ZT_SECURITY_MODEL_UOO115131-21.PDF

[6] NIST SP 1800-35B, "Implementing a Zero Trust Architecture"
https://www.nccoe.nist.gov/sites/default/files/2022-12/zta-nist-sp-1800-35b-preliminary-draft-2.pdf

[7] NIST SP 800-207, "Zero Trust Architecture"
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf