Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 1, 2017

M. Green
Cryptography Engineering LLC

October 31, 2016

Data Center use of Static Diffie-Hellman in TLS 1.3
<draft-green-tls-static-dh-in-tls13-00>

Abstract

Unlike earlier versions of TLS, current drafts of TLS 1.3 have instead adopted ephemeral-mode Diffie-Hellman and elliptic-curve Diffie-Hellman as the primary cryptographic key exchange mechanism used in TLS. This document describes an optional configuration for TLS servers that allows for the use of a static Diffie-Hellman secret for all TLS connections made to the server. Passive monitoring of TLS connections can be enabled by installing a corresponding copy of this key in each monitoring device.

Status of This Memo

Copyright Notice

1. Introduction

Unlike earlier versions of TLS, current drafts of TLS 1.3 [draft-ietf-tls-tls13-18] do not provide support for the RSA handshake -- and have instead adopted ephemeral-mode Diffie-Hellman and elliptic-curve Diffie-Hellman as the primary cryptographic key exchange mechanism used in TLS.

While ephemeral (EC) Diffie-Hellman is in nearly all ways an improvement over the TLS RSA handshake, it has a limitation in certain enterprise settings. Specifically, the use of ephemeral (PFS) ciphersuites is not compatible with enterprise network monitoring tools such as Intrusion Detection Systems (IDS) that must passively monitor intranet TLS connections made to endpoints under the enterprise's control. This includes TLS connections made from enterprise load balancers at the edge of the enterprise network to internal enterprise TLS servers. It does not include TLS connections traveling over the external Internet.

Such monitoring is ubiquitous and indispensable in some industries, and loss of this capability may slow adoption of TLS 1.3.

This document describes an optional configuration for TLS servers that allows for the use of a static Diffie-Hellman secret for all TLS connections made to the server. Passive monitoring of TLS connections can be enabled by installing a corresponding copy of this key in each monitoring device.

An advantage of this proposal is that it can be implemented using software modifications to the TLS server only, without the need to make changes to TLS client implementations.

2. Summary of the existing Diffie-Hellman handshake

In TLS 1.3, servers exchange keys using two primary modes, Ephemeral Diffie-Hellman (DHE) and Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). In a simplified view of the full handshake, the following steps occur:

1. The client generates an ephemeral public and private key, and transmits the public key within a "key_share" message, along with a random nonce (ClientHello.random).
2. The server generates an ephemeral public and private key, and transmits the public key within a "key_share" message, along with a random nonce (ServerHello.random).
3. The two parties now calculate a shared (EC) Diffie-Hellman secret by combining the other party's ephemeral public key with their own ephemeral secret.
4. A series of traffic and handshake keys is derived by combining this shared secret with various inputs from the handshake, including the ClientHello.random and ServerHello.random.
5. Data encryption is performed using these keys.

3. Using static (EC) Diffie-Hellman on the server

The proposal embodied in this draft modifies the standard TLS handshake summarized above in the following ways.

First, for each elliptic curve (and FF-DH parameter length) supported by the server, the server is provisioned with a random static (EC) Diffie- Hellman private key. This key is generated at server installation, and is rotated at periodic intervals appropriate for any long-term server key. These keys could also be generated at a central key management server and distributed (in a secure encrypted form) to many endpoint servers.

The static secret key is used to derive a fixed, static (EC) Diffie-Hellman public key.

All steps of the original handshake proceed as above, with the following modification to server behavior. Step (2) proceeds as follows:

2. The server transmits the static public key within a "key_share" message, along with a random nonce (ServerHello.random).

4. Security considerations

We now consider the security implications of the change described above:

i. The shift from fully-ephemeral (EC) Diffie-Hellman to partially static Diffie-Hellman affects the security properties offered by the TLS 1.3 handshake by eliminating the Perfect Forward Secrecy (PFS) property provided by the server. If a server is compromised and the private key is stolen, then an attacker who observes any TLS handshake (even one that occurred prior to the compromise) will be able to recover traffic encryption keys and will be able to decrypt traffic.

ii. As long as the server static secret key is not compromised, the resulting protocol will provide strong cryptographic security, as long as the Diffie-Hellman parameters (e.g., finite-field group or elliptic curve) are correctly generated and provide security at a sufficient cryptographic security level.

iii. Replay attacks are prevented due to the fact that the server generates a unique 32-byte ServerHello.random field using a strong random number generator, and this value is included in the traffic key derivation procedure.

iv. A flaw in the generation of finite-field Diffie-Hellman parameters or the use of an insecure implementation could leak some bits of the static secret key over time. This risk is not present in ephemeral DH implementations. Implementers should use care to avoid such pitfalls.

Thus the modification described in Section 4 represents a deliberate weakening of some security properties. Implementers who choose to include this capability should carefully consider the risks to their infrastructure of using a handshake without PFS. Static secret keys should be rotated regularly.

5. IANA Considerations

This document contains no actions for IANA.

6. Acknowledgements

This modification to TLS was initially suggested by Hugo Krawczyk.

7. Normative References

[draft-ietf-tls-tls13-18]
E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3", draft-ietf-tls-tls13-18 (work in progress), October 2016.

Author's Address

Matthew Green
Cryptography Engineering LLC
4506 Roland Ave
Baltimore, MD 21210
USA

Email: mgreen@cryptographyengineering.com