tls
Internet-Draft
Intended status: Standards Track
Expires: June 8, 2019

D. Gillmor
ACLU

December 5, 2018

TLS clients should reject static Diffie-Hellman
draft-dkg-tls-reject-static-dh-01

Abstract

This draft addresses problematic proposals that contradict the expected security properties of TLS. In particular, the ETSI "Middlebox Security Protocol" standard deliberately weakens the cryptographic guarantees of TLS unilaterally by the server, using static Diffie-Hellman keys where ephemeral keys are expected. Responsible TLS clients should avoid connecting to servers that appear to implement such a specification.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on June 8, 2019.

Table of Contents

1. Introduction

TLS 1.3 [RFC8446] promises strong cryptographic properties for a two-party protocol. These properties are the result of extensive engineering and analysis, and are intended to afford users of TLS baseline expectations of confidentiality, integrity, authentication, as well as more subtle properties like replay resistance and forward secrecy.

[draft-green-tls-static-dh-in-tls13-01] proposed the use of a pseudo-static DH share, and was discussed at length in the IETF TLS working group as a mechanism to modify the security properties of TLS for operations within the "enterprise datacenter". The working group failed to reach consensus on this draft, in large part because of the changes it created to the TLS security model, the relative lack of cryptanalysis those changes have received, and the risks to users on the broader Internet.

[MIDDLEBOX] was recently formalized by ETSI, and offers a very similar mechanism to [draft-green-tls-static-dh-in-tls13-01]. In particular, MIDDLEBOX addresses none of the concerns raised during the earlier discussion, and is not fit for the goals of TLS.

This document discusses how responsible TLS clients can avoid the risks inherent in such a design, by refusing connections to peers that implement it.

1.1. Key Words

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this docuoment are to be interpreted as described in [RFC2119].

2. Problems with static DH

[MIDDLEBOX] proposes the use of static Diffie-Hellman keys where TLS expects ephemeral Diffie-Hellman keys. Furthermore, it encourages the sharing of those secret keys with third parties ("middleboxes"). This section documents some of the known problems with this design.

2.1. Limited cryptanalysis

TLS 1.3 as specified has been subject to a substantial amount of cryptanalysis, including formal methods that provide security guarantees. Much of that cryptanalysis takes as a given that the ephemeral DH keys are never re-used. Deliberately re-using DH keys invalidates some of this cryptanalysis, and discards the formal guarantees provided.

2.2. Lack of forward secrecy

Standard ephemeral Diffie-Hellman key exchange permits simple forward secrecy by means of each peer discarding the secrets used to establish the session. Reusing a DH key requires retention of the key, which means that the expected forward secrecy properties are lost.

2.3. Confidentiality violation by middleboxes

A Middlebox which has access to the DH key of a given session can read the contents of the messages in that session by deriving the client_application_traffic_secret and server_application_traffic_secret and using it to decrypt ApplicationData messages. This appears to be the stated goal of [MIDDLEBOX] but typical TLS clients unwittingly connecting to such a

server may still expect confidentiality against third party eavesdropping. This implementation violates that expectation.

2.4. Message tampering by middleboxes

A Middlebox which has access to the DH key of a given session can derive all necessary secrets of the session, and is capable of modifying messages in flight without detection by either peer. This violates the integrity guarantees of TLS.

2.5. Session resumption by middleboxes

A middlebox with access to the DH key of a given session can derive the resumption_master_secret, and can also view any NewSessionTicket messages sent by the server. The middlebox can use that information to subsequently resume the client's old session. The middlebox can also replay any application-layer data that the server might use to establish client identity (e.g. passwords, HTTP cookies, or other bearer tokens).

Since many TLS servers associate client identity with a TLS session and/or application-layer bearer tokens, this effectively allows the middlebox to impersonate the client. This violates expectations of authenticity (because the server does not know whether a resuming client is really the expected client) and replay resistance (because the middlebox can replay any application layer data sent by the client to the server without the client's knowledge).

2.6. Static DH implementations are error-prone

Implementations of static DH schemes are known to be difficult to implement correctly. See for example [invalid-curves-TLS-ECDH]. Proposals of this nature are likely to introduce new forms of implementation error that would be avoided by standard implementations.

3. Mitigations against static DH

Given the concerns raised in Section 2, responsible TLS clients that want to provide the standard TLS guarantees need to implement clear mitigations against risky peers. This section documents useful mitigations.

3.1. TLS Clients MUST Reject server certificates marked for use with static DH

[MIDDLEBOX] suggests that most servers using the designated scheme will use a certificate with so-called "VisibilityInformation" stored in the "subjectAltName" X.509v3 extension (see [RFC5280]), as an "otherName" field with a specific "type-id" of 0.4.0.3523.3.1.

```
0.4.0.3523.3.1
{ itu-t(0)
  identified-organization(4)
  etsi(0)
  msp(3523)
  etls(3)
  visibility(1) }
```

Figure 1: OID of VisibilityInformation `type-id`

A TLS client that receives a Certificate message from the server where the end entity certificate contains any such element in its "subjectAltName" MUST terminate the TLS connection with a fatal "bad_certificate" alert.

3.2. Client detection and rejection of static DH

Annex A of [MIDDLEBOX] suggests that some servers may use pseudo-static Diffie-Hellman without this "subjectAltName" in their certificate.

To defend against leakage from these servers, responsible TLS clients that can afford to keep state SHOULD keep track of the DH shares sent by the server over the course of multiple connections.

If the TLS client notices that it has been offered the same DH share more than once, it SHOULD terminate the TLS connection upon handshake completion with a fatal "decrypt_error" alert.

3.3. Servers MUST avoid accidental DHE share reuse

Given the concerns in Section 2 and the necessary client mitigations in the subsections above, servers need to avoid giving the appearance of using non-ephemeral DH. Servers MUST NOT reuse ephemeral DH shares.

4. Security Considerations

This entire document is an attempt to address security considerations associated with the use of static Diffie-Hellman keys in TLS where ephemeral Diffie-Hellman keys are expected.

5. Privacy Considerations

5.1. Timing of rejection for detecting DH reuse

Clients that are not careful with timing may introduce a minor linkability concern when implementing the mitigation described in Section 3.2.

Consider a network adversary with the following capabilities:

- can observe some connections
- can actively interfere with other connections
- is willing to cause connection failures in order to link client sessions

Such an adversary may be able to identify a TLS client of a standard TLS server across different connections by:

- observing a successul connection, recording the server's "server_share" value in the "key_share" extension to "ServerHello"
- interfering sith subsequent connections to the same server from unknown clients
- each interference re-uses the server's previously-offered "server_share" value.

If the client rejects this repeated share early (e.g upon receipt of the "ServerHello", but before the handshake completes), then the network adversary can re-identify the client as being the one that saw the share recently.

Note that this linkability attack is mitigated by waiting until handshake completion to reject the server's offer, since a normal network adversary does not know the server's credentials, so it will not be able to complete the handshake legitimately. So rejection of the connection at end of handshake will not allow the server to distinguish the specific client from any other TLS client.

6. IANA Considerations

There are no IANA considerations for this document.

7. Acknowledgements

Thanks to numerous commenters on the tls@ietf.org mailing who explained why using static DH presents a risk to TLS users.

8. References

8.1. Normative References

[MIDDLEBOX] European Telecommunications Standards Institute, "Middlebox Security Protocol; Part 3: Profile for enterprise network and data centre access control", ETSI TS 103 523-3, October 2018.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

8.2. Informative References

[draft-green-tls-static-dh-in-tls13-01] Green, M., Droms, R., Housley, R., Turner, P., and S. Fenter, "Data Center use of Static Diffie-Hellman in TLS 1.3", July 2017.

[invalid-curves-TLS-ECDH] Jager, T., Schwenk, J., and J. Somorovsky, "Practical Invalid Curve Attacks on TLS-ECDH", September 2015.

Author's Address

Daniel Kahn Gillmor
ACLU

Email: dkg@fifthhorseman.net