

Network Working Group
Internet-Draft
Updates: [6698](#), [7250](#), [7671](#) (if approved)
Intended status: Informational
Expires: April 12, 2017

R. Barnes
M. Thomson
E. Rescorla
Mozilla
October 9, 2016

Unknown Key-Share Attacks on DNS-based Authentications of Named Entities
(DANE)
[draft-barnes-dane-uks-00](#)

Abstract

Unknown key-share attacks are a class of attacks that allow an attacker to deceive one peer of a secure communication as to the identity of the remote peer. When used with traditional, PKI-based authentication, TLS-based applications are generally safe from unknown key-share attacks. DNS-based Authentication of Named Entities (DANE), however, proposes that applications perform a different set of checks as part of authenticating a TLS connection. As a result, DANE as currently specified is likely to lead to unknown key-share attacks when clients support DANE for authentication. We describe these risks and some simple mitigations.

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 <http://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 April 12, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents

Internet-Draft

DANE UKS

October 2016

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Attack Synopsis	3
2.1.	Attack Example	3
3.	Mitigations	5
4.	IANA Considerations	7
5.	Security Considerations	7
6.	References	7
6.1.	Normative References	7
6.2.	Informative References	8
Appendix A.	Acknowledgements	9
	Authors' Addresses	9

[1.](#) Introduction

TLS is very widely used to secure application protocols, and in particular to authenticate the use of domain names for TLS servers. Traditionally, TLS has authenticated a server's use of a domain name by having the server present a certificate containing that name, then having the client verify that the certificate attests to the name of the server to which it was trying to connect (in addition to verifying that the certificate is issued by a trusted authority) [[RFC5280](#)] [[RFC6125](#)].

DNS-based Authentication of Named Entities (DANE) makes modifications to this process in order to accommodate the use of DNSSEC-signed assertions acquired outside of TLS instead of certificates provided in TLS [[RFC6698](#)]. This change, together with some recommended changes to TLS usage and operational practices, make it possible for an attacker to mount unknown key-share attacks against a TLS client that supports DANE.

In this document, we describe how unknown key-share attacks arise when a client supports DANE in the manner recommended by the DANE

specifications, and we propose some changes to DANE that remove these risks.

[2.](#) Attack Synopsis

In an unknown key-share attack [[UKS](#)], a malicious participant in a protocol claims to control a key that is in reality controlled by some other actor. The victim client will believe that he is talking to the attacker, when in reality he is talking to the victim server.

While this attack may sound less severe than attacks that let the attacker claim an identity that is not his own, it can be used to subvert identity-based access controls in the same way as an impersonation attack. For example, a malicious web site could use an unknown key-share attack to make a cross-origin request appear to be same-origin, circumventing security checks on cross-origin requests [[W3C.CR-cors-20130129](#)].

TLS with PKI-based authentication is not vulnerable to unknown key-share attacks because the server explicitly states its intended identity in its certificate and the client verifies that the server's asserted identity matches the client's intent.

When the client acquires DANE information out of band with respect to TLS, it risks exposing itself to unknown key-share attacks if it takes some additional steps recommended by the DANE specifications [[RFC7671](#)] [[RFC7250](#)].

- o If the client does not verify the identity in the server's certificate (as recommended in [Section 5.1 of \[RFC7671\]](#)), then an attacker can induce the client to accept an unintended identity for the server.
- o If the client allows the use of raw public keys in TLS [[RFC7250](#)], then it will not receive any indication of the server's identity in the TLS channel, and is thus unable to check that the server's identity is as intended.

[2.1.](#) Attack Example

In the natural version of this attack, the attacker convinces a client that it has a TLS connection to `attack.example.com` (operated by the attacker) when in reality, it has a TLS connection to `victim.example.org` (operated by some innocent third party). `victim.example.org` need not even be using DANE; it may have a ordinary Web PKI certificate. In order to mount this attack, the attacker provisions a TLSA record for `attack.example.com` authorizing `victim.example.com`'s public key.

Client	<code>attack.example.com</code>	<code>victim.example.org</code>
		pubkey=P
<code>_-443._tcp.attack.example.com TLSA?--></code>		
<code><-----TLSA usage=3 key=P-----</code>		
<code><=====TLS=====</code>	<code><=====TLS=====</code>	

When the client connects to `attack.example.com`, the attacker forwards the TLS messages to `victim.example.org`, which responds with its ordinary certificate. With a non-DANE TLS connection, this would be detected by the [\[RFC2818\]](#) or [\[RFC6125\]](#) certificate checks, but [\[RFC7671\]](#) specifically instructs the client not to look at the name in the certificate when DANE is in use. Instead, because since the public key matches the TLSA record for `attack.example.com`, the client accepts the connection as coming from `attack.example.com` - even though it's actually to `victim.example.org`.

Depending on the application being run over TLS, this attack can lead to different application-layer vulnerabilities. For example, if the client uses the same TLS client authentication with both servers, the attacker can convince the victim client to dereference a link authorizing some action on the victim server, for instance transferring money from the victim client to the attacker.

With a little more sophistication, the attacker can use this type of attack to violate firewall restrictions. Consider the case where the victim client and the victim server are behind the same firewall but the victim server is unreachable to the attacker. The attacker can exploit the client to recover content from the victim server by

combining this UKS attack with a DNS rebinding attack.

Client	attack.example.com	victim.example.org
IP=192.0.2.2	IP=198.51.100.1	IP=192.0.2.1
		pubkey=P
-_443._tcp.attack.example.com TLSA?-->		
<-----TLSA usage=3 key=P-----		
----- attack.example.com A? ----->		
<----- 192.0.2.1 -----		
<=====TLS=====>		

As before, the client connects to victim.example.org, thinking it is attack.example.com, with the result that any data retrieved is same origin to attack.example.com and therefore is accessible to script from the attacker. This attack was already possible with HTTP resources, but the UKS described here extends it to HTTPS resources.

There are several subtleties to note about this attack. First, it requires the attacker to provide the client with two IP addresses in sequence; first its true IP address (so it can retrieve the attacker's page), not shown, and then the victim server's IP address so that the client can contact the victim. This is known as DNS rebinding. Second, the attacker must somehow retrieve the victim server's public key (because it cannot contact the server directly). One possible way to do this is through the Certificate Transparency [[RFC6962](#)] logs.

[3.](#) Mitigations

At a high level, the mitigation to these attacks is to ensure that when two peers negotiate a secure connection, they agree not just on what public key the server is using, but also what name.

For TLS, this means that the server MUST assert some intended identity (or identities) by including that identity under a signature with its private key. In practice, there are two ways that this can happen. Either the DANE record can contain a self-signed EE certificate containing the identity, or the server can present a certificate in the handshake that contains the name, where it is

transitively authenticated via the Finished MAC (and the CertificateVerify in TLS 1.3 [[I-D.ietf-tls-tls13](#)]).

In order to avoid vulnerability to unknown key-share attacks, then, TLS clients MUST verify that the server's name appears in one of these two places:

- o Even when using DANE, TLS clients MUST verify that the certificate presented by the server represents the name they expect to connect to [[RFC6125](#)].
- o End entity certificates asserted through DANE (usage=3, selector=0) MUST contain the name being authenticated.
- o When using a full EE certificate provided directly in a TLSA record (usage=3, selector=0, match=0), clients MUST verify that the certificate represents the name they expect to connect to. If so, the client MAY accept the use of raw public keys in the resulting TLS connection. When raw public keys are used in TLS, the client MUST verify that the EE certificate presented in the TLSA record is validly self-signed.
- * It is only strictly necessary for the client to verify that the EE certificate is correctly self-signed when the certificate is asserted through DANE and raw public keys are used in the TLS handshake. When the certificate is presented in the handshake,

the name is authenticated by the Finished MAC or CertificateVerify signature (as noted above), so the client only needs to check that the name is correct.

- o When using a public key asserted through DANE (usage=3, selector=1) the server MUST NOT accept the use of raw public keys.
- o In general, TLS clients MUST NOT use raw public keys in TLS unless the client is identifying the server by its public key directly, as opposed to a name.

(Note that TLSA usages 0 and 1 are inherently not vulnerable to unknown key-share attacks, since they are added checks on top of the normal PKI-based authentication.)

The following table summarizes the above requirements for when raw public keys may be used and where the server's name must appear. "MAY*" indicates that the client MAY use raw public keys, but needs to perform some additional checks.

Usage	Selector	Match	Raw key?	Name must appear...
CA(2)	*	*	n/a	In TLS EE
EE(3)	Full(0)	Exact(0)	MAY*	In TLSA or TLS EE
EE(3)	Full(0)	Hash(1/2)	MUST NOT	In TLS EE
EE(3)	SPKI(1)	*	MUST NOT	In TLS EE

The risk of unknown key-share attacks can also be removed by carrying DANE records in the TLS handshake, as suggested in [\[I-D.ietf-tls-dnssec-chain-extension\]](#). In this case, the client MUST verify that the name for which DANE information is provided is the name it intended to connect to.

The directionality of these mitigations is important (server asserts; client verifies). One might think that the opposite order would also work, i.e., for the client to send a desired identity (e.g., in Server Name Indication [\[RFC6066\]](#) or the HTTP Host header field [\[RFC7230\]](#)) and the server to verify it before accepting the handshake. However, servers today display a wide variety of behaviors when presented with unknown SNI values (as would happen during an unknown key share attack). While some fail safely, some reroute to a default hostname. Thus, it is not possible for the client to ensure that the server would fail safe.

In the longer term, DANE's susceptibility to unknown key-share attacks could also be mitigated with a re-design of TLSA records themselves. If DANE records included (1) the names being vouched for, and (2) a signature by the key pair being asserted over the contents of the record, then DANE would effectively always be in the "EE / Full / Exact" case above, since the DANE record would have the same semantics as a self-signed certificate (at least in the ways that matter here). Then it would be safe to use all DANE cases with

raw public keys, since no name checks would need to be done at the TLS layer.

[4.](#) IANA Considerations

This document makes no request of IANA.

[5.](#) Security Considerations

This section intentionally left blank.

[6.](#) References

[6.1.](#) Normative References

- [I-D.ietf-tls-tls13]
Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", [draft-ietf-tls-tls13-16](#) (work in progress), September 2016.
- [RFC2818] Rescorla, E., "HTTP Over TLS", [RFC 2818](#), DOI 10.17487/RFC2818, May 2000, <<http://www.rfc-editor.org/info/rfc2818>>.
- [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, <<http://www.rfc-editor.org/info/rfc5280>>.
- [RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", [RFC 6125](#), DOI 10.17487/RFC6125, March 2011, <<http://www.rfc-editor.org/info/rfc6125>>.

of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", [RFC 6698](#), DOI 10.17487/RFC6698, August 2012, <<http://www.rfc-editor.org/info/rfc6698>>.

- [RFC6962] Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", [RFC 6962](#), DOI 10.17487/RFC6962, June 2013, <<http://www.rfc-editor.org/info/rfc6962>>.
- [RFC7250] Wouters, P., Ed., Tschofenig, H., Ed., Gilmore, J., Weiler, S., and T. Kivinen, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", [RFC 7250](#), DOI 10.17487/RFC7250, June 2014, <<http://www.rfc-editor.org/info/rfc7250>>.
- [RFC7671] Dukhovni, V. and W. Hardaker, "The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operational Guidance", [RFC 7671](#), DOI 10.17487/RFC7671, October 2015, <<http://www.rfc-editor.org/info/rfc7671>>.
- [UKS] Blake-Wilson, S. and A. Menezes, "Unknown Key-Share Attacks on the Station-to-Station (STS) Protocol", Lecture Notes in Computer Science 1560, Springer, pp. 154-170 , 1999.

[6.2.](#) Informative References

- [I-D.ietf-tls-dnssec-chain-extension] Shore, M., Barnes, R., Huque, S., and W. Toorop, "A DANE Record and DNSSEC Authentication Chain Extension for TLS", [draft-ietf-tls-dnssec-chain-extension-01](#) (work in progress), July 2016.
- [RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", [RFC 6066](#), DOI 10.17487/RFC6066, January 2011, <<http://www.rfc-editor.org/info/rfc6066>>.
- [RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", [RFC 7230](#), DOI 10.17487/RFC7230, June 2014, <<http://www.rfc-editor.org/info/rfc7230>>.
- [W3C.CR-cors-20130129] Kesteren, A., "Cross-Origin Resource Sharing", World Wide Web Consortium CR CR-cors-20130129, January 2013, <<http://www.w3.org/TR/2013/CR-cors-20130129>>.

[Appendix A](#). Acknowledgements

The considerations in this document are largely based on Martin Thomson and Eric Rescorla's work with Karthik Bhargavan on the analogous problem in DTLS-SRTP.

Authors' Addresses

Richard Barnes
Mozilla

Email: rlb@ipv.sx

Martin Thomson
Mozilla

Email: martin.thomson@gmail.com

Eric Rescorla
Mozilla

Email: ekr@rftm.com

