Independent Submission
Internet-Draft
Intended status: Standards Track
Expires: May 24, 2015

C. Mehner
USAA

November 20, 2014

# HTTP DANE Validation Assertion
# draft-cem-dane-assertion-00

Abstract

This document defines a new HTTP header that allows web host operators to instruct user agents to remember the hosts' request for DANE (DNS-Based Authentication of Named Entities) validation over a period of time. During that time, UAs will require that the host presents a certificate chain that will authenticate the Transport Layer Security connection using DANE. By having hosts explicitly state that they have adopted DANE, UAs will only expend resources attempting DANE validation on hosts that request it. Comments are solicited and should be addressed to the author

## 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 May 24, 2015.

## Copyright Notice

Table of Contents



## 1. Introduction

This document defines a new HTTP header that enables user agents (UAs) a web host to know which hosts upon which they should perform a DANE [RFC6698] validation. This is called an "HTTP DANE Validation Assertion" (HDVA). The goal of this proposal is to raise the adoption of DANE in web hosts by addressing the cost of attempting DANE Validation on every host via a mechanism that allows web hosts to declare that they use DANE. Using this header also can give hosts the functionality of HTTP-based public key pinning [I-D.ietf-websec-key-pinning] while gaining the greater flexibility of DANE.

UAs performing DANE validation on every HTTPS connection will not benefit from this header, however conformant UAs will use DANE on connections subsequent to the initial time the host is noted. Those hosts will not be able to detect and thwart a MITM attacking the UA's first connection to the host. However, the requirement that the MITM provide an X.509 certificate chain that can pass the UA's validation requirements without error (UAs are assumed to use either [RFC5280] or [RFC6698] to verify cryptographic identity) or control over a DNS-SEC zone mitigates this risk somewhat.

### 1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## 2. Server and Client Behavior

### 2.1. Response Header Field Syntax

The DANE-Validation HTTP response header field (DVA header field) indicates to a UA that it should perform DANE Validation ([RFC6698]) in regards to the host emitting the response message containing this header field.

Figure 1 describes the syntax (Augmented Backus-Naur Form) of the header fields, using the grammar defined in [RFC5234] and the rules defined in Section 3.2 of [RFC7230].

```
DANE-Validation-Directives = directive *( OWS ";" OWS directive )

directive             = directive-name [ "=" directive-value ]
directive-name        = token
directive-value       = token
                      / quoted-string
```

Figure 1: HDVA Header Syntax

Optional white space (OWS) is used as defined in Section 3.2.3 of [RFC7230]. Token and quoted-string are used as defined in Section 3.2.6 of [RFC7230].

The directives defined in this specification are described below. The overall requirements for directives are:

1. The order of appearance of directives is not significant.

2. A given directive MUST NOT appear more than once in a given header field. Directives are either optional or required, as stipulated in their definitions.

3. Directive names are case-insensitive.

4. UAs MUST ignore any header fields containing directives, or other header field value data, that do not conform to the syntax defined in this specification. In particular, UAs must not attempt to fix malformed header fields.

5. If a header field contains any directive(s) the UA does not recognize, the UA MUST ignore those directives.

6. If the DVA header field otherwise satisfies the above requirements (1 through 5), the UA MUST process the directives it recognizes.

Additional directives extending the semantic functionality of the header fields can be defined in other specifications. The first such specification will need to define a registry for such directives. Such future directives will be ignored by UAs implementing only this specification, as well as by generally non-conforming UAs.

When a connection passes DANE Validation after noting the DVA header, the host becomes a Known DANE Host.

### 2.1.1. The max-age Directive

The REQUIRED "max-age" directive specifies the number of seconds, after the reception of the DVA header field, during which the UA SHOULD regard the host (from whom the message was received) as a Known DANE Host.

The syntax of the max-age directive's REQUIRED value (after quoted-string unescaping, if necessary) is defined as:

```
max-age-value = delta-seconds
delta-seconds = 1*DIGIT
```

Figure 2: max-age Value Syntax

delta-seconds is used as defined in [RFC7234], Section 1.2.1.

See Section 2.3.2 for limitations on the range of values for max-age.

### 2.1.2. The includeSubDomains Directive

The OPTIONAL includeSubDomains directive is a valueless directive that, if present (i.e., it is "asserted"), signals to the UA that the DANE Policy applies to this DANE Host as well as any subdomains of the host's domain name.

### 2.1.3. The required Directive

The OPTIONAL required directive is a valueless directive that, if present (i.e., it is "asserted"), signals to the UA that it MUST perform a DANE validation and if no DANE information is received (via network lookup or cache) the UA MUST NOT start a TLS connection or it MUST abort the TLS handshake. Because Section 4.1 of [RFC6698] allows fallback from DANE to PKIX, this directive is in place for Host operators to force both PKIX and DANE validation to take place to provide the additional protection on the connection.

### 2.1.4. Examples

The HDVA header field below stipulates that the HDVA Policy is to remain in effect for one year (there are approximately 31536000 seconds in a year), and the policy applies only to the domain of the HDVA Host issuing it:

```
DANE-Validation: max-age=31536000
```

The HDVA header field below stipulates that the HDVA Policy is to remain in effect for approximately six months and that the policy applies to the domain of the issuing HDVA Host and all of its subdomains:

```
DANE-Validation: max-age=15768000 ; includeSubDomains
```

The max-age directive value can optionally be quoted:

```
DANE-Validation: max-age="31536000"
```

The HDVA header field below indicates that the UA must delete the entire HDVA Policy associated with the HDVA Host that sent the header field:

```
DANE-Validation: max-age=0
```

The HDVA header field below has exactly the same effect as the one immediately above because the includeSubDomains directive's presence in the HDVA header field is ignored when max-age is zero:

DANE-Validation: max-age=0; includeSubDomains

The HDVA header field below states that if the UA MUST receive and validate DANE information, and if it does not it MUST close the connection:

DANE-Validation: max-age=15768000; required

## 2.2. Server Processing Model

This section describes the processing model that DANE Hosts implement. The model has 2 parts: (1) the processing rules for HTTP request messages received over a secure transport (e.g. authenticated, non-anonymous TLS); and (2) the processing rules for HTTP request messages received over non-secure transports, such as TCP.

### 2.2.1. HTTP-over-Secure-Transport Request Type

When replying to an HTTP request that was conveyed over a secure transport, a DANE Host SHOULD include in its response exactly one DVA header field and MUST satisfy the grammar specified in Section 2.1.

Establishing a given host as a Known DANE Host, in the context of a given UA, is accomplished as follows:

1. Over the HTTP protocol running over secure transport, by correctly returning (per this specification) at least one valid DVA header field to the UA.

2. Through other mechanisms, such as a client-side pre-loaded Known DANE Host List.

### 2.2.2. HTTP Request Type

DANE Hosts MAY include the DVA header field in HTTP redirects conveyed over non-secure transport. Hosts may choose to do this if they wish to operate using the TLSA usages of DANE-EE or DANE-TA (as defined in [RFC7218].

## 2.3. User Agent Processing Model

The UA processing model relies on parsing domain names. Note that internationalized domain names SHALL be canonicalized according to the scheme in Section 10 of [RFC6797].

### 2.3.1. DANE-Validation Response Header Field Processing

If the UA receives, over a secure transport, an HTTP response that includes a DVA header field conforming to the grammar specified in Section 2.1, and there are no underlying secure transport errors or warnings (see Section 2.4), the UA MUST either:

- Note the host as a Known DANE Host if it is not already so noted (see Section 2.3.2)

or

- Update the UA's cached information for the Known DANE Host if any of the max-age, includeSubDomains, or required header field value directives convey information different from that already maintained by the UA.

The max-age value is essentially a "time to live" value relative to the time of the most recent observation of the DVA header field. If the max-age header field value token has a value of 0, the UA MUST remove its cached DANE Policy information (including the includeSubDomains directive, if asserted) if the DANE Host is Known, or, MUST NOT note this DANE Host if it is not yet Known.

If a UA receives more than one DVA header field in an HTTP response message over secure transport, then the UA MUST process only the first DVA header field.

If the UA receives the HTTP response over non-secure transport it MUST NOT note the host as a DVA host. To allow the validation of DANE-EE and DANE-TA TLSA usages, a UA MAY accept DVA headers as a 'hint' to perform DANE Validation on the connection.

If the DVA header is not a Valid DANE Header (see Section 2.4), the UA MUST ignore any present DVA header field(s). The UA MUST ignore any DVA header fields not conforming to the grammar specified in Section 2.1.

### 2.3.2. Noting a DANE Host - Storage Model

The Effective Date of a Known DANE Host is the time that the UA observed a Valid DANE Header for the host. The Effective Expiration Date of a Known DANE Host is the Effective Date plus the max-age. A Known DANE Host is "expired" if the Effective Expiration Date refers to a date in the past. The UA MUST ignore any expired Known DANE Hosts in its cache.

For example, if a UA is beginning to perform DANE Validation for a Known DANE Host and finds that the cached information for the host indicates an Effective Expiration Date in the past, the UA MUST NOT continue with DANE Validation for the host, and must consider the host to no longer be a Known DANE Host.

Known DANE Hosts are identified only by domain names, and never IP addresses. If the substring matching the host production from the Request-URI (of the message to which the host responded) syntactically matches the IP-literal or IPv4address productions from Section 3.2.2 of [RFC3986], then the UA MUST NOT note this host as a Known DANE Host.

Otherwise, if the substring does not congruently match an existing Known DANE Host's domain name, per the matching procedure specified in Section 8.2 of [RFC6797], then the UA MUST add this host to the Known DANE Host cache. The UA caches the following information:

- the DANE Host's domain name
- the Effective Expiration Date, or enough information to calculate it (the Effective Date and the value of the max-age directive)
- whether or not the includeSubDomains directive is asserted
- whether or not the required directive is asserted

If any other metadata from optional or future DVA header directives are present in the Valid DANE Header, and the UA understands them, the UA MAY note them as well.

UAs MAY set an upper limit on the value of max-age, so that UAs that have noted erroneous DANE Validation Assertions (whether by accident or due to attack) have some chance of aging out over time. If the server sets a max-age greater than the UA's upper limit, the UA MAY behave as if the server set the max-age to the UA's upper limit. For example, if the UA caps max-age at 5184000 seconds (60 days), and a DANE Host sets a max-age directive of 90 days in its Valid DANE Header, the UA MAY behave as if the max-age were effectively 60 days. (One way to achieve this behavior is for the UA to simply store a value of 60 days instead of the 90 day value provided by the DANE Host.) For UA implementation guidance on how to select a maximum max-age, see Section 3.1.

The UA MUST NOT modify any metadata of any superdomain matched Known DANE Host.

### 2.3.3. HTTP-Equiv <Meta> Element Attribute

UAs MUST NOT heed http-equiv="DANE-Validation" attribute settings on <meta> elements [W3C.REC-html401-19991224] in received content.

## 2.4. Noting DANE Hosts

Upon receipt of the DVA response header field, the UA notes the host as a Known DANE Host, storing the Host and associated directives in non-volatile storage (for example, along with the HSTS or HPKP metadata). The associated directives are collectively known as DANE Metadata.

The UA MUST note the Host as a DANE Host if and only if it received the DVA response header field over an error-free TLS connection. If the host is a DANE Host, this includes the validation added in Section 2.5.

If the DVA response header field does not meet the above criteria, the UA MUST NOT note the host as a DANE Host. A DVA response header field that meets all these criteria is known as a Valid DANE Header.

Whenever a UA receives a Valid DANE Header, it MUST set its DANE Metadata to the exact Effective Expiration Date (computed from max-age), and note any associated directives if present.

For forward compatibility, the UA MUST ignore any unrecognized DVA header directives, while still processing those directives it does recognize. Section 2.1 specifies the directives max-age, includeSubDomains, and required but future specifications and implementations might use additional directives.

## 2.5. Validating DANE Connections

When a UA connects to a Known DANE Host using a TLS connection, the UA SHOULD perform a DANE Validation on the Host, as soon as possible (e.g. immediately after receiving the Server Certificate message). A DANE Validation follows the procedure for comparing a certificate association from a TLSA record and a certificate from the TLS handshake as defined in [RFC6698].)

If no TLSA records were received for evaluation and the host's DANE Metadata includes an asserted required directive, the UA MUST terminate the connection.

If the connection has no errors in the DANE Validation, the UA will determine whether to apply any additional correctness checks such as

Pin Validation [I-D.ietf-websec-key-pinning], or applying an HTTP Strict Transport Security Policy [RFC6797].

## 2.6. Interactions With Preloaded DANE Host Lists

UAs MAY choose to implement additional sources of DANE Host information, such as through built-in lists of host information. Such UAs should allow users to override such additional sources, including disabling them from consideration.

The effective policy for a Known DANE Host that has both built-in hosts and hosts from previously observed DVA header response fields is implementation-defined.

## 2.7. TLSA Certificate Usages in DANE

HDVA is able to interoperate with UAs that support DANE and those that do not. For Hosts that use PKIX-TA or PKIX-EE certificate validation will occur both with and without DANE. Hosts that expect to use DANE-TA or DANE-EE should not expect to interoperate with UAs that do not support DANE. Conversely, hosts that choose PKIX-TA or PKIX-EE should not expect full interoperation with UAs that do not include a full list of trust authorities.

UAs that choose to accept and validate DVA headers over non-secure transport as a 'hint' to perform a DANE Validation MUST do so in according with Section 2.3.1 and MUST allow DANE-TA and DANE-EE usages for the initial connection and given a successful DANE Validation note the TLS connection as error-free.

# 3. Security Considerations

HTTP DANE-Validation Assertions allow hosts to strongly assert their intention for additional validation of their cryptographic identity. This document concerns the expression, conveyance, and enforcement of the DANE Validation policy.

See [RFC6698] for security considerations relating to a DANE Validation.

This document adds the concept of a required directive that requires the UA to receive TLSA records when communicating with a DANE Host over secure transport. When the host asserts the required directive there is additional risk of an active network attacker blocking DANE information from reaching the UA. This scenario would effectively create a denial of service for the victim of the attack. This is also a concern in some networks, which are configured in such a manner as to effectively block DANE information. If a host chooses

to assert the required directive, they should consider clients that may not be able to get DANE information and consider the associated risks when asserting that directive. Without the required directive, an active network attacker could potentially block DANE information from reaching the victim and force validation of the connection to precede without any DANE information this would circumvent the additional out-of-band checks and rely on the UA's normal cryptographic identity validation, which could allow an attacker to man-in-the-middle the connection with a certificate that would otherwise fail DANE Validation.

To help mitigate this risk UAs SHOULD cache TLSA records as mentioned in Section 8.2 of [RFC6698]. In addition, if a UA does a pre-fetch for IP address, they should also prefetch TLSA records for DANE Hosts. Furthermore, UAs can receive TLSA records through another medium such as TLS extensions, HTTP, or other methods, which may not be blocked or be faster than DNS itself.

To discover attacks on a host that does not assert the required directive, over a network where an attacker is blocking the TLSA records from reaching a UA, the host may also employ the Report-Only directive from [I-D.ietf-websec-key-pinning].

## 3.1. Maximum max-age

As mentioned in Section 2.3.2, UAs MAY cap the max-age value at some upper limit. There is a security trade-off in that low maximum values provide a narrow window of protection for users who visit the Known DANE Host only infrequently, while high maximum values may potentially result in a UA's inability to successfully perform DANE Validation on hosts that assert the required directive should they choose to remove the TLSA record from the domain. Also, if a host has removed the TLSA records, a long max-age would create longer initial connection times while the UA attempts to retrieve a non-existent TLSA record.

There is likely no ideal upper limit to the max-age directive that would satisfy all use cases. However, a value on the order of 60 days (5,184,000 seconds) may be considered a balance between the two competing security concerns.

## 3.2. Using includeSubDomains Safely

It may happen that DANE Hosts whose hostnames share a parent domain use different Valid DVA Headers. If a host whose hostname is a parent domain for another host sets the includeSubDomains directive, the two hosts' DANE policies may conflict with each other. For example, consider two Known DANE Hosts, example.com and

subdomain.example.com. Assume example.com sets a Valid DVA Header such as this:

```
DANE-Validation: max-age=12000; required; includeSubDomains
```

Figure 3: example.com Valid DVA Header

Assume subdomain.example.com sets a Valid DVA Header such as this:

```
DANE-Validation: max-age=12000;
```

Figure 4: subdomain.example.com Valid DVA Header

Assume a UA that has not previously noted either of these hosts as DANE Hosts. If the UA first contacts subdomain.example.com, it will note the host as a DANE Host, and attempt DANE Validation as normal on subsequent connections. If the UA does not receive a TLSA record for subdomain.example.com it will fall back to the UA's normal certificate path validation. If the UA then contacts example.com, it will note the DANE Host and require DANE Validation on future connections.

However, if the UA happened to visit example.com before subdomain.example.com, the UA would, due to example.com's use of the includeSubDomains and required directives, require a valid TLSA record to perform DANE Validation for subdomain.example.com. If such a record was not present or available, the UA will cancel the connection. Thus, depending on the order in which the UA observes the Valid DVA Headers for hosts example.com and subdomain.example.com, DANE Validation might or might not fail for subdomain.example.com, if it cannot receive any TLSA records. This can occur even if the certificate chain the UA receives for subdomain.example.com is perfectly valid.

Thus, DANE Host operators must use the includeSubDomains directive with care. For example, they may choose to use the required directive only on Hosts that do not assert the includeSubDomains directive, those that do not have any child domains, or to only use the required directive on HOSTs whose child domains are all assured to receive TLSA records via TLS extensions or some other pre-arranged means.

### 3.3. Interactions With Cookie Scoping

HTTP cookies [RFC6265] set by a Known DANE Host can be stolen by a network attacker who can forge web and DNS responses so as to cause a client to send the cookies to a phony subdomain of the host. To prevent this, hosts SHOULD set the "secure" attribute and precisely

scope the "domain" attribute on all security-sensitive cookies, such as session cookies. These settings tell the browser that the cookie should only be sent back to the specific host(s) (not any arbitrary subdomain of a given domain), and should only be sent over HTTPS (not HTTP).

## 4. IANA Considerations

IANA is requested to register the response headers described in this document in the "Message Headers" registry ([permanent-headers]), with the following parameters:

- Header Field Names should be "DANE-Validation"
- Protocol should be "http"
- Status should be "standard"
- Reference should be this document

## 5. Usability Considerations

To keep backwards compatibility with non-conforming UAs a host may choose to provide PKIX-TA or PKIX-EE TLSA records combined with the required directive in the DVA Header in order to provide a protection against imposter certificates. When the UA encounters a situation like that where it would prevent the connection from continuing, users will experience a denial of service. It is advisable for UAs to explain the reason why, i.e. that it was impossible to verify the confirmed cryptographic identity of the host.

It is advisable that UAs have a way for users to clear the current list of DANE Hosts, and to allow users to query the current state of DANE Hosts.

## 6. Acknowledgements

Thanks to the websec working group for the Public-Key-Pinning draft, from which this document draws heavily.

## 7. Normative References

[I-D.ietf-websec-key-pinning]
Evans, C., Palmer, C., and R. Sleevi, "Public Key Pinning Extension for HTTP", draft-ietf-websec-key-pinning-21 (work in progress), October 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[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, May 2008.

[RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265, April 2011.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.

[RFC6797] Hodges, J., Jackson, C., and A. Barth, "HTTP Strict Transport Security (HSTS)", RFC 6797, November 2012.

[RFC7218] Gudmundsson, O., "Adding Acronyms to Simplify Conversations about DNS-Based Authentication of Named Entities (DANE)", RFC 7218, April 2014.

[RFC7230] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, June 2014.

[RFC7234] Fielding, R., Nottingham, M., and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Caching", RFC 7234, June 2014.

[W3C.REC-html401-19991224] Raggett, D., Hors, A., and I. Jacobs, "HTML 4.01 Specification", World Wide Web Consortium Recommendation REC-html401-19991224, December 1999, <http://www.w3.org/TR/1999/REC-html401-19991224>.

[permanent-headers] Klyne, G., "Permanent Message Header Field Names", July 2014, <http://www.iana.org/assignments/message-headers/message-headers.xml#perm-headers/>.

Author's Address

Carl Mehner
USAA
9800 Fredericksburg
San Antonio, TX 78288
US

Email: carl.mehner@usaa.com