HTTP Working Group
Internet-Draft
Intended status: Experimental
Expires: February 15, 2018

E. Stark
Google

August 14, 2017

Expect-CT Extension for HTTP
draft-ietf-httpbis-expect-ct-02

Abstract

This document defines a new HTTP header, named Expect-CT, that allows web host operators to instruct user agents to expect valid Signed Certificate Timestamps (SCTs) to be served on connections to these hosts. When configured in enforcement mode, user agents (UAs) will remember that hosts expect SCTs and will refuse connections that do not conform to the UA's Certificate Transparency policy. When configured in report-only mode, UAs will report the lack of valid SCTs to a URI configured by the host, but will allow the connection. By turning on Expect-CT, web host operators can discover misconfigurations in their Certificate Transparency deployments and ensure that misissued certificates accepted by UAs are discoverable in Certificate Transparency logs.

Note to Readers

Discussion of this draft takes place on the HTTP working group mailing list (ietf-http-wg@w3.org), which is archived at https://lists.w3.org/Archives/Public/ietf-http-wg/.

Working Group information can be found at http://httpwg.github.io/; source code and issues list for this draft can be found at https://github.com/httpwg/http-extensions/labels/expect-ct.

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 February 15, 2018.

Copyright Notice

Table of Contents

1. Introduction

This document defines a new HTTP header that enables UAs to identify web hosts that expect the presence of Signed Certificate Timestamps (SCTs) [I-D.ietf-trans-rfc6962-bis] in future Transport Layer Security (TLS) [RFC5246] connections.

Web hosts that serve the Expect-CT HTTP header are noted by the UA as Known Expect-CT Hosts. The UA evaluates each connection to a Known Expect-CT Host for compliance with the UA's Certificate Transparency (CT) Policy. If the connection violates the CT Policy, the UA sends a report to a URI configured by the Expect-CT Host and/or fails the connection, depending on the configuration that the Expect-CT Host has chosen.

If misconfigured, Expect-CT can cause unwanted connection failures (for example, if a host deploys Expect-CT but then switches to a legitimate certificate that is not logged in Certificate Transparency logs, or if a web host operator believes their certificate to conform to all UAs' CT policies but is mistaken). Web host operators are advised to deploy Expect-CT with caution, by using the reporting feature and gradually increasing the interval where the UA remembers the host as a Known Expect-CT Host. These precautions can help web host operators gain confidence that their Expect-CT deployment is not causing unwanted connection failures.

Expect-CT is a trust-on-first-use (TOFU) mechanism. The first time a UA connects to a host, it lacks the information necessary to require SCTs for the connection. Thus, the UA will not be able to detect and thwart an attack on the UA's first connection to the host. Still,

Expect-CT provides value by 1) allowing UAs to detect the use of unlogged certificates after the initial communication, and 2) allowing web hosts to be confident that UAs are only trusting publicly-auditable certificates.

1.1. Requirements Language

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

1.2. Terminology

Terminology is defined in this section.

Certificate Transparency Policy is a policy defined by the UA concerning the number, sources, and delivery mechanisms of Signed Certificate Timestamps that are served on TLS connections. The policy defines the properties of a connection that must be met in order for the UA to consider it CT-qualified.

Certificate Transparency Qualified describes a TLS connection for which the UA has determined that a sufficient quantity and quality of Signed Certificate Timestamps have been provided.

CT-qualified See Certificate Transparency Qualified.

CT Policy See Certificate Transparency Policy.

Effective Expect-CT Date is the time at which a UA observed a valid Expect-CT header for a given host.

Expect-CT Host See HTTP Expect-CT Host.

HTTP Expect-CT is the overall name for the combined UA- and server-side security policy defined by this specification.

HTTP Expect-CT Host is a conformant host implementing the HTTP server aspects of HTTP Expect-CT. This means that an Expect-CT Host returns the "Expect-CT" HTTP response header field in its

HTTP response messages sent over secure transport.

Known Expect-CT Host is an Expect-CT Host that the UA has noted as such. See Section 2.3.3 for particulars.

UA is an acronym for "user agent". For the purposes of this specification, a UA is an HTTP client application typically actively manipulated by a user [RFC7230].

Unknown Expect-CT Host is an Expect-CT Host that the UA has not noted.

2. Server and Client Behavior

2.1. Response Header Field Syntax

The "Expect-CT" header field is a new response header defined in this specification. It is used by a server to indicate that UAs should

evaluate connections to the host emitting the header for CT compliance (Section 2.4).

Figure 1 describes the syntax (Augmented Backus-Naur Form) of the header field, using the grammar defined in RFC 5234 [RFC5234] and the rules defined in Section 3.2 of RFC 7230 [RFC7230].

```
Expect-CT           = #expect-ct-directive
expect-ct-directive = directive-name [ "=" directive-value ]
directive-name      = token
directive-value     = token / quoted-string
```

Figure 1: Syntax of the Expect-CT header field

Optional white space ("OWS") is used as defined in Section 3.2.3 of RFC 7230 [RFC7230]. "token" and "quoted-string" are used as defined in Section 3.2.6 of RFC 7230 [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 Expect-CT header field otherwise satisfies the above requirements (1 through 5), the UA MUST process the directives it recognizes.

2.1.1. The report-uri Directive

The OPTIONAL "report-uri" directive indicates the URI to which the UA SHOULD report Expect-CT failures (Section 2.4). The UA POSTs the reports to the given URI as described in Section 3.

The "report-uri" directive is REQUIRED to have a directive value, for which the syntax is defined in Figure 2.

```
report-uri-value = absolute-URI
```

Figure 2: Syntax of the report-uri directive value

"absolute-URI" is defined in Section 4.3 of RFC 3986 [RFC3986].

Hosts may set "report-uri"s that use HTTP or HTTPS. If the scheme in the "report-uri" is one that uses TLS (e.g., HTTPS), UAs MUST check Expect-CT compliance when the host in the "report-uri" is a Known Expect-CT Host; similarly, UAs MUST apply HSTS if the host in the "report-uri" is a Known HSTS Host.

Note that the report-uri need not necessarily be in the same Internet domain or web origin as the host being reported about.

UAs SHOULD make their best effort to report Expect-CT failures to the "report-uri", but they may fail to report in exceptional conditions. For example, if connecting the "report-uri" itself incurs an Expect-CT failure or other certificate validation failure, the UA MUST cancel the connection. Similarly, if Expect-CT Host A sets a "report-uri" referring to Expect-CT Host B, and if B sets a "report-uri" referring to A, and if both hosts fail to comply to the UA's CT Policy, the UA SHOULD detect and break the loop by failing to send reports to and about those hosts.

UAs SHOULD limit the rate at which they send reports. For example, it is unnecessary to send the same report to the same "report-uri" more than once.

2.1.2. The enforce Directive

The OPTIONAL "enforce" directive is a valueless directive that, if present (i.e., it is "asserted"), signals to the UA that compliance to the CT Policy should be enforced (rather than report-only) and that the UA should refuse future connections that violate its CT Policy. When both the "enforce" directive and "report-uri" directive (as defined in Figure 2) are present, the configuration is referred to as an "enforce-and-report" configuration, signalling to the UA both that compliance to the CT Policy should be enforced and that violations should be reported.

2.1.3. The max-age Directive

The "max-age" directive specifies the number of seconds after the reception of the Expect-CT header field during which the UA SHOULD regard the host from whom the message was received as a Known Expect-CT Host.

The "max-age" directive is REQUIRED to be present within an "Expect-

CT" header field. The "max-age" directive is REQUIRED to have a directive value, for which the syntax (after quoted-string unescaping, if necessary) is defined in Figure 3.

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

Figure 3: Syntax of the max-age directive value

"delta-seconds" is used as defined in Section 1.2.1 of RFC 7234 [RFC7234].

2.1.4. Examples

The following examples demonstrate valid Expect-CT response header fields:

```
Expect-CT: max-age=86400,enforce

Expect-CT: max-age=86400, enforce, report-uri="https://foo.example/report"

Expect-CT: max-age=86400,report-uri="https://foo.example/report"
```

Figure 4: Examples of valid Expect-CT response header fields

2.2. Server Processing Model

This section describes the processing model that Expect-CT 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, an Expect-CT Host SHOULD include in its response exactly one Expect-CT header field. The header field MUST satisfy the grammar specified in Section 2.1.

Establishing a given host as an Expect-CT 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 Expect-CT header field to the UA.

2. Through other mechanisms, such as a client-side preloaded Expect-CT Host list.

2.2.2. HTTP Request Type

Expect-CT Hosts SHOULD NOT include the Expect-CT header field in HTTP responses conveyed over non-secure transport. UAs MUST ignore any Expect-CT header received in an HTTP response conveyed over non-secure transport.

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. Expect-CT Header Field Processing

If the UA receives, over a secure transport, an HTTP response that includes an Expect-CT header field conforming to the grammar specified in Section 2.1, the UA MUST evaluate the connection on which the header was received for compliance with the UA's CT Policy, and then process the Expect-CT header field as follows.

If the connection complies with the UA's CT Policy (i.e. the connection is CT-qualified), then the UA MUST either:

- Note the host as a Known Expect-CT Host if it is not already so noted (see Section 2.3.3), or

- Update the UA's cached information for the Known Expect-CT Host if the "enforce", "max-age", or "report-uri" header field value directives convey information different from that already maintained by the UA. If the "max-age" directive has a value of 0, the UA MUST remove its cached Expect-CT information if the host was previously noted as a Known Expect-CT Host, and MUST NOT note this host as a Known Expect-CT Host if it is not already noted.

If the connection does not comply with the UA's CT Policy (i.e. is not CT-qualified), then the UA MUST NOT note this host as a Known Expect-CT Host.

If the header field includes a "report-uri" directive, and the connection does not comply with the UA's CT Policy (i.e. the connection is not CT-qualified), and the UA has not already sent an Expect-CT report for this connection, then the UA SHOULD send a report to the specified "report-uri" as specified in Section 3.

The UA MUST ignore any Expect-CT header field not conforming to the grammar specified in Section 2.1.

2.3.2. HTTP-Equiv <meta> Element Attribute

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

2.3.3. Noting Expect-CT

Upon receipt of the Expect-CT response header field over an error-free TLS connection (including the validation adding in Section 2.4), the UA MUST note the host as a Known Expect-CT Host, storing the host's domain name and its associated Expect-CT directives in non-volatile storage. The domain name and associated Expect-CT directives are collectively known as "Expect-CT metadata".

To note a host as a Known Expect-CT Host, the UA MUST set its Expect-CT metadata given in the most recently received valid Expect-CT header, as specified in Section 2.3.4.

For forward compatibility, the UA MUST ignore any unrecognized Expect-CT header directives, while still processing those directives it does recognize. Section 2.1 specifies the directives "enforce", "max-age", and "report-uri", but future specifications and implementations might use additional directives.

2.3.4. Storage Model

Known Expect-CT 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 Expect-CT Host.

Otherwise, if the substring does not congruently match an existing Known Expect-CT 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 Expect-CT Host cache. The UA caches:

- the Expect-CT Host's domain name,

- whether the "enforce" directive is present

- the Effective Expiration Date, which is the Effective Expect-CT Date plus the value of the "max-age" directive. Alternatively, the UA MAY cache enough information to calculate the Effective Expiration Date.

- the value of the "report-uri" directive, if present.

If any other metadata from optional or future Expect-CT header directives are present in the Expect-CT 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 Expect-CT hosts (whether by accident or due to attack) have some chance of recovering 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 5,184,000 seconds (60 days), and an Expect-CT Host sets a max- age directive of 90 days in its Expect-CT 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 Expect-CT host.)

2.4. Evaluating Expect-CT Connections for CT Compliance

When a UA connects to a Known Expect-CT Host using a TLS connection, if the TLS connection has errors, the UA MUST terminate the connection without allowing the user to proceed anyway. (This behavior is the same as that required by [RFC6797].)

If the connection has no errors, then the UA will apply an additional correctness check: compliance with a CT Policy. A UA should evaluate compliance with its CT Policy whenever connecting to a Known Expect-CT Host, as soon as possible. It is acceptable to skip this CT compliance check for some hosts according to local policy. For example, a UA may disable CT compliance checks for hosts whose validated certificate chain terminates at a user-defined trust anchor, rather than a trust anchor built-in to the UA (or underlying

platform).

An Expect-CT Host is "expired" if the effective expiration date refers to a date in the past. The UA MUST ignore any expired Expect-CT Hosts in its cache and not treat such hosts as Known Expect-CT hosts.

If a connection to a Known CT Host violates the UA's CT policy (i.e. the connection is not CT-qualified), and if the Known Expect-CT

Host's Expect-CT metadata indicates an "enforce" configuration, the UA MUST treat the CT compliance failure as a non-recoverable error.

If a connection to a Known CT Host violates the UA's CT policy, and if the Known Expect-CT Host's Expect-CT metadata includes a "report-uri", the UA SHOULD send an Expect-CT report to that "report-uri" (Section 3).

A UA that has previously noted a host as a Known Expect-CT Host MUST evaluate CT compliance when setting up the TLS session, before beginning an HTTP conversation over the TLS channel.

If the UA does not evaluate CT compliance, e.g. because the user has elected to disable it, or because a presented certificate chain chains up to a user-defined trust anchor, UAs SHOULD NOT send Expect-CT reports.

3. Reporting Expect-CT Failure

When the UA attempts to connect to a Known Expect-CT Host and the connection is not CT-qualified, the UA SHOULD report Expect-CT failures to the "report-uri", if any, in the Known Expect-CT Host's Expect-CT metadata.

When the UA receives an Expect-CT response header field over a connection that is not CT-qualified, if the UA has not already sent an Expect-CT report for this connection, then the UA SHOULD report Expect-CT failures to the configured "report-uri", if any.

3.1. Generating a violation report

To generate a violation report object, the UA constructs a JSON

object with the following keys and values:

- "date-time": the value for this key indicates the time the UA observed the CT compliance failure. The value is a string formatted according to Section 5.6, "Internet Date/Time Format", of [RFC3339].

- "hostname": the value is the hostname to which the UA made the original request that failed the CT compliance check. The value is provided as a string.

- "port": the value is the port to which the UA made the original request that failed the CT compliance check. The value is provided as an integer.

- "effective-expiration-date": the value indicates the Effective Expiration Date (see Section 2.3.4) for the Expect-CT Host that failed the CT compliance check. The value is provided as a string formatted according to Section 5.6, "Internet Date/Time Format", of [RFC3339].

- "served-certificate-chain": the value is the certificate chain as served by the Expect-CT Host during TLS session setup. The value is provided as an array of strings, which MUST appear in the order that the certificates were served; each string in the array is the Privacy-Enhanced Mail (PEM) representation of each X.509 certificate as described in [RFC7468].

- "validated-certificate-chain": the value is the certificate chain as constructed by the UA during certificate chain verification. (This may differ from the value of the "served-certificate-chain" key.) The value is provided as an array of strings, which MUST appear in the order matching the chain that the UA validated; each string in the array is the Privacy-Enhanced Mail (PEM) representation of each X.509 certificate as described in [RFC7468].

- "scts": the value represents the SCTs (if any) that the UA received for the Expect-CT host and their validation statuses. The value is provided as an array of JSON objects. The SCTs may

appear in any order. Each JSON object in the array has the following keys:

* A "version" key, with an integer value. The UA MUST set this value to "1" if the SCT is in the format defined in Section 3.2 of [RFC6962] and "2" if it is in the format defined in Section 4.6 of [I-D.ietf-trans-rfc6962-bis].

* The "status" key, with a string value that the UA MUST set to one of the following values: "unknown" (indicating that the UA does not have or does not trust the public key of the log from which the SCT was issued), "valid" (indicating that the UA successfully validated the SCT as described in Section 5.2 of [RFC6962] or Section 8.2.3 of [I-D.ietf-trans-rfc6962-bis]), or "invalid" (indicating that the SCT validation failed because of, e.g., a bad signature).

* The "source" key, with a string value that indicates from where the UA obtained the SCT, as defined in Section 3 or [RFC6962] and Section 6 of [I-D.ietf-trans-rfc6962-bis]. The UA MUST set the value to one of "tls-extension", "ocsp", or "embedded".

* The "serialized_sct" key, with a string value. If the value of the "version" key is "1", the UA MUST set this value to the base64 encoded [RFC4648] serialized "SignedCertificateTimestamp" structure from Section 3.2 of [RFC6962]. If the value of the "version" key is "2", the UA MUST set this value to the base64 encoded [RFC4648] serialized "TransItem" structure representing the SCT, as defined in Section 4.6 of [I-D.ietf-trans-rfc6962-bis].

3.2. Sending a violation report

The UA SHOULD report an Expect-CT failure when a connection to a Known Expect-CT Host does not comply with the UA's CT Policy and the host's Expect-CT metadata contains a "report-uri". Additionally, the UA SHOULD report an Expect-CT failure when it receives an Expect-CT header field which contains the "report-uri" directive over a connection that does not comply with the UA's CT Policy.

The steps to report an Expect-CT failure are as follows.

1. Prepare a JSON object "report object" with the single key "expect-ct-report", whose value is the result of generating a violation report object as described in Section 3.1.

2. Let "report body" by the JSON stringification of "report object".

3. Let "report-uri" be the value of the "report-uri" directive in the Expect-CT header field.

4. Send an HTTP POST request to "report-uri" with a "Content-Type" header field of "application/expect-ct-report+json", and an entity body consisting of "report body".

The UA MAY perform other operations as part of sending the HTTP POST request, for example sending a CORS preflight as part of [FETCH].

4. Security Considerations

When UAs support the Expect-CT header, it becomes a potential vector for hostile header attacks against site owners. If a site owner uses a certificate issued by a certificate authority which does not embed SCTs nor serve SCTs via OCSP or TLS extension, a malicious server operator or attacker could temporarily reconfigure the host to comply with the UA's CT policy, and add the Expect-CT header in enforcing mode with a long "max-age". Implementing user agents would note this as an Expect-CT Host (see Section 2.3.3). After having done this, the configuration could then be reverted to not comply with the CT policy, prompting failures. Note this scenario would require the

attacker to have substantial control over the infrastructure in question, being able to obtain different certificates, change server software, or act as a man-in-the-middle in connections.

Site operators could themselves only cure this situation by one of: reconfiguring their web server to transmit SCTs using the TLS extension defined in Section 6.5 of [I-D.ietf-trans-rfc6962-bis], obtaining a certificate from an alternative certificate authority which provides SCTs by one of the other methods, or by waiting for the user agents' persisted notation of this as an Expect-CT host to reach its "max-age". User agents may choose to implement mechanisms

for users to cure this situation, as noted in Section 7.

4.1. Maximum max-age

There is a security trade-off in that low maximum values provide a narrow window of protection for users that visit the Known Expect-CT Host only infrequently, while high maximum values might result in a denial of service to a UA in the event of a hostile header attack, or simply an error on the part of the site-owner.

There is probably no ideal maximum for the "max-age" directive. Since Expect-CT is primarily a policy-expansion and investigation technology rather than an end-user protection, a value on the order of 30 days (2,592,000 seconds) may be considered a balance between these competing security concerns.

4.2. Avoiding amplification attacks

Another kind of hostile header attack uses the "report-uri" mechanism on many hosts not currently exposing SCTs as a method to cause a denial-of-service to the host receiving the reports. If some highly-trafficked websites emitted a non-enforcing Expect-CT header with a "report-uri", implementing UAs' reports could flood the reporting host. It is noted in Section 2.1.1 that UAs should limit the rate at which they emit reports, but an attacker may alter the Expect-CT header's fields to induce UAs to submit different reports to different URIs to still cause the same effect.

5. Privacy Considerations

Expect-CT can be used to infer what Certificate Transparency policy is in use, by attempting to retrieve specially-configured websites which pass one user agents' policies but not another's. Note that this consideration is true of UAs which enforce CT policies without Expect-CT as well.

Additionally, reports submitted to the "report-uri" could reveal information to a third party about which webpage is being accessed and by which IP address, by using individual "report-uri" values for individually-tracked pages. This information could be leaked even if

client-side scripting were disabled.

Implementations must store state about Known Expect-CT Hosts, and hence which domains the UA has contacted.

Violation reports, as noted in Section 3, contain information about the certificate chain that has violated the CT policy. In some cases, such as organization-wide compromise of the end-to-end security of TLS, this may include information about the interception tools and design used by the organization that the organization would otherwise prefer not be disclosed.

Because Expect-CT causes remotely-detectable behavior, it's advisable that UAs offer a way for privacy-sensitive users to clear currently noted Expect-CT hosts, and allow users to query the current state of Known Expect-CT Hosts.

6. IANA Considerations

TBD

7. Usability Considerations

When the UA detects a Known Expect-CT Host in violation of the UA's CT Policy, users will experience denials of service. It is advisable for UAs to explain the reason why.

8. Authoring Considerations

8.1. HTTP Header

Expect-CT could be specified as a TLS extension or X.509 certificate extension instead of an HTTP response header. Using an HTTP header as the mechanism for Expect-CT introduces a layering mismatch: for example, the software that terminates TLS and validates Certificate Transparency information might know nothing about HTTP. Nevertheless, an HTTP header was chosen primarily for ease of deployment. In practice, deploying new certificate extensions requires certificate authorities to support them, and new TLS extensions require server software updates, including possibly to servers outside of the site owner's direct control (such as in the case of a third-party CDN). Ease of deployment is a high priority for Expect-CT because it is intended as a temporary transition

mechanism for user agents that are transitioning to universal Certificate Transparency requirements.

9. Changes

9.1. Since -01

- Change SCT reporting format to support both RFC 6962 and 6962-bis SCTs.

9.2. Since -00

- Editorial changes
- Change Content-Type header of reports to 'application/expect-ct-report+json'
- Update header field syntax to match convention (issue #327)
- Reference RFC 6962-bis instead of RFC 6962

10. Normative References

[FETCH] van Kesteren, A., "Fetch", n.d., <https://fetch.spec.whatwg.org/>.

[HTML] Hickson, I., Pieters, S., van Kesteren, A., Jaegenstedt, P., and D. Denicola, "HTML", n.d., <https://html.spec.whatwg.org/>.

[I-D.ietf-trans-rfc6962-bis] Laurie, B., Langley, A., Kasper, E., Messeri, E., and R. Stradling, "Certificate Transparency Version 2.0", draft-ietf-trans-rfc6962-bis-26 (work in progress), July 2017.

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

[RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <http://www.rfc-editor.org/info/rfc3339>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005,

<http://www.rfc-editor.org/info/rfc3986>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <http://www.rfc-editor.org/info/rfc4648>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[RFC6797] Hodges, J., Jackson, C., and A. Barth, "HTTP Strict Transport Security (HSTS)", RFC 6797, DOI 10.17487/RFC6797, November 2012, <http://www.rfc-editor.org/info/rfc6797>.

[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>.

[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>.

[RFC7234] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Caching", RFC 7234, DOI 10.17487/RFC7234, June 2014, <http://www.rfc-editor.org/info/rfc7234>.

[RFC7468] Josefsson, S. and S. Leonard, "Textual Encodings of PKIX, PKCS, and CMS Structures", RFC 7468, DOI 10.17487/RFC7468, April 2015, <http://www.rfc-editor.org/info/rfc7468>.

[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>.

Author's Address

Emily Stark
Google

Email: estark@google.com