Workgroup: TLS
Internet-Draft: draft-ietf-tls-wkech-02
Published: 21 April 2023
Intended Status: Experimental
Expires: 23 October 2023
Authors: S. Farrell R. Salz
Trinity College Dublin Akamai Technologies
B. Schwartz
Google LLC

A well-known URI for publishing ECHConfigList values.

Abstract

We propose use of a well-known URI at which an HTTP origin can inform an authoritative DNS server, or other interested parties, about this origin's Service Bindings, i.e. its "HTTPS" DNS records. These instructions can include Encrypted ClientHello (ECH) configurations, allowing the origin to publish and rotate its own ECH keys.

AUTHORS NOTE: This version proposes changing from the highly ECHConfig specific approach of -00 to a much more generic approach. The authors are seeking feedback from the Working Group as to which of these approaches may be more likely to garner rough consensus. If the WG feel this is worse than -00 we're fine with reverting.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Encrypted ClientHello (ECH) [I-D.ietf-tls-esni] for TLS1.3 [RFC8446] defines a confidentiality mechanism for server names and other ClientHello content in TLS. For many applications, that requires publication of ECHConflgList data structures in the DNS. An ECHConfigList structure contains a list of ECHConfig values. Each ECHConfig value contains the public component of a key pair that will typically be periodically (re-)generated by a web server. Many web infrastructures will have an API that can be used to dynamically update the DNS RR values containing ECHConfigList values. Some deployments however, will not, so web deployments could benefit from a mechanism to use in such cases.

We define such a mechanism here. Note that this is not intended for universal deployment, but rather for cases where the web server doesn't have write access to the relevant zone file (or equivalent). That zone file will eventually include an HTTPS or SVCB RR [I-D.ietf-dnsop-svcb-https] containing an ECHConfigList. This mechanism is extensible to deliver other kinds of information about the origin, but in this specification it only provides the functionality necessary to configure ECH.

We use the term "zone factory" for the entity that does have write access to the zone file. We assume the zone factory (ZF) can also make HTTPS requests to the web server with the ECH keys.

We propose use of a well-known URI [RFC8615] on the web server that allows ZF to poll for changes to ECHConfigList values. For example, if a web server generates new ECHConfigList values hourly and publishes those at the well-known URI, ZF can poll that URI. When ZF sees new values, it can check if those work, and if they do, then update the zone file and re-publish the zone.

[[The source for this is in https://github.com/sftcd/wkesni/ PRs are welcome there too.]]

2. Terminology

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Example use of the well-known URI for ECH

An example deployment could be as follows:

1. Web server generates new ECHConfigList values hourly at N past the hour via some regular, automated process (e.g. a cronjob)
2. ECHConfigList values are "current" for an hour, and remain usable for 3 hours from the time of generation
3. The cronjob updates the ECHConfigList values in a JSON resource at at https://$ORIGIN/.well-known/origin-svcb, as shown in Figure 1.
4. On the zone factory, an HTTP client retrieves this JSON resource. It attempts to connect to the origin using these values and confirms that they are working.
5. The zone factory observes that the JSON resource has an HTTP freshness lifetime of 3600 seconds, and chooses a DNS TTL of 1800. It updates the zone file for $ORIGIN and re-publishes the zone containing only the new ECHConfigList values.
6. When the TTL of the DNS records approaches the remaining freshness lifetime of the JSON resource, the zone factory attempts to refresh its cached copy of the JSON resource. If the resource has changed, it repeats this process.

4. The origin-svcb well-known URI

If a web server ($ORIGIN) wants to convey information to the Zone Factory, it publishes the JSON content defined in Section 5 at: https://$ORIGIN/.well-known/origin-svcb

The well-known URI defined here MUST be an https URL and therefore the zone factory verifies the correct $ORIGIN is being accessed. If there is any failure in accessing the well-known URI, then the zone factory MUST NOT modify the zone.

5. The JSON structure for origin service binding info

[[The JSON structure is a work in progress.]]

```
 {
     "endpoints": [{
         "priority": 1,
         "target": "cdn.example.",
         "ech": "AD7+DQA65wAgAC..AA=="
     }, {
         "priority": 1,
         "port": 8413,
         "ech": "AD7+DQA65wAgAC..AA=="
     }]
 }
```

Figure 1: Sample JSON for ECH without aliases

```
     {
         "alias": "cdn.example.net:443"
     }
```

Figure 2: Sample JSON with aliasing

The JSON file at the well-known URI MUST contain an object with either an "endpoints" key or an "alias" key. If the "endpoints" key is present, its value is an array whose elements represent HTTPS records in ServiceMode. Each element MAY contain one or more keys from the JSON HTTP Origin Info registry (see IANA Considerations). The initial registry entries are:

- priority: The value is a positive integer corresponding to the SvcPriority. If omitted, the zone factory SHOULD infer numerically increasing SvcPriority from the order of the endpoints array.
- target: The value is a string containing a fully qualified domain name, corresponding to the HTTPS record's TargetName. The default value is ".".
- port: The value is a non-negative integer, corresponding to the value of the "port" SvcParamKey.
- ech: The value is a string containing an ECHConfigList encoded in Base64 [RFC4648], corresponding to the value of the "ech" SvcParamKey.

An empty endpoint object corresponds to an HTTPS record with inferred SvcPriority, TargetName=".", and no ECH support. An empty record of this kind can be useful as a simple way to make use of the HTTPS RR type's HSTS behavior.

[[TODO: What does the zone factory do if it encounters an unrecognized field?]]

If the object contains an "alias" key, its value MUST be an "authority" (Section 3.2 of [RFC3986]). This indicates that $ORIGIN is hosted on the same endpoints as this target, and is equivalent to an HTTPS AliasMode record. A zone factory might implement this directive by publishing an AliasMode record, publishing a CNAME record, copying HTTPS records from the target zone, or fetching https://$TARGET/.well-known/origin-svcb" (if it exists).

This arrangement provides the following important properties:

- Origins can indicate that different ECHConfigs are used on different ports.
- Origins can indicate that multiple CDNs are in use, each with its own ECHConfig.
- Origins that simply alias to a single target can indicate this without copying the ECHConfig and other parameters, which can interfere with key rotation and other maintenance.
- "port" and "target" are generally sufficient to uniquely identify a ServiceMode record, so zone factories can use the endpoint list to add ECH to pre-existing ServiceMode records that may have other SvcParams.

6. Zone factory behaviour

The zone factory SHOULD check that the presented endpoints work and provide access to $ORIGIN before publication. A bespoke TLS client may be needed for this check, that does not require the ECHConfigList value to have already been published in the DNS. [[I guess that calls for the zone factory to know of a "safe" URL on $ORIGIN to try, or maybe it could use HTTP HEAD? Figuring that out is TBD. The ZF could also try a GREASEd ECH and see if the retry-configs it gets back is one of the ECHConfig values in the ECHConfigList.]]

A careful zone factory could explode the ECHConfigList value presented into "singleton" values with one public key in each and test each for each endpoint.

The zone factory SHOULD publish all the endpoints that are presented in the JSON file, and that pass the check above.

The zone factory MUST set a DNS TTL short enough that any generated records expire from DNS caches before the JSON object's HTTP cache lifetime expires. The zone factory MUST refresh the JSON object and regenerate the zone before it expires each time. This ensures that ECHConfigs are not used longer than intended by the origin, while permitting the zone factory to limit the TTL if desired.

7. Security Considerations

This document defines another way to publish ECHConfigList values. If the wrong keys were read from here and published in the DNS, then clients using ECH would do the wrong thing, likely resulting in denial of service, or a privacy leak, or worse, when TLS clients attempt to use ECH with a backend web site. So: Don't do that:-)

Although this configuration resource MAY be publicly accessible, general HTTP clients SHOULD NOT attempt to use this resource in lieu of HTTPS records queries through their preferred DNS server:

- The bootstrap connection would not be able to use ECH, so it would reveal all the information that ECH seeks to protect.
- The origin could serve the user with a uniquely identifying configuration, potentially resulting in an unexpected tracking vector.

8. Acknowledgements

Thanks to Niall O'Reilly for a quick review of -00.

9. IANA Considerations

[[TBD: IANA registration of a .well-known. Also TBD - how to handle I18N for $FRONT and $BACKEND within such a URL.]]

If approved, this specification requests the creation of an IANA registry named "JSON HTTP Origin Info" with a Standards Action registration policy, containing a field named "Name" whose value is a UTF-8 string.

10. Normative References

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[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>.

[RFC8615] Nottingham, M., "Well-Known Uniform Resource Identifiers (URIs)", RFC 8615, DOI 10.17487/RFC8615, May 2019, <https://www.rfc-editor.org/info/rfc8615>.

[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, <https://www.rfc-editor.org/info/rfc3986>.

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

[I-D.ietf-tls-esni] Rescorla, E., Oku, K., Sullivan, N., and C. A. Wood, "TLS Encrypted Client Hello", Work in Progress, Internet-Draft, draft-ietf-tls-esni-16, 6 April 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-esni-16>.

[I-D.ietf-dnsop-svcb-https] Schwartz, B. M., Bishop, M., and E. Nygren, "Service binding and parameter specification via the DNS (DNS SVCB and HTTPS RRs)", Work in Progress, Internet-Draft, draft-ietf-dnsop-svcb-https-12, 11 March 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-svcb-https-12>.

Appendix A. Change Log

[[RFC editor: please remove this before publication.]]

The -00 WG draft replaces draft-farrell-tls-wkesni-03.

Version 01 changed from a special-purpose design, carrying only ECHConfigs and port numbers, to a more general approach based on Service Bindings.

Version 02 is just a keep-alive

Authors' Addresses

Stephen Farrell
Trinity College Dublin
Dublin
2
Ireland

Phone: +353-1-896-2354
Email: stephen.farrell@cs.tcd.ie

Rich Salz
Akamai Technologies

Email: rsalz@akamai.com

Benjamin Schwartz
Google LLC

Email: bemasc@google.com