TLS
Internet-Draft
Intended status: Experimental
Expires: January 6, 2020

S. Farrell
Trinity College Dublin

July 5, 2019

A well-known URI for publishing ESNIKeys
draft-farrell-tls-wkesni-01

Abstract

We propose use of a well-known URI at which web servers can publish ESNIKeys as a way to help get those published in the DNS.

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 January 6, 2020.

Table of Contents

1. Introduction 2
2. Terminology 3
3. Example use of the well-known URI for ESNI 3
4. The esni well-known URI 3
5. The JSON structure for ESNIKeys 4
6. Zone factory behaviour 5
7. Security Considerations 5
8. Acknowledgements 6
9. IANA Considerations 6
10. Normative References 6
Appendix A. Change Log 7
Author's Address 7

1. Introduction

Encrypted Server Name Indication (ESNI) [I-D.ietf-tls-esni] for TLS1.3 [RFC8446] defines a confidentiality mechanism for server names in TLS. That requires publication of an ESNIKeys data structure in the DNS. An ESNIKeys structure contains the public component of a key pair that will typically be periodically (re-)generated by a web server. Many web servers will have an API that can be used to dynamically update ESNIKeys in the DNS. Some implementations/ deployments however, will not, so web server implementers 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 just for cases where the zone file (or equivalent) that includes the ESNIKeys RR is on some machine, which we here call a "zone factory," to which the web server doesn't have write access.

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

[[This idea could: a) wither on the vine, b) be published as it's own RFC, or c) end up as a PR for [I-D.ietf-tls-esni]. There is no absolute need for this to be in the RFC that defines ESNI, so (b) seems feasible if there's enough interest, hence this draft. 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 ESNI

An example deployment could be as follows:

- Web server generates new ESNIKeys hourly at N past the hour via a cronjob
- ESNIKeys are "current" for an hour, published with a TTL of 1800, and remain usable for 3 hours from the time of generation
- Web server has a set of "hidden" sites - the DNS name for each hidden web site is here represented as $HIDDEN, which will end up as a realSNI value to be encrypted inside an ESNI extension
- Web server has a "cover" site ($COVER), where $COVER will typically be the DNS name used in the ESNIKeys public_name field for ESNIKeys version 0xff02
- The cronjob creates creates a JSON file for each hidden site at https://$COVER/.well-known/esni/$HIDDEN.json
- Each JSON file contains an array with the ESNIKeys RR values for that particular $HIDDEN as shown in Figure 1 - the values in Figure 1 with ellipses are the RR values we want to eventually see in the DNS
- On the zone factory, a cronjob runs at N+3 past the hour, it knows all the names involved and checks to see if the content at those well-known URIs has changed or not
- If the content has changed the cronjob attempts to use the ESNIKeys, and for each $HIDDEN where that works, it updates the zone file and re-publishes the zone containing only the new ESNIKeys RR values

4. The esni well-known URI

When a web server ($COVER) wants to publish ESNIKeys information for a hidden site ($HIDDEN) then it provides the JSON content defined in Section 5 at: https://$COVER/.well-known/esni/$HIDDEN.json

The well-known URI defined here MUST be an https URL and therefore the zone factory verifies the correct $COVER 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 ESNIKeys

[[Since the specifics of the JSON structure in Figure 1 are very likely to change, this is mostly TBD. What is here for now, is what the author has currently implemented simply because it worked ok and was easy to do:-)]]

```
[
    {
        "ESNIKeys.version": 0xff01,
        "desired-ttl": 1800,
        "ESNIKeys": "/wH5QHc...="
    },
    {
        "ESNIKeys.version": 0xff02,
        "desired-ttl": 1800,
        "ESNIKeys": "FF02897...OA"
    }
]
```

Figure 1: Sample JSON

The JSON file at the well-known URI MUST contain an array with one or more elements. Each element of the array MUST have these fields:

- ESNIKeys.version: contains a number with the value of the version field of the ESNIKeys. This is needed (today) as different versions are published in the DNS differently. (Draft-02 used a TXT RR and is still all that is usable with some early test deployments, draft-03 uses a new RRTYPE from the experimental range.)
- desired-ttl: contains a number indicating the TTL that the web server would like to see used for this RR. The zone factory MUST NOT use a longer TTL.
- ESNIKeys: contains the RRVALUE value to be used, either as a base64 encoded string (for ESNIKeys.version of 0xff01) or as an ASCII-HEX string (for ESNIKeys.version of 0xff02).

The JSON file contains an array for a couple of reasons:

- While ESNI is still in draft form, it may be necessary to publish different versions of the ESNIKeys structure.
- For some deployments, the same $HIDDEN could be accessible, using ESNI, via different $COVER (or public_name) web servers.
- As ESNIKeys is (regrettably:-) an extensible structure, it may be necessary to publish different ESNIKeys values to get best interoperability.

6. Zone factory behaviour

The zone factory SHOULD check that the presented ESNIKeys values work with the $HIDDEN server before publication. A "special" TLS client may be needed for this check, that does not require the ESNIKeys value to have already been published in the DNS. [[I guess that could call for the zone factory to know of a "safe" URL on $HIDDEN to use, or maybe it could use HTTP HEAD? Figuring that out is TBD.]]

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

The zone factory SHOULD only publish ESNIKeys values that are in the latest version of the JSON file. This leaves the control of "expiry" with the web server, so long as the ESNIKeys presented actually work. [[An alternative could be to have the new values just be appended to the zone, but that'd require some form of "notAfter" value in the JSON file which seems unnecessary and more complex.]]

From the point of view of the zone factory, the KeyShareEntry values within each element of the JSON array are entirely independent. The zone factory MUST NOT assume that there is any specific relationship between the ESNIKeys values in one JSON structure, nor between the set of JSON structures for the set of $HIDDEN sites that share a $COVER.

The ESNI specification [I-D.ietf-tls-esni] defines how and where the ESNIKeys RR for $HIDDEN needs to be published in the DNS.

A possibly interesting (unintended) consequence of this design is that once a TLS client has first gotten ESNIKeys from the DNS for $HIDDEN with the draft-03 ESNIKeys structure containing the public_name field, the TLS client would know both $COVER and $HIDDEN and so could later probe for this .well-known as an alternative to doing so via DoT/DoH. Probably not something a web browser might do, but could be fun for other applications maybe.

7. Security Considerations

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

8. Acknowledgements

Thanks to Niall O'Reilly for a quick review.

9. IANA Considerations

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

10. Normative References

[I-D.ietf-tls-esni]
Rescorla, E., Oku, K., Sullivan, N., and C. Wood, "Encrypted Server Name Indication for TLS 1.3", draft-ietf-tls-esni-03 (work in progress), March 2019.

[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>.

Appendix A. Change Log

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

From -00 to -01:

- Re-structured a bit after re-reading rfc8615

Author's Address

Stephen Farrell
Trinity College Dublin
Dublin 2
Ireland

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