ECH Interoperability Report

Stephen Farrell

Trinity College Dublin/Tolerant Networks Ltd.

stephen.farrell@cs.tcd.ie

work-in-progress 2024/12/09:15:02 UTC

Abstract

The abstract.

Index Terms

Encrypted Client Hello (ECH), Interoperability

I. INTRODUCTION

Deployments of the Transport Layer Security (TLS [1]) protocol expose the name of the server (e.g. the web site DNS name) via the Server Name Indication (SNI) field in the first message sent (the ClientHello). The Encrypted Client Hello (ECH) [2] extension to TLS is a privacy-enhancing scheme that aims to address this leak.

The Open Technology Fund (OTF) [1] have funded the DEfO project [2] to develop ECH implementations for OpenSSL, and to otherwise encourage implementation and deployment of

---

[1] https://www.opentech.fund/

[2] https://defo.ie

ECH. This report is a deliverable for the DEfO project and describes the current state of ECH interoperability.

As we expect the implementation and deployment environment for ECH to change over time, this report will be updated as events warrant and is currently versioned based on the build-time of this PDF. The latest version can be found at https://github.com/defo-project/ech-interop-report/ blob/main/ech-interop-report.pdf. Comments, additions or corrections are welcome. Those can be sent via email to the author, or as issues or PRs (preferably for the latex files) in the github repository for this report which is https://github.com/defo-project/ech-interop-report.

II. SOFTWARE

This section describes libraries, clients and servers that support ECH.

A. *Libraries*

The libraries listed in Table I have some support for ECH. Of the libraries listed, ECH support for all of the OpenSSL variants, python and libcurl were developed by the DEfO project.

TABLE I: Libraries with ECH

Name	Details
OpenSSL-sftcd	Source: https://github.com/sftcd/openssl/tree/ECH-draft-13c Version: 3.4.0-dev fork of master ECH support: client and server, TLS only (no DTLS, no QUIC) Comment: this is the DEfO project's main development branch
OpenSSL-defo	Source: https://github.com/defo-project/openssl/ Version: 3.5.0-dev fork of master, same ECH code as OpenSSL-sftcd ECH support: client and server, TLS only (no DTLS, no QUIC) Comment: this is used for DEfO project CI builds and tests
OpenSSL-feature-branch	Source: https://github.com/openssl/openssl/tree/feature/ech Version: 3.5.0-dev feature branch in upstream repo ECH support: stub APIs so far, next PR will have ECH client, then server after

TABLE I: Libraries with ECH (Continued)

Name	Details
	Comment: this is the DEfO-project's target for ECH PRs, and is where ECH code will end up prior to eventually being merged to master
boringssl	Source: https://boringssl.googlesource.com/boringssl Version: boringssl doesn't do versions, last local build 2024-11-29 ECH support: ECH client and server, ECH for TLS, QUIC and (possibly) DTLS Comment: in production use (chromium et al), a little more limited in HPKE suites than OpenSSL - only KEMs are x25519 and p256
NSS	Source: https://github.com/nss-dev/nss.git Version: NSS 3.108 ECH support: ECH client and server, ECH for TLS (unsure of DTLS/QUIC) Comment: in production use (firefox), a little more limited in HPKE suites than OpenSSL - only KEM is x25519
WolfSSL	Source: https://github.com/wolfSSL/wolfssl.git Version: 5.7.4 ECH support: ECH client and server, ECH for TLS (unsure of DTLS/QUIC) Comment: - ECH not built by default (needs "–enable-ech") - fails when HelloRetryRequest seen - https://github.com/wolfSSL/wolfssl/issues/6802
gnuTLS	Source: https://gitlab.com/gnutls/gnutls/blob/master/README.md Version: work-in-progress ECH support: interop untested (by DEfO-project) at this time Comment: an ECH merge request exists but has yet to be merged https://gitlab.com/gnutls/gnutls/-/merge_requests/1748
golang	Source: https://go.googlesource.com/go or https://github.com/golang/go/ Version: 1.23 or later required ECH support: client only, server coming in 1.24 (server code is merged) Comment:

TABLE I: Libraries with ECH (Continued)

Name	Details
	- golang tests were just (2024-12-02) added to our smokeping tests
rustls	Source: https://github.com/rustls/rustls/ Version: 0.23.19 ECH support: client only Comment: - rustls tests were just (2024-12-03) added to our smokeping tests
python	Source: https://github.com/defo-project/cpython Version: python 3.13/3.14 ECH support: TLS client only Comment: - python tests were just (2024-12-05) added to our smokeping tests
libcurl	Source: part of https://github.com/curl/curl Version: 8.10.0-DEV ECH support: TLS client only Comment: not well tested other than via command line curl

B. Clients

Current versions of firefox (e.g. 133.0, used in our smokeping tests) and chromium (version 131.0.6778.85 is used in our tests) support ECH by default. Other chromium-derived browsers also support ECH, while we don't include them in our ongoing tests described below, we have manually verified that current versions of vivaldi, brave and opera also have ECH support. The Tor browser (a firefox derivative) doesn't seem to support ECH by default. To test if a browser supports ECH one can visit https://test.defo.ie/ or, for much more detail https://test.defo.ie/iframe_tests.html.

Browsers in general are still somewhat quirky in terms of whether or not ECH will succeed, even when ECH is properly setup. This seems at least partly due to the delay that can inherently occur between the browser receiving DNS answers for A/AAAA records, and the time at which the browser receives a DNS answer for an HTTPS record (containing ECH information). It seems to be the case that browsers may be "impatient" in that they start the TLS session without

attempting ECH if that delay is too long, and that that happens sufficiently often to be noticeable.

TODO: Characterise browser oddities via some JS?

The only command line tool supporting ECH of which we're aware is curl. The DEfO project contributed ECH code as an experimental feature for curl that was merged to the master branch in April 2024. As an experimental feature, ECH is not built by default nor part of a release, so is only currently available to those who build from source. Curl supports using OpenSSL, boringssl or Wolfssl libraries for ECH. Build instrucftions are at https://github.com/curl/curl/blob/master/docs/ECH.md.

As a scripting tool, often used as part of a TLS client, Python also fits here. The DEfO project Cpython build adds ECH support to python via the OpenSSL library. That build is available at https://github.com/defo-project/cpython with the new APIs required for ECH described at https://irl.github.io/cpython/library/ssl.html#encrypted-client-hello.

C. Servers

The servers listed in Table II support ECH. In each case, the ECH integration, using OpenSSL, was developed as part of the DEfO project.

TABLE II: Servers supporting ECH

Name	Details
lighttpd	Source: https://github.com/defo-project/lighttpd Version: 1.4.76.81 Comment: only shared-mode ECH
nginx	Source: https://github.com/defo-project/nginx Version: 1.27.3.4 Comment: both shared-mode and split-mode ECH
apache2	Source: https://github.com/defo-project/apache-httpd Version: 2.5.0.ech.351 Comment: only shared-mode ECH
haproxy	Source: https://github.com/defo-project/haproxy Version: 3.2.dev0.62

TABLE II: Servers supporting ECH (Continued)

Name	Details
	Comment: both shared-mode and split-mode ECH
OpenSSL s_server	Source: https://github.com/defo-project/openssl
	Version: 3.5.0-dev
	Comment: shared-mode only but can easily "force" HelloRetryRequest (HRR)

III. SERVICES

A. Test Services

We can separate services into test and operational services. Table III lists known test services supporting ECH. All of those below .ie and my-own.net were setup by the DEfO project.

TABLE III: Test Services with ECH

Name	Details
defo.ie	https://defo.ie/ech-check.php is often used to check ECH
	ECH keys are rotated hourly, private usable for 3 hours, ECHConfigList in DNS contains only latest public
draft-13.esni.defo.ie	different server technology (as listed in Table II) instances on different ports as listed at https://defo.ie/, e.g., https://draft-13.esni.defo.ie:10413 is served by nginx
	ECH keys are rotated hourly, private usable for 3 hours, ECHConfigList in DNS contains only latest public
test.defo.ie	hosts a number of ECH server setups with good and variously bad configurations - see https://test.defo.ie/iframes_tests which describes those and allows a browser to attempt connections to each via Iframes
	ECH keys/configs are static for these setups
foo.ie	https://foo.ie/ech-check.php was setup used to check the defo.ie setup was easily replicated
	ECH keys are rotated hourly, private usable for 3 hours, ECHConfigList in DNS contains only latest public
my-own.net	this was to test the impact of having the same ECH keys on port 443 (https://my-own.net/ech-check.php) and another port (https://my-own.net:8443/ech-check.php) - at one point that made a difference to browsers

TABLE III: Test Services with ECH (Continued)

Name	Details
	ECH keys are rotated hourly, private usable for 3 hours, ECHConfigList in DNS contains only latest public
tls-ech.dev	https://tls-ech.dev/ was setup by the boringssl developers as a test server that uses boringssl ECH keys/configs seem to be static for this setup
Cloudflare	https://cloudflare-ech.com/cdn-cgi/trace is a test page setup by cloudflare that reports on ECH success/failure apparently, the server implementation and infrastucrure are part of Cloudflare's normal setup a similar test service used to be available at https://crypto.cloudflare.com/cdn-cgi/trace but that was turned off around the time that ECH was re-enabled for Cloudflare customers ECH keys are rotated hourly, private usable for N hours, ECHConfigList in DNS contains only latest public TODO: Check what N for private key usability
rfc5746.mywaifu.best	this ECH-enabled web page (https://rfc5746.mywaifu.best/) seems to have been setup as an ECH test site by someone using DEfO artefacts (nginx and documentation) but without any contact having happened between the person who set that up and any of the DEfO-project participants the person who set this up documented some of that at https://ckcr4lyf.github.io/tech-notes/services/nginx/nginx-ech.html

B. Operational Services

The only operational ECH service we know of is Cloudflare's deployment. However, that is non-negligible. Cloudflare earlier enabled ECH but disabled it soon after in October 2023 [3] as it caused some back-end issues. They then re-enabled ECH in October 2024. Our understanding of Cloudflare's deployment is that ECH is enabled by default for their "free" tier customers, but that paying customers have to take action to enable ECH.

---

[3] https://community.cloudflare.com/t/early-hints-and-encrypted-client-hello-ech-are-currently-disabled-globally/567730

In contrast, non-paying customers are not provided with a control to disable only ECH - they have to downgrade to TLSv1.2 in order to disable ECH. This is something that has been seen in recent weeks after the Russian government started blocking use of ECH to Cloudflare. [4]

C. Domain Probe Data

As part of our DEfO test setup, we have a web page where one can enter a host name and port and a script on our server will check if ECH is enabled for a web server at the name and port. (That's at https://test.defo.ie/domainechprobe.php.) Data is only stored if there is an HTTPS resource record for the name and port, if there is not we store nothing. In cases where this is an HTTPS resource record, we only store the name and port, whether the HTTPS value includes an "ech=" field, if there is, whether or not ECH worked using our ECH-enabled curl as the client. We also store the HTTPS resource record value, unless there are CNAMES or other kinds of re-direction involved.

We can use this data to get some further insight into services for which ECH is enabled. For now, we are not publishing the raw data - while data for the most recent 50 queries is shown on the web page, if we wanted to publish the raw data, we'd have to enable some form of consent for users, and it's not clear that'd be a) easy or b) worthwhile. (We currently take no steps to try record who made which requests to this page.)

Figure 1 shows the (small) numbers of queries made since August 2024. We can see a noticeable uptick in accesses just-before and after Russia blocked Cloudflare's ECH service. We also see some usage (at least 15 cases) that appear to indicate some web site owners may be using this service to check whether or not they have successfully disabled ECH. (The pattern is two quick accesses to the same name/port, the first of which indicates ECH worked, and the second that the HTTPS RR no longer contains an "ech=" value.)

The 1244 queries involved 517 unique names (117 of which are under the ".ru" ccTLD). The most commonly queried name is "youtube.com" (131 times), for which ECH is not currently enabled. Some 354 names were only queried once, and 86 were queried twice.

Of the 1244 queries, 660 referred to a name that had an "ech=" in the relevant HTTPS record, referring to 298 unique names. Within the 660 HTTPS values that contain an "ech=" value, we see 276 unique ECHConfigList values (4 of which are corrupt values published for our DEfO test

---

[4]https://betanews.com/2024/11/20/encrypted-client-hello-didnt-solve-censorship-but-still-may-have-a-role-to-play/
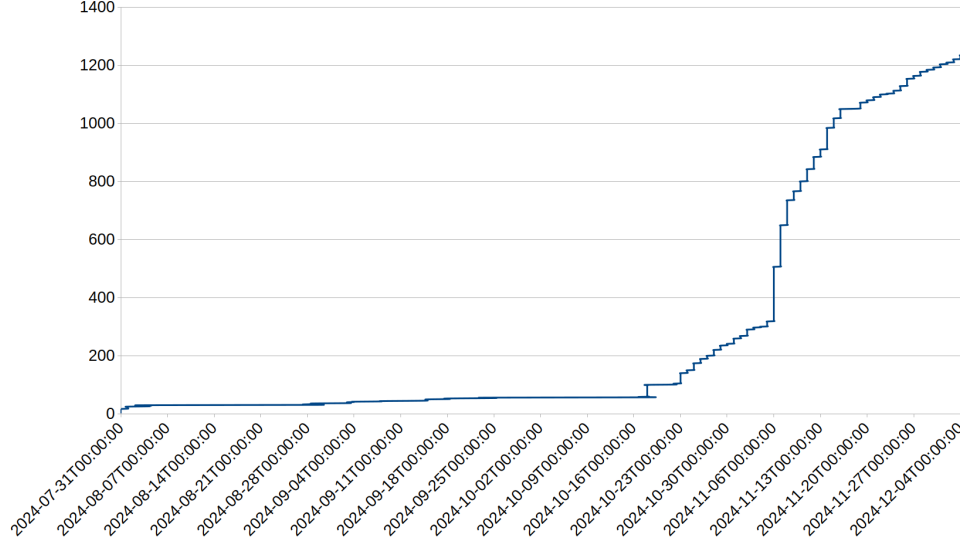
Fig. 1: Cumulative number of queries seen at https://test.defo.ie/domainechprobe.php versus time. The total number of queries is 1244.

services). Of the 298 unique names, 271 seem to be served by Cloudflare (based on the relevant name server names), 15 relate to test services described in Table III. At the time of writing, 10 of the names result in an NXDOMAIN or SERVFAIL response, 1 seems to correspond to a possible hobbyist site and 2 seem to (now) be parked domains or part of some advertising campaign. None of of those last 3 currently publish an HTTPS record.

So at least for the names entered to our domain probe page, we can seemingly conclude that Cloudflare have the only operational service at present.

IV. INTEROPERABILITY

Our primary interoperability testing is based on our "smokeping"-like tests. While smokeping measures latency for names/addreses, our tests aim to establish whether and which ECH configurations interoperate or fail to do so.

We mainly do this via a set of hourly cronjobs running on the test.defo.ie VM that attempt to access 67 URLs (some deliberately with broken configurations) from six different clients (described in Table IV. We report results for the most recent 6 runs at https://test.defo.ie/smokeping-summary.php.

TABLE IV: Smokeping clients

Name	Details
chromium	headless browser tests via selenium Version: 131.0.6778.85 Python script: https://github.com/defo-project/ech-dev-utils/blob/main/scripts/selenium_test.py ECH implementation based on boringssl
firefox	headless browser tests via selenium Version: 133.0 Python script: https://github.com/defo-project/ech-dev-utils/blob/main/scripts/selenium_test.py ECH implementation based on NSS
curl	bash script using curl Version: 8.11.1-DEV Bash script: https://github.com/defo-project/ech-dev-utils/blob/main/scripts/smoke_ech_curl.sh curl currently only pays attention the the first HTTPS resource record seen in DNS answers ECH implementation based on OpenSSL
golang	custom golang programme and bash script Version: golang 1.23 Golang programme: https://github.com/defo-project/ech-dev-utils/blob/main/scripts/ech_url.go Bash script: https://github.com/defo-project/ech-dev-utils/blob/main/scripts/smoke_ech_go.sh ECH implementation developed for golang
rustls	custom rustls programme and bash script Version: rustls 0.23.19 Rustls programme: https://github.com/defo-project/ech-dev-utils/blob/main/scripts/ech_url.rs

TABLE IV: Smokeping clients (Continued)

Name	Details
	Bash script: https://github.com/defo-project/ech-dev-utils/blob/main/scripts/smoke_ech_rs.sh
	ECH implementation developed for rustls
python	custom Cpython build and python script
	Version: python 3.13
	Python script: https://github.com/defo-project/ech-dev-utils/blob/main/scripts/ech_url.py
	Bash script: https://github.com/defo-project/ech-dev-utils/blob/main/scripts/smoke_ech_py.sh
	ECH implementation based on OpenSSL

There are some notes on these tests worth calling out in case someone wants to reproduce them:

- Some of the headless browser tests (e.g. with badly encoded ECHConfigList values) will cause selenium to throw exceptions, so test scripts need to catch those in order to properly determine that we got an expected failure

- Firefox will not attempt ECH to a name when the hostname of the test machine/VM is beneath the 2LD for that name, this causing unexpected failures if that hostname matches the origin of a test URL. In our case, we run both servers and test clients on the same VM ("test.defo.ie") so we have to set the hostname on the VM to something "peculiar" for tests to run as expected. (That of course causes "sudo" to complain which could disturb logs.)

- In the course of testing ECH in recent years, we have repeatedly made one misconfiguration error: using "." as the target name in HTTPS RRs for ports other then 443, when the correct thing do is to include the origin as the target name in such cases. That is when making an HTTPS RR for https://example.com:12345/ one needs to publish an HTTPS RR at "_12345._https.example.com" and the (presentation) value of that record needs to start with "1 example.com ..." where 1 is the priority and example.com is that targetName. So simply copying the value of the HTTPS RR for port 443 and re-publishing that for another port is not sufficient for interoperability. (But also does work for some clients in some cases, making it easier to accidentally do this and not notice.)

A. Results

The set of 67 test URLs covers all of the server technologies listed in Table II) though as all of those use OpenSSL for ECH, we direct most of the URLs towards one nginx instance, typically both (via a haproxy instance on) port 443 and direct to the nginx instance on port 15443. The set also include URLs for the test services listed in Table III. Some test URLs have "broken" configurations, e.g. with badly encoded HTTPS resource records, or have specification-conformant configurations that we expect may not work.

We have 6 clients testing hourly against 67 URLs, giving us 402 measurements per hour. The full set of URLs are too long to explain in detail here so we'll just show current results and describe a few cases where those results are more "interesting". The full set of URLs and the last 6 hours of test outcomes at https://test.defo.ie/smokeping-summary.php or web pages with (most of) the test URLs loaded via an Iframe can be accessed at https://test.defo.ie/iframe_tests_sort.html. That web page also includes some explanatory text about each test URL.

Table V below shows, for each test URL and each client, the percentage of expected and total results for a recent two day interval. [5]

TODO: Check expected values and sort out any self-caused anomalies.

TABLE V: Interop tests from 2024-12-06 00:00:00 to 2024-12-08 02:51:23.414218

num	url	chromium	curl	firefox	golang	rustls	python
1	https://2thenp-ng.test.defo.ie/echstat.php?format=json	1.00/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51
2	https://2thenp-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51
3	https://ap.test.defo.ie/echstat.php?format=json	0.96/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51
4	https://ap.test.defo.ie:15444/echstat.php?format=json	0.96/52	1.00/51	1.00/51	1.00/51	0.98/51	1.00/51
5	https://badalpn-ng.test.defo.ie/echstat.php?format=json	1.00/52	1.00/51	0.00/51	0.98/51	1.00/51	1.00/51
6	https://badalpn-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	1.00/51	0.00/51	1.00/51	0.98/51	1.00/51
7	https://bk1-ng.test.defo.ie/echstat.php?format=json	0.00/52	1.00/51	0.00/51	1.00/51	0.00/51	0.00/51
8	https://bk1-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	1.00/51	0.00/51	1.00/51	0.00/51	0.00/51
9	https://bk2-ng.test.defo.ie/echstat.php?format=json	0.00/52	1.00/51	0.00/51	1.00/51	0.00/51	0.00/51
10	https://bk2-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	1.00/51	0.00/51	1.00/51	0.00/51	0.00/51
11	https://bv-ng.test.defo.ie/echstat.php?format=json	0.00/52	1.00/51	0.00/51	1.00/51	0.00/51	0.00/51
12	https://bv-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	1.00/51	0.00/51	1.00/51	0.00/51	0.00/51
13	https://cloudflare-ech.com/cdn-cgi/trace	0.00/52	1.00/51	0.00/51	1.00/51	0.98/51	0.00/51
14	https://curves1-ng.test.defo.ie/echstat.php?format=json	1.00/52	1.00/51	1.00/51	1.00/51	0.57/51	1.00/51
15	https://curves1-ng.test.defo.ie:15443/echstat.php?format=json	0.90/52	1.00/51	1.00/51	1.00/51	0.37/51	1.00/51

[5]Table V is produced using https://github.com/defo-project/ech-dev-utils/blob/main/scripts/smokeping-summary-report.py.

TABLE V: Interop tests from 2024-12-06 00:00:00 to 2024-12-08 02:51:23.414218 (Continued)

num	url	chromium	curl	firefox	golang	rustls	python
16	https://curves2-ng.test.defo.ie/echstat.php?format=json	0.98/52	1.00/51	1.00/51	0.98/51	0.63/51	1.00/51
17	https://curves2-ng.test.defo.ie:15443/echstat.php?format=json	0.92/52	1.00/51	1.00/51	1.00/51	0.61/51	1.00/51
18	https://curves3-ng.test.defo.ie/echstat.php?format=json	0.00/52	1.00/51	0.00/51	0.00/51	0.45/51	1.00/51
19	https://curves3-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	1.00/51	0.00/51	0.00/51	0.49/51	1.00/51
20	https://defo.ie/ech-check.php	0.94/52	1.00/51	1.00/51	1.00/51	1.00/51	0.00/51
21	https://draft-13.esni.defo.ie:10413/	0.90/52	1.00/51	1.00/51	1.00/51	0.98/51	0.00/51
22	https://draft-13.esni.defo.ie:11413/	0.94/52	1.00/51	1.00/51	1.00/51	0.98/51	0.00/51
23	https://draft-13.esni.defo.ie:12413/	0.00/52	1.00/51	0.00/51	1.00/51	0.98/51	0.00/51
24	https://draft-13.esni.defo.ie:12414/	0.92/52	1.00/51	1.00/51	1.00/51	1.00/51	0.00/51
25	https://draft-13.esni.defo.ie:8413/stats	0.94/52	1.00/51	1.00/51	1.00/51	1.00/51	0.00/51
26	https://draft-13.esni.defo.ie:8414/stats	0.87/52	1.00/51	1.00/51	1.00/51	0.96/51	0.00/51
27	https://draft-13.esni.defo.ie:9413/	0.98/52	1.00/51	1.00/51	1.00/51	1.00/51	0.00/51
28	https://h1alpn-ng.test.defo.ie/echstat.php?format=json	0.96/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51
29	https://h1alpn-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	1.00/51	1.00/51	1.00/51	0.98/51	1.00/51
30	https://h2alpn-ng.test.defo.ie/echstat.php?format=json	0.94/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51
31	https://h2alpn-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	1.00/51	1.00/51	1.00/51	1.00/51	0.98/51
32	https://hidden.hoba.ie/	0.00/52	0.00/51	1.00/51	0.00/51	0.00/51	0.00/51
33	https://longalpn-ng.test.defo.ie/echstat.php?format=json	0.96/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51
34	https://longalpn-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51
35	https://ly.test.defo.ie/echstat.php?format=json	0.96/52	0.00/51	1.00/51	0.00/51	0.00/51	0.00/51
36	https://many-ng.test.defo.ie/echstat.php?format=json	0.94/52	1.00/51	1.00/51	0.98/51	1.00/51	1.00/51
37	https://many-ng.test.defo.ie:15443/echstat.php?format=json	0.21/52	1.00/51	0.00/51	1.00/51	1.00/51	1.00/51
38	https://min-ng.test.defo.ie/echstat.php?format=json	1.00/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51
39	https://min-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51
40	https://mixedalpn-ng.test.defo.ie/echstat.php?format=json	1.00/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51
41	https://mixedalpn-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51
42	https://mixedmode-ng.test.defo.ie/echstat.php?format=json	0.00/52	1.00/51	0.00/51	1.00/51	0.00/51	0.00/51
43	https://mixedmode-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	1.00/51	0.00/51	1.00/51	0.00/51	0.00/51
44	https://my-own.net/ech-check.php	0.94/52	1.00/51	1.00/51	1.00/51	1.00/51	0.00/51
45	https://my-own.net:8443/ech-check.php	0.79/52	1.00/51	1.00/51	1.00/51	1.00/51	0.00/51
46	https://myechtest.site/	0.00/52	1.00/51	0.00/51	1.00/51	0.00/51	1.00/51
47	https://ng.test.defo.ie/echstat.php?format=json	0.98/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51
48	https://ng.test.defo.ie:15443/echstat.php?format=json	0.92/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51
49	https://noaddr-ng.test.defo.ie/echstat.php?format=json	0.00/52	1.00/51	0.00/51	1.00/51	0.00/51	1.00/51
50	https://noaddr-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	1.00/51	0.00/51	1.00/51	0.00/51	1.00/51
51	https://p256-ng.test.defo.ie/echstat.php?format=json	0.00/52	1.00/51	0.00/51	0.00/51	0.00/51	1.00/51
52	https://p256-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	1.00/51	0.00/51	0.00/51	0.00/51	1.00/51
53	https://pthen2-ng.test.defo.ie/echstat.php?format=json	0.96/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51
54	https://pthen2-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	1.00/51	1.00/51	0.98/51	0.98/51	1.00/51
55	https://ss.test.defo.ie/stats	0.94/52	1.00/51	1.00/51	1.00/51	1.00/51	0.00/51
56	https://sshrr.test.defo.ie/stats	0.96/52	1.00/51	1.00/51	1.00/51	1.00/51	0.00/51

TABLE V: Interop tests from 2024-12-06 00:00:00 to 2024-12-08 02:51:23.414218 (Continued)

num	url	chromium	curl	firefox	golang	rustls	python
57	https://tls-ech.dev/	0.90/52	1.00/51	1.00/51	1.00/51	1.00/51	0.00/51
58	https://v1-ng.test.defo.ie/echstat.php?format=json	0.98/52	1.00/51	0.98/51	1.00/51	1.00/51	1.00/51
59	https://v1-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51
60	https://v2-ng.test.defo.ie/echstat.php?format=json	1.00/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51
61	https://v2-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51
62	https://v3-ng.test.defo.ie/echstat.php?format=json	0.94/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51
63	https://v3-ng.test.defo.ie:15443/echstat.php?format=json	0.96/52	0.98/51	1.00/51	0.92/51	1.00/51	1.00/51
64	https://v4-ng.test.defo.ie/echstat.php?format=json	0.00/52	1.00/51	0.00/51	1.00/51	0.00/51	0.00/51
65	https://v4-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	0.98/51	0.00/51	1.00/51	0.00/51	0.00/51
66	https://withext-ng.test.defo.ie/echstat.php?format=json	0.96/52	1.00/51	1.00/51	0.98/51	0.98/51	1.00/51
67	https://withext-ng.test.defo.ie:15443/echstat.php?format=json	0.00/52	1.00/51	1.00/51	1.00/51	1.00/51	1.00/51

The set of "interesting" things arising from Table V include

- Line 2 - https://2thenp-ng.test.defo.ie:15443/echstat.php?format=json works for all clients except chromium where it never works, yet the same configuration on port 443 works 100% of the time on all clients. That configuration has an ECHConfigList with two entries - the first an x25519 key and the second a p256 key - and the same ECHConfigList value is published for both ports. The error here however is in the configuration - the targetName for the port 15443 RR is "." rather than the hostname, so chromium here is correct and our test setup is wrong. We fixed a bug in https://github.com/defo-project/ech-dev-utils/blob/main/test-cases/test_cases_gen.py addressing this on 20241208.

  TODO: publish a "bad" config like that as a specific test

- Lines 5 and 6 - https://badalpn-ng.test.defo.ie/echstat.php?format=json and it's equivalent off port 443, work fine with all clients except Firefox and never work for firefox. The relevant DNS RR has an illegal alpn value (a '+' character). Hard to see this as other than firefox being over-fussy, though there could of course be some good reason to reject such alpn values.

- Line 7 and elsewhere - fixed a bug in https://github.com/defo-project/ech-dev-utils/blob/main/scripts/selenium_test.py where expected failures from tests using the "echstat_stat()" handler were being logged as test failures when they should be logged as expected.

- Line 13 - our "cf_check" result handler hadn't been changed to be applied with the change in Cloudflare's ECH test URL from "crypto.cloudflare.com" to "cloudflare-ech.com" which caused erroneous failure reports.

- Line 13 - our python test client is throwing an exception for the Cloudflare test URL:

```
[SSL: SSLV3_ALERT_ILLEGAL_PARAMETER] ssl/tls alert illegal parameter (_s
```

being investigated...

- Line 20 and similar - our python test client needed a tweak to not fail for tests not using the "echstat.php" server side script - fix was pushed on 20241209
- Line 32 - somehow certbot was no longer installed on test.defo.ie so we got a cert expiry for hidden.hoba.ie - re-instaled certbot to fix, which leads to some redundancy as we end up with ¿1 cert for some names (more or less all other name on that VM are below *.test.defo.ie)
- Line 35 - out python test client hits a certificate validation error for this one:

```
[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: unable to ge
```

being investigated...

TODO: based on the above, re-generate Table V in some hours

V. CONCLUSIONS

Conclude

ACKNOWLEDGEMENTS

Thanks to DEfO folks, OTF...

REFERENCES

[1] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, Aug. 2018. [Online]. Available: https://www.rfc-editor.org/info/rfc8446

[2] E. Rescorla, K. Oku, N. Sullivan, and C. A. Wood, "TLS Encrypted Client Hello," Internet Engineering Task Force, Internet-Draft draft-ietf-tls-esni-22, Sep. 2024, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-tls-esni/22/