IMPLICIT ECH AND IN-BAND UPDATES

IETF 122

Nick Sullivan
nicholas.sullivan+ietf@gmail.com

CURRENT ECH DEPLOYMENTS "STICK OUT"

Blocking of Cloudflare ECH in Russia, 2024-11-05 #417

Open

wkrp opened on Nov 8, 2024 · edited by wkrp Edits

[Discussion moved from #393 (comment). NTC threads are https://ntc.party/t/12837 (technical information) and https://ntc.party/t/12732 (discussion).]

Cloudflare's deployment of Encrypted Client Hello (ECH) is blocked in multiple networks in Russia since 2024-11-05. The blocking trigger is the presence of both of the following two elements in the Client Hello:

1. An SNI extension with the value `cloudflare-ech.com`.
2. An ECH extension.

Neither of these elements on its own is sufficient. That is, an SNI of cloudflare-ech.com *without* an ECH extension is not blocked, and ECH extensions that use an SNI other than cloudflare-ech.com are not blocked. In particular, you can still make ECH connections to servers that use a different public_name, such as defo.ie and tls-ech.dev; and GREASE ECH with SNI different from cloudflare-ech.com is not blocked.

INTEROPERABILITY HAZARD

6.1. Offering ECH

5. It SHOULD place the value of `ECHConfig.contents.public_name` in the "server_name" extension. Clients that do not follow this step, or place a different value in the "server_name" extension, risk breaking the retry mechanism described in Section 6.1.6 or failing to interoperate with servers that require this step to be done; see Section 7.1.

7.1. Client-Facing Server

Once the server has chosen the correct ECHConfig, it MAY verify that the value in the ClientHelloOuter "server_name" extension matches the value of ECHConfig.contents.public_name, and abort with an "illegal_parameter" alert if these do not match. This optional check allows the server to limit ECH connections to only use the public SNI values advertised in its ECHConfigs. The server MUST be careful not to unnecessarily reject connections if the same ECHConfig id or keypair is used in multiple ECHConfigs with distinct public names.

RECOVERY LOGIC

7.1. Client-Facing Server

- If the server is configured with any ECHConfigs, it MUST include the "encrypted_client_hello" extension in its EncryptedExtensions with the "retry_configs" field set to one or more ECHConfig structures with up-to-date keys. Servers MAY supply multiple ECHConfig values of different versions. This allows a server to support multiple versions at once.

6.1.6. Handshaking with ClientHelloOuter

If the server rejects ECH, the client proceeds with the handshake, authenticating for ECHConfig.contents.public_name as described in Section 6.1.7. If authentication or the handshake fails, the client MUST return a failure to the calling application. It MUST NOT use the retry configurations. It MUST NOT treat this as a secure signal to disable ECH.

6.1.7. Authenticating for the Public Name

When the server rejects ECH, it continues with the handshake using the plaintext "server_name" extension instead (see Section 7). Clients that offer ECH then authenticate the connection with the public name, as follows:

- The client MUST verify that the certificate is valid for ECHConfig.contents.public_name. If invalid, it MUST abort the connection with the appropriate alert.
- If the server requests a client certificate, the client MUST respond with an empty Certificate message, denoting no client certificate.

DISAMBIGUATION

If the server is unable to decrypt the ECH extension, it needs to decide whether to treat the connection as GREASE or ECH

1. If ECH: use a certificate that covers the public_name of the ECH config so the client can retry
2. If GREASE: use a certificate covering the outer SNI

OUTER SNI = PUBLIC NAME

Certificate:	public_name	=	outer SNI
Is GREASE	Continue, ignore config		Continue, ignore config
Is ECH	Retry ECH with config		Retry ECH with config

OUTER SNI + PUBLIC NAME ON SAME CERTIFICATE

Certificate:	public_name	+	outer SNI
Is GREASE	Continue, ignore config		Continue, ignore config
Is ECH	Retry ECH with config		Retry ECH with config

OUTER SNI != PUBLIC NAME

Certificate:	public_name	outer SNI
Is GREASE	Abort, retry no GREASE; or Try ECH with config	Continue, ignore config
Is ECH	Retry ECH with config	Abort, reject config

IMPLICIT ECH

- Extension in ECH that signals new server behavior
- Server MUST NOT abort when outer SNI does not match ECH config
- Outer SNI effectively deprecated for ECH connections - used for reachability only

IMPLICIT ECH

- Client Strategy for choosing outer SNI
  - Any choice of valid DNS names

- Server advertises its disambiguation strategy (public_name vs. outer SNI) or;
- (**NEW!** -MT and DJ) Client must support signed retry configurations

RETRY CONFIGURATION IS SIGNED

Certificate:	public_name	outer SNI
Is GREASE	Abort, retry no GREASE; or Try ECH with config	Continue, ignore config
Is ECH	Retry ECH with config	Retry ECH with config

HOW TO SIGN ECH CONFIGURATIONS

Many potential options

- Sign with private key of a Certificate that covers the public_name
  - Delegated Credential for safety(?)
- Sign with a Raw Public Key
  - Include a list of valid update keys(?)
- Sign with DNSSEC

IMPLICIT ECH AND IN-BAND UPDATES

IETF 122

Nick Sullivan
nicholas.sullivan+ietf@gmail.com