Signed Exchange Reporting for distributors

Authors: horo@chromium.org
Last Updated: 2019-Feb-22

Motivation:

Signed Exchange feature enables content publishers to sign their contents using their own private keys. User Agents (UAs) can trust the signed contents as if the contents are served from the publisher's origins even if they are served from other distributors' origins. Even if there is no network errors, UA may fail to load the signed content (example: the signature of the content has expired). This case is not covered by the Network Error Logging feature now. Both publishers and distributors can't recognize the errors in the user's environment.

Proposal:

Signed Exchange Reporting for distributors extends the Network Error Logging (NEL) to enable the distributors to investigate the signed exchange loading errors such as certificate verification errors.

Example:

A publisher (publisher.example) signed the article (https://publisher.example/article.html) as article.html.sxg. A distributor (distributor.example) is distributing the content at https://distributor.example/publisher.example/article.html.sxg and the certificate of publisher.example at https://distributor.example/publisher.example/cert.

If the distributor wants to investigate the signed exchange logs, the distributor sets the Report-To and NEL header in the HTTP response. This is same as the existing Network Error Logging feature.

```
Report-To: {"group": "sxg-errors",
            "max_age": 10886400,
            "endpoints": [{ "url": "https://report.distributor.example/" }] }
NEL: {"report_to": "sxg-errors", "max_age": 2592000}
```

Once UA receives the Report-To and NEL header, when UA failed to load (prefetch or navigation) the signed exchange content in the origin (https://distributor.example) because the

signature has expired, UA should send the report to the end point https://report.distributor.example/.

```
{
  "type": "signed-exchange",
  "age": 1,
  "url": "https://distributor.example/publisher.example/article.html.sxg",
  "user_agent": "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) ...",
  "body": {
    "phase": "sxg",
    "type": "sxg.signature_verification_error",
    "status_code": 200,
    "referrer": "https://www.example/",
    "method": "GET",
    "sxg": {
      "outer_url": "https://distributor.example/publisher.example/article.html.sxg",
      "inner_url": "https://publisher.example/article.html",
      "cert_url": ["https://distributor.example/publisher.example/cert"],
    }
  }
}
```

Security/Privacy Risks:

- Using this feature, the distributors can know whether the certificate URL is reachable or not from the user's environment. So if the report provides the detailed error information such as ”cert_fetch_error” or “cert_parse_error”, an evil attacker can do the port-scanning using it. To avoid the leaking of cross-origin information, Chrome will send only “ok” or “failed” when the origin (or IP address) of cert_url is different from the distributor.

Links:

Signed HTTP Exchanges: https://wicg.github.io/webpackage/draft-yasskin-http-origin-signed-responses.html
Loading Signed Exchanges: https://wicg.github.io/webpackage/loading.html
Network Error Logging: https://w3c.github.io/network-error-logging/

Webpackage spec issue: https://github.com/WICG/webpackage/issues/214
Webpackage spec PR: https://github.com/WICG/webpackage/pull/374

Network Error Logging spec issue: https://github.com/w3c/network-error-logging/issues/99
Network Error Logging spec PR: https://github.com/w3c/network-error-logging/pull/100