Expect-CT

In this lesson, we'll study the Expect-CT header.

We'll cover the following

Why `Expect-CT`?

The goal of `Expect-CT` is to inform the browser that it should perform additional background checks to ensure the certificate is genuine. When a server uses the `Expect-CT` header, it is requesting the client to verify that the certificates being used are present in public Certificate Transparency (CT) logs.

The Certificate Transparency initiative is an effort led by Google in order to:

> *[provide] an open framework for monitoring and auditing SSL certificates in nearly real time.*
>
> *Specifically, Certificate Transparency makes it possible to detect SSL certificates that have been mistakenly issued by a certificate authority or maliciously acquired from an otherwise unimpeachable certificate authority. It also makes it possible to identify certificate authorities that have gone rogue and are maliciously issuing certificates.*
>
> **certificate-transparency.org**

Note that a rogue server wouldn't set the expect-ct header, putting themselves on the line. A genuine server can ask clients to opt-in with all subsequent requests to be validated with CT moving forward. If the client gets tricked into connecting to a malicious server, the attack will never work as the SSL certificate won't pass the CT validation.

Sample header

Sample header

The header takes this form:

```
Expect-CT: max-age=3600, enforce, report-uri="https://ct.example.com/report"
```

In this example, the server is asking the browser to:

- enable CT verification for the current app for a period of one hour (3600 seconds)
- `enforce` this policy and prevent access to the app if a violation occurs
- send a report to the given URL if a violation occurs

The Certificate Transparency initiative's goal is to detect erroneously issued or malicious certificates (including rogue Certificate Authorities) earlier, faster, and more precisely than any other method before. By opting-in using the `Expect-CT` header, you can take advantage of this initiative to improve your app's security posture.

In the next lesson, we'll study `X-Frame-Options` which are meant to circumvent clickjacking attacks.