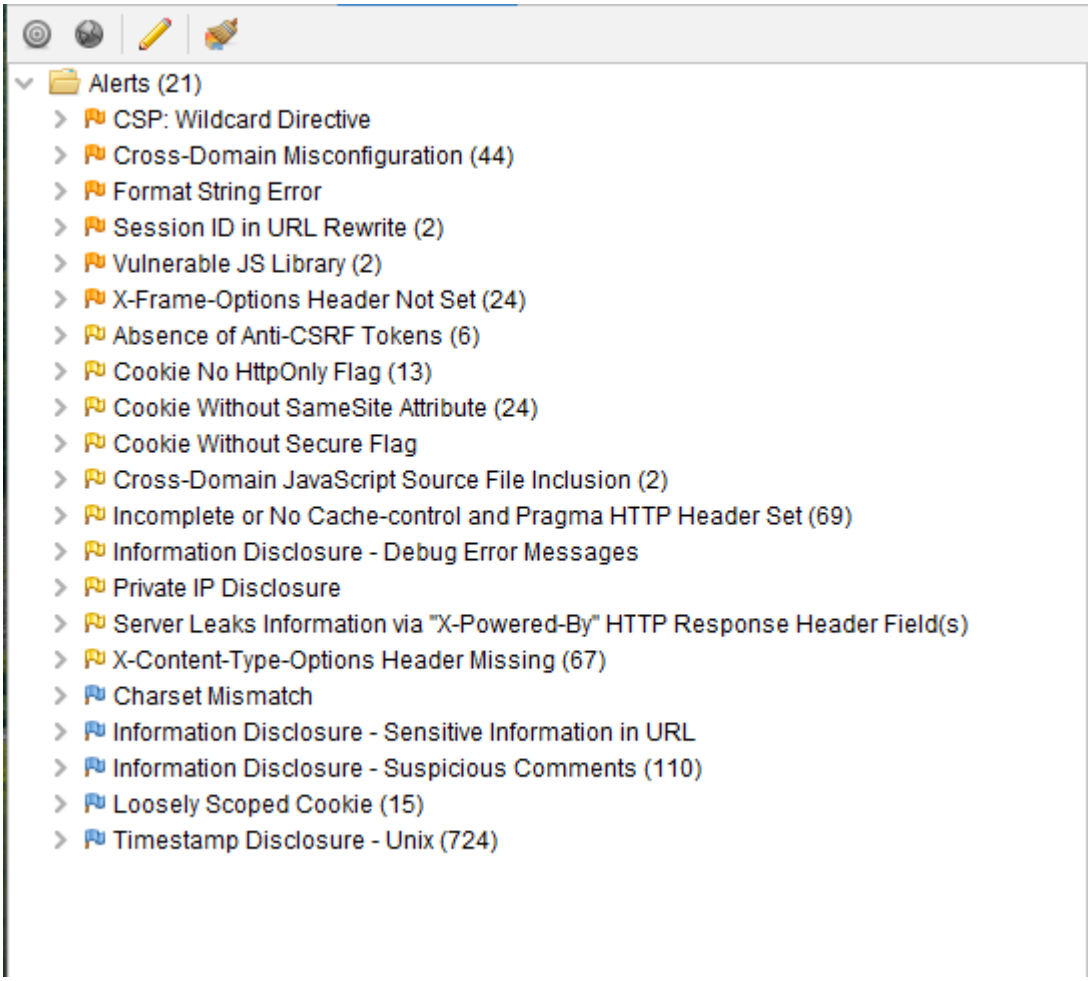
Zap Testing & Risk Mitigation

Summary of Alerts after running an active scan with ZAP:

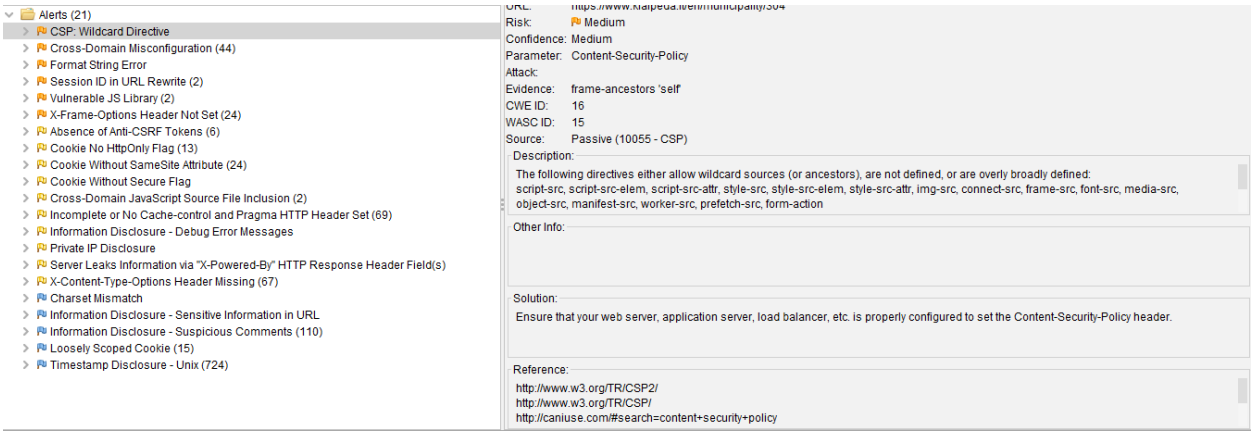


Scanning with ZAP did not discover any high security vulnerabilities, but it did reveal a bunch of medium and low-level security risks.

I focused on 3 separate medium risk vulnerabilities that are detailed below along with the steps I took to mitigate them.

Medium Vulnerability #1:

CSP: Wildcard Directive



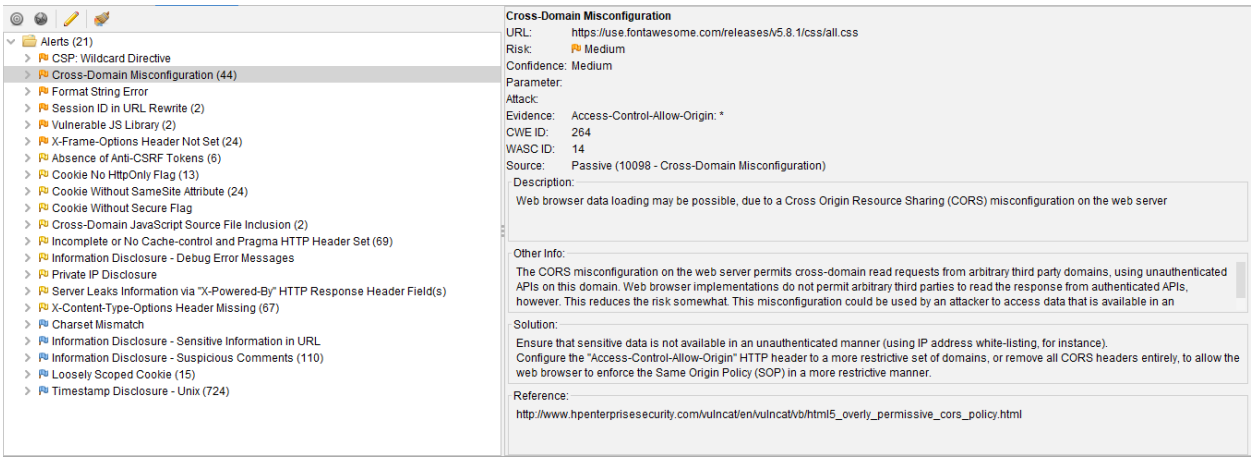
Mitigation:

To implement the Content-Security-Policy, I created the *web.config* file and added the following line of code to add the following explicit policy directives. This helps mitigate Cross Site Scripting (XCC) and data injection attacks.

```
<httpProtocol>
    <customHeaders>
        <add name="Content-Security-Policy" value="upgrade-insecure-requests; default-src 'self'; style-src 'self'; worker-src 'self';
            form-action 'self'; prefetch-src 'self'; frame-src 'self'; manifest-src 'self'; connect-src 'self';
            script-src 'self'; object-src 'none'; media-src 'none'; font-src 'none'; img-src 'none'; frame-ancestors 'self'" />
    </customHeaders>
</httpProtocol>
```

Medium Vulnerability #2:

Cross-Domain Misconfiguration



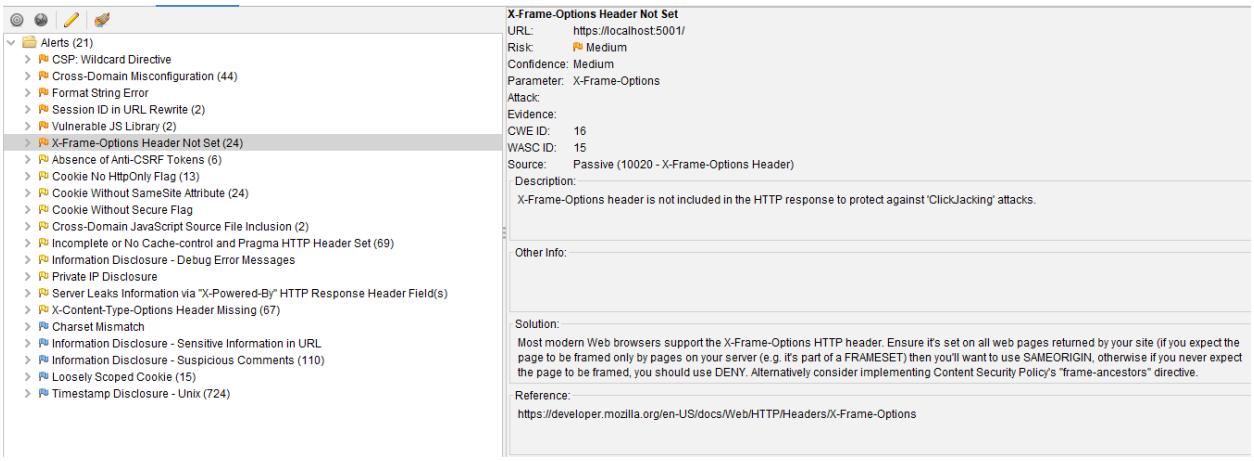
Mitigation:

To address this specific vulnerability, I added the following line of code to the *web.config's customHeaders* section. This specifies that the response header can be shared with the requesting code from the app's domain, which prevents any third-party origins from accessing the resource.

```
    <add name="Access-Control-Allow-Origin" value="domain" />
</customHeaders>
```

Medium Vulnerability #3:

X-Frame-Options Header Not Set



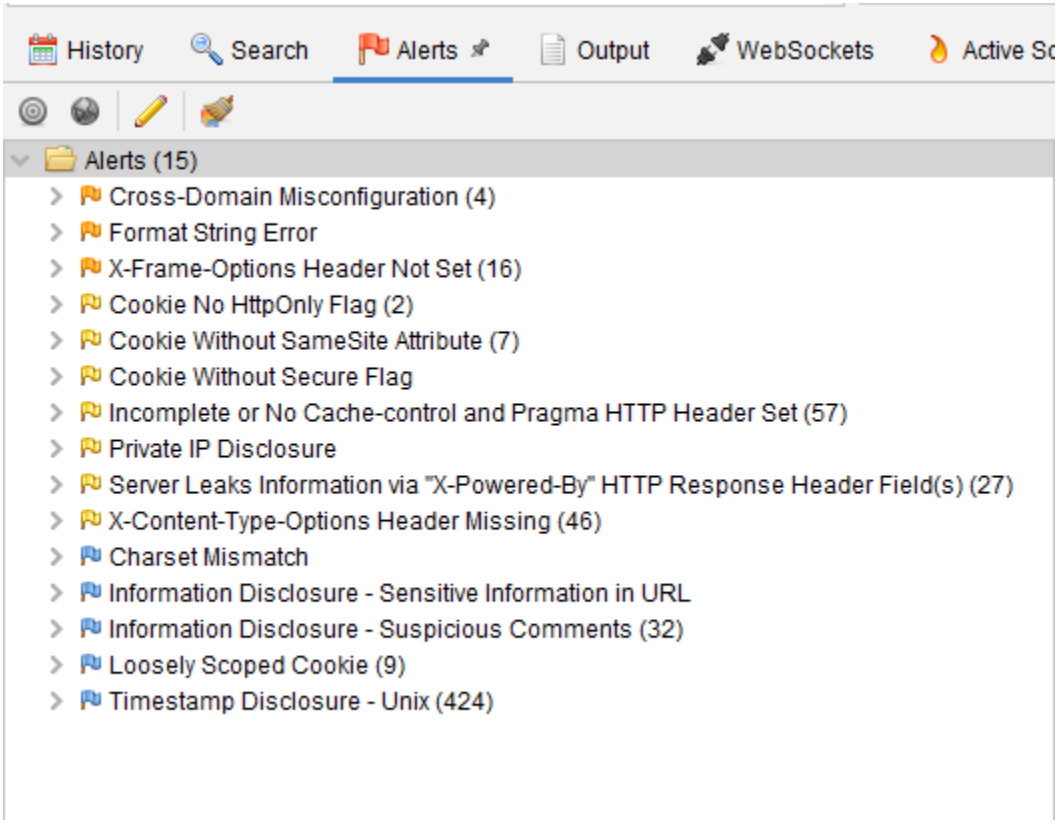
Mitigation:

To mitigate this vulnerability, I added the following line in the *Startup.cs* file, as well as the *X-Frame-Options* line and a *frame-ancestors* directive inside the *web.config* file. These steps help to mitigate the likelihood of clickjacking attacks against the website.

```
// mitigating X-Frame-Options exploitation
services.AddAntiforgery(options =>
{
    options.SuppressXFrameOptionsHeader = true;
});
```

```xml
<system.webServer>
    <httpProtocol>
        <customHeaders>
            <add name="X-Frame-Options" value="DENY" />
            <add name="Content-Security-Policy" value="upgrade-insecure-requests; default-src 'self'; style-src 'self'; worker-src 'self';
                form-action 'self'; prefetch-src 'self'; frame-src 'self'; manifest-src 'self'; connect-src 'self';
                script-src 'self'; object-src 'none'; media-src 'none'; font-src 'none'; img-src 'none'; frame-ancestors 'self'" />
            <add name="Access-Control-Allow-Origin" value="domain" />
        </customHeaders>
    </httpProtocol>
```

Summary of Alerts after implementing steps to mitigate security risks:



As evident from the screenshot, medium vulnerability alert count is reduced significantly. Low risk vulnerabilities were affected (lowered) as well.