Güvenlik açıklarını test etmek için OWASP ZAP uygulamasını kullandık. Bu uygulama 4 farklı çeşit uyarı verebiliyor:

Informational Priority Alert

Low Priority Alert: Genel iyileştirme gerektiren öneriler.

Medium Priority Alert: Önemli ama aciliyeti daha az (XSS).

High Priority Alert: Kritik sorunlar (SQL Injection).

Bizim projemiz, sorgulama sonucunda 3 Medium, 5 Low ve 9 Informational olmak üzere toplam 17 adet alert aldı.

**metin, ekran görüntüsü, yazılım, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturulduContent Security Policy (CSP) Header Not Set** (Medium)

CSP, web uygulamanızın hangi kaynaklardan içerik yükleyebileceğini belirleyen bir HTTP başlığıdır. Eksik olması durumunda, saldırganlar kötü niyetli JavaScript kodlarını yükleyebilir veya XSS saldırıları gerçekleştirebilir.

Çözüm olarak Startup.cs dosyasına   
app.Use(async (context, next) =>

{context.Response.Headers.Add("Content-Security-Policy", "default-src 'self';");

await next();});

middleware eklenerek problem giderilebilir.

**Cross-Domain Misconfiguratio**metin, ekran görüntüsü, yazılım, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu**n** (Medium)

Sunucunun CORS yapılandırması, Access-Control-Allow-Origin: \* gibi geniş izinler sunuyor. Bu, üçüncü taraf alanların uygulamanın kaynaklarına erişmesine olanak tanır. Hassas verilerin, istem dışı üçüncü taraflara sızdırılması riski barındırır.

Çözüm olarak Startup.cs dosyamıza

app.UseCors(builder =>

builder.WithOrigins("https://trusteddomain.com")

.AllowAnyMethod()

.AllowAnyHeader());

middleware eklenerek problem giderilebilir.

metin, yazılım, web sayfası, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu**Missing Anti-Clickjacking Header** (Medium)

Clickjacking, saldırganların bir kullanıcıyı, bilmeden zararlı bir işlem gerçekleştiren bir sayfaya tıklatmaya zorladığı bir saldırıdır. Bu saldırılar, sitenin başka bir web sitesinin çerçevesi içinde yüklenmesine izin veren eksik veya hatalı bir güvenlik başlığıyla kolaylaşır.

Çözüm olarak Startup.cs dosyasına   
app.Use(async (context, next) =>

{

context.Response.Headers.Add("X-Frame-Options", "SAMEORIGIN");

await next();

});

middleware eklenerek problem giderilebilir.

metin, sayı, numara, yazılım, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu**Cookie Without Secure Flag** (Low)

Bu sorun, hassas olmayan veriler için kritik bir tehdit oluşturmasa da oturum çerezleri veya hassas veriler içerdiğinde güvenlik açığı yaratabilir. Secure bayrağı olmayan çerezler, şifrelenmemiş HTTP bağlantıları üzerinden gönderilebilir. Bu, ortadaki adam (MITM) saldırılarına karşı çerezlerin savunmasız hale gelmesine neden olabilir.

Çözüm olarak Startup.cs dosyamızda

app.UseCookiePolicy(new CookiePolicyOptions

{

Secure = CookieSecurePolicy.Always,

HttpOnly = Microsoft.AspNetCore.CookiePolicy.HttpOnlyPolicy.Always

});

Şeklinde bir yapılandırma yapılabilir.

metin, yazılım, bilgisayar simgesi, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturuldu**Cross-Domain JavaScript Source File Inclusion** (Low)

Uygulamamız, üçüncü taraf bir alan adından (https://cdn.jsdelivr.net/) JavaScript dosyasını dahil ediyor. Üçüncü taraf kaynaklar güvenilir olmayabilir ve kötü niyetli içerik barındırabilir. Bu, XSS veya benzeri güvenlik açıklarını uygulamanıza dahil edebilir.

Çözüm olarak Üçüncü taraf bir alan adından JavaScript dosyaları dahil etmek yerine, bu dosyaları kendi sunucumuzda barındırabiliriz.

metin, yazılım, sayı, numara, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu**Server Leaks Version Information via "Server" HTTP Response Header** (Low)

HTTP yanıt başlığındaki "Server" alanı, sunucunun yazılım türünü ve sürümünü açık bir şekilde belirtiyor. Bu bilgiler, saldırganların belirli güvenlik açıklarını hedeflemesini kolaylaştırabilir.  
Çözüm olarak sunucumuzda Server başlığını tamamen devre dışı bırakabilir veya çok genel bir bilgi göstererek problemi çözebiliriz.

**Strict-Transport-Security Header Not Setmetin, yazılım, çizgi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu** (Low)

HSTS, web tarayıcılarına yalnızca HTTPS üzerinden sunucu ile bağlantı kurmalarını söyleyen bir güvenlik politikasıdır. HSTS olmadan, istemciler yanlışlıkla HTTP kullanabilir ve bu durum potansiyel olarak kötü niyetli "man-in-the-middle" (MitM) saldırılarına neden olabilir.

Çözüm olarak tüm HTTP isteklerini HTTPS' ye yönlendirebiliriz.

metin, yazılım, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**X-Content-Type-Options Header Missing** (Low)

X-Content-Type-Options başlığı ayarlanmamış. Bu, tarayıcının MIME-sniffing yapmasına izin vererek, yanıt gövdesinin belirtilen içerik türünden farklı bir içerik türü olarak yorumlanmasına yol açabilir. MIME-sniffing, tarayıcının içerik türünü belirtilenden farklı şekilde yorumlamasına neden olabilir. Örneğin, bir dosya türü "image/png" olarak ayarlanmış ancak tarayıcı tarafından farklı şekilde yorumlanmış olabilir.

Çözüm olarak sniffing konfigürasyonlarını gözden geçirilebilir.