OWASP Güvenlik Prensiplerinin Analizi ve Uyarlanması

1. Giriş ve Stratejik Çerçeve

Bu rapor, otomotiv, Nesnelerin İnterneti (IoT) ve API katmanlarını bir araya getiren çok disiplinli projenin siber güvenlik stratejisini oluşturmak üzere OWASP (Open Web Application Security Project) çerçevesinin nasıl uygulanacağını detaylandırmaktadır. Projenin araç içi ağlar, şarj altyapısı ve merkezi API'leri kapsayan karmaşık yapısı, standart güvenlik yaklaşımlarını yetersiz kılmaktadır. Bu bağlamda OWASP, yalnızca web uygulamaları için bir "Top 10" listesinden ibaret değildir; aksine, projenin her bir özgün bileşeni için (Otomotiv, IoT, API) özel rehberler sunan kapsamlı bir güvenlik çerçevesidir. Raporun temel amacı, bu özelleşmiş OWASP kaynaklarını kullanarak projeye özgü, bütünsel ve uygulanabilir bir siber güvenlik stratejisi geliştirmektir. Bu analiz, projenin "Security by Design" yaşam döngüsünün temelini oluşturacak ve güvenlik zafiyetlerinin dağıtım sonrası reaktif yöntemlerle düzeltilmesi yerine, geliştirme aşamasında proaktif olarak ortadan kaldırılmasını sağlayacaktır.

Projenin temel bileşenleri olan araç içi ağlar (CAN-Bus), elektrikli araç (EV) şarj altyapısı (OCPP) ve merkezi yönetim sistemleri (API, AI Modeli), hem operasyonel devamlılık hem de kullanıcı güvenliği açısından kritik öneme sahiptir. Bu bileşenlerden herhangi birinde meydana gelebilecek bir güvenlik ihlali, sadece veri sızıntısıyla kalmayıp fiziksel güvenlik riskleri ve hizmet kesintileri gibi ciddi sonuçlar doğurabilir.

Bu doğrultuda, bir sonraki bölümde projenin mimari katmanları ve bu katmanlara özgü tehdit yüzeyleri detaylı bir şekilde analiz edilecektir.

2. Proje Mimarisi ve Tehdit Yüzeylerinin Analizi

Projenin teknik mimarisi, her biri kendine özgü güvenlik riskleri barındıran farklı katmanlardan oluşmaktadır. Araç içi gömülü sistemlerden bulut tabanlı yönetim panellerine kadar uzanan bu yapı, her bir bileşenin farklı bir tehdit yüzeyi sunduğunu ve bu nedenle güvenlik stratejisinin katmanlı bir yaklaşımla ele alınması gerektiğini ortaya koymaktadır. Aşağıda projenin dört ana bileşeni, kapsamları ve temel güvenlik sorunları analiz edilmiştir.

2.1. Araç İçi Sistemler (Gömülü/Otomotiv)

- **Kapsam:** CAN-Bus iletişimi ve kritik Elektronik Kontrol Üniteleri (ECU).
- **Temel Sorunlar:** Mesaj enjeksiyonu (spoofing), mevcut mesajların tekrar gönderilmesi (replay attacks) ve kritik sistemlere yetkisiz erişim girişimleri.

- **Önerilen Çözüm Yaklaşımı:** Bu katmandaki güvenliğin, mesaj bütünlüğünü ve kaynağını kriptografik olarak güvence altına alacak bir Blockchain altyapısı ile sağlanması.

2.2. EV Şarj Altyapısı (IoT/Endüstriyel IoT)

- **Kapsam:** Şarj İstasyonları (EVSE) ve bu istasyonların merkezi sistemle iletişim kurmak için kullandığı OCPP (Open Charge Point Protocol).
- **Temel Sorunlar:** Zayıf şifreleme mekanizmaları, Man-in-the-Middle (MITM) saldırılarına açıklık, yetkisiz erişim, özellikle eski model şarj istasyonlarında bulunan güncel olmayan SSL/TLS protokol zafiyetleri ve yamalanmamış firmware açıkları.

2.3. İletişim ve API Katmanı (Backend)

- **Kapsam:** Şarj istasyonları ile merkezi sistem arasındaki API'ler ve V2X (Araçtan Her Şeye) iletişim protokolleri.
- **Temel Sorunlar:** Zayıf veya eksik kimlik doğrulama mekanizmaları, yetkilendirme eksiklikleri nedeniyle yetkisiz API kullanımı ve iletişim sırasında veri bütünlüğünün bozulması riskleri.

2.4. Analiz ve Yönetim Katmanı (Web/AI)

- **Kapsam:** Anomali tespitini gerçekleştiren yapay zeka sistemi ve sistemin yönetildiği web tabanlı panel (dashboard).
- **Temel Sorunlar:** Yönetim paneline yönelik standart web uygulama zafiyetleri (örn. XSS, Injection) ve yapay zeka modelinin kasıtlı olarak yanıltıcı verilerle beslenerek manipüle edilmesi (data poisoning).

Bu mimari katmanlar için en doğru ve etkili güvenlik rehberlerinin nasıl seçileceği, bir sonraki bölümde detaylı olarak ele alınacaktır.

3. İlgili OWASP Projelerinin Belirlenmesi ve Gerekçelendirilmesi

Projenin karmaşık yapısına en uygun güvenlik standartlarını belirlemek, mevcut kaynakları en verimli şekilde yönlendirmek ve en kritik riskleri önceliklendirmek için hayati bir adımdır. Bu nedenle, genel bir OWASP Top 10 listesi yerine, projenin her bir bileşeninin doğasına uygun, spesifik OWASP projelerini temel almak gerekmektedir. Bu yaklaşım, her bir teknoloji katmanının kendine has zafiyetlerini hedef alan, daha odaklı ve derinlemesine bir güvenlik analizi sağlar. Bu spesifik OWASP proje seçiminin temel çıktılarından biri, projenin "Hedef-2" kapsamında talep edilen kapsamlı güvenlik kontrol listesini oluşturmak olacaktır. Aşağıdaki tablo, proje bileşenlerini ilgili OWASP projeleriyle eşleştirmekte ve bu seçimin arkasındaki stratejik gerekçeleri açıklamaktadır.

Proje Bileşeni	İlgili OWASP Projesi	Gerekçe ve Uyarlama Alanı
OCPP / API	**OWASP API Security Top 10**	OCPP, temel olarak bir API protokolüdür. Projede belirtilen "Enerji Hırsızlığı" gibi işlevsel riskler, teknik olarak `API1:2023 - Broken Object Level Authorization` (bir kullanıcının başkasının şarj oturumunu durdurması) ve `API2:2023 - Broken Authentication` (sahte kimlikle şarj başlatma) gibi zafiyetlerle doğrudan ilişkilidir.
Şarj İstasyonu	**OWASP IoT Top 10**	Şarj istasyonları, sahada çalışan birer IoT cihazıdır. Tespit edilen "Zayıf şifreleme", "Firmware açıkları" ve "eski istasyonlardaki" güncel olmayan bileşen sorunları, `I1: Weak Passwords`, `I4: Lack of Secure Update Mechanism` ve

		`I7: Insecure Data Transfer` gibi temel IoT zafiyetleriyle birebir örtüşmektedir.
CAN-Bus	**OWASP Automotive Security**	Bu rehber, ISO 21434 standardına uyum sağlamak ve CAN-Bus mesaj enjeksiyonu, ECU manipülasyonu gibi doğrudan araç içi ağlara özgü riskleri yönetmek için zorunludur. Standart web veya IoT rehberleri bu alanda gerekli teknik derinliği sağlayamaz.
Firmware	**OWASP Firmware Security Testing Methodology**	"Eski şarj istasyonları" ve "firmware açıkları" sorunlarının kök neden analizi ve çözümü için bu metodoloji kritik öneme sahiptir. Bu rehber, mevcut istasyon firmware'lerinin statik ve dinamik olarak analiz edilerek gömülü zafiyetlerin tespit edilmesi için bir yol haritası sunar.
Yönetim Paneli	**OWASP Top 10 (Web Application)**	Anomali tespit sonuçlarının görüntülendiği ve sistemin yönetildiği

		web arayüzü, Cross-Site Scripting (XSS) ve SQL Injection gibi klasik web zafiyetlerine karşı korunmalıdır. Bu alanda standart OWASP Top 10 listesi en doğru referanstır.

Bu teorik eşleştirmelerin, projenin somut hedeflerine yönelik pratik güvenlik kontrollerine nasıl dönüştürüleceği bir sonraki bölümde detaylandırılacaktır.

4. OWASP Kontrollerinin Proje Hedeflerine Pratik Uyarlaması

Bu bölüm, önceki bölümlerde yapılan teorik zafiyet analizini, projenin spesifik risklerini ve hedeflerini doğrudan adresleyen somut, uygulanabilir güvenlik kontrollerine dönüştürmeyi amaçlamaktadır. Proje hedefleri ile ilgili OWASP zafiyetleri arasındaki bağlantıyı kurmak, güvenlik yatırımlarının iş hedeflerini doğrudan desteklemesini ve tespit edilen en kritik riskleri etkin bir şekilde azaltmasını sağlar.

Proje Hedefi/Riski: `Hedef-3: Enerji hırsızlığı ve sahte veri enjeksiyonu algoritması.`

- **İlgili OWASP Zafiyeti:** `OWASP API2:2023 - Broken Authentication` ve `API1:2023 - Broken Object Level Authorization.`
- **Analiz ve Uyarlama:** Enerji hırsızlığı senaryosu, teknik olarak bir saldırganın kimlik doğrulama (authentication) veya yetkilendirme (authorization) mekanizmalarındaki bir zafiyeti sömürerek hak sahibi olmadığı bir şarj seansını başlatmasıdır.
- **Önerilen Kontrol:** Kontrol mekanizması, her OCPP `StartTransaction` ve `StopTransaction` isteğinde, isteği yapan kullanıcının (`idTag`) sadece geçerli bir oturuma sahip olmasını yeterli görmemeli, ek olarak o kullanıcının spesifik olarak o şarj istasyonu ve o seans için işlem yapmaya *yetkili* olduğunun ayrıca doğrulanmasını yapmalıdır.

Proje Hedefi/Riski: `CAN-Bus mesaj enjeksiyonu, sahte mesaj, tekrar saldırıları.`

- **İlgili OWASP Zafiyeti:** `OWASP Automotive Security` rehberleri ve `OWASP IoT I7:2018 - Insecure Data Transfer & Storage`.

- **Analiz ve Uyarlama:** CAN-Bus protokolü, tasarımı gereği Güvenli Araç İçi İletişim (Secure On-board Communication - SecOC) prensiplerinden yoksundur. Bu durum, ağı dinleyen herhangi birinin mesajları okumasını, değiştirmesini (Tampering) veya sahte mesajlar göndermesini (Spoofing) mümkün kılar.
- **Önerilen Kontrol:** Projede önerilen **Blockchain çözümü**, bu temel zafiyet için birincil kontrol mekanizması olarak hizmet eder. Bu çözüm, mesaj özgünlüğü ve bütünlüğü için özel ve sağlam bir SecOC uygulaması görevi görerek her mesajın kaynağını ve içeriğini dağıtık bir defter üzerinden kriptografik olarak garanti altına alır ve bu OWASP zafiyetini etkin bir şekilde adresler.

Proje Hedefi/Riski: `Eski şarj istasyonlarındaki SSL zafiyetleri.`

- **İlgili OWASP Zafiyeti:** `OWASP IoT I5:2018 - Use of Insecure or Outdated Components` ve `I7:2018 - Insecure Data Transfer.`
- **Analiz ve Uyarlama:** Eski şarj istasyonlarının güncel olmayan ve zafiyet barındıran şifreleme kütüphaneleri veya protokolleri kullanması, iletişim kanalını Man-in-the-Middle (MITM) saldırılarına karşı savunmasız bırakır.
- **Önerilen Kontrol:** Bu kritik sorun için iki aşamalı bir çözüm stratejisi önerilmektedir: (1) **Yazılım Tabanlı Çözüm:** `OWASP Firmware Security Testing Methodology (FSTM)` kullanılarak istasyonların firmware'i analiz edilmeli ve eğer mümkünse güvenli protokolleri destekleyen bir yama geliştirilip dağıtılmalıdır. (2) **Donanım Tabanlı Çözüm:** Yamalamanın mümkün olmadığı durumlarda, istasyon ile merkezi sistem arasına, güvensiz iletişimi güvenli bir VPN tüneli içine alacak bir **"Güvenlik Ağ Geçidi" (Security Gateway)** donanımı eklenmelidir.

Bu kontrollerin daha geniş bir tehdit modellemesi çerçevesine nasıl entegre edileceği bir sonraki bölümde ele alınacaktır.

5. Tehdit Modellemesi ve STRIDE Entegrasyonu

Tehdit modellemesi, güvenlik açıklarını reaktif bir şekilde yamamaktan, potansiyel saldırı vektörlerini proaktif olarak öngörerek tasarım aşamasında engellemeye geçişi sağlar. Microsoft tarafından geliştirilen STRIDE metodolojisi, olası tehditleri sistematik olarak sınıflandırmak için etkili bir çerçeve sunar. Bu metodoloji, önceki bölümlerde belirlenen OWASP zafiyetlerini, bir saldırganın motivasyonunu yansıtan somut senaryolara dönüştürmemize yardımcı olur.

- **S**poofing (Kimlik Sahtekarlığı): Bir saldırganın, çalıntı veya sahte bir kullanıcı kimliği (`idTag`) ile OCPP API'sine istek göndererek ücretsiz bir şarj seansı başlatması senaryosu.

    - **Teknik Karşılığı:** `API2:2023 - Broken Authentication`.
- **T**ampering (Kurcalama/Veri Bütünlüğünü Bozma): CAN-Bus ağına sızan bir saldırganın, aracın hız verisini taşıyan bir mesajın içeriğini değiştirerek aracın yanlış bilgi işlemesine neden olması.
    - **Teknik Karşılığı:** `OWASP IoT I7:2018 - Insecure Data Transfer`.
- **R**epudiation (İnkar Edilemezlik): Bir işlemin (örneğin kritik bir ECU komutunun) göndericisi tarafından inkar edilmesi riski. Bu tehdit, projenin Blockchain altyapısı sayesinde her işlemin kriptografik olarak kanıtlanabilir ve değiştirilemez olmasıyla adreslenmektedir. Bu durum, teknik bir zafiyetten ziyade bir güvenlik garantisi sağlar.
- **I**nformation Disclosure (Bilgi İfşası): API'deki bir yetkilendirme hatası nedeniyle, `/charge-history` gibi bir uç noktanın, sadece istek atan kullanıcının değil, sistemdeki tüm kullanıcıların şarj geçmişi gibi hassas verileri döndürerek veri sızdırması riski.
    - **Teknik Karşılığı:** `API3:2023 - Broken Object Property Level Authorization`.
- **D**enial of Service (Hizmet Engelleme): Kötü niyetli bir aktörün, tek bir şarj istasyonuna saniyede binlerce sahte "şarj başlat" isteği göndererek istasyonun işlemci kaynaklarını tüketmesi ve meşru kullanıcıların hizmet almasını engellemesi.
    - **Teknik Karşılığı:** `API4:2023 - Unrestricted Resource Consumption`.
- **E**levation of Privilege (Yetki Yükseltme): Normal bir kullanıcının, API'deki bir fonksiyon seviyesi yetkilendirme hatasını sömürerek, sadece yönetici yetkileriyle çağrılması gereken "tüm istasyonların firmware'ini güncelle" gibi kritik bir fonksiyonu tetiklemesi riski.
    - **Teknik Karşılığı:** `API5:2023 - Broken Function Level Authorization`.

Bu tehdit analizleri ışığında, atılması gereken somut adımlar sonuç bölümünde bir eylem planı olarak özetlenmiştir.

6. Sonuç ve Stratejik Eylem Planı

Bu raporun ortaya koyduğu üzere, projenin başarısı, çok katmanlı ve karmaşık yapısının gerektirdiği bütünsel bir güvenlik stratejisinin benimsenmesine bağlıdır. OWASP çerçevesi, bu stratejiyi oluşturmak için kritik ve esnek bir araç olduğunu kanıtlamıştır. Sadece standart web zafiyetlerine odaklanmak yerine, projenin otomotiv, IoT ve API gibi her bir bileşeni için özelleşmiş OWASP rehberlerini temel almak, kapsamlı ve derinlemesine bir güvenlik duruşu sergilemeyi mümkün kılmaktadır. Yapılan analizler, teorik zafiyetlerin STRIDE metodolojisi ile somut risk senaryolarına dönüştürülmesini ve bu riskleri azaltacak pratik kontrol mekanizmalarının

belirlenmesini sağlamıştır. Bu bulgular doğrultusunda, projenin güvenlik olgunluğunu artırmak için aşağıdaki önceliklendirilmiş adımlar atılmalıdır:

1. **Rehberlerin Detaylı İncelenmesi:** Bu raporda tanımlanan 5 temel OWASP projesinin (API Security Top 10, IoT Top 10, Automotive Security, Firmware Security Testing Methodology, Web Application Top 10) resmi dokümanları ve kontrol listeleri, ilgili teknik ekipler tarafından detaylı bir şekilde incelenecektir.
2. **Güvenlik Kontrol Listesi'nin Oluşturulması:** İncelenen rehberlerden elde edilen bulgularla, projenin "Hedef-2: 50 Maddelik Güvenlik Kontrol Listesi"nin ilk taslağı oluşturulacaktır. Bu liste, geliştirme ve test süreçlerinde bir kılavuz olarak kullanılacaktır.
3. **API Güvenlik Testleri:** OCPP ve diğer backend API'leri, `OWASP API Security Top 10` listesindeki zafiyetlere karşı test edilecektir. Bu süreçte OWASP ZAP gibi otomatik güvenlik tarama araçlarından ve manuel penetrasyon testi tekniklerinden faydalanılacaktır.
4. **Firmware Analizi Planlaması:** Özellikle sahada bulunan eski model şarj istasyonlarının güvenlik risklerini yönetmek amacıyla, `OWASP FSTM` metodolojisine dayalı bir firmware analizi için detaylı bir takvim ve kaynak planlaması yapılacaktır.