T.C.

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜ

Ders: Ağ Güvenliği

PROJE RAPORU

Proje Konusu: Sunucu Güvenliğini Arttırmayı Amaçlayan Otomatik Bir Cloud Hardening Platformu

Hazırlayanlar:

Enes Smajli- B201210590, Soureya Rozzi Malli –B201210604,
Mahi Abdulhakim Mohammed- B201210605,

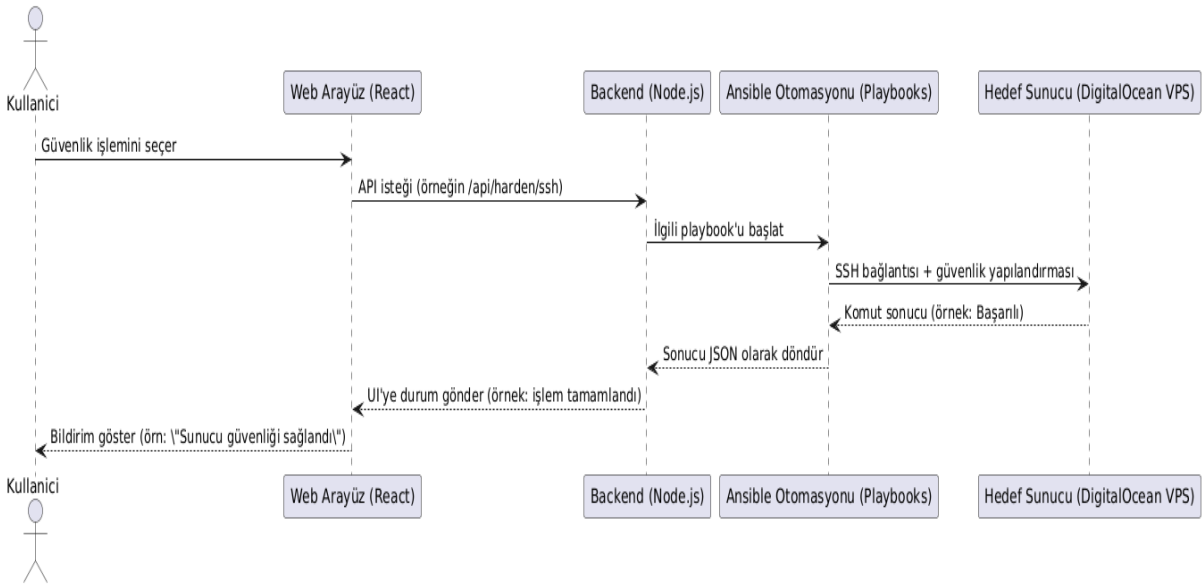
Projenin Amacı ve Hedefleri

Cloud Hardening Platformu, bulut üzerinde (özellikle DigitalOcean gibi servis sağlayıcılarda) çalışan sunucuların güvenlik seviyesini **otomatikleştirilmiş** ve **kullanıcı dostu** bir şekilde artırmayı amaçlar.
Manuel konfigürasyon hataları ve karmaşık güvenlik politikalarına ihtiyaç duymadan, sistem yöneticilerinin sunucularını:

- SSH erişim güvenliğini sağlamak,
- Güvenlik duvarı ayarlarını yönetmek,
- Kullanıcı ve yetki kontrollerini düzenlemek,
- Sistem güncellemelerini otomatikleştirmek,
- Giriş bannerları ile farkındalık yaratmak,

gibi kritik hardening görevlerini, basit arayüz üzerinden tek tıkla uygulayabilmeleri sağlanır.

Sequence diagramı:



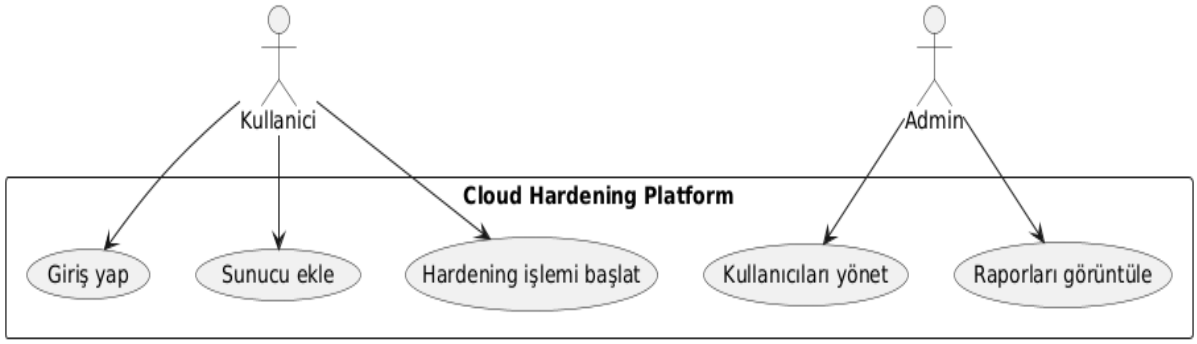
Bu diyagram, sistemdeki işlem akışını adım adım gösterir.

• Kullanıcı arayüzden bir güvenlik işlemi seçer.

• İstek backend'e gider, Ansible playbook çalıştırılır.

• Sunucuda işlem yapılır, sonuç kullanıcıya bildirilir.

Use Case Diagram (Kullanıcı ve Admin Etkileşimleri):



Bu UML diyagramı bir **Cloud Hardening Platform** (Bulut Güçlendirme Platformu) sisteminin actor ve use case ilişkilerini gösteriyor.

Aktörler: Kullanıcı ve Admin

İşlevler:

- Giriş yap, sunucu ekle, güvenlik işlemi başlat (her iki aktör)
- Kullanıcı yönet, rapor görüntüle (sadece Admin)

2. Sistem Mimarisi ve Bileşenler

2.1 Frontend (React.js + Firebase Authentication)

- Kullanıcıların platforma güvenli şekilde kayıt olup giriş yapmasını sağlar.
- Sunucu ekleme, hardening işlemlerini başlatma gibi işlemler için sezgisel ve hızlı bir arayüz sunar.
- Firebase Authentication ile oturum yönetimi ve kullanıcı güvenliği sağlanır.

2.2 Backend (Node.js + Express.js + Firebase Admin SDK)

- Frontend'den gelen istekleri REST API aracılığıyla karşılar.
- Kullanıcı isteklerine göre uygun Ansible playbooklarını tetikler.
- Playbooklar, Node.js üzerinden sistem komutu (`child_process.exec`) olarak çalıştırılır.
- Ansible playbooklardan dönen çıktı, işlem durumu ve hata mesajları toplanır, işlenir ve frontend'e iletilir.
- Firebase Admin SDK, kullanıcı yetkilendirme ve doğrulama için kullanılır.

2.3 Ansible Otomasyon Katmanı

- Güvenlik politikalarının uygulanmasını sağlayan YAML tabanlı playbooklar içerir:

    - SSH konfigürasyonu (root login kapatma, port değiştirme)
    - UFW firewall yönetimi
    - Kullanıcı yönetimi (kısıtlı sudo erişimi ile yeni kullanıcılar)
    - Banner ayarları
    - Sistem güncellemeleri
- Ansible, SSH üzerinden sunucuya bağlanır ve yapılandırmaları otomatik uygular.

2.4 Hedef Sunucular (DigitalOcean VPS)

- Platformun uyguladığı güvenlik politikalarının hedef aldığı uzak makineler.
- SSH ile erişim sağlanır ve gerekli konfigürasyonlar Ansible playbookları ile yapılır.

3. Çalışma Mantığı (İşlem Akışı)

1. **Kullanıcı**, web arayüzünden istediği güvenlik işlemini seçer veya yeni bir sunucu ekler.
2. Arayüz bu bilgiyi **backend**'e API çağrısı ile iletir.
3. Backend, gelen isteği doğrular ve uygun **Ansible playbook**'unu tetikler.
4. Ansible, SSH bağlantısı ile sunucuya bağlanır ve belirtilen güvenlik işlemlerini sırasıyla uygular.
5. İşlemin durumu ve çıktısı backend'e iletilir.
6. Backend, bu bilgiyi kullanıcının arayüzüne aktarır.
7. Kullanıcı, işlem sonucunu arayüzde onay, hata veya detaylar ile görür.

Bu yapı, işlemlerin hızlı, güvenli ve hatasız yürütülmesini sağlar.

4. Teknik ve Güvenlik Özellikler

- **JWT ve Firebase Authentication:** API erişimleri sadece doğrulanmış kullanıcılara açılır, yetkisiz erişimler engellenir.
- **Ansible + SSH Anahtarları:** Güvenli bağlantı ve otomatik konfigürasyon.
- **Çevresel Değişkenler:** Kritik parametreler `.env` dosyasında saklanır; bu dosyanın güvenliği önemlidir.
- **Otomasyonun Önemi:** İnsan hatalarını azaltır, standart güvenlik politikalarının tutarlı uygulanmasını sağlar.
- **Gerçek Zamanlı Geri Bildirim:** Kullanıcı arayüzünde anlık durum bilgisi sunulur, işlem takibi kolaylaşır.

5. Kullanılan Teknolojiler ve Araçlar

Katman	Teknoloji / Araç
Frontend	React.js, Tailwind CSS, Firebase Auth
Backend	Node.js, Express.js, Firebase Admin SDK
Otomasyon	Ansible (YAML playbook'ları)
Sunucu	DigitalOcean VPS, SSH

6. Sonuç

Bu proje, karmaşık ve kritik güvenlik işlemlerini otomatikleştirerek, sistem yöneticilerinin iş yükünü azaltmayı hedefler. Modern web teknolojileri ile güçlü otomasyon araçlarının birleşimi, bulut sunucularının güvenliğini etkin ve kolay yönetilebilir hale getirir.