## **DefectDojo Nedir?**

**DefectDojo**, güvenlik açığı yönetimi ve pentest takibi için geliştirilmiş **açık kaynaklı bir Application Vulnerability Management (AVM)** platformudur.

Amaç, birçok farklı güvenlik aracından çıkan çıktıları **tek bir yerde toplamak**, **zafiyetleri merkezi şekilde yönetmek**, **raporlamak** ve **geliştiricilerle iletişimi kolaylaştırmak**tır.

Başta OWASP tarafından başlatılan bu proje; büyük güvenlik ekiplerinin manuel işleri azaltmasını, otomasyonu artırmasını ve güvenlik sürecini daha verimli yönetmesini sağlar.

## **Neden Kullanılır?**

* Birden fazla tarama aracından (Burp Suite, Nikto, Nmap, ZAP, Trivy, SonarQube, Nessus, vs.) gelen çıktıları **tek çatı altında birleştirmek** için,
* Tespit edilen zafiyetleri **izlemek, sınıflandırmak, önceliklendirmek** ve çözüm süreçlerini belgelemek için,
* Geliştiriciler, güvenlik uzmanları ve yönetim ekipleri arasında **iletişimi merkezi hale getirmek** için,
* Güvenlik süreçlerinde **rapor üretimini ve otomasyonu sağlamak** için.

## Nasıl Kurulur? (Hızlı Kurulum - Docker)

DefectDojo’nun en kolay kurulumu **Docker Compose** ile yapılır:

* git clone https://github.com/DefectDojo/django-DefectDojo.git
* cd django-DefectDojo
* cp dojo/settings/settings.dist.py dojo/settings/settings.py
* docker-compose build
* docker-compose up

İlk kurulumda biraz zaman alabilir. Ardından aşağıdaki adresten erişebilirsin:

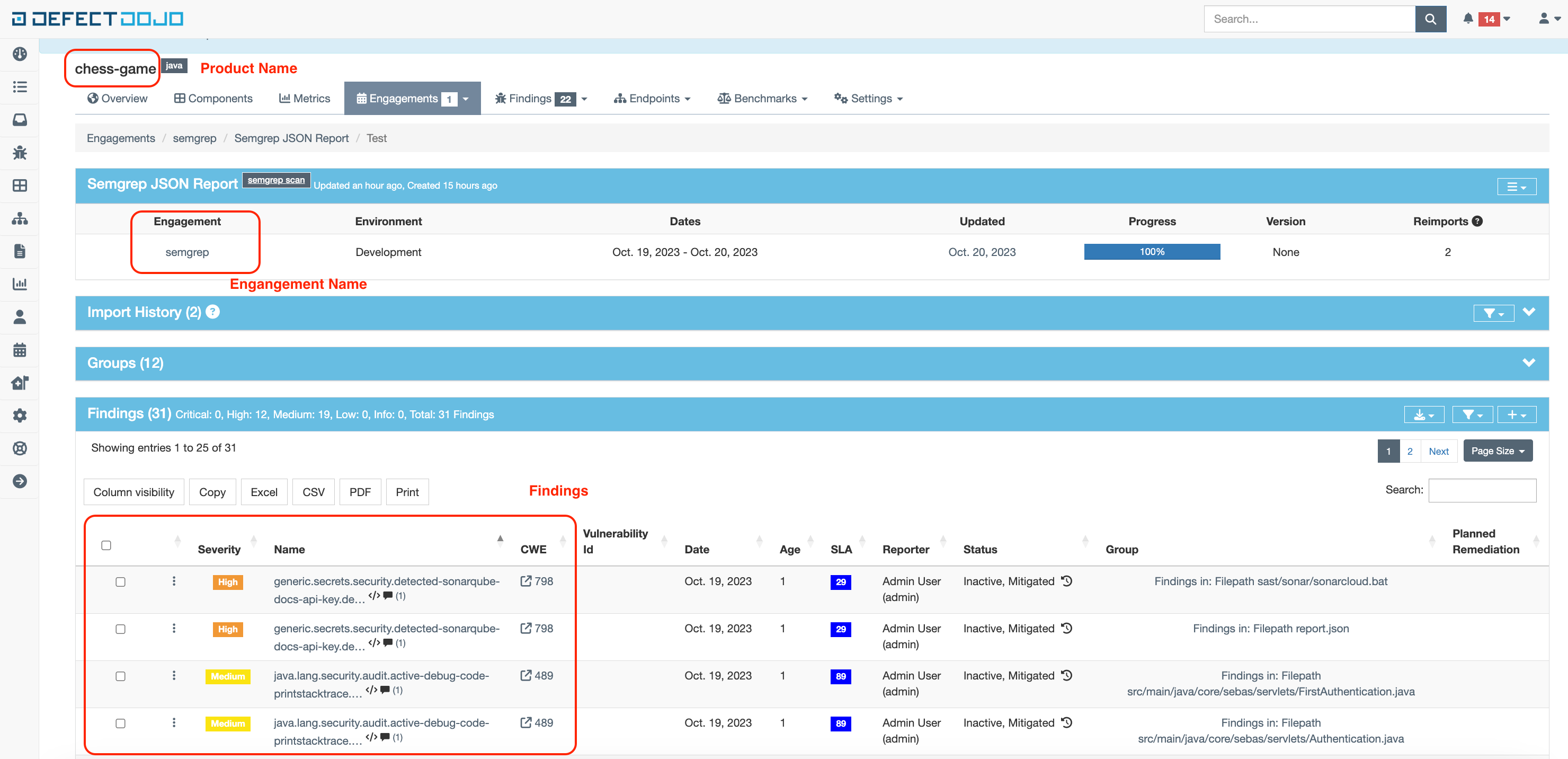
* [http://localhost:8080](http://localhost:8080/)
* Varsayılan kullanıcı: admin
* Varsayılan şifre: admin

**Not:** Kurulumdan sonra kullanıcı bilgilerini değiştirmen önerilir.

## **Kullanım Senaryosu**

### Aşamalar:

1. **Engagement (Proje)** oluştur → Örneğin: "Web Uygulama Pentest - Temmuz"
2. **Test Type** seç → (örn. Burp Scan, Manual Test, Static Analysis vs.)
3. Tarama sonucunu **upload et** → .json, .xml, .csv uzantılı çıktılar desteklenir.
4. DefectDojo, içeriği işler ve zafiyetleri **otomatik olarak listeler**.
5. Her zafiyet için:
   * Açıklama
   * Ciddiyet seviyesi (Critical/High/Medium/Low)
   * Mitigasyon önerisi
   * Durumu (Open/Resolved/Verified) gibi detaylar girilebilir.
6. Daha sonra gelişmeleri izlemek ve raporlamak için kullanılabilir.



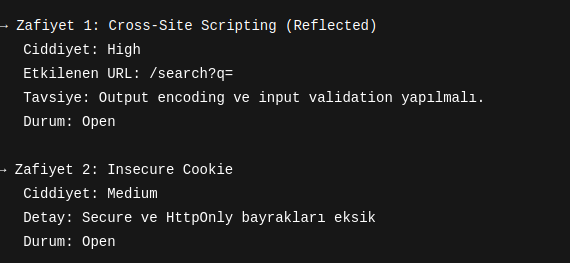
## **Desteklenen Güvenlik Araçları (Bazıları)**

* Burp Suite
* OWASP ZAP
* Nmap
* Trivy
* GitLab SAST
* Nikto
* Nessus
* SonarQube

Burp Suite'in JSON çıktısını yükledin:

* burp\_scan.json

**Örnek Çıktı:**



**Raporlama ekranlarında grafiksel olarak:**

* Zafiyet dağılımı
* Ciddiyet seviyelerine göre riskler
* Zamanla çözülen zafiyet sayısı gibi metrikler sunulur.