

Yüksek Güvenlikli Kod ve Uygulama Geliştirmek



[Download PDF Version](#)

Eğitim Süresi

- **Format 1**
 - **5 Gün**
 - **Ders Süresi:** 50 dakika
 - **Eğitim Saati:** 10:00 - 17:00
- **Format 2**
 - **12 Gün**
 - **Ders Süresi:** 50 dakika
 - **Eğitim Saati:** 10:00 - 17:00
- Her iki eğitim formatında eğitimler 50 dakika + 10 dakika moladır. 12:00-13:00 saatleri arasında 1 saat yemek arasındaki verilir. Günde toplam 6 saat eğitim verilir. 5 günlük formatta 30 saat eğitim, 12 günlük formatta toplam 72 saat eğitim verilmektedir. 12 saatlik eğitimde katılımcılar kod yazar ve eğitmenle birlikte sorulan sorulara ve taleplere uygun içerikler ve örnekler çalışılır.
- Eğitimler uzaktan eğitim formatında tasarlanmıştır. Her eğitim için teams linkleri gönderilir. Katılımcılar bu linklere girerek eğitimlere katılırlar. Ayrıca farklı remote çalışma araçları da eğitmen tarafından tüm katılımlara sunulur. Katılımcılar bu araçları kullanarak eğitimlere katılırlar.
- Eğitim içerisinde github ve codespace kullanılır. Katılımcılar bu platformlar üzerinden örnek projeler oluşturur ve eğitmenle birlikte eğitimlerde sorulan sorulara ve taleplere uygun içeriye

cevap verir. Katılımcılar bu araçlarla eğitimlerde sorulan sorulara ve taleplere uygun içeriğe cevap verir.

- Eğitim yapay zeka destekli kendi kendine öğrenme formasyonu ile tasarlanmıştır. Katılımcılar eğitim boyunca kendi kendine öğrenme formasyonu ile eğitimlere katılır. Bu eğitim formatı sayesinde tüm katılımcılar gelecek tüm yaşamlarında kendilerini güncellemeye devam edebilecekler ve her türlü sorunun karşısında çözüm bulabilecekleri yeteneklere sahip olacaklardır.

Yüksek Güvenlikli Kod Gelitmek

Teknolojinin hızla gelişmesiyle birlikte yazılım güvenliği, her geçen gün daha kritik bir öneme sahip olmaktadır. Siber saldırıların artan sıklığı, güvenlik açılarının hızla kötüye kullanılmasına yol açmakta ve hem bireyler hem de kurumlar için büyük riskler yaratmaktadır. Bu bağlamda, yazılım geliştiricilerinin güvenli yazılım geliştirme süreçleri hakkında derinlemesine bilgi sahibi olmaları hayatı önem taşır.

Eğitim Hedefi

Bu eğitimin temel hedefi, katılımcılara yazılım güvenliği konusunda kapsamlı bir bilgi ve beceri kazandırmaktır. Katılımcılar, yazılım geliştirme süreçlerinde karşılaşılan güvenlik açılarını tanıayıp etkili bir şekilde çözebilecek bilgiye sahip olacaklar. Eğitimin sonunda katılımcılar, güvenli kod yazma tekniklerini uygulayarak, yazılım projelerinde güvenlik önlemlerini entegre edebilecek, siber tehditlere karşı proaktif bir yaklaşım sergileyebilecektir.

- 1. Yazılım Güvenliği Temellerini Öğrenmek:** Güvenlik açıları, tehditler ve güvenli yazılım geliştirme kavramlarını anlamak.
- 2. Yaygın Güvenlik Açıkları ve Çözümleri:** OWASP Top Ten gibi yaygın güvenlik açılarını tespit etme ve bu açıları nasıl önleyebileceğini öğrenmek.
- 3. Güvenli Kod Yazma Teknikleri:** Güvenli yazılım geliştirme metodolojilerini uygulayarak, güvenli kod yazma becerisi kazanmak.
- 4. Güvenlik Testlerini Uygulamak:** Penetrasyon testleri ve güvenlik taramaları gibi test yöntemlerini öğrenmek ve uygulamak.
- 5. Güvenli Yazılım Mimarisi ve Tasarımı:** Güvenli yazılım mimarisi ve tasarım prensiplerini anlamak ve bu bilgileri projelere entegre etmek.
- 6. Güncel Güvenlik Trendlerini Takip Etmek:** Bulut güvenliği, IoT güvenliği gibi güncel konularda bilgi sahibi olmak ve gelecekteki tehditlere karşı hazırlıklı olmak.
- 7. Proaktif Güvenlik Yönetimi:** Güvenlik politikaları oluşturmak, güvenlik bilincini artırmak ve yazılım geliştirme süreçlerinde güvenlik önlemleri almak.

Eğitim İçeriği



1. Giriş ve Temel Kavramlar

- Kod güvenliği ve güvenlik açıklarının tanımı
- Yazılım geliştirme sürecinde güvenliğin önemi
- Güvenlik kavramlarının yazılım yaşam döngüsü üzerindeki etkisi
- Farklı güvenlik türleri: fiziksel güvenlik, ağ güvenliği, uygulama güvenliği

2. Yaygın Güvenlik Açıkları

- **OWASP Top Ten:**
 - SQL Enjeksiyonu
 - Kimlik Doğrulama ve Oturum Yönetimindeki Hatalar
 - Hassas Verilerin Açığa Çıkması
 - XML Harici Entiteler (XXE)
 - Güvenlik Yapılandırma Hataları
 - Kötü Güvenlik Denetimleri
 - İstemci Tarafı Hataları
 - Güvenli İletişim Hataları
 - Yazılım ve Bağımlılık Yönetimi Hataları
 - Yetersiz Güvenlik İzleme ve Kaydetme

• **SQL Enjeksiyonu (SQL Injection)**

Kullanıcı tarafından girilen verilerin doğrudan veritabanına sorgu olarak gönderilmesi, saldırganların kötü niyetli SQL kodları eklemesine olanak tanır. Bu, veritabanındaki hassas verilere erişim sağlayabilir.

- **XSS (Cross-Site Scripting)**

Kullanıcı tarafından girilen verilerin doğrulanmadan veya filtre edilmeden web sayfalarına dahil edilmesi, kötü niyetli JavaScript kodlarının kullanıcıların tarayıcılarında çalışmasına neden olabilir. Bu, kullanıcıların çalınması, kimlik avı (phishing) ve oturum çalınması gibi saldırılara yol açabilir.

- **CSRF (Cross-Site Request Forgery)**

Bir saldırganın, kullanıcıyı kandırarak, onun izni olmadan istemediği bir işlemi gerçekleştirmesini sağladığı bir saldırıdır. Kullanıcı, oturumu açıkken, zararlı bir bağlantıya tıklayarak, işlemi fark etmeden gerçekleştirebilir.

- **Yetersiz Girdi Doğrulaması ve Sanitizasyonu**

Kullanıcıdan gelen veri, uygulamanın beklediği formatta değilse veya doğrulama yapılmazsa, saldırganlar zararlı veri gönderebilir. Bu, örneğin, komut çalıştırma veya veritabanı sorgusu gibi işlemlere yol açabilir.

- **Yetkilendirme ve Kimlik Doğrulama Zayıflıkları**

Zayıf parola politikaları, kimlik doğrulama eksiklikleri veya yetki kontrolü hataları, kullanıcıların sisteme yetkisiz erişim sağlamasına neden olabilir. Ayrıca, oturum yönetimi hataları da önemli güvenlik açıklarıdır.

- **İnsecure Deserialization (Güvensiz Serileştirme)**

Uygulama, dışarıdan gelen verileri deserialize (serileştirme) ederek okurken, saldırganlar zararlı veriler gönderebilir. Bu, kod enjeksiyonu, saldırının sistemde çalışmasına yol açabilir.

- **Zayıf Şifreleme**

Verilerin yeterince güvenli bir şekilde şifrelenmemesi, şifreleme anahtarlarının kötü yönetilmesi veya şifreleme algoritmalarının zayıf olması, verilerin ele geçirilmesine neden olabilir. Özellikle hassas veriler (şifreler, kredi kartı numaraları) şifrelenmeden saklanması ciddi bir güvenlik açığı oluşturur.

- **İzin Kontrolü ve Erişim Hataları**

Yetersiz erişim kontrolü, kullanıcıların sadece yetkili oldukları verilere erişmesini engelmez. Bu, özellikle gizli verilere veya yönetici seviyesinde kaynaklara izinsiz erişim sağlanmasına yol açabilir.

- **Gizli Anahtarlar ve Şifreler Kodu İçinde Saklanması**

Yazılımda gizli anahtarlar veya şifreler düz metin olarak saklanması veya kaynak kodunda yer alması, siber saldırganlar için kolayca erişilebilen bir hedef haline gelir. Bu tür bilgiler güvenli bir şekilde saklanmalı ve yönetilmelidir.

- **İnsecure Direct Object References (IDOR)**

Bir kullanıcı, URL veya API üzerinden sistemdeki bir kaynağa (dosya, veri tabanı kaydı, vb.) izinsiz erişim sağlayabilir. Bu, genellikle URL parametreleri üzerinden gerçekleşen bir güvenlik açığıdır.

- **Bileşenlerin Güncel Olmaması (Outdated Components)**

Kullanılan yazılım bileşenlerinin, kütüphanelerin ve framework'lerin eski versiyonları, bilinen güvenlik açıklarına sahip olabilir. Güncel olmayan yazılım bileşenleri, güvenlik açıklarını artırır.

- **Hatalı veya Eksik Hata Yönetimi**

Hataların, kullanıcıya veya saldırganlara anlamlı bilgi sağlanmadan düzgün bir şekilde işlenmemesi, uygulamanın güvenlik açıklarının keşfedilmesine yol açabilir. Bu, saldırganlara sistemin zayıf noktalarını gösterebilir.

- **Server-Side Request Forgery (SSRF)**

SSRF saldırılarda, saldırgan sunucuya dış bir kaynağa (yerel ağlar veya harici hizmetler gibi) istekte bulunması için yönlendirme yapar. Bu, veritabanlarına veya yerel ağlara erişimi sağlayabilir.

- **Denial of Service (DoS) ve Distributed Denial of Service (DDoS)**

Sistemleri aşırı yükleyerek veya kötüye kullanarak, hizmetin çökmesine veya performansın ciddi şekilde düşmesine neden olabilir. Bu tür saldırılar, yazılımın ölçeklenebilirliğini ve dayanıklılığını test eder.

3. Güvenli Kod Yazma Prensipleri

- Kod güvenliği için en iyi uygulamalar
- Girdi doğrulama ve sanitizasyon
- Çıktı kodlaması ve güvenli veri işleme
- Doğru kimlik doğrulama ve yetkilendirme yöntemleri
- Güvenli oturum yönetimi
- Şifreleme yöntemleri ve veri koruma
- Hata yönetimi ve güvenlik bildirimleri

4. Güvenlik Araçları ve Teknolojileri

- Statik ve dinamik analiz araçları
- Güvenlik tarayıcıları ve güncel güvenlik açıkları veritabanları
- Güvenli yazılım geliştirme yaşam döngüsü (SDLC) araçları
- Kaynak kodu güvenlik analizi araçları: SonarQube, Veracode, Checkmarx
- Otomatik test ve sürekli entegrasyon araçları (CI/CD)

5. Güvenli Uygulama Geliştirme Süreci

- Güvenli uygulama geliştirme yaşam döngüsü (SDL)
- Proje planlama aşamasında güvenlik gereksinimlerinin belirlenmesi
- Tasarım aşamasında güvenlik mimarisi
- Geliştirme sürecinde güvenlik uygulamaları
- Test aşamasında güvenlik testleri ve değerlendirmeleri
- Dağıtım ve bakım aşamalarında güvenliğin sürdürülmesi

6. Güvenlik Testleri ve Değerlendirmeleri

- Penetrasyon testleri (Pentest) ve güvenlik değerlendirmeleri
- Manuel güvenlik testleri ve otomatik testlerin rolü
- Güvenlik açığı tarama yöntemleri
- Kırılganlık değerlendirme süreçleri
- Test sonuçlarının raporlanması ve iyileştirme önerileri

7. Güvenli Uygulama Mimarisi ve Tasarımı

- Güvenli yazılım mimarisi kavramları
- Servis odaklı mimari (SOA) ve mikro hizmet mimarisi güvenlik uygulamaları
- Güvenlik tasarım kalıpları (security design patterns)
- API güvenliği ve güvenli iletişim protokollerı

8. Güvenlik Yönetimi ve Politika Geliştirme

- Yazılım güvenliği politikalarının oluşturulması
- Güvenlik standartları ve düzenlemeleri (OWASP, ISO 27001, GDPR)
- Ekip içinde güvenlik bilincinin artırılması
- İlgili paydaşlarla güvenlik iletişimini ve işbirliği

9. Güncel Güvenlik Eğilimleri ve Gelecekteki Tehditler

- Bulut güvenliği ve uygulama güvenliği
- IoT güvenliği ve mobil uygulama güvenliği
- Yapay zeka ve makine öğrenimi ile güvenlik
- Gelecekteki güvenlik tehditleri ve savunma stratejileri

10. Pratik Uygulamalar ve Vaka Çalışmaları

- Gerçek dünya örnekleri ve vaka çalışmaları
- Güvenlik açılarının nasıl tespit edileceği ve düzeltileceği üzerine simülasyonlar
- Güvenli yazılım geliştirme projelerinde karşılaşılan zorluklar ve çözümleri
- Grup çalışmaları ve uygulamalı projelerle güvenlik becerilerinin pekiştirilmesi

Eğitim Yöntemi

- **Teorik Bilgi:** Güncel bilgiler ve konseptlerin anlatımı.
- **Uygulamalı Örnekler:** Gerçek senaryolarla pratik uygulamalar.
- **Etkileşimli Tartışmalar:** Katılımcıların aktif katılım sağlayacağı, soru-cevap şeklinde tartışmalar yapılacak oturumlar.
- **Proje Tabanlı Öğrenme:** Eğitimin son günü, katılımcıların öğrendiklerini pratikte uygulayacakları kapsamlı bir proje çalışması yapılacak.

Hedef Kitle

1. Yazılım Geliştiriciler ve Programcılar:

- Yazılım geliştiren ve güvenli yazılım uygulamaları oluşturmak isteyen yazılımcılar, yazılım güvenliği konularında bilgi ve becerilerini artırmak amacıyla bu eğitime katılabılır.
- Mevcut projelerde güvenlik önlemleri almayı hedefleyen yazılımcılar, güvenlik açıklarını tespit etme ve düzeltme becerisi kazanacaktır.

2. DevOps ve Sistem Yöneticileri:

- Yazılım geliştirme ve dağıtım süreçlerinde güvenliği sağlamakla sorumlu olan DevOps mühendisleri ve sistem yöneticileri, altyapı ve uygulama güvenliğini güçlendirerek yeni teknikler öğrenebilir.
- Güvenli altyapı yönetimi ve yazılım dağıtımını için gerekli bilgileri bu eğitimde bulacaklardır.

3. Yazılım Mimarları ve Tasarımcılar:

- Yazılımın güvenliğini tasarlama ve güvenli yazılım mimarları oluşturma konusunda bilgi sahibi olmak isteyen yazılım mimarları ve tasarımcıları, güvenlikli yazılım geliştirme ilkelerini öğrenmek için bu eğitimi alabilirler.
- Uygulama tasarımı ve sistem mimarisi aşamalarında güvenliği ön plana çeken mimarlar, eğitim ile kritik güvenlik tasarımlarını öğrenme fırsatı bulacaklardır.

4. Kalite Güvence (QA) ve Test Uzmanları:

- Yazılım ürünlerinin güvenlik testlerini gerçekleştiren ve güvenlik açıklarını tespit etmek için testler yapan QA uzmanları ve test mühendisleri, güvenlik testleri konusunda daha derinlemesine bilgi edinmek isteyebilir.
- Güvenlik açıklarını doğru şekilde test edebilmek ve yönetmek için gerekli teknik bilgiyi sağlayacaklardır.

5. Proje Yöneticileri ve Liderler:

- Yazılım projelerinin yönetiminden sorumlu olan, proje güvenliğini sağlamak isteyen proje yöneticileri ve liderler, güvenli yazılım geliştirme süreçlerini anlamak ve proje ekiplerini bu doğrultuda yönlendirmek amacıyla bu eğitimi alabilirler.
- Projelerinde güvenliği en baştan planlayarak, yazılım geliştirme döngüsünde güvenlik açıklarını minimize etmeyi hedefleyen yöneticiler için faydalı olacaktır.

6. Güvenlik Uzmanları:

- Yazılım güvenliği üzerine uzmanlaşmak isteyen ve güvenlik testlerini uygulayarak yazılımlarda güvenlik açılarını bulmak isteyen profesyoneller, bu eğitimde yeni güvenlik yöntemlerini öğrenebilirler.
- Güvenlik uzmanları, yazılım geliştirme süreçlerinde güvenlik tehditlerine karşı korunma yöntemlerini derinlemesine inceleyebilirler.

7. Yeni Başlayan Yazılım Geliştiriciler:

- Yazılım geliştirme alanında kariyerine yeni başlayan ve güvenli yazılım geliştirme konusuna ilgi duyan kişiler, bu eğitimi alarak temel güvenlik bilgilerini öğrenebilirler.
- Temel güvenlik kavramlarını anlamak ve uygulamalarında güvenliği sağlamak isteyen yeni yazılımcılar için temel düzeyde bir eğitim sağlanacaktır.

Katılımcılardan Beklentilerimiz

1. Aktif Katılım:

- Katılımcıların eğitim boyunca aktif bir şekilde derse katılmaları ve sorular sormaları beklenmektedir. Eğitim içeriği interaktif olduğundan, katılımcıların tartışmalara katılması, grup çalışmaları ve örnek vakalar üzerinden görüş paylaşmaları önemli olacaktır.

2. Temel Yazılım Bilgisi:

- Eğitim, yazılım geliştirme ve güvenlik konularında temel bilgiye sahip katılımcılara yönelik tasarlanmıştır. Katılımcıların yazılım geliştirme süreçleri hakkında temel bilgilere sahip olmaları ve temel programlama bilgisine sahip olmaları beklenmektedir.

3. Pratik Uygulamalara Katılım:

- Eğitimde verilen teorik bilgilerin pekiştirilmesi amacıyla pratik uygulamalar yapılacaktır. Katılımcıların uygulamalı bölümlere aktif katılım göstermeleri ve öğretendikleri bilgileri gerçek dünya senaryolarına uygulamaları beklenmektedir.

4. Öğrenmeye Açıklık:

- Eğitim süreci boyunca yeni güvenlik teknikleri ve yazılım geliştirme yöntemleri hakkında bilgi edinilecek ve mevcut bilgi seviyesini geliştirmeye yönelik bir yaklaşım benimsenmesi gerekmektedir. Katılımcıların yeni bilgilere açık olmaları ve gelişen güvenlik trendleri hakkında öğrenmeye istekli olmaları önemlidir.

5. Sürekli İletişim ve Geri Bildirim:

- Eğitim sırasında katılımcıların soruları ve geri bildirimleri eğitimin kalitesini artıracaktır. Katılımcılardan, eğitim sırasında ortaya çıkan herhangi bir soruyu eğitimlere yönetmeleri ve öğrenme sürecini verimli kılmak adına geri bildirimde bulunmaları beklenmektedir.

6. Proje ve Uygulama Çalışmaları:

- Katılımcıların eğitimde yer alan proje ve uygulama çalışmalarını zamanında ve dikkatlice tamamlamaları beklenmektedir. Bu çalışmalar, katılımcıların öğretendikleri teorik bilgileri pratikte nasıl uygulayacaklarını anlamalarına yardımcı olacaktır.

7. Ekip Çalışması ve İşbirliği:

- Eğitimde grup çalışmaları ve ekip içi işbirliği öne çıkacaktır. Katılımcıların, grup çalışmaları sırasında birbirleriyle işbirliği yapmaları, bilgi paylaşımı ve birlikte çözüm üretme yeteneklerini kullanmaları beklenmektedir.

8. Yazılım Güvenliği Bilincine Katkı:

- Katılımcıların, güvenli yazılım geliştirme konularında kazandıkları bilgileri işyerlerinde ve günlük projelerinde uygulamaları ve bu bilgiyi başkalarına aktararak yazılım güvenliği bilincinin arttırılmasına katkı sağlamaları beklenmektedir.

[Eğitim ana materyalleri, sadece eğitmenler için](#)