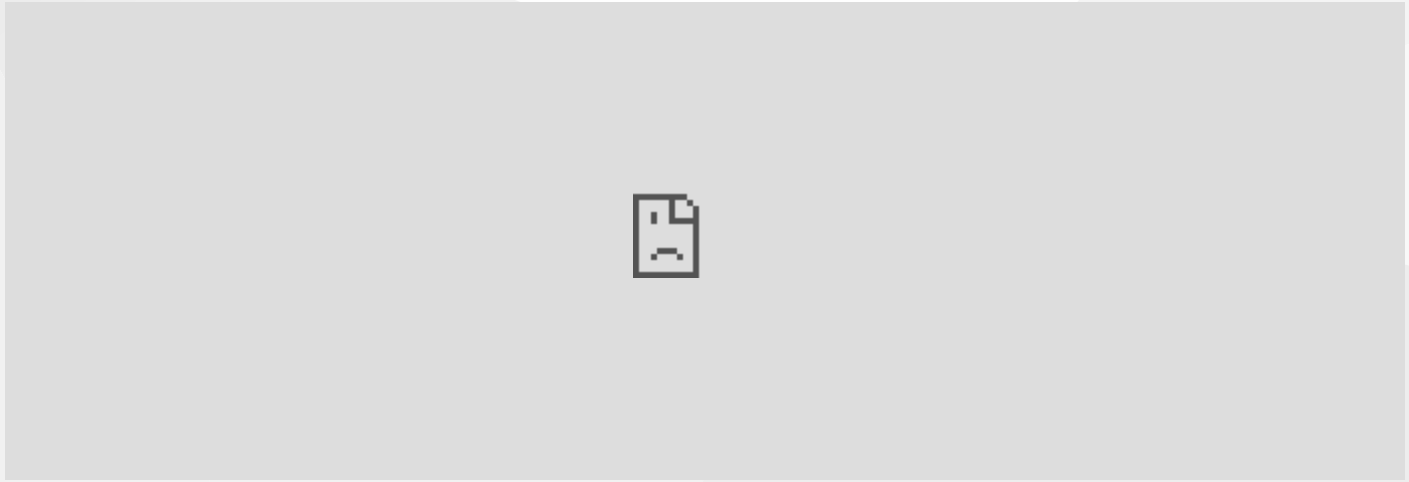
CE407 Güvenli Programlama

Hafta-1

Ders Planı ve İletişim, Güvenli Programlama ve Bilgisayar Virüsleri

İndir PDF, DOCX, SLIDE, PPTX



Outline

- Güvenli Programlama ve Bilgisayar Virüsleri
- Uygulama Koruma Planı
  - Kod Bölme
  - Kod Doğrulama
  - Zamanlama
  - Protokol İzleme
- Bilgisayar Virüsleri
  - Virüslerin Özellikleri
  - Virüs Türleri
  - Virüs Karşı Önlemleri
- Saldırı Ağaçları ve Güvenlik Modelleri
- Saldırı Yöntemleri
- Güvenli İletişim Hedefleri

Uygulama Koruma Planı (Application Protection Plan)

1. Kod Bölme (Split)

Teorik Açıklama:

Kod bölme, güvenilmeyen ortamda yürütülen işlemleri güvenilir bir ortama taşıma yöntemidir. Bu sayede güvenlik açıkları minimize edilir.

Uygulama:

- **Uygulama:** Bir istemci-sunucu modelinde şifreleme işlemlerini istemci yerine sunucuda gerçekleştiren bir sistem kurun. Bu, kritik işlemleri güvenli ortamda yürütmek için kullanılır.

2. Kod Doğrulama (Measure)

Teorik Açıklama:

Güvenilmeyen bir siteye ya da cihaza "Doğru kodu mu çalıştırıyorsun?" şeklinde sorular yönelterek, sistemin beklenen davranışları sergilediğini kontrol ederiz.

Uygulama:

- **Uygulama:** Bir uygulamanın çalışma sırasında belirli matematiksel problemlere doğru ve hızlı yanıt verip vermediğini kontrol eden bir sistem geliştirin. Bu sistem, doğruluğu kanıtlayamazsa işlem yapmaz.

3. Zamanlama (Time)

Teorik Açıklama:

Güvenilmeyen bir sistemde, işlem yapılması gereken bir zorluk hesaplatılır ve belirli bir zaman dilimi içerisinde cevap beklenir. Bu teknik, saldırganların analiz için yeterli zamanı bulmasını engeller.

Uygulama:

- **Uygulama:** Bir "Zaman Temelli Soru-Cevap" uygulaması oluşturun. Belirli bir süre içinde cevap alınmazsa oturum sonlandırılsın.

4. Protokol İzleme (Monitor)

Teorik Açıklama:

Veri transferi sırasında protokol akışını izleyerek, olası güvenlik açıklarını veya kötü niyetli işlemleri tespit ederiz.

Uygulama:

- **Uygulama:** Bir web sunucusunda yapılan HTTP isteklerini izleyen bir log sistemi oluşturun. Şüpheli istekler algılandığında kullanıcıyı engelleyin.

Bilgisayar Virüsleri

1. Virüslerin Özellikleri

- **Uyuma Durumu (Dormant):** Virüs bir süre sessiz kalabilir, algılanmaktan kaçınır.
- **Yayılma (Propagation):** Yeni dosyalara veya sistemlere bulaşır.
- **Tetikleme (Triggering):** Virüsün harekete geçeceği zamanı belirleyen olay.
- **Eylem (Action):** Zararlı işlem yapılır, bu genellikle "payload" denir.

Uygulama:

- **Uygulama:** Bir simülasyon oluşturun. Virüs uyuma durumunda beklesin, belirli bir tarihte etkinleşip bir dosya silme işlemi yapsın.

2. Virüs Türleri

- **Program/Dosya Virüsü:** Program dosyalarına bulaşır.
- **Makro Virüsü:** Word/Excel belgelerine bulaşır ve belge açıldığında çalışır.
- **Boot Sektörü Virüsü:** Sabit diskin önyükleme sektörüne bulaşır, bilgisayar başlatıldığında çalışır.

Uygulama:

- **Uygulama:** Farklı virüs türlerinin nasıl çalıştığını gösteren bir simülasyon oluşturun. Her virüs türü farklı tetikleyicilerle harekete geçsin.

3. Virüs Karşı Önlemleri

- **İmza Tabanlı Tespit (Signatures):** Virüsün bilinen kod parçalarına dayalı tespit yöntemidir.
- **Şifreleme:** Virüslerin kodlarının şifrelenmesi, imza tespitine karşı koruma sağlar.

Uygulama:

- **Uygulama:** Şifrelenmiş bir virüs simülasyonu oluşturun. Virüs kodu her çalıştırıldığında farklı bir anahtar ile şifrelenmiş olsun.

Güvenlik Modelleri ve Saldırı Ağaçları (Attack Trees)

1. Saldırı Ağacı Nedir?

Saldırı ağacı, bir saldırganın bir hedefe ulaşma stratejilerini anlamamızı sağlayan bir yapıdır. Bu model, güvenlik açıklarını görselleştirerek saldırılara karşı etkili savunmalar geliştirilmesine yardımcı olur.

Uygulama:

- **Uygulama:** Basit bir saldırı ağacı oluşturun. Örneğin, bir web uygulamasında SQL enjeksiyonundan başlayarak, veritabanına erişime kadar olan adımları modelleyin.

2. Maliyet Modelleme

Her saldırı adımının bir maliyeti vardır. Bu maliyetler saldırganın hedefe ulaşmasını zorlaştırmak için hesaplanabilir. Bir saldırı ağacında, maliyetler her bir düğüme atanır ve en az maliyetli yol hesaplanır.

Uygulama:

- **Uygulama:** Bir saldırı ağacında her adımın maliyetini hesaplayan bir simülasyon geliştirin. En düşük maliyetle hedefe ulaşmayı simüle edin.

Saldırı Yöntemleri (Attack Methods)

1. Dinamik Analiz (Dynamic Analysis)

Bir programın çalışırken hangi bölümlerinin tetiklendiğini ve hangi girdilerle nasıl davranışlar sergilediğini anlamaya yarar.

Uygulama:

- **Uygulama:** Bir yazılımın çalışma zamanında hangi işlevlerin çağrıldığını izleyen ve bu işlevlerin hangi girdilerle tetiklendiğini gösteren bir izleyici oluşturun.

2. Statik Analiz (Static Analysis)

Bir programın kaynak kodu veya derlenmiş halinin analiz edilmesi işlemidir. Bu analiz ile potansiyel güvenlik açıkları belirlenir.

Uygulama:

- **Uygulama:** Bir disassembler kullanarak, basit bir programın derlenmiş kodunu analiz edin ve zayıf noktaları tespit edin.

3. Program Düzenleme (Editing Phase)

Bir saldırgan, yazılımın iç işleyişini anladıktan sonra, lisans denetimlerini devre dışı bırakmak veya kısıtlamaları kaldırmak için programı düzenleyebilir.

Uygulama:

- **Uygulama:** Lisans denetimini atlamak için bir programın ikili dosyasını düzenleyin. Hangi kısıtlamaların kaldırıldığını izleyin.

Güvenli İletişim Hedefleri

- **Karşılıklı Kimlik Doğrulama:** İletişime giren iki tarafın birbirini doğrulaması.
- **Anahtar İptali:** Geçersiz anahtarların iptal edilmesi.
- **Yüksek Performans:** Güvenli iletişimde hız ve düşük gecikme süresi esastır.

Uygulama:

- **Uygulama:** İki tarafın karşılıklı olarak birbirini doğrulamasını sağlayan basit bir kimlik doğrulama protokolü oluşturun.

Haftanın Özeti ve Gelecek Hafta

Bu Hafta:

- Uygulama Koruma Planı
- Bilgisayar Virüsleri ve Türleri
- Saldırı Ağaçları ve Güvenlik Modelleri
- Saldırı Yöntemleri ve Güvenli İletişim Hedefleri

Gelecek Hafta:

- Veri Güvenliği
- Kriptografik Teknikler
- Uygulamalı Şifreleme

$1. Hafta - Sonu$