

Secure Coding in C and C++ for Medical Devices Eğitimi

Eğitim Hakkında

Secure Coding in C ve C++ 'da Güvenli Kodlama eğitimi, katılımcılara sağlık sektörü bağlamında temel siber güvenlik kavramlarını öğretir. Öğrenciler, C ve C++ 'da güvenli kod en iyi uygulamalarını ve veri ihlallerini önlemek için yaygın güvenlik testi tekniklerinden nasıl yararlanılacağını öğrenirler.

Neler Öğreneceksiniz

Temel siber güvenlik kavramlarına aşina olmayı,
Sağlık sektörünün güvenlik özellikleri hakkında bilgi edinmeyi,
C ve C++ kodunuzdaki güvenlik zorluklarının üstesinden gelmeyi,
Güvenlik açıklarını ve sonuçlarını belirlemeyi,
C ve C++ 'da en iyi güvenlik uygulamalarını öğrenmeyi.

Eğitim İçeriği

Siber Güvenlik Temelleri

Güvenlik nedir?
Tehdit ve risk
Siber güvenlik tehdit türleri

Güvensiz yazılımın sonuçları

Kısıtlamalar ve pazar

Yönetmelikler ve standartlar

Sağlık bilgi sistemleri için düzenlemeler
Tıbbi cihaz düzenlemeleri

Sağlık sektöründe siber güvenlik

Sağlık hizmetlerinde tehditler ve eğilimler
Tıbbi cihazlara yönelik tehditler
Eski sistemlerin sorunu

Arabellek Taşması

Montaj temelleri ve çağrı kuralları
x64 montaj temelleri
Kayıtlar ve adresleme
En yaygın talimatlar
X64'te çağrı kuralları

Bellek yönetimi güvenlik açıkları

Bellek yönetimi ve güvenliği
Gerçek dünyadaki güvenlik açıkları
Tampon güvenlik sorunları
Yığın üzerinde arabellek taşması
Yığın üzerinde arabellek taşması
İşaretçi manipülasyonu

En iyi uygulamalar ve bazı tipik hatalar

Güvenli olmayan işlevler
Güvenli olmayan işlevlerle başa çıkmak
Asctime () ile sorun nedir?
C ++ 'da std :: string kullanma

BOF'ye yol açan bazı tipik hatalar

Sonlandırılmamış dizeler
readlink () ve dize sonlandırma
C ++ 'da C stili dizeleri işleme
Kötü amaçlı dizi sonlandırma
Dize uzunluğu hesaplama hataları
Tek tek hatalar
VxWorks TCP 'Acil Veri' ayrıştırmada tek tek hata
Hiçbir şey tahsis etmemek

Yaygın Yazılım Güvenliği Zayıf Yönleri

Güvenlik özellikleri
Doğrulama
Parola yönetimi

Giriş doğrulama ilkeleri

Kara listeler ve beyaz listeler
Veri doğrulama teknikleri
Örnek olay incelemesi: Natus Xltek NeuroWorks 8'de eksik giriş doğrulaması
Ne doğrulanmalı: saldırı yüzeyi
Nerede doğrulanmalı: derinlemesine savunma
Nasıl doğrulanır: doğrulama ve dönüşümler
Çıktı temizleme
Kodlama zorlukları
Normal ifade ile doğrulama

Injection

Injection prensipleri
Injection saldırıları

Kod yerleştirme

Tamsayı işleme sorunları

İmzalı numaraları temsil etmek

Tamsayı görselleştirme

Tamsayı yükseltme

Tamsayı taşması

İmzalı / imzasız kafa karışıklığı

Tamsayı kesme

Örnek olay: WannaCry

En iyi uygulamalar

Dosyalar ve akışlar

Yol geçişi

Yol geçişi ile ilgili örnekler

Yol geçişi için en iyi uygulamalar

Biçim dizesi sorunları

Printf () ile ilgili sorun

Zaman ve durum

Yarış koşulları

Hatalar

Hata ve istisna işleme ilkeleri

Hata yönetimi

İstisna işleme

Kod kalitesi

Veri işleme

Kontrol akışı

Sinyal işleme

Nesne yönelimli programlama güçlükleri

Bellek ve işaretçiler

Dosya G / Ç

Savunmasız Bileşenleri Kullanma

Çevreyi değerlendirmek

Sertleştirme

Örnek olay incelemesi: Alaris Gateway İş İstasyonunda tedarik zinciri saldırısı

Güvenlik açığı yönetimi

Yama yönetimi

Hata ödül programları

Güvenlik açığı veritabanları

Güvenlik açığı derecelendirmesi - CVSS

DevOps, derleme süreci ve CI / CD

Güvenli olmayan derleyici optimizasyonu

Güvenlik Testi

Güvenlik testi ile işlevsel test karşılaştırması

Manuel ve otomatik yöntemler

Güvenlik testi teknikleri ve araçları
Kod analizi
Dinamik analiz

Güvenli kodlama ilkeleri

Matt Bishop'tan sağlam programlamanın ilkeleri
Saltzer ve Schröder'in güvenli tasarım ilkeleri
Ve şimdi ne olacak?
Yazılım güvenlik kaynakları ve daha fazla okuma
C ve C ++ kaynakları