C++ İLE GÜVENLİ KODLAMA EĞİTİMİ

3 GÜN

Digital Vizyon Akademi

www.digitalvizyon.net

İçindekiler

Eğitim Hakkında

C++ ile Güvenli Kodlama Eğitimi, C++ programlama dili kullanılarak yazılan kodların güvenli bir şekilde nasıl tasarlanacağını ve geliştirileceğini kapsamlı bir şekilde ele alan, programcılara yönelik değerli bir eğitim sunar. Eğitim, temel güvenlik prensipleri, güvenlik açıklarının tespiti, güvenli kodlama uygulamaları ve araçların etkili kullanımı konularında derinlemesine bilgi sağlar.

Güvenli kodlama, yazılım güvenliğinin önemli bir parçasıdır ve C++ ile güvenli kodlama eğitimi, kod güvenliğini sağlamak için gereken en iyi uygulamaları öğreten kritik bir kaynaktır. Bu eğitim, özellikle bellek yönetimi, veri doğrulama, hata işleme ve güvenlik politikaları gibi konularda C++ özelinde ipuçları ve teknikler içerir. Eğitimin içeriği, katılımcılara, güvenlik açıklarını nasıl önleyeceklerini ve var olan güvensiz kodları nasıl güvenli hale getireceklerini göstererek, onların profesyonel beceri setini zenginleştirir.

Ayrıca, eğitim sırasında katılımcılara, güvenlikle ilgili sorunları tespit etmeye ve çözmeye yardımcı olacak araçların kullanımı aktarılır. Bu araçlar, statik kod analizi ve dinamik analiz araçları gibi yazılım güvenliği açısından hayati öneme sahip olan araçları içerir. C++ ile güvenli kodlama eğitimi, bu araçları nasıl etkili bir şekilde entegre edecekleri ve kullanacakları konusunda katılımcılara yol gösterir.

Bu eğitim, aynı zamanda katılımcıların kodlarını sürekli olarak güncelleme ve test etmeleri için gereken becerileri de kazandırır. Bu sürekli geliştirme süreci, potansiyel güvenlik açıklarının erken tespit edilmesini ve bu açıkların etkili bir şekilde ele alınmasını sağlar.

Özetle, C++ ile güvenli kodlama eğitimi, programcıları güvenli yazılım geliştirme konusunda yetiştirir ve onlara, modern güvenlik standartlarına uygun olarak güvenli, verimli ve hatasız kod yazma becerilerini kazandırır. Bu, programcıların işlerinde daha başarılı olmalarına ve yazılım projelerini daha güvenli bir şekilde yönetmelerine olanak tanır. C++'ın karmaşık yapısından kaynaklanan güvenlik açıklarını anlama ve bu açıkları minimize etme becerisine sahip programcılar, endüstride daha çok talep gören profesyoneller haline gelir.

Neler Öğreneceksiniz?

C++ ile güvenli kodlama eğitiminde, şu konuların bir kısmı öğrenebilirsiniz:

- Güvenli kod yazma prensipleri: Güvenli kod yazmak için gerekli olan prensipleri öğrenebilirsiniz, bunlar arasında input doğrulama, sanal makine kullanımı, açık kaynak kodlu yazılımların kullanımı gibi konular bulunur.
- Güvenli kod yazma teknikleri: Kodlarınızı nasıl güvenli hale getirebileceğinizi öğrenebilirsiniz, bu teknikler arasında buffer overflow, SQL injection, cross-site scripting gibi tehlikelere karşı nasıl korunabileceğiniz gibi konular bulunur.
- Güvenli kod yazma araçları: Güvenli kod yazmak için kullanabileceğiniz araçlar hakkında bilgi edinebilirsiniz, bu araçlar arasında memory leak bulucular, güvenli kod analiz araçları gibi konular bulunur.

- Güvenli kod yazma standartları: Güvenli kod yazma standartları hakkında bilgi edinebilirsiniz ve bu standartları nasıl uygulayabileceğinizi öğrenebilirsiniz.

Bu konular sadece bir örnek olarak verilmiştir ve her eğitim programı farklı konuları ve ayrıntıları içerebilir. Ancak, bu konular genel olarak C++ ile güvenli kodlama eğitiminin içeriğini oluşturur.

Ön Koşullar

C++ ile güvenli kodlama eğitimi için ön koşullar şunlar olabilir:

- Programlama deneyimi: C++ ile güvenli kodlama eğitimi, C++ programlama dili hakkında temel bilgi sahibi olmanızı gerektirir. Öncelikle C++ programlama dili hakkında temel bilgi sahibi olmanız önkoşul olarak kabul edilir.
- Programlama pratiği: Programlama ile ilgili teorik bilgilerin yanı sıra, bu bilgilerin uygulamalı olarak nasıl kullanılabileceği hakkında da bilgi sahibi olmanız önkoşul olarak kabul edilir.
- İşletim sistemi ve yazılım: Eğitim programının gerektirdiği işletim sistemi ve yazılım araçlarını kullanabilmeniz gerekmektedir. Örneğin, eğitim programı C++ ile güvenli kodlama yapmak için bir Integrated Development Environment (IDE) gerektirebilir.

Bu önkoşullar genel bir örnek olarak verilmiştir ve her eğitim programı için farklı önkoşullar bulunabilir. Eğitim programına kayıt yapmadan önce, önkoşullar hakkında eğitim programını sunan kuruluştan bilgi alabilirsiniz.

Kimler Katılmalı

C++ ile güvenli kodlama eğitimi, şu kişiler katılabilir:

- C++ programcıları: Eğitim, C++ programcılarının güvenli kod yazma becerilerini geliştirmelerine yardımcı olacaktır.
- Güvenlik uzmanları: Eğitim, güvenlik uzmanlarının C++ ile güvenli kod yazma becerilerini geliştirmelerine yardımcı olacaktır.
- Bilgi güvenliği profesyonelleri: Eğitim, bilgi güvenliği profesyonellerinin C++ ile güvenli kod yazma becerilerini geliştirmelerine yardımcı olacaktır.
- İlgilenen herkes: Programlama ve güvenlik konularına ilgi duyan herkes, C++ ile güvenli kodlama eğitiminden yararlanabilir.

Bu liste genel bir örnek olarak verilmiştir ve her eğitim programına katılacak olanların profesyonel deneyim ve yeteneklerine göre farklılık gösterebilir. Eğitim programına kayıt yapmadan önce, kendinize en uygun olan programı seçmeniz önerilir.

Outline

Security

- Types of attacks: denial of service and data mining
- Vectors of attack: network, libraries, malware
- Defense in depth
- Classification of security flaws
- What Could Possibly Go Wrong?
- Always ask: what happens if this fails?
- What happens if the application crashes?
- What happens if an exception is thrown?
- Network problems?
- Operating system crashes?
- Protections failure (firewall, physical security, etc)
- What about programs launched from the application?
- Where does the application fail to?
- Fail securely

Coding Vulnerabilities

- Input validation: XML injection, SQL injection, path traversal, log forging
- Race Conditions: time-of-check to time-of-use. memory corruption
- Time and state
- Variable parameters
- Error and exception handling
- Automatic and controlled data conversions
- Memory locking, threads, and semaphores
- File Handling
- Cryptography
- Symmetric-key
- Asymmetric-key
- Hashing
- The dependency of randomization
- Password and key management
- Passwords and keys in memory

Client Authentication

- Web – basic
- Web – digest
- Biometrics
- Cryptographic
- Two-factor authentication

- Data Overflow
- Buffer overflow
- Array indexing
- Stack overflow & Stack smashing
- Overflow and index on the heap and the stack

Security Design Principles

- Fail-safes
- Mediation: did the data change since last checked?
- Separation of privileges
- Least privilege
- Psychological Acceptability
- CERT and Design Principles
    - CERT C++ coding standards
    - Addressing CERT requirements
    - Object-oriented design principles and design patterns
    - Testing, unit testing, and test-driven-development

Intel Architecture

- Processors, registers, memory
- Function calling conventions
- Stack frame & non-executable (NX) memory areas
- Recursion
- Address space layout randomization

Third-Party Code

- Any code that is not your own, including other internal groups
- Package management
- Vetting third-party code: source, reverse compilers
- Monitoring network connections