Secure coding in C and C++ for medical devices

Category: Networking & Security | **Vendor:** Various

Duration: 32.00 hours (4 days) **26.0 CPD Hours** **Rating:** ⭐ 4.6 (5,878 reviews)

Course Information

Delivery Format: Instructor Led - Online

Course Overview

Your application written in C and C++ works as intended, so you are done, right? But did you consider feeding in incorrect values? 16Gbs of data? A null? An apostrophe? Negative numbers, or specifically -1 or -2^31? Because that’s what the bad guys will do – and the list is far from complete.

The most important concern in the healthcare industry is naturally safety. However, once isolated medical devices became highly connected to date, which poses new kinds of security risks: from exposing sensitive patient information to denial of service. And remember, there is no safety without security!

Handling security needs a healthy level of paranoia, and this is what this course provides: a strong emotional engagement by lots of hands-on labs and stories from real life, all to substantially improve code hygiene. Mistakes, consequences, and best practices are our blood, sweat and tears.

All this is put in the context of medical devices developed in C and C++, and extended by core programming issues, discussing security pitfalls of these languages.

So that you are prepared for the forces of the dark side.

So that nothing unexpected happens.

Nothing.

About This Course

Your application written in C and C++ works as intended, so you are done, right? But did you consider feeding in incorrect values? 16Gbs of data? A null? An apostrophe? Negative numbers, or specifically -1 or -2^31? Because that’s what the bad guys will do – and the list is far from complete.

The most important concern in the healthcare industry is naturally safety. However, once isolated medical devices became highly connected to date, which poses new kinds of security risks: from exposing sensitive patient information to denial of service. And remember, there is no safety without security!

Handling security needs a healthy level of paranoia, and this is what this course provides: a strong emotional engagement by lots of hands-on labs and stories from real life, all to substantially improve code hygiene. Mistakes, consequences, and best practices are our blood, sweat and tears.

All this is put in the context of medical devices developed in C and C++, and extended by core programming issues, discussing security pitfalls of these languages.

So that you are prepared for the forces of the dark side.

So that nothing unexpected happens.

Nothing.

Who Should Attend

C/C++ developers developing medical devices

Prerequisites & Entry Requirements

General Prerequisites:

General C/C++ development

Learning Outcomes

Upon successful completion of this course, participants will be able to:

- Getting familiar with essential cyber security concepts
- Learning about security specialties of the healthcare sector
- Identify vulnerabilities and their consequences
- Learn the security best practices in C and C++
- Input validation approaches and principles

Detailed Course Outline

Day 1

- Cyber security basics
- What is security?
- Threat and risk
- Cyber security threat types – the CIA triad
- Cyber security threat types – the STRIDE model
- Consequences of insecure software
- Regulations and standards
- Healthcare data protection regulations
- Regulations for medical devices
- Cyber security in the healthcare sector
- Threats to medical devices
- Attackers and motivation
- The problem of legacy systems

Memory management vulnerabilities

- Assembly basics and calling conventions

- x64 assembly essentials
- Registers and addressing
- Most common instructions
- Calling conventions on x64
- Buffer overflow
- Memory management and security
- Vulnerabilities in the real world
- Buffer security issues
- Buffer overflow on the stack

Day 2

Memory management vulnerabilities

- Best practices and some typical mistakes
- Unsafe functions
- Dealing with unsafe functions
- Lab – Fixing buffer overflow
- What's the problem with asctime()?
- Lab – The problem with asctime()
- Using std::string in C++
- Some typical mistakes leading to BOF
- Unterminated strings
- readlink() and string termination
- Manipulating C-style strings in C++
- Malicious string termination
- Lab – String termination confusion
- String length calculation mistakes
- Off-by-one errors
- Case study – Off-by-one error in VxWorks TCP 'Urgent Data' parsing
- Allocating nothing

Memory management hardening

- Securing the toolchain
- Securing the toolchain in C and C++
- Compiler warnings and security
- Using FORTIFY_SOURCE
- Lab – Effects of FORTIFY
- AddressSanitizer (ASan)
- RELRO protection against GOT hijacking
- Heap overflow protection
- Stack smashing protection
- Runtime protections
- Runtime instrumentation
- Address Space Layout Randomization (ASLR)
- Non-executable memory areas

Common software security weaknesses

- Security features
- Authentication
- Authorization

Day 3

Common software security weaknesses

- Security features (continued)
- Password management

Common software security weaknesses

- Input validation
- Input validation principles
- Denylists and allowlists
- Case study – Improper input validation in Natus Xltek NeuroWorks 8
- What to validate – the attack surface
- Where to validate – defense in depth
- When to validate – validation vs transformations
- Output sanitization
- Encoding challenges
- Unicode challenges
- Validation with regex
- Regular expression denial of service (ReDoS)
- Lab – ReDoS in C
- Dealing with ReDoS
- Integer handling problems

Day 4

Common software security weaknesses

- Input validation
- Injection
- Process control
- Files and streams
- Format string issues

Time and state

- Race conditions

Errors

- Error and exception handling principles

- Error handling

- Exception handling

Code quality

- Code quality and security

- Data handling

- Object oriented programming pitfalls

Wrap up

- Secure coding principles

- And now what?

Skills You Will Learn:

- Getting familiar with essential cyber security concepts
- Learning about security specialties of the healthcare sector
- Identify vulnerabilities and their consequences
- Learn the security best practices in C and C++
- Input validation approaches and principles

Frequently Asked Questions

Q: What delivery options are available for Secure coding in C and C++ for medical devices?

We offer multiple delivery formats:

- Live Instructor-Led Classroom Online (Virtual/Live Online)
- Traditional Instructor-Led Classroom Training (ILT)
- On-site delivery at your offices anywhere in United Kingdom
- Private dedicated courses customized for your team

Q: How many CPD hours does this course provide?

The 4-day Secure coding in C and C++ for medical devices course provides up to 26.0 CPD hours of structured learning. CPD certificates can be provided upon request.

Q: What is the duration of the Secure coding in C and C++ for medical devices training?

The training takes place over 4 day(s), with each day lasting approximately 32.00 hours including breaks for lunch and refreshments.

Q: Do you provide corporate training for Secure coding in C and C++ for medical devices?

Yes, we provide corporate training, dedicated training, and closed classes for Secure coding in C and C++ for medical devices. Training can take place anywhere in United Kingdom including London, Manchester, Birmingham, Edinburgh, or live online allowing teams from across United Kingdom or internationally to attend.

Q: What related terms do people search for?

Popular related searches include: Secure coding in C and C++ for medical devices, Secure Coding, SECC-CCPPMD

Q: Why choose Nexus Human for Secure coding in C and C++ for medical devices?

Nexus Human is recognized as one of the leading training providers. Our trainers have won multiple awards including:

- Small Firms Best Trainer Award
- National Training Partner of the Year (Ireland) - Multiple Years
- Global Top 30 Instructor Awards (2012, 2019, 2021)
- Tech Excellence Award Nominations
- Learning Performance Institute (LPI) External Training Provider Sponsor 2024

Q: Are there any discount codes available?

Yes! Use discount code **PENPAL5** when booking your Secure coding in C and C++ for medical devices training. Please note that only one discount code can be used per booking and cannot be combined with other special offers.

Nexus Human

Professional Training & Development

Email: info@nexushuman.com

Website: www.nexushuman.com

Phone: +353 1 889 8700 (Ireland) | +44 20 8142 8961 (UK)

© 2026 Nexus Human. All rights reserved. This brochure was generated on 13/11/2026.