

Tutorials

Monday, March 13, 2006

Sheraton National Hotel

8:00 AM - 5:00 PM	T1: Secure Coding in C and C++
8:00 AM - 5:00 PM	T2: Security Patterns and Secure Systems Design Using UML
8:00 AM - 12:00 PM	T3: SaFSec – Integrating Safety and Security Assurance
1:00 PM - 5:00 PM	T4: Correctness by Construction – A Manifesto for High Integrity Engineering

T1: Secure Coding in C and C++

8:00 AM - 5:00 PM

Presenter

Robert C. Seacord, Senior Vulnerability Analyst, SEI/CMU

Audience

The Secure Coding in C and C++ tutorial should be useful to anyone involved in the development or maintenance of software in C and C++.

- For a C/C++ programmer, this tutorial will teach you how to identify common programming errors that result in software vulnerabilities, understand how these errors are exploited, and implement a solution in a secure fashion.
- For a software project manager, this tutorial identifies the risks and consequences of software vulnerabilities to guide investments in developing secure software.

Abstract

Secure Coding in C and C++ provides practical advice on secure practices in C and C++ programming. Producing secure programs requires secure designs. However, even the best designs can lead to insecure programs if developers are unaware of the many security pitfalls inherent in C and C++ programming. This tutorial provides a detailed explanation of common programming errors in C and C++ and describes how these errors can lead to code that is vulnerable to exploitation. The tutorial concentrates on security issues intrinsic to the C and C++ programming languages and associated libraries. It does not emphasize security issues involving interactions with external systems such as databases and web servers, as these are rich topics on their own. The intent is that this tutorial be useful to anyone involved in developing secure C and C++ programs regardless of the specific application.

- For a computer science student, this tutorial will teach you programming practices that will help you to avoid developing bad habits and enable you to develop secure programs during your professional career.
- For a security analyst, this tutorial provides a detailed description of common vulnerabilities, identifies ways to detect these vulnerabilities, and offers practical avoidance strategies.

Software Vulnerabilities are common and pervasive. As technology advances in to the most remote areas of our life, it is important to understand and learn from the past coding practices. Software

companies are quickly recognizing the effect that vulnerabilities pose to the customer, and their own market success. Major software companies have begun to require that their software developers take courses specifically on secure coding practices.

Prerequisites

The Secure Coding in C and C++ tutorial requires a basic to intermediate understanding of the C and C++ programming languages. It does not require an understanding of application security.

Presenter Biography

Robert C. Seacord is a senior vulnerability analyst at the CERT/Coordination Center (CERT/CC) at the Software Engineering Institute (SEI) in Pittsburgh, PA. Robert is the author of Secure Coding in C and C++ (Addison-Wesley, 2005) and coauthor Building Systems from Commercial Components (Addison-Wesley, 2002) and Modernizing Legacy Systems (Addison-Wesley, 2003) as well as more than 40 papers on software security, component-based software engineering, Web-based system design, legacy-system modernization, component repositories and search engines, and user interface design and development.

Robert is part time faculty member at the University of Pittsburgh where he has taught Software Engineering

Robert started programming professionally for IBM in 1982, working in communications and operating system software, processor development, and software engineering. Robert also has worked at the X Consortium, where he developed and maintained code for the Common Desktop Environment and the X Window System. He also is actively involved in the JTC1/SC22/WG14 international standardization working group for the C programming language.

T2: Security Patterns and Secure Systems Design Using UML

8:00 AM - 5:00 PM

Presenter

Prof. Eduardo B. Fernandez, Florida Atlantic University

Abstract

Analysis and design patterns are well established as a convenient and reusable way to build high-quality object-oriented software. Patterns combine experience and good practices to develop basic models that can be used for new designs. Security patterns join the extensive knowledge accumulated about security with the structure provided by patterns to provide guidelines for secure system design and evaluation. We show a variety of security patterns and their use in the construction of secure systems. These patterns include Authentication, Authorization, Role-based Access Control, Firewalls, Protected Execution Environment, and others. We combine some of these patterns to build Single-Sign-On architectures, web services authorization, authorized applications, and others. We apply these patterns through a secure system development method that use different mechanisms based on a hierarchical architecture whose layers define the scope of each security mechanism. First, the possible attacks to the application are defined from the actions in each use case. Then the rights of the users are defined from extended Use Cases using a Role-Based Access Control (RBAC) model. These rights are then reflected in the conceptual class model. We then define additional security constraints that apply to distribution and concurrency aspects, as well as to user interfaces. We use a catalog of security patterns that help defining the security mechanisms at each architectural level and at each development stage. The patterns are shown using UML models and examples are taken from our forthcoming book "Security Patterns". Attendees will be able to understand security patterns and how can they be used to build secure systems. .

Outline of tutorial:

- Introduction
- Internet security issues---recent attacks, vulnerabilities, threats
- Object-oriented design and patterns--- need for good software engineering, analysis and design patterns
- A methodology for secure systems design
- Security models and their patterns---policies, access matrix, multilevel models, RBAC
- Relating attacks to use cases.
- Defining authorizations from use cases---nonfunctional aspects of use cases, RBAC and security policies
- Authorized conceptual model, UML as an access control model
- Secure system architectures---effect of distribution and user interfaces, firewalls, and IDS.
- Web application servers and components---mapping RBAC to components, J2EE and .NET
- Patterns for web services
- Coordination across levels---mapping of authorizations across architectural levels
- Analysis of a case study (open discussion)
- Conclusions

Presenter: Prof. Eduardo B. Fernandez, Professor, Dept. of Computer Science and Eng., Florida Atlantic University

Eduardo B. Fernandez (Eduardo Fernandez-Buglioni) is a professor in the Department of Computer Science and Engineering at Florida Atlantic University in Boca Raton, Florida. He has published numerous papers on authorization models, object-oriented analysis and design, and fault-tolerant systems. He has written three books on these subjects. He has lectured all over the world at both academic and industrial meetings. He has created and taught several graduate and undergraduate courses and industrial tutorials. His current interests include patterns for object-oriented design and web services security. He holds a MS degree in Electrical Engineering from Purdue University and a Ph.D. in Computer Science from UCLA. He is a Senior Member of the IEEE, and a Member of ACM. He is an active consultant for industry, including assignments with IBM, Allied Signal, Motorola, Harris, Lucent, and others. He is also a frequent proposal reviewer for NSF. He is a co-author of M.Schumacher, E.B.Fernandez, F. Buschmann, D. Hybertson, and P. Sommerlad, *Security Patterns*, Wiley, 2006.

T3: SaFSec – Integrating Safety and Security Assurance

8:00 AM - 12:00 PM

Presenter

Samantha Lautieri, SafSec Project Manager

Audience

Developers – managers and engineers – and Purchasers – managers and integrators – of

- products;

- sub-systems;
- systems of systems;

Who desire an insight into how they can develop and integrate sub-systems and products using a risk directed approach including learning

- how safety and security objectives and processes overlap;
- how to achieve assurance (and certification) of a safer, more secure system at lower cost;
- how integrating COTS or module upgrades into a system can be made easier and reduce re-certification effort and costs.

Abstract

SafSec is a methodology that was developed at Praxis HIS, United Kingdom, for, and funded by, the Ministry of Defence (MoD) and key contractors. The objectives for the SafSec Methodology were to:

- integrate safety and security certification practices to reduce effort and cost
- to tackle the problems posed by upgrades, obsolescence and integrated modular avionics.

The SafSec methodology is contained within a standard and supplemented by a guidance document that has been produced as a result of:

- consultation and review with a wide variety of stakeholders, including regulators and policy authors, main defence contractors, procurers, and academics; and
- validation on two case studies where the participating companies both had interests in the success of a SafSec type methodology and are main contractors to UK MoD.

A further explanation of SafSec can be found on the SafSec website www.safsec.com.

Contents

The objectives of the tutorial are to impart knowledge on how safety and security analysis and development processes can be practiced in an integrated fashion that will maximise the effectiveness of the design assurances and minimise the cost.

The objectives will be met by:

- 1 Providing a short background to both the SafSec project and the methodology.
- 2 Explaining the key concepts of the SafSec Methodology, including the use of a module boundary contract to aid in the re-use and re-certification of the product and/or system components.
- 3 Explaining how current best practice can be used to design, develop and demonstrate a system that is adequately safe and secure for the context of use.
- 4 Demonstrating how following the SafSec Methodology will assist with compliance against particular safety and security standards.

The tutorial will address use of the SafSec Standard and Guidance Document that is available at www.safsec.com from the resources page.

Presenter Biography

Samantha Lautieri, SafSec Project Manager, Praxis High Integrity Systems has been a technical project manager on the SafSec project for more than 3 years. She was involved in the original research and the first definition of the SafSec Methodology and has been an integral part in the validation of the methodology on 2 case studies as well as the most recent updates to the methodology and guidance.

T4: Correctness by Construction – A Manifesto for High Integrity Engineering

1:00 PM - 5:00 PM

Presenters

Peter Amey & Rod Chapman, Praxis High Integrity Systems

Audience

Technical and managerial professionals involved in systems engineering, software engineering, or safety and security who desire knowledge of engineering methods to realize low defect rates and high resilience to change for long-life high integrity systems

Tutorial Abstract:

Two of the major recurring issues in systems and software engineering are unacceptably high defect rates, and lack of ability to make changes over a significant period of time in a reliable manner. These issues frequently cause major impact on the usability and maintainability of systems, and this impact becomes intolerable when components of the system become unmaintainable, or when high levels of security accreditation or safety certification must be achieved.

This tutorial presents the Correctness by Construction high integrity approach to systems and software engineering. Correctness by Construction is based on a set of principles, distilled from practical project experience, to realize systems and software engineering outputs with very low defect rate and very high resilience to change. These principles can be applied most effectively to new developments and upgrades. However some of the same principles can also be applied retrospectively to improve the maintainability and upgradability of existing systems.

Topics include:

1. The challenges that we face
2. How Correctness by Construction addresses the challenges
 - a) Validating the concepts of operation
 - b) Risk-directed requirements engineering
 - c) Formal specification
 - d) Rigorous design and implementation based on static analysis
 - e) Making the case for safety and security
 - f) Maintaining correctness through long-term upgrades

3. Correctness by Construction in action – project examples

Why attendees will benefit from the Correctness by Construction tutorial

The challenge of producing software that has an acceptably-low defect rate is hard enough. To do so under time and budget pressures is harder still. Yet the pressures grow ever stronger and the trust we place in software-intensive systems continues to rise. There is a pressing need to find ways of producing high-integrity software at an acceptable cost.

Over the past 16 years, Praxis has evolved a practical approach to software development which delivers exceptionally low defect rates with very high productivity. The approach eschews the use of fashionable technologies and silver bullets preferring an engineering approach which favours error prevention and relentless error elimination at every stage of development. The result is a development process that generates a continuous forward flow from requirements to implementation.

The tutorial will describe how this approach, which has come to be called Correctness by Construction works and how it has been applied to significant, real-world projects.

Presenter Biographies:

Peter Amey is an aeronautical engineer by original professional training. He served as an engineering officer in the Royal Air Force and spent several years at the Boscombe Down test establishment working on the certification of aircraft armament systems. Peter joined Program Validation Limited in 1992 to develop SPARK and the SPARK Examiner and continues that work today with Praxis High Integrity Systems. As well developing SPARK he has used it on major programmes including Tornado, Eurofighter and the Lockheed C130J. Peter, now Chief Technical Officer for software engineering at Praxis, teaches SPARK and Ada on a regular basis and has lectured widely on the development of critical systems.

Roderick Chapman received MEng and DPhil degrees from the University of York, England in 1991 and 1995 respectively. He is currently products manager at Praxis Critical Systems, leading the design and development of the SPARK language and toolset. Before joining SPARK team, Rod was involved in the implementation high-integrity real-time and embedded systems, including SHOLIS (the first system implemented to the Def Stan 00-55 SIL4 standard), the Lockheed Martin C130J Mission Computer, and the MULTOS CA. Rod has presented tutorial, papers and panel sessions at many conferences, including SIGAda, Ada Europe, and STC, and remains a member of the Ada95 HRG.

Both presenters were featured in the September 2005 issue of IEEE Spectrum in the feature article "The Exterminators"