Training Catalogue
22/05/2020

KAÏNA-COM

TRAINING CATALOGUE

From threats to code

KSE005 – From threats to code

Reference	KSE005
Experience	☒ Beginner ☒ Intermediate □ Advanced
Duration	Training Program: • 2 days
Training Method	□ I: i-learning, individual training (web-based training) ☒ V: v-learning, virtual class □ C: c-learning, classroom training **KAÏNA-COM** LE CARRÉ HAUSSMANN II, 6 Allée de la Connaissance 77127 Lieusaint - France
Price	1.390,50 € HT
Prerequisite	Participants will have a solid understanding of TCP/IP networking, and be proficient in at least one programming language – C/C++, C#, PHP, Java or JavaScript.
Audience	If you develop software products that attach to a network – products such as medical devices, SaaS applications or mobile medical apps – you should attend.

Continued on next page

KSE005 – From threats to code, Continued

Objective

"From threats to code" is a concentrated, fast-moving, introduction to developing secure code for the entire software development team from program manager to implementation engineer.

We introduce a threat-analytic approach based on understanding what threats really count and in the second day, we dive into right software security assessment and secure coding to mitigate threats such as Shellcode and buffer overflow attacks.

Continued on next page

KSE005 – From threats to code, Continued

Course Contents

Course Contents :

Day #1 - An Introduction to threat modeling and analysis

Table 1: KSE005 - Course Contents (Day#1)

Chapter	Description
Ideology	• Why bother modeling? • Why security defenses don’t work • Why risk management is broken • Bridging the valley of death between IT and security • A secure SDLC (software development life-cycle) for an unsecure world
Security metrics	• Escaping the hamster wheel of pain • Defining security metrics – What makes a good metric, bad metric, what is not a metric? – Modelers versus measurers
How to measure anything	• Asset valuation • Threat damage to asset • Probability of occurrence
Threat modeling and analysis objectives and drivers	• Qualitative or quantitative? • Is there ROI on security? • Compliance drivers: Industry, Government, Vendor-neutral standards
Threat modeling building blocks	• Threats / attack scenarios • Assets • Vulnerabilities • Countermeasures – Encryption – Network monitoring – Auditing activity logs and data flows – Input validation – Error handling

Continued on next page

KSE005 – From threats to code, Continued

Course Contents, continued

Chapter	Description
Analyzing your threat model	• Analyzing your threat model and building a cost-effective security countermeasure plan
Pulling it all together	• A class exercise
Software vulnerability fundamentals	• Vulnerabilities – Security Policies – Security expectations • Classifying vulnerabilities – Design vulnerabilities – Implementation vulnerabilities – Operational vulnerabilities – Gray areas • Common threads – Input and data flow – Trust relationships – Assumptions and misplaced trust – Interfaces – Environmental attacks – Exceptional conditions

Continued on next page

KSE005 – From threats to code, Continued

Course Contents, continued

Day #2 – An Introduction to secure coding

Table 2: KSE005 - Course Contents (Day#2)

Chapter	Description
Design review	• Software design fundamentals – Algorithms – Abstraction and decomposition – Trust relationships – Principles of software design – Fundamental design flaws • Enforcing security policy – Authentication – Authorization – Accountability – Confidentiality – Integrity – Availability • Threat modeling of software – Data collection – Attack trees – Prioritizing
Operational review	• Exposure – Attack surface – Insecure defaults – Access control – Unnecessary services – Secure channels – Spoofing – Network profiles • Countermeasures – Development-based – Host-based – Network-based

Continued on next page

KSE005 – From threats to code, Continued

Course Contents, continued

Chapter	Description
Software vulnerabilities	• Buffer overflows – Process memory layout – Stack overflows – Off-by-one errors – Heap overflows – Global and static data overflows • Shellcode – Writing the code – Finding your code in memory • Protection mechanisms – Stack cookies – Heap hardening – Non-executable stack and help protection • Address space layout – Randomization – SafeSEH – Function pointer obfuscation
Windows objects and the file system	• Processes and threads – Process loading – ShellExecute and ShellExecuteEx – DLL loading – Services • File access – File permissions – File IO API – Links
Windows messaging	• Window messages • Shatter attack

Continued on next page

KSE005 – From threats to code, Continued

Course Contents, continued

Chapter	Description
Network vulnerabilities in practice	• TCP connections, an overview • TCP streams – TCP spoofing – Connection fabrication – Connection tampering – Blind reset attacks – Blind data injection attacks – TCP segment fragmentation spoofing
The End	• Summary • Q&A • Course’s Evaluation