Real Time College

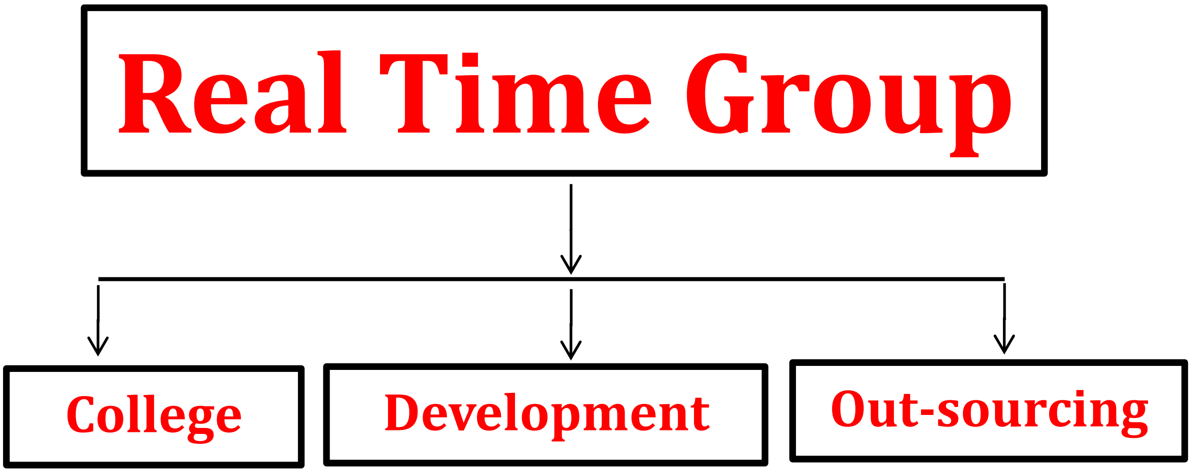
Course: Security

Duration: 10 Hours
Hands-On-Training: 0%

Real Time Group is a multi-disciplinary dynamic and innovative Real-Time O.S. and Embedded Software Solutions Center, established in 2007.

Providing Bare-Metal and Embedded Linux solutions, professional services and consulting, end-to-end flexible system infrastructure, outsourcing, integration and training services for Hardware, Software and RT-OS \ Embedded Systems.

The company is divided into the following three Divisions:



rt-ed.co.il rt-dev.com rt-hr.co.il

Training Division:

Professional Training Services for Hardware, Software, RT-OS and Embedded systems industries.

We provide the knowledge and experience needed to enable professional engineers to Develop, Integrate and QA Hardware, Software and Networking Projects.

In order to insure experience, all courses are practical – hands-on-training.
The latest Development, QA and Automation equipment which are adopted by the industry are used.

All students are supplied with Development-Boards for home-work and course projects.

Course Overview:

Web applications are inherently insecure, as aptly illustrated by a pile of recent events. Insecurity is however not fundamental to the web platform. As a matter of fact, the modern web offers a variety of powerful security features that help stop a hacker. Unfortunately, not many developers have the knowledge and skills to leverage these security features to their full potential.

Who should attend:

- Developers of WEB applications client side.
- Developers of WEB applications sever side.

Prerequisite:

- Understand how HTTP comunication works
- Basic knowledge of Web vulnearabilities
- Basic knowledge of Linux and Computer usage
- Node JS, Angular/React

Security

1 Security threats

a. Injection
b. Broken authentication
c. Sensitive data exposure
d. XML external entities (XXE)
e. Broken access control
f. Security misconfiguration
g. Cross-site scripting (XSS)
h. Insecure deserialization
i. Using components with known vulnerabilities
j. Insufficient logging and monitoring

2 Security standards

3 Security technologies overview

4 Best practices and recommendation

a. Client side Angular/React Best practices recommendation
b. Node JS Best practices recommendation
c. Server (networking) practices recommendation