|  |  |  |  |
| --- | --- | --- | --- |
| **Last Edition:** | 23 November 2018 | **Version No.** | 18.1 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1.** | **Course Title:** | | | | | Web Application Security (WAS) | | | | | | | | | | | |
| **2.** | **Course Code:** | | | | | INFANL01-9 | | | | | | | | | | | |
| **3.** | **Course Team:** | | | | | Babak Basharirad (Course Leader) | | | | | | | | | | | |
|  |  | | | | | Second Teacher | | | | | | | | | | | |
|  |  | | | | | Third Teacher | | | | | | | | | | | |
| **4.** | **Rationale and Synopsis:** | | | | | | | | | | | | | | | | |
|  | quantity and importance of data entrusted to web applications is growing, and defenders need to learn how to secure them. Traditional network defences, such as firewalls, fail to secure web applications. This course introduces these potential risks and helps students to better understand web application vulnerabilities, thus enabling them to properly defend organizations’ web assets. | | | | | | | | | | | | | | | | |
| **5.** | **Year and Semester offered:** | | | | | Year 2 / Sem 1 | | | | | | | | | | | |
| **6.** | **Prerequisite:** | | | | | Introduction to Web Programming | | | | | | | | | | | |
| **7.** | **Credit Value:** | | | | | 3 EC | | | | | | | | | | | |
| **8.** | **Student Learning Time (SLT)** [hours] | | | | | | | | | | | | | | | | |
|  | L = Lecture  T = Tutorial  P = Practical  V = Virtual Learning  A = Assessment  O = Other | | | | Face to Face | | | | | | | | Guided Learning | | Ind.  Learning | | Total  Learning  Time |
|  |  | | | | L | T | P | V | | A | O | | Total | |  | |  |
|  |  | | | | 7 | - | 14 | - | | 3 | - | | 24 | | 54 | | 78 |
| **9.** | **Learning outcomes:** | | | | | | | | | | | | | | | | |
|  | On completion of this module, students will be able to:   1. Apply a detailed methodology to your web application penetration tests. 2. Analyse the results from automated web testing tools to remove false positives and validate findings. Manually discover key web application flaws. 3. Analyse traffic between the client and the server application using proxy tools to find security issues within the client-side application code. 4. Create configurations and test payloads within other web attacks. | | | | | | | | | | | | | | | | |
| **10.** | **Assessment \*:** | | | | | | | | | | | | | | | | |
|  |  | | | | | | | | Learning Objectives for assessment | | | | | | | | |
|  | Class Test |  | - | % | | | | | LO 1 | | | LO 2 | | LO 3 | | LO 4 | |
|  | Final Exam |  | 100 | % | | | | | LO 1 | | | LO 2 | | LO 3 | | LO 4 | |
|  | Assignment |  | - | % | | | | | LO 1 | | | LO 2 | | LO 3 | | LO 4 | |
|  | \* regardless of assessment type, students need to obtain 50% of marks for each LO to successfully pass the module. | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **11.** | **Content of the module and the SLT per topic** [hours]**:** | | | | | | |
|  | **Week** | | **Topics** | | | Class | Ind. |
|  | **1** | | **Introduction** | | | **3** | **6.45** |
|  |  | |  | * HTTP * Sessions * HTTPs | |  |  |
|  | **2** | | **Passing Data to Subsystems** | | | **3** | **6.45** |
|  |  | |  | * SQL Injection * Shell Command Injection * Talking to Programs Written in C/C++ | |  |  |
|  | **3** | | **User Input** | | | **3** | **6.45** |
|  |  | |  | * Definition * Validating Input * Handling Invalid Input | |  |  |
|  | **4** | | **Output Handling: The Cross-site Scripting Problem** | | | **3** | **6.45** |
|  |  | |  | * Introduction * The Problem and the Solution * Browser Character Sets | |  |  |
|  | **5** | | **Web Trojans** | | | **3** | **6.45** |
|  |  | |  | * Introduction * The Problem and the Solution | |  |  |
|  | **6** | | **Passwords and Other Secrets** | | | **3** | **6.45** |
|  |  | |  | * Crypto-Stuff * Password-based Authentication * Secret Identifiers * Secret Leakage * Availability of Server-side Code | |  |  |
|  | **7** | | **Enemies of Secure Code** | | | **3** | **6.45** |
|  |  | |  | * Ignorance * Mess * Deadlines | |  |  |
|  | **8** | | **Exam and Assignment Submission** | | | **3** | **6.45** |
|  |  | |  | * Written Exam | |  |  |
|  | **Total SLT (hours)** | | | | | **24** | **54** |
| **12.** | **References and Supporting Materials:** | | | | | | |
| **Main Reference(s):** | | | | | | |
| **1.** | **Title:** | | | Innocent Code: A Security Wake-Up Call for Web Programmers 1st Edition | | |
|  | **Author(s):** | | | Sverre H. Huseby | | |
|  | **Pub. Year:** | | | 2004 | | |
| **Additional Reference(s):** | | | | | | |
| **1.** | **Title:** | | | Web Security Testing Cookbook: Systematic Techniques to Find Problems Fast | | |
|  | **Author(s):** | | | Paco Hope, Ben Walther | | |
|  | **Pub. Year:** | | | 2008 | | |
| **2.** | **Title:** | | | The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws | | |
|  | **Author(s):** | | | Dafydd Stuttard, Marcus Pinto | | |
|  | **Pub. Year:** | | | 2011 | | |