Principali informazioni sull'insegnamento	**A.A. 2020-2021**
Titolo insegnamento	Cyber Security
Corso di studio	Informatica e Tecnologie per la Produzione del Software
Crediti formativi	6 (4+2)
Denominazione inglese	Cyber Security
Obbligo di frequenza	No
Lingua di erogazione	Italiano

Docente responsabile	Nome Cognome	Indirizzo Mail
	Paolo Mignone	paolo.mignone@uniba.it

Dettaglio crediti formativi	Ambito disciplinare	SSD	Crediti
Lezioni Frontali	INFORMATICO	ING-INF/05	4
Esercitazioni/Laboratorio	INFORMATICO	ING-INF/05	2

Modalità di erogazione	
Periodo di erogazione	II semestre
Anno di corso	Terzo Anno
Modalità di erogazione	Lezioni frontali e Laboratorio

Organizzazione della didattica	
Ore totali	150
Ore di corso	62 (32 lezioni frontali e 30 esercitazioni/laboratorio)
Ore di studio individuale	88

Calendario	
Inizio attività didattiche	01/03/2021
Fine attività didattiche	04/06/2021

Syllabus	
Prerequisiti	Conoscenze di base di programmazione e linguaggi di programmazione.
Risultati di apprendimento previsti	• *Conoscenza e capacità di comprensione* Lo studente apprenderà a comprendere le criticità di sicurezza in sistemi software, assumendo due prospettive: quella dell'utente malizioso, interessato ad attaccare i sistemi stessi e, specularmente, quella dello sviluppatore consapevole, interessato a mitigare le minacce di sicurezza esistenti nei sistemi. • *Conoscenza e capacità di comprensione applicate* Lo studente apprenderà, con esempi pratici del mondo reale, le varie tecniche di attacco e di difesa adottate in sistemi software. In particolare, saranno

	affrontate tecniche di difesa e attacco basate sulla memoria per sistemi software tradizionali e tecniche per sistemi software web-based. • *Autonomia di giudizio* Lo studente acquisirà una autonomia di giudizio in quanto dovrà essere in grado di decidere quali strategie per prevenire, mitigare e rilevare possibili attacchi, nonché applicare tecniche di revisione del codice per rilevare potenziali vulnerabilità in componenti software. • *Abilità comunicative* Lo studente sarà in grado di descrivere, attraverso gli argomenti trattati nel corso, le scelte intraprese per rafforzare la sicurezza dei sistemi software, durante la fase di sviluppo. Ciò migliorerà le sue capacità di comunicazione nei confronti dei committenti e degli utilizzatori dei sistemi software stessi. • *Capacità di apprendere* I concetti appresi non saranno solamente utili per i casi esposti durante le lezioni frontali, ma risulteranno generali e applicabili anche a contesti differenti (esempio, sistemi software implementati in linguaggi di programmazione diversi). Questo migliorerà la capacità di apprendimento dello studente, che sarà in grado di gestire situazioni analoghe, seppur diverse, rispetto a quelle esposte durante il corso.
Contenuti di insegnamento	**1. Attacchi e Difese alla memoria a basso livello** Buffer overflow, format string attack, attacchi return to libc e ROP (Return-Oriented Programming). Memory-safety enforcement, fat pointers, integrità del flusso di controllo (CFI). **2. Crittografia** Cifrari a blocchi simmetrici e asimmetrici, trapdoor, principio di Kerckhoffs, Feistel ciphers, AES, **3. Sicurezza Web** SQL injection, stored and reflected Cross-site scripting (XSS), Cross-site request forgery (CSRF), Session hijacking e difese basate su validazione dell'input, linee guida per lo sviluppo di web application sicure in Java. **4. Revisione statica e automatica del codice** Analisi statica ed esecuzione simbolica. Presentazione di principi chiave e compromessi esistenti nell'utilizzo di strumenti di analisi statica, quali *taint analysis* e *whitebox fuzz testing*. **5. Controllo degli accessi** Access matrix model, Access Control Lists e Capabilities, UNIX file permissions **6. Penetration testing** Panoramica di obiettivi, tecniche e strumenti fondamentali.

Programma	
Testi di riferimento	Jon Erickson, Hacking, 2nd Edition The Art of Exploitation. 2008, 488 pp., ISBN-13: 978-1-59327-144-2. Gary McGraw, Software Security Library Boxed Set, First Edition, 2006, 1392 pp., ISBN: 978-0321418708 OWASP Foundation, OWASP Testing Guide. 2009 - Version 3.0. F. Long, D. Mohindra, R.C. Seacord, D. F. Sutherland, D. Svoboda. Java Coding Guidelines. Addison-Wesley, 2014. Dispense a cura del docente.
Note ai testi di riferimento	Slide fornite dal docente.
Metodi didattici	Lezioni frontali con il supporto di slide.
Metodi di valutazione	Prova orale + Presentazione di un caso di studio
Criteri di valutazione	L'esame consiste in una prova orale, con presentazione di un caso di studio di una attività di penetration testing di tipo Capture The Flag (CTF) o nella presentazione di una web application sicura. La prova orale mira a verificare la conoscenza dei concetti teorici esposti durante le lezioni frontali. La presentazione del caso di studio è volta a verificare le capacità dello studente nell'identificare le debolezze di un sistema e saper prontamente proporre/applicare le giuste misure di sicurezza.