Principali informazioni sull'insegnamento	
Titolo insegnamento	Progettazione di Sistemi Sicuri
Corso di studio	Sicurezza Informatica (sede di Taranto)
Crediti formativi	6 (4+2P)
Denominazione inglese	Secure Systems Design
Obbligo di frequenza	No
Lingua di erogazione	Italiano

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

Dettaglio credi formativi	Ambito disciplinare	SSD	Crediti
	INFORMATICO	ING-INF/05	6 (4+2P)

Modalità di erogazione	
Periodo di erogazione	I semestre
Anno di corso	Secondo Anno
Modalità di erogazione	Lezioni frontali

Organizzazione della didattica	
Ore totali	150
Ore di corso	32 lezioni frontali
Ore di studio individuale	68 (lezioni frontali) + 50 (progetto)

Calendario	
Inizio attività didattiche	05/10/2020
Fine attività didattiche	13/01/2021

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 coinvolte nella progettazione di 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 durante la fase di sviluppo.

	• *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 adottare in fase di progettazione dei sistemi software. In particolare, sarà in grado di valutare e applicare tecniche di implementazione 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 di basso livello, basati sulla memoria** Stack smashing, attacchi con stringhe di formato, attacchi di accesso alla memoria obsoleti e ROP (Return-Oriented Programming). **2. Difese contro gli attacchi basati sulla memoria** Stack canaries, dati non eseguibili (W+X o DEP), randomizzazione del layout dello spazio degli indirizzi (ASLR), memory-safety enforcement (ad es. SoftBound), integrità del flusso di controllo (CFI). **3. Sicurezza Web** SQL injection, Cross-site scripting (XSS), Cross-site request forgery (CSRF) e Session hijacking e difese basate su validazione dell'input. **4. Modelli di progettazione sicura** Modelli di minaccia e principi di progettazione della sicurezza. Idee di organizzazione. Esempi positivi e negativi di progettazione di sistemi nel mondo reale. **5. 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*. **6. Test di penetrazione** 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.
Note ai testi di riferimento	Slide fornite dal docente.
Metodi didattici	Lezioni frontali con il supporto di slide.
Metodi di valutazione	Prova orale + Progetto
Criteri di valutazione	L'esame consiste in una prova orale, con presentazione di un progetto. La prova orale mira a verificare la conoscenza dei concetti teorici esposti durante le lezioni frontali. Il progetto mira a valutare le capacità di progettazione e messa in pratica delle tecniche di progettazione sicura trattate nel corso.