fişa disciplinei

1. Date despre program

1.1 Instituţia de învăţământ superior	**Universitatea Babeş-Bolyai, Cluj-Napoca**
1.2 Facultatea	**Facultatea de Matematică şi Informatică**
1.3 Departamentul	**Departamentul de Informatică**
1.4 Domeniul de studii	**Informatică**
1.5 Ciclul de studii	**Licenţă**
1.6 Programul de studiu / Calificarea	**Informatică**

2. Date despre disciplină

2.1 Denumirea disciplinei			**Securitate Software**				
2.2 Titularul activităţilor de curs			**Lect. Dr. Radu DRAGOŞ**				
2.3 Titularul activităţilor de seminar			**Drd. Raul TOŞA**				
2.4 Anul de studiu	**3**	2.5 Semestrul	**5**	2.6. Tipul de evaluare	**C**	2.7 Regimul disciplinei	**Opţională**

3. Timpul total estimat (ore pe semestru al activităţilor didactice)

3.1 Număr de ore pe săptămână	**4**	Din care: 3.2 curs	**2**	3.3 seminar / laborator / proiect	**0 + 1 + 1**
3.4 Total ore din planul de învăţământ	**56**	Din care: 3.5 curs	**28**	3.6 seminar / laborator / proiect	**0 + 1 + 1**

Distribuţia fondului de timp:	ore
Studiul după manual, suport de curs, bibliografie şi notiţe	**28**
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate şi pe teren	**28**
Pregătire seminarii/laboratoare, teme, referate, portofolii şi eseuri	**38**
Tutoriat	0
Examinări	0
Alte activităţi:	0

3.7 Total ore studiu individual	**94**
3.8 Total ore pe semestru	**150**
3.9 Numărul de credite	**6**

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	· Arhitectura sistemelor de calcul · Sisteme de operare · Structuri de date şi algoritmi · Baze de date · Programare web
4.2 de competenţe	Programare în C, cunoştinţe de bază ale arhitecturii Intel x86, elemente de bază în programarea web şi SQL.

5. Condiţii (acolo unde este cazul)

5.1 De desfăşurare a cursului	· Sală dotată cu video-proiector
5.2 De desfăşurare a seminarului/laboratorului	· Laborator cu calculatoare cu Visual Studio (C++)

6. Competenţele specifice acumulate

Competențe profesionale	**C1.1** Cunoaşterea noţiunilor, conceptelor şi principiilor teoretice şi practice aferente domeniului general al securităţii informaţiilor. Identificarea şi înţelegerea problemelor de securitate ce pot apărea în contextul unor limbaje specifice. **C1.2** Evaluarea unor proiecte software existente şi identificarea greşelilor de securitate din punctul de vedere al arhitecturii, modului de programare sau procedurilor de testare. Abilitatea de a efectua *code review*, cu scopul de a elimina erori ce afecteaza nivelul de securitate al aplicaţiilor software. **C1.4** Dezvoltarea unor module software sau a unor utilitare care să ajute la asigurarea unui înalt grad de securitate. Propunerea unor scenarii şi modalităţi de testare a unor proiecte existente, cu scopul de a asigura / verifica calitatea lor din punctul de vedere al securităţii. **C2.1** Cunoaşterea principiilor şi noţiunilor de bază necesare proiectării, dezvoltării şi întreţinerii unui cod sigur din punctul de vedere al securităţii (*secure coding*). Cunoaşterea claselor uzuale de software şi unelte de securitate. Cunoaşterea arhitecturilor de sisteme de operare şi platformelor necesare dezvoltării soluţiilor de securitate. **C2.4** Identificarea şi utilizarea unor criterii şi metode adecvate pentru evaluarea, la diferite niveluri de abstractizare, a aplicaţiilor informatice din punctul de vedere al securităţii acestora. Abilitatea de a evalua şi documenta, din perspectiva securităţii, calitatea aplicaţiilor informatice. **C6.1** Înţelegerea arhitecturilor sistemelor de calcul şi a modelelor de comunicare a acestora prin reţele. Abilitatea de a proiecta canale de comunicare solide din punctul de vedere al securităţii informaţiilor transmise.
Competențe transversale	**CT1**. Aplicarea şi promovarea deprinderilor de proiectare şi codare defensivă, sigură si eficientă. Îmbunătăţirea abilităţilor profesionale prin valorificarea noii perspective dobândite, cea a securităţii sistemelor software.

7. Obiectivele disciplinei (reieşind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	Capacitatea evaluării caracteristicilor de securitate ale unei aplicaţii software la nivelul codului sursă. Dobândirea deprinderilor fundamentale minimale de scriere a unui cod sursă fără vulnerabilităţi.
7.2 Obiectivele specifice	1. Cunoaşterea mecanismelor de bază ce definesc securitatea sistemului şi a mediului software în care se execută o aplicaţie (i.e. modelul de securitate), cum ar fi: permisiunile de acces, politicile de securitate, interacţiunea cu mediul exterior etc. 2. Cunoaşterea principalelor tipuri de vulnerabilităţi software, precum: utilizarea datelor utilizator nevalidate corespunzător, interacţiunea necontrolată directă sau indirectă cu mediul exterior aplicaţiei etc. 3. Deprinderea unor tehnici eficiente de studiere şi evaluare a unui cod sursă din perspectiva securităţii şi capacitatea de a identifica posibile vulnerabilităţi. 4. Capacitatea de a evalua implicaţiile unei vulnerabilităţi descoperite. 5. Cunoaşterea tehnicilor şi a bibliotecilor de funcţii utile în scrierea unui cod sursă fără vulnerabilităţi şi capacitatea de a le utiliza în situaţii reale.

8. Conținuturi

8.1 Curs		Metode de predare	Observaţii
1	Concepte şi aspecte de bază referitoare la vulnerabilităţile software şi la metodele şi uneltele de dezvoltare a unui software fără vulnerabilităţi şi de evaluare a unui software din perspectiva posibilelor vulnerabilităţi	Expunere la tablă, prezen tare cu video proiectorul, discuţii, probleme scurte	
2	Vulnerabilităţi de corupere a memoriei (*buffer/integer overflow etc.*)		
3	Vulnerabilităţi specifice limbajului C: limite aritmetice (de reprezentare), conversii de tip, pointeri etc.		
4	Vulnerabilităţi în componentele structurale ale unei aplicaţii software (*Program building blocks*)		
5	Vulnerabilităţi în utilizarea şi manipularea şirurilor de caractere şi meta-caractere		
6	Vulnerabilităţi specifice sistemelor de operare UNIX		
7	Vulnerabilităţi specifice sistemelor de operare Windows		
8	Vulnerabilităţi de sincronizare (în situaţii de concurenţă)		
9	Vulnerabilităţi Web: injectare cod SQL, XSS, XSRF etc.		
10	Vulnerabilităţi de criptografie: parole vulnerabile, numere aleatoare previzibile etc.		
11	Vulnerabilităţi specifice codului aplicaţiilor ce folosesc comunicarea în reţelele de calculatoare		
12	Metode de proiectare corectă a aplicaţiilor din perspectiva securităţii: principii de proiectare, definirea modelului de riscuri (threat modeling), evaluare design etc.		
13	Metode de implementare corectă a unei aplicaţii software din perspectiva securităţii: metode şi modele de dezvoltare a aplicaţiilor (Waterfall, agile), cele mai frecvente şi mai periculoase riscuri şi vulnerabilităţi, tehnici de defensive de scriere a codului (*defensive coding techniques*)		
14	Metode de evaluare a (codului) unei aplicaţii din perspectiva securităţii: asigurarea calităţii, testare, gestiunea vulnerabilităţilor identificate		
8.2 Seminar / laborator / proiect		Metode de predare	Observaţii
1	Unelte utile în identificarea şi evaluarea vulnerabilităţilor unui cod sursă: navigatoare prin codul sursă, depanatoare, navigatoare prin codul executabil (binar), testare *fuzzy*	Scurte expuneri la tablă, tutoriale, ghiduri de lucru, demonstraţii *live*, explicaţii suplimentare, discuţii, propunerea spre	

		rezolvare a unor probleme de diferite tipuri şi grade de complexitate.	
2	Tehnici de evitare, detecţie şi evaluare a vulnerabilităţilor de corupere a memoriei si specifice limbajului C		
3	Tehnici de evitare, detecţie şi evaluare a vulnerabilităţilor de utilizare şi gestionare a şirurilor de caractere şi meta-caractere		
4	Tehnici de evitare, detecţie şi evaluare a vulnerabilităţilor specifice sistemului de operare Linux		
5	Tehnici de evitare, detecţie şi evaluare a vulnerabilităţilor sistemelor de operare Windows		
6	Tehnici de evitare, detecţie şi evaluare a vulnerabilităţilor de sincronizare		
7	Tehnici de evitare, detecţie şi evaluare a vulnerabilităţilor aplicaţiilor Web si aplicaţiilor de reţea		

Bibliografie

1. M. Down, J. McDonald, J. Schuh, „*The Art of Software Security Assessment. Identifying and Preventing Software Vulnerabilities*”, Addison-Wesley, 2007
2. M. Howard, D. LeBlanc, J. Viega, „*24 Deadly Sins of Software Security. Programming* Flows *and How to Fix Them*”, McGraw Hill, 2010
3. M. Howard, D. LeBlanc, „*Writing Secure Code for Windows Vista*”, Microsoft Press, 2007
4. G. McGraw, „*Software Security:Building Security In*”, Addison-Wesley, 2006
5. R. Seacord, „CERT C Coding Standard: 98 Rules for Developing Safe, Reliable, and Secure Systems”, Addison-Wesley, 2nd edition, 2014
6. -, „*Common Weaknesses Enumeration* (WCE)”, on-line: http://cwe.mitre.org/data/index.html

9. Coroborarea conţinuturilor disciplinei cu aşteptările reprezentanţilor comunităţii epistemice, asociaţiilor profesionale şi angajatori reprezentativi din domeniul aferent programului

Se realizează prin discuţii periodice cu reprezentanţi ai angajatorilor semnificativi din domeniul securităţii informaţiei.

Cursuri referitoare la aspecte de securitate în dezvoltarea aplicaţiilor şi domenii adiacente (de exemplu teste de penetrare) sunt prezente în cadrul multor alte masterate din domeniul securităţii calculatoarelor şi a informaţiilor, la universităţi din ţară şi străinătate, cum ar fi:

- *Securitatea sistemelor software*, Master de Securitatea informaţiei, Universitatea Al. I. Cuza, Iaşi, Facultatea de calculatoare, http://profs.info.uaic.ro/~webdata/planuri/master/MISS1FS03.pdf
- *Securitatea sistemelor şi aplicaţiilor*, Master de Securitatea tehnologiei informaţiei, Academia Tehnică Militară, Bucureşti, http://mta.ro/masterat/masterinfosec/curricula2013.html
- *Secure Software Systems*, Master of Science in Information Security, Carnegie Mellon University, SUA, http://www.ini.cmu.edu/degrees/msis/courses.html
- *Software Security*, Master in Information Security, Royal Holloway University of London, Information Security Group, https://www.royalholloway.ac.uk/isg/documents/pdf/coursespecs(msc)/modules201314/iy5607softwaresecurityspec1314.pdf

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	· Abilitatea de definire a conceptelor specifice problemelor de securitate la nivel de cod sursă şi de expunere a metodelor de evaluare şi dezvoltare corectă a unui cod sursă din perspectiva securităţii. · Abilitatea de rezolvare a unor probleme specifice domeniului. · Prezenţă, (inter)activitate în timpul orelor de curs.	Test online (grilă) sau lucrare scrisă (grilă), şi/sau prezentarea unei teme de cercetare din domeniul cursului.	50%
10.5 Seminar/laborator	· Abilitatea de rezolvare a unor probleme specifice domeniului · Prezenţă, (inter)activitate în timpul orelor de laborator/proiect.	Realizarea activităţilor de laborator şi/sau rezolvarea temelor de casă şi/sau a unor probleme în cadrul unui test practic	50%
10.6 Standard minim de performanță			
· Capacitatea de a defini vulnerabilităţile software fundamentale, precum: buffer-overflow, injectare code SQL, XSS etc. · Capacitatea de a identifica vulnerabilităţile software fundamentale şi de a corecta codul (demonstrate în cadrul exerciţiilor de laborator şi a evaluării finale).			

Data completării	Titular de curs	Titular de seminar / laborator / proiect
3 Mai 2016	*Lect. Dr. Radu DRAGOŞ*	*Drd. Raul TOŞA*

Data avizării în departament

.....

Director de departament

….....