UNIVERSITY OF THE FRASER VALLEY

ORIGINAL COURSE IMPLEMENTATION DATE: January 2005
REVISED COURSE IMPLEMENTATION DATE: September 2018
COURSE TO BE REVIEWED: (six years after UEC approval) January 2024
Course outline form version: 09/15/14

OFFICIAL UNDERGRADUATE COURSE OUTLINE FORM

Note: The University reserves the right to amend course outlines as needed without notice.

Course Code and Number: COMP 325	**Number of Credits:** 3 Course credit policy (105)
Course Full Title: Malicious Software and Attack Prevention **Course Short Title (if title exceeds 30 characters):** Malicious Soft. & Attack Prev.	
Faculty: Faculty of Professional Studies	**Department (or program if no department):** Computer Information Systems

Calendar Description:

Students will learn about the vulnerabilities inherent in computer programs. Topics studied will include stack and buffer overflows, race conditions, file operations, string handling, interprocess communication, injection attacks. C and assembly language examples will be used.

Note: Students with credit for CIS 325 cannot take this course for further credit.

Prerequisites (or NONE):	COMP 155 or CIS 221. Note: COMP 256 is recommended.
Corequisites (if applicable, or NONE):	
Pre/corequisites (if applicable, or NONE):	

Equivalent Courses (cannot be taken for additional credit)

Former course code/number: **CIS 325**

Cross-listed with:

Equivalent course(s):

Note: Equivalent course(s) should be included in the calendar description by way of a note that students with credit for the equivalent course(s) cannot take this course for further credit.

Transfer Credit

Transfer credit already exists: ☐ Yes ☒ No

Transfer credit requested (OReg to submit to BCCAT):
☐ Yes ☒ No (if yes, fill in transfer credit form)

Resubmit revised outline for articulation: ☐ Yes ☐ No

To find out how this course transfers, see **bctransferguide.ca**.

Total Hours: 45

Typical structure of instructional hours:

Lecture hours	45
Seminars/tutorials/workshops	
Laboratory hours	
Field experience hours	
Experiential (practicum, internship, etc.)	
Online learning activities	
Other contact hours:	
Total	**45**

Special Topics

Will the course be offered with different topics?
☐ Yes ☒ No

If yes, different lettered courses may be taken for credit:
☐ No ☐ Yes, repeat(s) ☐ Yes, no limit

Note: The specific topic will be recorded when offered.

Expected frequency of course offerings (every semester, annually, every other year, etc.): Once per year

Department / Program Head or Director: Daniel Harris	**Date approved:**	September 2017
Faculty Council approval	**Date approved:**	October 13, 2017
Campus-Wide Consultation (CWC)	**Date of posting:**	November 17, 2017
Dean/Associate VP: Tracy Ryder Glass	**Date approved:**	October 13, 2017
Undergraduate Education Committee (UEC) approval	**Date of meeting:**	January 26, 2018

Learning Outcomes

Upon successful completion of this course, students will be able to:

- Explain inherent vulnerabilities in modern software architecture
- Interpret issues resulting in stack and buffer overflows
- Explain race conditions
- Describe file operation vulnerabilities
- Identify how string format and string handling vulnerabilities can be attacked
- Describe reasons for validating interprocess communication
- Describe injection attacks

Prior Learning Assessment and Recognition (PLAR)

☒ Yes ☐ No, PLAR cannot be awarded for this course because

Typical Instructional Methods (guest lecturers, presentations, online instruction, field trips, etc.; may vary at department's discretion)

Lectures and hands-on laboratory.

Grading system: Letter Grades: ☒ Credit/No Credit: ☐ Labs to be scheduled independent of lecture hours: Yes ☐ No ☒

NOTE: The following sections may vary by instructor. Please see course syllabus available from the instructor.

Typical Text(s) and Resource Materials (if more space is required, download Supplemental Texts and Resource Materials form)

	Author (surname, initials)	Title (article, book, journal, etc.)	Current ed.	Publisher	Year
1.	Erickson, Jon	Hacking, The Art of Exploitation	☒	No Starch Press Inc.	
2.	Kaziol, Litchfield, Aitel, Anley, Eren, Mehta, Hassell	The Shellcoder's Handbook, Discovering and Exploiting Security Holes	☒	Wiley Publishing	
3.	Pfleeger and Pfleeger	Security in Computing	☒	Prentice Hall	
4.	Holland and McGraw	Exploiting Software, How to Break Code	☒	Addison-Wesley	
5.	Vienna and McGraw	Building Secure Software, How to Avoid Security Problems the Right Way	☒	Addison-Wesley	

Required Additional Supplies and Materials (software, hardware, tools, specialized clothing, etc.)

None

Typical Evaluation Methods and Weighting

Final exam:	35 %	Assignments:	30 %	Midterm exam:	35 %	Practicum:	%
Quizzes/tests:	%	Lab work:	%	Field experience:	%	Shop work:	%
Other:	%	Other:	%	Other:	%	Total:	100%

Details (if necessary):

Typical Course Content and Topics

- C and assembler review
- Inter-process communications
- Input validation
- Buffer overflow
- Format string attacks
- File handling
- Injection attacks
- Stack overflows
- Heap overflows
- String handling
- Integer overflows and underflows
- Secure storage and encryption
- Access control problems
- Race conditions
- Forensics
- Social Engineering
- Privacy