Université du Québec en Outaouais **Département d'informatique et d'ingénierie**

Sigle : CYB6073 Gr. 01
Titre : Analyse statique du logiciel pour la cybersécurité
Session : Automne 2026 **Horaire et local**
Professeur.e : Nguena Timo, Omer

1. Description du cours paraissant à l'annuaire :

Objectifs

Au terme de ce cours, l'étudiant.e sera en mesure d'appliquer des techniques d'analyse statique de code et d'utiliser des outils qui implémentent ces techniques à diverses fins.

Contenu

Rappel du langage C et mesure de complexité d'un programme C. Objectifs d'analyse du code : exploration, compréhension, détection de défauts, etc. Défauts logiciels et conséquences sur le fonctionnement : division par zéro, décisions erronées, dépassement de la mémoire, etc. Analyse des domaines des variables, analyse du flot d'exécution. Technique et outils d'exécution symbolique de code C et application au test, à la vérification et à la cybersécurité. Réduction de la complexité du code. Introduction à l'interprétation abstraite.

Descriptif – Annuaire

2. Objectifs spécifiques du cours :

1. Analyser la structure et le comportement potentiel d'un programme C en examinant sa syntaxe, ses structures de contrôle, ses fonctions, ses variables et ses opérations sur la mémoire, ainsi qu'en estimant sa complexité.

2. Construire et exploiter des représentations intermédiaires d'un programme, notamment l'arbre syntaxique abstrait (AST), le graphe de flot de contrôle (CFG) et les représentations du flot de données, afin de faciliter son analyse automatisée.

3. Identifier et caractériser des défauts logiciels et des vulnérabilités liés notamment aux erreurs de calcul, aux décisions erronées, aux dépassements de mémoire, aux accès hors limites, aux pointeurs invalides et aux erreurs de gestion de la mémoire.

4. Appliquer des techniques d'analyse du flot de contrôle et du flot de données pour étudier les chemins d'exécution possibles, les définitions et utilisations des variables ainsi que les conditions susceptibles de conduire à un comportement incorrect ou vulnérable.

5. Analyser les domaines de valeurs des variables en déterminant les contraintes qui s'appliquent à leurs valeurs et en utilisant des techniques telles que l'analyse de constantes, les intervalles et l'analyse des pointeurs et des alias.

6. Appliquer l'exécution symbolique et la résolution de contraintes pour explorer les chemins d'exécution d'un programme C, générer des cas de test et rechercher des conditions susceptibles de provoquer des défauts ou des violations de propriétés de sécurité.

7. Expliquer et mettre en œuvre les principes fondamentaux de l'interprétation abstraite pour raisonner sur les comportements possibles d'un programme sans exécuter toutes ses entrées ni tous ses chemins d'exécution.

8. Utiliser et comparer des outils d'analyse statique pour détecter, caractériser et diagnostiquer des défauts et des vulnérabilités dans des programmes C, en interprétant leurs résultats et en tenant compte de leurs limites.

9. Réduire la complexité de l'analyse d'un programme en identifiant les chemins, les instructions ou les propriétés pertinents pour l'objectif d'analyse, tout en préservant la validité des conclusions recherchées.

10. Corriger et réévaluer un programme vulnérable en s'appuyant sur les résultats des analyses, afin de vérifier que les défauts ciblés ont été corrigés et que les modifications n'introduisent pas de nouveaux problèmes détectables.

11. Consulter et exploiter des bases de données de faiblesses et de vulnérabilités logicielles, notamment CWE (Common Weakness Enumeration) et CVE (Common Vulnerabilities and Exposures), afin de classifier les faiblesses détectées dans le code, de comprendre leurs conséquences potentielles sur la sécurité et de rechercher les vulnérabilités connues associées.

3. Stratégies pédagogiques :

Enseignement en non présentiel

Évaluation en présentiel

L'apprentissage s'articulera autour de trois approches complémentaires :

- Approche conceptuelle : exposés magistraux illustrés, permettant d'acquérir les fondements de l'analyse statique, les représentations des programmes, l'analyse des flots de contrôle et de données, l'exécution symbolique et l'interprétation abstraite.
- Approche pratique : exercices dirigés et laboratoires utilisant des programmes C, des outils d'analyse statique et des solveurs de contraintes pour mettre en œuvre les techniques étudiées et interpréter les résultats obtenus.
- Approche par problèmes et études de cas : analyse de programmes comportant des défauts et des vulnérabilités réalistes, classification des faiblesses à l'aide de référentiels tels que CWE et CVE, puis proposition et vérification de correctifs.

l'évaluation des apprentissages combinera des exercices théoriques, des travaux pratiques et une étude de cas intégratrice.

4. Heures de disponibilité ou modalités pour rendez-vous :

Mercredi 11h30-12h30 , par teams

5. Plan détaillé du cours sur 15 semaines :

Semaine	Thèmes	Dates
1	**Présentation du plan du cours** **Motivation à l'analyse statique du logiciel pour la cybersécurité**	11 sept. 2026
2	**Bases de données de vulnérabilités**	18 sept. 2026

	Vulnérabilités dans les logiciels C	
3	Jour férié à l'UQO	25 sept. 2026
4	**Outils de détection de vulnérabilité à base de règles lexicales** **Technique d'analyse lexicale et théorie des automates finis** **Introduction à ANTLR pour l'analyse lexicale**	02 oct. 2026
5	**Limites de l'analyse lexicale pour la détection de vulnérabilité** **Outils de détection de vulnérabilité à base de règles syntaxiques** **Technique d'analyse syntaxique et représentation de programme par des arbres syntaxiques** **Remise des descriptions de projet de programmation - partie 1** **Remise des descriptions de projet de détection de vulnérabilité - partie 1**	09 oct. 2026
6	**Semaine des études**	16 oct. 2026
7	**Technique d'analyse du flot de données et de contrôle** **Outils et application à la détection de vulnérabilité** **Conception d'un outil d'analyse du flot de données et de contrôle**	23 oct. 2026
8	**En PRESENTIEL** **Évaluation 1** **Analyse par contraintes et modèles** • **Logique des propositions et SAT** • **Encodage binaire** • **Transformation d'un programme en formule booléenne** • **Vérification bornée de code C** **Outils et application à la détection de vulnérabilité**	30 oct. 2026
9	**Technique d'analyse symbolique** • **Contrainte et solveur SMT**	06 nov. 2026

	• **Transformation d'un problème de détection de vulnérabilité en contrainte SMT** **Outils et application à la détection des vulnérabilités**	
10	**En Présentiel** **Évaluation 2** **Technique d'analyse de pointeurs et mémoire (1)** **Outils et application à la détection de vulnérabilités** **Remise des descriptions de projet de programmation - partie 2** **Remise des descriptions de projet de détection de vulnérabilité - partie 2**	13 nov. 2026
11	**Interprétation abstraite (1)** **Outils et application à la détection de vulnérabilité**	20 nov. 2026
12	**Interprétation abstraite (2)** **Outils et application à la détection de vulnérabilité**	27 nov 2026
13	**Évaluation 3** **Étude de la complexité des techniques d'analyse statique** **Estimation de la complexité du code** **Reduction de la complexité du code et efficacité de l'analyse statique**	4 déc. 2026
14	**Cours en Présentiel:** **Examen de fin de session**	11 déc. 2026
15	**Partie 1 :** **- Présentation du projet de programmation** **Partie 2 :** **- Présentation du projet de détection de vulnérabilité, des outils et méthodes utilisables pour la détection**	18 déc. 2026
6. Évaluation du cours :		

- 3 Évaluations: 30%
- projets de détection et de corrections de vulnérabilité: 30%
- 1 projet de programmation d'une méthode d'analyse statique : 20%
- 1 Examen final : 20 %

7. Politiques départementales et institutionnelles :

- Politique du département d'informatique et d'ingénierie relative à la tenue des examens
- Note sur le plagiat et sur la fraude
- Politique relative à la qualité de l'expression française écrite chez les étudiants et les étudiantes de premier cycle à l'UQO
- Absence aux examens : cadre de gestion, demande de reprise d'examen (formulaire)

Tolérance ZÉRO en matière de violence à caractère sexuel.

Le Bureau d'intervention et de prévention en matière de harcèlement (BIPH) a pour mission d'accueillir, soutenir et guider toute personne vivant une situation de harcèlement, de discrimination ou de violence à caractère sexuel. Le BIPH oriente ses actions afin de prévenir les violences à caractère sexuel pour que nous puissions étudier, travailler et s'épanouir dans un milieu sain et sécuritaire.

Vous vivez ou êtes une personne témoin d'une situation de violence à caractère sexuel ? Vous êtes une personne membre de la communauté étudiante ou une personne membre du personnel, autant à Gatineau qu'à Ripon et St-Jérôme, l'équipe du BIPH est là pour vous, sans jugement et en toute confidentialité.

Ensemble, participons à une culture de respect.

Pour de plus amples renseignements consultez UQO.ca/biph ou écrivez-nous au Biph@uqo.ca

8. Principales références :

1. https://dwheeler.com/flawfinder/
2. Bennett, Gareth, et al. "Semgrep*: Improving the limited performance of static application security testing (sast) tools." *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*. 2024.
3. Kroening, Daniel, and Michael Tautschnig. "CBMC–C Bounded Model Checker: (Competition Contribution)." *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.
4. Baldoni, Roberto, et al. "A survey of symbolic execution techniques." *ACM Computing Surveys (CSUR)* 51.3 (2018): 1-39.
5. Xu, Baowen, et al. "A brief survey of program slicing." *ACM SIGSOFT Software Engineering Notes* 30.2 (2005): 1-36.
6. Lipp, Stephan, Sebastian Banescu, and Alexander Pretschner. "An empirical study on the effectiveness of static C code analyzers for vulnerability detection." *Proceedings of the 31st ACM SIGSOFT international symposium on software testing and analysis*. 2022.
7. Linz, Peter, and Susan H. Rodger. *An introduction to formal languages and automata*. Jones & Bartlett Learning, 2022.

9. Page Web du cours :

https://moodle.uqo.ca