МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Волинський національний університет імені Лесі Українки
Факультет інформаційних технологій і математики
Кафедра комп'ютерних наук та кібербезпеки

СИЛАБУС

вибіркового освітнього компонента

БЕЗПЕЧНЕ ПРОГРАМУВАННЯ

підготовки бакалавра

Луцьк – 2026

Силабус освітнього компонента «Безпечне програмування**»** підготовки бакалавра, всіх галузей знань, всіх спеціальностей, за всіма освітніми програмами.

Розробник: Гаращенко В.В. старший викладач кафедри комп'ютерних наук та кібербезпеки.

Погоджено
Гарант освітньо-професійної програми:  Гришанович Т. О.

Силабус освітнього компонента затверджено та погоджено на засіданні кафедри комп'ютерних наук та кібербезпеки

протокол № 6 від 15.01.2026 р.

Завідувач кафедри:  Гришанович Т. О.

І. Опис освітнього компонента

Найменування показників	Характеристика освітнього компонента
	Вибірковий
Денна форма навчання	Рік підготовки 4
150/5 кредитів	Семестр 8
	Лекції 10 год.
	Лабораторні 20 год.
	Самостійна робота 110 год.
ІНДЗ: немає	Консультації 10 год.
	Форма контролю: залік

ІІ. Інформація про викладача

ПІП Гаращенко Володимир Вікторович
Науковий ступінь -
Вчене звання -
Посада ст. викладач кафедри комп'ютерних наук та кібербезпеки
Контактна інформація ел. скринька: Harashchenko.Volodymyr@vnu.edu.ua
Дні занять https://ps.vnu.edu.ua/cgi-bin/timetable.cgi?n=700

ІІІ. Опис освітнього компонента

1. Анотація освітнього компонента

Освітній компонент «Безпечне програмування» складено з урахуванням можливості формування індивідуальної освітньої траєкторії здобувачів освіти підготовки бакалавра та орієнтований на формування професійних компетентностей у сфері розробки програмного забезпечення з урахуванням вимог кібербезпеки.

Дисципліна поглиблює знання студентів щодо принципів Secure SDLC (Secure Software Development Life Cycle), моделей загроз, безпечної архітектури програмних систем, а також методів запобігання, виявлення та усунення вразливостей на різних етапах життєвого циклу програмного продукту.

Особлива увага приділяється практичним аспектам аналізу коду, роботі з інструментами автоматизованого тестування безпеки, впровадженню механізмів захисту даних, управлінню доступом та протидії типовим кібератакам.

2. Пререквізити та постреквізити

Пререквізити

Для успішного засвоєння дисципліни здобувач повинен володіти:

- знаннями з об'єктно-орієнтованого програмування (C/C++, Java, Python, C#);
- розумінням алгоритмів та структур даних;
- базовими знаннями з операційних систем;
- знаннями принципів побудови комп'ютерних мереж;

- основами баз даних та SQL;
- навичками розробки веб- або десктоп-застосунків.

Постреквізити

Результати навчання з дисципліни можуть бути використані під час:

- виконання кваліфікаційної роботи бакалавра;
- проєктування архітектури програмних систем;
- проведення аудиту безпеки програмного коду;
- роботи на посадах Software Engineer, Security Engineer, Application Security Analyst, DevSecOps Engineer.

3. Мета і завдання освітнього компонента

Мета дисципліни

Формування у здобувачів системного бачення безпеки програмного забезпечення та практичних навичок розробки стійких до атак програмних систем відповідно до сучасних міжнародних стандартів кібербезпеки.

Завдання дисципліни

- сформувати розуміння моделей загроз та принципів threat modeling;
- навчити ідентифікувати типові вразливості (Buffer Overflow, SQL Injection, XSS, CSRF, Insecure Deserialization, Race Conditions тощо);
- опанувати методи статичного (SAST), динамічного (DAST) та інтерактивного (IAST) аналізу безпеки;
- вивчити принципи безпечної архітектури (principle of least privilege, defense in depth, secure by design);
- навчитися реалізовувати механізми автентифікації, авторизації, шифрування та захисту даних;
- опанувати принципи безпечної роботи з пам’яттю;
- навчитися інтегрувати перевірки безпеки в CI/CD процеси;
- сформувати навички аналізу реальних кейсів експлуатації вразливостей;
- розвинути вміння документувати ризики та формувати рекомендації щодо їх усунення.

4. Soft Skills

Освітній компонент сприяє розвитку таких загальних компетентностей:

- **Критичне та аналітичне мислення** – здатність аналізувати архітектуру системи з точки зору потенційного зловмисника та виявляти слабкі місця ще на етапі проєктування.
- **Системне мислення** – розуміння взаємозв’язків між компонентами програмної системи та впливу змін на рівень безпеки.
- **Увага до деталей** – здатність виявляти помилки валідації даних, некоректні обробники винятків, неправильну конфігурацію доступу.
- **Етична відповідальність та правова грамотність** – усвідомлення правових аспектів роботи з персональними даними та відповідальності за розголошення вразливостей.
- **Problem Solving** – здатність знаходити оптимальні технічні рішення для усунення ризиків без значного впливу на продуктивність системи.

- **Командна взаємодія** – ефективна комунікація з розробниками, тестувальниками та DevOps-інженерами під час впровадження політик безпеки.
- **Навички технічної аргументації** – уміння обґрунтувати необхідність рефакторингу або зміни архітектури з позиції безпеки.
- **Стресостійкість** – здатність приймати рішення в умовах виявлення критичних уразливостей або під час інцидентів безпеки.
- **Безперервне навчання (Lifelong Learning)** – готовність постійно оновлювати знання відповідно до розвитку нових типів атак і технологій захисту.

5. Структура освітнього компонента.

Назви змістових модулів і тем	Усього	Лек.	Лабор.	Сам. роб.	Конс.	Форма контролю / Бали
Змістовий модуль 1. Практичні технології безпечного програмування та захищеної розробки програмних систем						
Тема 1. Принципи безпечного програмування та реалізація базових механізмів захисту в коді.	23	2	4	15	2	Звіт/6/6
Тема 2. Реалізація захисту від ін'єкційних та вебвразливостей у прикладних системах.	23	2	4	15	2	Звіт /6/6
Тема 3. Програмна реалізація криптографічного захисту та механізмів автентифікації.	23	2	4	15	2	Звіт /6/6
Тема 4. Автоматизований аналіз безпеки коду та інтеграція перевірок у процес розробки.	23	2	4	15	2	Звіт /6/6
Тема 5. Побудова безпечної архітектури застосунку та реалізація практик DevSecOps.	23	2	4	15	2	Звіт/6/6
Разом за модулем 1	115	10	20	75	10	60
Контрольна робота				20		20
Тестування				15		20
Всього годин/Балів	150	10	20	110	10	100

6. Завдання для самостійного опрацювання.

№ з/п	Тема (опрацювати)	Кількість год.
1	Опрацювання та аналіз лекційного матеріалу	20
2	Опрацювання додаткових джерел та відео-роликів мережі Інтернет	15
3	Підготовка до лабораторних робіт	20
4	Підготовка до контрольної роботи	20
5	Підготовка до тестування	15

6	Підготовка відповідей на питання до заліку	20

IV. Політика оцінювання

Політика викладача щодо здобувача освіти

Усі учасники освітнього процесу повинні дотримуватись вимог чинного законодавства України, Статуту і Правил внутрішнього розпорядку ВНУ імені Лесі Українки, загально-прийнятих моральних принципів, правил поведінки та корпоративної культури; підтримувати атмосферу доброзичливості, відповідальності, порядності й толерантності. Атмосфера на заняттях повинна бути творчою, відкритою до конструктивної критики. Недопустимі запізнення на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття; списування. Очікується, що всі студенти відвідають усі лекції і практичні заняття курсу.

Політика щодо академічної доброчесності

Дотримання академічної доброчесності здобувачами передбачає: самостійне виконання завдань поточного та підсумкового контролю (для осіб з особливим освітніми потребами ця вимога застосовується з урахуванням їх індивідуальних потреб і можливостей); посилання на джерела інформації у разі використання ідей, тверджень, відомостей; дотримання норм законодавства про авторське право; надання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності.

Порушенням академічної доброчесності вважається: академічний плагіат, самоплагіат, фабрикація, фальсифікація, списування. За порушення академічної доброчесності здобувачі освіти можуть бути притягнені до такої академічної відповідальності: повторне проходження оцінювання; повторне проходження відповідного освітнього компонента освітньої програми.

Під час модульного та підсумкового контролю (заліку) студентам заборонено користуються такими засобами як мобільний телефон, планшет, конспект, навчальна література, інші джерела інформації, в тому числі Інтернет-ресурси.

Політика щодо дедлайнів та перескладання

Усі передбачені завдання мають бути виконані у встановлений термін. Несвоєчасно виконані завдання оцінюються на нижчу оцінку. Виключенням можуть бути завдання, які не вдалося зробити з поважних причин, в такому випадку студент може доробити вказані завдання у вказаний термін.

Якщо здобувач вищої освіти був відсутній на заняттях з будь-якої причини, то він (вона) вивчає матеріал самостійно, використовуючи навчальні посібники, конспекти лекцій, матеріали дистанційного курсу, у випадку розміщення його на платформі дистанційного навчання Moodle, виконує всі домашні завдання (завдання подані на самостійну роботу). Прозвітуватися про виконання завдань можна, використовуючи дистанційний курс, прикріпивши виконанні завдання у відповідні комірки та попередити викладача про здане завдання, або під час консультацій або надіслати виконане завдання на корпоративну пошту викладача. Зворотній зв'язок з викладачем для з’ясування всіх питань: використання форуму, чату дистанційного курсу, корпоративної пошти університету або відповідної бесіди у певному месенджері.

Можливість визнання результатів навчання, отриманих у формальній, неформальній та інформальній освіті

Під час вивчення освітнього компонента можливе визнання інших результатів навчання, отриманих у формальній, неформальній та/або інформальній освіті. Порядок визнання результатів навчання для здобувачів вищої освіти, набутих у: формальній освіті (академічна мобільність студентів на території України чи поза її межами, для студентів, які переводяться, поновлюються з інших ЗВО (вітчизняних чи іноземних); неформальній та/або інформальній освіті здійснюється згідно «ПОЛОЖЕННЯ про визнання результатів навчання, отриманих у формальній, неформальній та/або інформальній освіті у Волинському національному університеті імені Лесі Українки».

Можливість отримати додаткові (бонусні) бали

Відповідно до пункту 4.5 Положення про поточне та підсумкове оцінювання знань здобувачів вищої освіти Волинського національного університету імені Лесі Українки здобувачам освіти, які брали участь у роботі конференцій, підготовці наукових публікацій, в олімпіадах, конкурсах студентських наукових робіт, спортивних змаганнях, мистецьких конкурсах тощо й досягли значних результатів, може бути присуджено додаткові (бонусні) бали, які зараховуються як результати поточного контролю з відповідного ОК. Систему бонусних балів погоджує науково-методична комісія факультету (інституту).

І так, здобувачі освіти мають можливість отримати додаткові бали за вказаний вид робіт з ОК «Програмування» відповідно до таблиці витягу з протоколу № 1 засідання НМТ ФІТІМ ВНУ ім. Лесі Українки від 3.09.2025 р.

Системи бонусних балів для здобувачів освіти

Вид діяльності	Рівень / результат	Кількість бонусних балів
Студентські олімпіади	І місце	7
	ІІ місце	5
	ІІІ місце	3
	Участь в олімпіаді	2
Конкурси студентських наукових робіт	Диплом І ступеня	7
	Диплом ІІ ступеня	5
	Диплом ІІІ ступеня	3
Підготовка наукових публікацій	Публікація в WoS / Scopus	10
	Фахова стаття	7
	Нефахова стаття	5
	Публікація тез	2
Участь у конференціях	Виступ на конференції	2
Першість України з командного програмування	І місце	10
	ІІ місце	8
	ІІІ місце	6
	Участь	4

V. Підсумковий контроль

Підсумковий контроль з даної дисципліни передбачено у вигляді заліку.

Оцінювання здійснюється за 100-бальною шкалою. Оцінка включає в себе оцінювання всіх видів запланованої навчальної роботи протягом семестру: нараховується за якісне виконання лабораторних, індивідуальних робіт та виконання самостійної роботи. Максимальна кількість балів, яку може отримати студент під час поточного оцінювання, у випадку заліку, за семестр – 100 балів.

Залік викладач виставляє за результатами поточної роботи за умови, що здобувач освіти виконав ті види навчальної роботи, які визначено силабусом ОК. У випадку, якщо

здобувач освіти не відвідував окремі аудиторні заняття (з поважних причин), на консультаціях він має право відпрацювати пропущені заняття та добрати ту кількість балів, яку було визначено на пропущені теми. У дату складання заліку викладач записує у відомість суму поточних балів, які здобувач освіти набрав під час поточної роботи (шкала від 0 до 100 балів).

У випадку, якщо здобувач освіти протягом поточної роботи набрав менше як 60 балів, він складає залік під час ліквідації академічної заборгованості. У цьому випадку бали, набрані під час поточного оцінювання анульовуються. Максимальна кількість балів на залік під час ліквідації академічної заборгованості, як правило, 100. У день складання заліку за основною сесією заборонено проводити додаткові опитування здобувача освіти, а також здобувач освіти не має права доздавати будь-який вид робіт, передбачений силабусом освітнього компоненту. На заліку, під час ліквідації академічної заборгованості, здобувач отримує комплексне завдання, яке охоплює всі теми і всі форми контролю, які пропонувалися при вивченні освітнього компонента. Порядок проведення заліку-ліквідації – залік відбувається у вигляді виконання комплексного завдання.

Питання до заліку-ліквідації та приклади практичних завдань

1. Поясніть принципи Secure by Design та Security by Default.
2. У чому полягає принцип мінімальних привілеїв і як його реалізувати програмно?
3. Що таке валідація та санітизація вхідних даних? Наведіть приклади реалізації.
4. Які помилки обробки винятків можуть призводити до витоку інформації?
5. Поясніть механізм виникнення переповнення буфера та способи його запобігання.
6. Як реалізувати контроль доступу на рівні програмного коду?
7. Що таке безпечне логування та яких даних не можна логувати?
8. Поясніть механізм SQL Injection та способи його усунення (prepared statements, ORM).
9. У чому різниця між XSS та CSRF?
10. Як програмно реалізувати захист від XSS?
11. Як працює CSRF-токен і як його реалізувати у веб-застосунку?
12. Що таке Broken Access Control? Наведіть приклад реалізації вразливості.
13. Поясніть небезпеку Command Injection та способи захисту.
14. Які практики допомагають уникнути вразливостей при роботі з файлами?
15. Поясніть різницю між хешуванням і шифруванням.
16. Чому для зберігання паролів використовують bcrypt або Argon2?
17. Як реалізувати безпечну автентифікацію користувачів у веб-застосунку?
18. Що таке JWT і які ризики його неправильного використання?
19. Поясніть принципи реалізації RBAC.
20. Як забезпечити захищену передачу даних між клієнтом і сервером?
21. Які помилки виникають при неправильному управлінні криптографічними ключами?
22. Поясніть відмінність між SAST та DAST.
23. Що таке CVE та як перевірити залежності проєкту на наявність вразливостей?
24. Які типові вразливості виявляє статичний аналіз коду?
25. Що таке secure code review і які його етапи?
26. Як інтегрувати перевірку безпеки у CI/CD pipeline?
27. Які переваги автоматизованого тестування безпеки?
28. Поясніть принцип Defense in Depth.

29. Що таке Zero Trust і як його реалізувати у програмній системі?
30. Які типові вразливості REST API та способи їх усунення?
31. Як забезпечити контроль доступу у мікросервісній архітектурі?
32. Які ризики виникають при використанні контейнерів (Docker) та як їх мінімізувати?
33. Що таке DevSecOps і як інтегрувати безпеку в процес розробки?
34. Опишіть базові кроки реагування на інцидент безпеки в програмному проєкті.

Приклади практичних завдань

Приклад практичного завдання 1. Валідація введення

Завдання:

1. Напишіть програму, яка приймає від користувача текст або числа.
2. Додайте перевірку, щоб користувач не міг ввести небезпечні символи (наприклад, <, >, ', ", ;).
3. Продемонструйте, як неправильна валідація може викликати помилку або вразливість.

Очікуваний результат: програма приймає тільки коректні дані, показує приклад неправильного введення та пояснює, чому це небезпечно.

Приклад практичного завдання 2. Хешування пароля

Завдання:

1. Напишіть просту програму, яка приймає пароль від користувача.
2. Використайте бібліотеку для хешування (наприклад, `hashlib` у Python).
3. Збережіть хеш у змінній та продемонструйте перевірку пароля на вході.

Очікуваний результат: програма не зберігає пароль у відкритому вигляді, а перевіряє правильність введеного пароля через порівняння хешів.

Приклад практичного завдання 3. Виявлення простої вразливості

Завдання:

1. Візьміть невеликий готовий скрипт (Python або Java), який працює з текстовим введенням користувача.
2. Проаналізуйте код і знайдіть, де може виникнути небезпека (наприклад, використання `eval()` або прямий запис даних у файл без перевірки).
3. Виправте код, зробивши його безпечним.

Очікуваний результат: виправлений код без простих вразливостей, короткий опис того, що було небезпечним.

Шкала оцінювання знань здобувачів освіти з освітніх компонентів, де формою контролю є залік

Оцінка в балах	Лінгвістична оцінка
90–100	Зараховано
82–89	
75–81	

67–74	
60–66	
1–59	Незараховано (необхідне перескладання)

VII. Рекомендована література та інтернет-ресурси

1. Савченко В. М., Мнушка О. В. Сучасні технології безпечного програмування : навч.-метод. посібник [Електронний ресурс] / В. М. Савченко, О. В. Мнушка. – Харків : НТУ “ХПІ”, 2025. – 112 с. – Режим доступу: https://repository.kpi.kharkov.ua/entities/publication/0a622f52-1f78-4429-a8da-afcaf360583a
2. Давидов В. В., Семенов С. Г., Кучук Н. В. та ін. Сучасні технології безпечного програмування : навч.-метод. посібник [Електронний ресурс] / В. В. Давидов [та ін.]; Нац. техн. ун-т “Харків. політехн. ін-т”. – Харків : Форт, 2021. – 117 с. – Режим доступу: https://repository.kpi.kharkov.ua/items/be2f4a9e-09e3-48b6-a25c-3acae016caee
3. Савченко В. М., Мнушка О. В. Modern Secure Programming Technologies. Workshops [Електронний ресурс] / V. Savchenko, O. Mnushka. – Kharkiv : FOP Brovin O. V., 2024. – 136 с. – Режим доступу: https://repository.kpi.kharkov.ua/entities/publication/dedea443-801f-47f4-be27-75113e71e88a
4. Спасітєлєва С. О. Технології безпечного програмування [Електронний ресурс] / С. О. Спасітєлєва. – Київ : Ін-т Київ. ун-ту ім. Б. Грінченка. – Навч.-метод. мат-ли. – Режим доступу: https://elibrary.kubg.edu.ua/id/eprint/27448/
5. Горбенко І. Д., Олешко О. І., Герцог А. М. Безпека програмного забезпечення та безпечне програмування [Електронний ресурс] / І. Д. Горбенко, О. І. Олешко, А. М. Герцог // Прикладна радіоелектроніка : наук.-техн. журн. – Х. : ХНУРЕ, 2011. – № 2. – С. 244–248. – Режим доступу: https://openarchive.nure.ua/entities/publication/b77f60c5-c086-48f2-9108-1446e3eff6dc
6. OWASP Foundation. OWASP Top 10 – 2021 : The Ten Most Critical Web Application Security Risks [Електронний ресурс]. – Режим доступу: https://owasp.org/Top10/2025/
7. Силабус освітнього компонента “Сучасні технології безпечного програмування” [Електронний ресурс]. – Режим доступу: https://web.kpi.kharkov.ua/cep/wp-content/uploads/sites/217/2025/09/SP1_Suchasni_tehnologiyi_bezpechnogo_programuvannya.pdf
8. Викладач - Кібербезпека. Лекція 7.1 Безпечне програмування. SDLC. Атаки. Проблеми якості, вразливості. Захист., 2024. *YouTube*. URL: https://www.youtube.com/watch?v=exOIBCZCLKQ