ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
СТАВРОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ

C=RU, S=Ставропольский край, L=г. Ставрополь, T=проректор по учебной работе, OU=ректорат, O="ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ "" СТАВРОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ""", СНИЛС=15230589951, ИНН=263411176761, E=s.a.maksimovich@stgau.ru, G=Светлана Александровна, SN=Максимович, CN=Максимович Светлана Александровна

УТВЕРЖДАЮ

Директор/Декан
факультета цифровых технологий
Шлаев Дмитрий Валерьевич

_____

_____

«___» _____ 20___ г.

Рабочая программа дисциплины

Б1.О.14 Технологии разработки защищенного ПО

09.04.03 Прикладная информатика

Искусственный интеллект в кибербезопасности

магистр

очная

Ставрополь, 2026

1. Цель дисциплины

приобретение компетенций в области обеспечения информационной безопасности разрабатываемого ПО путем освоения передовых языковых средств (Rust), методологий безопасного кодинга, статического и динамического анализа, обеспечивающих минимизацию человеческого фактора и исключение критических уязвимостей на этапе компиляции.

2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Процесс изучения дисциплины направлен на формирование следующих компетенций ОП ВО и овладение следующими результатами обучения по дисциплине:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Перечень планируемых результатов обучения по дисциплине
ОПК-2 Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач	ОПК-2.1 Понимает методологические основы современных информационно-коммуникационных и интеллектуальных технологий для решения профессиональных задач	**знает** методологические основы современных информационно-коммуникационных и интеллектуальных технологий для решения профессиональных задач **умеет** применять методологические основы современных информационно-коммуникационных и интеллектуальных технологий для решения профессиональных задач **владеет навыками** навыками работы с современными информационно-коммуникационными и интеллектуальными технологиями для решения профессиональных задач
ОПК-2 Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач	ОПК-2.3 Разрабатывает оригинальные алгоритмы и программные средства, в том числе с использованием интеллектуальных технологий, для решения профессиональных задач	**знает** алгоритмы и программные средства, в том числе с использованием интеллектуальных технологий, для решения профессиональных задач **умеет** разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием интеллектуальных технологий, для решения профессиональных задач **владеет навыками** навыками разработки оригинальных алгоритмов и программных средств с использованием интеллектуальных технологий для решения профессиональных задач
ОПК-5 Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем	ОПК-5.2 Модернизирует программное и аппаратное обеспечения информационных и автоматизированных систем для решения профессиональных задач	**знает** программное и аппаратное обеспечения информационных и автоматизированных систем для решения профессиональных задач **умеет** модернизировать программное и аппаратное обеспечения информационных и автоматизированных систем для решения

		профессиональных задач **владеет навыками** навыками модернизации программного и аппаратного обеспечения информационных и автоматизированных систем для решения профессиональных задач
ОПК-5 Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем	ОПК-5.3 Разрабатывает и выбирает программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач	**знает** программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач **умеет** выбирать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач **владеет навыками** навыками разработки программного и аппаратного обеспечения информационных и автоматизированных систем для решения профессиональных задач
ПК-2 Способен управлять жизненным циклом программных продуктов и проектами в области информационных технологий	ПК-2.1 Управляет процессом разработки программного обеспечения, осуществляя планирование, контроль исполнения и корректировку планов	**знает** как осуществлять планирование, контроль исполнения и корректировку планов при разработке ПО **умеет** управлять процессом разработки программного обеспечения, осуществляя планирование, контроль исполнения и корректировку планов **владеет навыками** навыками управления процессом разработки программного обеспечения, осуществляя планирование, контроль исполнения и корректировку планов
ПК-4 Способен администрировать средства защиты информации и обеспечивать безопасность информационных систем	ПК-4.1 Администрирует средства защиты информации прикладного и системного программного обеспечения (ОС, СУБД, приложения), настраивая политики безопасности и контролируя их соблюдение	**знает** средства защиты информации прикладного и системного программного обеспечения (ОС, СУБД, приложения), настраивая политики безопасности и контролируя их соблюдение **умеет** администрировать средства защиты информации прикладного и системного программного обеспечения (ОС, СУБД, приложения), настраивая политики безопасности и контролируя их соблюдение **владеет навыками** навыками администрирования средств защиты информации прикладного и системного ПО, настраивая политики безопасности и контролируя их соблюдение

3. Место дисциплины в структуре образовательной программы

Дисциплина «Технологии разработки защищенного ПО» является дисциплиной обязательной части программы.

Изучение дисциплины осуществляется в 3семестре(-ах).

Для освоения дисциплины «Технологии разработки защищенного ПО» студенты используют знания, умения и навыки, сформированные в процессе изучения дисциплин:

Ознакомительная практика

Прикладная статистика и анализ данных

Проектирование инфраструктуры и архитектуры ИС

Современные интеллектуальные системы и технологии

Управление информационной безопасностью

Безопасность облачных технологий

Программирование высокопроизводительных систем

Машинное обучение в кибербезопасности

Управление проектами жизненного цикла ИС

Освоение дисциплины «Технологии разработки защищенного ПО» является необходимой основой для последующего изучения следующих дисциплин:

Подготовка к сдаче и сдача государственного экзамена

Подготовка к процедуре защиты и защита выпускной квалификационной работы

Технологическая (проектно-технологическая) практика

Преддипломная практика

Технологии непрерывной разработки и безопасности

Поведенческий анализ и мониторинг защищенности с использованием ИИ

Обнаружение и анализ вредоносного программного обеспечения

4. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу с обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины «Технологии разработки защищенного ПО» в соответствии с рабочим учебным планом и ее распределение по видам работ представлены ниже.

Семестр	Трудоемкость час/з.е.	Контактная работа с преподавателем, час			Самостоятельная работа, час	Контроль, час	Форма промежуточной аттестации (форма контроля)
		лекции	практические занятия	лабораторные занятия			
3	108/3	12		24	72		За
в т.ч. часов: в интерактивной форме		4		6			

Семестр	Трудоемкость час/з.е.	Внеаудиторная контактная работа с преподавателем, час/чел					
		Курсовая работа	Курсовой проект	Зачет	Дифференцированный зачет	Консультации перед экзаменом	Экзамен
3	108/3			0.12			

5. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

№	Наименование раздела/темы	Семестр	Количество часов					Формы текущего контроля успеваемости и промежуточной аттестации	Оценочное средство проверки результатов достижения индикаторов компетенций	Код индикаторов достижения компетенций
			всего	Лекции	Семинарские занятия		Самостоятельная работа			
					Практические	Лабораторные				
1.	1 раздел. Технологии разработки защищенного ПО									
1.1.	Введение в безопасную разработку. Модели угроз и экосистема Rust	3	6	2		4	12		Устный опрос	ОПК-2.1, ОПК-2.3, ОПК-5.2, ОПК-5.3, ПК-2.1, ПК-4.1
1.2.	Управление памятью как основа защищенного ПО	3	6	2		4	12	КТ 1	Тест	ОПК-2.1, ОПК-2.3, ОПК-5.2, ОПК-5.3, ПК-2.1, ПК-4.1
1.3.	Аудит кода и статический анализ безопасности (SAST)	3	6	2		4	12		Устный опрос	ОПК-2.1, ОПК-2.3, ОПК-5.2, ОПК-5.3, ПК-2.1, ПК-4.1
1.4.	Криптографические примитивы и безопасное хранение данных	3	6	2		4	12		Устный опрос	ОПК-2.1, ОПК-2.3, ОПК-5.2, ОПК-5.3, ПК-2.1, ПК-4.1

1.5.	Сетевые взаимодействия и защита от инъекций	3	6	2		4	12	КТ 2	Тест	ОПК-2.1, ОПК-2.3, ОПК-5.2, ОПК-5.3, ПК-2.1, ПК-4.1
1.6.	Финальная сборка, аудит и безопасный релиз (DevSecOps)	3	6	2		4	12		Устный опрос	ОПК-2.1, ОПК-2.3, ОПК-5.2, ОПК-5.3, ПК-2.1, ПК-4.1
	Промежуточная аттестация	За								
	Итого		108	12		24	72			
	Итого		108	12		24	72			

5.1. Лекционный курс с указанием видов интерактивной формы проведения занятий

Тема лекции (и/или наименование раздел) (вид интерактивной формы проведения занятий)/ (практическая подготовка)	Содержание темы (и/или раздела)	Всего, часов / часов интерактивных занятий/ практическая подготовка
Введение в безопасную разработку. Модели угроз и экосистема Rust	Жизненный цикл безопасной разработки (SSDLC). Классификация уязвимостей (CWE, OWASP Top 10). Почему C/C++ опасны для критических систем. Философия безопасности Rust: владение (Ownership), заимствование (Borrowing), времена жизни (Lifetimes) как механизмы статической защиты памяти.	2/2
Управление памятью как основа защищенного ПО	Сравнение моделей памяти: ручное управление (C), сборка мусора (Java/Go) и модель владения (Rust). Безопасность в многопоточных средах: Send и Sync. Обработка ошибок без исключений (Result<T, E> и Option<T>). Предотвращение состояния гонки данных (Data Races) через систему типов.	2/2
Аудит кода и статический анализ безопасности (SAST)	Инструменты статического анализа для защищенного ПО: Clippy, Rust Analyzer, MIRI (интерпретатор неопределенного поведения). Интеграция с CI/CD. Работа с небезопасным кодом (unsafe Rust). Когда он допустим, а когда — критическая уязвимость. Инкапсуляция unsafe в безопасные абстракции.	2/-
Криптографические примитивы и безопасное хранение данных	Криптография для разработчика: хеширование (пароли), симметричное/асимметричное шифрование. Ошибки конфигурации	2/2

	библиотек.Использование крипто-крейтов (ring, rust-crypto, orion). Безопасная работа с секретами в памяти (Zeroize). Атаки по сторонним каналам (время, питание) — защита на уровне кода.	
Сетевые взаимодействия и защита от инъекций	Безопасный сетевой стек в Rust (tokio + rustls). Отказ от OpenSSL в пользу криптографии на Rust. Защита от XSS, SQL-инъекций и Command Injection на бэкенде. Проектирование устойчивых микросервисов: защита от DDoS (таймауты, лимиты), валидация входных данных через типажи (Traits) без потери производительности.	2/2
Финальная сборка, аудит и безопасный релиз (DevSecOps)	Деплой защищенного ПО: оптимизация размера (strip, link-time optimization), контейнеризация (Docker + distroless). Анализ зависимостей (SBOM — Software Bill of Materials). Методология тестирования на проникновение (Pentest) для Rust-приложений. Фаззинг-тестирование (cargo-fuzz и libfuzzer) для поиска логических уязвимостей.	2/-
Итого		12

5.2.2. Лабораторные занятия с указанием видов проведения занятий в интерактивной форме

Наименование раздела дисциплины	Формы проведения и темы занятий (вид интерактивной формы проведения занятий)/(практическая подготовка)	Всего, часов / часов интерактивных занятий/ практическая подготовка	
		вид	часы
Введение в безопасную разработку. Модели угроз и экосистема Rust	Настройка окружения (Rustup, Cargo, Clippy). Реализация «безопасного» аналога классической уязвимости (UAF/Double Free) на C++ и демонстрация того, как Rust не дает скомпилировать такой код.	лаб.	4
Управление памятью как основа защищенного ПО	Разработка модуля безопасной работы с файловой системой на Rust. Обработка критических ошибок без паники (panic). Использование std::fs и кастомных типов ошибок.	лаб.	4
Аудит кода и статический анализ безопасности (SAST)	удит чужого кода на Rust с использованием cargo-audit и cargo-deny. Поиск и исправление UB (неопределенного поведения) в сырых указателях.	лаб.	4
Криптографические примитивы и безопасное хранение данных	Реализация сервиса для безопасного хранения паролей (менеджер секретов) на Rust с использованием zeroize для затирания ключей из памяти и argon2 для хеширования.	лаб.	4
Сетевые взаимодействия и защита от инъекций	Сборка защищенного HTTP/TLS сервера (actix-web + rustls). Написание middleware для валидации и санитизации входных JSON-данных, иммунная к инъекциям.	лаб.	4
Финальная сборка,	Комплексная работа: Сборка итогового	лаб.	4

аудит и безопасный релиз (DevSecOps)	микро-сервиса, запуск cargo-fuzz на критический парсер входных данных, составление отчета о безопасности (Threat Modeling Report) и релизного артефакта.		

5.3. Курсовой проект (работа) учебным планом не предусмотрен

5.4. Самостоятельная работа обучающегося

Темы и/или виды самостоятельной работы	Часы
Самостоятельно изучение лекционного материала и подготовка к лабораторным занятиям	12
Самостоятельное изучение лекционного материала и подготовка к лабораторным занятиям	12
самостоятельное изучение лекционного материала и подготовка к лабораторным занятиям	12
Самостоятельное изучение лекционного материала и подготовка к лабораторным занятиям	12
Самостоятельное изучение лекционного материала и подготовка к лабораторным занятиям	12
Самостоятельное изучение лекционного материала и подготовка к лабораторным занятиям	12

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

Учебно-методическое обеспечение для самостоятельной работы обучающегося по дисциплине «Технологии разработки защищенного ПО» размещено в электронной информационно-образовательной сре-де Университета и доступно для обучающегося через его личный кабинет на сайте Университета. Учебно-методическое обеспечение включает:

1. Рабочую программу дисциплины «Технологии разработки защищенного ПО».

2.Методические рекомендации для организации самостоятельной работы обучающегося по дисциплине «Технологии разработки защищенного ПО».

3. Методические рекомендации по выполнению письменных работ () (при наличии).

4. Методические рекомендации по выполнению контрольной работы студентами заочной формы обучения (при наличии)

5. Методические указания по выполнению курсовой работы (проекта) (при наличии).

Для успешного освоения дисциплины, необходимо самостоятельно детально изучить представленные темы по рекомендуемым источникам информации:

№ п/п	Темы для самостоятельного изучения	Рекомендуемые источники информации (№ источника)		
		основная (из п.8 РПД)	дополнительная (из п.8 РПД)	метод. лит. (из п.8 РПД)
1	Введение в безопасную разработку. Модели угроз и экосистема Rust. Самостоятельно изучение лекционного материала и подготовка к лабораторным занятиям	Л1.1, Л1.2	Л2.1	Л3.1, Л3.2
2	Управление памятью как основа защищенного ПО. Самостоятельное изучение лекционного материала и подготовка к лабораторным занятиям	Л1.1, Л1.2	Л2.1	Л3.1, Л3.2
3	Аудит кода и статический анализ безопасности (SAST). самостоятельное изучение лекционного материала и подготовка к лабораторным занятиям	Л1.1, Л1.2	Л2.1	Л3.1, Л3.2
4	Криптографические примитивы и безопасное хранение данных. Самостоятельное изучение лекционного материала и подготовка к лабораторным занятиям	Л1.1, Л1.2	Л2.1	Л3.1, Л3.2
5	Сетевые взаимодействия и защита от инъекций. Самостоятельное изучение лекционного материала и подготовка к лабораторным занятиям	Л1.1, Л1.2	Л2.1	Л3.1, Л3.2
6	Финальная сборка, аудит и безопасный релиз (DevSecOps). Самостоятельное изучение лекционного материала и подготовка к лабораторным занятиям	Л1.1, Л1.2	Л2.1	Л3.1, Л3.2

7. Фонд оценочных средств (оценочных материалов) для проведения промежуточной аттестации обучающихся по дисциплине «Технологии разработки защищенного ПО»

7.1. Перечень индикаторов компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Индикатор компетенции (код и содержание)	Дисциплины/элементы программы (практики, ГИА), участвующие в формировании индикатора компетен-ции	

7.2. Критерии и шкалы оценивания уровня усвоения индикатора компетенций, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Оценка знаний, умений и навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций по дисциплине «Технологии разработки защищенного ПО» проводится в форме текущего контроля и промежуточной аттестации.

Текущий контроль проводится в течение семестра с целью определения уровня усвоения обучающимися знаний, формирования умений и навыков, своевременного выявления преподавателем недостатков в подготовке обучающихся и принятия необходимых мер по её корректировке, а также для совершенствования методики обучения, организации учебной работы и оказания индивидуальной помощи обучающемуся.

Промежуточная аттестация по дисциплине «Технологии разработки защищенного ПО» проводится в виде Зачет.

За знания, умения и навыки, приобретенные студентами в период их обучения, выставляются оценки «ЗАЧТЕНО», «НЕ ЗАЧТЕНО». (или «ОТЛИЧНО», «ХОРОШО», «УДОВЛЕТВОРИТЕЛЬНО», «НЕУДОВЛЕТВОРИТЕЛЬНО» для дифференцированного зачета/экзамена)

Для оценивания знаний, умений, навыков и (или) опыта деятельности в университете применяется балльно-рейтинговая система оценки качества освоения образовательной программы. Оценка проводится при проведении текущего контроля успеваемости и промежуточных аттестаций обучающихся. Рейтинговая оценка знаний является интегрированным показателем качества теоретиче-ских и практических знаний и навыков студентов по дисциплине.

Состав балльно-рейтинговой оценки студентов очной формы обучения

Для студентов очной формы обучения знания по осваиваемым компетенциям формируются на лекционных и практических занятиях, а также в процессе самостоятельной подготовки.

В соответствии с балльно-рейтинговой системой оценки, принятой в Университете студентам начисляются баллы по следующим видам работ:

№ контрольной точки	Оценочное средство результатов индикаторов достижения компетенций	Максимальное количество баллов
3 семестр		
КТ 1	Тест	15
КТ 2	Тест	15
Сумма баллов по итогам текущего контроля		**30**
Посещение лекционных занятий		20
Посещение практических/лабораторных занятий		20
Результативность работы на практических/лабораторных занятиях		30
Итого		100

№ контрольной точки	Оценочное средство результатов индикаторов достижений компетенций	Максимальное количество баллов	Критерии оценки знаний студентов
3 семестр			

КТ 1	Тест	15	11-15 баллов выставляется обучающемуся, если тестовые задания выполняются на 85% и выше; 8-10 баллов выставляется обучающемуся, если тестовые задания выполняются на 70 - 84%; 5-7 баллов выставляется обучающемуся, если тестовые задания выполняются на 55 – 69 %; 1-4 балла выставляется обучающемуся, если тестовые задания выполняются на 45 – 54%; 0 баллов выставляется обучающемуся, если тестовые задания выполняются на 44% и меньше.
КТ 2	Тест	15	11-15 баллов выставляется обучающемуся, если тестовые задания выполняются на 85% и выше; 8-10 баллов выставляется обучающемуся, если тестовые задания выполняются на 70 - 84%; 5-7 баллов выставляется обучающемуся, если тестовые задания выполняются на 55 – 69 %; 1-4 балла выставляется обучающемуся, если тестовые задания выполняются на 45 – 54%; 0 баллов выставляется обучающемуся, если тестовые задания выполняются на 44% и меньше.

Критерии и шкалы оценивания результатов обучения на промежуточной аттестации

При проведении итоговой аттестации «зачет» («дифференцированный зачет», «экзамен») преподавателю с согласия студента разрешается выставлять оценки («отлично», «хорошо», «удовлетворительно», «зачет») по результатам набранных баллов в ходе текущего контроля успеваемости в семестре по выше приведенной шкале.

В случае отказа – студент сдает зачет (дифференцированный зачет, экзамен) по приведенным выше вопросам и заданиям. Итоговая успеваемость (зачет, дифференцированный зачет, экзамен) не может оцениваться ниже суммы баллов, которую студент набрал по итогам текущей и промежуточной успеваемости.

При сдаче (зачета, дифференцированного зачета, экзамена) к заработанным в течение семестра студентом баллам прибавляются баллы, полученные на (зачете, дифференцированном зачете, экзамене) и сумма баллов переводится в оценку.

Критерии и шкалы оценивания ответа на зачете

По дисциплине «Технологии разработки защищенного ПО» к зачету допускаются студенты, выполнившие и сдавшие практические работы по дисциплине, имеющие ежемесячную аттестацию и без привязке к набранным баллам. Студентам, набравшим более 65 баллов, зачет выставляется по результатам текущей успеваемости, студенты, не набравшие 65 баллов, сдают зачет по вопросам, предусмотренным РПД. Максимальная сумма баллов по промежуточной аттестации (зачету) устанавливается в 15 баллов

Вопрос билета	Количество баллов
Теоретический вопрос	до 5
Задания на проверку умений	до 5

Задания на проверку навыков	до 5

Теоретический вопрос

5 баллов выставляется студенту, полностью освоившему материал дисциплины или курса в соответствии с учебной программой, включая вопросы рассматриваемые в рекомендованной программой дополнительной справочно-нормативной и научно-технической литературы, свободно владеющему основными понятиями дисциплины. Требуется полное понимание и четкость изложения ответов по экзаменационному заданию (билету) и дополнительным вопросам, заданных экзаменатором. Дополнительные вопросы, как правило, должны относиться к материалу дисциплины или курса, не отраженному в основном экзаменационном задании (билете) и выявляют полноту знаний студента по дисциплине.

4 балла заслуживает студент, ответивший полностью и без ошибок на вопросы экзаменационного задания и показавший знания основных понятий дисциплины в соответствии с обязательной программой курса и рекомендованной основной литературой.

3 балла дан недостаточно полный и недостаточно развернутый ответ. Логика и последовательность изложения имеют нарушения. Допущены ошибки в раскрытии понятий, употреблении терминов. Студент не способен самостоятельно выделить существенные и несущественные признаки и причинно-следственные связи. Студент может конкретизировать обобщенные знания, доказав на примерах их основные положения только с помощью преподавателя. Речевое оформление требует поправок, коррекции.

2 балла дан неполный ответ, представляющий собой разрозненные знания по теме вопроса с существенными ошибками в определениях. Присутствуют фрагментарность, нелогичность изложения. Студент не осознает связь данного понятия, теории, явления с другими объектами дисциплины. Отсутствуют выводы, конкретизация и доказательность изложения. Речь неграмотная. Дополнительные и уточняющие вопросы преподавателя не приводят к коррекции ответа студента не только на поставленный вопрос, но и на другие вопросы дисциплины.

1 балл дан неполный ответ, представляющий собой разрозненные знания по теме вопроса с существенными ошибками в определениях. Присутствуют фрагментарность, нелогичность изложения. Студент не осознает связь данного понятия, теории, явления с другими объектами дисциплины. Отсутствуют выводы, конкретизация и доказательность изложения. Речь неграмотная. Дополнительные и уточняющие вопросы преподавателя не приводят к коррекции ответа студента не только на поставленный вопрос, но и на другие вопросы дисциплины.

0 баллов - при полном отсутствии ответа, имеющего отношение к вопросу.

Задания на проверку умений и навыков

5 баллов Задания выполнены в обозначенный преподавателем срок, письменный отчет без замечаний. Работа выполнена в полном объеме с соблюдением необходимой последовательности.

4 балла Задания выполнены в обозначенный преподавателем срок, письменный отчет с небольшими недочетами.

2 баллов Задания выполнены с задержкой, письменный отчет с недочетами. Работа выполнена не полностью, но объем выполненной части таков, что позволяет получить правильные результаты и выводы.

1 баллов Задания выполнены частично, с большим количеством вычислительных ошибок, объем выполненной части работы не позволяет сделать правильных выводов.

0 баллов Задания выполнены, письменный отчет не представлен или работа выполнена не полностью, и объем выполненной части работы не позволяет сделать правильных выводов.

7.3. Примерные оценочные материалы для текущего контроля успеваемости и промежуточной аттестации по итогам освоения дисциплины «Технологии разработки защищенного ПО»

Какой механизм языка Rust гарантирует безопасность работы с памятью на этапе компиляции?

А) Сборщик мусора (Garbage Collector)

B) Система владения (Ownership) с правилами заимствования и временами жизни

C) Виртуальная машина с байт-кодом

D) Динамическая проверка типов во время выполнения

Что произойдет при попытке скомпилировать код, который создает изменяемую ссылку (&mut) на данные, уже имеющие неизменяемую ссылку (&) в той же области видимости?

A) Компиляция пройдет успешно, но будет предупреждение

B) Компиляция не удастся с ошибкой проверки заимствований (borrow checker)

C) Программа упадет во время выполнения с паникой (panic)

D) Данные будут автоматически скопированы в новую область памяти

Какая уязвимость НЕВОЗМОЖНА в безопасной части Rust (без использования unsafe)?

A) SQL-инъекция

B) Переполнение буфера (Buffer Overflow)

C) Гонка данных (Data Race) в многопоточном коде

D) Отказ в обслуживании (DoS) через бесконечный цикл

Какой трейт (trait) гарантирует безопасность передачи владения данными между потоками в Rust?

A) Clone

B) Drop

C) Send

D) Default

Какой инструмент статического анализа рекомендуется использовать для проверки "стиля" и обнаружения распространенных ошибок в коде Rust?

A) Valgrind

B) Clippy

C) GDB

D) AddressSanitizer

Что означает ключевое слово unsafe в Rust?

A) Код будет выполняется медленнее

B) Код доверяется программисту и может выполнять операции, не проверяемые компилятором на безопасность

C) Код будет автоматически удален из финальной сборки

D) Код работает только в отладочном режиме (debug)

Как правильно обработать ситуацию, когда результат операции может отсутствовать (например, поиск в коллекции)?

A) Использовать panic!() для немедленного завершения

B) Использовать тип Option<T> и сопоставление с образцом (match)

C) Использовать null и проверять на null

D) Игнорировать результат и продолжить выполнение

Какая библиотека (крейт) рекомендуется для безопасного затирания секретов (паролей, ключей) из памяти?

A) serde

B) tokio

C) zeroize

D) log

Что такое SBOM (Software Bill of Materials) в контексте безопасной разработки?

A) Инструмент для профилирования производительности

B) Перечень всех компонентов и зависимостей, используемых в проекте

C) Методология тестирования на проникновение

D) Система управления версиями кода

Какой механизм в Rust позволяет создавать абстракции с нулевой стоимостью (zero-cost abstractions)?

A) Виртуальные таблицы (vtable) как в C++

B) Монады как в Haskell

C) Дженерики (Generics) и мономорфизация на этапе компиляции

D) Динамическая диспетчеризация через dyn Trait

Сравнительный анализ моделей управления памятью в C++, Go и Rust с точки зрения безопасности.

Акцент: уязвимости памяти (UAF, Double Free, Buffer Overflow) и способы их предотвращения.

Моделирование угроз по методологии STRIDE для приложений, разработанных на Rust.

Акцент: применение к микросервисной и клиент-серверной архитектуре.

Эволюция SSDLC (Secure Software Development Lifecycle): от Waterfall к DevSecOps.

Акцент: место Rust в современном безопасном конвейере разработки.

Система владения (Ownership) в Rust как архитектурный паттерн безопасности.

Акцент: сравнение с умными указателями в C++ (unique_ptr, shared_ptr).

Правила заимствования и времена жизни (Lifetimes): как компилятор Rust предотвращает висячие ссылки.

Акцент: примеры кода, демонстрирующие ошибки компиляции.

Обработка ошибок без исключений: типы Option<T> и Result<T, E> как способ защиты от непредвиденных состояний.

Акцент: сравнение с механизмами исключений в Java/Python.

Clippy и Rust Analyzer как инструменты Security Code Review.

Акцент: категории линтов, обнаружение потенциальных уязвимостей, настройка в CI/CD.

MIRI — интерпретатор неопределенного поведения: поиск UB в коде на Rust.

Акцент: примеры UB в unsafe-коде и их обнаружение.

Фаззинг-тестирование с cargo-fuzz: методология и примеры обнаружения уязвимостей.

Акцент: интеграция фаззинга в процесс разработки, кейсы из реальных проектов.

Правила безопасного использования unsafe в Rust: инкапсуляция и аудит.

Акцент: когда unsafe необходим и как минимизировать его поверхность.

Разработка безопасной обертки над C-библиотекой на примере OpenSSL или SQLite.

Акцент: работа с FFI, защита от ошибок работы с сырыми указателями.

Сравнение криптографических библиотек в экосистеме Rust: ring, orion, rust-crypto, dalek.

Акцент: безопасность API, производительность, поддержка алгоритмов.

Безопасное хеширование паролей на Rust: реализация Argon2 с использованием крейта argon2.

Акцент: настройка параметров, защита от атак перебора.

Защита секретов в памяти с использованием крейта zeroize.

Акцент: предотвращение утечек через дампы памяти, отладчики и анализ ядра.

Безопасный сетевой стек на Rust: реализация TLS-сервера с использованием tokio и rustls.

Акцент: сравнение с OpenSSL, конфигурация, защита от атак на TLS.

Защита REST API на Rust от инъекционных атак (SQL, Command Injection).

Акцент: использование строгой типизации и валидации через типажи.

Проектирование устойчивого к DDoS микросервиса на Rust (таймауты, лимиты, backpressure).

Акцент: настройка tokio, использование tower middleware.

Контейнеризация Rust-приложений: создание минимальных Docker-образов и distroless-контейнеров.

Акцент: минимизация поверхности атаки, сканирование уязвимостей образов.

Составление SBOM (Software Bill of Materials) для Rust-проектов: инструменты и форматы.

Акцент: cargo-audit, cargo-deny, интеграция с CI/CD.

Применение Rust в критической инфраструктуре: опыт использования в Android, AWS, Microsoft.

Акцент: конкретные проекты (Android Rust, Firecracker, Azure IoT Edge), статистика безопасности, перспективы.

Какой механизм языка Rust гарантирует безопасность работы с памятью на этапе компиляции?

A) Сборщик мусора (Garbage Collector)

B) Система владения (Ownership) с правилами заимствования и временами жизни

C) Виртуальная машина с байт-кодом

D) Динамическая проверка типов во время выполнения

Что произойдет при попытке скомпилировать код, который создает изменяемую ссылку (&mut) на данные, уже имеющие неизменяемую ссылку (&) в той же области видимости?

A) Компиляция пройдет успешно, но будет предупреждение

B) Компиляция не удастся с ошибкой проверки заимствований (borrow checker)

C) Программа упадет во время выполнения с паникой (panic)

D) Данные будут автоматически скопированы в новую область памяти

Какая уязвимость НЕВОЗМОЖНА в безопасной части Rust (без использования unsafe)?

A) SQL-инъекция

B) Переполнение буфера (Buffer Overflow)

C) Гонка данных (Data Race) в многопоточном коде

D) Отказ в обслуживании (DoS) через бесконечный цикл

Какой трейт (trait) гарантирует безопасность передачи владения данными между потоками в Rust?

A) Clone

B) Drop

C) Send

D) Default

Какой инструмент статического анализа рекомендуется использовать для проверки "стиля" и обнаружения распространенных ошибок в коде Rust?

A) Valgrind

B) Clippy

C) GDB

D) AddressSanitizer

Что означает ключевое слово unsafe в Rust?

A) Код будет выполняется медленнее

B) Код доверяется программисту и может выполнять операции, не проверяемые компилятором на безопасность

C) Код будет автоматически удален из финальной сборки

D) Код работает только в отладочном режиме (debug)

Как правильно обработать ситуацию, когда результат операции может отсутствовать (например, поиск в коллекции)?

A) Использовать panic!() для немедленного завершения

B) Использовать тип Option<T> и сопоставление с образцом (match)

C) Использовать null и проверять на null

D) Игнорировать результат и продолжить выполнение

Какая библиотека (крейт) рекомендуется для безопасного затирания секретов (паролей, ключей) из памяти?

A) serde

B) tokio

C) zeroize

D) log

Что такое SBOM (Software Bill of Materials) в контексте безопасной разработки?

A) Инструмент для профилирования производительности

B) Перечень всех компонентов и зависимостей, используемых в проекте

C) Методология тестирования на проникновение

D) Система управления версиями кода

Какой механизм в Rust позволяет создавать абстракции с нулевой стоимостью (zero-cost abstractions)?

A) Виртуальные таблицы (vtable) как в C++

B) Монады как в Haskell

C) Дженерики (Generics) и мономорфизация на этапе компиляции

D) Динамическая диспетчеризация через dyn Trait

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

а) Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

основная

Л1.1 Баранова Е. К., Бабаш А. В. Информационная безопасность и защита информации [Электронный ресурс]:учеб. пособие; ВО - Бакалавриат. - Москва: Издательский Центр РИО□, 2022. - 336 с. – Режим доступа: http://znanium.com/catalog/document?id=393765

Л1.2 Мякишев Д. В. Разработка программного обеспечения АСУ ТП на основе объектно-ориентированного подхода [Электронный ресурс]:метод. пособие; ВО - Специалитет. - Вологда: Инфра-Инженерия, 2024. - 128 с. – Режим доступа: https://znanium.ru/catalog/document?id=452449

дополнительная

Л2.1 Лауферман О. В., Лыгина Н. И. Разработка программного продукта: профессиональные стандарты, жизненный цикл, командная работа [Электронный ресурс]:учеб. пособие; ВО - Бакалавриат. - Новосибирск: НГТУ, 2019. - 75 с. – Режим доступа: https://e.lanbook.com/book/152251

б) Методические материалы, разработанные преподавателями кафедры по дисциплине, в соответствии с профилем ОП.

Л3.1 Прохорова О. В. Информационная безопасность и защита информации [Электронный ресурс]:учебник ; ВО - Бакалавриат. - Санкт-Петербург: Лань, 2020. - 124 с. – Режим доступа: https://e.lanbook.com/book/133924

Л3.2 Дергачев К. В., Титарев Д. В. Защита информации: лабораторный практикум [Электронный ресурс]:учеб. пособие ; ВО - Бакалавриат, Магистратура. - Москва: Русайнс, 2024. - 158 с. – Режим доступа: https://book.ru/book/952795

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

№	Наименование ресурса сети «Интернет»	Электронный адрес ресурса
1	расов, А. В. Разработка защищенного программного обеспечения : учебное пособие / А. В. Красов. — Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2023. — 90 с. — Текст : электронный // Лань : электронно-библиотечная система.— URL: https://e.lanbook.com/book/425906 (дата обращения: 19.06.2026). — Режим доступа: для авториз. пользователей.	https://e.lanbook.com/book/425906
2	Казарин, О. В. Программно-аппаратные средства защиты информации. Защита программного обеспечения : учебник и практикум для вузов / О. В. Казарин, А. С. Забабурин. — Москва : Юрайт, 2025. — 312 с. — (Высшее образование). — ISBN 978-5-534-18998-6. — Текст : электронный // Юрайт : электронно-библиотечная система. — URL: https://urait.ru/bcode/567283 (дата обращения: 19.06.2026). — Режим доступа: для авториз. пользователей.	https://urait.ru/bcode/567283

10. Методические указания для обучающихся по освоению дисциплины

Специфика изучения дисциплины обусловлена формой обучения студентов, ее местом в подготовке бакалавров и временем, отведенным на освоение курса рабочим учебным планом.

Курс обучения делится на время, отведенное для занятий, проводимых в аудиторной форме (лекции, практические и лабораторные занятия) и время, выделенное на внеаудиторное освоение дисциплины, большую часть из которого составляет самостоятельная работа студента.

Лекционная часть учебного курса для студентов проводится в форме обзоров по основным те -мам. Практические и лабораторные занятия предусмотрены для закрепления теоретических знаний, углубленного рассмотрения наиболее сложных проблем дисциплины, выработки навыков структурно-логического построения учебного материала и отработки навыков самостоятельной подготовки.

Самостоятельная работа студента включает в себя изучение теоретического материала курса, выполнение практических заданий, подготовку к контрольно-обобщающим мероприятиям.

Для освоения курса дисциплины студенты должны:

− изучить материал лекционных и практических занятий в полном объеме по разделам курса;

− выполнить задание, отведенное на самостоятельную работу: подготовить и защитить реферат по утвержденной преподавателем теме, подготовиться к собеседованию, тестированию, защите практических работ, контрольной работе;

− продемонстрировать сформированность компетенций, закрепленных за курсом дисциплины во время мероприятий текущего и промежуточного контроля знаний.

Посещение лекционных и практических занятий для студентов очной и заочной формы является обязательным.

Уважительными причинами пропуска аудиторных занятий является:

− освобождение от занятий по причине болезни, выданное медицинским учреждением,

− распоряжение по деканату, приказ по вузу об освобождении в связи с участием в внутривузовских, межвузовских и пр. мероприятиях,

− официально оформленное свободное посещение занятий.

Пропуски отрабатываются независимо от их причины.

Пропущенные темы лекционных занятий должны быть законспектированы в тетради для лекций, конспект представляется преподавателю для ликвидации пропуска. Пропущенные практические занятия отрабатываются в виде устной защиты практического занятия во время консультаций по дисциплине.

Контроль сформированности компетенций в течение семестра проводится в форме устного опроса на практических занятиях, защиту практических работ, выполнения контрольных работ, написания тестового контроля по теоретическому курсу дисциплины.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства и информационных справочных систем (при необходимости).

11.1 Перечень лицензионного программного обеспечения

1. Kaspersky Endpoint Security 12.11 - Антивирус
2. Microsoft Windows Server STDCORE AllLngLicense/Software AssurancePack Academic OLV 16Licenses LevelE AdditionalProduct CoreLic 1Year - Серверная операционная система
3. OPERA - Система управления отелем

11.3 Перечень программного обеспечения отечественного производства

1. Kaspersky Endpoint Security 12.11 - Антивирус

При осуществлении образовательного процесса студентами и преподавателем используются следующие информационно справочные системы: СПС «Консультант плюс», СПС «Гарант».

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

№ п/п	Наименование специальных помещений и помещений для самостоятельной работы	Номер аудитории	Оснащенность специальных помещений и помещений для самостоятельной работы
1	Учебная аудитория для проведения занятий всех типов (в т.ч. лекционного, семинарского, практической подготовки обучающихся), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	315/НК	Оснащение: специализированная мебель на 250 посадочных мест, трибуна для лектора – 1 шт., президиум – 1 шт., видеостена из 9 бесшовный ЖК дисплеев Mercury Full HD 55” ширина-3,1 м высота - 1,7 м , АРМ на основе Intel Core i3 , Монитор Dell 21.5", Клавиатура + мышь , Источник бесперебойного питания 650ВА, Монитор ЖК размер экрана: Dell 21.5", широкоформатная матрица VA с разрешением 1920×1080, отношением сторон 16:9 - 3шт.,микрофонная система Restmoment RX-812 -1шт, Restmoment RX-D58 микрофон делегата -4шт.,АМС настенный громкоговоритель мониторного типа - 6шт., DSPPA микшер-усилитель - 1шт., магнитно-маркерная доска – 1 шт., учебно-наглядные пособия в виде тематических презентаций, информационные плакаты, подключение к сети «Интернет», доступ в электронную информационно-образовательную среду университета, выход в корпоративную сеть университета.
2	Помещение для самостоятельной работы обучающихся, подтверждающее наличие материально-технического обеспечения, с перечнем основного оборудования		
		214/НК библиотека	Специализированная мебель на 130 посадочных мест, персональные компьютеры, моноблоки – 80 шт., копир А3 - 3, принтер матричный - 2, МФУ ч/б – 7 шт., МФУ цветной – 2 шт., принтер ч/б – 8 шт., принтер цветн. - 2 шт., сканер – 2 шт., сканеры штрих-кода - 5, наушники - 10 шт., Wi-Fi оборудование, подключение к сети «Интернет», доступ к российским и международным ресурсам и базам данных, доступ к электронно-библиотечным системам, доступ в электронную информационно-образовательную среду университета. Открытый доступ к фонду учебной, научной и художественной литературы.

13. Особенности реализации дисциплины лиц с ограниченными возможностями здоровья

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники и учебные пособия, иная учебная литература, специальные технические средства обучения коллективного и индивидуального пользования, предоставление услуг ассистента (помощника), оказывающего обучающимся необходимую техническую помощь, а также услуги сурдопереводчиков и тифлосурдопереводчиков.

а) для слабовидящих:

- на промежуточной аттестации присутствует ассистент, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (он помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе записывая под диктовку);
- задания для выполнения, а также инструкция о порядке проведения промежуточной аттестации оформляются увеличенным шрифтом;
- задания для выполнения на промежуточной аттестации зачитываются ассистентом;
- письменные задания выполняются на бумаге, надиктовываются ассистенту;
- обеспечивается индивидуальное равномерное освещение не менее 300 люкс;
- студенту для выполнения задания при необходимости предоставляется увеличивающее устройство;

в) для глухих и слабослышащих:

- на промежуточной аттестации присутствует ассистент, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (он помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе записывая под диктовку);
- промежуточная аттестация проводится в письменной форме;
- обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости поступающим предоставляется звукоусиливающая аппаратура индивидуального пользования;
- по желанию студента промежуточная аттестация может проводиться в письменной форме;

д) для лиц с нарушениями опорно-двигательного аппарата (тяжелыми нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей):

- письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту;
- по желанию студента промежуточная аттестация проводится в устной форме.

Рабочая программа дисциплины «Технологии разработки защищенного ПО» составлена на основе Федеральный государственный образовательный стандарт высшего образования - магистратура по направлению подготовки 09.04.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 916).

Автор (ы)

_____ доц. КИС, кэн Ермакова Анна Николаевна

Рецензенты

_____ доц. КИС, кпн Богданова Светлана Викторовна

_____ доц. КИС, ктн Рачков Валерий Евгеньевич

Рабочая программа дисциплины «Технологии разработки защищенного ПО» рассмотрена на заседании Кафедра информационных систем протокол № 9 от 07.04.2026 г. и признана соответствующей требованиям ФГОС ВО и учебного плана по направлению подготовки 09.04.03 Прикладная информатика

Заведующий кафедрой _____ Березницкий Андрей Сергеевич

Рабочая программа дисциплины «Технологии разработки защищенного ПО» рассмотрена на заседании учебно-методической комиссии Факультет цифровых технологий протокол № 2 от 08.04.2026 г. и признана соответствующей требованиям ФГОС ВО и учебного плана по направлению подготовки 09.04.03 Прикладная информатика

Руководитель ОП _____