

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Б1.О.20
(индекс дисциплины)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Обеспечение безопасности при разработке программного обеспечения
(наименование дисциплины)

по направлению подготовки

09.03.04 Программная инженерия
направленность (профиль)

Программная инженерия с применением ИИ-технологий

Форма обучения: заочная

Год набора: 2024

Общая трудоемкость: 5 ЗЕ

Распределение часов дисциплины по семестрам

Семестр	9	Итого
Форма контроля	экзамен	
Вид занятий		
Лекции	4	4
Лабораторные	-	-
Практические	-	-
ККР	1	1
Промежуточная аттестация	0,35	0,35
Контактная работа	5,35	5,35
Самостоятельная работа	166	166
Контроль	8,65	8,65
Итого	180	180

Рабочую программу составила:

Доцент института цифровых технологий, к.эк.наук, Раченко Т.А.

(должность, ученое звание, степень, Фамилия И.О.)

Рецензирование рабочей программы дисциплины:



Отсутствует



Рецензент

(должность, ученое звание, степень, Фамилия И.О.)

Рабочая программа составлена на основании ФГОС ВО и учебного плана направления подготовки

09.03.04 Программная инженерия

(код и наименование направления подготовки, специальности в соответствии с ФГОС ВПО)

Срок действия рабочей программы дисциплины до «31» августа 2031 г.

УТВЕРЖДЕНО

На заседании института цифровых технологий

(протокол заседания № 1 от «05» сентября 2025 г.)

1. Цель освоения дисциплины

Цель – формирование у обучающихся компетенций в области обеспечения безопасности при разработке программного обеспечения, в том числе для систем с применением технологий искусственного интеллекта, включая методы защиты данных, моделей и процессов разработки.

Задачи:

1. Изучение типовых уязвимостей программного обеспечения и методов их предотвращения, с акцентом на уязвимости в системах, использующих ИИ.
2. Знакомство с принципами проектирования безопасного программного обеспечения, включая защиту данных на всех этапах жизненного цикла разработки.
3. Изучение методов и средств аутентификации и авторизации пользователей в распределённых системах обработки данных.
4. Знакомство с криптографическими методами и средствами защиты данных, включая современные подходы (гомоморфное шифрование, дифференциальная приватность).
5. Изучение протоколов безопасной передачи данных и методов защиты данных при передаче в облачные и распределённые хранилища.
6. Изучение методов обеспечения целостности данных, в том числе для наборов данных, используемых в обучении моделей.
7. Освоение навыков использования инструментальных средств обеспечения безопасности программного обеспечения, включая инструменты для анализа безопасности данных и моделей ML (TensorFlow Privacy, Adversarial Robustness Toolbox и др.).
8. Формирование умения анализировать уязвимости программного обеспечения и разрабатывать политику информационной безопасности для проектов в области программной инженерии с применением ИИ.
9. Овладение приёмами предотвращения, обнаружения и нейтрализации угроз безопасности программных систем, включая атаки на модели машинного обучения (отравление данных, инверсия моделей, атаки с подбором выходных данных).

2. Место дисциплины в структуре ОПОП ВО

Дисциплина относится к **обязательной части** блока Б1.

Дисциплины, на освоении которых базируется данная дисциплина:

Информационные системы и технологии, Управление проектами разработки программного обеспечения, Базы данных и управление данными, Обеспечение качества кода и код ревью.

Дисциплины, для которых необходимы знания, умения, навыки, приобретаемые в результате изучения данной дисциплины:

Выполнение и защита выпускной квалификационной работы.

3. Планируемые результаты обучения

Формируемые и контролируемые компетенции	Индикаторы достижения компетенций	Планируемые результаты обучения
ОПК-3. Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	ОПК-3.4. Применение ИКТ-инструментов для решения профессиональных задач. Соблюдение требований информационной безопасности	Знать: основные принципы и методы управления жизненным циклом данных с учётом информационной безопасности; методы и инструменты обеспечения информационной безопасности в распределённых системах, включая системы с применением ИИ. Уметь: применять инструменты анализа данных для выявления тенденций и закономерностей в контексте безопасности; обосновывать выбор решений на основе анализа данных и оценки рисков; применять методы оптимизации управления жизненным циклом данных с учётом безопасности; оценивать и минимизировать риски в распределённых системах, включая системы ИИ. Владеть: навыками работы с инструментами анализа данных (Excel, Python, Pandas, Spark) для задач безопасности; навыками использования технологий для управления жизненным циклом данных; навыками разработки мер по обеспечению информационной безопасности, включая защиту данных и моделей ИИ.
ОПК-5. Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.4. Установка программного обеспечения. Настройка аппаратных компонентов. Интеграция системных компонентов	Знать: современные тенденции развития аппаратного и программного обеспечения, включая платформы для ИИ и больших данных; принципы обеспечения отказоустойчивости систем. Уметь: подбирать оптимальные конфигурации оборудования под конкретные задачи, включая обучение и инференс моделей; обеспечивать совместимость компонентов системы; автоматизировать процессы установки и настройки. Владеть: навыками работы с системами автоматизированного развертывания (Docker, Kubernetes); методами мониторинга работоспособности систем; технологиями резервного копирования и восстановления данных, включая версионирование данных и моделей.

4. Структура и содержание дисциплины Обеспечение безопасности при разработке программного обеспечения

Модуль	Вид учебной работы	Наименование тем занятий	Семестр	Объём, ч	Баллы	Формы текущего контроля
1. Основы безопасности ПО и управления данными	лекция	Тема 1. Введение в безопасность при разработке программного обеспечения. Особенности безопасности систем с применением ИИ.			–	–
	самост.	Изучение лекционного материала, подготовка к выполнению практических работ			–	–
	лекция	Тема 1.1. Методы оптимизации управления жизненным циклом распределённых данных с учётом информационной безопасности. Приватность данных и дифференциальная приватность.			–	–
	самост.	Изучение лекционного материала, подготовка к практическим работам			–	–
2. Безопасность в сетевых технологиях и системах ИИ	лекция	Тема 2. Принципы информационной безопасности. Проектирование безопасности для систем сбора и обработки больших данных.			–	–
	практ. самост.	Практическая работа №1. Разработка веб-приложения с функцией редактирования заметок и обеспечение базовой безопасности (Flask, защита от CSRF,		–		Отчёт по практической работе
	практ. самост.	Практическая работа №2. Статический анализ кода (SAST) с помощью Bandit, устранение уязвимостей.		–		Отчёт по практической работе

Модуль	Вид учебной работы	Наименование тем занятий	Семестр	Объём, ч	Баллы	Формы текущего контроля
	практ. са-мост.	Практическая работа №3. Динамическое тестирование (DAST) с помощью OWASP ZAP, защита от		–		Отчёт по практической работе
	практ. са-мост.	Практическая работа №4. Защита от SQL-инъекций, тестирование с помощью SQLMap.		–		Отчёт по практической работе
	практ. са-мост.	Практическая работа №5. Обеспечение безопасности базы данных PostgreSQL (настройка pg_hba.conf, SSL, прав доступа).		–		Отчёт по практической работе
	практ. са-мост.	Практическая работа №6. Разработка плана безопасности и DevSecOps-интеграция (CI/CD с Bandit).		–		Отчёт по практической работе
	практ. са-мост.	Практическая работа №7. Мониторинг безопасности и реагирование на инциденты (SIEM Lite).		–		Отчёт по практической работе
	самост.	Подготовка к выполнению практических работ, изучение инструментов (OWASP ZAP, Bandit, SQLMap,			–	–
	лекция	Тема 3. Технология осуществления оптимизации управления жизненным циклом данных. Безопасность конвейеров обработки данных.			–	–
	самост.	Изучение лекционного материала			–	–

Модуль	Вид учебной работы	Наименование тем занятий	Семестр	Объём, ч	Баллы	Формы текущего контроля
	лекция	Тема 4. Инструменты обеспечения безопасности на этапе разработки. Инструменты для анализа безопасности данных и моделей (TensorFlow Privacy, Adversarial Robustness Toolbox и др.).			–	–
	самост.	Изучение лекционного материала			–	–
3. Разработка прикладных задач с учётом требований безопасности (в т.ч. для ИИ)	лекция	Тема 5. Оптимизация управления жизненным циклом данных. Обеспечение безопасности при использовании внешних наборов данных и моделей.		–	–	–
	самост.	Изучение лекционного материала			–	–
	лекция	Тема 6. Угрозы безопасности и методы их предотвращения. Атаки на модели машинного обучения (отравление данных, инверсия, уклонение).		–	–	–
	самост.	Изучение лекционного материала			–	–
	лекция	Тема 7. Реагирование на угрозы безопасности. Создание систем мониторинга и защиты для AI-сервисов.		–	–	–
	самост.	Изучение лекционного материала, подготовка к ККР и экзамену			–	–
	ККР	Комплексная контрольная работа				Оценка уровня знаний и умений

Модуль	Вид учебной работы	Наименование тем занятий	Семестр	Объём, ч	Баллы	Формы текущего контроля
	экзамен	Экзамен (устно или письменно по билетам)		–		Оценка уровня знаний и умений
Итого						

Текущий рейтинг (сумма баллов за практические работы) + балл за ККР + балл за экзамен. Максимальный итоговый балл – 100.

Все практические работы выполняются студентом **самостоятельно** в рамках часов, отведённых на самостоятельную работу. Отчёты по практическим работам сдаются и защищаются в установленные сроки (в период семестра). ККР выполняется и сдаётся на проверку. Экзамен проводится по окончании семестра.

5. Образовательные технологии

В процессе изучения дисциплины используется технология традиционного обучения (лекции, практические работы, самостоятельная работа студента)

Методические указания по освоению дисциплины

Для успешного освоения дисциплины необходимы посещение студентами лекционных и практических занятий, самостоятельная работа студентов с лекционным материалом и учебной литературой.

Изучение дисциплины требует систематического и последовательного накопления знаний, следовательно, пропуски отдельных тем не позволяют глубоко освоить предмет.

В ходе лекционных занятий полезно задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

Студент может дополнить список предложенной литературы современными источниками, не представленными в списке, и в дальнейшем использовать собственные подготовленные учебные материалы при написании курсовых и выпускных квалификационных работ.

Студентам следует

- при подготовке к практическим занятиям обязательно использовать не только лекции, учебную литературу, но и другие источники;
- в начале занятий задавать преподавателю вопросы по материалу, вызвавшему затруднения в его понимании и использовании при решении задач, предложенных для самостоятельного решения;
- на занятиях доводить каждую задачу до окончательного ответа, демонстрировать понимание проведенных расчетов (рассуждений), в случае затруднений обращаться к преподавателю.

Для того чтобы практические занятия приносили максимальную пользу, необходимо помнить, что решение задач проводится по рассмотренному на лекциях материалу и связано, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться студентом на практических занятиях как в результате обсуждения и анализа лекционного материала, так и в процессе решения задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (что очень важно) для активной проработки лекционного материала.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений (рассуждений, преобразований) составить краткий план решения проблемы (задачи). Решение задач следует излагать подробно, вычисления (рассуждения, преобразования) располагать в строгом порядке. Решение при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Полезно (если это возможно) решать задачу несколькими способами и сравнивать полученные результаты. Решение задач определённого типа нужно продолжать до приобретения твердых навыков в их решении.

Самостоятельная работа студентов по предмету организуется в следующих формах:

- 1) самостоятельное изучение основного теоретического материала, ознакомление с дополнительной литературой, Интернет-ресурсами;
- 2) решение профессиональных задач из реальной предметной области.

В качестве учебно-методического обеспечения самостоятельной работы используется основная и дополнительная литература по предмету, Интернет-ресурсы, материал лекций, указания, выданные преподавателем при проведении практических работ.

Подготовка к зачету способствует закреплению, углублению и систематизации знаний, получаемых в процессе обучения. Готовясь к зачету, студент ликвидирует имеющиеся пробелы в знаниях, упорядочивает свои знания. На зачете студент демонстрирует как теоретические знания, приобретённые в процессе обучения по данной учебной дисциплине, так и навыки их практического использования при решении задач.

Необходимо ориентировать студентов на систематическую подготовку к занятиям в течение семестра, поскольку это позволит освоить основы изучаемой дисциплины, а время экзаменационной сессии можно будет использовать для систематизации уже имеющихся знаний.

7. Оценочные средства

7.1. Паспорт оценочных средств

Семестр	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
	ОПК-3; ОПК-5	Тестовы вопросы Вопросы к экзамену. Отчеты по практическим работам 1-7

7.2. Типовые задания или иные материалы, необходимые для текущего контроля

Отчеты по практическим работам

(наименование оценочного средства)

Типовые примеры заданий

Практическая работа №1. Разработка веб-приложения с функцией редактирования заметок и обеспечение базовой безопасности

Цель работы: освоить методы безопасной разработки веб-приложений на примере создания функционала редактирования заметок, включая применение безопасных практик кодирования, предотвращение типовых уязвимостей (SQL-инъекции, CSRF, IDOR) и использование современных инструментов анализа безопасности.

1. Задание:

Разработать веб-приложение на Python с использованием Flask и SQLAlchemy, реализующее CRUD-операции для управления заметками.

2. Обеспечить:

- использование ORM (запрет на конкатенацию SQL-строк);
- защиту от CSRF через Flask-WTF;
- имитацию защиты от IDOR с проверкой прав на основе сессии;
- валидацию и санитизацию ввода;
- наличие минимум двух HTML-шаблонов (главная страница и форма редактирования).

Форма отчета:

Исходный код приложения (структура папок, файлы app.py, шаблоны). Скриншоты работающего приложения (главная страница, форма редактирования). Краткий отчет (1 стр.) с

описанием архитектуры, перечнем применённых мер безопасности, пояснением защиты от CSRF и IDOR.

Практическая работа №2. Статический анализ кода и устранение уязвимостей

Цель работы: научиться использовать инструменты статического анализа кода (SAST) для выявления потенциальных уязвимостей на этапе разработки и устранять найденные проблемы.

Задание:

На основе приложения, созданного в ПР №1:

1. Провести статический анализ с помощью Bandit.
2. Проанализировать отчёт, выявить уязвимости (например, hardcoded secrets, debug mode).
3. Устранить уязвимости:
 - вынести секретные данные в переменные окружения (через python-dotenv);
 - отключить режим отладки для production;
 - исправить прочие найденные проблемы.
4. Повторно выполнить анализ и подтвердить устранение.

Форма отчета:

Команды запуска Bandit, скриншоты отчётов до и после исправлений.

Фрагменты кода до и после исправлений.

Краткое описание каждой исправленной уязвимости.

Практическая работа №3. Динамическое тестирование (DAST) и защита от OWASP

Цель работы: освоить методы динамического тестирования безопасности веб-приложений, научиться защищать приложение от распространённых уязвимостей (отсутствие заголовков безопасности, CSRF).

Задание:

На основе приложения из ПР №2:

1. Провести активное сканирование с помощью OWASP ZAP.
2. Проанализировать отчёт ZAP, выявить основные уязвимости (например, отсутствие CSP, X-Frame-Options, уязвимость к CSRF).
3. Реализовать защиту:
 - добавить middleware для установки HTTP-заголовков безопасности (CSP, HSTS, X-Frame-Options, X-Content-Type-Options);
 - реализовать полноценную защиту от CSRF (Flask-WTF, если не было в ПР №1);
 - скрыть информацию о сервере (запуск через Gunicorn с кастомным server_name).
4. Повторно просканировать и подтвердить устранение уязвимостей.

Форма отчета:

1. Скриншоты настройки и запуска сканирования в ZAP.
2. Отчёты ZAP до и после исправлений.
3. Код реализованных защитных механизмов.
4. Краткое описание каждой исправленной уязвимости.

Практическая работа №4. Защита от SQL-инъекций и тестирование с помощью

Цель работы: углубить понимание механизма SQL-инъекций, научиться защищать приложение с помощью параметризованных запросов и тестировать его устойчивость с помощью специализированных инструментов.

Задание:

Дополнить приложение из ПР №3 функционалом аутентификации (регистрация, логин), используя намеренно уязвимые SQL-запросы с конкатенацией строк (через sqlite3).

1. Провести ручное тестирование SQL-инъекций (обход аутентификации, UNION-атака, time-based) и составить чек-лист.
2. Провести тестирование с помощью SQLMap для уязвимой версии.
3. Переписать уязвимые запросы с использованием параметризованных запросов SQLAlchemy ORM.
4. Повторно протестировать с помощью SQLMap, убедиться в отсутствии уязвимостей.

Форма отчета:

Скриншоты успешных SQL-инъекций (до исправления).

Чек-лист тестирования с payloads.

Команды и результаты SQLMap до и после исправления.

Код уязвимого и безопасного варианта.

Описание принципа работы параметризованных запросов.

Практическая работа №5. Обеспечение безопасности базы данных PostgreSQL

Цель работы: научиться настраивать комплексную безопасность СУБД PostgreSQL на уровне сети, аутентификации, авторизации и шифрования.

Задание:

1. Настроить права доступа через pg_hba.conf: разрешить подключения только с 127.0.0.1 и одного доверенного IP.
2. Реализовать шифрование: сгенерировать SSL-сертификаты через OpenSSL, настроить postgresql.conf для обязательного использования SSL.
3. Создать пользователя приложения с минимальными привилегиями (только SELECT, INSERT), отозвать права у PUBLIC.
4. Настроить брандмауэр ОС для разрешения подключений к порту PostgreSQL только с доверенного IP.

Форма отчета:

Фрагменты конфигурационных файлов (pg_hba.conf, postgresql.conf) с комментариями.

Команды генерации сертификатов и настройки пользователя.

Правила брандмауэра (Windows Firewall / pfctl).

Результаты тестов: успешное подключение с разрешённого IP, блокировка с запрещённого.

Практическая работа №6. Разработка плана безопасности и DevSecOps-интеграция

Цель работы: научиться системно подходить к обеспечению безопасности приложения, разработать комплексный план безопасности и интегрировать проверки в процесс разработки (DevSecOps).

Задание:

1. Разработать план безопасности для приложения из ПР №4 с учётом настроек из ПР №5, включающий:
 - цели и требования безопасности;
 - оценку угроз (на основе ранее проведённых тестов);
 - перечень реализованных мер защиты;
 - план регулярного тестирования (SAST, DAST, пентесты).
2. Реализовать CI/CD pipeline (например, через GitHub Actions), который автоматически:
 - запускает Bandit при каждом пуше;
 - блокирует слияние кода при обнаружении критических уязвимостей.

Форма отчета:

Документ с планом безопасности (структурированный).

Ф

Скриншоты из интерфейса CI/CD, показывающие успешный и неуспешный запуски (с блокировкой).

Л

Практическая работа №7. Мониторинг безопасности и реагирование на инциденты

о

н

Цель работы: научиться настраивать базовую систему мониторинга безопасности (SIEM-подобную) для сбора, анализа и реагирования на события безопасности веб-приложения и базы данных.

у

Задание:

- а 3. Настроить централизованный сбор логов Flask и PostgreSQL в файлы.
- ц 4. Написать Python-скрипт-анализатор, который в реальном времени (или по расписанию) сканирует логи на предмет подозрительных событий.
- и 5. Реализовать автоматическое оповещение (вывод в консоль, запись в файл тревог, отправка email) при обнаружении инцидентов.
- С 6. Сформировать ежедневный отчёт о безопасности (количество запросов, инцидентов, их типы).

/

Форма отчета:

Конфигурации логирования (Flask, PostgreSQL).

Исходный код скрипта-анализатора.

Скриншоты консоли с оповещениями.

Примеры файлов security_alerts.log и daily_security_report.txt.

Краткое описание архитектуры системы мониторинга.

п

Общие требования к оформлению отчётов по практическим работам

и Отчёт выполняется в текстовом редакторе, шрифт Times New Roman, 14 pt, межстрочный интервал 1,5, поля – 2 см.

е Объём отчёта – не менее 5 страниц (без учёта приложений).

р

Все графические материалы (схемы, диаграммы, скриншоты) должны быть подписаны и иметь ссылки в тексте.

Код скриптов и конфигурационные файлы могут быть вынесены в приложения.

Отчёт сдаётся преподавателю в электронном виде в установленный срок.

Распределение баллов по работам:

Практическая работа	Макс. балл
№1. Разработка веб-приложения с базовой безопасностью	6
№2. Статический анализ кода (SAST)	6
№3. Динамическое тестирование (DAST) и защита от OWASP Top 10	6
№4. Защита от SQL-инъекций, тестирование SQLMap	6
№5. Обеспечение безопасности базы данных PostgreSQL	7
№6. Разработка плана безопасности и DevSecOps-интеграция	7
№7. Мониторинг безопасности и реагирование на инциденты	7
Итого	45

Критерии оценки отчётов по практическим работам (применительно к максимальному баллу за работу):

Уровень выполнения	Баллы	Критерии
Высокий	100% от макс. балла (6 или 7)	Работа выполнена в полном объёме с соблюдением необходимой последовательности действий; отчёт оформлен аккуратно, чётко, без ошибок; вывод исчерпывающий и доказательный. При защите отчёта студент ответил на все вопросы по теме, хорошо ориентируется в материале, умеет определить взаимосвязь факторов и их влияние на конечную цель, умеет графически отобразить важнейшие функциональные зависимости.
Хороший	70–85% от макс. балла (4–5 балла для работ с макс. 6; 5–6 для работ с макс. 7)	Работа выполнена в полном объёме с соблюдением необходимой последовательности действий; студент без ошибок выполнил отчёт, вывод исчерпывающий. При защите отчёта

Уровень выполнения	Баллы	Критерии
		хорошо разбирается в материале, но не уверен и неполно отвечает на вопросы. Способность к обобщению причинно-следственных связей важнейших факторов выражена недостаточно.
Удовлетворительный	50–65% от макс. балла (3–4 балла для работ с макс. 6; 3,5–4,5 для работ с макс. 7)	Работа выполнена не полностью, но объём выполненной части таков, что позволяет получить правильные результаты и выводы; выполнена с несущественными замечаниями. Вывод по работе не раскрывает сути работы. Владение понятийным аппаратом темы недостаточно.
Неудовлетворительный	менее 50% от макс. балла (1–2 балла для работ с макс. 6; 1–3 балла для работ с макс. 7)	Студент выполнил работу не полностью, или объём выполненной части работы не позволяет сделать правильных выводов. В ответах на вопросы есть грубые ошибки. Нет знания принципиальных теоретических положений темы.

Оценочные средства для промежуточной аттестации по итогам освоения дисциплины

Типовые примеры тестовых заданий

1. Какая из перечисленных уязвимостей относится к категории OWASP Top 10 «Нарушение контроля доступа» (Broken Access Control)?

- а) SQL-инъекция
- б) IDOR (Insecure Direct Object References)
- в) XSS
- г) CSRF

2. Какой инструмент используется для статического анализа безопасности кода (SAST) на языке Python?

- а) OWASP ZAP
- б) Bandit
- в) SQLMap
- г) Wireshark

3. Что такое CSRF-атака?

- а) внедрение вредоносного скрипта в веб-страницу
- б) выполнение несанкционированных действий от имени аутентифицированного пользователя
- в) перебор паролей
- г) перехват сетевого трафика

4. Какой метод защиты наиболее эффективен против SQL-инъекций?

- а) экранирование кавычек вручную
- б) использование параметризованных запросов / ORM
- в) скрывание ошибок БД
- г) ограничение длины ввода

5. Что означает аббревиатура DAST?

- а) Dynamic Application Security Testing
- б) Data Analysis Security Tool

в) Distributed Application Security Testing

г) Design Analysis and Security Testing

6. Какой HTTP-заголовок запрещает загрузку страницы во фрейме (защита от clickjacking)?

а) Content-Security-Policy

б) Strict-Transport-Security

в) **X-Frame-Options**

г) X-Content-Type-Options

7. Какая атака на модели машинного обучения заключается в добавлении вредоносных данных в обучающую выборку?

а) атака уклонения (evasion)

б) инверсия модели (model inversion)

в) **отравление данных (data poisoning)**

г) атака с подбором выходных данных (membership inference)

8. Для чего используется инструмент SQLMap?

а) для статического анализа кода

б) **для автоматического обнаружения и эксплуатации SQL-инъекций**

в) для динамического тестирования веб-приложений

г) для мониторинга сетевого трафика

9. Что такое дифференциальная приватность (differential privacy)?

а) метод шифрования данных

б) **метод защиты приватности при анализе данных, добавляющий шум в результаты запросов**

в) способ аутентификации

г) протокол безопасной передачи данных

10. Какой инструмент используется для динамического тестирования безопасности веб-приложений (DAST)?

а) Bandit

б) **OWASP ZAP**

в) Git

г) Docker

11. Что из перечисленного является лучшей практикой хранения секретов (паролей, ключей API) в коде?

а) хранить в комментариях

б) хранить в отдельном файле, добавленном в репозиторий

в) **хранить в переменных окружения или в менеджере секретов**

г) хранить в имени переменной

12. Какая атака на модели машинного обучения направлена на получение информации об обучающих данных через анализ выходов модели?

а) отравление данных

б) атака уклонения

в) **инверсия модели (model inversion)**

г) атака «отказ в обслуживании»

13. Какой заголовок HTTP обеспечивает защиту от MIME-сниффинга?

а) X-Frame-Options

б) Strict-Transport-Security

в) **X-Content-Type-Options: nosniff**

г) Content-Security-Policy

14. Что такое DevSecOps?

а) замена ручного тестирования автоматизированным

б) **интеграция практик безопасности на всех этапах DevOps-цикла**

- в) выделение безопасности в отдельную команду
- г) отказ от статического анализа кода

15. Какой метод защиты от CSRF-атак является наиболее распространённым в веб-приложениях?

- а) проверка Referer-заголовка
- б) **использование CSRF-токенов**
- в) ограничение по IP-адресу
- г) капча

16. Что из перечисленного НЕ является уязвимостью OWASP Top 10?

- а) SQL-инъекция
- б) межсайтовый скриптинг (XSS)
- в) **использование ORM**
- г) недостаточное логирование и мониторинг

17. Какая атака заключается в подборе входных данных, заставляющих модель ИИ ошибаться (например, добавление невидимого шума к изображению)?

- а) отравление данных
- б) **атака уклонения (evasion attack)**
- в) инверсия модели
- г) атака перебора

18. Какой из перечисленных инструментов используется для мониторинга и анализа сетевого трафика?

- а) Bandit
- б) OWASP ZAP
- в) **Wireshark**
- г) SQLMap

19. Что такое HSTS (HTTP Strict Transport Security)?

- а) **механизм, принудительно использующий HTTPS для всех соединений с сайтом**
- б) метод защиты от XSS
- в) протокол безопасной аутентификации
- г) способ шифрования cookies

20. Какой инструмент автоматизации может использоваться для встраивания проверок безопасности (SAST) в CI/CD-пайплайн?

- а) OWASP ZAP
- б) **GitHub Actions (совместно с Bandit)**
- в) SQLMap
- г) Wireshark

Критерии оценки за пройденный тест:

- 100 баллов выставляется обучающемуся, если он ответил правильно на все вопросы случайной выборки 30 тестовых заданий;
- 0-99 баллов выставляется обучающемуся в зависимости от количества верных ответов на вопросы случайной выборки 30 тестовых заданий.

7.3.2. Пример задания ККР

Комплексная контрольная работа по дисциплине «Обеспечение безопасности при разработке программного обеспечения»

Цель работы: систематизировать и закрепить знания, умения и навыки в области обеспечения безопасности при разработке программного обеспечения, полученные в ходе выполнения

практических работ, а также продемонстрировать способность самостоятельно анализировать уязвимости и разрабатывать меры защиты для веб-приложения.

Задачи:

1. Проанализировать предоставленный фрагмент кода веб-приложения на наличие типовых уязвимостей (SQL-инъекции, отсутствие CSRF-защиты, недостаточная валидация ввода и др.).
2. Предложить и обосновать комплекс мер по устранению выявленных уязвимостей с учётом современных практик безопасной разработки (использование ORM/параметризованных запросов, CSRF-токены, валидация и санитизация данных).
3. Разработать план безопасности для приложения, включающий настройку защитных HTTP-заголовков (CSP, HSTS, X-Frame-Options), статический анализ кода (SAST) и динамическое тестирование (DAST).
4. Составить конфигурацию CI/CD-пайплайна (например, GitHub Actions), автоматизирующего запуск SAST-инструмента (Bandit) и блокирующего слияние кода при обнаружении критических уязвимостей.
5. Оформить отчёт, содержащий описание выявленных проблем, предложенные исправления (с фрагментами кода до/после), план безопасности, конфигурацию CI/CD и выводы.

Ход выполнения

1. **Анализ уязвимостей.** Внимательно изучить предоставленный преподавателем фрагмент кода веб-приложения (Flask + SQLite/PostgreSQL). Выявить не менее трёх различных типов уязвимостей (например, SQL-инъекция из-за конкатенации строк, отсутствие CSRF-защиты, недостаточная валидация пользовательского ввода, хранение секретов в коде и т.п.).

2. **Разработка мер защиты.** Для каждой выявленной уязвимости предложить способ устранения:

- переход на параметризованные запросы / ORM;
- внедрение CSRF-защиты (Flask-WTF);
- вынос секретов в переменные окружения;
- добавление валидации и санитизации ввода;
- настройку защитных HTTP-заголовков (CSP, HSTS, X-Frame-Options, X-Content-Type-Options).

3. **План безопасности.** Составить документ, включающий:

- перечень реализованных мер защиты;
- описание процедур регулярного тестирования (статический анализ кода — Bandit, динамическое тестирование — OWASP ZAP);
- план реагирования на инциденты (мониторинг логов, оповещение).

4. **CI/CD-интеграция.** Написать конфигурационный файл для GitHub Actions (или другого CI/CD-инструмента), который:

- при каждом push / pull request запускает Bandit для статического анализа;
- в случае обнаружения уязвимостей с высоким или критическим уровнем серьёзности блокирует слияние (merge) кода.

5. **Оформление отчёта.** Подготовить отчёт в текстовом редакторе (шрифт Times New Roman, 14 pt, межстрочный интервал 1,5, поля 2 см). Отчёт должен содержать:

- титульный лист;
- исходный анализируемый фрагмент кода (или ссылку на него);
- таблицу выявленных уязвимостей с указанием типа, местоположения и потенциальных последствий;
- фрагменты кода до и после исправления для каждой уязвимости;
- план безопасности (структурированный документ);
- конфигурацию CI/CD-пайплайна с комментариями;
- выводы по работе (достигнутые результаты, какие меры оказались наиболее эффективными).

Отчёт по работе представить в электронном виде в формате .docx.

Процедура оценивания

Оценка выполненной работы проводится по критериям:

1. **Наличие всей существенной информации по работе** – полнота анализа уязвимостей, полнота предложенных мер защиты.
2. **Точность и полнота предоставляемых сведений** – корректность выявления уязвимостей, правильность выбора способов их устранения.
3. **Непротиворечивость приводимой информации** – логическая связь между выявленными проблемами и предложенными решениями.
4. **Правильность интерпретаций и выводов, которые сделаны по результатам работы** – обоснованность выводов, понимание последствий каждой уязвимости.
5. **Степень достижения студентом поставленной цели** – полнота выполнения всех пунктов задания.
6. **Обоснованность применяемого решения** – аргументация выбора конкретных инструментов и методов защиты.
7. **Грамотность (содержательная) используемых формулировок** – корректное использование профессиональной терминологии.

Критерии оценки курсовой контрольной работы (максимальный балл – 15)

Уровень	Баллы	Критерии
Высокий	13–15	Работа выполнена в полном объёме с соблюдением необходимой последовательности действий; выявлены все существенные уязвимости (не менее трёх), предложены корректные и обоснованные способы их устранения; план безопасности структурирован, содержит все необходимые разделы; CI/CD-конфигурация работоспособна и соответствует заданию; отчёт оформлен аккуратно, чётко, без ошибок; выводы исчерпывающие и доказательные. При защите отчёта студент ответил на все вопросы по теме, хорошо ориентируется в материале, умеет определить взаимосвязь факторов и их влияние на конечную цель.

Уровень	Баллы	Критерии
Хороший	10–12	Работа выполнена в полном объёме с соблюдением необходимой последовательности действий; студент выявил основные уязвимости и предложил способы их устранения, но допущены незначительные неточности (например, не полностью учтены все возможные векторы атак, или в CI/CD-конфигурации отсутствует блокировка при критических уязвимостях). Отчёт выполнен без существенных ошибок, выводы в целом верны. При защите отчёта студент хорошо разбирается в материале, но не уверен и неполно отвечает на вопросы. Способность к обобщению причинно-следственных связей выражена недостаточно.
Удовлетворительный	7–9	Работа выполнена не полностью (например, выявлено менее трёх уязвимостей, или предложены не все необходимые меры защиты, или отсутствует один из разделов плана безопасности), но объём выполненной части позволяет получить правильные результаты и выводы по отдельным аспектам. Отчёт выполнен с несущественными замечаниями. Выводы не полностью раскрывают суть работы. Владение понятийным аппаратом темы недостаточно.
Неудовлетворительный	0–6	Студент выполнил работу не полностью, или объём выполненной части не позволяет сделать правильных выводов. Уязвимости выявлены неверно или не выявлены вовсе. Предложенные меры защиты некорректны или отсутствуют. В ответах на вопросы есть грубые ошибки. Нет знания принципиальных теоретических положений темы.

7.3.3. Вопросы к промежуточной аттестации (экзамену)

1. Что такое обеспечение безопасности при разработке программного обеспечения? Почему это важно?
2. Какие основные угрозы безопасности существуют при разработке ПО?
3. Какие этапы разработки ПО требуют особого внимания к безопасности?
4. Что такое уязвимости в программном обеспечении? Приведите примеры.
5. Какие методы и инструменты используются для выявления уязвимостей в ПО?
6. Что такое DevSecOps? Как этот подход влияет на безопасность ПО?
7. Какие стандарты безопасности существуют для разработки ПО? Приведите примеры.
8. Что такое «безопасный код»? Какие практики помогают писать безопасный код?
9. Как обеспечивается безопасность в процессе непрерывной интеграции и доставки?
10. Что такое «безопасное программирование»? Какие принципы лежат в его основе?
11. Какие протоколы безопасности используются для защиты данных при передаче по сети?
12. Что такое SSL/TLS? Как они работают?
13. Какие функции выполняют межсетевые экраны в обеспечении безопасности?
14. Какие типы межсетевых экранов существуют? Приведите примеры.
15. Как настраиваются правила межсетевых экранов для защиты сети?
16. Что такое VPN? Как он обеспечивает безопасность передачи данных?
17. Какие инструменты используются для мониторинга сетевого трафика и выявления аномалий?
18. Что такое IDS и IPS? В чем их различия?
19. Как протоколы безопасности и межсетевые экраны взаимодействуют для защиты сети?

20. Какие меры безопасности следует предпринять при настройке удалённого доступа к сети?
21. Какие угрозы безопасности существуют для баз данных?
22. Какие меры безопасности следует предпринять при разработке баз данных?
23. Что такое SQL-инъекция? Как её предотвратить?
24. Какие методы аутентификации и авторизации используются для защиты баз данных?
25. Что такое шифрование данных в базе данных? Какие алгоритмы шифрования используются?
26. Как обеспечивается целостность данных в базе данных?
27. Какие инструменты используются для мониторинга безопасности баз данных?
28. Что такое «безопасная разработка баз данных»? Какие практики это включает?
29. Как обеспечивается резервное копирование и восстановление баз данных?
30. Какие стандарты безопасности существуют для баз данных? Приведите примеры.
31. Что такое сканирование сети? Какие цели оно преследует?
32. Какие инструменты используются для сканирования сети? Приведите примеры.
33. Как проводится сканирование сети на уязвимости?
34. Что такое портовое сканирование? Какие порты считаются уязвимыми?
35. Как сканирование сети помогает выявить потенциальные угрозы?
36. Какие меры безопасности следует предпринять после проведения сканирования сети?
37. Что такое Nmap? Какие функции он выполняет?
38. Как проводится сканирование сети на наличие вредоносного ПО?
39. Какие ограничения существуют при сканировании сети?
40. Как сканирование сети влияет на производительность сети?
41. Какие принципы управления доступом реализуются в СУБД?
42. Какие методы шифрования данных применяются для защиты информации?
43. Какие подходы используются для резервного копирования и восстановления данных?
44. Как осуществляется аудит и мониторинг активности в базах данных?
45. Какие механизмы контроля прав доступа применяются для минимизации рисков?
46. Что такое криптографическая защита данных и как она реализуется?
47. Какие типы атак на веб-приложения наиболее распространены?
48. Как реализуется безопасность в микросервисной архитектуре?
49. Какие методы защиты от XSS-атак применяются в веб-разработке?
50. Как организуется управление инцидентами безопасности в ИТ-системах?
51. Какие принципы безопасности лежат в основе архитектуры приложений?
52. Как обеспечивается безопасность контейнеров в процессе разработки ПО?
53. Какие методы защиты от CSRF-атак применяются в веб-разработке?
54. Как реализуется безопасная аутентификация в распределённых системах?
55. Какие подходы используются для управления секретами в DevOps-практиках?
56. Как обеспечивается безопасность API в современных приложениях?
57. Какие методы защищают от атак типа «человек посередине» (Man-in-the-Middle)?
58. Что такое SAST и DAST? В чем их различия и как они применяются?
59. Какие принципы безопасности следует учитывать при проектировании облачных приложений?
60. Как реализуется безопасная работа с открытым исходным кодом в коммерческих проектах?
61. Какие особенности безопасности возникают при работе с большими данными (распределённое хранение, потоковая обработка)?
62. Что такое дифференциальная приватность и как она применяется для защиты данных в системах машинного обучения?
63. Какие существуют типы атак на модели машинного обучения (отравление данных, атаки с подбором выходных данных, инверсия моделей)? Приведите примеры.

64. Как обеспечить безопасность конвейера данных (data pipeline) при обучении и инференсе моделей?
65. Какие инструменты используются для оценки безопасности моделей машинного обучения (например, Adversarial Robustness Toolbox, TensorFlow Privacy)?
66. Что такое приватность при обучении с федеративным подходом (federated learning) и каковы риски?
67. Как обеспечить безопасное хранение и передачу наборов данных, содержащих персональные данные?
68. Какие методы криптографической защиты данных (гомоморфное шифрование, шифрование с сохранением порядка) актуальны для обработки больших данных?
69. Как организовать мониторинг и аудит доступа к данным в системах ИИ?
70. Какие требования безопасности предъявляются к API, через которые осуществляется взаимодействие с AI-сервисами?
71. Какие этапы жизненного цикла данных требуют оптимизации с учётом безопасности?
72. Какие методы оптимизации управления распределёнными данными (сегментирование, репликация, кэширование) влияют на безопасность?
73. Как настроить безопасность в распределённых системах обработки данных (Apache)?
74. Какие политики шифрования и управления ключами применяются в распределённых хранилищах данных?

Практические кейсы:

Кейс 1. В веб-приложении обнаружена SQL-инъекция. Разработайте план немедленного реагирования и долгосрочного устранения уязвимости. Какие инструменты используете для проверки?

Кейс 2. При динамическом тестировании OWASP ZAP выявлено отсутствие HTTP-заголовков безопасности (CSP, HSTS, X-Frame-Options). Какие заголовки необходимо добавить и как это повлияет на безопасность приложения?

Кейс 3. Модель машинного обучения, развёрнутая в промышленной среде, стала выдавать неожиданные результаты при подаче специально сформированных запросов. Предположите причину (атака уклонения) и предложите меры защиты.

Кейс 4. Разрабатывается система, которая собирает данные с IoT-устройств, передаёт их в облако для обучения моделей и предоставляет результаты через веб-сервис. Укажите ключевые точки контроля безопасности и предложите меры защиты на каждом этапе.

Кейс 5. В CI/CD pipeline (GitHub Actions) необходимо интегрировать проверки безопасности: статический анализ кода, сканирование зависимостей, проверку секретов. Опишите, как это можно реализовать, и какие инструменты использовать.

Кейс 6. Проект использует открытые наборы данных для обучения модели. Как проверить их на наличие вредоносных вкраплений (отравление данных)? Предложите процедуру верификации.

Кейс 7. В корпоративной сети обнаружены несанкционированные SSH-подключения к серверу с обучающими данными. Определите возможные источники угрозы и разработайте план защиты.

Кейс 8. Для облачного хранилища, содержащего персональные данные, требуется настроить шифрование и управление ключами. Какие механизмы вы предложите (шифрование на стороне клиента, управление ключами в KMS, ротация ключей)?

Кейс 9. При развёртывании модели через API были зафиксированы аномально высокие запросы от одного IP-адреса. Какие меры необходимо принять для предотвращения атаки перебора (brute-force) или DoS?

Кейс 10. Разработайте план обеспечения безопасности для системы, использующей федеративное обучение на мобильных устройствах. Укажите меры защиты на стороне клиента и сервера.

Критерии оценивания экзамена при прохождении итогового тестирования по БРС

Семестр	Форма проведения промежуточной аттестации	Критерии и нормы оценки	
	Экзамен (по накопительному рейтингу)	«отлично»	рейтинговый балл 85-100
		«хорошо»	рейтинговый балл 70-84
		«удовлетворительно»	рейтинговый балл 55-69
		«неудовлетворительно»	рейтинговый балл 0-54

8. Учебно-методическое и информационное обеспечение дисциплины

Обязательная литература

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно-методическое пособие, практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
	Баранова Е. К.	Криптографические методы защиты информации : лаб. практикум : учеб. пособие / Е. К. Баранова, А. В. Бабаш. - Москва : КноРус, 2015. - 196 с. : ил. + CD. - (Бакалавриат). - Библиогр. в конце гл. - ISBN 978-5-406-03802-4 : 250-00. - ISBN 205-00.	Учебное пособие		
	Фороузан Б. А.	Криптография и безопасность сетей [Электронный ресурс] : учеб. пособие / Б. А. Фороузан ; пер. с англ. под ред. А. Н. Берлина. - Москва : ИНТУИТ, 2017 ; Саратов : Вузовское образование, 2017. - 782 с. : ил. - (Основы информационных технологий).	Учебное пособие		ЭБС «IPRbooks»
	Хорев П. Б.	Программно-аппаратная защита информации [Электронный ресурс] : учеб. пособие / П. Б. Хорев. - 2-е изд., испр. и доп. - Москва : Форум : ИНФРА-М, 2015. - 352 с. - (Высшее образование). - ISBN 978-5-00091-004-7.	Учебное пособие		ЭБС

Дополнительная литература

№ п/п	Авторы, со- ставители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно-методиче- ское пособие, практикум, др.)	Год из- дания	Количество в научной биб- лиотеке / Наименова- ние ЭБС
	Кукина Е. Г.	Введение в криптографию [Электронный ресурс] : сборник задач и упражнений / Е. Г. Кукина, В. А. Романьков. - Омск : ОмГУ, 2013. - 9	Учебное пособие		ЭБС
	Никифоров С. Н.	Защита информации [Электронный ресурс] : учеб. пособие / С. Н. Никифоров. - Санкт-Петербург : СПбГАСУ, 2015. - 383 с. : ил. - ISBN	Учебное пособие		ЭБС
	Спицын В. Г.	Информационная безопасность вычислительной техники [Электронный ресурс] : учебное пособие / В. Г. Спицын. - Томск : Эль Контент, 2011. - 148 с. - ISBN 978-5-4332-0020-3.	Учебное пособие		ЭБС
	Федин Ф. О.	Информационная безопасность [Электронный ресурс] : учебное пособие / Ф. О. Федин, В. П. Офицеров, Ф. Ф. Федин ; [под ред. В. А. Дикарева]. - Москва : МГПУ, 2011. - 260 с.	Учебное пособие		ЭБС

Перечень профессиональных баз данных и информационных справочных систем

№ пп	Наименование	Ссылка
	Springer Nature (Полнотекстовая коллекция журналов)	
	Springer eBooks (Полнотекстовая коллекция электронных книг издательства Springer)	
	«Кодекс»	
	Техэксперт	
	Федеральная служба по техническому и экспортному контролю	
	Kaggle (датасеты с метками безопасности)	Kaggle датасеты: полное руководство по поиску и использованию для анализа данных - DataLopata

Перечень программного обеспечения

№ п/п	Наименование ПО	Количество лицензий	Реквизиты договора (дата, номер, срок действия)
	Visual Studio Code (VS Code)	неограниченный	Бесплатное ПО, лицензия
		неограниченный	Бесплатное ПО, лицензия
		неограниченный	Бесплатное ПО, лицензия
		неограниченный	Бесплатное ПО, лицензия
		неограниченный	Бесплатное ПО, лицензия
		неограниченный	Бесплатное ПО, лицензия
	OWASP ZAP (Zed Attack Proxy)	неограниченный	Бесплатное ПО, лицензия
		неограниченный	Бесплатное ПО, лицензия

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

№ п/п	Наименование оборудованных учебных кабинетов, лабораторий, мастерских и др. объектов для проведения практических и лабораторных занятий	Перечень основного оборудования
	Компьютерный класс. Помещение для самостоятельной работы. Учебная аудитория для проведения занятий семинарского типа. Учебная аудитория для курсового проектирования (выполнения курсовых работ).	Столы ученические, стулья ученические, ПК с выходом в сеть Интернет

	Учебная аудитория для проведения групповых и индивидуальных консультаций. Учебная аудитория для проведения занятий текущего контроля и промежуточной аттестации. (Г-401)	
	Компьютерный класс. Учебная аудитория для проведения занятий лекционного типа. Учебная аудитория для проведения занятий семинарского типа. Учебная аудитория для проведения лабораторных работ. Учебная аудитория для курсового проектирования (выполнения курсовых работ). Учебная аудитория для проведения групповых и индивидуальных консультаций. Учебная аудитория для проведения занятий текущего контроля и промежуточной аттестации. (УЛК-408)	Компьютер (монитор 17", системный блок Intel (R) Gb / 80 Gb), маршрутизатор 2801 Router, коммутатор Catalyst, экран/интерактивная доска Smart Board ТВ, проектор Acer P1303W., стол преподавательский, стол ученический, стол компьютерный, стул, доска аудиторная (маркерная).
	Учебная аудитория для проведения занятий лекционного типа. Учебная аудитория для проведения занятий семинарского типа. Учебная аудитория для курсового проектирования (выполнения курсовых работ). Учебная аудитория для проведения групповых и индивидуальных консультаций. Учебная аудитория для проведения занятий текущего контроля и промежуточной аттестации. (УЛК-418)	Стол ученический двухместный (моноблок), доска аудиторная 3-х секционная (меловая), стол преподавательский, стул, проектор Acer