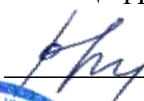


Федеральное государственное образовательное бюджетное учреждение
высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)

ЦЕНТР КОМПЕТЕНЦИЙ «ЦИФРОВАЯ ЭКОНОМИКА»

УТВЕРЖДАЮ

Директор Центра компетенций
«Цифровая экономика»



Н.Ю. Сурова

«01» августа 2026 г.



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА

«Кибербезопасность. СПО (базовый уровень)»

Учебная нагрузка: 145 академических часа.

МОСКВА 2026

1. Пояснительная записка

Актуальность

Программа базового уровня предназначена для студентов учреждений среднего профессионального образования (СПО), знакомых с базовыми алгоритмическими конструкциями, но не имеющих систематических знаний в области программирования и информационной безопасности. Ее актуальность обусловлена стремительным ростом киберугроз в условиях цифровой трансформации – большинство студентов активно используют цифровые сервисы, но не обладают практическими навыками защиты личных данных, распознавания угроз и безопасной разработки. Программа восполняет этот пробел, формируя у обучающихся культуру кибербезопасности и навыки программирования на Python в контексте реальных задач защиты информации.

Программа имеет выраженную профориентационную ценность: информационная безопасность является одной из наиболее динамично развивающихся сфер ИТ-индустрии, а обучение позволяет студентам погрузиться в практические задачи ИБ-специалиста – от шифрования данных до анализа веб-уязвимостей. Программа помогает обучающимся осознать, как полученные навыки соотносятся с кадровыми потребностями рынка, и служит надёжным фундаментом для выбора будущей профессии и трудоустройства на позиции младшего аналитика ИБ, специалиста SOC или разработчика безопасного кода. Содержание программы ориентировано на формирование общих и профессиональных компетенций в соответствии с требованиями ФГОС СПО, что повышает ее практическую значимость для дальнейшего профессионального развития.

Отличительные особенности программы:

- обучение через призму информационной безопасности – базовый синтаксис Python (переменные, условия, циклы, функции) осваивается на задачах валидации паролей, шифрования файлов, анализа логов и моделирования атак;
- модульная структура, что обеспечивает системность знаний и последовательное усвоение материала;
- проектная деятельность – итогом обучения становится разработка и защита итогового проекта, что позволяет применить все полученные навыки в законченном продукте и сформировать портфолио на GitHub.

Новизна программы:

- использование ИИ-инструментов как объекта контроля – обучающиеся учатся проверять и исправлять код, сгенерированный нейросетями, что развивает критическое мышление, навыки отладки и понимание ограничений ИИ;
- формирование культуры кибербезопасности с первых занятий – особое внимание уделяется триаде информационной безопасности, правовым и этическим рамкам, Доктрине ИБ РФ;
- связь с индустрией – в программу интегрированы кейс-задачи без единственно верного ответа, включая задачу от аккредитованной ИТ-компании, что позволяет студентам погрузиться в реальные бизнес-контексты и актуальные проблемы отрасли, а также формирует первичное представление о профессиональной деятельности в сфере информационной безопасности.

Описание программы

Программа направлена на формирование у обучающихся базовых знаний и практических навыков в области программирования и информационной безопасности.

Программа включает в себя четыре модуля, охватывающих основные аспекты безопасной разработки на Python, криптографической защиты данных, анализа уязвимостей и проектной деятельности для решения прикладных задач кибербезопасности.

Формы реализации – очная без применения дистанционных образовательных технологий (в том числе, с применением средств электронного обучения).

Общий объем программы 145 академических часов.

Первый модуль посвящён основам программирования на Python в контексте информационной безопасности. Обучающиеся знакомятся с базовым синтаксисом языка, изучают работу с переменными, условиями, циклами и функциями, осваивают работу с файловой системой и обработку исключений. Все темы отрабатываются на практических задачах: от валидации паролей и анализа ввода пользователя до написания скриптов для базовой проверки целостности данных. В результате обучающиеся получают знания о стандартных типах данных и управляющих конструкциях Python, умеют работать с файлами и обрабатывать ошибки ввода-вывода, а также владеют навыками написания структурированного, безопасного кода с использованием принципов «защитного программирования».

Второй модуль – прикладная защита данных. В рамках этого модуля обучающиеся изучают фундаментальные различия между кодированием, хешированием и шифрованием, осваивают применение стандартных библиотек Python (hashlib, base64, secrets) для решения практических задач. Студенты учатся генерировать криптостойкие случайные значения, вычислять и сравнивать хеши файлов, а также реализовывать простые алгоритмы симметричного шифрования. Особое внимание уделяется проверке целостности данных и безопасному хранению паролей. В результате обучающиеся понимают принципы криптографической защиты, умеют применять хеширование для верификации файлов, владеют навыками генерации безопасных токенов и паролей, а также осознают границы применимости различных криптографических методов.

Третий модуль посвящён анализу веб-уязвимостей и сетевой безопасности. Обучающиеся знакомятся с триадой информационной безопасности (конфиденциальность, целостность, доступность), изучают типовые веб-уязвимости, включая инъекции и межсайтовый скриптинг, и осваивают методы их нейтрализации. В модуле рассматриваются принципы анализа сетевых логов и проверки базовых HTTP-заголовков безопасности. Уникальной особенностью модуля является работа с кодом, сгенерированным ИИ-инструментами: студенты учатся находить, анализировать и исправлять уязвимости в чужом коде. В результате обучающиеся умеют распознавать базовые веб-угрозы, владеют навыками безопасной обработки пользовательского ввода, имеют навыки написания простых анализаторов логов для выявления подозрительной активности и понимают правовые и этические рамки деятельности специалиста по информационной безопасности, включая Доктрину ИБ РФ.

Четвёртый модуль — проектная деятельность и погружение в профессию. В этом модуле обучающиеся применяют весь спектр полученных знаний для разработки и защиты итогового проекта, что позволяет создать законченный продукт для портфолио на GitHub. Модуль включает решение кейс-задач без единственно верного ответа, в том числе от аккредитованной ИТ-компании, что позволяет студентам погрузиться в реальные бизнес-контексты и актуальные проблемы отрасли. В результате обучающиеся получают опыт завершённой инженерной деятельности, умеют презентовать и защищать свои решения, владеют навыками командной работы и распределения ролей, а также формируют первичное представление о профессиональной деятельности и карьерных треках в сфере информационной безопасности.

Программа предназначена для обучающихся учреждений среднего профессионального образования, знакомых с базовыми алгоритмическими конструкциями, но не имеющих

систематических знаний в области программирования и информационной безопасности, и желающих получить практико-ориентированные навыки в этой востребованной сфере.

Программа обеспечена учебно-методическими ресурсами для качественного освоения содержания учебного материала, что также обеспечивает результативность освоения модулей. Обучающимся обеспечен доступ к современному практико-ориентированному содержанию электронных образовательных и информационных ресурсов. Предусмотрены материально-технические условия – программное обеспечение и оборудование.

Преподавательский состав включает IT-специалистов, имеющих большой педагогический стаж в области преподавания языков программирования и обеспечения информационной безопасности.

Аннотация

«Кибербезопасность. СПО (базовый уровень)» – дополнительная общеобразовательная программа технической направленности, ориентированная на практическое освоение программирования и системных принципов кибербезопасности в контексте профессиональной деятельности. Программа рассчитана на базовый уровень сложности.

Дополнительная общеобразовательная программа предусматривает поэтапное освоение языка Python и ключевых концепций защиты информации с переходом к реализации криптографических алгоритмов, анализу уязвимостей и разработке безопасных приложений.

Программа направлена на формирование у обучающихся по программам среднего профессионального образования практических навыков программирования на Python в контексте задач информационной безопасности, включая шифрование и хеширование данных, анализ веб-уязвимостей, работу с логами, а также разработку и защиту проекта.

Программа предназначена для студентов учреждений среднего профессионального образования, знакомых с базовыми алгоритмическими конструкциями, но не имеющих систематических знаний в области программирования и информационной безопасности, и желающих получить практико-ориентированные навыки в этой востребованной сфере.

Программа формирует следующие навыки:

- программирование на Python и работа со стандартными библиотеками для задач информационной безопасности;
- применение алгоритмов шифрования и хеширования для защиты данных, включая генерацию криптостойких случайных значений;
- разработка простых анализаторов логов для выявления подозрительной активности;
- распознавание, анализ и устранение типовых веб-уязвимостей, в том числе в коде, сгенерированном ИИ-инструментами;
- безопасная работа с файловой системой и пользовательскими данными, отладка и тестирование программного кода.

Программа способствует формированию целостного понимания информационной безопасности, её этических и правовых аспектов. Обучающиеся научатся использовать цифровые технологии осознанно и ответственно, критически оценивая риски, соблюдая культуру цифровой гигиены и понимая правовую ответственность специалиста по информационной безопасности.

Входные требования к обучающимся: необходимо базовое знакомство с алгоритмическими конструкциями (ветвления, циклы) и начальные навыки работы с компьютером. Специальной подготовки в области программирования и информационной безопасности не требуется.

Программа базового уровня сочетает глубокую техническую подготовку с развитием командных навыков и правовой грамотности. Содержание и материалы программы разработаны с учётом возрастных особенностей целевой аудитории и обеспечивают

последовательное усвоение материала от базового синтаксиса Python до разработки и защиты итогового проекта. Предусмотрена система повторения: каждый модуль опирается на навыки предыдущего, а в четвёртом модуле обучающиеся интегрируют все полученные знания в итоговом проекте. Программа построена на дифференцированном подходе и предусматривает практические задания двух уровней сложности.

Программа ориентирована на вклад в развитие дальнейшей образовательной траектории в сфере ИТ за счёт включения элементов содержания, соответствующих общим и профессиональным компетенциям ФГОС СПО по специальности 09.02.07 «Информационные системы и программирование», а также практических заданий, направленных на их формирование. Начиная со второго модуля, программа предусматривает вариативность образовательной траектории, которая выражается в предоставлении обучающимся выбора контекста практических заданий.

В программу интегрированы кейс-задачи, не имеющие единственно верного ответа, включая задачу, представленную аккредитованной ИТ-организацией. В практических заданиях предусмотрены основы коллективной работы: парное программирование, командный аудит кода с распределением ролей и критериями оценки индивидуального вклада каждого участника.

Программа включает работу с использованием искусственного интеллекта (ИИ): нейросети используются как инструмент для поиска уязвимостей и анализа кода, а обучающиеся учатся проверять и исправлять сгенерированный ИИ код. Предусмотрена «Политика использования ИИ-инструментов», регламентирующая допустимые и запрещённые виды деятельности. Правовой блок программы охватывает Доктрину информационной безопасности РФ, защиту персональных данных, уголовную ответственность в сфере компьютерной информации и этику ответственного раскрытия уязвимостей.

Промежуточные аттестации более чем наполовину состоят из практико-ориентированных заданий на написание кода, а итоговая аттестация реализована в формате, устойчивом к ИИ-генерации, и включает тестирование и защиту проекта с демонстрацией результатов проектной работы.

Результаты освоения программы востребованы как для дальнейшего обучения, так и в профессиональной деятельности. Изучение базовых методов защиты информации, освоение инструментов программирования и практических подходов к решению задач формируют прочную основу для успешного старта и уверенного продвижения в сфере информационной безопасности.

Цель программы: формирование у обучающихся по программам среднего профессионального образования практических навыков программирования на Python в контексте задач информационной безопасности, включая шифрование и хеширование данных, анализ веб-уязвимостей, работу с логами, а также разработку и защиту проекта.

Задачи:

1. Освоить базовые конструкции языка Python и применять их для решения прикладных задач информационной безопасности.
2. Изучить алгоритмы шифрования и хеширования с использованием стандартных библиотек Python, освоить различие между кодированием и шифрованием, а также принципы проверки целостности данных.
3. Сформировать умения распознавать, анализировать и устранять типовые веб-уязвимости, как в собственном, так и в предоставленном коде, включая код, сгенерированный ИИ-инструментами.
4. Воспитать культуру цифровой гигиены и этичного поведения.
5. Сформировать опыт завершённой инженерной деятельности: от выбора идеи проекта до его защиты.

Задачи обучения, развития, воспитания:

Задачи обучения:

сформировать отношение к компьютерной технике и современным техническим устройствам как к профессиональному инструменту для решения практических задач в области информационной безопасности и разработки безопасного программного обеспечения;

сформировать мотивацию к изучению и практическому применению программных IT-инструментов для создания и реализации проектов в сфере защиты информации, востребованных на рынке труда;

сформировать системное представление о программных разработках с помощью языка программирования Python, включая объектно-ориентированное программирование, работу с API и файловой системой;

сформировать представление о триаде информационной безопасности, правовых и этических нормах деятельности специалиста по информационной безопасности;

сформировать представление о фундаментальных различиях между кодированием, хешированием и шифрованием, а также о принципах проверки целостности данных;

изучить стандартные и специализированные библиотеки Python для решения задач информационной безопасности, включая генерацию криптостойких случайных значений;

изучить методы анализа сетевых логов, проверки базовых HTTP-заголовков безопасности, а также принципы распознавания, анализа и устранения типовых веб-уязвимостей.

Задачи воспитания:

сформировать нормы культурного проявления социальной активности и гражданской позиции в отношении тенденций развития отечественных IT-технологий в области информационной безопасности и импортозамещения;

сформировать морально-этическую основу для развития профессиональной позиции в отношении использования технологий искусственного интеллекта и нейронных сетей, включая критическую оценку сгенерированного кода и понимание ограничений ИИ;

сформировать основу для развития культуры профессионального общения в IT-сфере, в том числе навыков командной работы, парного программирования и распределения ролей при выполнении проектных задач;

воспитать ответственное отношение к соблюдению правовых и этических норм в сфере информационной безопасности, включая знание уголовной ответственности за правонарушения в сфере компьютерной информации.

Задачи развития:

способствовать развитию аналитического, системного и критического мышления, необходимого для решения профессиональных задач в сфере информационной безопасности;

развивать творческий подход к поиску и устранению уязвимостей, а также к проектированию защищённых программных решений;

развивать профессиональный тезаурус и понимание принципов безопасной разработки, соответствующих общим и профессиональным компетенциям ФГОС СПО;

развивать навыки отладки, тестирования и статического анализа кода, включая обработку граничных случаев и ошибок;

способствовать формированию первичных навыков формирования профессионального портфолио и мотивации к дальнейшему профессиональному развитию и трудоустройству в отрасли.

Формы и режим занятий

Формы занятий – теоретическая подготовка, практическая работа, самостоятельная работа.

Объем часов в неделю – до 6 академических часов.

Описание планируемых результатов обучения:

Сформированные у обучающихся базовые умения и навыки в области кибербезопасности.

По окончании обучения по программе базового уровня обучающиеся будут: иметь представление:

- о триаде информационной безопасности;
- об этических и правовых нормах деятельности в сфере информационной безопасности;
- о принципах работы веб-уязвимостей и методах их нейтрализации;
- о современных киберугрозах, включая социальную инженерию (фишинг) и атаки с использованием ИИ.

знать:

- базовый синтаксис языка Python и стандартные модули для задач ИБ;
- фундаментальные различия между кодированием, хешированием и шифрованием;
- основные веб-уязвимости и принципы безопасной работы с пользовательским вводом;
- принципы анализа сетевых логов и проверки базовых HTTP-заголовков безопасности.

уметь:

- писать скрипты для шифрования и хеширования данных с использованием стандартных библиотек Python;
- разрабатывать простые анализаторы логов для выявления подозрительной активности;
- распознавать базовые веб-угрозы и применять методы их нейтрализации;
- использовать модуль secrets для генерации криптостойких случайных значений;
- работать с файловой системой.

владеть:

- инструментами стандартной библиотеки Python;
- навыками безопасной работы с файловой системой и пользовательскими данными;
- методами отладки и тестирования программного кода (граничные случаи, обработка ошибок);
- базовыми навыками командной работы.

У обучающегося будут сформированы:

- культура цифровой гигиены и ответственного поведения в сети;
- понимание правовой ответственности и этических границ деятельности специалиста по информационной безопасности;

- критическое мышление при работе с ИИ-инструментами;
- навыки командной работы и распределения ролей;
- первичные навыки формирования профессионального портфолио;
- мотивация к дальнейшему изучению ИТ и информационной безопасности, профессиональному развитию и успешному трудоустройству в отрасли, в том числе к участию в профильных олимпиадах, соревнованиях и чемпионатах.

Планируемые результаты обучения по программе направлены на формирование у обучающихся следующих компетенций, предусмотренных ФГОС СПО по специальности

09.02.07 «Информационные системы и программирование» (утвержден Приказом Министерства образования и науки РФ от 09.12.2016 № 1547):

общие компетенции (ОК):

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности.

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях.

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде.

профессиональные компетенции (ПК):

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 4.4. Обеспечивать защиту программного обеспечения компьютерных систем программными средствами.

Планируемые результаты обучения измеримы и проверяемы через систему текущего контроля, промежуточной аттестации по каждому модулю и итогового контроля по ДОП.

Личностные результаты:

ответственное отношение к учению, способность к саморазвитию и самообразованию, осознанному выбору и построению дальнейшей индивидуальной траектории образования на базе ориентировки в профессиональных предпочтениях, с учётом устойчивых познавательных интересов и уважительного отношения к труду;

готовность и способность к самообразованию на протяжении всей жизни как условию успешной профессиональной и общественной деятельности;

формирование профессиональной позиции и гражданской ответственности в отношении обеспечения информационной безопасности, включая знание Доктрины информационной безопасности РФ и уголовной ответственности в сфере компьютерной информации;

понимание правовой ответственности и этических границ деятельности специалиста по информационной безопасности, включая этику ответственного раскрытия уязвимостей.

Метапредметные результаты:

умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в познавательной деятельности, развивать её мотивы и интересы;

умение самостоятельно планировать альтернативные пути достижения целей, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;

умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий и требований, корректировать их в соответствии с изменяющейся ситуацией;

умение самостоятельно определять цели деятельности и составлять её планы; контролировать и корректировать деятельность; использовать все возможные ресурсы для реализации планов деятельности; выбирать успешные стратегии в различных ситуациях;

умение продуктивно общаться и взаимодействовать в процессе совместной деятельности, учитывать позиции других участников, эффективно разрешать конфликты;

развитие аналитического, системного и критического мышления, необходимого для решения профессиональных задач в сфере информационной безопасности, включая анализ кода, сгенерированного ИИ-инструментами.

Предметные результаты:

представление о триаде информационной безопасности, правовых и этических нормах деятельности специалиста по информационной безопасности;

представление о фундаментальных различиях между кодированием, хешированием и шифрованием, а также о принципах проверки целостности данных;

представление о программных инструментах и библиотеках, используемых для решения профессиональных задач в области защиты информации;

практико-ориентированные навыки и умения написания программного кода на языке программирования Python в соответствии с техническим заданием по разработке безопасных приложений;

навыки и умения шифрования и хеширования данных с использованием стандартных библиотек Python, генерации криптостойких случайных значений;

навыки и умения разработки анализаторов логов, распознавания и устранения типовых веб-уязвимостей, а также безопасной обработки пользовательского ввода.

2. Учебный план

Наименование модулей/тем программы	Всего, час	Виды учебных занятий				Формы контроля
		Теория	Практика	Самостоятельная работа	Контроль/ аттестация, ак. час	
Модуль 1. Вход в ИБ: право, мышление, базовые инструменты	36	9	18	7	2	
Тема 1.1. Знакомство с платформой и ролями в ИБ	2	2	0	0	0	
Тема 1.2. Ответственное использование ИТ: персональные данные, безопасность, цифровой след и этика ИИ	3	2	0	1	0	
Тема 1.3. Как мыслит атакующий: шаги атаки и цифровой след	11	2	6	2	1	Выполнение тестовых заданий для текущего

Наименование модулей/тем программы	Всего, час	Виды учебных занятий				Формы контроля
		Теория	Практика	Самостоятельная работа	Контроль/ аттестация, ак. час	
						контроля знаний и навыков по пройденному материалу
Тема 1.4. Безопасная разработка	5	1	3	1	0	
Тема 1.5. Случайность в ИБ. Кейс-задача	14	2	9	3	0	
Промежуточная аттестация по модулю 1	1	0	0	0	1	Тестирование и выполнение практико-ориентированных заданий на написание программного кода
Модуль 2. Прикладная защита данных: фишинг, шифрование, хеширование	37	7	21	7	2	
Тема 2.1. Фишинг изнутри: письмо, легенда, триггеры	5	1	3	1	0	
Тема 2.2. Шифрование и кодирование: от криптографии к битовым операциям	10	2	6	2	0	
Тема 2.3. Хеширование и проверка целостности данных	6	1	3	1	1	Выполнение тестовых заданий для текущего контроля знаний и навыков по пройденному материалу

Наименование модулей/тем программы	Всего, час	Виды учебных занятий				Формы контроля
		Тео- рия	Прак- тика	Самостоя- тельная работа	Контроль/ аттестация, ак. час	
Тема 2.4. Стеганография и анализатор подозрительных сообщений. Кейс-задача	5	1	3	1	0	
Тема 2.5. Аутентификация и авторизация в информационных системах	10	2	6	2	0	
Промежуточная аттестация по модулю 2	1	0	0	0	1	Тестирование и выполнение практико-ориентированных заданий на написание программного кода
Модуль 3. Защитник на практике: сети, веб, логи, CTF	36	7	21	6	2	
Тема 3.1. Сети и веб: IP, порты, протоколы, заголовки	10	2	6	2	0	
Тема 3.2. CSV-логи и выявление подозрительной активности	5	1	3	1	0	
Тема 3.3. OWASP на понятном языке: XSS, SQL-инъекции и небезопасный ввод	5	1	3	0	1	Выполнение тестовых заданий для текущего контроля знаний и навыков по пройденному материалу

Наименование модулей/тем программы	Всего, час	Виды учебных занятий				Формы контроля
		Теория	Практика	Самостоятельная работа	Контроль/ аттестация, ак. час	
Тема 3.4. Методы оценки защищённости информационных систем: где заканчивается обучение и начинается ответственность	10	2	6	2	0	
Тема 3.5. Социальная инженерия. Кейс-задача	5	1	3	1	0	
Промежуточная аттестация по модулю 3	1	0	0	0	1	Тестирование и выполнение практико-ориентированных заданий на написание программного кода
Модуль 4. Итоговый проект: инструмент для кибербезопасности на Python	36	6	18	9	3	
Тема 4.1. Искусственный интеллект в атаках и защите	5	1	3	1	0	
Тема 4.2. Методы командной проверки защищенности программных модулей	10	2	6	2	0	
Тема 4.3. Выбор финального проекта: 3 варианта на выбор	6	1	3	1	1	Выполнение тестовых заданий для текущего контроля знаний и навыков

Наименование модулей/тем программы	Всего, час	Виды учебных занятий				Формы контроля
		Теория	Практика	Самостоятельная работа	Контроль/ аттестация, ак. час	
						по пройденному материалу
Тема 4.4. Разработка и отладка проекта	8	1	3	4	0	
Тема 4.5. Оформление проекта	5	1	3	1	0	
Промежуточная аттестация по модулю 4	1	0	0	0	1	Тестирование и выполнение практико-ориентированных заданий на написание программного кода
Итоговый контроль*	1	0	0	0	1	Комплексное задание: тест и защита проекта
Итого	145	29	78	29	9	
ИТиСИ	Объем часов ИТиСИ не входит в общую трудоемкость ДОП, но участие в ИТиСИ обязательно для всех успешно прошедших Итоговый контроль/аттестацию по ДОП					

* Часы итогового контроля входят в 4 модуль.

3. Календарно-тематическое планирование

Название ДОП	№ потока	Дата начала обучения по ДОП	№ модуля ДОП	Начало обучения по модулю ДОП	Календарный период (количество дней) длительности модуля	Дата окончания обучения по ДОП
Кибербезопасность. СПО (базовый уровень)	1	2026-10-01	1	2026-10-01	61	2027-05-28
			2	2026-12-01	67	
			3	2027-02-08	54	
			4	2027-04-05	54	

4. Рабочая программа

Порядковый номер модуля	Наименование темы	Лекции. Количество академических часов	Содержание теоретической подготовки	Практические занятия. Количество академических часов	Содержание практической работы	Самостоятельная работа. Количество академических часов	Содержание самостоятельной работы
1	Вход в ИБ: право, мышление, базовые инструменты						
1	Тема 1.1. Знакомство с платформой и ролями в ИБ	2	Три основные роли в информационной безопасности: пентестер (атакующий/red-team), защитник (defender/blue team), аналитик ИБ. Базовые понятия: угроза (потенциальная возможность нарушения безопасности), уязвимость (слабое место в системе), атакующий (лицо, пытающееся нарушить безопасность). Триада CIA: конфиденциальность (Confidentiality), целостность (Integrity), доступность (Availability) - фундамент модели ИБ. Платформа LMS: структура кабинета,	0	Практическая работа в указанной теме Дополнительной общеобразовательной программы не предусмотрена и проводиться не будет.	0	Самостоятельная работа и самоконтроль в указанной теме Дополнительной общеобразовательной программы не предусмотрены и проводиться не будут.

			форматы заданий, сроки выполнения. Команды Red Team (имитация атакующих) и Blue Team (защита инфраструктуры).				
1	Тема 1.2. Ответственное использование ИТ: персональные данные, безопасность, цифровой след и этика ИИ	2	Доктрина информационной безопасности РФ - стратегический документ государства в сфере ИБ. Статьи УК РФ: ст. 272 (неправомерный доступ), ст. 273 (вредоносные программы), ст. 274 (нарушение правил эксплуатации). 152-ФЗ - федеральный закон о защите персональных данных. NDA (Non-Disclosure Agreement) - соглашение о неразглашении, обязательное перед пентестом. Bug bounty - легальный формат поиска уязвимостей с вознаграждением. Score - границы тестирования. GDPR - европейский регламент по защите персональных данных. РП - информация, позволяющая	0	Практическая работа в указанной теме Дополнительной общеобразовательной программы не предусмотрена и проводиться не будет.	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.

			идентифицировать человека.				
1	Тема 1.3. Как мыслит атакующий: шаги атаки и цифровой след	2	Пять шагов модели кибератаки: сбор информации → выбор слабого места → проникновение → закрепление → достижение цели. OSINT (Open Source Intelligence) - разведка по открытым источникам информации. Цифровой след - совокупность всех данных о пользователе в сети (история поиска, соцсети, метаданные). CVE (Common Vulnerabilities and Exposures) - каталог уникальных идентификаторов известных уязвимостей. MITRE ATT&CK - база знаний о тактиках, техниках и процедурах (TTP) атакующих.	6	Краткое повторение теории, необходимой для выполнения практических заданий. Разбор с обучающимися примеров моделирования этапов кибератаки и сбора информации из открытых источников. Выполнение обучающимися практических заданий по выявлению уязвимостей цифрового следа. Подробное описание практических заданий приложено к ДОП. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».	2	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.
1	Тема 1.4. Безопасная разработка	1	Принцип «не доверяй вводу» (Never Trust User Input) - любой ввод извне считается потенциально опасным. Валидация - проверка ввода на соответствие правилам.	3	Краткое повторение теории о принципах безопасной разработки и валидации ввода. Разбор с обучающимися примеров проверки строк на запрещённые	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление

			Белый список безопаснее чёрного списка. Санитизация - удаление или замена опасных символов во вводе. <code>html.escape()</code> - функция Python для экранирования HTML-символов, защита от XSS. Параметризованные запросы - наиболее надёжный метод защиты от SQL-инъекций.		символы и валидации пароля. Выполнение обучающимися практических заданий. Подробное описание практических заданий приложено к ДОП. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.
1	Тема 1.5. Случайность в ИБ. Кейс-задача	2	Модуль <code>random</code> - детерминированный ГПСЧ (Mersenne Twister), непригоден для криптографических целей. 624 последовательных числа Mersenne Twister - достаточно для предсказания всей последовательности. Модуль <code>secrets</code> - криптографически стойкий генератор, использует <code>os.urandom()</code> из энтропии ОС. 4-значный PIN: 10 000 комбинаций (10^4). Блокировка после N неудачных попыток -	9	Краткое повторение теории о генерации случайных чисел, различиях между <code>random</code> и <code>secrets</code>, принципах защиты. Разбор с обучающимися примеров моделирования подбора PIN-кода с блокировкой и генерации одноразовых кодов. Выполнение обучающимися практических заданий, а также решение кейс-задачи по анализу и исправлению уязвимостей в системе аутентификации.	3	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.

			основная защита от brute force. Entropy pool - накопитель энтропии из различных источников системы (мышь, клавиатура, сеть).		Подробное описание практических заданий приложено к ДОП. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		
2	Прикладная защита данных: фишинг, шифрование, хеширование						
2	Тема 2.1. Фишинг изнутри: письмо, легенда, триггеры	1	Фишинг - вид социальной инженерии, обман пользователя для получения конфиденциальной информации. Триггеры фишинга: срочность, страх, любопытство, жадность - основные психологические механизмы. Spear phishing - целенаправленная атака на конкретного человека с использованием персональной информации. Whaling - фишинг против топ-менеджеров и VIP-персон. Clone phishing - копирование легитимного письма с заменой	3	Краткое повторение теории о видах фишинга, психологических триггерах и защите от них. Разбор с обучающимися примеров анализа и выявления признаков атаки. Выполнение обучающимися практических заданий по анализу текста письма на наличие маркеров фишинга на Python, в том числе вариативного задания. Подробное описание практических заданий приложено к ДОП. Дополнительные упражнения по теме представлены в методической разработке «Материалы для	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.

			ссылки/вложения на вредоносное. DMARC - механизм проверки подлинности отправителя email. X-Frame-Options - защита от clickjacking.		преподавателя по проведению практических занятий».		
2	Тема 2.2. Шифрование и кодирование: от криптографии к битовым операциям	2	Шифрование - преобразование данных с использованием ключа. Кодирование - просто формат представления, ключ не нужен. XOR (исключающее ИЛИ) - побитовая операция: $A \oplus A = 0$, $A \oplus 0 = A$. Коммутативна и самообратима. Шифр Цезаря - сдвиг алфавита на фиксированное число позиций. Простейший шифр подстановки. Base64 - кодирование бинарных данных в текстовый формат, НЕ шифрование. Симметричное шифрование (AES) - один ключ для шифрования и дешифрования. Асимметричное шифрование (RSA) - пара ключей: открытый и закрытый.	6	Краткое повторение теории о различиях между шифрованием и кодированием, алгоритмах XOR, Цезаря и Base64. Разбор с обучающимися примеров реализации шифра Цезаря и XOR-шифрования. Выполнение обучающимися практических заданий по написанию функций для шифрования и дешифрования с использованием XOR и Цезаря, а также кодирования и декодирования Base64. Подробное описание практических заданий приложено к ДОП. Дополнительные упражнения по теме представлены в методической разработке «Материалы для	2	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.

					преподавателя по проведению практических занятий».		
2	Тема 2.3. Хеширование и проверка целостности данных	1	Хеш-функция - одностороннее преобразование данных в строку фиксированной длины. Необратимость - невозможность восстановить исходные данные по хешу. Лавинный эффект - малое изменение входа вызывает большое изменение хеша (~50% бит). MD5 - устаревшая хеш-функция (128 бит), уязвима к коллизиям. SHA-256 - рекомендуемый стандарт (256 бит). Соль (salt) - случайные данные, добавляемые к паролю перед хешированием для защиты от радужных таблиц. HMAC - код аутентификации сообщения на основе хеша с ключом.	3	Краткое повторение теории о хеш-функциях, свойствах MD5 и SHA-256, использовании соли. Разбор с обучающимися примеров вычисления хешей и проверки целостности данных. Выполнение обучающимися практических заданий по написанию функций для расчёта хешей, а также проверки целостности строк и файлов, в том числе вариативного задания. Подробное описание практических заданий приложено к ДОП. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.
2	Тема 2.4. Стеганография и анализатор подозрительных	1	Стеганография - скрытие факта передачи сообщения. Криптография - скрытие содержания сообщения.	3	Краткое повторение теории о стеганографии, LSB-методе, метаданных и правовых аспектах дроппинга. Разбор с	1	Повторение и систематизация пройденного материала по ключевым понятиям и

	сообщений. Кейс-задача		LSB (Least Significant Bit) - метод стеганографии, замена младших битов пикселей изображения. Метаданные - информация о файле: автор, дата, программа, GPS-координаты. Стегоанализ - процесс обнаружения скрытых сообщений в файлах. Дроппинг - размещение вредоносного ПО на компьютере жертвы (ст. 159 и ст. 174.1 УК РФ).		обучающимися примеров анализа подозрительных сообщений. Выполнение обучающимися практических заданий по реализации стеганографического метода в тексте, а также решение кейс-задачи по разработке анализатора подозрительных вакансий и признаков дроппинга. Подробное описание практических заданий приложено к ДОП. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.
2	Тема 2.5. Аутентификация и авторизация в информационных системах	2	Аутентификация - подтверждение личности пользователя. Авторизация - проверка прав доступа. Двухфакторная аутентификация (2FA) - два фактора: что-то, что знаешь + что-то, что имеешь. JWT (JSON Web Token) - стандарт для	6	Краткое повторение теории о аутентификации, авторизации, многофакторной аутентификации. Разбор с обучающимися примеров реализации простой системы аутентификации. Выполнение	2	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий

			передачи токенов авторизации в формате JSON. bcrypt - адаптивный алгоритм хеширования паролей с настраиваемой сложностью. Session hijacking - перехват идентификатора сессии пользователя. OAuth - протокол авторизации для предоставления доступа сторонним приложениям без передачи пароля.		обучающимися практических заданий по созданию функций для аутентификации пользователей, а также оценки стойкости паролей. Подробное описание практических заданий приложено к ДОП. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.
3	Защитник на практике: сети, веб, логи, СТФ						
3	Тема 3.1. Сети и веб: IP, порты, протоколы, заголовки	2	IP-адрес - уникальный числовой идентификатор устройства в сети. TCP/IP: 65535 портов (0–65535). HTTP - порт 80, HTTPS (с шифрованием TLS/SSL) - порт 443, SSH - порт 22, FTP - порт 21. DNS - система доменных имён, преобразующая имена (google.com) в IP-адреса. HTTP-заголовки - метаданные запроса/ответа: Content-Type, Authorization, User-Agent, Cookie. Клиент-	6	Краткое повторение теории о сетевых протоколах, IP-адресации, портах, HTTP-заголовках. Разбор с обучающимися примеров анализа URL и проверки подозрительных паттернов. Выполнение обучающимися практических заданий по написанию функций для анализа URL и проверки HTTP-заголовков безопасности с использованием Python.	2	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.

			серверная архитектура: клиенты отправляют запросы, серверы обрабатывают и отвечают.		Выполнение вариативного задания. Подробное описание практических заданий приложено к ДОП. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		
3	Тема 3.2. CSV-логи и выявление подозрительной активности	1	CSV (Comma-Separated Values) - текстовый файл с разделителями-запятыми для табличных данных. Лог - хронологическая запись событий: входы, ошибки, действия пользователей. Подозрительный IP - адрес с аномальной активностью: ≥ 3 неудачных попыток входа с одного адреса. Модуль csv в Python - чтение и запись CSV-файлов. collections.Counter - подсчёт частоты элементов. ISO 8601 - стандартный формат даты/времени в логах	3	Краткое повторение теории о формате CSV, структуре логов и методах выявления аномалий. Разбор с обучающимися примеров чтения и анализа CSV-файлов с логами. Выполнение обучающимися практических заданий по написанию скрипта для обработки CSV-логов, выявления подозрительной активности. Подробное описание практических заданий приложено к ДОП. Дополнительные упражнения по теме представлены в методической разработке	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.

			(YYYY-MM-DDTHH:MM:SS).		«Материалы для преподавателя по проведению практических занятий».		
3	Тема 3.3. OWASP на понятном языке: XSS, SQL-инъекции и небезопасный ввод	1	OWASP (Open Web Application Security Project) - открытый проект по безопасности веб-приложений. OWASP Top 10 - 10 наиболее критичных рисков безопасности веб-приложений. XSS (Cross-Site Scripting) - внедрение вредоносного JavaScript на веб-страницу. A03:2021. SQL-инъекция - внедрение вредоносного SQL-кода в запрос к базе данных. A01:2021. Параметризованные запросы (prepared statements) - разделение кода и данных, защита от SQL-инъекций. Экранирование HTML: & → &, < → <, > → > - защита от XSS.	3	Краткое повторение теории об OWASP Top 10, XSS и SQL-инъекциях, методах защиты. Разбор с обучающимися примеров опасного кода и его исправления. Выполнение обучающимися практических заданий по построению безопасных SQL-запросов и обнаружению XSS-паттернов в строках. Выполнение вариативного задания. Подробное описание практических заданий приложено к ДОП. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».	0	Самостоятельная работа и самоконтроль в указанной теме Дополнительной общеобразовательной программы не предусмотрены и проводиться не будут.
3	Тема 3.4. Методы оценки защищённости	2	CTF (Capture The Flag) - соревнования по решению задач ИБ.	6	Краткое повторение теории о форматах CTF, Bug Bounty,	2	Повторение и систематизация пройденного

	информационных систем: где заканчивается обучение и начинается ответственность		Jeopardy-style - задачи по категориям. Attack-Defense CTF - командное соревнование: защита своих сервисов и атака на сервисы соперников. Bug Bounty - легальная программа вознаграждения за найденные уязвимости. Responsible disclosure - ответственное раскрытие: сообщение владельцу → 90 дней → публикация. CVSS - система оценки критичности уязвимостей (0–10). 9.0–10.0 - критический уровень. Zero-day - уязвимость, о которой неизвестно разработчику и для которой нет патча.		ответственном раскрытии и CVSS. Разбор с обучающимися примеров анализа уязвимостей. Выполнение обучающимися практических заданий на уязвимости. Выполнение вариативного задания. Подробное описание практических заданий приложено к ДОП. Проведение проверочного тестирования для закрепления изученного материала. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.
3	Тема 3.5. Социальная инженерия. Кейс-задача	1	Социальная инженерия - манипуляция людьми для получения конфиденциальной информации. Pretexting - создание выдуманного сценария (легенды) для установления контакта. Tailgating - следование за	3	Краткое повторение теории о методах социальной инженерии, фишинге, претекстинге. Разбор с обучающимися примеров фишинговых URL и сообщений. Выполнение обучающимися	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических

			авторизованным лицом в защищённую зону без пропуска. Baiting - оставление заражённого USB-носителя в надежде на любопытство жертвы. Vishing - фишинг через голосовые звонки. Smishing - фишинг через SMS.		практических заданий, а также решение кейс-задачи по расследованию фишинговой атаки и автоматизация анализа логов/URL. Подробное описание практических заданий приложено к ДОП. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.
4	Итоговый проект: инструмент для кибербезопасности на Python						
4	Тема 4.1. Искусственный интеллект в атаках и защите	1	Deepfake - поддельное видео/аудио, созданное с помощью ИИ для имитации реальных людей. LLM-фишинг - использование больших языковых моделей для создания убедительных фишинговых писем. Prompt injection - атака на LLM через внедрение вредоносных инструкций в промпт. Data poisoning - внесение искажённых данных в обучающую выборку ИИ. Adversarial attacks - специально подготовленные данные	3	Краткое повторение теории о роли ИИ в атаках и защите Разбор с обучающимися примеров анализа текстов на признаки манипуляции и использования ИИ для генерации кода. Выполнение обучающимися практических заданий по поиску признаков манипуляции или ошибки и улучшению предупреждения о фишинге. Подробное описание практических	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.

			для обмана ИИ-моделей. Галлюцинации ИИ - генерация правдоподобного, но недостоверного контента.		заданий приложено к ДОП. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		
4	Тема 4.2. Методы командной проверки защищенности программных модулей	2	Аудит кода - систематическая проверка исходного кода на наличие уязвимостей и ошибок безопасности. Типовые ошибки: небезопасная конкатенация SQL, отсутствие валидации, хранение паролей в открытом виде. Роли в команде: аналитик (поиск проблем), исправляющий (внесение изменений), документатор (описание). Чек-лист безопасности: валидация ввода, параметризованные запросы, хеширование, обработка ошибок. Индивидуальный вклад - обязательный критерий оценки группового задания.	6	Краткое повторение теории об аудите кода, типовых ошибках безопасности. Разбор с обучающимися примеров поиска уязвимостей в коде. Выполнение обучающимися практических заданий по аудиту кода и выявлению опасных паттернов. Подробное описание практических заданий приложено к ДОП. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».	2	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.

4	Тема 4.3. Выбор финального проекта: 3 варианта на выбор	1	Варианты итогового проекта из трех направлений. Разбор структуры, требований к коду и документу.	3	Краткое повторение теории, необходимой для выполнения практических заданий. Разбор с обучающимися примеров трёх направлений итогового проекта и критериев их оценки. Выбор обучающимися одного из вариантов проекта. Выполнение обучающимися практических заданий по проектированию структуры программы и созданию первого рабочего прототипа. Подробное описание практических заданий приложено к ДОП. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопро верки с автоматической системой оценивания.
4	Тема 4.4. Разработка и отладка проекта	1	Доработка проекта: отладка на граничных случаях, тестирование ввода, проверка обработки ошибок. Самопроверка по чек-	3	Краткое повторение теории, необходимой для выполнения практических заданий. Разбор с обучающимися примеров отладки кода и	4	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам

			листу безопасности: валидация, хеширование, secrets вместо random. Читаемый код: осмысленные имена переменных, комментарии, логическая структура. Документация: README-файл с описанием, инструкцией по запуску, примерами использования.		тестирования на граничных случаях. Выполнение обучающимися практических заданий по доработке собственного проекта, проверке обработки ошибок. Подробное описание практических заданий приложено к ДОП. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.
4	Тема 4.5. Оформление проекта	1	Финальная доработка: полировка интерфейса, обработка последних замечаний. Репетиция защиты: демонстрация основных функций, хронометраж доклада (3–5 минут). Подготовка ответов на вопросы: почему выбран алгоритм, как обеспечивается безопасность.	3	Краткое повторение теории, необходимой для выполнения практических заданий. Разбор с обучающимися примеров написания технической документации (README) и подготовки к публичному выступлению. Выполнение обучающимися практических заданий по финальной доработке проекта, оформлению	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с

				файла README, репетиции защиты и подготовке ответов на технические вопросы комиссии. Подробное описание практических заданий приложено к ДОП. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		автоматической системой оценивания.
--	--	--	--	--	--	-------------------------------------

5. Формы аттестации и оценочные материалы, включая примеры контрольных заданий

5.1 Текущий контроль

В таблице ниже представлены по каждому из модулей:

- количество часов текущего контроля;
- формы текущего контроля с подробным описанием процедуры оценивания результатов обучения;
- диагностические инструменты с примерами (контрольные задания, материалы, промежуточные тесты и задачи);
- показатели и критерии оценивания;
- шкала оценивания.

Текущий контроль. Количество ак. часов	1	1	1	1
Текущий контроль. Формы контроля	Тестирование	Тестирование	Тестирование	Тестирование
Текущий контроль. Диагностические инструменты	Задание в виде теста. Вопрос 1 Кто такой пентестер (этичный хакер)? А) Специалист, который защищает системы от атак	Задание в виде теста. Вопрос 1 Что такое валидация ввода? А) Замена специальных символов на безопасные	Задание в виде теста. Вопрос 1 Задание с кратким ответом. Впишите только число.	Задание в виде теста. Вопрос 1 Что такое CTF (Capture The Flag) в контексте ИБ? А) Игра для развлечения

Б) Специалист, который анализирует логи безопасности В) Специалист, который пишет документацию Г) Специалист, который легально тестирует системы на уязвимости Правильный ответ: Г) Специалист, который легально тестирует системы на уязвимости Вопрос 2 Что такое Доктрина информационной безопасности РФ? А) Уголовный кодекс РФ Б) Стратегический документ государства В) Правила работы в интернете Г) Инструкция для системных администраторов Правильный ответ: Б) Стратегический документ государства Вопрос 3 Задание с кратким ответом. Впишите только число. Сколько основных ролей в ИБ рассмотрено в теме 1.1 (пентестер, защитник, аналитик)? Ответ: _____	Б) Шифрование введённых данных В) Проверка ввода на соответствие заданным правилам Г) Очистка строки от опасных символов Правильный ответ: В) Проверка ввода на соответствие заданным правилам Вопрос 2 Что такое фишинг? А) Обман пользователя для получения конфиденциальной информации Б) Подбор пароля методом перебора В) Перехват сетевого трафика Г) Взлом сервера через уязвимость Правильный ответ: А) Обман пользователя для получения конфиденциальной информации Вопрос 3 Что означает принцип «не доверяй вводу» (Never Trust User Input)? А) Весь ввод извне считается потенциально опасным Б) Ввод нужно шифровать перед обработкой	Сколько уровней риска обычно определяет анализатор подозрительных сообщений (низкий, средний, высокий)? Ответ: _____ Правильный ответ: 3 Вопрос 2 Что такое стеганография? А) Метод шифрования данных Б) Хеширование данных В) Скрытие факта передачи сообщения Г) Сжатие файлов Правильный ответ: В) Скрытие факта передачи сообщения Вопрос 3 Чем стеганография отличается от криптографии? А) Стеганография скрывает факт передачи, криптография — содержание Б) Стеганография скрывает содержание, криптография — факт передачи В) Стеганография сложнее криптографии Г) Разницы нет	Б) Соревнование по хакерству В) Соревнование по решению задач информационной безопасности Г) Тест на знание программирования Правильный ответ: В) Соревнование по решению задач информационной безопасности Вопрос 2 Что такое Bug Bounty? А) Программа вознаграждения за найденные уязвимости Б) Лицензия на пентестинг В) Сертификат специалиста Г) Штраф за найденные уязвимости Правильный ответ: А) Программа вознаграждения за найденные уязвимости Вопрос 3 Что такое deepfake? А) Метод аутентификации Б) Поддельное видео или аудио, созданное с помощью ИИ В) Тип вредоносного ПО Г) Глубокое шифрование Правильный ответ: Б) Поддельное видео или
---	--	---	---

	Правильный ответ: 3 Вопрос 4 Кто такой защитник (defender) в ИБ? А) Специалист, который проводит OSINT Б) Специалист, который защищает инфраструктуру от атак В) Специалист, который атакует системы Г) Специалист, который пишет вредоносное ПО Правильный ответ: Б) Специалист, который защищает инфраструктуру от атак Вопрос 5 Что такое OSINT? А) Тип вредоносного ПО Б) Операционная система для пентестинга В) Протокол защищённой передачи данных Г) Разведка по открытым источникам информации Правильный ответ: Г) Разведка по открытым источникам информации	В) Нужно спрашивать подтверждение у пользователя Г) Пользователь должен ввести пароль дважды Правильный ответ: А) Весь ввод извне считается потенциально опасным Вопрос 4 Задание с кратким ответом. Впишите только число. Сколько символов минимум должен содержать пароль по стандартным критериям безопасности (длина + сложность)? Ответ: _____ Правильный ответ: 8 Вопрос 5 Какой модуль Python использует алгоритм Mersenne Twister для генерации псевдослучайных чисел? А) secrets Б) random В) hashlib Г) os Правильный ответ: Б) random	Правильный ответ: А) Стеганография скрывает факт передачи, криптография — содержание Вопрос 4 Что такое аутентификация? А) Проверка прав доступа Б) Резервное копирование В) Подтверждение личности пользователя Г) Шифрование данных Правильный ответ: В) Подтверждение личности пользователя Вопрос 5 Что такое IP-адрес? А) Уникальный числовой идентификатор устройства в сети Б) Номер телефона В) Пароль доступа Г) Имя компьютера Правильный ответ: А) Уникальный числовой идентификатор устройства в сети	аудио, созданное с помощью ИИ Вопрос 4 Задание с кратким ответом. Впишите только число. Какой стандартный срок (в днях) даётся на исправление уязвимости при responsible disclosure? Ответ: _____ Правильный ответ: 90 вопрос 5 Что такое социальная инженерия? А) Проектирование социальных сетей Б) Анализ социальных графов В) Разработка UI/UX Г) Манипуляция людьми для получения конфиденциальной информации Правильный ответ: Г) Манипуляция людьми для получения конфиденциальной информации
Текущий контроль. Описание процедуры оценивания Показатели и	Критерий оценивания: правильность ответа на тестовый вопрос. Показатель оценивания: количество правильных	Критерий оценивания: правильность ответа на тестовый вопрос. Показатель оценивания: количество правильных ответов на	Критерий оценивания: правильность ответа на тестовый вопрос. Показатель оценивания: количество правильных	Критерий оценивания: правильность ответа на тестовый вопрос. Показатель оценивания: количество правильных

критерии оценивания	ответов на тестовые задания. За каждый правильный ответ начисляется 1 балл. В случае неправильного ответа баллы не начисляются.	тестовые задания. За каждый правильный ответ начисляется 1 балл. В случае неправильного ответа баллы не начисляются.	ответов на тестовые задания. За каждый правильный ответ начисляется 1 балл. В случае неправильного ответа баллы не начисляются.	ответов на тестовые задания. За каждый правильный ответ начисляется 1 балл. В случае неправильного ответа баллы не начисляются.
Текущий контроль. Шкала оценивания, нижнее значение	0	0	0	0
Текущий контроль. Шкала оценивания, верхнее значение	5	5	5	5
Текущий контроль. Шкала оценивания, минимальный проходной балл для успешной сдачи	3	3	3	3

5.2 Промежуточная аттестация

В таблице ниже представлены по каждому из модулей:

- количество часов промежуточной аттестации;
- формы контроля (промежуточной аттестации) с подробным описанием процедуры оценивания результатов обучения;
- диагностические инструменты с примерами (контрольные задания, материалы, промежуточные тесты и задачи);
- показатели и критерии оценивания;
- шкала оценивания.

Аттестация по итогам модуля. Количество ак. часов	1	1	1	1
Аттестация по итогам модуля. Формы контроля	Тестирование и выполнение практико-ориентированных	Тестирование и выполнение практико-ориентированных заданий	Тестирование и выполнение практико-ориентированных	Тестирование и выполнение практико-ориентированных

	заданий на написание программного кода	на написание программного кода	заданий на написание программного кода	заданий на написание программного кода
Аттестация по итогам модуля. Диагностические инструменты	Тест и практико-ориентированные задания. Вопрос 1 Сколько основных ролей в ИБ рассматривается в теме (пентестер, защитник, аналитик)? Ответ: _____ Правильный ответ: 3 Вопрос 2 Сколько дней стандартно даётся на исправление уязвимости? Ответ: _____ Правильный ответ: 90 Вопрос 3 Сколько шагов включает модель кибератаки? Ответ: _____ Правильный ответ: 5 Вопрос 4 Кто такой пентестер? А) Тестирует на уязвимости Б) Анализирует логи В) Пишет документацию Г) Защищает системы	Тест и практико-ориентированные задания. Вопрос 1 Сколько основных признаков фишинга в письме (минимум)? Ответ: _____ Правильный ответ: 3 Вопрос 2 Сколько бит в одном байте? Ответ: _____ Правильный ответ: 8 Вопрос 3 Сколько символов в hex представлении MD5? Ответ: _____ Правильный ответ: 32 Вопрос 4 Что такое фишинг? А) DDoS атака Б) Обман пользователя В) Вирус Г) Взлом сервера Правильный ответ: Б) Обман пользователя Вопрос 5 Что такое шифрование? А) Преобразование в нечитаемый вид Б) Копирование данных В) Сжатие данных Г) Удаление данных	Тест и практико-ориентированные задания. Вопрос 1 Сколько бит в IPv4 адресе? Ответ: _____ Правильный ответ: 32 Вопрос 2 Сколько полей в типичной CSV записи лога? Ответ: _____ Правильный ответ: 5 Вопрос 3 Сколько типов XSS атак существует? Ответ: _____ Правильный ответ: 3 Вопрос 4 Что такое IP адрес? А) Сетевой адрес устройства Б) Номер порта С) Протокол Д) Имя компьютера Правильный ответ: А) Сетевой адрес устройства Вопрос 5 Что такое CSV файл? А) База данных Б) Архив С) Изображение Д) Текстовый файл с разделителями Правильный ответ: Д) Текстовый файл с разделителями	Тест и практико-ориентированные задания. Вопрос 1 Сколько участников минимально нужно в команде аудита? Ответ: _____ Правильный ответ: 3 Вопрос 2 Сколько языков поддерживает типичная LLM? Ответ: _____ Правильный ответ: 50 Вопрос 3 Сколько типов логирования рассматривается? Ответ: _____ Правильный ответ: 3 Вопрос 4 Что такое признаки стиля? А) Формат файла Б) Кодировка В) Размер Г) Характеристики текста Правильный ответ: Г) Характеристики текста Вопрос 5 Что такое обучение модели? А) Сжатие Б) Удаление данных В) Настройка параметров Г) Шифрование

	Правильный ответ: А) Тестирует на уязвимости Вопрос 5 Что такое NDA? А) Соглашение о неразглашении Б) Нормативный документ В) Национальная директива Г) Новый документ Правильный ответ: А) Соглашение о неразглашении Вопрос 6 Что такое OSINT? А) Вредоносное ПО Б) Разведка по открытым источникам В) Операционная система Г) Протокол Правильный ответ: Б) Разведка по открытым источникам Вопрос 7 Какие роли в ИБ? (Выберите все правильные ответы) ☐ Пентестер ☐ Защитник ☐ Аналитик ☐ Менеджер	Правильный ответ: А) Преобразование в нечитаемый вид Вопрос 6 Что такое хеш-функция? А) Шифрование Б) Кодирование В) Сжатие Г) Одностороннее преобразование Правильный ответ: Г) Одностороннее преобразование Вопрос 7 Признаки фишингового письма? (Выберите все правильные ответы) ☐ Срочность ☐ Ошибки ☐ Незнакомый отправитель ☐ Подпись Правильный ответ: Срочность, Ошибки, Незнакомый отправитель ТРЕНАЖЁРЫ КОДА Вопрос 8 Напишите функцию check_phishing_email(sender, subject, body), которая проверяет письмо на признаки фишинга. Шаблон кода:	Вопрос 6 Что такое XSS? А) CSRF атака Б) SQL инъекция С) Межсайтовый скриптинг Д) DDoS атака Правильный ответ: С) Межсайтовый скриптинг Вопрос 7 Какие протоколы относятся к прикладному уровню? (Выберите все правильные ответы) ☐ HTTP ☐ FTP ☐ TCP ☐ IP Правильный ответ: HTTP, FTP ТРЕНАЖЁРЫ КОДА Вопрос 8 Напишите функцию check_url(url), которая принимает URL как строку и проверяет наличие подозрительных паттернов (IP вместо домена, лишние @, опечатки). Шаблон кода: def check_url(url): # TODO: напишите код pass Поле для вашего кода: _____ ЭТАЛОННОЕ РЕШЕНИЕ: import re def check_url(url):	Правильный ответ: В) Настройка параметров Вопрос 6 Что такое аудит кода? А) Проверка безопасности Б) Компиляция В) Документирование Г) Тестирование Правильный ответ: А) Проверка безопасности Вопрос 7 Какие признаки манипуляции в тексте? (Выберите все правильные ответы) ☐ Срочность ☐ Страх ☐ Авторитет ☐ Спокойствие Правильный ответ: Срочность, Страх, Авторитет ТРЕНАЖЁРЫ КОДА Вопрос 8 Напишите функцию calculate_individual_contribution(team_results, member_id), которая рассчитывает вклад участника. Шаблон кода: def calculate_individual_contribution(team_results, member_id): # TODO: напишите код pass Поле для вашего кода:
--	---	--	---	--

	Правильные ответы: Пентестер, Защитник, Аналитик ТРЕНАЖЁРЫ КОДА Вопрос 8 Напишите функцию count_roles(roles), которая принимает список ролей и возвращает количество уникальных ролей. Шаблон кода: def count_roles(roles): # TODO: напишите код pass Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: def count_roles(roles): return len(set(roles)) Вопрос 9 Напишите функцию classify_threat(threat_type), которая классифицирует угрозу по типу. Шаблон кода: def classify_threat(threat_type): # TODO: напишите код pass Поле для вашего кода:	<pre>def check_phishing_email(sender, subject, body): # TODO: напишите код pass Поле для вашего кода:</pre> ЭТАЛОННОЕ РЕШЕНИЕ: def check_phishing_email(sender, subject, body): suspicious_domains = ['fake.com', 'phish.net'] urgent_words = ['срочно', 'urgent', 'немедленно'] <pre>score = 0 if any(domain in sender for domain in suspicious_domains): score += 1 if any(word in subject.lower() for word in urgent_words): score += 1 if len(sender) > 50: score += 1</pre> return 'suspicious' if score >= 2 else 'safe' Вопрос 9 Напишите функцию extract_domain(email),	<pre>suspicious = [] if re.search(r'd{1,3}\.d{1,3}\.d{1,3}\. \d{1,3}', url): suspicious.append('ip_in_url') if url.count('@') > 1: suspicious.append('multiple_at') known = ['google.com', 'yandex.ru', 'vk.com'] for k in known: if k.replace('o', '0') in url or k.replace('e', '3') in url: suspicious.append('typo') return suspicious Вопрос 9 Напишите функцию parse_http_response(response_text), которая парсит HTTP ответ и извлекает статус-код. Шаблон кода: def parse_http_response(response_text): # TODO: напишите код Pass Поле для вашего кода:</pre> ЭТАЛОННОЕ РЕШЕНИЕ: def parse_http_response(response_text): lines = response_text.split("\n") if lines: first_line = lines[0] parts = first_line.split(' ') if len(parts) >= 2: return int(parts[1])	ЭТАЛОННОЕ РЕШЕНИЕ: def calculate_individual_contribution(team_results, member_id): member = team_results.get(member_id, {}) findings = member.get('findings_count', 0) fixes = member.get('fixes_count', 0) docs = member.get('documentation_score', 0) total = findings * 3 + fixes * 2 + docs return {'member_id': member_id, 'contribution_score': total} Вопрос 9 Напишите функцию analyze_anomalies(data_points), которая находит аномалии в наборе данных. Шаблон кода: def analyze_anomalies(data_points): # TODO: напишите код pass Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: def analyze_anomalies(data_points): if not data_points:
--	---	---	--	--

	ЭТАЛОННОЕ РЕШЕНИЕ: <pre>def classify_threat(threat_type): threats = {'hacker': 'attacker', 'virus': 'malware', 'flood': 'disaster'} return threats.get(threat_type, 'unknown')</pre> Вопрос 10 Напишите функцию <code>check_nda(days)</code>, которая проверяет, не истёк ли срок NDA (90 дней) Шаблон кода: <pre>def check_nda(days): # TODO: напишите код pass</pre> Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: <pre>def check_nda(days): return 'valid' if days <= 90 else 'expired'</pre> Вопрос 11 Напишите функцию <code>parse_articles(text)</code>, которая извлекает номера статей УК из текста.	которая извлекает домен из email адреса. Шаблон кода: <pre>def extract_domain(email): # TODO: напишите код Pass</pre> Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: <pre>def extract_domain(email): if '@' in email: return email.split('@')[1] return None</pre> Вопрос 10 Напишите функцию <code>xor_cipher(text, key)</code>, которая шифрует текст XOR операцией. Шаблон кода: <pre>def xor_cipher(text, key): # TODO: напишите код pass</pre> Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: <pre>def xor_cipher(text, key): result = [] for i, char in enumerate(text): result.append(chr(ord(char) ^ ord(key[i % len(key)]))) return ''.join(result)</pre> Вопрос 11	return None Вопрос 10 Напишите функцию <code>count_logins_from_csv(csv_lines)</code>, которая читает CSV-строки с логам и подсчитывает успешные и неуспешные входы. Шаблон кода: <pre>def count_logins_from_csv(csv_lines): # TODO: напишите код pass</pre> Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: <pre>def count_logins_from_csv(csv_lines): success = 0 failed = 0 for line in csv_lines: if 'success' in line.lower(): success += 1 elif 'failed' in line.lower() or 'error' in line.lower(): failed += 1 return {'success': success, 'failed': failed}</pre> Вопрос 11 Напишите функцию <code>find_suspicious_ips(log_entries, threshold)</code>, которая находит IP с количеством неуспешных входов $\geq$ threshold. Шаблон кода:	<pre>return [] avg = sum(data_points) / len(data_points) threshold = avg * 2 return [x for x in data_points if x > threshold]</pre> Вопрос 10 Напишите функцию <code>compare_texts(texts)</code>, которая сравнивает 3 текста и определяет, какой от ИИ. Шаблон кода: <pre>def compare_texts(texts): # TODO: напишите код pass</pre> Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: <pre>def compare_texts(texts): results = [] for i, text in enumerate(texts): words = len(text.split()) unique_words = len(set(text.lower().split())) ratio = unique_words / words if words > 0 else 0 results.append({'index': i, 'uniqueness': ratio}) return sorted(results, key=lambda x: x['uniqueness'])</pre> Вопрос 11 Напишите функцию <code>detect_sql_injection(code)</code>, которая ищет SQL-инъекции в коде.
--	--	--	---	--

	Шаблон кода: <pre>def parse_articles(text): # TODO: напишите код pass</pre> Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: <pre>def parse_articles(text): import re return re.findall(r'ст\.s*(\d+)', text)</pre> Вопрос 12 Напишите функцию <code>extract_domains(urls)</code>, которая извлекает домены из списка URL. Шаблон кода: <pre>def extract_domains(urls): # TODO: напишите код pass</pre> Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: <pre>def extract_domains(urls): domains = [] for url in urls: if '://' in url: domain = url.split('/')[1].split('.')[0]]</pre>	Напишите функцию <code>caesar_encrypt(text, shift)</code>, которая шифрует текст шифром Цезаря. Шаблон кода: <pre>def caesar_encrypt(text, shift): # TODO: напишите код pass</pre> Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: <pre>def caesar_encrypt(text, shift): result = [] for char in text: if char.isalpha(): base = ord('A') if char.isupper() else ord('a') result.append(chr((ord(char) - base + shift) % 26 + base)) else: result.append(char) return ".join(result)</pre> Вопрос 12 Напишите функцию <code>calculate_md5(text)</code>, которая вычисляет MD5 хеш текста. Шаблон кода: <pre>def calculate_md5(text): # TODO: напишите код pass</pre>	<pre>def find_suspicious_ips(log_entries, threshold): # TODO: напишите код pass</pre> Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: <pre>def find_suspicious_ips(log_entries, threshold): ip_counts = {} for entry in log_entries: if 'failed' in entry.lower(): parts = entry.split(',') ip = parts[0].strip() if parts else None if ip: ip_counts[ip] = ip_counts.get(ip, 0) + 1 return [ip for ip, count in ip_counts.items() if count >= threshold]</pre> Вопрос 12 Напишите функцию <code>sanitize_html_input(user_input)</code>, которая экранирует опасные HTML символы в пользовательском вводе. Шаблон кода: <pre>def sanitize_html_input(user_input): # TODO: напишите код pass</pre> Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: <pre>def sanitize_html_input(user_input):</pre>	Шаблон кода: <pre>def detect_sql_injection(code): # TODO: напишите код pass</pre> Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: <pre>def detect_sql_injection(code): dangerous_patterns = ['SELECT * FROM', 'DROP TABLE', 'INSERT INTO', '--', 'OR 1=1'] return [p for p in dangerous_patterns if p in code.upper()]</pre> Вопрос 12 Напишите функцию <code>detect_manipulation(text)</code>, которая находит признаки манипуляции в тексте (срочность, страх, авторитет). Шаблон кода: <pre>def detect_manipulation(text): # TODO: напишите код pass</pre> Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: <pre>def detect_manipulation(text): manipulation_patterns = { 'urgency': ['срочно', 'немедленно', 'сейчас'], 'fear': ['опасность', 'угроза', 'риск'], 'authority': ['эксперт', 'официально', 'гарантировано']</pre>
--	--	--	---	---

domains.append(domain) return domains Вопрос 13 Напишите функцию count_unique_ips(ip_list) , которая подсчитывает уникальные IP. Шаблон кода: def count_unique_ips(ip_list) : # TODO: напишите код pass Поле для вашего кода: _____	Поле для вашего кода: _____	return (user_input .replace('&', '&') .replace('<', '<') .replace('>', '>') .replace('"', '"') .replace("'", ''')) Вопрос 13 Напишите функцию build_safe_query(template, params), которая строит параметризованный SQL запрос из шаблона и словаря параметров. Шаблон кода: def build_safe_query(template, params): # TODO: напишите код pass Поле для вашего кода: _____	} found = [] for category, patterns in manipulation_patterns.items(): if any(p in text.lower() for p in patterns): found.append(category) return found Вопрос 13 Напишите функцию generate_audit_report(findings), которая генерирует отчёт по найденным уязвимостям. Шаблон кода: def generate_audit_report(findings): # TODO: напишите код pass Поле для вашего кода: _____
ЭТАЛОННОЕ РЕШЕНИЕ: def count_unique_ips(ip_list) : return len(set(ip_list)) Вопрос 14 Напишите функцию validate_email(email), которая проверяет корректность email. Шаблон кода: def validate_email(email): # TODO: напишите код pass Поле для вашего кода: _____	ЭТАЛОННОЕ РЕШЕНИЕ: import hashlib def calculate_md5(text): return hashlib.md5(text.encode()). hexdigest() Вопрос 13 Напишите функцию verify_integrity(data, expected_hash), которая проверяет целостность данных. Шаблон кода: def verify_integrity(data, expected_hash): # TODO: напишите код pass Поле для вашего кода: _____	ЭТАЛОННОЕ РЕШЕНИЕ: def build_safe_query(template, params): query = template for key, value in params.items(): query = query.replace(f':{key}', str(value)) return query Вопрос 14 Напишите функцию parse_ctf_flag(text), которая извлекает флаг формата flag {...} из текста. Шаблон кода: def parse_ctf_flag(text): # TODO: напишите код	ЭТАЛОННОЕ РЕШЕНИЕ: def generate_audit_report(findings): report = { 'total_findings': len(findings), 'critical': sum(1 for f in findings if f.get('severity') == 'critical'), 'high': sum(1 for f in findings if f.get('severity') == 'high'), 'medium': sum(1 for f in findings if f.get('severity') == 'medium'), 'recommendations': [f['recommendation'] for f in findings]

	ЭТАЛОННОЕ РЕШЕНИЕ: <pre>def validate_email(email): return '@' in email and len(email) > 5</pre> Вопрос 15 Напишите функцию generate_otp(), которая генерирует 6-значный OTP код. Шаблон кода: <pre>def generate_otp(): # TODO: напишите код</pre> pass Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: <pre>import secrets def generate_otp(): return str(secrets.randbelow(900000) + 100000)</pre>	которая извлекает метаданные из имени файла. Шаблон кода: <pre>def extract_metadata(filename): # TODO: напишите код</pre> pass Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: <pre>def extract_metadata(filename): metadata = {} if '.' in filename: metadata['extension'] = filename.split('.')[-1] metadata['name'] = filename.split('.')[0] metadata['length'] = len(filename) return metadata</pre> Вопрос 15 Напишите функцию generate_session_token(use_r_id), которая генерирует сессионный токен. Шаблон кода: <pre>def generate_session_token(use_r_id): # TODO: напишите код</pre> pass Поле для вашего кода:	pass Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: <pre>import re def parse_ctf_flag(text): match = re.search(r'flag\{([^\}]+\)}', text) return match.group(1) if match else None</pre> Вопрос 15 Напишите функцию analyze_url_patterns(url), которая ищет подозрительные паттерны в URL: IP вместо домена, опечатки в известных доменах, лишние символы @. Шаблон кода: <pre>def analyze_url_patterns(url): # TODO: напишите код</pre> pass Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: <pre>import re def analyze_url_patterns(url): findings = [] if re.search(r'd{1,3}\.d{1,3}\.d{1,3}\.d{1,3}', url): findings.append('ip_instead_of_dom ain')</pre>	<pre>} return report</pre> Вопрос 14 Напишите функцию check_xss_vulnerability(code), которая ищет XSS уязвимости в коде. Шаблон кода: <pre>def check_xss_vulnerability(code): # TODO: напишите код</pre> pass Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ: <pre>def check_xss_vulnerability(code): xss_patterns = ['innerHTML', 'eval(', 'document.write', '<script>'] return [p for p in xss_patterns if p in code]</pre> Вопрос 15 Напишите функцию detect_ai_text(text), которая анализирует текст на признаки генерации ИИ (шаблонность, отсутствие эмоций). Шаблон кода: <pre>def detect_ai_text(text): # TODO: напишите код</pre> pass Поле для вашего кода: ЭТАЛОННОЕ РЕШЕНИЕ:
--	--	---	--	---

		ЭТАЛОННОЕ РЕШЕНИЕ: <pre>import secrets def generate_session_token(user_id):) return f"{user_id}_{token}"</pre>	<pre>known = ['google.com', 'yandex.ru', 'vk.com'] for k in known: typo = k.replace('o', '0').replace('e', '3') if typo in url: findings.append(f'typo_of_{k}') if url.count('@') > 1: findings.append('multiple_at_signs') return findings</pre>	<pre>def detect_ai_text(text): ai_patterns = ['в заключение', 'важно отметить', 'следует учитывать'] score = sum(1 for p in ai_patterns if p in text.lower()) return 'likely_ai' if score >= 2 else 'likely_human'</pre>
Аттестация по итогам модуля. Описание процедуры оценивания Показатели и критерии оценивания	Критерий оценивания: для тестовой части – правильность ответа на тестовый вопрос; для практико-ориентированных заданий – правильность работы программы. Показатели оценивания: для тестовой части – количество правильных ответов на тестовые задания; для практико-ориентированных заданий – количество заданий, в которых программа работает правильно. За каждый правильный ответ/правильно выполненное практико-ориентированное задание начисляется 1 балл. В случае	Критерий оценивания: для тестовой части – правильность ответа на тестовый вопрос; для практико-ориентированных заданий – правильность работы программы. Показатели оценивания: для тестовой части – количество правильных ответов на тестовые задания; для практико-ориентированных заданий – количество заданий, в которых программа работает правильно. За каждый правильный ответ/правильно выполненное практико-ориентированное задание начисляется 1 балл. В случае неправильного ответа/неправильно выполненное практико-	Критерий оценивания: для тестовой части – правильность ответа на тестовый вопрос; для практико-ориентированных заданий – правильность работы программы. Показатели оценивания: для тестовой части – количество правильных ответов на тестовые задания; для практико-ориентированных заданий – количество заданий, в которых программа работает правильно. За каждый правильный ответ/правильно выполненное практико-ориентированное задание начисляется 1 балл. В случае неправильного ответа/неправильно выполненное практико-ориентированное задание баллы не начисляются.	Критерий оценивания: для тестовой части – правильность ответа на тестовый вопрос; для практико-ориентированных заданий – правильность работы программы. Показатели оценивания: для тестовой части – количество правильных ответов на тестовые задания; для практико-ориентированных заданий – количество заданий, в которых программа работает правильно. За каждый правильный ответ/правильно выполненное практико-ориентированное задание начисляется 1 балл. В случае неправильного ответа/неправильно выполненное практико-ориентированное задание баллы не начисляются.

	неправильного ответа/неправильно выполненное практико- ориентированное задание баллы не начисляются.	ориентированное задание баллы не начисляются.		
Аттестация по итогам модуля. Шкала оценивания, нижнее значение	0	0	0	0
Аттестация по итогам модуля. Шкала оценивания, верхнее значение	15	15	15	15
Аттестация по итогам модуля. Шкала оценивания, минимальный проходной балл для успешной сдачи	8	8	8	8

5.3 Итоговый контроль

Итоговый контроль. Количество академических часов	1
Итоговый контроль. Формы контроля	Комплексное задание: тест и защита проекта. Итоговый контроль не входит в 4 модуль (является отдельным элементом ДОП).
Итоговый контроль. Диагностические инструменты	Комплект заданий: тестовые вопросы и процедура защиты проекта
Итоговый контроль. Показатели и критерии оценивания	Показатели и критерии (с весами) оценивания: Этап 1. Тестирование Тестовая часть итогового контроля выполняется и проверяется автоматизированно на Образовательной платформе. Каждый верный ответ оценивается в 1 балл, неверный или отсутствующий — 0 баллов. Максимальный балл 1 этапа – 10 Минимальный пороговый балл – 5 Этап 2. Защита проекта (с презентацией) Критерии и веса внутри проектной части (каждый критерий оценивается членами комиссии по шкале от 0 до 10): 1. Содержательная часть (качество продукта, глубина проработки) - вес 0,7 0 – 4 - Код работает с ошибками или не запускается. Реализована только часть заявленной функциональности (менее 50% функций). Отсутствуют ключевые функции. Нарушены требования безопасности. Код нечитаем: отсутствуют комментарии, осмысленные имена переменных, логическая структура. Не обработаны ошибки и граничные случаи. Описание проекта отсутствует или содержит минимум информации. Проект не демонстрирует понимания темы. 5 – 7 – Код запускается и выполняет основные функции без критических ошибок. Реализованы все минимально необходимые функции для проекта. Соблюдены базовые требования безопасности. Код в целом читаем, но есть замечания к структуре или именованию переменных. Обработаны основные ошибки, но не все граничные случаи. Описание проекта присутствует, содержит описание проекта и примеры использования. Проект демонстрирует базовое понимание темы и способность применить изученные инструменты. Структура проекта логична, но без дополнительных улучшений. 8 – 10 - Код работает без ошибок, все функции выполняют заявленную функциональность корректно. Реализованы все заявленные функции + дополнительные улучшения. Строго соблюдены требования безопасности. Код читаемый: осмысленные имена переменных, логическая структура, отступы, комментарии к функциям и ключевым блокам. Полностью обработаны ошибки и граничные случаи (программа корректно работает при любом вводе). Описание проекта содержит инструкции по запуску, примеры использования. Проект демонстрирует глубокое понимание

	темы, творческий подход и способность применять знания на практике. Структура проекта логична и соответствует лучшим практикам разработки. Обучающийся уверенно отвечает на технические вопросы и обосновывает принятые решения. 2. Визуальное оформление презентации - вес 0,3 0 – 4 - Слайды перегружены текстом; нет единого стиля; шрифты нечитаемы; изображения низкого качества; есть ошибки в оформлении. 5 – 7 – Текст и графика сбалансированы; единый стиль прослеживается; информация структурирована; визуализация данных присутствует. 8 – 10 - Дизайн эстетичный и функциональный; все визуальные элементы работают на раскрытие темы. Максимальный балл 2 этапа – 10 Минимальный пороговый балл – 6 Алгоритм расчета оценки/балла итогового контроля по ДОП Итоговый показатель аттестации рассчитывается в 20-балльной шкале как сумма результатов обоих этапов: Максимально возможное значение: 20 баллов.
Итоговый контроль. Примеры заданий	Вариант 1 1. Тест. Ответьте на тестовые вопросы, выбрав один из предложенных вариантов ответа. В отдельных вопросах нужно будет ввести свой ответ в поле для ответа. Вопрос 1 Как проверить целостность скачанного файла? А) Вычислить хеш файла и сравнить с эталонным хешем Б) Проверить дату создания файла В) Открыть файл в текстовом редакторе Г) Сравнить размер файла с оригиналом Правильный ответ: А) Вычислить хеш файла и сравнить с эталонным хешем Вопрос 2 Какой метод защиты от SQL-инъекций является наиболее надёжным? А) Проверка длины строки Б) Чёрный список SQL-ключевых слов В) Параметризованные запросы Г) Экранирование кавычек вручную Правильный ответ: В) Параметризованные запросы Вопрос 3 Какой метод защиты от brute force наиболее эффективен для коротких PIN-кодов? А) Шифрование базы данных Б) Использование CAPTCHA В) Блокировка после N неудачных попыток Г) Увеличение сложности пароля Правильный ответ: В) Блокировка после N неудачных попыток Вопрос 4 Задание с кратким ответом. Впишите только число.

	Сколько основных ролей в ИБ рассмотрено в курсе (пентестер, защитник, аналитик)? Ответ: _____ Правильный ответ: 3 Вопрос 5 Что такое инцидент информационной безопасности? А) Смена пароля сотрудником Б) Установка нового сервера В) Плановое обновление программного обеспечения Г) Событие, нарушающее конфиденциальность, целостность или доступность информации Правильный ответ: Г) Событие, нарушающее конфиденциальность, целостность или доступность информации Вопрос 6 Задание с кратким ответом. Впишите только число Сколько уровней риска обычно определяет анализатор подозрительных сообщений (низкий, средний, высокий)? Ответ: _____ Правильный ответ: 3 Вопрос 7 В чём главное отличие шифрования от кодирования? А) Кодирование сложнее шифрования Б) Шифрование обратимо, кодирование нет В) Шифрование требует ключ, кодирование — нет Г) Разницы нет Правильный ответ: В) Шифрование требует ключ, кодирование — нет Вопрос 8 Какой федеральный закон регулирует защиту персональных данных в РФ? А) Федеральный закон № 152-ФЗ Б) Федеральный закон № 98-ФЗ В) Федеральный закон № 187-ФЗ Г) Федеральный закон № 149-ФЗ Правильный ответ: А) Федеральный закон № 152-ФЗ Вопрос 9 Какой шаг является первым в стандартной модели кибератаки? А) Достижение цели Б) Проникновение в систему В) Сбор информации (разведка) Г) Закрепление в системе Правильный ответ: В) Сбор информации (разведка) Вопрос 10 Что такое whaling в контексте фишинга? А) Атака на обычных пользователей Б) Массовая рассылка спама В) Атака на серверы компании Г) Целевая атака на топ-менеджеров и VIP-персон
--	---

	Правильный ответ: Г) Целевая атака на топ-менеджеров и VIP-персон 2. Защита проекта. Представьте с использованием презентационных материалов результаты проектной работы на курсе. Полный комплект оценочных материалов (7 вариантов заданий) представлен в документе «Оценочные материалы для проведения итогового контроля» в составе УМК в приложении к ДОП.
Итоговый контроль. Шкала оценивания, нижнее значение	0
Итоговый контроль. Шкала оценивания, верхнее значение	20
Итоговый контроль. Шкала оценивания, минимальный проходной балл	11

6. Организационно-педагогические условия

6.1 Методическое обеспечение: методы, формы и технологии, применяемые при реализации программы

Методы обучения: модульный, проблемный и контекстный подходы.

Формы занятий: лекции с демонстрацией мультимедийных материалов, практические работы, самостоятельная работа.

Технологии: традиционные (объяснительно-иллюстративные) и интерактивные (групповые обсуждения, решение кейсов).

6.2 Материально-техническое обеспечение

Наименование требуемого оборудования и программного обеспечения:

Рабочее место обучающегося: оборудование и мебель, необходимые для обучения, включая расходные материалы.

Рабочее место педагога: оборудование (включая демонстрационное) и мебель.

Иное оборудование и мебель.

Сетевое оборудование и подключение к сети Интернет.

Программное обеспечение.

Средства индивидуальной защиты и охраны труда.

Зонирование площадки.

Компьютер/ноутбук

Windows/Linux (не ранее 2020)

Интернет от 2 Мбит/сек

Веб-браузер (не ранее 2020)

Микрофон, наушники и колонки

Клавиатура, мышь

Проектор/интерактивная доска

7. Учебно-методические материалы

Ниже представлен перечень методических разработок и материалов для обучения:

Методические разработки

Пакет методических разработок содержит:

- Методическую разработку «Материалы для преподавателя по проведению практических занятий».
- Методическую разработку «Материалы для преподавателя по организации самостоятельной работы и самоконтроля/самопроверки».
- Оценочные материалы текущего контроля и промежуточной аттестации.
- Оценочные материалы для проведения итогового контроля по ДОП

Материалы для обучения

Пакет материалов модуля содержит:

- Презентации для теоретических занятий.
- Презентации для практических занятий.
- Описание практических заданий.
- Задания для самоконтроля (самопроверки).
- Задания для самостоятельной работы.
- Задания системы контроля: текущего контроля, промежуточной аттестации.

8. Перечень источников информационного сопровождения (учебная литература и др.)

Учебная литература:

1. Руководство по автоматизации задач ИБ и анализу кода на Python: Костин, В. Н. Методы и средства защиты компьютерной информации. Информационная безопасность компьютерных сетей: учебное пособие / В. Н. Костин. — Москва: Инфра-Инженерия, 2022. — 216 с. — ISBN 978-5-9729-0831-8.
2. Учебник по правовым основам и базовым понятиям ИБ: Внуков, А. А. Основы информационной безопасности: защита информации: учебное пособие для СПО / А. А. Внуков. — 2-е изд., перераб. и доп. — Москва: Юрайт, 2021. — 161 с. — (Профессиональное образование). — ISBN 978-5-534-13554-1.
3. Практическое пособие по криптографическим методам и хеш-функциям: Рацеев, С. М. Криптографические методы защиты информации и их основы: лабораторный практикум: учебное пособие для СПО / С. М. Рацеев. — Санкт-Петербург: Лань, 2022. — 212 с. — (Среднее профессиональное образование). — ISBN 978-5-8114-9344-0.
4. Лутц, М. Python / М. Лутц. — 5-е изд. — Москва: Диалектика, 2020. — 320 с. — ISBN 978-5-907114-60-9.

Электронные информационные ресурсы:

1. Образовательная платформа по Python <https://pythonist.ru/>
2. pythonturbo обучение разработке на Python <https://pythonturbo.ru/category/basics/>

Директор Центра компетенций
«Цифровая экономика»



Н.Ю. Сурова