ЦЕНТРАЛЬНЫЙ УНИВЕРСИТЕТ

УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«24» июня 2025 г.
Протокол № 2

Рабочая программа дисциплины (модуля)
«Информационная безопасность»

Направление подготовки: *02.04.01 Математика и компьютерные науки*

Направленность (профиль) подготовки: *Машинное обучение*

Квалификация (степень) выпускника: *магистр*

Форма обучения: *очная*

Срок освоения программы: *2 года*

Год набора: *2025*

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 0215B0C000D1B3FEB44A37EE631E0C27D4
Владелец: Ивашкевич Евгений Васильевич
Действителен: с 13.01.2026 до 13.04.2027

Электронный документ

Москва
2025

Содержание

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Информационная безопасность» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – магистратура по специальности 02.04.01 Математика и компьютерные науки, профиль Машинное обучение, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 810 от 23.08.2017 года.

Изучение дисциплины (модуля) «Информационная безопасность» дает знания в области информационной безопасности становятся критически важными для защиты личной и корпоративной информации. Кроме того, осознание принципов информационной безопасности способствует созданию безопасной цифровой среды, что является необходимым условием для устойчивого развития бизнеса и общества в целом.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки магистратуры по направлению 02.04.01 Математика и компьютерные науки, профиль Машинное обучение и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений, как дисциплина по выбору.

Дисциплина (модуль) изучается на 2 курсе в 4 семестре, доступна для прохождения при условии успешного завершения дисциплин (модулей) «Алгоритмы и структуры данных. Часть 2», «Промышленная разработка».

Цель изучения дисциплины (модуля): формирование у студентов знаний и навыков, необходимых для защиты информации и информационных систем от угроз, рисков и атак.

Задачи изучения дисциплины (модуля):

- изучить методы выявления и анализа угроз безопасности в программных продуктах;
- освоить применение криптографических технологий для защиты данных и аутентификации;
- развить умение разрабатывать безопасное программное обеспечение с учетом современных требований;
- научиться оценивать риски и выбирать эффективные стратегии защиты информации;
- приобрести навыки использования специализированных инструментов для тестирования и повышения безопасности приложений.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

- основные принципы и методы обеспечения информационной безопасности программных систем;
- современные криптографические алгоритмы и их применение в разработке;
- распространённые типы уязвимостей и угроз для веб-приложений и системного уровня.

уметь:

- обнаруживать и анализировать типовые уязвимости в веб-приложениях и системах;
- формулировать и реализовывать базовые меры защиты информации на уровне приложения и системы;
- оценивать риски, связанные с реализацией функциональности, и делать выбор в пользу безопасных решений.

владеть:

- навыками разработки программного обеспечения с учётом требований безопасности;

— навыками применения криптографических механизмов (шифрования, хеширования, цифровой подписи) в программных решениях;
— навыками использования инструментов для оценки безопасности и тестирования приложений.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
УК-6.	Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки	УК-6.1.	Знает основные методы самооценки и анализа своей деятельности, а также принципы управления временем и целеполагания
		УК-6.2.	Умеет ставить реалистичные и достижимые цели, определять приоритеты в своей деятельности, а также разрабатывать и внедрять планы по совершенствованию своих навыков и компетенций на основе полученной самооценки
		УК-6.3.	Имеет практический опыт применения методов самооценки в своей профессиональной деятельности, включая участие в тренингах, семинарах и проектах, направленных на развитие личной эффективности и профессионального роста
ОПК-2.	Способен создавать и исследовать новые математические модели в естественных науках, совершенствовать и разрабатывать концепции, теории и методы	ОПК-2.1.	Знает основные математические модели и методы, используемые в естественных науках, включая статистическое моделирование, дифференциальные уравнения и численные методы, а также современные подходы к исследованию и анализу данных
		ОПК-2.2.	Умеет разрабатывать и адаптировать математические модели для решения конкретных проблем в естественных науках, проводить их анализ и верификацию, а также интерпретировать полученные результаты в контексте научных исследований
		ОПК-2.3.	Имеет практический опыт создания и исследования математических моделей в рамках научных проектов или

			исследований, включая участие в публикациях, конференциях или коллаборациях, где были разработаны и апробированы новые концепции и методы
ПК-3.	Способен решать задачи профессиональной деятельности, формулировать результат, увидеть следствия полученного результата	ПК-3.1.	Знает основные принципы и методы решения задач профессиональной деятельности, а также способы формулирования и представления результатов, включая анализ последствий и их значимость в контексте проекта
		ПК-3.2.	Умеет применять математические и компьютерные методы для решения конкретных задач, формулировать четкие и обоснованные результаты, а также анализировать их последствия для дальнейших действий и решений
		ПК-3.3.	Имеет практический опыт в решении профессиональных задач, включая участие в проектах, где были получены результаты и проанализированы их следствия, что способствовало принятию обоснованных решений
ПК-4.	Способен публично представлять собственные и известные научные результаты	ПК-4.1.	Знает основные принципы эффективного публичного выступления, методы визуализации данных и основные требования к научным презентациям, включая структуру и содержание
		ПК-4.2.	Умеет четко и логично формулировать свои научные результаты, адаптируя их для различных аудиторий, а также использовать визуальные средства для улучшения восприятия информации
		ПК-4.3.	Имеет практический опыт участия в научных конференциях, семинарах или других мероприятиях, где успешно представлял свои и известные научные результаты, получая обратную связь и взаимодействуя с аудиторией

3. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы					ТКУ (текущий контроль успеваемости)
		Очная форма					
		Аудиторная работа			Контроль	Самостоятельная работа	
		Лекции	Семинары (практические занятия)	Консультации			
1	Основные криптографические алгоритмы	4	4	8		13	Домашние задания Пентесты
2	Безопасность на уровне операционной системы	4	4	8		14	Домашние задания Пентесты
3	Безопасность веб-приложений	3	3	6		9	Домашние задания
4	Организационные аспекты информационной безопасности	4	4	8		14	Домашние задания Пентесты
	Зачет				4		
	Итого:	***15***	***15***	***30***	***4***	***50***	
Объем дисциплины (модуля) (в ак. ч.)		***114***					
Объем дисциплины (модуля) (в зач. ед.)		***3***					

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Основные криптографические алгоритмы	Основы криптографии. Генератор псевдослучайных чисел. Метод оценки эффективности ГПСЧ Сеть Фейстеля. Хэши. Коллизии хэшей и радужные таблицы Блочные шифры. AES как пример блочного шифра. Ассиметричное шифрование и цифровая подпись. Алгоритм RSA, Diffie-Helman.
2	Безопасность на уровне операционной системы	Управление доступом в Linux. Пользователи и их права. DAC модель, ACL PAM как средство авторизации в систему. Механизм TOTP. Настройка аудита Изоляция в Linux. CGroups, Namespaces, Capabilities. Бинарные уязвимости и эксплуатация libc Безопасность сетевого стека Linux. Интерфейсы, туннели, firewall, теория PKI и работы сертификатов
3	Безопасность веб-приложений	Теория и практика Web уязвимостей. SQL injection, CSRF, path traversial, XSS Теория и практика Web уязвимостей. RCE, LFI, SSTI, IDOR, SSRF, DoS и Race Conditions
4	Организационные аспекты информационной безопасности	Протокол авторизации и аутентификации. Рассматриваем варианты на основе LDAP и KeyCloak. OIDC и OpenID Spring Security Инфраструктура хранения секретов. Vault Основы контейнерной безопасности. Container Escapes

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Безопасность веб-приложений. Разведка, защита, нападение - СПб:Питер, 2022: ISBN. - Текст : электронный. - URL: https://znanium.com/catalog/product/2125433.

2. Фленов, М. Е. Linux глазами хакера : практическое руководство / М. Е. Фленов. - 6-е изд., перераб. и доп. - Санкт-Петербург : БХВ-Петербург, 2021. - 416 с. - ISBN 978-5-9775-6699-5. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2139146.

Дополнительная литература:

1. Скабцов, Н. Аудит безопасности информационных систем : практическое руководство / Н. Скабцов. - Санкт-Петербург : Питер, 2018. - 272 с. - (Серия «Библиотека программиста»). - ISBN 978-5-4461-0662-2. - Текст : электронный. - URL: https://znanium.com/catalog/product/1760857.

2. Лось, А. Б. Криптографические методы защиты информации для изучающих компьютерную безопасность : учебник для вузов / А. Б. Лось, А. Ю. Нестеренко, М. И. Рожков. — 2-е изд., испр. — Москва : Издательство Юрайт, 2025. — 424 с. — (Высшее образование). — ISBN 978-5-534-12474-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/560426.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

— столами и стульями;

— компьютерной техникой;

— механическими калькуляторами;

— специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и

обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование"	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Распределенные системы:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		

Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Информационная безопасность» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, консультации, домашние задания, пентесты, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удается, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Семинар — это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

Консультации – структурированные встречи, на которых преподаватели предоставляют индивидуальную или групповую помощь в освоении учебного материала, обсуждении вопросов и решении проблем, возникающих в процессе обучения.

Консультации могут включать разъяснение сложных тем, подготовку к экзаменам и помощь в выполнении проектных работ, что способствует более глубокому пониманию предмета и улучшению академической успеваемости.

Домашнее задание – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал. Использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

Пентесты (penetration testing) – это практика проверки приложений или систем на уязвимости путём имитации действий злоумышленника. Это контролируемая "взлом-сессия", цель которой не разрушить, а найти слабые места, прежде чем их найдут настоящие атакующие.

Бонусные баллы — это оценки, которые студенты могут получить за выполнение дополнительных заданий.

Формат бонусных баллов позволяет студентам улучшить общую оценку по дисциплине (модулю) и стимулирует углубленное изучение материала.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Информационная безопасность»

Оценивание уровня учебных достижений обучающихся по дисциплине (модулю) осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме ***зачета***, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом:

Десятибалльная оценка	**Пятибалльная оценка**	**Оценка за зачет**	**Общая характеристика результата обучения по дисциплине (модулю)**
10	Отлично	Зачтено	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину (модуль). Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других
9	Отлично	Зачтено	
8	Отлично	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
7	Хорошо	Зачтено	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
6	Хорошо	Зачтено	
5	Удовлетворительно	Зачтено	Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	Зачтено	
3	Не сдан	Не зачтено	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	Не зачтено	
1	Не сдан	Не зачтено	

Дисциплина (модуль) «Информационная безопасность» оценивается следующим образом:

Активность	Вес	Описание
Домашние задания	40%	За каждое из заданий можно набрать 10 баллов

Электронный документ

Активность	Вес	Описание
Пентесты	30%	Практика проверки приложений или систем на уязвимости путём имитации действий злоумышленника
Зачет	30%	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по дисциплине (модулю)

В рамках изучения дисциплины (модуля) возможно получение бонусных баллов.

Формула расчёта итоговой оценки по дисциплине (модулю) «Информационная безопасность»: «0,4 × среднее за домашние задания + 0,3 × среднее за пентесты + 0,3 × зачет».

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание

ОБЩАЯ ЧАСТЬ

1. Соберите и запустите OWASP Juice-Shop (https://github.com/juice-shop/juice-shop?ysclid=mi7fqa34a4717354142) локально – это специальная песочница, в которой можно обкатывать те или иные побеги/эксплуатации уязвимостей
2. Прогоните собранный образ через Trivy, соберите репорт о найденных уязвимостях в зависимостях
3. Поднимите DefectDojo, настройте выгрузку скана
4. Разберите CRITICAL уязвимости в зависимостях, продемонстрируйте атаки на минимум 1 уязвимость, найденную TRIVY. Изучите исходный код зависимостей, посмотрите, почему оно возникает. Предложите способ митигации
5. Выберите любой open-source на ваш выбор сканер исходного кода на уязвимости. Прогоните его на коде juice-shop, результат загоните в Dojo. Изучите вывод, найдите минимум 2 уязвимости, на которых будете тренироваться. Проверьте выполняемость вектора атаки, приведите способ митигации (устранения)

DEFENCE БЛОК

6. Склонируйте репо с кодом к себе в Gitlab/Github
7. Настройте DevSecOps пайплайн. Для этого на базе GitlabCI и локального раннера (если вы не разворачивали DD на сервере с белым IP) или Jenkins (потребуется локально поднять) напишите пайплайн, который при каждом новом коммите/MR в main-ветку будет запускать сканирования и завозить результаты в Trivy. Подумайте, как избежать дублирования CVE. Инструменты CI/CD и их выбор для задачи – ваша зона ответственности

ATTACK БЛОК

6. Решить задачу - http://158.160.84.161:5050 и найти флаг
7. Тут Blind SQL
8. Тут определенно WAF
9. Тут точно недостаточно одних лишь рук, нужно скриптить
10.

```
CREATE TABLE IF NOT EXISTS users (
    id INTEGER PRIMARY KEY,
    username TEXT NOT NULL,
    email TEXT NOT NULL
)


CREATE TABLE IF NOT EXISTS secrets (
    id INTEGER PRIMARY KEY,
    secret_key TEXT NOT NULL,
    secret_value TEXT NOT NULL
)
```

Результат прикладывать отчетом PDF с описанием и картинками, что и как делали

Домашнее задание

Цель задания

Понять и реализовать ключевые механизмы контейнеризации на уровне операционной системы Linux, создав изолированную среду для произвольного процесса.

Задача

Написать программу на любом языке программирования (C, Python, Go, Rust и т.д.), которая:

1. **Запускает произвольный процесс**
2. **Запускает его во всех неймспейсах Linux**
3. **Ограничивает capabilities процесса**
4. **Применяет cgroups для ограничения ресурсов**
5. **Настраивает seccomp фильтр**
6. **Применяет AppArmor профиль**

Подробные инструкции

1. Запуск произвольного процесса

Что это значит: Ваша программа должна запускать другой процесс (например, `/bin/bash` или `/bin/ls`).

Как сделать:

- Используйте системный вызов `fork()` для создания дочернего процесса
- В дочернем процессе используйте `execve()` для запуска целевой программы

Пример на C:

c

```c
pid_t pid = fork();
if (pid == 0) {
    char *args[] = {"/bin/bash", NULL};
    execve(args[0], args, NULL);
    perror("execve failed");
    exit(1);

}
```

2. Запуск во всех неймспейсах Linux

Что это значит: Неймспейсы изолируют различные аспекты системы (сеть, процессы, файловая система и т.д.).

Основные неймспейсы:

- `CLONE_NEWPID` - изоляция дерева процессов
- `CLONE_NEWNET` - изоляция сети
- `CLONE_NEWNS` - изоляция файловой системы
- `CLONE_NEWUTS` - изоляция hostname
- `CLONE_NEWIPC` - изоляция IPC
- `CLONE_NEWUSER` - изоляция пользователей
- `CLONE_NEWCGROUP` - изоляция cgroups

Как сделать:

- Используйте флаг `CLONE_NEW*` при вызове `clone()` или `unshare()`
- Для применения ко всем неймспейсам используйте комбинацию всех флагов

Пример на C:

c

```c
#define _GNU_SOURCE
#include <sched.h>
#include <sys/wait.h>

int main() {
   pid_t pid = clone(child_func, child_stack + STACK_SIZE,
               CLONE_NEWPID | CLONE_NEWNET | CLONE_NEWNS |
               CLONE_NEWUTS | CLONE_NEWIPC | CLONE_NEWUSER |
               CLONE_NEWCGROUP | SIGCHLD, NULL);

   waitpid(pid, NULL, 0);

   return 0;
}

int child_func(void *arg) {
   system("/bin/ls");
   return 0;

}
```

3. Ограничение capabilities

Что это значит: Capabilities - это разрешения, которые определяют, что процесс может делать в системе.

Основные capabilities:

- `CAP_NET_RAW` - возможность использовать RAW сокеты
- `CAP_SYS_ADMIN` - административные привилегии
- `CAP_DAC_OVERRIDE` - обход проверок прав доступа

Как сделать:

- Используйте системные вызовы `capset()` и `capget()`
- Или утилиту `libcap` для более простой работы

Пример на C с libcap:

c

```c
#include <sys/capability.h>

void drop_capabilities() {
   cap_t caps = cap_init();

   // SET CAP_NET_RAW
   cap_value_t cap_list[] = {CAP_NET_RAW};
   cap_set_flag(caps, CAP_PERMITTED, 1, cap_list, CAP_SET);
   cap_set_flag(caps, CAP_EFFECTIVE, 1, cap_list, CAP_SET);

   cap_set_proc(caps);
   cap_free(caps);

}
```

4. Применение cgroups

Что это значит: Control groups (cgroups) ограничивают использование ресурсов (CPU, память, I/O).

Как сделать:

- Создайте директорию в `/sys/fs/cgroup/`
- Настройте лимиты через файлы в этой директории
- Добавьте PID процесса в `cgroup.procs`

Пример через командную строку:

```
bash
sudo mkdir /sys/fs/cgroup/memory/my_container
sudo mkdir /sys/fs/cgroup/cpu/my_container


echo "100000000" | sudo tee
/sys/fs/cgroup/memory/my_container/memory.limit_in_bytes
echo "50000" | sudo tee /sys/fs/cgroup/cpu/my_container/cpu.cfs_quota_us
echo "100000" | sudo tee /sys/fs/cgroup/cpu/my_container/cpu.cfs_period_us

# Добавляем процесс
echo $PID | sudo tee /sys/fs/cgroup/memory/my_container/cgroup.procs

echo $PID | sudo tee /sys/fs/cgroup/cpu/my_container/cgroup.procs
```

Пример на Python:

```
python
import os

def setup_cgroups(pid):
    # Создаем cgroup для памяти
    os.makedirs('/sys/fs/cgroup/memory/my_container', exist_ok=True)

    # Устанавливаем лимит памяти 100MB
    with open('/sys/fs/cgroup/memory/my_container/memory.limit_in_bytes', 'w') as f:
        f.write('100000000')

    # Добавляем процесс
    with open('/sys/fs/cgroup/memory/my_container/cgroup.procs', 'w') as f:

        f.write(str(pid))
```

5. Настройка seccomp фильтра

Что это значит: Seccomp фильтрует системные вызовы, которые может выполнять процесс.

Как сделать:

- Используйте `prctl()` с `PR_SET_SECCOMP`
- Или `seccomp()` системный вызов
- Определите, какие системные вызовы разрешить

Пример на C:

```c
#include <seccomp.h>
#include <sys/prctl.h>

void setup_seccomp() {
    scmp_filter_ctx ctx;

    // Инициализируем контекст seccomp
    ctx = seccomp_init(SCMP_ACT_KILL); // Убиваем процесс при нарушении

    // Разрешаем базовые системные вызовы
    seccomp_rule_add(ctx, SCMP_ACT_ALLOW, SCMP_SYS(read), 0);
    seccomp_rule_add(ctx, SCMP_ACT_ALLOW, SCMP_SYS(write), 0);
    seccomp_rule_add(ctx, SCMP_ACT_ALLOW, SCMP_SYS(exit), 0);
    seccomp_rule_add(ctx, SCMP_ACT_ALLOW, SCMP_SYS(exit_group), 0);

    // Запрещаем опасные вызовы
    seccomp_rule_add(ctx, SCMP_ACT_KILL, SCMP_SYS(clone), 0);
    seccomp_rule_add(ctx, SCMP_ACT_KILL, SCMP_SYS(fork), 0);
    seccomp_rule_add(ctx, SCMP_ACT_KILL, SCMP_SYS(kill), 0);

    // Применяем фильтр
    seccomp_load(ctx);
    seccomp_release(ctx);

}
```

6. Применение AppArmor профиля

Что это значит: AppArmor - система mandatory access control, ограничивающая доступ к файлам и ресурсам.

Как сделать:

- Создайте профиль AppArmor
- Загрузите его в ядро
- Назначьте профиль процессу

Шаги:

1. **Создайте профиль** в `/etc/apparmor.d/containername`:

text
```
#include <tunables/global>

profile containername flags=(attach_disconnected,mediate_deleted) {
 #include <abstractions/base>

 # Разрешаем базовые операции
 /bin/bash ix,
 /bin/ls ix,
 /etc/ld.so.cache r,

 # Запрещаем запись в системные директории
 deny /etc/** w,
 deny /usr/** w,
 deny /var/** w,

}
```

2. **Загрузите профиль:**

bash
```
sudo apparmor_parser -r /etc/apparmor.d/containername
```

3. **Назначьте профиль процессу** в коде:

c
```
#include <sys/apparmor.h>

void apply_apparmor_profile() {
   char *profile = "containername";
   if (aa_change_profile(profile) < 0) {
      perror("Failed to apply AppArmor profile");

   }

}
```

Примерные описания заданий для пентестов

1. **Пентест на слабость криптографических реализаций:** Студент получает доступ к учебному приложению, использующему AES-шифрование для хранения данных. Задача: имитировать атаку на слабый ключ или уязвимость в генераторе псевдослучайных чисел (например, предсказуемость на основе радужных таблиц), расшифровать данные и предложить улучшения (например, использование более сильных ключей или RSA для цифровой подписи). Критерии: успешная декодировка, анализ коллизий хэшей, отчет с рекомендациями.

2. **Пентест на управление доступом и изоляцию в Linux:** Студент получает виртуальную машину с Linux, где настроены DAC/ACL и CGroups. Задача: имитировать эксплуатацию бинарных уязвимостей (например, через libc) для повышения привилегий, обхода namespaces или capabilities, а также проверить настройки PAM и TOTP. Критерии: обнаружение уязвимостей в аудите, демонстрация успешной эскалации прав, предложения по усилению firewall и PKI-сертификатов.

3. **Пентест на сетевую безопасность Linux:** Студент работает с сетевым стеком Linux (интерфейсы, туннели). Задача: имитировать атаку на firewall или сертификаты, используя уязвимости в сетевых интерфейсах (например, подмена сертификатов), и проверить изоляцию через namespaces. Критерии: успешная эксплуатация, анализ логов аудита, рекомендации по настройке туннелей и PKI.

4. **Пентест на инъекции и XSS:** Студент получает учебное веб-приложение. Задача: имитировать SQL injection, XSS или CSRF для извлечения данных или выполнения несанкционированных действий, а также проверить на path traversal и RCE. Критерии: успешная эксплуатация уязвимостей, демонстрация LFI/SSTI/IDOR, отчет с мерами защиты (например, фильтрация вводов).

5. **Пентест на DoS и SSRF:** Студент тестирует веб-приложение на уязвимости типа DoS, Race Conditions или SSRF. Задача: имитировать атаки для вызова отказов в обслуживании или доступа к внутренним ресурсам, анализируя сетевые взаимодействия. Критерии: воспроизведение атак, оценка рисков, предложения по ограничению запросов.

6. **Пентест на аутентификацию и авторизацию:** Студент получает систему с LDAP/KeyCloak или Spring Security. Задача: имитировать обход OIDC/OpenID или эксплуатацию уязвимостей в протоколе авторизации (например, подмена токенов), а также проверить инфраструктуру секретов в Vault. Критерии: успешная несанкционированная аутентификация, анализ рисков, рекомендации по усилению.

7. **Пентест на контейнерную безопасность:** Студент работает с контейнерами (Docker/Kubernetes). Задача: имитировать container escape через уязвимости в изоляции или секретах, проверяя управление доступом и хранение данных в Vault. Критерии: демонстрация выхода из контейнера, оценка рисков, предложения по настройке безопасности (например, ограничение capabilities).

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Какой из перечисленных алгоритмов относится к симметричному шифрованию? а) RSA б) AES в) Диффи-Хеллман г) Эллиптические кривые	б	ОПК-2
2.	Что из перечисленного является основным механизмом контроля целостности в ОС? а) Журналирование б) Аутентификация	а	ПК-3

	в) Контроль доступа на основе ролей (RBAC) г) Хэш-функции		
3.	Какой метод наиболее эффективен для защиты веб-приложения от CSRF-атак? а) Использование HTTPS б) Внедрение токенов CSRF в) Использование сложных паролей г) Ограничение доступа по IP	б	ПК-4
4.	Назовите алгоритм асимметричного шифрования, основанный на свойствах эллиптических кривых.	Эллиптические кривые (ECC)	ОПК-2
5.	Как называется протокол, позволяющий двум сторонам безопасно обмениваться ключами через незащищённый канал?	Диффи-Хеллман	ПК-3
6.	Какой элемент безопасности ОС отвечает за проверку личности пользователя?	Аутентификация	УК-6
7.	Как называется процесс управления временем жизни пользователя в веб-приложении после входа в систему?	Управление сессиями	ПК-3
8.	Как называется документ, регламентирующий требования и правила по информационной безопасности в организации?	Политика безопасности	УК-6
9.	Назовите инструмент для хранения секретов, важный для самооценки инфраструктуры безопасности.	Vault	УК-6
10.	Назовите сеть, используемую в блочных шифрах.	Фейстель / Feistel	ОПК-2
11.	Назовите метод для взлома хэшей с использованием радужных таблиц.	радужные таблицы / rainbow tables	ПК-3
12.	Назовите модель управления доступом в Linux.	DAC / discretionary access control	ПК-4
13.	Назовите механизм авторизации в Linux.	PAM	УК-6
14.	Назовите технологию изоляции в Linux.	Namespaces / namespaces	ОПК-2
15.	Назовите тип уязвимостей, связанных с libc.	бинарные уязвимости / binary vulnerabilities	ПК-3
16.	Назовите тип веб-уязвимости для инъекции SQL.	SQL injection	ПК-4
17.	Назовите тип веб-уязвимости для выполнения кода.	RCE / remote code execution	УК-6
18.	Назовите тип веб-уязвимости для обхода каталогов.	path traversal	ОПК-2
19.	Назовите протокол для аутентификации.	OIDC / OpenID Connect	ПК-3
20.	Назовите фреймворк для безопасности приложений.	Spring Security	ПК-4