ФИНАНСОВЫЙ УНИВЕРСИТЕТ
Учебно-методическое пособие

ФИНАНСОВЫЙ УНИВЕРСИТЕТ
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

ОСНОВЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

для направления «Программная инженерия»

ПРАКТИЧЕСКИЙ УЧЕБНЫЙ МАРШРУТ

Краткий пересказ учебной программы → понятные объяснения
→ примеры из разработки ПО → практические задания → Roadmap подготовки

МАРШРУТ ОБУЧЕНИЯ



144 часа | 5 тематических блоков | 1 итоговый проект

Учебный маршрут по дисциплине «Основы информационной безопасности»

ФИНАНСОВЫЙ УНИВЕРСИТЕТ
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

КАК ПОЛЬЗОВАТЬСЯ ЭТИМ PDF

Это не учебник, который нужно читать подряд.
Это маршрут: теория → пример → практика → самопроверка.

01 ЧТО НУЖНО ПОНЯТЬ

Краткая теория без академических формулировок.

Сначала поймите, о какой проблеме информационной безопасности идёт речь.

02 ПРОСТЫМИ СЛОВАМИ

Перевод терминологии рабочей программы на язык разработчика.

Определение → понятное объяснение → связь с программной системой.

03 ПРИМЕР

Конкретная ситуация из разработки ПО.

Увидьте, как теоретическое понятие проявляется в реальном сценарии.

04 ЧТО ДОЛЖЕН УМЕТЬ РАЗРАБОТЧИК

Практический результат обучения.

После блока должно быть понятно, какое действие вы умеете выполнить.

05 САМОПРОВЕРКА

3–5 вопросов после каждого раздела.

Отвечайте без подсказки. Если не получается — вернитесь к теории.

06 ПРАКТИКА

Небольшое задание на применение материала.

Переход от «знаю термин» к «умею применить его к системе».

07 ВРЕМЯ

Для каждого блока указана рекомендуемая нагрузка.

Планируйте обучение заранее и отмечайте фактически затраченное время.

ПРИМЕР УЧЕБНОГО БЛОКА

Изучение: 2 часа

Теория	Пример	Практика	Самопроверка
50 мин	20 мин	40 мин	10 мин

ПОНЯТЬ → ОБЪЯСНИТЬ → ПРИМЕНИТЬ → ПРОВЕРИТЬ

ROADMAP

ОСНОВЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ • ПРОГРАММНАЯ ИНЖЕНЕРИЯ

Последовательность тем подробного учебного материала

01	**Основы информационной безопасности** Понятия, государственная политика, нормативная база, жизненный цикл ПО	24 ч
02	**Уязвимости и угрозы** Активы, уязвимости, угрозы и практическая связь между ними	28 ч
03	**Триада CIA** Конфиденциальность, целостность и доступность	4 ч
04	**Модель угроз** Активы, нарушители, сценарии, последствия и меры защиты	6 ч
05	**База угроз ФСТЭК** Систематизация угроз при анализе информационной системы	3 ч
06	**Современные технологии и информационные риски** Доступ, атаки, веб-приложения и средства защиты	30 ч
07	**Человек и информационная безопасность** Политики, человеческий фактор, социальная инженерия и практика	30 ч
08	**Организация защиты информационной системы** Технические, организационные и правовые меры	32 ч
09	**Непрерывность бизнеса** Резервирование, восстановление и планы действий	5 ч
10	**Инциденты информационной безопасности** Обнаружение, реагирование и фиксация инцидентов	6 ч
11	**Безопасность программного обеспечения** Безопасность на этапах разработки и эксплуатации	8 ч
12	**Криптография** Шифрование, хеширование и электронная подпись	6 ч
13	**Итоговый практический проект** Анализ безопасности небольшого веб-приложения	10 ч

1. Блок №1. Что такое информационная безопасность

Время: 24 часа • Базовый блок: терминология, государственная политика, нормативная база и место ИБ в жизненном цикле ПО.

Главная мысль для разработчика

Информационная безопасность — это не только антивирус. Для программиста это способность спроектировать и поддерживать систему так, чтобы данные и функции не были неправомерно раскрыты, изменены, уничтожены или сделаны недоступными.

1.1. Базовые понятия — 3 ч

Изучить: информация; информационная безопасность; национальная безопасность; информационные ресурсы; информационная система; защита информации.

Простыми словами. Сначала нужно понять объект защиты. В программной системе это не только база данных. Защищать приходится аккаунты, API, серверы, ключи, журналы, документы и сами функции приложения.

Что искать в системе

- данные пользователей и сотрудников
- учётные записи и права доступа
- API и административные функции
- базы данных и резервные копии

Что может быть целью защиты

- раскрытие данных
- несанкционированное изменение
- удаление или потеря данных
- остановка сервиса

Пример: интернет-магазин

Пользователь → Web-приложение → API → База данных. В каждой точке нужно задать вопрос: какие данные проходят здесь, кто может ими пользоваться и что произойдёт, если доступ получит посторонний?

1.2. Информационная безопасность государства — 4 ч

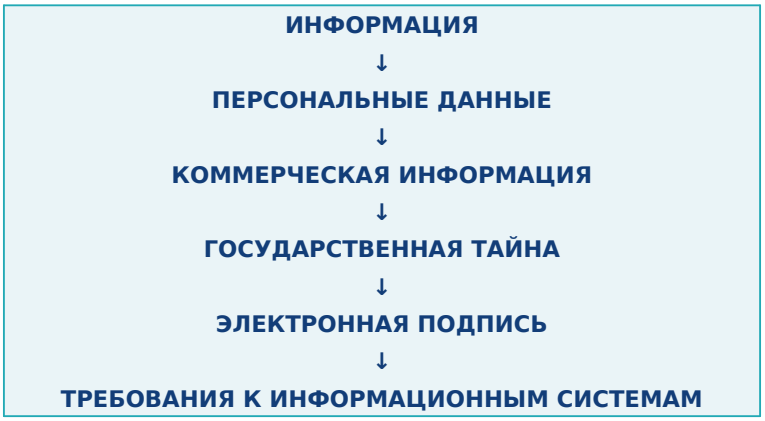
Изучить: государственные органы; национальные интересы; информационный суверенитет; основные направления государственной политики; нормативные документы.

Как читать эту тему программисту

Не нужно превращать раздел в изучение государственного управления. Практический вопрос звучит так: какие требования государства могут влиять на создаваемую мной информационную систему?

1.3. Нормативная база — 7 ч

Сначала построить карту объектов, а уже затем разбирать требования и документы, которые относятся к ним.



1.4. Информационная безопасность в жизненном цикле ПО — 4 ч

ТРЕБОВАНИЯ ↓ ПРОЕКТИРОВАНИЕ ↓ РАЗРАБОТКА ↓ ТЕСТИРОВАНИЕ	**РАЗВЁРТЫВАНИЕ ↓ ЭКСПЛУАТАЦИЯ ↓ ОБНОВЛЕНИЕ**

На каждом этапе задайте один вопрос: какая проблема безопасности может возникнуть здесь? Например, на этапе требований — какие данные надо защищать; при разработке — можно ли обойти проверку; при эксплуатации — как обнаружить инцидент.

1.5. Практика — 4 ч

Задание: интернет-магазин

- ☐ Определить, какие данные и компоненты защищаются.
- ☐ Определить, от кого или от чего их нужно защищать.
- ☐ Указать этап жизненного цикла, где возникает риск.
- ☐ Предложить способ защиты.

1.6. Самопроверка — 2 ч

Вопросы

- ☐ Что является активом информационной системы?
- ☐ Чем защита информации отличается от установки одного средства защиты?
- ☐ Почему безопасность должна учитываться до написания кода?
- ☐ Какие данные интернет-магазина требуют защиты?
- ☐ На каком этапе жизненного цикла может возникнуть ошибка безопасности?

2. Блок №2. Уязвимости и угрозы

Время: 28 часов • Один из ключевых разделов для разработчика: активы, уязвимости, угрозы и их связь.

2.1. Что такое актив — 2 ч

Актив

Актив — то, что имеет ценность для системы и требует защиты. Например: база данных пользователей, учётная запись администратора, API, сервер, ключ подписи.

2.2. Уязвимость — 4 ч

Простая формула: уязвимость = слабое место системы, которое может быть использовано для реализации угрозы.

Пароль хранится в открытом виде → **Уязвимость** → **Получение доступа к БД** → **Компрометация**

2.3. Угроза — 4 ч

Разобрать: антропогенные; техногенные; внутренние; внешние; преднамеренные; случайные угрозы.

Внешние / преднамеренные

- фишинг
- кража учётных данных
- атака на веб-приложение
- вредоносное ПО

Внутренние / случайные

- ошибка сотрудника
- неправильная настройка
- утрата носителя
- сбой оборудования

Связка, которую нужно запомнить

Актив отвечает на вопрос «что защищаем?». Уязвимость — «где слабое место?». Угроза — «что может использовать это слабое место?». Риск — «какие последствия и насколько они значимы?»

2.4. Практическая связка — 18 ч

Рабочая схема анализа

☐ Составить перечень активов учебной системы.

☐ Для каждого актива найти минимум одну уязвимость.

☐ Определить угрозу, которая может использовать уязвимость.

☐ Описать возможное последствие.

☐ Предложить защитную меру и способ проверки.

3. Триада CIA

Время: 4 часа • Визуальный каркас для оценки последствий нарушения безопасности информации.

Информация доступна только тем, кому она разрешена.	Данные не изменены неправомерно.	Система и данные доступны, когда это необходимо.

Конфиденциальность

- Хакер получил базу пользователей.
- Пароль или персональные данные раскрыты.
- Доступ получили не те лица.

Целостность

- Изменён баланс счёта.
- Подменён заказ.
- Изменены записи или настройки.

Доступность

Пример: сервер интернет-магазина перестал отвечать. Даже если данные не украдены и не изменены, система не выполняет свою функцию — нарушена доступность.

Задание

☐ Для пяти сценариев определить: конфиденциальность, целостность или доступность.

☐ Если нарушено несколько свойств — указать все.

☐ Для каждого сценария назвать одну защитную меру.

4. Модель угроз

Время: 6 часов • Простой алгоритм, который превращает общий разговор об угрозах в конкретный анализ.

1. ЧТО ЗАЩИЩАЕМ?	2. ОТ КОГО?	3. ЧТО МОЖЕТ ПРОИЗОЙТИ?	4. КАКАЯ УЯЗВИМОСТЬ?	5. КАКОЙ УЩЕРБ?	6. КАК ЗАЩИТИТЬ?

Пример системы: API интернет-магазина с авторизацией и базой данных.

Шаг	Пример ответа
Актив	учётные записи, заказы, персональные данные, БД
Нарушитель	внешний злоумышленник или пользователь с недостаточными ограничениями
Сценарий	попытка получить доступ к чужому заказу
Уязвимость	ошибка проверки прав на стороне API
Последствие	раскрытие чужих данных
Мера	серверная авторизация + проверка принадлежности ресурса + тесты

Практика

- ☐ Для API интернет-магазина составить 5 потенциальных угроз.
- ☐ Для каждой угрозы указать актив, уязвимость и последствие.
- ☐ Назначить защитную меру.

5. База угроз ФСТЭК

Время: 3 часа • Инструмент систематизации угроз, который используется в учебной работе с моделями угроз.

Как понимать базу угроз

Это не список, который нужно механически заучить. Для учебного маршрута важно понять её назначение: систематизировать возможные угрозы и использовать их как основу при анализе информационной системы.

Как работать с ней

Сначала

- описать систему
- определить активы
- определить нарушителей
- найти потенциальные точки воздействия

Затем

- сопоставить сценарий с угрозой
- оценить последствия
- выбрать меры защиты
- зафиксировать результат

В программах курса база данных угроз ФСТЭК России включена в тему угроз. В PDF её лучше показывать как рабочий инструмент, а не как отдельный справочник для заучивания.

Мини-практика

☐ Взять один сценарий угрозы для API.

☐ Найти соответствующую запись в базе угроз.

☐ Записать название/идентификатор угрозы и объект воздействия.

☐ Объяснить своими словами, почему сопоставление подходит.

6. Блок №3. Современные технологии и информационные риски

Время: 30 часов • Связка организационно-правовых мер, доступа к ресурсам, атак, веб-систем и средств защиты.

6.1. Доступ к информационным ресурсам — 4 ч

КТО?	→	К ЧЕМУ?	→	С КАКИМ УРОВНЕМ ДОСТУПА?
Junior Developer	→	**dev**	→	**только необходимые права**
Senior Developer	→	**dev + staging**	→	**расширенные права по роли**
Administrator	→	**production**	→	**административные права**

6.2. Основные виды атак — 6 ч

• вредоносное ПО; • фишинг; • социальная инженерия; • атаки на веб-приложения; • кража учётных данных; • утечки данных.

6.3. Безопасность веб-приложений — 6 ч

Что вводит пользователь?	Кто имеет право?	Что проверяет сервер?	Что хранится и кто читает?

6.4. Программно-аппаратные средства защиты — 5 ч

Назначение

- антивирусные средства
- межсетевые экраны
- контроль доступа
- средства защиты данных

Контроль и наблюдение

- мониторинг
- регистрация событий
- выявление подозрительной активности
- проверка состояния защиты

6.5. Практический кейс — 5 ч

Сценарий: пользователь сообщает, что его аккаунт был взломан

☐ Определить, что могло произойти.

☐ Определить, какие данные могли быть скомпрометированы.

☐ Назвать реализованные угрозы.

☐ Предложить меры, которые могли предотвратить инцидент.

6.6. Самопроверка — 4 ч

Проверить себя: зачем разграничивать доступ; чем отличается доступ к dev/staging/production; где проходит граница доверия в веб-приложении; какие данные могут утечь; какие средства защиты закрывают разные типы рисков.

7. Блок №4. Человек как элемент информационной безопасности

Время: 30 часов • Человек рассматривается не отдельно от технологий, а как часть системы защиты.

7.1. Политика безопасности — 6 ч

Простое объяснение
Политика безопасности отвечает на вопрос: что пользователям разрешено, а что запрещено делать с системой?

минимум 12 символов	только необходимые права	при подозрении на компрометацию

7.2. Человеческий фактор — 6 ч

Слабый пароль • передача учётной записи • открытие подозрительной ссылки • установка неизвестного ПО • неправильная настройка доступа.

7.3. Социальная инженерия — 6 ч

Сценарий
«Здравствуйте, это техническая поддержка. Назовите код из SMS». Студент должен определить: что пытается сделать злоумышленник; какую информацию он хочет получить; почему человек может поверить; какое правило безопасности должно сработать.

7.4. Осведомлённость сотрудников — 4 ч

ПЕРЕД ДЕЙСТВИЕМ СПРОСИ СЕБЯ

1. Кто отправитель?
2. Зачем ему эта информация?
3. Действительно ли он имеет право её получать?
4. Куда ведёт ссылка?
5. Что произойдёт, если я ошибусь?

7.5. Практика — 5 ч

Разбор ситуаций

- ☐ Разобрать 3–5 коротких сообщений или рабочих сценариев.
- ☐ Для каждого определить признаки социальной инженерии.
- ☐ Определить, какие данные пытаются получить.
- ☐ Сформулировать правило, которое должен выполнить сотрудник.

7.6. Самопроверка — 3 ч

Вопросы

- ☐ Зачем нужна политика безопасности?
- ☐ Почему техническая защита не отменяет человеческий фактор?
- ☐ Какие признаки указывают на попытку социальной инженерии?
- ☐ Какие действия пользователя могут привести к компрометации?

8. Блок №5. Как организовать защиту информационной системы

Время: 32 часа • Финальный крупный блок: технические, организационные и правовые меры, непрерывность и реагирование.

8.1. Три вида мер защиты — 5 ч

Firewall, шифрование, резервное копирование	регламенты, обучение, разграничение прав	требования законодательства, договоры, ответственность

9. Непрерывность бизнеса — 5 ч

Сценарий
Интернет-магазин перестал работать на 8 часов. Вопрос: как сделать так, чтобы компания продолжила работать?

Изучить: резервирование; резервное копирование; восстановление; планы действий; показатели бесперебойности.

10. Инцидент информационной безопасности — 6 ч

ОБНАРУЖЕНИЕ	ФИКСАЦИЯ	АНАЛИЗ	ЛОКАЛИЗАЦИЯ	УСТРАНЕНИЕ	ВОССТАНОВЛЕНИЕ	УЛУЧШЕНИЕ

После восстановления важно не закрывать инцидент формально. Нужно проанализировать причины и изменить защиту так, чтобы повторение стало менее вероятным.

11. Безопасность программного обеспечения

Время: 8 часов • Максимально практическая адаптация материала к Программной инженерии.

Этап разработки	Вопрос безопасности
Анализ требований	Что необходимо защищать?
Проектирование	Где находятся риски?
Разработка	Как избежать уязвимостей?
Тестирование	Можно ли обойти защиту?
Развёртывание	Безопасно ли настроена система?
Эксплуатация	Как обнаруживать инциденты?
Обновление	Не появилась ли новая уязвимость?

Что должен унести разработчик

Безопасность не добавляется одним финальным этапом. Она проходит через требования, архитектуру, код, тесты, конфигурацию, эксплуатацию и обновления.

Практика

- ☐ Взять небольшую функцию приложения.
- ☐ Описать требования безопасности к ней.
- ☐ Найти потенциальную уязвимость на этапе проектирования.
- ☐ Добавить негативный тест, который должен обнаружить проблему.

12. Криптография — отдельный мини-модуль

Время: 6 часов • Без сложной математики: понять назначение основных механизмов и их место в ПО.

Шифрование



Хеширование



Электронная подпись



Связать с программированием: где шифруются данные, где хранится хеш, где проверяется подпись, как происходит аутентификация и авторизация.

Самопроверка

- ☐ Чем шифрование отличается от хеширования?
- ☐ Для чего нужна электронная подпись?
- ☐ Какой механизм отвечает за проверку неизменности данных?
- ☐ Где криптография встречается в обычном веб-приложении?

13. Итоговый практический проект

Время: 10 часов • «Анализ информационной безопасности небольшого веб-приложения».

ПОЛЬЗОВАТЕЛЬ → WEB-САЙТ → API → BACKEND → DATABASE

Задача проекта — пройти весь маршрут курса на одном объекте: от определения активов до действий после инцидента.

ЭТАП	ВОПРОС
Шаг 1	Какие активы необходимо защищать?
Шаг 2	Какие угрозы существуют?
Шаг 3	Какие уязвимости могут использоваться?
Шаг 4	Какие последствия возможны?
Шаг 5	Какие меры защиты необходимы?
Шаг 6	Какие организационные правила нужны?
Шаг 7	Как обнаружить инцидент?
Шаг 8	Что делать после инцидента?

Финальный результат

Проект считается завершённым, если студент может объяснить

что защищается → от чего защищается → какая уязвимость существует → какое последствие возможно → какая мера выбрана → как её проверить → как обнаружить инцидент → что делать после него.

Важно: итоговый проект не требует построения реальной защищённой инфраструктуры. Его цель — показать последовательное применение понятий курса к программной системе.

Итог разделов 1–13

После этих разделов студент должен видеть информационную безопасность как часть инженерного процесса, а не как отдельный набор программ.

термины, активы, требования	угрозу и уязвимость	CIA и последствия	технические, организационные и правовые меры	тесты, мониторинг, реагирование

Перед переходом к следующей части курса проверьте

- ☐ Я могу назвать активы информационной системы.
- ☐ Я отличаю угрозу от уязвимости.
- ☐ Я умею использовать CIA для описания последствий.
- ☐ Я могу построить простую модель угроз.
- ☐ Я могу выбрать защитную меру и объяснить, зачем она нужна.
- ☐ Я могу связать безопасность с этапом разработки ПО.
- ☐ Я могу описать базовый порядок реагирования на инцидент.