

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
по выполнению практических работ по дисциплине  
«Защита программ и данных»  
для студентов направления подготовки  
10.04.01 «Информационная безопасность»  
направленность (профиль)  
«Математическое и программное обеспечение защиты информации»

Ставрополь 2026

## СОДЕРЖАНИЕ

Введение

Соответствие практических занятий темам рабочей программы дисциплины

Общие требования к выполнению и оформлению практических занятий

Практическое занятие № 1. Анализ угроз безопасности программного обеспечения и построение модели нарушителя

Практическое занятие № 2. Оценка процессов разработки программного обеспечения на соответствие требованиям ГОСТ Р 56939. Формирование требований доверия

Практическое занятие № 3. Обеспечение эксплуатационной безопасности программного обеспечения: контроль целостности и замкнутая программная среда

Практическое занятие № 4. Исследование механизмов защиты программ от анализа и модификации

Практическое занятие № 5. Контроль отсутствия недеklarированных возможностей в исходных текстах программ

Практическое занятие № 6. Анализ отечественной нормативной базы в области защиты программного обеспечения. Работа с Единым реестром российских программ

Список рекомендуемой литературы

## Введение

Настоящие методические указания предназначены для проведения практических занятий по дисциплине «Защита программ и данных» для обучающихся по направлению подготовки 10.04.01 «Информационная безопасность», направленность (профиль) «Математическое и программное обеспечение защиты информации».

Тематика практических занятий полностью соответствует содержанию раздела 5 рабочей программы дисциплины: каждому практическому занятию поставлена в соответствие тема рабочей программы и объём часов, предусмотренный учебным планом. Общий объём практических занятий составляет 27 астрономических часов в 1 семестре.

В ходе практических занятий формируются компетенции ПК-3 «Способен проводить инструментальный анализ защищённости компьютерной системы и осуществлять поиск уязвимостей программного обеспечения» и ПК-4 «Проведение экспертизы при расследовании компьютерных преступлений, правонарушений и инцидентов» по индикаторам ИД-2 ПК-3, ИД-3 ПК-3, ИД-3 ПК-4, ИД-5 ПК-4.

Последовательность занятий отражает логику дисциплины: от постановки задачи защиты программного обеспечения и анализа угроз (занятие 1) через организационно-технологические основания доверия (занятие 2) и практические механизмы эксплуатационной безопасности (занятие 3) к исследованию защитных механизмов исполняемого кода (занятие 4), контролю отсутствия недеklarированных возможностей (занятие 5) и правовым основаниям профессиональной деятельности (занятие 6).

Практические занятия выполняются с использованием отечественного программного обеспечения, включённого в Единый реестр российских программ для электронных вычислительных машин и баз данных, а также свободно распространяемых средств, применяемых в учебных целях. Конкретный состав программного обеспечения указан в описании каждого занятия и соответствует разделу 9 рабочей программы дисциплины.

### Соответствие практических занятий темам рабочей программы дисциплины

| № ПЗ  | Тема практического занятия                                                                                                     | Тема рабочей программы дисциплины (раздел 5)                                                                                      | Объём, астр. ч. |
|-------|--------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|-----------------|
| 1     | Анализ угроз безопасности программного обеспечения и построение модели нарушителя                                              | Тема 1. Введение в теорию и практику защиты программного обеспечения                                                              | 4               |
| 2     | Оценка процессов разработки программного обеспечения на соответствие требованиям ГОСТ Р 56939. Формирование требований доверия | Тема 2. Основания теории и практики защиты программного обеспечения                                                               | 4               |
| 3     | Обеспечение эксплуатационной безопасности программного обеспечения: контроль целостности и замкнутая программная среда         | Тема 3. Методы обеспечения технологической и эксплуатационной безопасности программного обеспечения                               | 5               |
| 4     | Исследование механизмов защиты программ от анализа и модификации                                                               | Тема 4. Защита программ от анализа и модификации                                                                                  | 5               |
| 5     | Контроль отсутствия недеklarированных возможностей в исходных текстах программ                                                 | Тема 5. Исследование программного обеспечения на предмет отсутствия недеklarированных возможностей                                | 5               |
| 6     | Анализ отечественной нормативной базы в области защиты программного обеспечения. Работа с Единым реестром российских программ  | Тема 6. Краткое описание отечественных нормативных актов, регламентирующих деятельность в области защиты программного обеспечения | 4               |
| ИТОГО |                                                                                                                                |                                                                                                                                   | 27              |

## **Общие требования к выполнению и оформлению практических занятий**

Каждое практическое занятие выполняется по индивидуальному варианту, номер которого назначается преподавателем. Выполнение работы по чужому варианту не засчитывается.

Порядок работы над практическим занятием включает три этапа: подготовка (изучение теоретической части и рекомендуемой литературы в часы самостоятельной работы), выполнение (работа в лаборатории по установленному порядку) и защита (представление отчёта и собеседование по контрольным вопросам).

Обучающийся, не изучивший теоретическую часть, к выполнению работы не допускается. Готовность к работе проверяется преподавателем в начале занятия путём опроса по вопросам базового уровня.

### **Требования к оформлению отчёта**

Отчёт по практическому занятию оформляется в текстовом редакторе (МойОфис Текст, Р7-Офис. Документы или эквивалент) в соответствии со следующими требованиями: шрифт Times New Roman, кегль 14, межстрочный интервал полуторный, поля: левое – 3 см, правое – 1,5 см, верхнее и нижнее – 2 см, абзацный отступ – 1,25 см, выравнивание по ширине.

Структура отчёта: титульный лист с указанием темы практического занятия и номера варианта; цель работы; краткие теоретические сведения (не более одной страницы, своими словами); исходные данные варианта; описание процесса выполнения каждого задания с промежуточными результатами; итоговые таблицы и заключения, предусмотренные заданиями; выводы о проделанной работе; список использованных источников.

Иллюстрации (снимки экрана) размещаются по центру страницы, имеют подпись вида «Рисунок 1 – Настройка контроля целостности» и ссылку в тексте. Таблицы имеют заголовок вида «Таблица 1 – Перечень актуальных угроз» и сквозную нумерацию. Снимки экрана без пояснений в тексте не засчитываются.

Отчёт представляется преподавателю в электронном виде в формате ODT или PDF не позднее срока, установленного графиком самостоятельной работы. Защита отчёта проводится в форме собеседования по контрольным вопросам.

### **Критерии оценивания практического занятия**

Оценка «отлично» – все задания выполнены в полном объёме, результаты корректны и обоснованы, отчёт оформлен без нарушений требований, обучающийся свободно отвечает на контрольные вопросы, в том числе повышенного уровня, и способен обосновать принятые решения.

Оценка «хорошо» – все задания выполнены, допущены отдельные неточности в оформлении или обоснованиях, обучающийся отвечает на контрольные вопросы базового уровня и на большую часть вопросов повышенного уровня.

Оценка «удовлетворительно» – задания выполнены не менее чем на две трети, обучающийся отвечает на вопросы базового уровня, испытывает затруднения в обосновании результатов.

Оценка «неудовлетворительно» – задания не выполнены либо выполнены менее чем наполовину, отчёт не оформлен, обучающийся не отвечает на вопросы базового уровня.

## **Практическое занятие № 1**

### **Анализ угроз безопасности программного обеспечения и построение модели нарушителя**

**Соответствие рабочей программе:** Тема 1. Введение в теорию и практику защиты программного обеспечения.

**Формируемые индикаторы:** ИД-2 ПК-3, ИД-3 ПК-3.

**Объём:** 4 астр. ч.

**Цель работы:** сформировать навыки выявления угроз безопасности программного обеспечения на этапах его жизненного цикла и построения модели нарушителя с использованием Банка данных угроз безопасности информации ФСТЭК России.

#### **Задачи работы:**

- изучить состав и границы программного обеспечения как объекта защиты;
- освоить работу с Банком данных угроз безопасности информации ФСТЭК России;
- приобрести навыки построения модели нарушителя и определения актуальности угроз.

#### **Теоретическая часть**

##### ***1.1. Программное обеспечение как объект защиты***

В классической постановке задач защиты информации объектом защиты выступают сведения, обрабатываемые в информационной системе, а программное обеспечение рассматривается как средство их обработки. Такой подход оказывается недостаточным, поскольку само программное обеспечение обладает самостоятельной ценностью и может выступать как предметом посягательства, так и инструментом реализации угрозы в отношении обрабатываемых данных.

При определении границ объекта защиты в состав программного обеспечения включают: исходные тексты программ; результаты сборки – исполняемые файлы, динамические библиотеки, модули ядра; конфигурационные файлы и файлы настроек; сопроводительную и эксплуатационную документацию; средства сборки и тестирования; используемые сторонние компоненты и библиотеки. Каждый из перечисленных элементов имеет собственный набор угроз, и упущение любого из них при анализе приводит к неполноте результата.

Отдельного внимания заслуживают сторонние компоненты. Современное программное изделие на 60–90 процентов состоит из заимствованного кода – библиотек, фреймворков, пакетов из внешних репозиторий. Контроль этой части объекта защиты выделяется в самостоятельную задачу защиты цепочки поставки программного обеспечения.

##### ***1.2. Угрозы безопасности на этапах жизненного цикла***

Жизненный цикл программного обеспечения включает формирование требований, проектирование, разработку, тестирование, распространение, ввод в эксплуатацию, сопровождение и снятие с эксплуатации. Принципиальная особенность угроз безопасности программного обеспечения состоит в разрыве во времени между внесением дефекта и его проявлением: закладка, внесённая на этапе разработки, может быть активирована спустя годы эксплуатации.

Стоимость устранения дефекта возрастает по мере продвижения по жизненному циклу. Дефект, выявленный на этапе формирования требований, устраняется правкой документа; тот же дефект, выявленный в эксплуатируемой системе, требует выпуска обновления, его доставки всем пользователям и проверки установки, а в отдельных случаях – расследования инцидента.

*Таблица 1 – Характерные угрозы на этапах жизненного цикла программного обеспечения*

| Этап жизненного цикла   | Характерные угрозы                                                                                                  | Меры противодействия                                                          |
|-------------------------|---------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| Формирование требований | Неполнота или отсутствие требований по безопасности; неучтённые сценарии нарушения                                  | Анализ требований по безопасности, моделирование угроз                        |
| Проектирование          | Архитектурные решения, не допускающие последующего контроля; отсутствие разграничения доверия                       | Экспертиза архитектуры, выделение доверенных компонентов                      |
| Разработка              | Внесение программных закладок; ошибки кодирования, приводящие к уязвимостям; заимствование недоверенных компонентов | Экспертиза исходных текстов, статический анализ, контроль состава компонентов |
| Тестирование            | Неполнота покрытия кода тестами; отсутствие проверок на некорректные входные данные                                 | Оценка покрытия, динамический анализ, фаззинг-тестирование                    |
| Распространение         | Подмена дистрибутива; компрометация канала доставки                                                                 | Электронная подпись дистрибутива, доверенные репозитории                      |
| Ввод в эксплуатацию     | Небезопасная конфигурация по умолчанию; сохранение отладочных механизмов                                            | Контроль настроек, удаление отладочных интерфейсов                            |
| Сопровождение           | Использование компонентов с известными уязвимостями; подмена обновления                                             | Управление уязвимостями, подписанные обновления                               |
| Снятие с эксплуатации   | Сохранение остаточных данных и учётных записей                                                                      | Гарантированное удаление данных, отзыв доступов                               |

### ***1.3. Уязвимость, недеklarированная возможность, программная закладка***

Три ключевых понятия дисциплины нередко смешиваются, между тем различие между ними имеет практическое значение, поскольку определяет метод выявления и меру противодействия.

Уязвимость – недостаток программного обеспечения, использование которого нарушителем приводит к нарушению конфиденциальности, целостности или доступности информации. Уязвимость, как правило, возникает непреднамеренно, вследствие ошибки проектирования или кодирования. Выявляется тестированием, статическим и динамическим анализом, отслеживается по банкам данных уязвимостей.

Недекларированные (недокументированные) возможности – функциональные возможности программного обеспечения, не описанные в документации, при использовании которых возможно нарушение безопасности информации. Признак недеklarированной возможности – расхождение между документацией и фактическим поведением программы, а не наличие ошибки как таковой.

Программная закладка – преднамеренно внесённый в программу фрагмент, реализующий недеklarированные возможности. Отличается от прочих недеklarированных возможностей наличием умысла, что определяет и специфику противодействия: организационные меры контроля разработчиков, экспертиза исходных текстов, разграничение полномочий при внесении изменений в код.

### ***1.4. Банк данных угроз безопасности информации ФСТЭК России***

Банк данных угроз безопасности информации ФСТЭК России (bdu.fstec.ru) является основным отечественным источником систематизированных сведений об угрозах и уязвимостях и ведётся ФСТЭК России во взаимодействии с заинтересованными организациями.

Банк содержит два основных раздела. Раздел угроз включает записи вида УБИ.ХХХ, каждая из которых описывает наименование угрозы, источник, объект воздействия и нарушаемое свойство безопасности информации. Раздел уязвимостей включает записи вида

BDU:20XX-XXXXX с указанием уязвимого программного обеспечения, типа ошибки, уровня опасности по метрике CVSS и способа устранения.

При работе с банком следует учитывать, что запись об угрозе носит обобщённый характер и требует конкретизации применительно к рассматриваемому объекту. Формулировка вида «угроза внедрения кода» без указания того, какой именно компонент, каким способом и с какими последствиями подвергается воздействию, не является результатом анализа.

Порядок определения актуальных угроз установлен Методикой оценки угроз безопасности информации ФСТЭК России. Методика предусматривает последовательное определение негативных последствий, объектов воздействия, источников угроз, способов реализации и, как итог, перечня актуальных угроз.

### **1.5. Модель нарушителя и определение актуальности угроз**

Модель нарушителя представляет собой описание предполагаемых субъектов, способных реализовать угрозы, и включает виды нарушителей (внешний, внутренний), их категории, цели и потенциал.

Потенциал нарушителя – совокупность доступных ему знаний, средств, времени и возможностей. Разграничение по потенциалу принципиально: угроза, реализуемая только нарушителем с высоким потенциалом, может быть признана неактуальной для системы, не представляющей интереса для такого нарушителя, тогда как угрозы, реализуемые нарушителем с базовым потенциалом, актуальны практически всегда.

Угроза признаётся актуальной при одновременном выполнении трёх условий: имеется источник угрозы, обладающий необходимым потенциалом; существует способ реализации угрозы применительно к рассматриваемому объекту воздействия; реализация угрозы приводит к наступлению негативных последствий. Отсутствие любого из условий позволяет признать угрозу неактуальной, что должно быть письменно обосновано.

*Таблица 2 – Категории нарушителей и их потенциал*

| Категория нарушителя                        | Вид        | Возможности                                                                  | Потенциал          |
|---------------------------------------------|------------|------------------------------------------------------------------------------|--------------------|
| Отдельное физическое лицо                   | Внешний    | Использование общедоступных средств и опубликованных сведений об уязвимостях | Базовый            |
| Пользователь системы                        | Внутренний | Легальный доступ к части функций и данных, знание порядка работы             | Базовый            |
| Администратор системы                       | Внутренний | Полный доступ к настройкам, средствам защиты и журналам                      | Базовый повышенный |
| Разработчик программного обеспечения        | Внутренний | Доступ к исходным текстам и средствам сборки, возможность внесения закладок  | Базовый повышенный |
| Конкурирующая организация                   | Внешний    | Привлечение специалистов, приобретение специализированных средств анализа    | Средний            |
| Специальная служба иностранного государства | Внешний    | Выявление ранее неизвестных уязвимостей, воздействие на цепочку поставки     | Высокий            |

**Оборудование и материалы:** персональный компьютер с установленной операционной системой Astra Linux Special Edition (или РЕД ОС), офисным пакетом «Р7-Офис» либо «МойОфис Стандартный», справочной правовой системой «КонсультантПлюс» или «Гарант» и доступом к сети «Интернет».

**Указания по технике безопасности:** к выполнению практических занятий допускаются обучающиеся, ознакомившиеся с правилами работы в лаборатории и

прошедшие инструктаж по технике безопасности. Работы, связанные с анализом исполняемого кода, выполняются исключительно на учебных образцах, предоставленных преподавателем, в изолированном сегменте лабораторной сети. Подключение личных носителей информации и запуск посторонних программ не допускаются. Исследование чужого программного обеспечения без разрешения правообладателя недопустимо и влечёт ответственность, предусмотренную законодательством Российской Федерации.

### **Порядок выполнения работы**

1. Ознакомиться с теоретической частью настоящих указаний и рекомендуемой литературой.
2. Получить у преподавателя номер варианта и соответствующее ему описание программного изделия.
3. Описать объект защиты: назначение изделия, состав программных компонентов, используемые сторонние библиотеки, режимы работы, обрабатываемая информация.
4. Определить границы объекта защиты и обосновать включение или исключение отдельных компонентов.
5. Для каждого этапа жизненного цикла изделия определить возможные источники угроз и описать не менее двух угроз на этап, оформив результат по образцу таблицы 1.
6. Открыть раздел угроз Банка данных угроз безопасности информации ФСТЭК России и, используя фильтры по объекту воздействия и нарушаемому свойству безопасности, отобрать угрозы, применимые к рассматриваемому изделию.
7. Отобрать не менее десяти угроз и оформить их в виде таблицы с полями: идентификатор УБИ, наименование, источник, объект воздействия, нарушаемое свойство безопасности, конкретизация применительно к варианту.
8. Обратиться к разделу уязвимостей банка и проверить наличие зарегистрированных уязвимостей для сторонних компонентов, входящих в состав изделия; зафиксировать найденные записи.
9. Построить модель нарушителя: определить виды и категории нарушителей, их цели, возможности и потенциал; обосновать исключение категорий, признанных неактуальными.
10. Определить негативные последствия, наступающие при реализации каждой из отобранных угроз.
11. Сформировать перечень актуальных угроз, для каждой неактуальной угрозы привести письменное обоснование исключения.
12. Сформулировать предложения по мерам защиты, направленным на нейтрализацию актуальных угроз.
13. Оформить отчёт и подготовиться к собеседованию по контрольным вопросам.

### **Варианты заданий**

*Таблица 3 – Варианты заданий*

| № варианта | Программное изделие                                                | Особенности                      |
|------------|--------------------------------------------------------------------|----------------------------------|
| 1          | Клиент-серверная система обработки персональных данных сотрудников | Веб-интерфейс, внешняя СУБД      |
| 2          | Мобильное приложение дистанционного банковского обслуживания       | Работа в недоверенной среде      |
| 3          | Встроенное программное обеспечение промышленного контроллера       | Отсутствие механизмов обновления |
| 4          | Система электронного документооборота организации                  | Электронная подпись документов   |

| № варианта | Программное изделие                                             | Особенности                                      |
|------------|-----------------------------------------------------------------|--------------------------------------------------|
| 5          | Средство антивирусной защиты рабочих станций                    | Работа с привилегиями ядра                       |
| 6          | Автоматизированная система управления технологическим процессом | Объект критической информационной инфраструктуры |
| 7          | Веб-сервис предоставления государственных услуг                 | Публичная доступность, высокая нагрузка          |
| 8          | Программный комплекс бухгалтерского учёта                       | Интеграция с внешними сервисами                  |
| 9          | Система видеонаблюдения с распознаванием образов                | Обработка биометрических данных                  |
| 10         | Программное обеспечение медицинской информационной системы      | Специальные категории персональных данных        |

### Содержание отчёта

Отчёт оформляется в соответствии с общими требованиями, приведёнными во вводной части настоящих указаний, и дополнительно содержит все таблицы и заключения, предусмотренные порядком выполнения работы.

### Контрольные вопросы

1. Что понимается под программным обеспечением как объектом защиты и какие элементы включаются в его состав?
2. Почему сторонние компоненты требуют отдельного рассмотрения при анализе угроз?
3. Перечислите этапы жизненного цикла программного обеспечения и характерные для них угрозы.
4. Чем обусловлен рост стоимости устранения дефекта по мере продвижения по жизненному циклу?
5. В чём различие между уязвимостью, недекларированной возможностью и программной закладкой?
6. Какой признак позволяет отличить недекларированную возможность от ошибки реализации?
7. Какие разделы содержит Банк данных угроз безопасности информации ФСТЭК России?
8. Какова структура записи об угрозе и записи об уязвимости в банке данных?
9. Что означает конкретизация угрозы применительно к объекту защиты?
10. Что такое модель нарушителя и какие сведения она содержит?
11. Что такое потенциал нарушителя и как он определяется?
12. При каких условиях угроза признаётся актуальной?
13. Каким образом обосновывается исключение угрозы из перечня актуальных?
14. Какие негативные последствия рассматриваются при оценке угроз?

**Рекомендуемая литература:** 1, 2, 5, 7 (см. общий список литературы)

## Практическое занятие № 2

### Оценка процессов разработки программного обеспечения на соответствие требованиям ГОСТ Р 56939. Формирование требований доверия

**Соответствие рабочей программе:** Тема 2. Основания теории и практики защиты программного обеспечения.

**Формируемые индикаторы:** ИД-2 ПК-3, ИД-3 ПК-3.

**Объём:** 4 астр. ч.

**Цель работы:** сформировать навыки сопоставления процессов разработки программного обеспечения с мерами безопасной разработки и определения требований доверия к программному изделию.

#### **Задачи работы:**

- освоить состав мер безопасной разработки программного обеспечения;
- приобрести навыки оценки процесса разработки на соответствие национальному стандарту;
- научиться определять требуемый уровень доверия к программному изделию.

#### **Теоретическая часть**

##### ***2.1. Понятие доверенного программного обеспечения***

Доверенным считается программное обеспечение, для которого получены обоснованные свидетельства того, что оно выполняет только заявленные функции и не содержит недеklarированных возможностей. Ключевое слово в этом определении – свидетельства: доверие не является свойством, присущим программе, а представляет собой результат проведённых проверок, зафиксированный документально.

Принципиальное следствие такого понимания состоит в том, что доверие невозможно установить исключительно по готовому продукту. Проверка бинарного файла даёт ограниченный результат, поскольку часть свойств программы проявляется только при определённых условиях, а полный перебор состояний вычислительно неосуществим. Поэтому свидетельства формируются на протяжении всего процесса создания программного обеспечения, что и составляет содержание концепции безопасной разработки.

Из этого следует и практическое требование: разработчик, претендующий на признание своего изделия доверенным, должен не только применять меры безопасной разработки, но и сохранять свидетельства их применения – протоколы анализа, отчёты инструментальных средств, записи об обработке обнаруженных дефектов.

##### ***2.2. Структура и состав мер ГОСТ Р 56939***

ГОСТ Р 56939 «Защита информации. Разработка безопасного программного обеспечения. Общие требования» устанавливает состав мер, реализуемых разработчиком на этапах жизненного цикла программного обеспечения. Стандарт не предписывает конкретных инструментов и методик, оставляя их выбор за разработчиком, но требует, чтобы соответствующая мера была реализована и документирована.

Меры группируются по этапам жизненного цикла. На этапе анализа требований выполняется определение требований по безопасности и моделирование угроз. На этапе проектирования – экспертиза архитектуры. На этапе разработки – статический анализ исходных текстов, экспертиза исходных текстов, применение безопасных практик программирования. На этапе тестирования – динамический анализ, функциональное тестирование, фазинг-тестирование, оценка покрытия. На этапах поставки и сопровождения – управление конфигурацией, обеспечение защиты среды разработки, обработка обращений пользователей об обнаруженных ошибках и уязвимостях.

*Таблица 4 – Группы мер безопасной разработки и формируемые свидетельства*

| Группа мер                          | Содержание                                                   | Формируемые свидетельства                           |
|-------------------------------------|--------------------------------------------------------------|-----------------------------------------------------|
| Анализ требований по безопасности   | Выявление и документирование требований, моделирование угроз | Перечень требований, модель угроз                   |
| Экспертиза архитектуры              | Проверка архитектурных решений на реализуемость требований   | Заключение по результатам экспертизы                |
| Статический анализ исходных текстов | Автоматизированный поиск дефектов без выполнения программы   | Отчёты анализатора, записи об устранении дефектов   |
| Экспертиза исходных текстов         | Просмотр кода специалистом, не являющимся его автором        | Протоколы просмотра, перечень замечаний             |
| Динамический анализ и тестирование  | Проверка поведения программы при выполнении, фаззинг         | Протоколы тестирования, отчёты о покрытии           |
| Управление конфигурацией            | Контроль состава и изменений компонентов                     | Журнал изменений, перечень версий компонентов       |
| Защита среды разработки             | Разграничение доступа к репозиторию и средствам сборки       | Регламенты, журналы доступа                         |
| Обработка обращений                 | Приём и рассмотрение сообщений об ошибках и уязвимостях      | Реестр обращений, сведения о выпущенных обновлениях |

### **2.3. Уровни доверия к средствам защиты информации**

Требования доверия, установленные ФСТЭК России, задают шесть уровней; нумерация построена по убыванию строгости – первый уровень является наиболее строгим, шестой наименее. Уровень доверия определяет глубину проводимых испытаний, объём представляемых разработчиком материалов и, в частности, обязательность контроля отсутствия недеklarированных возможностей.

Выбор требуемого уровня доверия определяется характером защищаемой информации и классом информационной системы. Для государственных информационных систем и значимых объектов критической информационной инфраструктуры применяются более строгие уровни; для систем, обрабатывающих информацию ограниченного доступа, не относящуюся к государственной тайне, – менее строгие.

Практическое значение уровня доверия состоит в том, что он задаёт объём работ, которые разработчик обязан выполнить и подтвердить, ещё до подачи изделия на сертификационные испытания. Определение уровня доверия на раннем этапе позволяет заранее спланировать соответствующие процессы и избежать переработки готового изделия.

*Таблица 5 – Соотношение уровней доверия и объёма испытаний (обобщённо)*

| Уровень доверия | Область применения                                                                                 | Характер проводимых проверок                                                                       |
|-----------------|----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| 1–3             | Информационные системы, обрабатывающие сведения, составляющие государственную тайну                | Углублённый анализ исходных текстов, полный контроль отсутствия недеklarированных возможностей     |
| 4               | Значимые объекты критической информационной инфраструктуры, государственные информационные системы | Анализ исходных текстов, контроль отсутствия недеklarированных возможностей в установленном объёме |
| 5–6             | Информационные системы персональных данных, иные системы с информацией ограниченного доступа       | Функциональное тестирование, анализ уязвимостей, контроль в сокращённом объёме                     |

### **2.4. Общие критерии оценки защищённости**

ГОСТ Р ИСО/МЭК 15408 вводит формализованный аппарат описания требований безопасности, применимый к любому изделию информационных технологий. Основными понятиями являются объект оценки, профиль защиты, задание по безопасности и оценочный уровень доверия.

Профиль защиты описывает независимый от конкретной реализации набор требований безопасности для класса изделий определённого назначения. Задание по безопасности описывает требования применительно к конкретному изделию и служит основой для проведения его оценки. Такое разделение позволяет заказчику формулировать требования, не привязываясь к продукту, а разработчику – демонстрировать соответствие своего продукта заявленным требованиям.

Оценочный уровень доверия задаёт глубину оценки: от подтверждения функционального тестирования до формальной верификации проекта. Рост уровня сопровождается существенным ростом трудоёмкости и стоимости оценки, поэтому выбор уровня представляет собой компромисс между требуемой уверенностью и располагаемыми ресурсами.

### ***2.5. Модели безопасности и их прикладное значение***

Формальные модели безопасности компьютерных систем задают правила разграничения доступа и условия, при которых система признаётся безопасной. Дискреционные модели описывают доступ через матрицу прав субъектов на объекты, мандатные – через сопоставление меток конфиденциальности, ролевые – через назначение полномочий ролям, а не отдельным субъектам.

Применительно к защите программного обеспечения особое значение имеет субъектно-ориентированная модель, описывающая порождение новых субъектов доступа. Программа, будучи запущенной, порождает субъект, наследующий определённые полномочия; корректность этого порождения и составляет условие сохранения безопасности системы. Нарушение данного условия – типичный механизм действия программной закладки.

Модели скрытых каналов описывают передачу информации способами, не предусмотренными штатными механизмами взаимодействия: через занятость общих ресурсов, задержки в обработке, объём и частоту обращений. Скрытые каналы представляют собой предмет отдельного анализа при высоких уровнях доверия.

**Оборудование и материалы:** персональный компьютер с установленной операционной системой Astra Linux Special Edition (или РЕД ОС), офисным пакетом «Р7-Офис» либо «МойОфис Стандартный», доступом к справочным правовым системам «КонсультантПлюс» или «Гарант», к официальному сайту ФСТЭК России и к сети «Интернет».

**Указания по технике безопасности:** к выполнению практических занятий допускаются обучающиеся, ознакомившиеся с правилами работы в лаборатории и прошедшие инструктаж по технике безопасности. Работы, связанные с анализом исполняемого кода, выполняются исключительно на учебных образцах, предоставленных преподавателем, в изолированном сегменте лабораторной сети. Подключение личных носителей информации и запуск посторонних программ не допускаются. Исследование чужого программного обеспечения без разрешения правообладателя недопустимо и влечёт ответственность, предусмотренную законодательством Российской Федерации.

### **Порядок выполнения работы**

1. Ознакомиться с теоретической частью настоящих указаний и рекомендуемой литературой.
2. Получить у преподавателя номер варианта и описание процесса разработки условного программного изделия.
3. Изучить описание процесса: состав участников, применяемые средства, порядок внесения изменений, применяемые проверки.

4. Составить перечень мер безопасной разработки, установленных ГОСТ Р 56939 и применимых к рассматриваемому изделию.

5. Для каждой меры определить, реализована ли она в описанном процессе, реализована частично либо не реализована; результат оформить в виде таблицы соответствия.

6. Для каждой реализованной меры указать, какое свидетельство формируется и где оно хранится.

7. Определить характер обрабатываемой изделием информации и класс информационной системы, в которой оно применяется.

8. На основании определённого класса обосновать требуемый уровень доверия к изделию.

9. Определить, требуется ли для данного уровня доверия проведение контроля отсутствия недекларированных возможностей и в каком объёме.

10. Сформировать перечень недостающих мер и предложить план их внедрения с указанием этапа жизненного цикла, ответственного и ожидаемого свидетельства.

11. Оценить трудоёмкость внедрения недостающих мер в относительных единицах и расставить приоритеты.

12. Оформить отчёт и подготовиться к собеседованию по контрольным вопросам.

### Варианты заданий

Таблица 6 – Варианты заданий

| № варианта | Программное изделие                                      | Условия применения                         |
|------------|----------------------------------------------------------|--------------------------------------------|
| 1          | Средство криптографической защиты информации             | Информационная система персональных данных |
| 2          | Межсетевой экран уровня веб-приложений                   | Государственная информационная система     |
| 3          | Система обнаружения вторжений                            | Значимый объект КИИ                        |
| 4          | Средство защиты от несанкционированного доступа          | Государственная информационная система     |
| 5          | Операционная система общего назначения                   | Информационная система органа власти       |
| 6          | Система управления базами данных                         | Информационная система персональных данных |
| 7          | Средство доверенной загрузки                             | Значимый объект КИИ                        |
| 8          | Средство антивирусной защиты                             | Государственная информационная система     |
| 9          | Система управления событиями информационной безопасности | Информационная система организации         |
| 10         | Средство контроля съёмных носителей                      | Информационная система персональных данных |

### Содержание отчёта

Отчёт оформляется в соответствии с общими требованиями, приведёнными во вводной части настоящих указаний, и дополнительно содержит все таблицы и заключения, предусмотренные порядком выполнения работы.

### Контрольные вопросы

1. Какое программное обеспечение считается доверенным и что понимается под свидетельствами доверия?

2. Почему доверие невозможно установить исключительно по готовому продукту?

3. Назовите основные группы мер, установленных ГОСТ Р 56939.

4. Какие свидетельства формируются при реализации мер безопасной разработки?

5. В чём состоит различие между статическим и динамическим анализом кода?

6. Для чего проводится моделирование угроз на этапе анализа требований?
7. Что такое экспертиза исходных текстов и почему её проводит не автор кода?
8. Сколько уровней доверия установлено требованиями ФСТЭК России и как они соотносятся по строгости?
9. От чего зависит выбор требуемого уровня доверия к изделию?
10. Что такое профиль защиты и чем он отличается от задания по безопасности?
11. Что задаёт оценочный уровень доверия по ГОСТ Р ИСО/МЭК 15408?
12. Какую роль играет управление конфигурацией в обеспечении доверия?
13. Как организуется обработка обращений пользователей об обнаруженных уязвимостях?
14. В чём состоит содержание субъектно-ориентированной модели компьютерной системы?
15. Что такое скрытый канал передачи информации и какие виды скрытых каналов известны?

**Рекомендуемая литература:** 1, 2, 4, 5, 6 (см. общий список литературы)

### **Практическое занятие № 3**

#### **Обеспечение эксплуатационной безопасности программного обеспечения: контроль целостности и замкнутая программная среда**

**Соответствие рабочей программе:** Тема 3. Методы обеспечения технологической и эксплуатационной безопасности программного обеспечения.

**Формируемые индикаторы:** ИД-2 ПК-3, ИД-3 ПК-3.

**Объем:** 5 астр. ч.

**Цель работы:** сформировать навыки настройки механизмов контроля целостности программных компонентов и замкнутой программной среды, а также организации управления уязвимостями и обновлениями.

##### **Задачи работы:**

- освоить настройку механизмов контроля целостности программных компонентов;
- приобрести навыки формирования замкнутой программной среды;
- научиться организовывать управление уязвимостями с использованием отечественных источников данных.

#### **Теоретическая часть**

##### ***3.1. Технологическая и эксплуатационная безопасность***

Технологическая безопасность обеспечивается на этапе создания программного обеспечения и направлена на исключение внесения дефектов и закладок. Её составляющими являются контроль исходных текстов, управление сборкой, защита среды разработки, использование доверенных репозиторий и контроль состава заимствованных компонентов.

Эксплуатационная безопасность обеспечивается в процессе применения программного обеспечения и направлена на исключение несанкционированной модификации и запуска постороннего кода. Её составляющими являются контроль целостности, разграничение доступа, изоляция процессов, доверенная загрузка и замкнутая программная среда.

Разделение носит условный характер: обе группы мер решают общую задачу и дополняют друг друга. Технологические меры не защищают от подмены компонента после поставки, эксплуатационные – не выявляют закладку, внесённую в исходный код и попавшую в подписанный дистрибутив. Достаточный уровень защищённости достигается только их совместным применением.

##### ***3.2. Контроль целостности программных компонентов***

Контроль целостности состоит в вычислении эталонных значений контрольных характеристик защищаемых объектов и последующей периодической проверке соответствия текущих значений эталонным. В качестве контрольной характеристики применяются значения криптографических хеш-функций либо электронная подпись.

Объектами контроля выступают исполняемые файлы и динамические библиотеки, модули ядра, конфигурационные файлы, сценарии автозапуска, а также сами эталонные значения. Последнее существенно: если перечень эталонов не защищён, нарушитель модифицирует файл и пересчитает эталон, после чего механизм контроля перестанет срабатывать.

Режимы контроля различаются моментом проведения проверки. Контроль при загрузке выполняется однократно при запуске системы и обладает наименьшими накладными расходами. Контроль по расписанию проводится периодически. Контроль

перед запуском выполняется при каждом обращении к объекту и обеспечивает наибольшую полноту при наибольших накладных расходах.

При обнаружении нарушения целостности механизм должен блокировать использование изменённого объекта, зарегистрировать событие в журнале с указанием объекта, времени и учётной записи, а также известить администратора безопасности. Простая регистрация факта без блокирования не является достаточной мерой.

### ***3.3. Замкнутая программная среда и доверенная загрузка***

Замкнутая программная среда ограничивает перечень программ, разрешённых к запуску, заранее определённым списком. Всё, что не входит в список, к выполнению не допускается независимо от прав запускающего пользователя. Данный механизм противодействует запуску принесённых нарушителем программ, в том числе вредоносных, даже если они не обнаруживаются средством антивирусной защиты.

В отечественных операционных системах замкнутая программная среда реализуется на основе электронной подписи исполняемых файлов и модулей ядра. Ядро при загрузке модуля или запуске исполняемого файла проверяет подпись; при её отсутствии либо несоответствии запуск блокируется. Такая реализация одновременно решает задачу контроля целостности, поскольку модификация подписанного файла нарушает подпись.

Доверенная загрузка обеспечивает контроль целостности компонентов, участвующих в запуске системы, – загрузчика, ядра, начального образа файловой системы – до передачи им управления. Без доверенной загрузки прочие механизмы могут быть обойдены подменой загружаемого образа системы.

Практическое внедрение замкнутой программной среды требует предварительной инвентаризации: необходимо составить полный перечень программ, требуемых для работы, включая вспомогательные утилиты и компоненты, вызываемые опосредованно. Неполнота перечня приводит к отказам в работе прикладного программного обеспечения после включения механизма.

### ***3.4. Управление уязвимостями и обновлениями***

Управление уязвимостями представляет собой непрерывный процесс, включающий инвентаризацию используемых программных компонентов с указанием версий, регулярный контроль сведений о выявленных уязвимостях, оценку применимости уязвимости к конкретной конфигурации, планирование и установку обновлений и последующую проверку результата.

Инвентаризация является основой процесса: невозможно отследить уязвимости компонента, о наличии которого в системе неизвестно. Перечень компонентов должен поддерживаться в актуальном состоянии и включать не только устанавливаемые пакеты, но и библиотеки, входящие в состав прикладного программного обеспечения.

Оценка применимости позволяет избежать избыточных работ. Уязвимость, зарегистрированная для компонента, может не иметь значения для конкретной конфигурации, если уязвимая функция не используется или доступ к ней ограничен архитектурой системы. Обоснование неприменимости оформляется документально наравне с решением об установке обновления.

Обновления должны устанавливаться только из доверенного источника с проверкой электронной подписи. Установка обновления из непроверенного источника представляет собой самостоятельную угрозу – воздействие на цепочку поставки, при котором нарушитель получает доступ ко всем системам, установившим подменённое обновление.

Таблица 7 – Этапы процесса управления уязвимостями

| Этап                | Содержание                                                                       | Результат                                      |
|---------------------|----------------------------------------------------------------------------------|------------------------------------------------|
| Инвентаризация      | Формирование перечня программных компонентов с указанием версий                  | Актуальный перечень компонентов                |
| Мониторинг          | Регулярный контроль сведений об уязвимостях по БДУ ФСТЭК России                  | Перечень зарегистрированных уязвимостей        |
| Оценка применимости | Анализ конфигурации на предмет возможности использования уязвимости              | Перечень применимых уязвимостей с обоснованием |
| Планирование        | Определение сроков и порядка установки обновлений                                | План устранения                                |
| Устранение          | Установка обновлений из доверенного источника либо применение компенсирующих мер | Устранённые уязвимости                         |
| Контроль            | Проверка фактического устранения и целостности обновлённых компонентов           | Протокол проверки                              |

**Оборудование и материалы:** персональный компьютер с установленной операционной системой Astra Linux Special Edition (или РЕД ОС), средством защиты информации от несанкционированного доступа Secret Net Studio (при наличии), средством антивирусной защиты Kaspersky Endpoint Security или Dr.Web, офисным пакетом «Р7-Офис» либо «МойОфис Стандартный», доступом к Банку данных угроз ФСТЭК России.

**Указания по технике безопасности:** к выполнению практических занятий допускаются обучающиеся, ознакомившиеся с правилами работы в лаборатории и прошедшие инструктаж по технике безопасности. Работы, связанные с анализом исполняемого кода, выполняются исключительно на учебных образцах, предоставленных преподавателем, в изолированном сегменте лабораторной сети. Подключение личных носителей информации и запуск посторонних программ не допускаются. Исследование чужого программного обеспечения без разрешения правообладателя недопустимо и влечёт ответственность, предусмотренную законодательством Российской Федерации.

### Порядок выполнения работы

1. Ознакомиться с теоретической частью настоящих указаний и рекомендуемой литературой.
2. Получить у преподавателя номер варианта и соответствующий ему перечень защищаемых программных компонентов.
3. Провести инвентаризацию установленного программного обеспечения, зафиксировать наименования и версии компонентов.
4. Сформировать перечень объектов, подлежащих контролю целостности, обосновать включение каждого объекта.
5. Настроить механизм контроля целостности для выбранных объектов, вычислить и сохранить эталонные значения контрольных характеристик.
6. Определить и обосновать режим проведения контроля (при загрузке, по расписанию, перед запуском).
7. Проверить защищённость самого перечня эталонных значений от модификации.
8. Выполнить контролируемую модификацию одного из защищаемых файлов и убедиться в срабатывании механизма.
9. Зафиксировать соответствующую запись в журнале событий, привести её содержание в отчёте.
10. Восстановить исходное состояние модифицированного файла и убедиться в снятии признака нарушения целостности.
11. Составить перечень программ, необходимых для работы, для формирования замкнутой программной среды.

12. Настроить замкнутую программную среду в соответствии с составленным перечнем.

13. Проверить блокирование запуска программы, не входящей в перечень; зафиксировать реакцию системы и запись в журнале.

14. Проверить работоспособность прикладного программного обеспечения после включения замкнутой программной среды.

15. Провести проверку компонентов, выявленных при инвентаризации, по Банку данных угроз ФСТЭК России; составить перечень зарегистрированных уязвимостей.

16. Для каждой уязвимости оценить применимость к текущей конфигурации, обосновать вывод.

17. Сформировать план устранения применимых уязвимостей с указанием сроков и способов.

18. Оформить отчёт со снимками экрана и подготовиться к собеседованию по контрольным вопросам.

### Варианты заданий

Таблица 8 – Варианты заданий

| № варианта | Защищаемые компоненты                              | Дополнительное условие  |
|------------|----------------------------------------------------|-------------------------|
| 1          | Исполняемые файлы системных служб                  | Контроль при загрузке   |
| 2          | Конфигурационные файлы средств защиты              | Контроль по расписанию  |
| 3          | Модули ядра операционной системы                   | Контроль перед запуском |
| 4          | Динамические библиотеки прикладного ПО             | Контроль по расписанию  |
| 5          | Сценарии автоматического запуска                   | Контроль при загрузке   |
| 6          | Исполняемые файлы офисного пакета                  | Контроль перед запуском |
| 7          | Файлы настроек сетевых служб                       | Контроль по расписанию  |
| 8          | Компоненты средства антивирусной защиты            | Контроль перед запуском |
| 9          | Файлы учётных записей и разграничения доступа      | Контроль при загрузке   |
| 10         | Исполняемые файлы системы управления базами данных | Контроль по расписанию  |

### Содержание отчёта

Отчёт оформляется в соответствии с общими требованиями, приведёнными во вводной части настоящих указаний, и дополнительно содержит все таблицы и заключения, предусмотренные порядком выполнения работы.

### Контрольные вопросы

1. В чём различие между технологической и эксплуатационной безопасностью программного обеспечения?

2. Почему технологические и эксплуатационные меры не заменяют друг друга?

3. Как реализуется механизм контроля целостности программных компонентов?

4. Какие объекты подлежат контролю целостности и почему перечень эталонов также нуждается в защите?

5. Охарактеризуйте режимы проведения контроля целостности и области их применения.

6. Какие действия должны выполняться при обнаружении нарушения целостности?

7. Что такое замкнутая программная среда и на чём основана её реализация?

8. Почему замкнутая программная среда эффективна против вредоносного программного обеспечения, не обнаруживаемого антивирусом?

9. Какие подготовительные работы необходимы перед включением замкнутой программной среды?

10. Какую роль играет доверенная загрузка в обеспечении безопасности?

11. Как организуется изоляция процессов в современных операционных системах?

12. Перечислите этапы процесса управления уязвимостями.

13. Почему инвентаризация является основой управления уязвимостями?

14. Что понимается под оценкой применимости уязвимости и как она обосновывается?

15. Почему обновления должны устанавливаться только из доверенного источника?

**Рекомендуемая литература:** 1, 2, 3, 5, 7 (см. общий список литературы)

## Практическое занятие № 4

### Исследование механизмов защиты программ от анализа и модификации

**Соответствие рабочей программе:** Тема 4. Защита программ от анализа и модификации.

**Формируемые индикаторы:** ИД-3 ПК-3, ИД-3 ПК-4.

**Объём:** 5 астр. ч.

**Цель работы:** сформировать навыки оценки стойкости механизмов защиты исполняемого кода от статического и динамического анализа и выработки предложений по их усилению.

#### **Задачи работы:**

- освоить методы анализа программных реализаций;
- изучить приёмы противодействия дизассемблированию и отладке;
- приобрести навыки оценки стойкости механизмов защиты исполняемого кода.

#### **Теоретическая часть**

##### ***4.1. Задача анализа программных реализаций***

Задача анализа программных реализаций состоит в сборе детальной информации о функционировании программы. Актуальность задачи определяется многообразием программных средств обработки информации и применяемых в них средств защиты, значительным разбросом в уровне подготовленности разработчиков, а также широким распространением программных средств зарубежного производства, документация на которые может быть неполной либо не соответствовать фактическому поведению.

В рамках дисциплины анализ рассматривается как инструмент защиты: специалист по защите информации применяет его для оценки собственных механизмов защиты, выявления недекларированных возможностей в применяемом программном обеспечении и подготовки экспертных заключений при расследовании инцидентов.

Решение задачи анализа традиционно разбивается на этапы: определение формата и структуры исследуемого файла; выявление применённых средств упаковки и защиты; восстановление логики работы отдельных функций; определение взаимосвязей между функциями; формирование общего описания алгоритма работы программы.

##### ***4.2. Методы анализа программных реализаций***

Статический метод предполагает исследование программы без её выполнения. Инструментами служат дизассемблеры, декомпиляторы, средства просмотра структуры файла и шестнадцатеричные редакторы. Достоинство метода – полнота охвата кода, включая ветви, не выполняющиеся при обычной работе; недостаток – трудоёмкость и невозможность непосредственного наблюдения значений данных времени выполнения.

Динамический метод предполагает исследование программы в процессе выполнения. Инструментами служат отладчики, средства трассировки и перехвата вызовов, эмуляторы. Достоинство метода – наблюдение фактических значений данных и последовательности вызовов; недостаток – охват только тех ветвей, которые фактически выполнялись, и возможность обнаружения факта анализа самой исследуемой программой.

Метод экспериментов с «чёрным ящиком» предполагает многократное выполнение программы с различными входными данными и сравнительный анализ получаемых результатов без доступа к внутреннему устройству. Название заимствовано из математической теории автоматов, где чёрным ящиком именуется система, о которой известны только входы и выходы. Метод применим, когда исследование кода невозможно

или нецелесообразно; его недостатками являются неполнота результата, значительные затраты времени и невозможность обнаружения ветвей, активируемых редкими условиями.

Типичным примером применения метода «чёрного ящика» служит определение формата заголовков бинарного файла данных: последовательное создание файлов с известным, минимально различающимся содержимым и сравнение полученных результатов позволяет установить назначение отдельных полей заголовка.

#### **4.3. Противодействие статическому анализу**

Обфускация состоит в преобразовании кода и данных к виду, сохраняющему функциональность, но затрудняющему понимание. Применяются переименование идентификаторов, усложнение потока управления введением избыточных переходов и предикатов, преобразование констант в вычисляемые выражения, объединение и разделение функций.

Упаковка и шифрование исполняемых модулей исключают непосредственное дизассемблирование: в файле присутствует лишь распаковщик и зашифрованный образ кода, восстанавливаемый в памяти во время выполнения. Определение применённого упаковщика является первым шагом анализа и выполняется по сигнатурам средствами вида Detect It Easy.

Самогенерируемый код формируется программой во время выполнения и отсутствует в файле в готовом виде. Такой приём делает статический анализ принципиально неполным, поскольку исследуемые инструкции не существуют до момента запуска.

Приёмы, направленные против конкретных алгоритмов дизассемблирования, используют особенности их работы. Линейное дизассемблирование обрабатывает буфер последовательно до его конца, независимо от того, что инструкции управления потоком приводят к выполнению лишь части буфера. Вставка байта с данными после инструкции условного перехода приводит к тому, что линейный дизассемблер интерпретирует данные как начало инструкции и далее разбирает код со смещением. Преждевременное завершение функции инструкцией возврата в неожиданном месте нарушает работу рекурсивного дизассемблера, восстанавливающего границы функций.

*Таблица 9 – Приёмы противодействия анализу и их характеристики*

| Приём                                | Против какого анализа направлен              | Способ выявления                                                      |
|--------------------------------------|----------------------------------------------|-----------------------------------------------------------------------|
| Обфускация потока управления         | Статический                                  | Построение графа потока управления, выявление недостижимых предикатов |
| Упаковка и шифрование модуля         | Статический                                  | Сигнатурный анализ, аномально высокая энтропия секций                 |
| Самогенерируемый код                 | Статический                                  | Наблюдение изменения содержимого памяти при выполнении                |
| Вставка ложного байта                | Статический (линейное дизассемблирование)    | Сопоставление результатов линейного и рекурсивного разбора            |
| Преждевременный возврат из функции   | Статический (рекурсивное дизассемблирование) | Анализ фактических границ функций при выполнении                      |
| Проверка флага отладки процесса      | Динамический                                 | Поиск обращений к соответствующим системным функциям                  |
| Измерение времени выполнения участка | Динамический                                 | Поиск обращений к счётчику тактов процессора                          |
| Поиск программных точек останова     | Динамический                                 | Анализ проверок содержимого собственного кода                         |
| Контроль контрольной суммы кода      | Динамический                                 | Выявление вычисления хеша от собственных секций                       |

#### **4.4. Противодействие динамическому анализу**

Обнаружение отладочной среды основано на выявлении её характерных признаков. Наиболее прямой способ – обращение к системным функциям, сообщающим о наличии

присоединённого отладчика. Способ прост в реализации и столь же прост в обходе, поэтому применяется в сочетании с другими.

Измерение времени выполнения участка кода использует то обстоятельство, что пошаговое выполнение под отладчиком занимает на порядки больше времени, чем обычное. Для измерения применяется счётчик тактов процессора, считываемый инструкцией `rdtsc`: разность значений до и после участка, превышающая пороговое значение, свидетельствует об отладке.

Поиск программных точек останова основан на том, что установка такой точки означает замену первого байта инструкции специальным однобайтовым кодом прерывания. Программа может просматривать собственный код в поисках этого значения. Противодействием служат аппаратные точки останова, использующие отладочные регистры процессора и не изменяющие код, либо редактирование маршрута выполнения непосредственно при отладке.

Контроль контрольной суммы собственного кода позволяет обнаружить как установку программных точек останова, так и внесение исправлений в исполняемый файл. Приём эффективен против модификации, но требует, чтобы сам механизм подсчёта был защищён от отключения.

#### ***4.5. Контроль целостности кода и защита от копирования***

Электронная подпись исполняемого кода подтверждает его происхождение и неизменность с момента подписания. Проверка подписи операционной системой при запуске исключает выполнение модифицированного файла и является наиболее надёжным механизмом контроля целостности, поскольку опирается на криптографические, а не эвристические основания.

Защита от копирования реализуется программными механизмами (привязка к параметрам оборудования, активация через сервер, лицензионные файлы) и программно-аппаратными (электронные ключи, подключаемые к вычислительной системе). Программно-аппаратные механизмы обеспечивают более высокую стойкость, поскольку часть логики защиты выносится за пределы исследуемой вычислительной системы.

Типовым способом преодоления программно-аппаратной защиты является эмуляция ключа – создание программного модуля, воспроизводящего его ответы. Противодействием служат перенос части вычислений внутрь ключа, применение криптографических протоколов с невоспроизводимыми ответами и контроль времени отклика.

#### ***4.6. Оценка стойкости механизмов защиты***

Абсолютная защита исполняемого кода от анализа принципиально недостижима: программа, выполняемая на вычислительной системе, находящейся под полным контролем исследователя, может быть исследована. Задача защиты состоит не в исключении анализа, а в повышении его трудоёмкости до уровня, при котором затраты превышают ценность получаемого результата.

Стойкость механизма защиты оценивается по трудоёмкости его преодоления, выражаемой в требуемой квалификации исследователя, необходимых средствах и затратах времени. Оценка носит сравнительный характер и должна пересматриваться по мере развития средств анализа.

При оценке следует учитывать совокупный эффект: последовательное применение нескольких независимых механизмов повышает трудоёмкость сильнее, чем усложнение любого из них по отдельности, поскольку исследователь обязан преодолеть каждый.

**Оборудование и материалы:** персональный компьютер с установленной операционной системой Astra Linux Special Edition (или РЕД ОС), средствами Detect It Easy, x64dbg, Cutter (radare2), шестнадцатеричным редактором HxD, офисным пакетом «Р7-Офис» либо «МойОфис Стандартный»; учебные образцы исполняемых файлов,

предоставленные преподавателем. Работа выполняется в изолированном сегменте лабораторной сети.

**Указания по технике безопасности:** к выполнению практических занятий допускаются обучающиеся, ознакомившиеся с правилами работы в лаборатории и прошедшие инструктаж по технике безопасности. Работы, связанные с анализом исполняемого кода, выполняются исключительно на учебных образцах, предоставленных преподавателем, в изолированном сегменте лабораторной сети. Подключение личных носителей информации и запуск посторонних программ не допускаются. Исследование чужого программного обеспечения без разрешения правообладателя недопустимо и влечёт ответственность, предусмотренную законодательством Российской Федерации.

### **Порядок выполнения работы**

1. Ознакомиться с теоретической частью настоящих указаний и рекомендуемой литературой.
2. Получить у преподавателя учебный образец исполняемого файла согласно номеру варианта и зафиксировать его контрольную сумму.
3. Определить формат файла, разрядность, целевую платформу и предполагаемый компилятор.
4. Средствами Detect It Easy определить наличие упаковщика или средства защиты; зафиксировать результат.
5. Проанализировать структуру заголовков и таблицу секций, оценить энтропию секций, сделать вывод о наличии шифрования.
6. Выполнить дизассемблирование образца, построить граф потока управления основных функций.
7. Сопоставить результаты линейного и рекурсивного разбора, выявить участки, разбираемые по-разному.
8. Выявить и описать применённые приёмы противодействия дизассемблированию, указав для каждого признак обнаружения.
9. Выполнить запуск образца под отладчиком и зафиксировать поведение программы.
10. Выявить и описать применённые приёмы противодействия отладке, указав, каким образом программа обнаруживает отладочную среду.
11. Определить, применяется ли в образце контроль целостности собственного кода, и описать механизм его реализации.
12. Свести выявленные механизмы защиты в таблицу с оценкой стойкости каждого по трудоёмкости преодоления (квалификация, средства, время).
13. Оценить совокупную стойкость защиты образца и определить наиболее слабое звено.
14. Сформулировать не менее трёх предложений по усилению защиты исследованного образца с обоснованием ожидаемого эффекта.
15. Оформить отчёт и подготовиться к собеседованию по контрольным вопросам.

### **Варианты заданий**

*Таблица 10 – Варианты заданий*

| № варианта | Учебный образец                               | Исследуемый механизм                        |
|------------|-----------------------------------------------|---------------------------------------------|
| 1          | Образец 1: консольное приложение без упаковки | Обфускация потока управления                |
| 2          | Образец 2: упакованное консольное приложение  | Определение упаковщика, распаковка в памяти |

| № варианта | Учебный образец                                         | Исследуемый механизм                            |
|------------|---------------------------------------------------------|-------------------------------------------------|
| 3          | Образец 3: приложение с проверкой флага отладки         | Обнаружение отладочной среды                    |
| 4          | Образец 4: приложение с временными проверками           | Измерение времени выполнения участка            |
| 5          | Образец 5: приложение с контролем контрольной суммы     | Контроль целостности собственного кода          |
| 6          | Образец 6: приложение с ложными байтами после переходов | Противодействие линейному дизассемблированию    |
| 7          | Образец 7: приложение с преждевременным возвратом       | Противодействие рекурсивному дизассемблированию |
| 8          | Образец 8: приложение с самогенерируемым кодом          | Формирование кода во время выполнения           |
| 9          | Образец 9: приложение с лицензионным файлом             | Программная защита от копирования               |
| 10         | Образец 10: приложение с комбинированной защитой        | Совокупная оценка стойкости                     |

### Содержание отчёта

Отчёт оформляется в соответствии с общими требованиями, приведёнными во вводной части настоящих указаний, и дополнительно содержит все таблицы и заключения, предусмотренные порядком выполнения работы.

### Контрольные вопросы

1. В чём заключается задача анализа программных реализаций и чем определяется её актуальность?
2. На какие этапы разбивается решение задачи анализа программных реализаций?
3. Какие методы применяются для решения задачи анализа программных реализаций?
4. Каковы достоинства и недостатки статического метода анализа?
5. Каковы достоинства и недостатки динамического метода анализа?
6. Почему метод экспериментов с «чёрным ящиком» так называется?
7. В чём состоит суть метода экспериментов с «чёрным ящиком» и каковы его недостатки?
8. Какие типичные примеры анализа программ методом «чёрного ящика» вам известны?
9. Что такое обфускация и какие её виды применяются?
10. Каким образом упаковка исполняемого модуля препятствует статическому анализу?
11. На чём основана методика дизассемблирования с добавлением ложного байта?
12. Для какого алгоритма дизассемблирования характерна работа до достижения конца буфера?
13. Каким образом программа может обнаружить факт своей отладки?
14. Для чего используется инструкция чтения счётчика тактов процессора при противодействии отладке?
15. Что такое программные и аппаратные точки останова и чем они различаются?
16. Как электронная подпись исполняемого кода обеспечивает контроль его целостности?
17. Какие механизмы защиты от копирования применяются и в чём различие программных и программно-аппаратных решений?

18. Что такое эмуляция ключа и какие меры противодействия ей известны?

19. От чего зависит оценка стойкости механизма защиты и почему абсолютная защита недостижима?

**Рекомендуемая литература:** 1, 2, 3, 4 (см. общий список литературы)

## Практическое занятие № 5

### Контроль отсутствия недеklarированных возможностей в исходных текстах программ

**Соответствие рабочей программе:** Тема 5. Исследование программного обеспечения на предмет отсутствия недеklarированных возможностей.

**Формируемые индикаторы:** ИД-3 ПК-3, ИД-3 ПК-4, ИД-5 ПК-4.

**Объём:** 5 астр. ч.

**Цель работы:** сформировать навыки проведения статического анализа исходных текстов программ на предмет отсутствия недеklarированных возможностей и оформления результатов исследования.

#### Задачи работы:

- освоить состав работ по контролю отсутствия недеklarированных возможностей;
- приобрести навыки работы с инструментальными средствами статического анализа кода;
- научиться оформлять протокол испытаний и заключение по результатам исследования.

#### Теоретическая часть

##### 5.1. Недекларированные возможности и уровни контроля

Контроль отсутствия недеklarированных возможностей представляет собой совокупность работ, проводимых испытательной лабораторией в отношении программного обеспечения с целью подтверждения того, что оно не содержит функциональных возможностей, не описанных в документации и способных привести к нарушению безопасности информации.

Работы проводятся по уровням контроля, различающимся глубиной исследования. Чем строже уровень, тем больший объём проверок выполняется и тем шире состав представляемых разработчиком материалов. Наименее строгие уровни ограничиваются контролем состава документации и исходного состояния программного обеспечения, наиболее строгие требуют полного анализа исходных текстов, включая контроль фактических маршрутов выполнения.

Принципиальным условием проведения работ является наличие исходных текстов. Контроль отсутствия недеklarированных возможностей по бинарному коду принципиально ограничен, поскольку восстановление исходной логики по машинному коду неполно и неоднозначно.

Таблица 11 – Состав работ по уровням контроля (обобщённо)

| Вид работ                                           | Менее строгие уровни | Более строгие уровни             |
|-----------------------------------------------------|----------------------|----------------------------------|
| Контроль состава и содержания документации          | Выполняется          | Выполняется в расширенном объёме |
| Контроль исходного состояния ПО                     | Выполняется          | Выполняется                      |
| Статический анализ исходных текстов                 | В сокращённом объёме | В полном объёме                  |
| Контроль полноты и отсутствия избыточности          | Выполняется          | Выполняется                      |
| Контроль связей функциональных объектов             | Не выполняется       | Выполняется                      |
| Контроль маршрутов выполнения                       | Не выполняется       | Выполняется                      |
| Динамический анализ, контроль фактических маршрутов | Не выполняется       | Выполняется                      |
| Отчётность по результатам испытаний                 | Выполняется          | Выполняется                      |

### ***5.2. Контроль состава документации и исходного состояния***

Контроль состава и содержания документации предполагает проверку наличия всех предусмотренных документов и соответствия описания фактическому составу программного обеспечения. Именно документация задаёт эталон декларированных возможностей: возможность, не описанная в ней, является недеklarированной по определению, независимо от того, вредоносна она или нет.

Контроль исходного состояния программного обеспечения состоит в фиксации состава и характеристик представленных материалов: перечня файлов, их размеров, дат изменения и контрольных сумм. Зафиксированное исходное состояние служит основанием для последующей проверки того, что исследовались именно те материалы, которые были представлены, и что в ходе испытаний они не изменялись.

Отдельной проверкой является контроль соответствия исходных текстов загрузочному коду: сборка представленных исходных текстов в контролируемых условиях и сопоставление результата с представленным исполняемым файлом. Расхождение означает, что исполняемый файл содержит код, отсутствующий в исходных текстах.

### ***5.3. Статический анализ исходных текстов***

Контроль полноты и отсутствия избыточности исходных текстов направлен на выявление кода, не относящегося к реализации заявленных функций. Избыточными признаются неиспользуемые функции, недостижимые ветви, отладочные вставки, закомментированные фрагменты, содержащие функциональность. Каждый такой фрагмент подлежит объяснению разработчиком либо удалению.

Контроль связей функциональных объектов по управлению устанавливает, какие функции вызывают какие; результатом является граф вызовов. Контроль связей по информации устанавливает, какие данные передаются между функциями; результатом является граф потоков данных. Совместный анализ обоих графов позволяет выявить неочевидные пути передачи информации, в том числе к внешним интерфейсам.

Контроль маршрутов выполнения функциональных объектов состоит в построении перечня возможных путей выполнения программы и сопоставлении их с описанными в документации сценариями работы. Маршрут, приводящий к выполнению функции, не предусмотренной документацией, является признаком недеklarированной возможности.

Инструментальные средства статического анализа автоматизируют перечисленные работы, однако не заменяют экспертную оценку. Любой статический анализатор порождает как ложные срабатывания (сообщения о дефектах там, где их нет), так и пропуски. Обработка результатов включает классификацию срабатываний по критичности и разделение их на истинные и ложные с обоснованием каждого решения.

### ***5.4. Динамический анализ и покрытие кода***

Динамический анализ дополняет статический и состоит в выполнении программы с фиксацией фактически пройденных маршрутов. Сопоставление фактических маршрутов с построенным ранее перечнем возможных маршрутов позволяет выявить участки кода, не выполняющиеся ни при одном из штатных сценариев, – такие участки требуют особого внимания.

Покрытие кода характеризует долю программы, выполненную при проведении тестирования. Различают покрытие по операторам, по ветвям и по условиям; последнее является наиболее строгим. Низкое покрытие означает, что значительная часть кода не исследовалась, и снижает обоснованность вывода об отсутствии недеklarированных возможностей.

Фаззинг-тестирование состоит в подаче на вход программы искажённых, граничных и случайных данных с целью выявления некорректной обработки. Метод особенно результативен применительно к разборщикам форматов данных и сетевым протоколам, где ошибки обработки входных данных наиболее вероятны.

### ***5.5. Индивидуальные признаки программы и идентификация авторства***

Индивидуальные признаки программы – устойчивые особенности, отражающие индивидуальный стиль разработчика и применённые им средства. К ним относятся особенности оформления кода и именования идентификаторов, характерные конструкции и алгоритмические решения, содержание комментариев, служебные и отладочные строки, сведения о среде сборки, остающиеся в метаданных файла, орфографические особенности текстовых сообщений.

Совокупность индивидуальных признаков позволяет решать задачи установления авторства кода, выявления заимствований, а также обнаружения родства между различными программами. Отдельно взятый признак доказательного значения, как правило, не имеет; вывод формируется по совокупности.

При проведении экспертизы по фактам компьютерных инцидентов индивидуальные признаки программы сопоставляются с иными данными: сведениями об учётных записях, журналами доступа, временными характеристиками, что позволяет установить участников события, их роли, места и условия, при которых информация была создана, модифицирована или удалена.

### ***5.6. Оформление результатов исследования***

Результаты работ оформляются протоколом испытаний и заключением. Протокол фиксирует ход работ: перечень исследованных материалов, применённые средства и методики, выполненные проверки и полученные промежуточные результаты. Заключение содержит итоговый вывод.

Заключение должно включать: основание проведения исследования; сведения об объекте исследования и его исходном состоянии; перечень применённых методов и средств; описание выявленных особенностей программного обеспечения; вывод о наличии либо отсутствии недеklarированных возможностей; при выявлении – описание каждой из них с указанием условий активации и возможных последствий.

Формулировки заключения должны быть проверяемы. Вывод, не подкреплённый описанием конкретных проверок и их результатов, не имеет доказательного значения. Равным образом недопустима подмена вывода об отсутствии недеklarированных возможностей выводом об отсутствии выявленных недеklarированных возможностей без указания объёма проведённых проверок.

**Оборудование и материалы:** персональный компьютер с установленной операционной системой Astra Linux Special Edition (или РЕД ОС), средствами анализа кода AppChecker, Solar appScreener либо PT Application Inspector, программно-аппаратным комплексом «АК-ВС 3» (при наличии), офисным пакетом «Р7-Офис» либо «МойОфис Стандартный»; учебный программный проект, предоставленный преподавателем.

**Указания по технике безопасности:** к выполнению практических занятий допускаются обучающиеся, ознакомившиеся с правилами работы в лаборатории и прошедшие инструктаж по технике безопасности. Работы, связанные с анализом исполняемого кода, выполняются исключительно на учебных образцах, предоставленных преподавателем, в изолированном сегменте лабораторной сети. Подключение личных носителей информации и запуск посторонних программ не допускаются. Исследование чужого программного обеспечения без разрешения правообладателя недопустимо и влечёт ответственность, предусмотренную законодательством Российской Федерации.

## Порядок выполнения работы

1. Ознакомиться с теоретической частью настоящих указаний и рекомендуемой литературой.
2. Получить учебный программный проект согласно номеру варианта вместе с прилагаемой документацией.
3. Зафиксировать исходное состояние проекта: перечень файлов, их размеры и контрольные суммы; оформить результат в виде таблицы.
4. Провести контроль состава и содержания документации, сопоставить описанные в ней функции с фактическим составом проекта.
5. Составить перечень декларированных функций, подлежащих проверке.
6. Выполнить контроль полноты и отсутствия избыточности исходных текстов; зафиксировать выявленные неиспользуемые и недостижимые фрагменты.
7. Провести статический анализ исходных текстов имеющимся инструментальным средством, сохранить отчёт.
8. Классифицировать выявленные срабатывания по степени критичности.
9. Разделить срабатывания на истинные и ложные, для каждого решения привести обоснование.
10. Построить граф вызовов функций проекта, выявить функции, не вызываемые из основного потока выполнения.
11. Построить перечень маршрутов выполнения и сопоставить их со сценариями, описанными в документации.
12. Выявить участки кода, не соответствующие ни одному документированному сценарию; описать условия, при которых они выполняются.
13. Выполнить динамический анализ проекта, оценить покрытие кода тестами.
14. Выявить индивидуальные признаки исследуемой программы и оформить их перечень.
15. Оформить протокол испытаний с указанием применённых средств и выполненных проверок.
16. Составить заключение о наличии либо отсутствии недеklarированных возможностей с указанием объёма проведённых проверок.
17. Оформить отчёт и подготовиться к собеседованию по контрольным вопросам.

## Варианты заданий

Таблица 12 – Варианты заданий

| № варианта | Учебный программный проект                  | Особенность для выявления                  |
|------------|---------------------------------------------|--------------------------------------------|
| 1          | Утилита обработки текстовых файлов          | Неиспользуемые функции                     |
| 2          | Клиент сетевого сервиса                     | Недокументированное сетевое взаимодействие |
| 3          | Библиотека криптографических преобразований | Ослабленная генерация ключей               |
| 4          | Модуль разграничения доступа                | Недостижимая ветвь проверки прав           |
| 5          | Разборщик формата данных                    | Отсутствие проверки входных данных         |
| 6          | Служба журналирования событий               | Условная запись во внешний файл            |
| 7          | Утилита резервного копирования              | Отладочные вставки в рабочем коде          |
| 8          | Модуль аутентификации пользователей         | Служебная учётная запись в коде            |
| 9          | Обработчик очереди заданий                  | Скрытая передача данных между функциями    |
| 10         | Модуль формирования отчётов                 | Избыточность исходных текстов              |

## **Содержание отчёта**

Отчёт оформляется в соответствии с общими требованиями, приведёнными во вводной части настоящих указаний, и дополнительно содержит все таблицы и заключения, предусмотренные порядком выполнения работы.

## **Контрольные вопросы**

1. Что понимается под недекларированными возможностями программного обеспечения?
2. Какие виды работ входят в состав контроля отсутствия недекларированных возможностей?
3. Чем различаются уровни контроля и от чего зависит выбор уровня?
4. Почему проведение контроля требует наличия исходных текстов?
5. Какую роль играет документация при определении недекларированных возможностей?
6. В чём состоит контроль исходного состояния программного обеспечения?
7. Как проверяется соответствие исходных текстов загрузочному коду?
8. В чём состоит контроль полноты и отсутствия избыточности исходных текстов?
9. Что такое функциональный объект и маршрут его выполнения?
10. Чем контроль связей по управлению отличается от контроля связей по информации?
11. Какие задачи решает динамический анализ и что понимается под покрытием кода?
12. Какие виды покрытия кода различают и какое из них является наиболее строгим?
13. Что такое фаззинг-тестирование и в каких случаях оно наиболее результативно?
14. Что такое ложное срабатывание статического анализатора и как оно обрабатывается?
15. Какие индивидуальные признаки программы позволяют идентифицировать её автора?
16. Почему вывод об авторстве формируется по совокупности признаков?
17. Какие сведения должны быть отражены в протоколе испытаний и в заключении?
18. В чём различие между выводом об отсутствии недекларированных возможностей и выводом об отсутствии выявленных недекларированных возможностей?

**Рекомендуемая литература:** 1, 2, 4, 5, 6 (см. общий список литературы)

## **Практическое занятие № 6**

### **Анализ отечественной нормативной базы в области защиты программного обеспечения. Работа с Единым реестром российских программ**

**Соответствие рабочей программе:** Тема 6. Краткое описание отечественных нормативных актов, регламентирующих деятельность в области защиты программного обеспечения.

**Формируемые индикаторы:** ИД-3 ПК-4, ИД-5 ПК-4.

**Объём:** 4 астр. ч.

**Цель работы:** сформировать навыки применения отечественных нормативных правовых актов и национальных стандартов в области защиты программного обеспечения и подбора отечественных программных средств по Единому реестру.

#### **Задачи работы:**

- изучить состав отечественной нормативной базы в области защиты программного обеспечения;
- приобрести навыки извлечения требований из нормативных документов;
- освоить работу с Единым реестром российских программ для ЭВМ и баз данных.

#### **Теоретическая часть**

##### **6.1. Федеральное законодательство**

Основу правового регулирования составляет Федеральный закон «Об информации, информационных технологиях и о защите информации», устанавливающий понятия информации, информационных технологий, обладателя информации и оператора информационной системы, а также общие требования к защите информации и ответственность за их несоблюдение.

Федеральный закон «О безопасности критической информационной инфраструктуры Российской Федерации» определяет порядок категорирования объектов критической информационной инфраструктуры и требования по обеспечению их безопасности. Для программного обеспечения, применяемого на значимых объектах, устанавливаются повышенные требования, включая необходимость его соответствия требованиям по безопасности информации.

Федеральный закон «О персональных данных» устанавливает обязанности оператора по обеспечению безопасности персональных данных, что определяет требования к применяемым средствам защиты и к программному обеспечению, участвующему в их обработке.

Доктрина информационной безопасности Российской Федерации определяет национальные интересы в информационной сфере и стратегические направления их обеспечения, в том числе развитие отечественной отрасли информационных технологий и снижение зависимости от зарубежных программных решений.

##### **6.2. Документы ФСТЭК России**

ФСТЭК России является федеральным органом исполнительной власти, осуществляющим функции по обеспечению безопасности информации в ключевых системах информационной инфраструктуры и противодействию техническим разведкам. Издаваемые ФСТЭК России приказы устанавливают состав и содержание мер защиты информации для различных категорий информационных систем.

Наряду с приказами ФСТЭК России издаёт методические документы, определяющие порядок выполнения отдельных работ, – в частности, методику оценки угроз безопасности

информации, методику оценки уровня критичности уязвимостей, требования доверия к средствам защиты информации.

ФСТЭК России ведёт Банк данных угроз безопасности информации и государственный реестр сертифицированных средств защиты информации. Вторым источником имеет непосредственное практическое значение: он позволяет установить, какие средства защиты допустимо применять в информационной системе определённого класса.

### **6.3. Национальные стандарты**

Национальные стандарты задают технические требования и порядок проведения проверок. Их применение, как правило, добровольно, однако становится обязательным при ссылке на стандарт в нормативном правовом акте или в договоре.

*Таблица 13 – Основные национальные стандарты в области защиты программного обеспечения*

| Обозначение          | Наименование (кратко)                                                  | Область применения                       |
|----------------------|------------------------------------------------------------------------|------------------------------------------|
| ГОСТ Р 56939         | Разработка безопасного программного обеспечения. Общие требования      | Процессы разработки, состав мер          |
| ГОСТ Р 58412         | Угрозы безопасности информации при разработке программного обеспечения | Моделирование угроз этапа разработки     |
| ГОСТ Р ИСО/МЭК 15408 | Критерии оценки безопасности информационных технологий                 | Формализация требований и оценка изделий |
| ГОСТ Р 51583         | Порядок создания автоматизированных систем в защищённом исполнении     | Организация работ по созданию систем     |
| ГОСТ Р 50922         | Защита информации. Основные термины и определения                      | Терминологическая основа                 |

### **6.4. Сертификация средств защиты информации**

Сертификация представляет собой форму подтверждения соответствия средства защиты информации установленным требованиям, проводимую аккредитованными испытательными лабораториями и органами по сертификации. Результатом является сертификат соответствия с указанием требований, которым соответствует изделие, и условий его применения.

Для средства защиты информации существенны два параметра: класс защиты, определяющий функциональные возможности, и уровень доверия, определяющий глубину проведённых проверок. Оба параметра указываются в сертификате и подлежат сопоставлению с требованиями, предъявляемыми к информационной системе, в которой планируется применение изделия.

Применение несертифицированных средств защиты в информационных системах, для которых сертификация обязательна, является нарушением установленных требований и влечёт последствия, предусмотренные законодательством.

### **6.5. Единый реестр российских программ для ЭВМ и баз данных**

Единый реестр российских программ для электронных вычислительных машин и баз данных ведётся Минцифры России и содержит сведения о программных продуктах, отвечающих установленным требованиям к отечественному происхождению: исключительное право должно принадлежать российскому лицу, продукт должен свободно распространяться на территории Российской Федерации, а доля выплат иностранным лицам по лицензионным договорам не должна превышать установленного предела.

Наличие продукта в реестре имеет прямые правовые последствия. При закупках для государственных и муниципальных нужд установлен запрет на допуск программного обеспечения, происходящего из иностранных государств, при наличии в реестре

российского аналога с сопоставимыми характеристиками. Для значимых объектов критической информационной инфраструктуры установлены требования по преимущественному использованию отечественного программного обеспечения.

При работе с реестром следует различать факт включения продукта в реестр и факт его сертификации. Это независимые процедуры: включение в реестр подтверждает отечественное происхождение, сертификация – соответствие требованиям по безопасности информации. Средство защиты информации, применяемое в государственной информационной системе, должно, как правило, удовлетворять обоим условиям.

Подбор отечественного аналога зарубежного программного средства требует сопоставления не наименований, а функциональных характеристик: перечня решаемых задач, поддерживаемых форматов и протоколов, совместимости с применяемыми операционными системами, возможностей интеграции с существующими системами.

#### ***6.6. Ответственность за правонарушения в сфере компьютерной информации***

Уголовный кодекс Российской Федерации содержит главу, посвящённую преступлениям в сфере компьютерной информации. Статья 272 устанавливает ответственность за неправомерный доступ к охраняемой законом компьютерной информации, повлёкший её уничтожение, блокирование, модификацию либо копирование.

Статья 273 устанавливает ответственность за создание, распространение или использование компьютерных программ, заведомо предназначенных для несанкционированного уничтожения, блокирования, модификации, копирования компьютерной информации или нейтрализации средств её защиты. Существенно, что состав образует уже создание такой программы, независимо от факта её применения.

Статья 274 устанавливает ответственность за нарушение правил эксплуатации средств хранения, обработки или передачи компьютерной информации, а статья 274.1 – за неправомерное воздействие на критическую информационную инфраструктуру Российской Федерации.

Для специалиста по защите информации знание указанных норм имеет двоякое значение: оно необходимо как для квалификации действий нарушителя при расследовании инцидентов, так и для правильной организации собственной деятельности, поскольку многие применяемые в профессии средства и методы при использовании без надлежащих оснований образуют состав правонарушения.

**Оборудование и материалы:** персональный компьютер с установленной операционной системой Astra Linux Special Edition (или РЕД ОС), офисным пакетом «Р7-Офис» либо «МойОфис Стандартный», доступом к справочным правовым системам «КонсультантПлюс» или «Гарант», к официальному сайту ФСТЭК России и к Единому реестру российских программ.

**Указания по технике безопасности:** к выполнению практических занятий допускаются обучающиеся, ознакомившиеся с правилами работы в лаборатории и прошедшие инструктаж по технике безопасности. Работы, связанные с анализом исполняемого кода, выполняются исключительно на учебных образцах, предоставленных преподавателем, в изолированном сегменте лабораторной сети. Подключение личных носителей информации и запуск посторонних программ не допускаются. Исследование чужого программного обеспечения без разрешения правообладателя недопустимо и влечёт ответственность, предусмотренную законодательством Российской Федерации.

#### **Порядок выполнения работы**

1. Ознакомиться с теоретической частью настоящих указаний и рекомендуемой литературой.

2. Получить у преподавателя номер варианта и описание условной информационной системы.
3. Определить вид информационной системы, категорию обрабатываемой информации и применимый режим защиты.
4. Определить перечень нормативных правовых актов, применимых к защите программного обеспечения данной системы; оформить перечень с указанием полных реквизитов документов.
5. Определить перечень применимых национальных стандартов и обосновать необходимость их применения.
6. Извлечь из выбранных документов требования, предъявляемые к разработке и применению программного обеспечения, свести их в таблицу с указанием источника требования.
7. Определить, требуется ли для применяемых в системе средств защиты информации наличие сертификата соответствия и какого класса.
8. Составить перечень программных средств, необходимых для функционирования системы (операционная система, офисный пакет, средство антивирусной защиты, средство защиты от несанкционированного доступа, средство анализа защищённости).
9. Подобрать в Едином реестре российских программ отечественные средства, обеспечивающие выполнение выявленных требований; для каждого указать реестровую запись и правообладателя.
10. Проверить по государственному реестру ФСТЭК России наличие сертификатов соответствия у подобранных средств защиты информации.
11. Составить сравнительную таблицу «зарубежное средство – отечественный аналог» с сопоставлением функциональных характеристик и обоснованием возможности замещения.
12. Указать ограничения и риски, возникающие при переходе на отечественные аналоги, и предложить меры по их снижению.
13. Оформить отчёт и подготовиться к собеседованию по контрольным вопросам.

### Варианты заданий

Таблица 14 – Варианты заданий

| № варианта | Условная информационная система                              | Категория информации                               |
|------------|--------------------------------------------------------------|----------------------------------------------------|
| 1          | Информационная система органа исполнительной власти субъекта | Служебная информация ограниченного распространения |
| 2          | Информационная система медицинской организации               | Персональные данные специальных категорий          |
| 3          | Автоматизированная система управления энергообъекта          | Информация значимого объекта КИИ                   |
| 4          | Информационная система образовательной организации           | Персональные данные обучающихся                    |
| 5          | Информационная система кредитной организации                 | Банковская тайна, персональные данные              |
| 6          | Система электронного документооборота предприятия            | Коммерческая тайна                                 |
| 7          | Информационная система транспортного предприятия             | Информация объекта КИИ                             |
| 8          | Государственная информационная система предоставления услуг  | Персональные данные заявителей                     |
| 9          | Информационная система промышленного предприятия             | Коммерческая тайна, информация КИИ                 |
| 10         | Информационная система научной организации                   | Результаты интеллектуальной деятельности           |

## **Содержание отчёта**

Отчёт оформляется в соответствии с общими требованиями, приведёнными во вводной части настоящих указаний, и дополнительно содержит все таблицы и заключения, предусмотренные порядком выполнения работы.

## **Контрольные вопросы**

1. Какие федеральные законы составляют правовую основу защиты информации в Российской Федерации?
2. Какие обязанности возлагает на оператора Федеральный закон «О персональных данных»?
3. Что устанавливает Федеральный закон о безопасности критической информационной инфраструктуры?
4. Каково содержание Доктрины информационной безопасности Российской Федерации?
5. Какие функции выполняет ФСТЭК России в области защиты информации?
6. Какие вопросы регулируются приказами и методическими документами ФСТЭК России?
7. Каково назначение ГОСТ Р 56939 и ГОСТ Р 58412?
8. Что устанавливает ГОСТ Р ИСО/МЭК 15408?
9. В каких случаях применение национального стандарта становится обязательным?
10. Каков порядок сертификации средств защиты информации?
11. Чем различаются класс защиты и уровень доверия средства защиты информации?
12. Кто ведёт Единый реестр российских программ и каковы условия включения в него?
13. Какие правовые последствия влечёт наличие программного продукта в Едином реестре?
14. В чём различие между включением продукта в реестр и его сертификацией?
15. Каким образом производится подбор отечественного аналога зарубежного программного средства?
16. Какая ответственность предусмотрена статьями 272–274.1 Уголовного кодекса Российской Федерации?
17. Почему состав преступления по статье 273 образует уже создание вредоносной программы?

**Рекомендуемая литература:** 1, 2, 5, 6, 8 (см. общий список литературы)

### Список рекомендуемой литературы

1. Казарин, О. В. Программно-аппаратные средства защиты информации. Защита программного обеспечения : учебник и практикум для вузов / О. В. Казарин, А. С. Забабурин. – Москва : Издательство Юрайт, 2023. – 312 с.
2. Проскурин, В. Г. Защита программ и данных : учеб. пособие для вузов / В. Г. Проскурин. – Москва : Академия, 2012. – 208 с.
3. Хаулет, Т. Инструменты безопасности с открытым исходным кодом / Т. Хаулет. – 2-е изд., испр. – Москва : ИНТУИТ, 2016. – 566 с.
4. Девянин, П. Н. Модели безопасности компьютерных систем : учебное пособие / П. Н. Девянин. – Москва : Академия, 2005. – 144 с.
5. ГОСТ Р 56939. Защита информации. Разработка безопасного программного обеспечения. Общие требования.
6. ГОСТ Р 58412. Защита информации. Разработка безопасного программного обеспечения. Угрозы безопасности информации при разработке программного обеспечения.
7. Банк данных угроз безопасности информации ФСТЭК России. – URL: <https://bdu.fstec.ru>
8. Единый реестр российских программ для ЭВМ и баз данных. – URL: <https://reestr.digital.gov.ru>

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

по организации самостоятельной работы по дисциплине

«Защита программ и данных»

для студентов направления подготовки

10.04.01 «Информационная безопасность»

направленность (профиль)

«Математическое и программное обеспечение защиты информации»

Ставрополь 2026

## **СОДЕРЖАНИЕ**

### **Введение**

1. Общая характеристика самостоятельной работы обучающегося
2. Методические рекомендации по отдельным видам самостоятельной работы
3. Технологическая карта самостоятельной работы обучающегося
4. График самостоятельной работы обучающегося
5. Самостоятельное изучение тем дисциплины
6. Паспорт фонда оценочных средств для проверки самостоятельной работы
  - 6.1. Вопросы для собеседования по практическим занятиям
  - 6.2. Вопросы для собеседования по самостоятельному изучению тем дисциплины
  - 6.3. Примерные тестовые задания
7. Вопросы к промежуточной аттестации (зачёту)
8. Критерии оценивания результатов самостоятельной работы
9. Список рекомендуемой литературы

## **Введение**

Настоящие методические указания определяют порядок организации самостоятельной работы обучающихся по дисциплине «Защита программ и данных» и составлены в соответствии с рабочей программой дисциплины и учебным планом направления подготовки 10.04.01 «Информационная безопасность», направленность (профиль) «Математическое и программное обеспечение защиты информации», год начала подготовки 2026.

Объём самостоятельной работы по дисциплине составляет 54 астрономических часа в 1 семестре, что соответствует объёму контактной работы и отражает характер дисциплины: значительная часть учебного материала носит нормативный и справочный характер и подлежит проработке обучающимся самостоятельно.

Указания содержат перечень тем для самостоятельного изучения, задания по каждой теме, вопросы для самоконтроля, примерные тестовые задания и вопросы к промежуточной аттестации, а также критерии оценивания результатов.

### **1. Общая характеристика самостоятельной работы обучающегося**

Основными видами учебной работы по достижению результатов освоения дисциплины являются лекции, практические занятия и самостоятельная работа обучающихся. Лабораторные работы учебным планом по данной дисциплине не предусмотрены.

На лекциях раскрываются основные положения и понятия курса, формируются знания в области защиты программ и данных, развивается логическое мышление, формируется научное мировоззрение. На практических занятиях формируются умения и навыки, необходимые для решения задач профессиональной деятельности.

Приступая к изучению учебной дисциплины, необходимо ознакомиться с рабочей программой, получить в библиотеке рекомендованные учебные пособия, а также получить у ведущего преподавателя в электронном виде конспекты лекций и методические указания по проведению практических занятий.

Самостоятельная работа обучающихся по дисциплине включает следующие виды деятельности:

- самостоятельное изучение тем дисциплины по установленному перечню с составлением конспекта;
- работу с нормативными правовыми актами и национальными стандартами;
- подготовку к практическим занятиям, включая изучение теоретической части соответствующих методических указаний;
- оформление отчётов по выполненным практическим занятиям;
- подготовку к собеседованию по результатам практических занятий и по темам самостоятельного изучения;
- подготовку к компьютерному тестированию;
- подготовку к промежуточной аттестации.

В случае пропуска учебного занятия обучающийся может воспользоваться содержанием соответствующих разделов учебно-методического комплекса дисциплины для самоподготовки и освоения темы. Для самоконтроля используются вопросы, приведённые в разделе 6 настоящих указаний.

### **2. Методические рекомендации по отдельным видам самостоятельной работы**

#### **2.1. Работа с лекционным материалом**

Конспект лекции подлежит проработке в течение ближайших дней после занятия, пока сохраняется контекст изложения. Проработка состоит в восстановлении сокращений, дополнении конспекта сведениями из рекомендованной литературы и формулировании вопросов по неясным положениям.

Особое внимание следует уделять определениям. Дисциплина насыщена близкими по звучанию, но различными по содержанию понятиями (уязвимость и недекларированная возможность, класс защиты и уровень доверия, включение в реестр и сертификация). Смешение таких понятий является типичной ошибкой на промежуточной аттестации.

## ***2.2. Работа с нормативными правовыми актами и стандартами***

Нормативные документы следует изучать по действующим редакциям, получаемым из справочных правовых систем «КонсультантПлюс» или «Гарант» либо с официального сайта ФСТЭК России. Использование текстов, размещённых на неофициальных ресурсах, не допускается: они могут содержать устаревшие редакции.

При работе с документом целесообразно фиксировать его полные реквизиты (вид, дата, номер, наименование), поскольку в отчётах и заключениях ссылка на документ без реквизитов не имеет доказательного значения. Рекомендуется вести собственный перечень изученных документов с краткой характеристикой предмета регулирования каждого.

## ***2.3. Подготовка к практическому занятию***

Подготовка включает изучение теоретической части методических указаний к соответствующему занятию, повторение материала лекции по связанной теме и ознакомление с порядком выполнения работы. Обучающийся должен ясно представлять, какой результат он должен получить и в каком виде его оформить.

Готовность к занятию проверяется преподавателем; обучающийся, не изучивший теоретическую часть, к выполнению работы не допускается и выполняет её в дополнительно назначенное время.

## ***2.4. Оформление отчёта по практическому занятию***

Отчёт оформляется в соответствии с требованиями, приведёнными в методических указаниях по проведению практических занятий. Наиболее распространённой ошибкой является подмена описания результата набором снимков экрана: иллюстрация без пояснения того, что на ней показано и какой вывод из этого следует, не засчитывается.

Выводы по работе должны соотноситься с целью работы и содержать ответ на вопрос, достигнута ли она и какими средствами. Формулировки общего характера («работа выполнена, навыки получены») выводами не являются.

## ***2.5. Подготовка к собеседованию и тестированию***

Собеседование проводится по контрольным вопросам, приведённым в разделе 6 настоящих указаний, и разделяется на вопросы базового и повышенного уровня. Вопросы базового уровня проверяют знание определений и состава процедур, повышенного – понимание оснований принимаемых решений и умение обосновать выбор.

Подготовка к тестированию предполагает проработку всех тем дисциплины, поскольку тестовые задания охватывают материал равномерно. Примерные тестовые задания приведены в подразделе 6.3.

## ***2.6. Подготовка к промежуточной аттестации***

Промежуточная аттестация по дисциплине проводится в форме зачёта в 1 семестре. К зачёту допускаются обучающиеся, выполнившие и защитившие все практические занятия.

Подготовку целесообразно строить не по порядку вопросов, а по темам дисциплины, объединяя связанные вопросы. Перечень вопросов приведён в разделе 7 настоящих указаний.

### 3. Технологическая карта самостоятельной работы обучающегося

| Коды реализуемых компетенций, индикатора(ов)     | Вид деятельности обучающихся                                                                                                                                    | Средства и технологии оценки                                            | СРС | Контактная работа | Всего |
|--------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|-----|-------------------|-------|
| 1 семестр                                        |                                                                                                                                                                 |                                                                         |     |                   |       |
| ИД-2 ПК-3<br>ИД-3 ПК-3<br>ИД-3 ПК-4<br>ИД-5 ПК-4 | Самостоятельное изучение тем дисциплины; подготовка к практическим занятиям; оформление отчётов по практическим занятиям; подготовка к промежуточной аттестации | Собеседование, проверка конспектов и отчётов, компьютерное тестирование | 54  | –                 | 54    |
| Итого за 1 семестр                               |                                                                                                                                                                 |                                                                         | 54  | –                 | 54    |

### 4. График самостоятельной работы обучающегося

График распределяет объём самостоятельной работы по темам дисциплины и увязывает его со сроками представления результатов.

Таблица 1 – График самостоятельной работы

| № темы | Содержание самостоятельной работы                                          | Объём, астр. ч. | Форма отчётности               | Срок представления |
|--------|----------------------------------------------------------------------------|-----------------|--------------------------------|--------------------|
| 1      | Изучение темы 1, подготовка к практическому занятию № 1, оформление отчёта | 8               | Конспект, отчёт, собеседование | 2–3-я неделя       |
| 2      | Изучение темы 2, подготовка к практическому занятию № 2, оформление отчёта | 8               | Конспект, отчёт, собеседование | 4–5-я неделя       |
| 3      | Изучение темы 3, подготовка к практическому занятию № 3, оформление отчёта | 10              | Конспект, отчёт, собеседование | 6–8-я неделя       |
| 4      | Изучение темы 4, подготовка к практическому занятию № 4, оформление отчёта | 10              | Конспект, отчёт, собеседование | 9–11-я неделя      |
| 5      | Изучение темы 5, подготовка к практическому занятию № 5, оформление отчёта | 10              | Конспект, отчёт, собеседование | 12–14-я неделя     |
| 6      | Изучение темы 6, подготовка к практическому занятию № 6, оформление отчёта | 8               | Конспект, отчёт, собеседование | 15–16-я неделя     |
| –      | ИТОГО                                                                      | 54              | –                              | –                  |

### 5. Самостоятельное изучение тем дисциплины

Перечень тем для самостоятельного изучения соответствует разделу 5 рабочей программы дисциплины. По каждой теме составляется конспект и выполняется приведённое задание.

| № п/п | Темы для самостоятельного изучения                                                                  | Основная литература | Дополнительная литература | Интернет-ресурсы |
|-------|-----------------------------------------------------------------------------------------------------|---------------------|---------------------------|------------------|
| 1     | Тема 1. Введение в теорию и практику защиты программного обеспечения                                | 1, 2                | 1                         | 1–4              |
| 2     | Тема 2. Основания теории и практики защиты программного обеспечения                                 | 1, 2                | 1                         | 1–4              |
| 3     | Тема 3. Методы обеспечения технологической и эксплуатационной безопасности программного обеспечения | 1, 2                | 1                         | 1–4              |
| 4     | Тема 4. Защита программ от анализа и модификации                                                    | 1, 2                | 1                         | 1–4              |
| 5     | Тема 5. Исследование программного обеспечения на предмет отсутствия недеklarированных возможностей  | 1, 2                | 1                         | 1–4              |

| № п/п | Темы для самостоятельного изучения                                                                                                | Основная литература | Дополнительная литература | Интернет-ресурсы |
|-------|-----------------------------------------------------------------------------------------------------------------------------------|---------------------|---------------------------|------------------|
| 6     | Тема 6. Краткое описание отечественных нормативных актов, регламентирующих деятельность в области защиты программного обеспечения | 1, 2                | 1                         | 1–4              |

### **Задания по темам самостоятельного изучения:**

#### **Тема 1. Введение в теорию и практику защиты программного обеспечения**

Объём: 8 астр. ч. Задание: Составить конспект по составу программного обеспечения как объекта защиты; подготовить обзор пяти записей Банка данных угроз ФСТЭК России, относящихся к программному обеспечению.

#### **Тема 2. Основания теории и практики защиты программного обеспечения**

Объём: 8 астр. ч. Задание: Составить конспект по составу мер ГОСТ Р 56939; подготовить схему соотношения профиля защиты, задания по безопасности и оценочного уровня доверия.

#### **Тема 3. Методы обеспечения технологической и эксплуатационной безопасности программного обеспечения**

Объём: 10 астр. ч. Задание: Составить конспект по механизмам контроля целостности; подготовить сравнение режимов проведения контроля по полноте и накладным расходам.

#### **Тема 4. Защита программ от анализа и модификации**

Объём: 10 астр. ч. Задание: Составить конспект по приёмам противодействия анализу; подготовить таблицу соответствия приёмов защиты и способов их выявления.

#### **Тема 5. Исследование программного обеспечения на предмет отсутствия недеklarированных возможностей**

Объём: 10 астр. ч. Задание: Составить конспект по составу работ контроля отсутствия недеklarированных возможностей; подготовить проект структуры заключения по результатам исследования.

#### **Тема 6. Краткое описание отечественных нормативных актов, регламентирующих деятельность в области защиты программного обеспечения**

Объём: 8 астр. ч. Задание: Составить конспект по составу нормативной базы; подготовить перечень из пяти отечественных программных средств с указанием реестровых записей.

### **6. Паспорт фонда оценочных средств для проверки самостоятельной работы**

| Этап формирования компетенции (№ темы) | Тип контроля  | Вид контроля              | Компонент фонда оценочных средств                                     | Количество элементов, шт. |
|----------------------------------------|---------------|---------------------------|-----------------------------------------------------------------------|---------------------------|
| 1 семестр                              |               |                           |                                                                       |                           |
| 1–6                                    | Текущий       | Устный                    | Вопросы для собеседования по практическим занятиям                    | 98                        |
| 1–6                                    | Текущий       | Устный                    | Вопросы для собеседования по самостоятельному изучению тем дисциплины | 61                        |
| 1–6                                    | Текущий       | Компьютерное тестирование | Фонд тестовых заданий                                                 | 22                        |
| 1–6                                    | Промежуточный | Устный                    | Вопросы к зачёту                                                      | 50                        |

## **6.1. Вопросы для собеседования по практическим занятиям**

### **Практическое занятие № 1. Анализ угроз безопасности программного обеспечения и построение модели нарушителя**

*Базовый уровень:*

1. Что понимается под программным обеспечением как объектом защиты и какие элементы включаются в его состав?
2. Почему сторонние компоненты требуют отдельного рассмотрения при анализе угроз?
3. Перечислите этапы жизненного цикла программного обеспечения и характерные для них угрозы.
4. Чем обусловлен рост стоимости устранения дефекта по мере продвижения по жизненному циклу?
5. В чём различие между уязвимостью, недекларированной возможностью и программной закладкой?
6. Какой признак позволяет отличить недекларированную возможность от ошибки реализации?
7. Какие разделы содержит Банк данных угроз безопасности информации ФСТЭК России?

*Повышенный уровень:*

1. Какова структура записи об угрозе и записи об уязвимости в банке данных?
2. Что означает конкретизация угрозы применительно к объекту защиты?
3. Что такое модель нарушителя и какие сведения она содержит?
4. Что такое потенциал нарушителя и как он определяется?
5. При каких условиях угроза признаётся актуальной?
6. Каким образом обосновывается исключение угрозы из перечня актуальных?
7. Какие негативные последствия рассматриваются при оценке угроз?

### **Практическое занятие № 2. Оценка процессов разработки программного обеспечения на соответствие требованиям ГОСТ Р 56939. Формирование требований доверия**

*Базовый уровень:*

1. Какое программное обеспечение считается доверенным и что понимается под свидетельствами доверия?
2. Почему доверие невозможно установить исключительно по готовому продукту?
3. Назовите основные группы мер, установленных ГОСТ Р 56939.
4. Какие свидетельства формируются при реализации мер безопасной разработки?
5. В чём состоит различие между статическим и динамическим анализом кода?
6. Для чего проводится моделирование угроз на этапе анализа требований?
7. Что такое экспертиза исходных текстов и почему её проводит не автор кода?
8. Сколько уровней доверия установлено требованиями ФСТЭК России и как они соотносятся по строгости?

*Повышенный уровень:*

1. От чего зависит выбор требуемого уровня доверия к изделию?
2. Что такое профиль защиты и чем он отличается от задания по безопасности?
3. Что задаёт оценочный уровень доверия по ГОСТ Р ИСО/МЭК 15408?

4. Какую роль играет управление конфигурацией в обеспечении доверия?
5. Как организуется обработка обращений пользователей об обнаруженных уязвимостях?
6. В чём состоит содержание субъектно-ориентированной модели компьютерной системы?
7. Что такое скрытый канал передачи информации и какие виды скрытых каналов известны?

### **Практическое занятие № 3. Обеспечение эксплуатационной безопасности программного обеспечения: контроль целостности и замкнутая программная среда**

#### *Базовый уровень:*

1. В чём различие между технологической и эксплуатационной безопасностью программного обеспечения?
2. Почему технологические и эксплуатационные меры не заменяют друг друга?
3. Как реализуется механизм контроля целостности программных компонентов?
4. Какие объекты подлежат контролю целостности и почему перечень эталонов также нуждается в защите?
5. Охарактеризуйте режимы проведения контроля целостности и области их применения.
6. Какие действия должны выполняться при обнаружении нарушения целостности?
7. Что такое замкнутая программная среда и на чём основана её реализация?
8. Почему замкнутая программная среда эффективна против вредоносного программного обеспечения, не обнаруживаемого антивирусом?

#### *Повышенный уровень:*

1. Какие подготовительные работы необходимы перед включением замкнутой программной среды?
2. Какую роль играет доверенная загрузка в обеспечении безопасности?
3. Как организуется изоляция процессов в современных операционных системах?
4. Перечислите этапы процесса управления уязвимостями.
5. Почему инвентаризация является основой управления уязвимостями?
6. Что понимается под оценкой применимости уязвимости и как она обосновывается?
7. Почему обновления должны устанавливаться только из доверенного источника?

### **Практическое занятие № 4. Исследование механизмов защиты программ от анализа и модификации**

#### *Базовый уровень:*

1. В чём заключается задача анализа программных реализаций и чем определяется её актуальность?
2. На какие этапы разбивается решение задачи анализа программных реализаций?
3. Какие методы применяются для решения задачи анализа программных реализаций?
4. Каковы достоинства и недостатки статического метода анализа?
5. Каковы достоинства и недостатки динамического метода анализа?
6. Почему метод экспериментов с «чёрным ящиком» так называется?
7. В чём состоит суть метода экспериментов с «чёрным ящиком» и каковы его недостатки?

8. Какие типичные примеры анализа программ методом «чёрного ящика» вам известны?
9. Что такое обфускация и какие её виды применяются?
10. Каким образом упаковка исполняемого модуля препятствует статическому анализу?

*Повышенный уровень:*

1. На чём основана методика дизассемблирования с добавлением ложного байта?
2. Для какого алгоритма дизассемблирования характерна работа до достижения конца буфера?
3. Каким образом программа может обнаружить факт своей отладки?
4. Для чего используется инструкция чтения счётчика тактов процессора при противодействии отладке?
5. Что такое программные и аппаратные точки останова и чем они различаются?
6. Как электронная подпись исполняемого кода обеспечивает контроль его целостности?
7. Какие механизмы защиты от копирования применяются и в чём различие программных и программно-аппаратных решений?
8. Что такое эмуляция ключа и какие меры противодействия ей известны?
9. От чего зависит оценка стойкости механизма защиты и почему абсолютная защита недостижима?

## **Практическое занятие № 5. Контроль отсутствия недекларированных возможностей в исходных текстах программ**

*Базовый уровень:*

1. Что понимается под недекларированными возможностями программного обеспечения?
2. Какие виды работ входят в состав контроля отсутствия недекларированных возможностей?
3. Чем различаются уровни контроля и от чего зависит выбор уровня?
4. Почему проведение контроля требует наличия исходных текстов?
5. Какую роль играет документация при определении недекларированных возможностей?
6. В чём состоит контроль исходного состояния программного обеспечения?
7. Как проверяется соответствие исходных текстов загрузочному коду?
8. В чём состоит контроль полноты и отсутствия избыточности исходных текстов?
9. Что такое функциональный объект и маршрут его выполнения?

*Повышенный уровень:*

1. Чем контроль связей по управлению отличается от контроля связей по информации?
2. Какие задачи решает динамический анализ и что понимается под покрытием кода?
3. Какие виды покрытия кода различают и какое из них является наиболее строгим?
4. Что такое фаззинг-тестирование и в каких случаях оно наиболее результативно?
5. Что такое ложное срабатывание статического анализатора и как оно обрабатывается?
6. Какие индивидуальные признаки программы позволяют идентифицировать её автора?

7. Почему вывод об авторстве формируется по совокупности признаков?
8. Какие сведения должны быть отражены в протоколе испытаний и в заключении?
9. В чём различие между выводом об отсутствии недеklarированных возможностей и выводом об отсутствии выявленных недеklarированных возможностей?

## **Практическое занятие № 6. Анализ отечественной нормативной базы в области защиты программного обеспечения. Работа с Единым реестром российских программ**

### *Базовый уровень:*

1. Какие федеральные законы составляют правовую основу защиты информации в Российской Федерации?
2. Какие обязанности возлагает на оператора Федеральный закон «О персональных данных»?
3. Что устанавливает Федеральный закон о безопасности критической информационной инфраструктуры?
4. Каково содержание Доктрины информационной безопасности Российской Федерации?
5. Какие функции выполняет ФСТЭК России в области защиты информации?
6. Какие вопросы регулируются приказами и методическими документами ФСТЭК России?
7. Каково назначение ГОСТ Р 56939 и ГОСТ Р 58412?
8. Что устанавливает ГОСТ Р ИСО/МЭК 15408?
9. В каких случаях применение национального стандарта становится обязательным?

### *Повышенный уровень:*

1. Каков порядок сертификации средств защиты информации?
2. Чем различаются класс защиты и уровень доверия средства защиты информации?
3. Кто ведёт Единый реестр российских программ и каковы условия включения в него?
4. Какие правовые последствия влечёт наличие программного продукта в Едином реестре?
5. В чём различие между включением продукта в реестр и его сертификацией?
6. Каким образом производится подбор отечественного аналога зарубежного программного средства?
7. Какая ответственность предусмотрена статьями 272–274.1 Уголовного кодекса Российской Федерации?
8. Почему состав преступления по статье 273 образует уже создание вредоносной программы?

## **6.2. Вопросы для собеседования по самостоятельному изучению тем дисциплины «Защита программ и данных»**

### **Тема 1. Введение в теорию и практику защиты программного обеспечения**

1. Программное обеспечение как объект защиты: состав и границы.
2. Защита цепочки поставки программного обеспечения.
3. Угрозы безопасности программного обеспечения на этапах жизненного цикла.
4. Зависимость стоимости устранения дефекта от этапа его выявления.
5. Соотношение понятий «уязвимость», «недекларированная возможность», «программная закладка».

6. Классификация вредоносного программного обеспечения.
7. Структура и назначение Банка данных угроз безопасности информации ФСТЭК России.
8. Модель нарушителя: виды, категории, потенциал.
9. Условия признания угрозы актуальной.

## **Тема 2. Основания теории и практики защиты программного обеспечения**

1. Формальные модели безопасности компьютерных систем и их применение к программному обеспечению.
2. Дискреционные, мандатные и ролевые модели разграничения доступа.
3. Субъектно-ориентированная модель компьютерной системы; операция порождения субъекта доступа.
4. Модели взаимодействия программной закладки с атакуемой системой.
5. Скрытые каналы передачи информации и их виды.
6. Понятие доверенного программного обеспечения и свидетельств доверия.
7. Уровни доверия и их соотношение по строгости требований.
8. Жизненный цикл безопасной разработки программного обеспечения.
9. Профиль защиты, задание по безопасности, оценочный уровень доверия.

## **Тема 3. Методы обеспечения технологической и эксплуатационной безопасности программного обеспечения**

1. Соотношение технологической и эксплуатационной безопасности.
2. Защита среды разработки и доверенные репозитории.
3. Механизмы контроля целостности программных компонентов.
4. Режимы проведения контроля целостности.
5. Замкнутая программная среда: назначение и реализация.
6. Доверенная загрузка вычислительной системы.
7. Изоляция процессов и разграничение доступа.
8. Организация антивирусной защиты; антивирусный мониторинг информационных потоков.
9. Управление уязвимостями: этапы и результаты.
10. Порядок установки обновлений из доверенного источника.

## **Тема 4. Защита программ от анализа и модификации**

1. Задача анализа программных реализаций и этапы её решения.
2. Статический и динамический методы анализа: достоинства и недостатки.
3. Метод экспериментов с «чёрным ящиком»: сущность, применение, недостатки.
4. Обфускация кода и данных.
5. Упаковка и шифрование исполняемых модулей.
6. Самогенерируемый код.
7. Приёмы противодействия линейному и рекурсивному дизассемблированию.
8. Способы обнаружения программой отладочной среды.
9. Программные и аппаратные точки останова.
10. Контроль целостности исполняемого кода и электронная подпись.

11. Защита программ от копирования; эмуляция электронных ключей.
12. Оценка стойкости механизмов защиты.

#### **Тема 5. Исследование программного обеспечения на предмет отсутствия недекларированных возможностей**

1. Уровни контроля отсутствия недекларированных возможностей.
2. Контроль состава и содержания документации.
3. Контроль исходного состояния программного обеспечения.
4. Контроль соответствия исходных текстов загрузочному коду.
5. Контроль полноты и отсутствия избыточности исходных текстов.
6. Контроль связей функциональных объектов по управлению и по информации.
7. Контроль маршрутов выполнения функциональных объектов.
8. Покрываемость кода: виды и значение при оценке полноты исследования.
9. Фаззинг-тестирование.
10. Инструментальные средства статического анализа; обработка ложных срабатываний.
11. Индивидуальные признаки программы и идентификация авторства кода.
12. Состав протокола испытаний и заключения по результатам исследования.

#### **Тема 6. Краткое описание отечественных нормативных актов, регламентирующих деятельность в области защиты программного обеспечения**

1. Федеральное законодательство в области защиты информации.
2. Требования по безопасности критической информационной инфраструктуры.
3. Доктрина информационной безопасности Российской Федерации.
4. Приказы и методические документы ФСТЭК России.
5. Национальные стандарты в области разработки безопасного программного обеспечения.
6. Порядок сертификации средств защиты информации; класс защиты и уровень доверия.
7. Единый реестр российских программ: назначение и условия включения.
8. Правовые последствия включения продукта в Единый реестр.
9. Ответственность за правонарушения в сфере компьютерной информации.

#### **6.3. Примерные тестовые задания**

Тестовые задания приведены с указанием правильного ответа и предназначены для самоконтроля.

*Таблица 2 – Примерные тестовые задания*

| № | Содержание задания                                                                         | Правильный ответ                                       |
|---|--------------------------------------------------------------------------------------------|--------------------------------------------------------|
| 1 | В чём заключается задача анализа программных реализаций?                                   | Сбор детальной информации о функционировании программы |
| 2 | Какие методы применяются для решения задачи анализа программных реализаций?                | Статический, динамический, метод «чёрного ящика»       |
| 3 | Инструкция чтения счётчика тактов процессора при противодействии отладке используется для: | сверки времени выполнения участка кода                 |
| 4 | Для какого алгоритма дизассемблирования характерна работа до достижения конца буфера?      | линейного дизассемблирования                           |
| 5 | Методика дизассемблирования с добавлением ложного байта заключается в:                     | добавлении после условных переходов байтов с данными   |

| №  | Содержание задания                                                                                                | Правильный ответ                                             |
|----|-------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| 6  | Какая инструкция помещает в стек адрес возврата?                                                                  | call                                                         |
| 7  | Использование какой инструкции позволяет затруднить дизассемблирование путём преждевременного завершения функции? | retn                                                         |
| 8  | Для противодействия проверке контрольной суммы кода можно использовать:                                           | аппаратные точки останова                                    |
| 9  | Программная точка останова реализуется путём:                                                                     | замены первого байта инструкции специальным кодом прерывания |
| 10 | Что подтверждает электронная подпись исполняемого файла?                                                          | происхождение и неизменность кода с момента подписания       |
| 11 | Недекларированные возможности – это:                                                                              | функциональные возможности, не описанные в документации      |
| 12 | Программная закладка отличается от прочих недекларированных возможностей:                                         | наличием умысла при её внесении                              |
| 13 | Какой документ задаёт эталон декларированных возможностей программы?                                              | эксплуатационная документация                                |
| 14 | Контроль полноты и отсутствия избыточности исходных текстов направлен на выявление:                               | кода, не относящегося к реализации заявленных функций        |
| 15 | Наиболее строгим видом покрытия кода является покрытие:                                                           | по условиям                                                  |
| 16 | Замкнутая программная среда обеспечивает:                                                                         | запуск только программ из заранее определённого перечня      |
| 17 | Какой источник ведёт ФСТЭК России для учёта угроз и уязвимостей?                                                  | Банк данных угроз безопасности информации                    |
| 18 | Угроза признаётся актуальной при наличии:                                                                         | источника, способа реализации и негативных последствий       |
| 19 | Единый реестр российских программ ведёт:                                                                          | Минцифры России                                              |
| 20 | Включение продукта в Единый реестр подтверждает:                                                                  | отечественное происхождение продукта                         |
| 21 | Ответственность за создание вредоносных компьютерных программ установлена:                                        | статьёй 273 Уголовного кодекса Российской Федерации          |
| 22 | ГОСТ Р 56939 устанавливает требования к:                                                                          | разработке безопасного программного обеспечения              |

## 7. Вопросы к промежуточной аттестации (зачёту)

1. Программное обеспечение как объект защиты: состав, границы, особенности.
2. Угрозы безопасности программного обеспечения на этапах жизненного цикла.
3. Уязвимость, недекларированная возможность, программная закладка: определения и различия.
4. Классификация вредоносного программного обеспечения.
5. Банк данных угроз безопасности информации ФСТЭК России: структура и порядок использования.
6. Модель нарушителя: виды, категории, потенциал.
7. Порядок определения актуальных угроз безопасности информации.
8. Формальные модели безопасности компьютерных систем.
9. Субъектно-ориентированная модель компьютерной системы.
10. Модели взаимодействия программной закладки с атакуемой системой.
11. Скрытые каналы передачи информации.
12. Понятие доверенного программного обеспечения и свидетельств доверия.
13. Состав мер безопасной разработки программного обеспечения по ГОСТ Р 56939.
14. Уровни доверия к средствам защиты информации.
15. Профиль защиты, задание по безопасности, оценочный уровень доверия.
16. Технологическая безопасность программного обеспечения: состав мер.

17. Эксплуатационная безопасность программного обеспечения: состав мер.
18. Контроль целостности программных компонентов: механизмы и режимы.
19. Замкнутая программная среда: назначение и реализация.
20. Доверенная загрузка вычислительной системы.
21. Изоляция процессов и разграничение доступа.
22. Средства и методы защиты от программных закладок.
23. Антивирусный мониторинг информационных потоков.
24. Организационные и административные меры антивирусной защиты.
25. Управление уязвимостями и обновлениями программного обеспечения.
26. Задача анализа программных реализаций: постановка и этапы решения.
27. Статический метод анализа программ: содержание, средства, ограничения.
28. Динамический метод анализа программ: содержание, средства, ограничения.
29. Метод экспериментов с «чёрным ящиком».
30. Обфускация кода и данных.
31. Упаковка и шифрование исполняемых модулей; самогенерируемый код.
32. Приёмы противодействия дизассемблированию.
33. Приёмы противодействия отладке.
34. Контроль целостности исполняемого кода; электронная подпись кода.
35. Защита программ от копирования; программно-аппаратные механизмы.
36. Оценка стойкости механизмов защиты программного обеспечения.
37. Контроль отсутствия недеklarированных возможностей: уровни и состав работ.
38. Контроль исходного состояния программного обеспечения и соответствия исходных текстов загрузочному коду.
39. Контроль полноты и отсутствия избыточности исходных текстов.
40. Контроль связей функциональных объектов и маршрутов выполнения.
41. Динамический анализ и покрытие кода; фаззинг-тестирование.
42. Инструментальные средства анализа защищённости и поиска уязвимостей.
43. Индивидуальные признаки программы и идентификация авторства кода.
44. Оформление результатов исследования: протокол испытаний и заключение.
45. Федеральное законодательство в области защиты информации.
46. Документы ФСТЭК России в области защиты информации.
47. Национальные стандарты в области разработки безопасного программного обеспечения.
48. Порядок сертификации средств защиты информации.
49. Единый реестр российских программ для ЭВМ и баз данных.
50. Ответственность за правонарушения в сфере компьютерной информации.

## **8. Критерии оценивания результатов самостоятельной работы**

Оценивание результатов самостоятельной работы проводится по совокупности представленных конспектов, отчётов по практическим занятиям и результатов собеседования.

«Зачтено» выставляется обучающемуся, выполнившему все виды самостоятельной работы, представившему конспекты по всем темам и защитившему все отчёты по практическим занятиям, знающему основные положения дисциплины, способному

обосновать применяемые методы и средства защиты программного обеспечения, а также сослаться на соответствующие нормативные документы.

«Не зачтено» выставляется обучающемуся, не выполнившему предусмотренные виды самостоятельной работы либо не представившему конспекты и отчёты, не знающему основных положений дисциплины и не способному дать ответ на вопросы базового уровня.

Отдельные виды самостоятельной работы оцениваются следующим образом: конспект по теме считается выполненным, если он охватывает все вопросы, приведённые в подразделе 6.2 для соответствующей темы; отчёт по практическому занятию оценивается по критериям, приведённым в методических указаниях по проведению практических занятий; результат тестирования считается положительным при доле правильных ответов не менее шестидесяти процентов.

## **9. Список рекомендуемой литературы**

### **9.1. Основная литература**

1. Казарин, О. В. Программно-аппаратные средства защиты информации. Защита программного обеспечения : учебник и практикум для вузов / О. В. Казарин, А. С. Забабурин. – Москва : Издательство Юрайт, 2023. – 312 с.

2. Проскурин, В. Г. Защита программ и данных : учеб. пособие для вузов / В. Г. Проскурин. – Москва : Академия, 2012. – 208 с.

3. Хаулет, Т. Инструменты безопасности с открытым исходным кодом / Т. Хаулет. – 2-е изд., испр. – Москва : ИНТУИТ, 2016. – 566 с.

### **9.2. Дополнительная литература**

1. Девянин, П. Н. Модели безопасности компьютерных систем : учебное пособие / П. Н. Девянин. – Москва : Академия, 2005. – 144 с.

2. ГОСТ Р 56939. Защита информации. Разработка безопасного программного обеспечения. Общие требования.

3. ГОСТ Р 58412. Защита информации. Разработка безопасного программного обеспечения. Угрозы безопасности информации при разработке программного обеспечения.

### **9.3. Интернет-ресурсы**

1. Электронный каталог научной библиотеки СКФУ. – URL: <http://catalog.ncfu.ru/catalog/ncfu>

2. ЭБС «Университетская библиотека онлайн». – URL: <http://biblioclub.ru>

3. Банк данных угроз безопасности информации ФСТЭК России. – URL: <https://bdu.fstec.ru>

4. Единый реестр российских программ для ЭВМ и баз данных. – URL: <https://reestr.digital.gov.ru>

### **9.4. Материально-техническое и программное обеспечение**

Помещения для самостоятельной работы обучающихся, оснащённые компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к электронной информационно-образовательной среде СКФУ. Программное обеспечение – в соответствии с разделом 9 рабочей программы дисциплины: отечественные программные средства, включённые в Единый реестр российских программ для ЭВМ и баз данных, а также свободно распространяемое программное обеспечение, используемое в учебных целях.