

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Узунов Федор Владимирович

Должность: Ректор

Дата подписания: 30.09.2026 08:54:43

Уникальный программный ключ:

fd935d10451b860e912264c0378f8448457b6d6607f94788008e39877e6bcb55

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
«ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ»
«УНИВЕРСИТЕТ ЭКОНОМИКИ И УПРАВЛЕНИЯ»**

Факультет экономики, управления и юриспруденции

Кафедра «Управление и бизнес-информатика»



УТВЕРЖДАЮ

Проректор по учебно-методической работе

Г.П. Узунова Г.П. Узунова

«02» февраля 2026 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

ЗАЩИЩЕННЫЕ МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ

Направление подготовки

10.03.01 «Информационная безопасность»

Профиль

Безопасность компьютерных систем

Квалификация выпускника: бакалавр

Для всех

форм обучения

Симферополь, 2026 г.

1. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ

Средства оценивания в ходе текущего контроля:

- устные опросы в ходе семинарских занятий;
- рефераты;
- тестирование;
- практические и ситуационные задания, выполняемые в ходе семинарского занятия или рекомендуемые для самостоятельной работы.

Коды компетенции	Результаты освоения ОПОП	Перечень планируемых результатов обучения по дисциплине
ПК-6	Способен организовывать технологический процесс защиты информации ограниченного доступа в соответствии с нормативными правовыми актами и нормативными	ПК-6.1. Знает правовые основы организации защиты государственной тайны и конфиденциальной информации, задачи органов защиты государственной тайны и служб защиты информации на предприятиях, нормативные правовые акты и стандарты по лицензированию деятельности в области обеспечения защиты государственной тайны, технической защиты конфиденциальной информации, нормативные методические документы ФСБ России, ФСТЭК России в области защиты информации; ПК-6.2. Владеет навыками организации и обеспечения режима секретности

1.1 Вопросы к текущему контролю

1. Концепция «Secure by Design» в мобильной разработке. В чем отличие подхода безопасности «с нуля» от добавления защиты на этапе тестирования?
2. Модель угроз OWASP Mobile Top 10. Назовите 5 наиболее критичных категорий уязвимостей и приведите пример для каждой.
3. Различия в моделях безопасности iOS и Android. Сравните механизмы «песочницы» (sandbox), управления правами (Permissions) и проверки приложений (App Review vs Google Play Protect).
4. Моделирование угроз (Threat Modeling) для мобильного приложения. Опишите этапы процесса (по методологии STRIDE) на примере приложения мобильного банкинга.
5. Безопасность архитектуры клиент-сервер. Почему логика проверки прав доступа (Access Control) никогда не должна полностью полагаться на мобильное устройство?
6. Небезопасное хранение данных (Insecure Data Storage). Сравните риски хранения данных в Shared Preferences (Android) / UserDefaults (iOS) и в SQLite/Realm базах данных без шифрования.
7. Хранение криптографических ключей. Сравните использование KeyStore (Android) и Keychain (iOS) с хранением ключей в коде или конфигурационных файлах.
8. Android KeyStore System и iOS Keychain. Как эти механизмы обеспечивают защиту ключей даже при компрометации операционной системы (Root/Jailbreak)?
9. Скрытие данных в файловой системе. Почему удаление файлов стандартными средствами не гарантирует их уничтожение и как работает «Data Wiping» (затирание)?

10. Локальная аутентификация (Biometrics). Как правильно интегрировать биометрию (FaceID/TouchID/Fingerprint) с использованием CryptoObject для защиты критических операций?
11. Анализ трафика мобильного приложения. Почему использование HTTP вместо HTTPS недопустимо? Опишите риски перехвата данных в публичных Wi-Fi сетях.
12. Pinning SSL/TLS сертификатов (Certificate/Public Key Pinning). Как этот механизм защищает от атак с использованием поддельных доверенных сертификатов и каковы сложности его поддержки?
13. Безопасность API (Server-Side). Почему проверка лимитов (Rate Limiting) и фильтрация входных данных на сервере критически важны, даже если клиентское приложение «безопасно»?
14. Атака Man-in-the-Middle (MITM) на мобильное приложение. Как злоумышленник может обойти проверку сертификата в приложении, если она реализована некорректно (например, Accept All Certificates)?
15. Использование WebView. Какие риски несет включение JavaScript-интерфейса (addJavascriptInterface) и как правильно настроить безопасный WebView?
16. Управление сессиями. Почему использование случайного идентификатора сессии (Session ID) в заголовках предпочтительнее передачи логина/пароля при каждом запросе?
17. Недостатки реализации собственного механизма аутентификации. Почему разработчикам не рекомендуется писать свою систему входа, а следует использовать OAuth 2.0 / OpenID Connect?
18. JWT (JSON Web Tokens) в мобильных приложениях. Где безопасно хранить токен доступа (Access Token) и токен обновления (Refresh Token)? Почему хранить Refresh Token в UserDefaults опасно?
19. Авторизация на стороне клиента (Client-Side Authorization). В чем опасность доверять UI мобильного приложения (например, скрывать кнопку «Админка», если пользователь не админ)?
20. Многофакторная аутентификация (MFA). Сравните эффективность SMS-кодов, Push-уведомлений и TOTP (Google Authenticator) для защиты мобильного приложения.
21. Реверс-инжиниринг (Reverse Engineering). Зачем злоумышленники декомпилируют APK/IPA файлы? Какие секреты (ключи API, алгоритмы) они ищут?
22. Обфускация кода. Как инструменты ProGuard/R8 (Android) и Swift Obfuscator (iOS) защищают приложение от анализа и взлома логики?
23. Небезопасное хранение секретов в коде. Почему жестко закодированные (hardcoded) ключи API в исходном коде считаются критической уязвимостью, даже если код скомпилирован?
24. Root/Jailbreak Detection. Почему обнаружение взлома устройства важно для банковских приложений и какие методы обхода (bypass) используют хакеры?
25. Dynamic Instrumentation (Frida/Ghidra). Как злоумышленники могут изменять логику работы запущенного приложения «на лету» (например, заставить функцию isPremiumUser() всегда возвращать true)?
26. Защита от Overlay-атак (Tapjacking). Как в Android вредоносное приложение может нарисовать невидимое окно поверх легитимного для кражи нажатий? Как этому противодействовать?
27. Безопасность IPC (Inter-Process Communication). Риски неправильной настройки Intent (Android) или Custom URL Schemes (iOS), позволяющие другим приложениям вызывать функции вашего.
28. Защита от отладки (Anti-Debugging). Как приложение может определить, что его пытаются запустить в отладчике, и зачем это нужно?
29. Push-уведомления. Являются ли они безопасным каналом для передачи конфиденциальных данных (например, кодов подтверждения)?

30. Безопасность сборки и CI/CD. Риски утечки секретов через build.gradle или Info.plist в системе автоматической сборки и способы их защиты (Secrets Management).

1.2. Темы рефератов

1. Сравнительный анализ моделей безопасности iOS и Android. Сравнение механизмов «песочницы» (Sandbox), управления привилегиями и процессов проверки приложений в App Store и Google Play.
2. Применение методологии STRIDE для моделирования угроз мобильного приложения для интернет-банкинга.
3. Концепция «Secure by Design» в мобильной разработке: интеграция практик безопасности на этапе формирования требований и проектирования архитектуры.
4. Особенности безопасности гибридных мобильных приложений (React Native, Flutter) по сравнению с нативной разработкой (Swift/Kotlin).
5. Анализ поверхности атаки (Attack Surface Analysis) современного мессенджера: векторы угроз от UI до фоновых сервисов.
6. Небезопасное хранение данных (Insecure Data Storage): анализ рисков использования Shared Preferences (Android) и UserDefaults (iOS) для хранения токенов и персональных данных.
7. Использование аппаратных модулей безопасности: сравнение Android Keystore System и iOS Keychain для хранения криптографических ключей.
8. Локальное шифрование баз данных. Сравнительный анализ SQLCipher и Realm Database Encryption для защиты данных в покое.
9. Безопасная реализация биометрической аутентификации: использование CryptoObject в Android и LAContext в iOS для привязки криптографических операций к биометрии.
10. Проблема остаточных данных: анализ информации, остающейся в кэше клавиатуры, снимках экрана и оперативной памяти после закрытия приложения.
11. Анализ перехвата трафика мобильных приложений: методы обхода SSL Pinning (Certificate/Public Key Pinning) с использованием инструментов Frida и Burp Suite.
12. Безопасность API в мобильных приложениях: уязвимости Broken Object Level Authorization (BOLA) и массовое назначение (Mass Assignment) на стороне сервера.
13. Использование протоколов OAuth 2.0 и OpenID Connect в мобильных приложениях: безопасные паттерны хранения Refresh Tokens.
14. Защита от атак типа «человек посередине» (MITM): сравнение эффективности различных реализаций SSL Pinning.
15. Безопасность Push-уведомлений: риски передачи конфиденциальной информации через каналы Apple Push Notification (APNs) и Firebase Cloud Messaging (FCM).
16. Критический анализ реализации JWT (JSON Web Tokens) в мобильных клиентах: риски кражи токенов и стратегии безопасного хранения.
17. Недостатки авторизации на стороне клиента (Client-Side Authorization): почему скрывание кнопок в интерфейсе не является защитой.
18. Многофакторная аутентификация (MFA) в мобильных устройствах: сравнение устойчивости SMS-кодов, Push-подтверждений и TOTP.
19. Глубокие ссылки (Deep Links) и универсальные ссылки (Universal Links): уязвимости перехвата управления (URL Scheme Hijacking) и способы защиты.
20. Безопасная реализация функции «Остаться в системе» (Remember Me): баланс между удобством пользователя и рисками компрометации устройства.
21. Реверс-инжиниринг мобильных приложений: использование инструментов Ghidra и Hopper для извлечения Hardcoded-ключей и алгоритмов.
22. Обфускация и минификация кода: эффективность ProGuard/R8 (Android) и Swift Shield (iOS) против статического анализа.

23. Методы обнаружения Root (Android) и Jailbreak (iOS): анализ популярных техник и способы их обхода (Bypass) злоумышленниками.
24. Dynamic Instrumentation с помощью Frida: примеры модификации логики приложения «на лету» (Runtime Manipulation) и методы защиты от внедрения кода.
25. Безопасность IPC (Inter-Process Communication): анализ уязвимостей экспорта Activity (Android) и Custom URL Schemes (iOS).
26. Защита от Overlay-атак (Tapjacking) в Android: механизмы предотвращения рисования вредоносных окон поверх легитимных приложений.
27. Анализ безопасности WebView: риски включения JavaScript-интерфейса (addJavascriptInterface) и загрузки ложического контента.
28. Уязвимости парсинга данных: атаки на десериализацию JSON/XML в мобильных приложениях и их последствия.
29. Безопасность сборки и CI/CD: предотвращение утечки секретов через конфигурационные файлы (build.gradle, Info.plist) в публичных репозиториях.
30. Анализ побочных каналов (Side-Channel Attacks): утечка информации через энергопотребление или время выполнения операций при криптографических вычислениях на смартфоне.

1.3 Тестовые задания

1. **В чем заключается основная опасность жесткого кодирования (hardcoding) API-ключей или секретов непосредственно в исходном коде мобильного приложения?** а) Это замедляет запуск приложения. б) Злоумышленник может извлечь их с помощью инструментов реверс-инжиниринга (декомпиляции) и использовать в своих целях. в) Это увеличивает размер установочного файла (APK/IPA). г) Это приводит к быстрой разрядке аккумулятора. *Правильный ответ: б*
2. **Какой механизм в Android и iOS используется для безопасного хранения криптографических ключей, защищая их даже при компрометации (Root/Jailbreak) операционной системы?** а) Shared Preferences / UserDefaults б) Обычный текстовый файл в папке Documents в) Android Keystore System и iOS Keychain / Secure Enclave г) База данных SQLite в открытом виде *Правильный ответ: в*
3. **Какова главная цель использования SSL/TLS Pinning (закрепления сертификата) в мобильном приложении?** а) Для ускорения загрузки данных с сервера. б) Для защиты от атак «человек посередине» (MITM) с использованием поддельных доверенных сертификатов. в) Для сжатия передаваемого трафика. г) Для автоматического обновления приложения. *Правильный ответ: б*
4. **Уязвимость «Insecure Data Storage» (Небезопасное хранение данных) чаще всего возникает, когда разработчик...** а) Использует слишком сложные пароли. б) Сохраняет конфиденциальные данные (токены, ПДн) в незащищенных хранилищах типа Shared Preferences (Android) или UserDefaults (iOS). в) Применяет HTTPS вместо HTTP. г) Использует биометрическую аутентификацию. *Правильный ответ: б*
5. **Как называется механизм изоляции, при котором каждое мобильное приложение работает в своей собственной среде и не имеет прямого доступа к данным других приложений?** а) Виртуальная машина б) Антивирусный сканер в) «Песочница» (Sandbox) г) Брандмауэр *Правильный ответ: в*
6. **Почему использование WebView с включенным JavaScript-интерфейсом (addJavascriptInterface в Android) считается опасным?** а) Это делает интерфейс некрасивым. б) Это позволяет любому веб-сайту, открытому в WebView, вызывать нативные функции устройства и потенциально выполнять вредоносный код. в) Это блокирует работу браузера по умолчанию. г) Это требует подключения к Wi-Fi. *Правильный ответ: б*

7. **В чем заключается атака «Tapjacking» (или Overlay Attack) в Android?** а) Перехват нажатий клавиш на физической клавиатуре. б) Отрисовка вредоносным приложением невидимого или прозрачного слоя поверх легитимного приложения для кражи нажатий пользователя. в) Удаленное управление устройством через Bluetooth. г) Подмена иконки приложения на рабочем столе. *Правильный ответ: б*
8. **Какова роль Refresh Token в архитектуре аутентификации на базе JWT и OAuth 2.0?** а) Для шифрования основного токена доступа. б) Для получения нового Access Token после истечения его срока действия без повторного ввода логина и пароля. в) Для хранения настроек пользователя. г) Для отображения аватара пользователя. *Правильный ответ: б*
9. **Что такое «Jailbreak» (iOS) или «Root» (Android) с точки зрения безопасности приложения?** а) Процесс обновления операционной системы. б) Получение привилегированного доступа к системе, позволяющий обходить механизмы безопасности и «песочницы». в) Установка официального приложения из магазина. г) Очистка кэша устройства. *Правильный ответ: б*
10. **Почему проверка прав доступа (например, «является ли пользователь администратором») только на стороне мобильного клиента является критической уязвимостью?** а) Потому что клиентское приложение можно модифицировать (реверс-инжиниринг) или обмануть, заставив его считать пользователя администратором. б) Потому что это замедляет работу интерфейса. в) Потому что сервер может не выдержать нагрузки. г) Потому что это запрещено правилами магазинов приложений. *Правильный ответ: а*
11. **Какой инструмент обычно используется для динамического анализа (внедрения кода «на лету») и изменения логики работы запущенного мобильного приложения?** а) Adobe Photoshop б) Microsoft Word в) Frida г) Калькулятор *Правильный ответ: в*
12. **В чем разница между авторизацией (Authorization) и аутентификацией (Authentication)?** а) Это одно и то же. б) Аутентификация — это проверка «кто вы?», а авторизация — проверка «что вам разрешено делать?». в) Аутентификация — это выход из системы, а авторизация — вход. г) Аутентификация используется только в вебе, а авторизация — только в мобильных приложениях. *Правильный ответ: б*
13. **Какое хранилище данных в iOS специально предназначено для хранения небольших фрагментов конфиденциальных данных, таких как пароли или токены, с автоматическим аппаратным шифрованием?** а) NSUserDefaults б) Keychain в) NSCache г) FileManager.default *Правильный ответ: б*
14. **Что защищает механизм обфускации кода (например, ProGuard/R8 для Android)?** а) От перегрева процессора устройства. б) От реверс-инжиниринга, делаая декомпилированный код нечитаемым и сложным для анализа. в) От потери интернет-соединения. г) От случайного удаления приложения пользователем. *Правильный ответ: б*
15. **Какая уязвимость (из OWASP API Top 10) возникает, когда злоумышленник меняет ID объекта в запросе (например, /api/user/100 на /api/user/101) и получает доступ к чужим данным из-за отсутствия проверки прав на сервере?** а) Broken Object Level Authorization (BOLA) / IDOR б) SQL Injection в) Denial of Service г) Cross-Site Scripting *Правильный ответ: а*
16. **Для безопасного хранения криптографических ключей в Android используется системное хранилище _____, которое защищает их от извлечения даже при наличии прав суперпользователя. (Правильный ответ: Keystore / Android Keystore System)**
17. **В iOS хранение небольших секретных данных, таких как токены или пароли, осуществляется в защищенном хранилище _____, интегрированном с аппаратным модулем Secure Enclave. (Правильный ответ: Keychain)**

18. Уязвимость, возникающая при сохранении конфиденциальных данных в Shared Preferences (Android) или UserDefaults (iOS) в открытом виде, классифицируется OWASP как небезопасное _____ данных. (Правильный ответ: хранение / *Insecure Data Storage*)
19. Механизм _____ (Pinning) используется в мобильных приложениях для защиты от атак «человек посередине» (MITM) путем привязки приложения к конкретному сертификату сервера. (Правильный ответ: закрепления / *Certificate Pinning*)
20. Процесс анализа и обратной разработки скомпилированного файла приложения (APK или IPA) для поиска секретов или логики называется _____-инжинирингом. (Правильный ответ: реверс / *Reverse*)
21. В Android каждый процесс приложения выполняется в изолированной среде, называемой «_____» (Sandbox), что предотвращает доступ одного приложения к данным другого. (Правильный ответ: песочница)
22. Атака _____ (Tarpjacking) заключается в том, что вредоносное приложение рисует прозрачный слой поверх легитимного интерфейса для перехвата нажатий пользователя. (Правильный ответ: перекрытие / *Overlay*)
23. Для защиты исходного кода от чтения и модификации после компиляции применяется процесс _____, например, с использованием инструментов ProGuard или R8. (Правильный ответ: обфускации)
24. В архитектуре аутентификации на базе JWT для обновления просроченного токена доступа без повторного ввода пароля используется токен _____. (Правильный ответ: обновления / *Refresh Token*)
25. Получение полного контроля над операционной системой iOS (обход ограничений Apple) называется _____ (Jailbreak). (Правильный ответ: джейлбрейк)
- 26. Соответствие между инструментом и его назначением**

Инструмент	Назначение
1. Frida	А. Инструмент для локального хранения ключей и проведения криптографических операций в Android.
2. Android Keystore	Б. Инструмент динамической инструментации («внедрения») кода для изменения логики работающего приложения.
3. Dex2jar / Ghidra	В. Межсетевой экран (прокси) для перехвата и анализа HTTP/HTTPS трафика между мобильным устройством и сервером.
4. Burp Suite	Г. Инструменты для декомпиляции Android-приложения в читаемый Java-код и его последующего анализа.

Правильные ответы: 1–Б, 2–А, 3–Г, 4–В.

27. Соответствие между уязвимостью OWASP Mobile Top 10 и описанием

Категория OWASP	Описание
1. M1 (Improper Platform Usage)	А. Хранение паролей или ключей в открытом виде в файловой системе или базе данных устройства.
2. M2 (Insecure Data Storage)	Б. Небезопасная реализация передачи данных (отказ от HTTPS или неверная проверка сертификатов).
3. M3 (Insecure Communication)	В. Неправильное использование функций ОС или API (например, незащищенный Content Provider в Android).
4. M9 (Reverse Engineering)	Г. Анализ логики приложения и извлечение секретов из бинарного файла из-за отсутствия обфускации.

Правильные ответы: 1–В, 2–А, 3–Б, 4–Г.

28. Соответствие между методом аутентификации и его характеристикой

Метод	Характеристика
1. EAP-TLS	А. Самый небезопасный метод, использующий передачу хеша пароля внутри защищенного TLS-туннеля (уязвим к перебору).

Метод	Характеристика
2. PEAP-MSCHAPv2	Б. Требует наличия цифровых сертификатов как на сервере, так и на каждом мобильном клиенте (наиболее высокий уровень безопасности).
3. EAP-FAST	В. Использует защищенный туннель и «временные» учетные данные (PAC) для аутентификации без сертификатов на клиенте.
4. OAuth 2.0	Г. Стандарт делегирования доступа, использующий токены (Access/Refresh) вместо передачи пароля приложению.

Правильные ответы: 1–Б, 2–А, 3–В, 4–Г.

29. Соответствие между API платформы и его функцией

API / Компонент	Функция
1. LAContext (iOS)	А. Отображение веб-страниц внутри приложения (требует настройки безопасного WebView).
2. BiometricPrompt (Android)	Б. Класс для работы с Keychain (хранилищем секретов) в iOS.
3. WebView	В. Системный диалог для биометрической аутентификации в iOS.
4. Keychain (iOS API)	Г. Системный диалог для биометрической аутентификации в Android.

Правильные ответы: 1–В, 2–Г, 3–А, 4–Б.

30. Соответствие между типом сборки и целью использования

Тип сборки	Цель
1. Debug (отладочная)	А. Максимальная производительность, отсутствие логов, обфускация кода, защита от отладки.
2. Release (релизная)	Б. Удобство разработки, наличие подробных логов, возможность подключения отладчика, отсутствие обфускации.
3. Root/Jailbreak сборка	В. Специальная версия приложения, которая проверяет наличие взлома ОС и ограничивает функционал (например, в банковских приложениях).
4. Secure (защищенная)	Г. Модифицированная версия приложения, предназначенная для запуска на устройствах с правами суперпользователя для тестирования защиты.

Правильные ответы: 1–Б, 2–А, 3–Г, 4–В.

1.4 Практические и ситуационные задания

Задание 1. Сравните модели безопасности iOS и Android. Опишите различия в механизмах изоляции приложений («песочница»), проверки приложений (App Review vs Google Play Protect) и управления привилегиями.

Задание 2. Концепция «Secure by Design» в мобильной разработке. Объясните, почему исправление уязвимости на этапе проектирования обходится дешевле, чем после публикации приложения в сторах.

Задание 3. Примените методологию STRIDE для моделирования угроз мобильного приложения мобильного банкинга. Приведите по одному примеру угрозы для каждой буквы аббревиатуры.

Задание 4. Опишите роль сервера бэкенда в обеспечении безопасности. Почему проверка прав доступа (Access Control) и лимитов (Rate Limiting) никогда не должна полностью полагаться на логику мобильного клиента?

Задание 5. Анализ поверхности атаки (Attack Surface) мобильного приложения. Перечислите не менее 5 потенциальных точек входа для злоумышленника (от UI до фоновых сервисов).

Задание 6. Уязвимость «Insecure Data Storage». Сравните риски хранения данных в Shared Preferences (Android) / UserDefaults (iOS) и в зашифрованных базах данных (например, SQLCipher).

Задание 7. Android Keystore System и iOS Keychain. Как эти механизмы обеспечивают защиту ключей даже в случае полной компрометации операционной системы (Root/Jailbreak)?

Задание 8. Локальная аутентификация (Biometrics). Объясните разницу между «простой» биометрией (только для разблокировки UI) и использованием CryptoObject (Android) / LAContext с политиками (iOS) для защиты криптографических операций.

Задание 9. Очистка данных (Data Wiping). Почему стандартное удаление файла в мобильной ОС не гарантирует уничтожение информации и какие методы безопасного затирания (overwriting) существуют?

Задание 10. Хранение токенов сессии. Сравните безопасность хранения JWT в Shared Preferences, HttpOnly Cookie (в WebView) и системном Keystore. Какой метод наиболее устойчив к краже?

Задание 11. SSL/TLS Pinning. Опишите механизм защиты от атак «человек посередине» (MITM). Какие сложности возникают при обновлении сертификатов на сервере при использовании Pinning?

Задание 12. Анализ перехвата трафика. Опишите, как с помощью Burp Suite и Frida можно обойти некорректную проверку SSL-сертификата (например, если в коде разрешены все хосты).

Задание 13. Безопасность API. Разбор уязвимости BOLA (Broken Object Level Authorization / IDOR) на примере запроса GET /api/v1/user/{id}. Как должен выглядеть безопасный код на стороне сервера?

Задание 14. Использование WebView. Какие критические риски несет включение addJavascriptInterface (Android) или hasOnlySecureContent (iOS) и как правильно настроить безопасный WebView?

Задание 15. Push-уведомления (FCM/APNs). Являются ли они безопасным каналом для передачи конфиденциальных данных (например, кодов подтверждения)? Обоснуйте ответ.

Задание 16. OAuth 2.0 и OpenID Connect. В чем разница между Access Token и Refresh Token? Где безопаснее хранить Refresh Token в мобильном приложении?

Задание 17. Недостатки реализации собственного механизма аутентификации. Почему разработчикам рекомендуется использовать стандартные протоколы (OAuth2/OIDC) вместо «самописных» систем входа?

Задание 18. Авторизация на стороне клиента (Client-Side Authorization). В чем заключается опасность доверять мобильному UI (например, скрывать кнопку «Админка», если пользователь не админ)?

Задание 19. Многофакторная аутентификация (MFA). Сравните устойчивость к атакам SMS-кодов, Push-уведомлений и TOTP (Google Authenticator).

Задание 20. Глубокие ссылки (Deep Links) и Universal Links. Опишите уязвимость перехвата ссылок (Link Hijacking) и способы её предотвращения через проверку сертификатов и доменов.

Задание 21. Реверс-инжиниринг (Reverse Engineering). Зачем злоумышленники декомпилируют APK/IPA файлы? Какие инструменты (Ghidra, Hopper, jadx) они используют?

Задание 22. Обфускация кода. Как инструменты ProGuard/R8 (Android) и Swift Shield/Obfuscator (iOS) защищают приложение от анализа и взлома логики?

Задание 23. Hardcoded-секреты. Почему жестко закодированные API-ключи в исходном коде считаются критической уязвимостью, даже если код скомпилирован в бинарный файл?

Задание 24. Root/Jailbreak Detection. Почему обнаружение взлома устройства критически важно для банковских приложений? Опишите два метода определения Jailbreak и два метода их обхода (Jailbreak Bypass).

Задание 25. Dynamic Instrumentation (Frida). Покажите на примере, как с помощью Frida можно заставить функцию `isPremiumUser()` всегда возвращать `true`, не пересобирая приложение.

Задание 26. Атака Overlay (Tapjacking) в Android. Как вредоносное приложение может нарисовать невидимое окно поверх легитимного для кражи нажатий? Как этому противодействовать программно?

Задание 27. Безопасность IPC (Inter-Process Communication). Риски неправильной настройки Intent (Android) или Custom URL Schemes (iOS), позволяющие другим приложениям вызывать функции вашего.

Задание 28. Защита от отладки (Anti-Debugging). Как приложение может определить, что его пытаются запустить в отладчике, и зачем это нужно злоумышленнику?

Задание 29. Атаки на цепочку поставок (Supply Chain). Риски использования скомпрометированных сторонних библиотек (SDK) и менеджеров зависимостей (CocoaPods, Maven).

Задание 30. Безопасность сборки и CI/CD. Как предотвратить утечку секретов (паролей от сторов, ключей шифрования) через конфигурационные файлы (`build.gradle`, `Info.plist`) в публичных репозиториях Git?

2. КРИТЕРИИ ОЦЕНИВАНИЯ ПРИ ПРОВЕДЕНИИ ТЕКУЩЕГО КОНТРОЛЯ

Вид контроля	Наименование работы	Наименование оценочных средств	Шкала оценивания
Текущий контроль	Вопросы для обсуждения на занятиях; Устные опросы по ранее изученному материалу; Письменные работы: рефераты, тестовые задания; Ситуационные и практические задания; Рефераты и доклады по темам (вопросам), вынесенным на самостоятельную работу.	Оценка выступлений на практическом (семинарском) занятии, проверка заданий и аудиторных работ, устный опрос, оценивание докладов, рефератов	отлично хорошо удовлетворительно неудовлетворительно

Критерии оценивания устных ответов обучающихся

Шкала оценивания	Характеристика оценивания
отлично	оценивается ответ, который показывает прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа.
хорошо	оценивается ответ, обнаруживающий прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа. Однако допускается одна - две неточности в ответе.
удовлетворительно	оценивается ответ, свидетельствующий в основном о знании процессов изучаемой предметной области, отличающийся недостаточной глубиной и полнотой раскрытия темы; знанием основных вопросов теории; слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры; недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа.
неудовлетворительно	оценивается ответ, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы; незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов; неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Допускаются серьезные ошибки в содержании ответа.

Критерии оценивания работы обучающихся на практических и семинарских занятиях

Шкала оценивания	Показатели	Критерии
Отлично	1. Полнота выполнения практического и тестового задания (полнота ответа); 2. Своевременность выполнения задания; 3. Последовательность и рациональность выполнения практического задания (логичность и четкость ответа);	Задание решено самостоятельно. При этом составлен правильный алгоритм решения задания, в логических рассуждениях, в выборе формул и решении нет ошибок, получен верный ответ, задание решено рациональным способом. Дан правильный и исчерпывающий ответ на поставленные теоретические и тестовые вопросы, в которых обучающийся показал всестороннее системное знание программного материала, усвоение основной и дополнительной литературы, четкое владение понятийным аппаратом.
Хорошо	4. Правильность ответов на вопросы; 5. Самостоятельность решения (владение дополнительным материалом); 6. Знание нормативно-законодательной базы и терминологии курса	Задание решено с помощью преподавателя. При этом составлен правильный алгоритм решения задания, в логическом рассуждении и решении нет существенных ошибок; правильно сделан выбор формул для решения; есть объяснение решения, но задание решено нерациональным способом или допущено не более двух несущественных ошибок, получен верный ответ. На поставленные теоретические и тестовые вопросы, при которых обучающийся показал достаточный уровень знаний основного программного материала: освоение информации лекционного курса и учебных пособий, овладение понятийным аппаратом, методикой исследований при попытке анализа различных ситуаций.
Удовлетворительно		Задание решено с подсказками преподавателя. Задание решено в общем виде. Обучающийся показал средний уровень знаний основного программного материала, но не мог убедительно аргументировать свой ответ, ошибся в использовании понятийного аппарата, показал недостаточные знания литературных источников.
Неудовлетворительно		Задание не решено. Обучающийся продемонстрировал значительные пробелы в знаниях основного программного материала, не аргументировал свой ответ, показал неудовлетворительные знания понятийного аппарата и специальной литературы.

Критерии оценивания рефератов

Форам контроля	Критерии оценивания	Шкала оценивания
Реферат	Выполнены все методические требования к написанию реферата: определена тема, актуальность; содержание соответствует теме и плану реферата; сделан краткий анализ различных точек зрения по рассматриваемой проблеме;	зачтено

	грамотно использованы первоисточники при анализе содержания; выводы сформулированы лаконично с логическими доказательствами; соблюдены требования по внешнему оформлению. Тема реферата раскрыта в полном объеме.	
	Нарушены требования по написанию и оформлению реферата. Несоответствие темы реферата. В тексте не отражены существенные моменты. Тема реферата не раскрыта до конца. Сделаны необоснованные выводы.	не зачтено

Критерии оценивания тестов

Средство контроля	Критерии оценивания – процент положительных ответов	Шкала оценивания
Тестирование	90-100	отлично
	70-89	хорошо
	40-69	удовлетворительно
	< 39	неудовлетворительно

3. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Средства оценивания в ходе промежуточной аттестации:

– вопросы для зачета;

–практические задания для зачета.

Коды компетенции	Результаты освоения ОПОП	Перечень планируемых результатов обучения по дисциплине
ПК-6	Способен организовывать технологический процесс защиты информации ограниченного доступа в соответствии с нормативными правовыми актами и нормативными	ПК-6.1. Знает правовые основы организации защиты государственной тайны и конфиденциальной информации, задачи органов защиты государственной тайны и служб защиты информации на предприятиях, нормативные правовые акты и стандарты по лицензированию деятельности в области обеспечения защиты государственной тайны, технической защиты конфиденциальной информации, нормативные методические документы ФСБ России, ФСТЭК России в области защиты информации; ПК-6.2. Владеет навыками организации и обеспечения режима секретности

3.1. Вопросы к зачету

6. Концепция «Secure by Design» в мобильной разработке. В чем отличие подхода безопасности «с нуля» от добавления защиты на этапе тестирования?

7. Модель угроз OWASP Mobile Top 10. Назовите 5 наиболее критичных категорий уязвимостей и приведите пример для каждой.
8. Различия в моделях безопасности iOS и Android. Сравните механизмы «песочницы» (sandbox), управления правами (Permissions) и проверки приложений (App Review vs Google Play Protect).
9. Моделирование угроз (Threat Modeling) для мобильного приложения. Опишите этапы процесса (по методологии STRIDE) на примере приложения мобильного банкинга.
10. Безопасность архитектуры клиент-сервер. Почему логика проверки прав доступа (Access Control) никогда не должна полностью полагаться на мобильное устройство?
11. Небезопасное хранение данных (Insecure Data Storage). Сравните риски хранения данных в Shared Preferences (Android) / UserDefaults (iOS) и в SQLite/Realm базах данных без шифрования.
12. Хранение криптографических ключей. Сравните использование KeyStore (Android) и Keychain (iOS) с хранением ключей в коде или конфигурационных файлах.
13. Android KeyStore System и iOS Keychain. Как эти механизмы обеспечивают защиту ключей даже при компрометации операционной системы (Root/Jailbreak)?
14. Скрытие данных в файловой системе. Почему удаление файлов стандартными средствами не гарантирует их уничтожение и как работает «Data Wiping» (затирание)?
15. Локальная аутентификация (Biometrics). Как правильно интегрировать биометрию (FaceID/TouchID/Fingerprint) с использованием CryptoObject для защиты критических операций?
16. Анализ трафика мобильного приложения. Почему использование HTTP вместо HTTPS недопустимо? Опишите риски перехвата данных в публичных Wi-Fi сетях.
17. Pinning SSL/TLS сертификатов (Certificate/Public Key Pinning). Как этот механизм защищает от атак с использованием поддельных доверенных сертификатов и каковы сложности его поддержки?
18. Безопасность API (Server-Side). Почему проверка лимитов (Rate Limiting) и фильтрация входных данных на сервере критически важны, даже если клиентское приложение «безопасно»?
19. Атака Man-in-the-Middle (MITM) на мобильное приложение. Как злоумышленник может обойти проверку сертификата в приложении, если она реализована некорректно (например, Accept All Certificates)?
20. Использование WebView. Какие риски несет включение JavaScript-интерфейса (addJavascriptInterface) и как правильно настроить безопасный WebView?
21. Управление сессиями. Почему использование случайного идентификатора сессии (Session ID) в заголовках предпочтительнее передачи логина/пароля при каждом запросе?
22. Недостатки реализации собственного механизма аутентификации. Почему разработчикам не рекомендуется писать свою систему входа, а следует использовать OAuth 2.0 / OpenID Connect?
23. JWT (JSON Web Tokens) в мобильных приложениях. Где безопасно хранить токен доступа (Access Token) и токен обновления (Refresh Token)? Почему хранить Refresh Token в UserDefaults опасно?
24. Авторизация на стороне клиента (Client-Side Authorization). В чем опасность доверять UI мобильного приложения (например, скрывать кнопку «Админка», если пользователь не админ)?
25. Многофакторная аутентификация (MFA). Сравните эффективность SMS-кодов, Push-уведомлений и TOTP (Google Authenticator) для защиты мобильного приложения.
26. Реверс-инжиниринг (Reverse Engineering). Зачем злоумышленники декомпилируют APK/IPA файлы? Какие секреты (ключи API, алгоритмы) они ищут?
27. Обфускация кода. Как инструменты ProGuard/R8 (Android) и Swift Obfuscator (iOS) защищают приложение от анализа и взлома логики?

28. Небезопасное хранение секретов в коде. Почему жестко закодированные (hardcoded) ключи API в исходном коде считаются критической уязвимостью, даже если код скомпилирован?
29. Root/Jailbreak Detection. Почему обнаружение взлома устройства важно для банковских приложений и какие методы обхода (bypass) используют хакеры?
30. Dynamic Instrumentation (Frida/Ghidra). Как злоумышленники могут изменять логику работы запущенного приложения «на лету» (например, заставить функцию `isPremiumUser()` всегда возвращать `true`)?
31. Защита от Overlay-атак (Tapjacking). Как в Android вредоносное приложение может нарисовать невидимое окно поверх легитимного для кражи нажатий? Как этому противодействовать?
32. Безопасность IPC (Inter-Process Communication). Риски неправильной настройки Intent (Android) или Custom URL Schemes (iOS), позволяющие другим приложениям вызывать функции вашего.
33. Защита от отладки (Anti-Debugging). Как приложение может определить, что его пытаются запустить в отладчике, и зачем это нужно?
34. Push-уведомления. Являются ли они безопасным каналом для передачи конфиденциальных данных (например, кодов подтверждения)?
26. Безопасность сборки и CI/CD. Риски утечки секретов через `build.gradle` или `Info.plist` в системе автоматической сборки и способы их защиты (Secrets Management).

3.2. Практические задания к зачету с оценкой

Задание 1. Сравните модели безопасности iOS и Android. Опишите различия в механизмах изоляции приложений («песочница»), проверки приложений (App Review vs Google Play Protect) и управления привилегиями.

Задание 2. Концепция «Secure by Design» в мобильной разработке. Объясните, почему исправление уязвимости на этапе проектирования обходится дешевле, чем после публикации приложения в сторах.

Задание 3. Примените методологию STRIDE для моделирования угроз мобильного приложения мобильного банкинга. Приведите по одному примеру угрозы для каждой буквы аббревиатуры.

Задание 4. Опишите роль сервера бэкенда в обеспечении безопасности. Почему проверка прав доступа (Access Control) и лимитов (Rate Limiting) никогда не должна полностью полагаться на логику мобильного клиента?

Задание 5. Анализ поверхности атаки (Attack Surface) мобильного приложения. Перечислите не менее 5 потенциальных точек входа для злоумышленника (от UI до фоновых сервисов).

Задание 6. Уязвимость «Insecure Data Storage». Сравните риски хранения данных в Shared Preferences (Android) / UserDefaults (iOS) и в зашифрованных базах данных (например, SQLCipher).

Задание 7. Android Keystore System и iOS Keychain. Как эти механизмы обеспечивают защиту ключей даже в случае полной компрометации операционной системы (Root/Jailbreak)?

Задание 8. Локальная аутентификация (Biometrics). Объясните разницу между «простой» биометрией (только для разблокировки UI) и использованием `CryptoObject` (Android) / `LAContext` с политиками (iOS) для защиты криптографических операций.

Задание 9. Очистка данных (Data Wiping). Почему стандартное удаление файла в мобильной ОС не гарантирует уничтожение информации и какие методы безопасного затирания (overwriting) существуют?

Задание 10. Хранение токенов сессии. Сравните безопасность хранения JWT в Shared Preferences, HttpOnly Cookie (в WebView) и системном Keystore. Какой метод наиболее устойчив к краже?

Задание 11. SSL/TLS Pinning. Опишите механизм защиты от атак «человек посередине» (MITM). Какие сложности возникают при обновлении сертификатов на сервере при использовании Pinning?

Задание 12. Анализ перехвата трафика. Опишите, как с помощью Burp Suite и Frida можно обойти некорректную проверку SSL-сертификата (например, если в коде разрешены все хосты).

Задание 13. Безопасность API. Разбор уязвимости BOLA (Broken Object Level Authorization / IDOR) на примере запроса GET /api/v1/user/{id}. Как должен выглядеть безопасный код на стороне сервера?

Задание 14. Использование WebView. Какие критические риски несет включение addJavascriptInterface (Android) или hasOnlySecureContent (iOS) и как правильно настроить безопасный WebView?

Задание 15. Push-уведомления (FCM/APNs). Являются ли они безопасным каналом для передачи конфиденциальных данных (например, кодов подтверждения)? Обоснуйте ответ.

Задание 16. OAuth 2.0 и OpenID Connect. В чем разница между Access Token и Refresh Token? Где безопаснее хранить Refresh Token в мобильном приложении?

Задание 17. Недостатки реализации собственного механизма аутентификации. Почему разработчикам рекомендуется использовать стандартные протоколы (OAuth2/OIDC) вместо «самописных» систем входа?

Задание 18. Авторизация на стороне клиента (Client-Side Authorization). В чем заключается опасность доверять мобильному UI (например, скрывать кнопку «Админка», если пользователь не админ)?

Задание 19. Многофакторная аутентификация (MFA). Сравните устойчивость к атакам SMS-кодов, Push-уведомлений и TOTP (Google Authenticator).

Задание 20. Глубокие ссылки (Deep Links) и Universal Links. Опишите уязвимость перехвата ссылок (Link Hijacking) и способы её предотвращения через проверку сертификатов и доменов.

Задание 21. Реверс-инжиниринг (Reverse Engineering). Зачем злоумышленники декомпилируют APK/IPA файлы? Какие инструменты (Ghidra, Hopper, jadx) они используют?

Задание 22. Обфускация кода. Как инструменты ProGuard/R8 (Android) и Swift Shield/Obfuscator (iOS) защищают приложение от анализа и взлома логики?

Задание 23. Hardcoded-секреты. Почему жестко закодированные API-ключи в исходном коде считаются критической уязвимостью, даже если код скомпилирован в бинарный файл?

Задание 24. Root/Jailbreak Detection. Почему обнаружение взлома устройства критически важно для банковских приложений? Опишите два метода определения Jailbreak и два метода их обхода (Jailbreak Bypass).

Задание 25. Dynamic Instrumentation (Frida). Покажите на примере, как с помощью Frida можно заставить функцию isPremiumUser() всегда возвращать true, не пересобирая приложение.

Задание 26. Атака Overlay (Tapjacking) в Android. Как вредоносное приложение может нарисовать невидимое окно поверх легитимного для кражи нажатий? Как этому противодействовать программно?

Задание 27. Безопасность IPC (Inter-Process Communication). Риски неправильной настройки Intent (Android) или Custom URL Schemes (iOS), позволяющие другим приложениям вызывать функции вашего.

Задание 28. Защита от отладки (Anti-Debugging). Как приложение может определить, что его пытаются запустить в отладчике, и зачем это нужно злоумышленнику?

Задание 29. Атаки на цепочку поставок (Supply Chain). Риски использования скомпрометированных сторонних библиотек (SDK) и менеджеров зависимостей (CocoaPods, Maven).

Задание 30. Безопасность сборки и CI/CD. Как предотвратить утечку секретов (паролей от сторов, ключей шифрования) через конфигурационные файлы (build.gradle, Info.plist) в публичных репозиториях Git?

4. КРИТЕРИИ ОЦЕНИВАНИЯ ПРИ ПРОВЕДЕНИИ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Шкала оценивания уровня сформированности компетенции (по пятибалльной системе) (зачет с оценкой)

Шкала оценивания	Уровень освоение компетенции	Критерии оценивания
Зачет	Базовый уровень освоения компетенции	Дан правильный и исчерпывающий ответ на вопрос с отражением дополнительного материала. Обучающийся демонстрирует глубокое знание теоретического материала, умение грамотно, логично и аргументировано излагать свои мысли.
Незачет	Неудовлетворительный уровень	Отсутствует ответ или в ответе есть грубые ошибки в изложении материала, свидетельствующие о незнании и непонимании соответствующего программного материала, отсутствует понимание сути вопроса.

Текущий контроль и промежуточная аттестация осуществляются в соответствии с «Положением о текущем контроле успеваемости и промежуточной аттестации обучающихся в Автономной некоммерческой организации «Образовательная организация высшего образования» «Университет экономики и управления».

Вид промежуточной аттестации – зачет.

Форма проведения промежуточной аттестации – письменный зачет.