Техническое задание
на разработку механизма защиты от пересборки приложения

Заказчик: ТОО TLT GAMES
Пользователи: физические лица, скачивающие приложение в Google Play
Срок: 5 – 6 недель
Дата: 18.11.2026
Страна разработки: Казахстан

1. Проблема.

В текущей стадии проекта приложение использует только базовые встроенные механизмы безопасности операционной системы, чего недостаточно для защиты от современных методов атак. Злоумышленник может подменять возвращаемые значения, обходить авторизацию и проверку подписей, что нелегально и может нанести вред пользователям.

2. Назначение и цели создания.

Для безопасной работы приложения требуется создать механизм защиты приложения от модификации, пересборки, декомпиляции, иного несанкционированного использования.

3. Требования к плагину.

3.1. *Стек и совместимость*:
- версии Unity: 2021.3 LTS и выше;
- бэкенд сборки: IL2CPP (поддержка архитектур ARMv7 и ARM64);
- ОС**:** Android 10/ API 29 - Android 16/ API 36;
- языки разработки**:** C# (Unity API), C++ (Android NDK для критических проверок), Java/Kotlin (взаимодействие с Android SDK);

3.2. *Интеграция*.
Подключение плагина должно требовать минимальных правок в коде основного приложения. Плагин не должен конфликтовать с базовой минимизацией R8/ProGuard.

3.3. *Учёт оригинальных методов Android по валидации бэкенда*, в т.ч. Play Integrity API, ссылка на документацию Google:
https://developer.android.com/google/play/integrity?authuser=6&hl=ru.

3.4. Объём защиты.

3.4.1. Обязательно - Проверка целостности и подписи (базовый функционал, ставка 30%):

- проверка подписи APK/AAB (сверка SHA-256 хеша сертификата подписи в нативном коде),
- проверка целостности файлов (проверка хеш-сумм критически важных файлов);

3.4.2. Обязательно - Защита от декомпиляции и обфускация (базовый функционал, ставка 25%):
- шифрование критически важных строковых констант с расшифровкой в рантайме,
- обфускация управляющего потока (Control Flow Flattening) для ключевых классов и методов (список классов передаётся Заказчиком),
- очистка отладочной информации и стриппинг символов в нативных .so-библиотеках.

3.4.3. Обязательно - Обнаружение запуска в скомпрометированной среде, RASP (базовый функционал, ставка 25%):
- обнаружение наличия Root-прав (проверка бинарников su, системных разделов, обнаружение Magisk/KernelSU и аналогов),
- детект подключения отладчиков (проверка TracerPid,, блокировка через ptrace),
- выявление фреймворков динамического инструментария (Frida, Xposed/LSPosed — проверка открытых портов, карт памяти /proc/self/maps, характерных библиотек).

3.4.4. Обязательно - Детектирование запуска на эмуляторах (базовый функционал, ставка 10%):
- проверка системных свойств Build, специфичных файлов и драйверов QEMU/Genymotion/BlueStacks/Nox.

3.4.5. Обязательно - Реакция на угрозу (базовый функционал, ставка 10%):
- при определении любой из обязательных проверок плагин должен передавать код события в приложение (через callback) либо принудительно завершать процесс приложения (настраивается конфигурацией).

3.4.6. Желательно - Защита от перехвата трафика (расширенный функционал, ставка 10%):
- встроенный SSL Pinning или контроль целостности хранилища сертификатов.

3.4.7. Желательно - Обнаружение запуска приложения в виртуальных контейнерах/клонерах (расширенный функционал, ставка 10%).

4. Этапы, сроки условия оплаты.

Ориентировочный цикл разработки составляет 5-6 недель в зависимости от реализуемого фукционала плагина.

Этап 1. *Проектирование и архитектура* (1 неделя): согласование модели угроз, API взаимодействия с хост-приложением, архитектуры нативного ядра. Результат: архитектурный документ.

Этап 2. *Разработка базового ядра* (2 недели): реализация обязательных мер защиты (пункты 3.4.1 – 3.4.5 ТЗ). Результат: альфа-версия плагина (.aar).

Этап 3. *Расширенный функционал* (1 неделя): реализация дополнительных мер защиты (пункты 3.4.6 и 3.4.7 ТЗ). Результат: релиз-кандидат (RC).

Этап 4. П*ентест и стабилизация* (1–2 недели): нагрузочное тестирование, проведение пентеста, исправление найденных недочётов, подготовка документации. Результат: финальный релиз.

Размер оплаты за готовый релизный вариант плагина – от 1 500 000 ₸. Размер итоговой стоимости работ зависит от объёма выполненной работы (см. пункты 3.4.1 – 3.4.7). Окончательная стоимость работ согласовывается Заказчиком и Исполнителем и указывается в акте сдачи-приёмки.

5. Порядок проверки и критерии приемки.

5.1. Влияние на производительность и размер (KPI).

5.1.1. Увеличение размера APK/AAB: не более 10% от исходного размера релизной сборки.
5.1.2. Время холодного старта (Cold Start): замедление запуска приложения не более чем на 1 с (замеряется на тестовом устройстве Заказчика).
5.1.3. Потребление ОЗУ и CPU: отсутствие утечек памяти; пиковый прирост потребления RAM при проверках — не более 20 МБ.

5.1.4. Стабильность (Crash-free): отсутствие ложных срабатываний на чистых (не взломанных) устройствах популярных производителей (Samsung, Xiaomi, Google Pixel, Honor, Huawei, Realmi).

5.2. Перечень проверяемых векторов атак, порядок приёмки.

Исполнитель предоставляет демо-приложение с интегрированным плагином.

Команда Заказчика проводит нагрузочное тестирование (пункты 5.1.1 -5.1.4), пентест по критериям, указанным ниже (пункты 5.2.1 – 5.2.5 ТЗ).

По итогам тестирования формируется отчёт.

Плагин считается принятым, если успешно блокируются следующие базовые сценарии:

5.2.1. Декомпиляция APK через JADX/Apktool: ключевые строки не читаются в открытом виде, логика защищённых методов не восстанавливается линейно.

5.2.2. Пересборка APK (изменение любого байта в classes.dex или ресурсах) и переподпись тестовым ключом: приложение не запускается / фиксирует нарушение целостности.

5.2.3. Запуск приложения на устройстве с Magisk (включая режим Zygisk/DenyList): факт наличия Root определяется.

5.2.4. Запуск на эмуляторах Bluestacks, Nox: факт эмуляции определяется.

5.2.5. Подключение к процессу через frida-server или adb jdwp / gdb: отладка/хукинг пресекаются.

Если в ходе приёмки выявлен обход защиты стандартными инструментами из списка выше, Исполнитель обязуется устранить уязвимость в рамках гарантийной доработки за 5 (пять) рабочих дней.

6. Порядок передачи прав и документации.

Исключительные права на разработанный плагин (включая исходный код на C#//C++, скрипты сборки и конфигурационные файлы) в полном объёме переходят к Заказчику с момента подписания акта сдачи-приёмки.

Ограничения на Open Source. Исполнитель гарантирует, что в коде не используются библиотеки с вирусными лицензиями (GPL, AGPL), требующими раскрытия исходного кода продукта Заказчика. Допускаются только пермиссивные лицензии (MIT, Apache 2.0, BSD) по предварительному согласованию.

Передаваемые материалы:
- Исходный код плагина в Git-репозитории с историей коммитов.
- Скомпилированные артефакты (.jar / .aar) для быстрой интеграции.
- Документация: руководство по подключению в CI/CD, описание настроек конфигурации, инструкция по обновлению ключей/хешей сертификатов.

Гарантийная поддержка. Исполнитель обязуется бесплатно устранять выявленные баги и конфликты с новыми минорными версиями приложения в течение 3 (трёх) месяцев после подписания акта.

7. Требования к Исполнителю.

7.1. уверенное владение C#/C++, знание Unity, Android NDK;

7.2. опыт работы с Android Keystore System, работы с обфускаторами (ProGuard, R8, DexGuard и пр.);
7.3. знание инструментов анализа, защиты от декомпиляции (Jadx, IDA Pro и пр.), обратной разработки, шифрования;
7.4. умение реализовать безопасное сетевое взаимодействие (TLS 1.3, SSL Pinning);
7.5. опыт проведения SAST, DAST.