Презентація орієнтована на Андроїд девелоперів в яких вже є досвід роботи з платформою андроїд і які тільки починають інтегрувати С++. Або С++ девів які починають працювати з Андроїд платформою.

В цілому в презентації буде йти мова про типові помилки з якими можуть стикатись інженери і короткі поради як уникати таких проблем.

Я не буду розповідати про JNI підходи і як з ним працювати а фокусуюсь на тих проблемах з якими ми дуже часто зтикаємося на нашому проекті

**Слайд 1**

Для чого взагалі використовувати

Для чого ми використовуємо НДК на проекті?

Структура проекту. Проблеми андроїда для роботи з НДК

**Слайд 2**

Використання глобальних і локальних посилань. Приклад з проетку.

Як виглядає креш через оверфлов таблиці локальних посилань. Приклад з коду.

Поради по вирішенню таких проблем. Код.

Те саме про глобальні посилання. Приклади

**Слайд 3**

Робота з нативними потоками. Приклад з проекту.

Invocation API?

Приклад крешів. Код. Шляхи вирішення

**Слайд Crash handling**

Зменшення розміру Зрізання різних символів приводить до таких наслідків як повністю обфускований стек крешу повністю обфускований стек крешу.

Приклад типового крешу який можна знайти на гугл дев консолі. Це корисно для різного захисту від пен тестерів але також і своїх інженерів.

Відміна від обфускованих джава файлів. Маппінги

ndk-stack tool

Інструмент ndk-stack дозволяє фільтрувати обфусковані стеки, як вони з'являються на виході adb logcat. Він також замінює будь-яку адресу в спільній бібліотеці за допомогою відповідних значень <source-file>: <number-line> з вашого вихідного коду, що полегшує їх виявлення.

Користування

Деталі креш стеку. Важливі деталі

JNI Table reference overflow

Що таке JNI reference? Local, global and weak