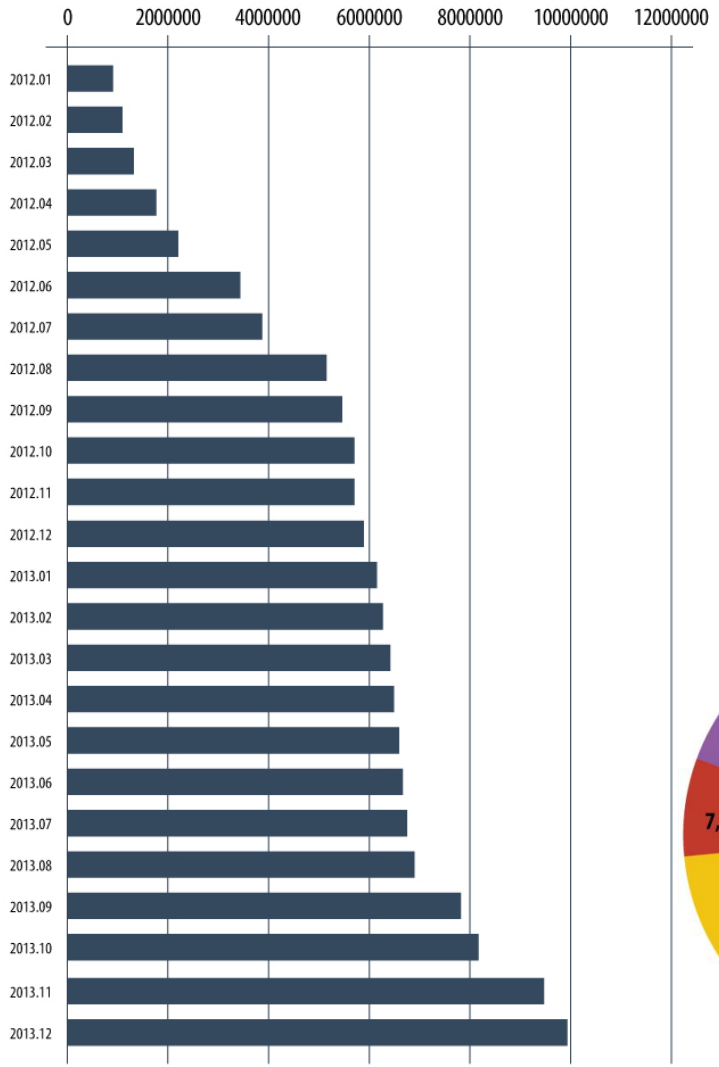
НЕОБИТ

НОВЫЕ
БЕЗОПАСНЫЕ
ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ

# ПРИМЕНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ АНАЛИЗА БЕЗОПАСНОСТИ ПРИЛОЖЕНИЙ ANDROID

ПАВЛЕНКО Е. Ю., РУКОВОДИТЕЛЬ ПРОЕКТОВ ООО «НЕОБИТ»

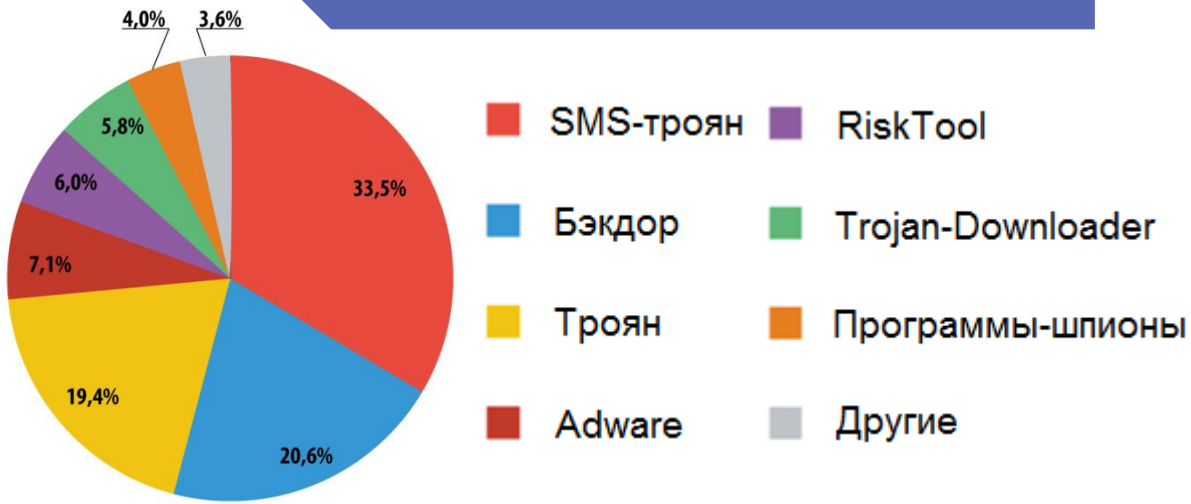
# ВПО для Android




**Инфицирование легальных веб-ресурсов**

**Распространение через альтернативные магазины приложений**

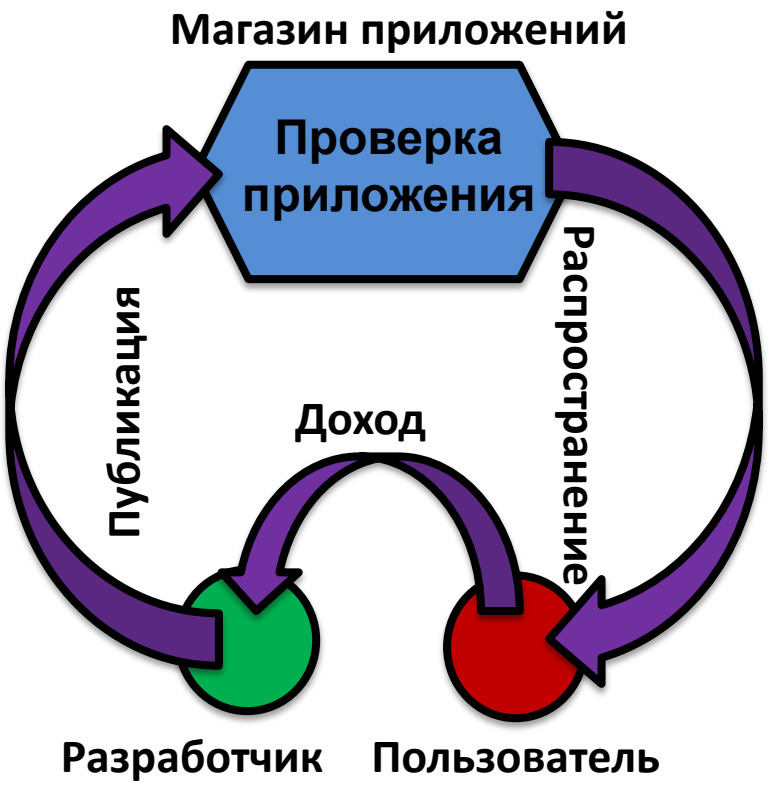
**Распространение через ботнеты**






НЕОБИТ

# Магазин приложений Google Play




**Подготовка к публикации**

- Зарегистрировать учетную запись разработчика на Google Play
  - Прочесть и принять «Соглашение о распространении программного обеспечения»
  - Оплатить $25 в качестве регистрационного сбора
  - Указать информацию о разработчике
- Зарегистрировать аккаунт продавца в Google Wallet
- С использованием консоли разработчика Google опубликовать приложение

- Правила в отношении контента
- Распространение приложений
- Правила в отношении рекламы
- Спам



НЕОБИТ

# Система защиты от вредоносного ПО Google Bouncer

Google Bouncer осуществляет анализ приложений в Google Play с целью их проверки на наличие вирусов и вредоносного кода.

Задачами Bouncer являются:

- ✓ Обнаружение в приложении известных вирусов и других вредоносных программных средств
- ✓ Анализ безопасности поведения приложения

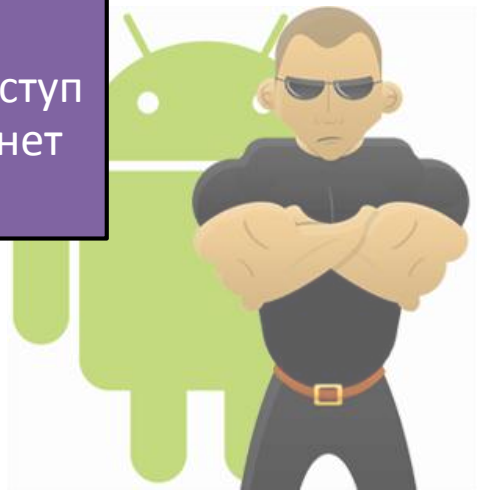
Виртуальная среда QEMU, эмулирующая Android

Выполнение в течение 5 минут

Разрешает доступ к сети Интернет

Динамический анализ приложения

Выполняется на инфраструктуре компании Google



# Обнаружение и обход Google Bouncer

## Содержимое виртуального Android-смартфона

- download/cat.jpg
- download/lady-gaga-300.jpg
- DCIM/Camera/IMG_20120302_142816.jpg
- android/data/passwords.txt

Почтовый аккаунт:
Miles.Karlson@gmail.com
В адресной книге один контакт:
Michelle.k.levin@gmail.com
Android ID: 9774d56d682e549c

## Обнаружение виртуальной среды

- телефон никогда не заряжается;
- не работает акселерометр;
- не работает фотокамера;
- /proc/cpuinfo: goldfish;
- getprop attributes: ro.kernel.qemu;
- /sys/qemu_trace и т.д.

QEMU

## Обнаружение инфраструктуры Google

```
$ whois 74.125.19.84 | grep OrgName
OrgName:        Google Inc.
$ whois 173.194.99.18 | grep OrgName
OrgName:        Google Inc.
```

WHOIS

## Временная задержка

Используется специальный внутренний таймер для отложенного исполнения своей полезной нагрузки

# Эффективность мобильных антивирусов

Современные мобильные антивирусы имеют следующий функционал:

- Поиск известных вирусов по сигнатурам
- Блокировка опасных сайтов при переходе по ссылке
- Проверка ссылок, полученных в SMS-сообщениях
- Дополнительные функции

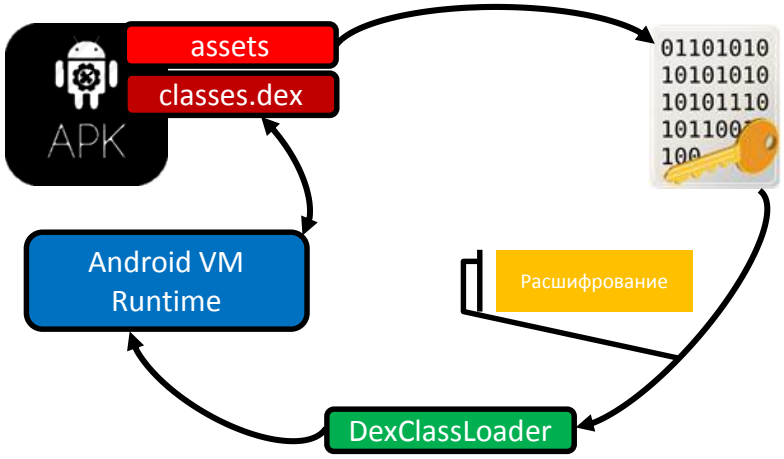

| Показатель выявления после изменений | | | | | | |
|---|---|---|---|---|---|---|
| Переименование пакетов | Изменение строк в байт-коде | Изменение манифеста | Изменение ресурсов | Переименование классов в байт-коде | Изменение, классов, методов и строк в байт-коде | Всё перечисленное с перемешиванием кода |
| 21 / 56 | 13 / 56 | 23 / 56 | 24 / 56 | 21 / 56 | 8 / 56 | 0 / 56 |

- Своя тестовая программа ни одним антивирусом обнаружена не была
- Вирусная программа Android.Trojan.MMarket с показателем: 40/56

**При наличии исходного кода:** → **Динамическая подгрузка байт-кода** → **Использование нативного кода**

# Методы обхода мобильных антивирусов

## Динамическая загрузка зашифрованных модулей

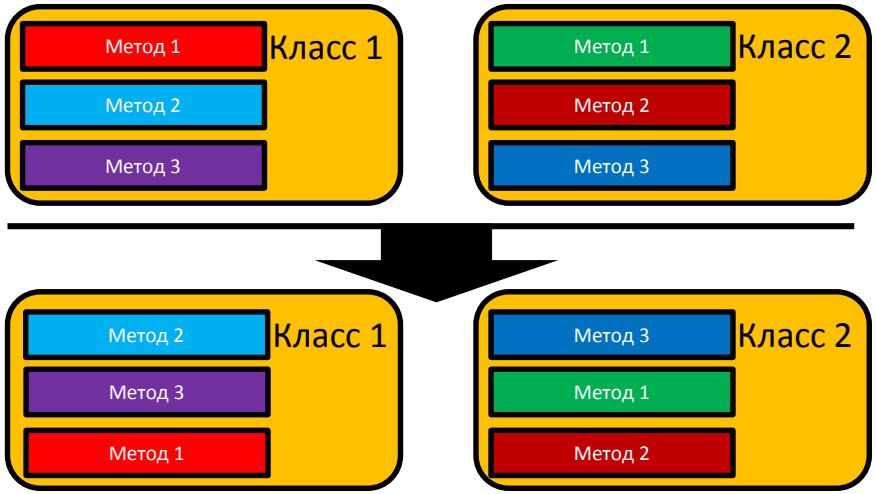



## Обфускация кода

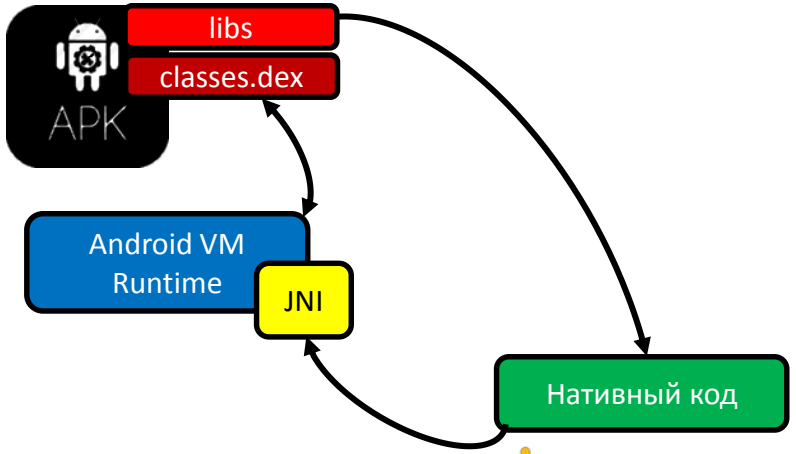
```
.method public addPart(Ljava/lang/String;Lorg/apache/http/entity/mime/content/ContentBody;)V
    .locals 1
    .param p1, "name"    # Ljava/lang/String;
    .param p2, "contentBody"    # Lorg/apache/http/entity/mime/content/ContentBody;
```

```
method public SJWsdgSqdf(Ljava/lang/String;LwbJCsmLydN/neJdUgzjll/SpFRrIKlLe/ISkuGKaMRS/YRiYEXGOat/GIm
    .locals 1
    .param p1, "IqKKIGMolhAHYQuRgIvR"    # Ljava/lang/String;
    .param p2, "mOvcnqaHveRtGUFry8wI"    # LwbJCsmLydN/neJdUgzjll/SpFRrIKlLe/ISkuGKaMRS/YRiYEXGOat/GIml
```

## Перемешивание кода



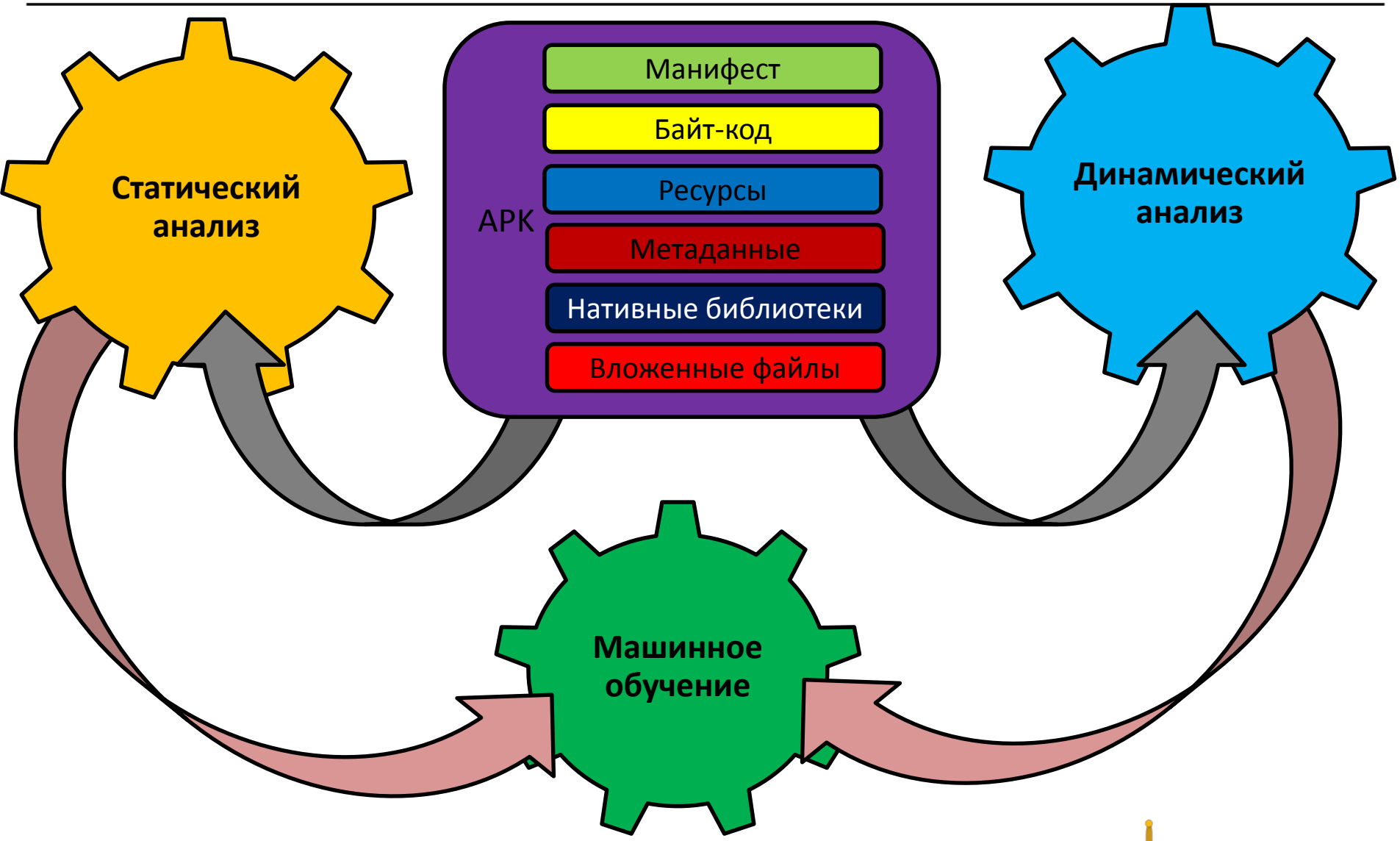

## Использование нативных библиотек



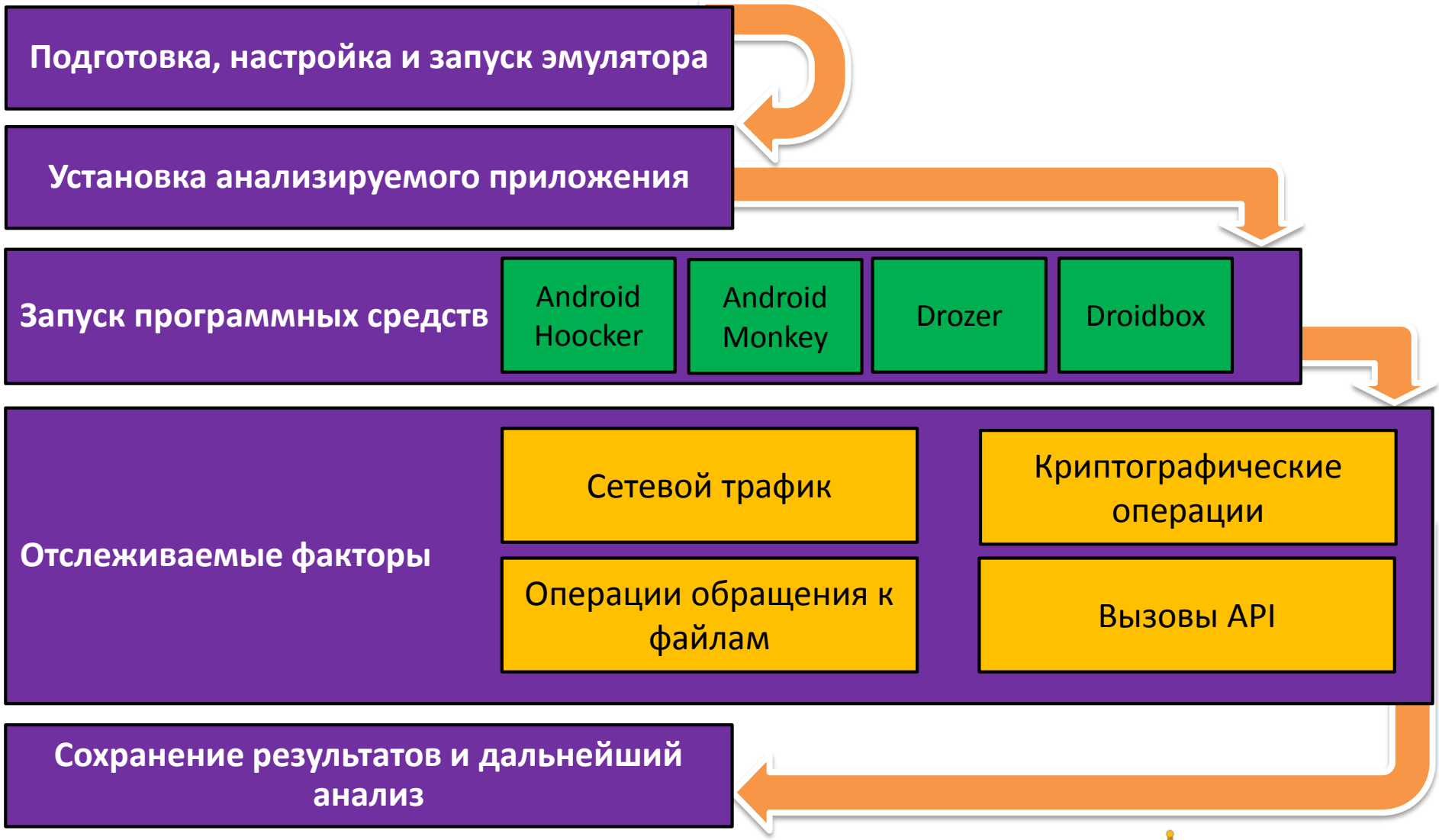



НЕОБИТ

# Схема анализа приложений



# Динамический анализ






НЕОБИТ

# Статический анализ


Анализ манифеста
Анализ байт-кода
→ [1,0,1, … , 0]

## Опасные разрешения

READ_CALENDAR
WRITE_CALENDAR
CAMERA
READ_CONTACTS
WRITE_CONTACTS
GET_ACCOUNTS
ACCESS_FINE_LOCATION
ACCESS_COARSE_LOCATION
RECORD_AUDIO
READ_PHONE_STATE
MODIFY_PHONE_STATE
CALL_PHONE
READ_CALL_LOG
WRITE_CALL_LOG
ADD_VOICEMAIL
USE_SIP
PROCESS_OUTGOING_CALLS
BODY_SENSORS
SEND_SMS
RECEIVE_SMS
READ_SMS
RECEIVE_WAP_PUSH
RECEIVE_MMS
READ_EXTERNAL_STORAGE
WRITE_EXTERNAL_STORAGE

## Запрашиваемая информация об устройстве

getLac()
getCid()
getCallState()
getCellLocation()
getDataActivity()
getDataState()
getDeviceId()
getDeviceSoftwareVersion()
getNeighboringCellInfo()
getNetworkCountryIso()
getNetworkOperator()
getNetworkOperatorName()
getNetworkType()
getPhoneType()
getSimCountryIso()
getSimOperator()
getSimOperatorName()
getSimSerialNumber()
getSimState()
getSubscriberId()
getVoiceMailAlphaTag()
getVoiceMailNumber()
getPackageInfo()

## Выполнение кода

Ljava/lang/System->loadLibrary
Ljava/lang/Runtime->exec
Ljava/lang/ClassLoader->DexClassLoader

## Доступ к директориям

/system/xbin/su
/efs/
/system/init.d
/system/etc/hosts
/system/lib/
/system/build.prop

## Доступ к персональной информации

Landroid/location/LocationManager->getProviders

## BroadcastReceiver

android.app.action.ACTION_PASSWORD_CHANGED
android.app.action.ACTION_PASSWORD_EXPIRING
android.app.action.ACTION_PASSWORD_FAILED
android.app.action.ACTION_PASSWORD_SUCCEEDED
android.app.action.DEVICE_ADMIN_DISABLED
android.app.action.DEVICE_ADMIN_DISABLE_REQUESTED
android.app.action.DEVICE_ADMIN_ENABLED
android.app.action.LOCK_TASK_ENTERING
android.app.action.LOCK_TASK_EXITING
android.intent.action.ACTION_POWER_CONNECTED
android.intent.action.ACTION_POWER_DISCONNECTED
android.intent.action.ACTION_SHUTDOWN
android.intent.action.APPLICATION_RESTRICTIONS_CHANGED
android.intent.action.BATTERY_CHANGED
android.intent.action.BOOT_COMPLETED
android.intent.action.CONFIGURATION_CHANGED
android.intent.action.CONTENT_CHANGED
android.intent.action.DATA_SMS_RECEIVED
android.intent.action.DATE_CHANGED
android.intent.action.EXTERNAL_APPLICATIONS_AVAILABLE
android.intent.action.EXTERNAL_APPLICATIONS_UNAVAILABLE
android.intent.action.FETCH_VOICEMAIL
android.intent.action.MANAGE_PACKAGE_STORAGE
android.intent.action.NEW_OUTGOING_CALL
android.intent.action.PACKAGE_ADDED
android.intent.action.PACKAGE_CHANGED
android.intent.action.PACKAGE_DATA_CLEARED
android.intent.action.PACKAGE_INSTALL
android.intent.action.PACKAGE_NEEDS_VERIFICATION
android.intent.action.PACKAGE_REMOVED
android.intent.action.PHONE_STATE
android.intent.action.REBOOT
android.net.wifi.NETWORK_IDS_CHANGED
android.provider.Telephony.SMS_RECEIVED



НЕОБИТ

# Формальное описание

$A = \{a_1, a_2, \dots, a_m\}$ - множество объектов-приложений - множество объектов кластеризации. Множество характеристик объектов $P = \{p_1, p_2, \dots, p_n\}$. Предполагаем, что в результате статического и динамического анализа каждому объекту $a_i \in A$ ставится в соответствие некоторый вектор $x_i = (x_1^i, x_2^i, \dots, x_n^i)$, где $x_j^i$- значение признака $p_j \in P$, $x_j^i \in \{0,1\}, i = \overline{1,m}, \ j = \overline{1,n}$. На множестве объектов имеется разбиение на конечное число непересекающихся классов $\Omega_k, \ k = 1, \dots, l, \ \cup_{k=1}^{l} \Omega_k = A$. Информация о вхождении некоторого объекта $a$ в какой-либо класс представляется в виде вектора $\{I_1(a), I_2(a), \dots, I_k(a)\}$, где $I_j(a)$ несет информацию о принадлежности объекта $a$ к классу $\Omega_j$:

$$I_j(a) = \begin{cases} 1, & \text{если } a \in \Omega_k \\ 0, & \text{если } a \notin \Omega_k \\ \Delta, & \text{если неопределенность} \end{cases}$$

Решение о принадлежности объекта $a$ к классу $\Omega_j$ определяется на основе меры близости между объектом и ядром кластера.

Для решения данной проблемы можно использовать методы нечеткой кластеризации, позволяющие каждому объекту принадлежать с различной степенью нескольким или всем кластерам одновременно. Тогда множество нечетких кластеров $C = \{\text{опасный}, \text{безопасный}, \text{потенциально опасный}\}$ можно задать матрицей разбиения: $P = [\mu_{ki}], \mu_{ki} \in [0,1], k = \overline{1,m}, \ \ i = \overline{1,c}$, где $\mu_{ki}$ – степень принадлежности объекта $k$ к кластеру $i$, $c$ – количество кластеров, $m$ – количество объектов.

При этом: $\sum_{i=1}^{c} \mu_{ki} = 1, k = \overline{1,m}$ ; $0 < \sum_{k=1}^{M} \mu_{ki} < m, i = \overline{1,c}$.

# Макет системы анализа приложений




НЕОБИТ

# Применение различных мер близости на этапе кластеризации


**Евклидово расстояние.**

$$\rho(x, v_k) = \sqrt{\sum_{k=1}^{n} (x_k - v_k)^2}$$

**Квадрат евклидова расстояния.**

$$\rho(x, v_k) = \sum_{k=1}^{n} (x_k - v_k)^2$$

**Расстояние Хэмминга.**

$$\rho(x, v_k) = \sum_{k=1}^{n} |x_k - v_k|$$

**Расстояние Чебышева.**

$$\rho(x, v_k) = \max_{k=1,\dots,n} |x_k - v_k|$$

**Индекс отделимости.**

Характеризует насколько отделены кластеры друг от друг, измеряя минимальное расстояние между их центрами.

$$S(c) = \frac{\sum_{i=1}^{c} \sum_{k=1}^{N} \mu_{ij}^2 \|x_k - v_i\|^2}{N \cdot \min_{i,j} \|v_j - v_i\|^2}$$

**Индекс Хие-Бени.**

Определяет правильность выбора количества кластеров для алгоритма с-средних.

$$XB(c) = \frac{\sum_{i=1}^{c} \sum_{k=1}^{N} \mu_{ij}^m \|x_k - v_i\|^2}{N \cdot \min_{i,k} \|x_k - v_i\|^2}$$

**Индекс разбиения.**

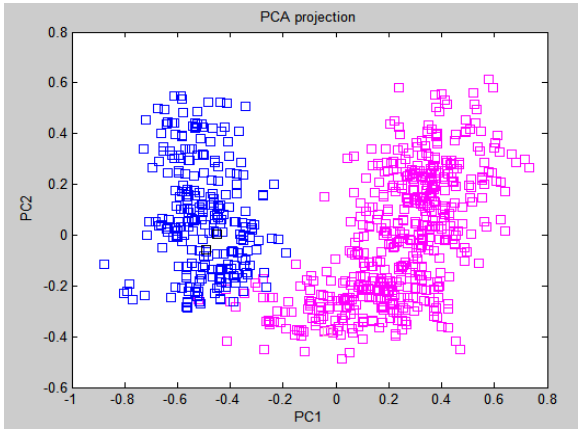
Используется для сравнения различных разбиений при одинаковом количестве кластеров.

$$SC(c) = \sum_{i=1}^{c} \frac{\sum_{k=1}^{N} \mu_{ik}^m \|x_k - v_i\|^2}{\sum_{k=1}^{N} \mu_{ik} \sum_{j=1}^{c} \|v_j - v_i\|^2}$$

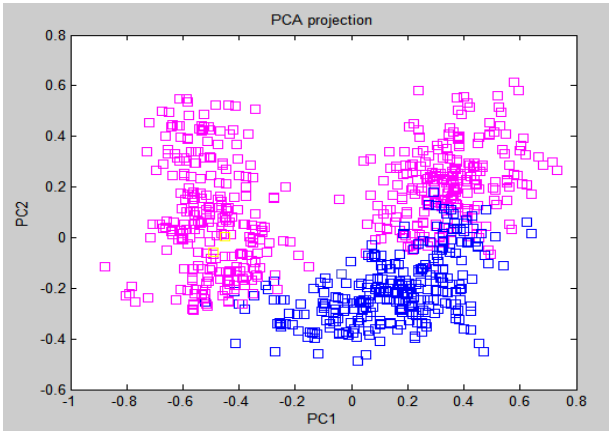


НЕОБИТ

# Нечеткая кластеризация



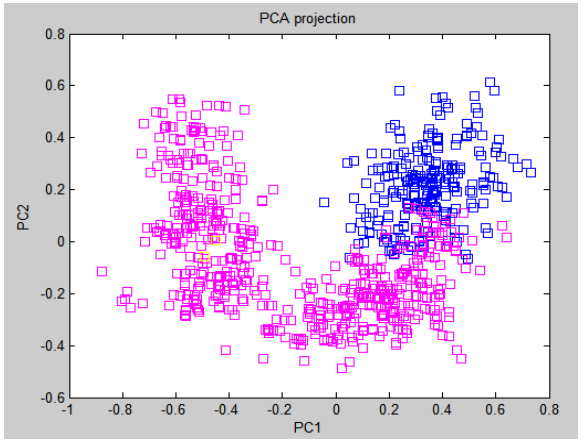

Кластер 1 - Опасный



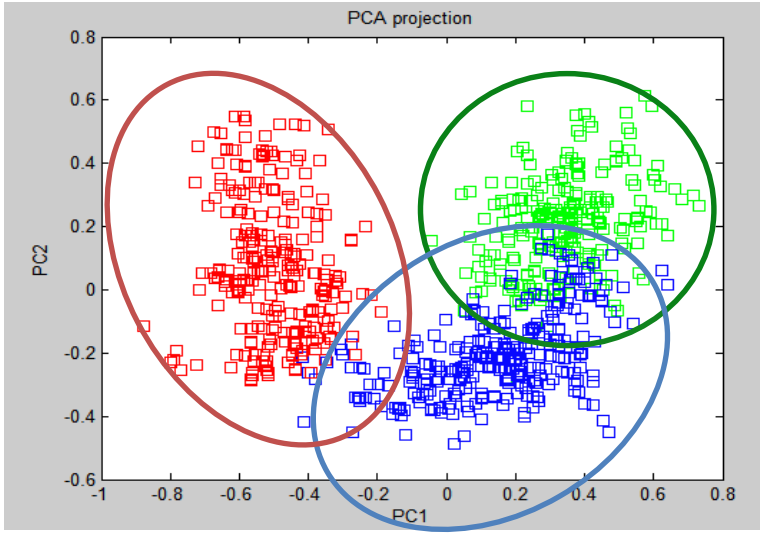

Кластер 2 – Потенциально опасный

**Значение функции принадлежности $\mu$:**

- $\mu = 0$
- $0 < \mu \leq 0.2$
- $0.2 < \mu \leq 0.4$
- $0.4 < \mu \leq 0.6$
- $0.6 < \mu \leq 0.8$
- $0.8 < \mu < 1$
- $\mu = 1$




Кластер 3 - Безопасный






НЕОБИТ

# Примеры приложений

| Название приложения | Описание | Полученные меры близости | Результат |
|---|---|---|---|
|  Pocket Tanks | Игра, аркадные танковые сражения | $\rho\,(x, v_1) = 3.2880$<br>$\rho(x, v_2) = 3.2506$<br>$\rho(x, v_3) = 2.6689$ | **Безопасный** |
|  TimeZone Fixer | Обновление файлов временной зоны | $\rho\,(x, v_1) = 3.0452$<br>$\rho(x, v_2) = 2.8584$<br>$\rho(x, v_3) = 2.9651$ | **Потенциально опасный** |
|  Complexity | Троянская программа | $\rho\,(x, v_1) = 2.0452$<br>$\rho(x, v_2) = 2.8579$<br>$\rho(x, v_3) = 2.9660$ | **Опасный** |
|  BatteryLife | Троянская программа | $\rho(x, v_1) = 1.7486$<br>$\rho(x, v_2) = 2.7536$<br>$\rho\,(x, v_3) = 1.9536$ | **Опасный** |
|  UniversalAndroot | Троянская программа | $\rho\,(x, v_1) = 1.4518$<br>$\rho(x, v_2) = 4.4528$<br>$\rho(x, v_3) = 5.2218$ | **Опасный** |
|  FBReader | Программа для чтения электронных книг | $\rho\,(x, v_1) = 3.0476$<br>$\rho(x, v_2) = 2.8733$<br>$\rho(x, v_3) = 2.3411$ | **Безопасный** |

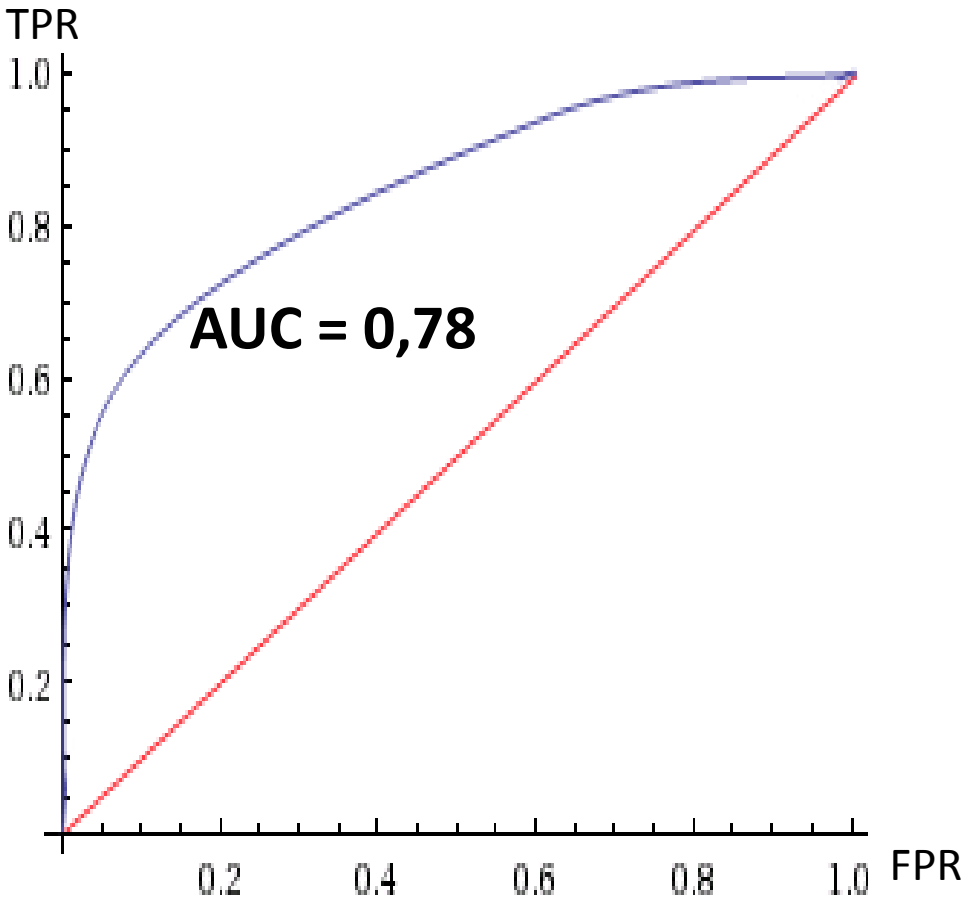
# Примеры анализа приложений

| Название приложения | Описание | Полученные меры близости | Результат |
|---|---|---|---|
|  Stagefright Detector | Анализатор уязвимостей | $\rho(x, v_1) = 1.9502$<br>$\rho(x, v_2) = 1.8495$<br>$\rho(x, v_3) = 2.3102$ | **Потенциально опасный** |
|  DroidDream | Троянская программа | $\rho(x, v_1) = 3.1366$<br>$\rho(x, v_2) = 3.5103$<br>$\rho(x, v_3) = 3.1665$ | **Опасный** |
|  Cameringo | Эффекты для фотографий | $\rho(x, v_1) = 3.4106$<br>$\rho(x, v_2) = 3.6959$<br>$\rho(x, v_3) = 3.3989$ | **Безопасный** |
|  WalkMate | Шагометр для мобильных устройств | $\rho(x, v_1) = 2.9476$<br>$\rho(x, v_2) = 2.3173$<br>$\rho(x, v_3) = 2.1167$ | **Безопасный** |
|  Steamy Screen | Приложение-бот | $\rho(x, v_1) = 2.7369$<br>$\rho(x, v_2) = 2.7824$<br>$\rho(x, v_3) = 3.1429$ | **Опасный** |
|  Chinese Eye | Троянская программа | $\rho(x, v_1) = 3.0499$<br>$\rho(x, v_2) = 3.0560$<br>$\rho(x, v_3) = 3.9426$ | **Опасный** |

# Оценка качества проводимого анализа

| Результаты экспериментов | |
|---|---|
| Всего приложений проанализировано | 1000 |
| Вредоносное программное обеспечение | 403 |
| Безопасные приложения | 597 |
| Истинно-положительное значение (true-positive, TP) | 333 |
| Ложно-положительное значение false-positive, FP) | 144 |
| Истинно-отрицательное значение (true-negative, TN) | 453 |
| Ложно-отрицательное значение (false-negative, FN) | 70 |

$$Sensitivity = \frac{TP}{TP+FN} = 0.83 \qquad Specificity = \frac{TN}{TN+FP} = 0.76$$






НЕОБИТ

# Выводы на основе результатов исследования

1. Google Play в основном регламентирует юридические аспекты информационной безопасности
2. Google Bouncer не является сложной интеллектуальной системой, которая способна выявить ВПО
3. Антивирусы не могут защитить от нового или сильно измененного старого ВПО
4. Статический и динамический анализ приложений для ОС Google Android дает достаточное количество сведений для анализа безопасности приложения
5. Кластеризация способна обеспечить достаточно высокую вероятность определения безопасности приложения для ОС Google Android

# НЕОБИТ

Санкт-Петербург, ул. Гжатская, 21 литер Г
Тел./факс: (812) 535-28-06
Сайт: neo-bit.ru
Почта: info@neo-bit.ru