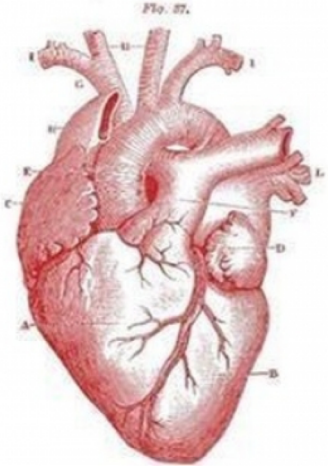
android: he protec, but he also attac

Гузенко Александр

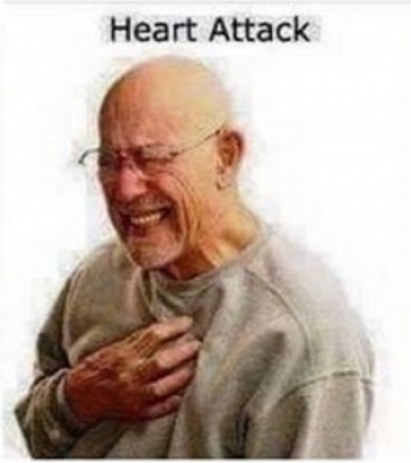
Tinkoff.ru

почему в названии доклада ошибка?

he protec



but he also attac



he protec

but he also attac

linus_irl

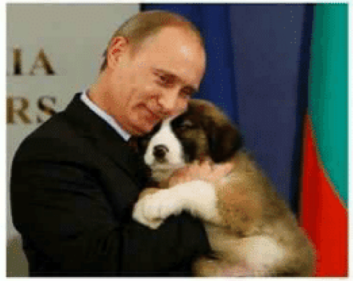
He protec

He also attac

But most importantly

He zucc

He protec



He attac

But most importantly

He hac

а нужно ли мне думать о безопасности?

обязательно, если вы разрабатываете:

банковское приложение

Tinkoff
Сбербанк
…

электронные кошельки

Qiwi
Я.Деньги
…

социальные сети

ВК
Facebook
Pure
…

приложения для хранения данных

Google Drive
Dropbox
…

любые другие приложения содержащие **«приватную" информацию о пользователе**


И ТАК
СОЙДЕТ!

проблемы реальны

check point опросил

443

ИТ- и ИБ-специалистов

результаты исследования показали, что защита большинства компаний отстаёт на 10 лет

аналитики check point обращают внимание на опасность таких секторов, как

банковское дело

торговля

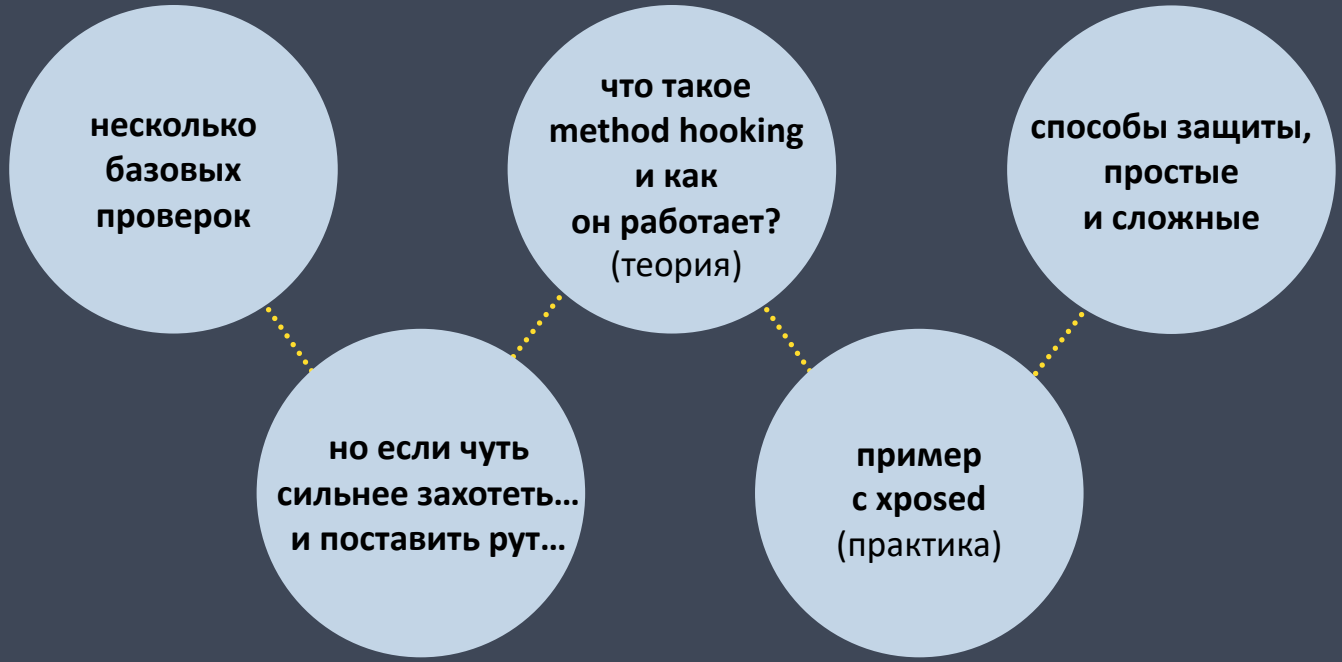
производство

согласно отчету 2018 Check Point более 300 мобильных приложений, распространяющихся через официальные магазины

содержат вредоносный код

о чем не будем говорить

план действий



дисклеймер

сначала обычные пользователи (простые смертные)

а не под дебагом ли ты, милок?

```
fun isDebugBuild(): Boolean = BuildConfig.DEBUG || 0 !=
(getApplicationInfo().flags and FLAG_DEBUGGABLE)
```

ИЛИ

```
fun isBeingDebugged(): Boolean =
Debug.isDebuggerConnected()
```

проверка на эмулятор

```
fun isRunningOnEmulator(): Boolean {
  val buildDetails = (Build.FINGERPRINT + Build.DEVICE + Build.MODEL + Build.BRAND + Build.PRODUCT +
Build.MANUFACTURER + Build.HARDWARE).toLowerCase()

  return buildDetails.run {
           contains("generic")
          ||contains("unknown")
          ||contains("emulator")
          ||contains("sdk")
          ||contains("genymotion")
          ||contains("x86")
          ||contains("goldfish")
          ||contains("test-keys")
          ||contains("google_sdk")
  }
}
```

установщик

```
val installer: String? = packageManager.getInstallerPackageName(context.packageName)
```

null == developer

PLAY_STORE_APP_ID == **"com.android.vending"**;

AMAZON_APP_ID == **"com.amazon.venezia"**;

SAMSUNG_APP_STORE_ID == **"com.sec.android.app.samsungapps"**;

подпись приложения

подпись — это гарант того, от кого вы устанавливаете приложение

подпись приложения

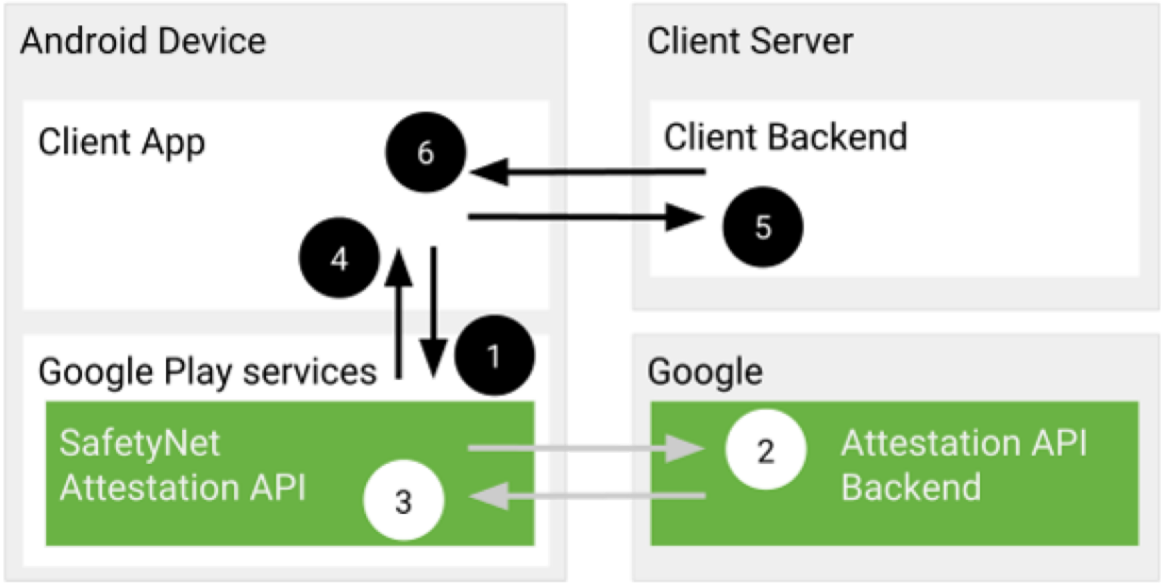
```
private fun isSignedWithDebugKeystore(): Boolean {
  var debuggable = false
  val pInfo = packageManager.getPackageInfo(packageName, GET_SIGNING_CERTIFICATES)
  val signatures = pInfo.signingInfo.signingCertificateHistory
  val cf = CertificateFactory.getInstance("X.509")
  for (i in signatures.indices) {
    val stream = ByteArrayInputStream(signatures[i].toByteArray())
    val cert = cf.generateCertificate(stream) as X509Certificate
    debuggable = cert.subjectX500Principal == X500Principal("CN=Android Debug,O=Android,C=US")
    if (debuggable) return false
  }
  return debuggable
}
```

подпись приложения

```
fun checkAppSignature(context: Context): Boolean {
  val packageInfo = packageManager.getPackageInfo(packageName, GET_SIGNING_CERTIFICATES)
  val signatures = packageInfo.signingInfo.apkContentsSigners
  for (signature in signatures) {
     val md = MessageDigest.getInstance("SHA")
     md.update(signature.toByteArray())
     val currentSignature = Base64.encodeToString(md.digest(), Base64.DEFAULT)
     if (SIGNATURE == currentSignature) {
        return true
     }
  }
  return false
}
```

SafetyNet

protect against security threats with SafetyNet



SafetyNet

```
SafetyNet.getClient(this).attest(nonce, API_KEY)
{
  "ctsProfileMatch": true,
  "basicIntegrity": true,
  "nonce": "R2Rra24fVm5xa2Mg",
  "timestampMs": 9860437986543,
  "apkPackageName": "com.package.name.of.requesting.app",
  "apkCertificateDigestSha256": ["base64 encoded, SHA-256 hash of the
                    certificate used to sign requesting app"],
  "apkDigestSha256": "base64 encoded, SHA-256 hash of the app's APK"
}
```

бугагашенька

а теперь
Root пользователи

а не под рутом ли ты, милок?

```
private fun canExecuteSuCommand(): Boolean {
  return try {
    Runtime.getRuntime().exec("su")
    true
  } catch (localIOException: IOException) {
    false
  }
}
```

или

RootBeer 

всё на поверхности

**APK —
просто архив**

**APKTool —
легко достанет
вам читаемый код**

method hooking

перехват (англ. hooking) — технология, позволяющая изменить стандартное поведение тех или иных компонентов информационной системы

method hooking

original

```
int getGold(int x, int y) {
        return x + y;
}
```

hooked

```
int hookGetGold(int x, int y) {
        y += 500;
        x += 500;
        return x + y;
}
```

method hooking

```
cout << getGold(5, 5)
```

ожидание	реальность
getGold: 10	hookGetGold: 1010

method hooking

original

.....

```
cout << getGold(5, 5)
```

.....

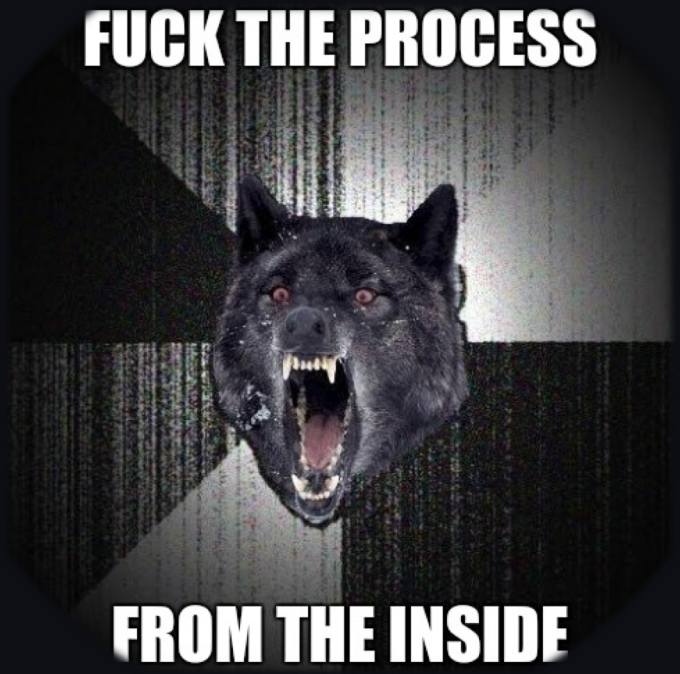
hooked

jmp (место в памяти нашего hooked метода)

```
int getGold(int x, int y) {
    return x + y;
}
```

jmp (место в памяти после getGold метода)

method hooking



покажите мне код

```
/*
 * This method will open /proc/<pid>/maps and search for the specified
 * library base address.
 */
uintptr_t findLibrary( const char *library, pid_t pid = -1 ) {
    char filename[0xFF] = {0},
         buffer[1024] = {0};
    FILE *fp = NULL;
    uintptr_t address = 0;

    sprintf( filename, "/proc/%d/maps", pid == -1 ? _pid : pid );

    fp = fopen( filename, "rt" );
    if( fp == NULL ){
        perror("fopen");
        goto done;
    }

    while( fgets( buffer, sizeof(buffer), fp ) ) {
        if( strstr( buffer, library ) ){
            address = (uintptr_t)strtoul( buffer, NULL, 16 );
            goto done;
        }
    }

```

покажите мне код

```
/*
 * This method will open /proc/<pid>/maps and search for the specified
 * library base address.
 */
uintptr_t findLibrary( const char *library, pid_t pid = -1 ) {
    char filename[0xFF] = {0},
         buffer[1024] = {0};
    FILE *fp = NULL;
    uintptr_t address = 0;
```

```
        goto done;
    }

    while( fgets( buffer, sizeof(buffer), fp ) ) {
        if( strstr( buffer, library ) ){
            address = (uintptr_t)strtoul( buffer, NULL, 16 );
            goto done;
        }
    }
```

покажите мне код

```
/*
 * This method will open /proc/<pid>/maps and search for the specified
 * library base address.
 */
uintptr_t findLibrary( const char *library, pid_t pid = -1 ) {
    char filename[0xFF] = {0},
         buffer[1024] = {0};

    sprintf( filename, "/proc/%d/maps", pid == -1 ? _pid : pid );

    fp = fopen( filename, "rt" );
    if( fp == NULL ){
        perror("fopen");
        goto done;
    }

            address = (uintptr_t)strtoul( buffer, NULL, 16 );
            goto done;
        }
    }
```

покажите мне код

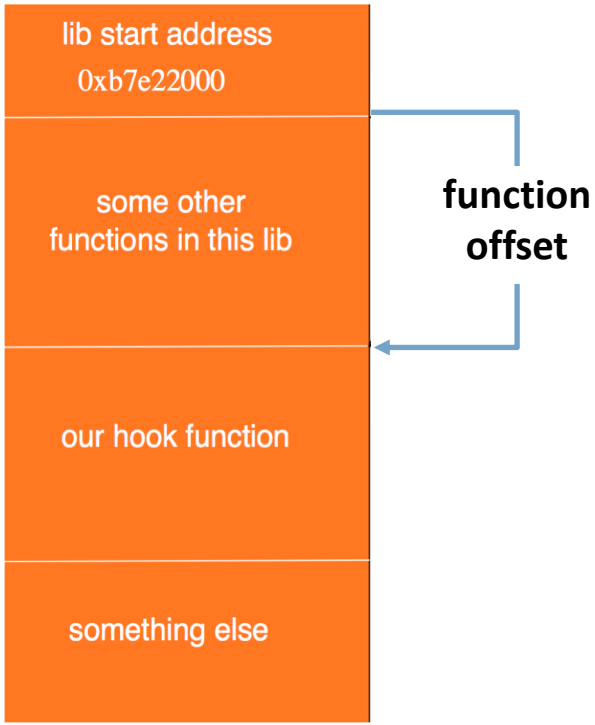
```
/*
 * This method will open /proc/<pid>/maps and search for the specified
 * library base address.
 */
uintptr_t findLibrary( const char *library, pid_t pid = -1 ) {
    char filename[0xFF] = {0},
         buffer[1024] = {0};
    FILE *fp = NULL;
    uintptr_t address = 0;

    sprintf( filename, "/proc/%d/maps", pid == -1 ? _pid : pid );

    fp = fopen( filename, "rt" );
```

```
    while( fgets( buffer, sizeof(buffer), fp ) ) {
        if( strstr( buffer, library ) ){
            address = (uintptr_t)strtoul( buffer, NULL, 16 );
            goto done;
        }
    }
```

инъекция библиотеки

если бы всё было просто…

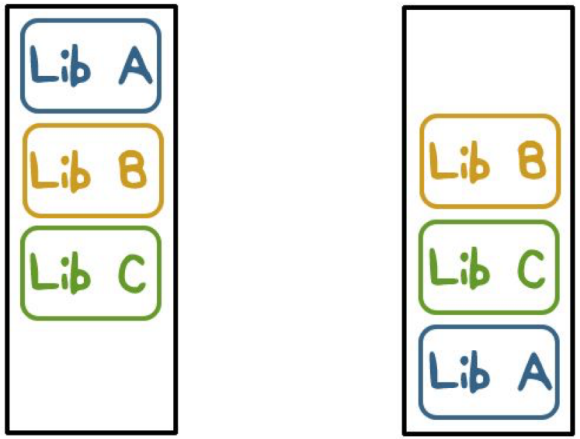
libc function address
=
libc start address
+
function offset



ASLR

Address Space Layout Randomization

- Stack, heap, main executable, and dynamic libraries.



Memory Layout

method hooking

мы знаем адрес mmap в нашем процессе

мы знаем адрес библиотеки /system/lib/libc.so (в котором лежит функция mmap) **в нашем процессе**

мы знаем адрес где загружена библиотека в стороннем процессе

знаем, что отступ от начал всегда одинаковый

injector

```
/*
 * Compute the delta of the local and the remote modules and apply it to
 * the local address of the symbol ... BOOM, remote symbol address!
 */
void *findFunction( const char* library, void* local_addr ){
    uintptr_t local_handle, remote_handle;

    local_handle = findLibrary( library, getpid() );
    remote_handle = findLibrary( library );

    remote_address = (void *) (uintptr_t)remote_handle + ((uintptr_t)local_handle - (uintptr_t)local_addr);
    return remote_address;
}
```

REMOTE_ADDRESS = REMOTE_HANDLE + (LOCAL_ADDR - LOCAL_HANDLE)

функция Ptrace()

#include <sys/ptrace.h>

long **ptrace**(enum __ptrace_request **request**, pid_t **pid**,

void ***addr**, void ***data**);

коротко об инъекции

ARM Call Convention

PC (R15)

(англ. program counter) — регистр процессора, который указывает, какую команду нужно выполнять следующей

LR (R14)

(англ. Link register) — регистр, хранящий адрес, куда будем возвращаться после выполнения функции

R0-R3

вспомогательные регистры. Сначала хранят первые 4 аргумента, после выполнения в **R0** лежит значение выполнившейся функции

R13

регистр, в котором хранится адрес на стек

ARM Call Convention

используем ptrace_getregs для сохранения текущих регистров

расставляем регистры в соответствии с конвенцией (r0 – r3, если мало – кладём в стек)

ставим 0 в lr, чтобы мы смогли поймать sigsegv после вызова функции

кладём в pc адрес функции

обновляем регистры с помощью ptrace_setregs

стартуем функцию с помощью ptrace_cont и ждём sigsegv (потому что 0 в lr)

забираем и сохраняем данные из регистра r0

восстанавливаем оригинальные регистры

покажите мне код

```
unsigned long call( void *function, int nargs, ... ) {
    int i = 0;
    struct pt_regs regs = {{0}}, rbackup = {{0}};

    // get registers and backup them
    trace( PTRACE_GETREGS, 0, &regs );
    memcpy( &rbackup, &regs, sizeof(struct pt_regs) );

    va_list vl;
    va_start(vl,nargs);

    for( i = 0; i < nargs; ++i ){
        unsigned long arg = va_arg( vl, long );

        // fill R0-R3 with the first 4 arguments
        if( i < 4 ){
            regs.uregs[i] = arg;
        }
        // push remaining params onto stack
        else {
            regs.ARM_sp -= sizeof(long) ;
            write( (size_t)regs.ARM_sp, (uint8_t *)&arg, sizeof(long) );
        }
    }

    va_end(vl);

```

покажите мне код

```
unsigned long call( void *function, int nargs, ... ) {
    int i = 0;
    struct pt_regs regs = {{0}}, rbackup = {{0}};

    va_list vl;
    va_start(vl,nargs);

    for( i = 0; i < nargs; ++i ){
        unsigned long arg = va_arg( vl, long );

        // fill R0-R3 with the first 4 arguments
        if( i < 4 ){
            regs.uregs[i] = arg;
        }
        // push remaining params onto stack
        else {
            regs.ARM_sp -= sizeof(long) ;
            write( (size_t)regs.ARM_sp, (uint8_t *)&arg, sizeof(long) );
        }
    }

    va_end(vl);
```

покажите мне код

```
unsigned long call( void *function, int nargs, ... ) {
    int i = 0;
    struct pt_regs regs = {{0}}, rbackup = {{0}};

    // get registers and backup them
    trace( PTRACE_GETREGS, 0, &regs );
    memcpy( &rbackup, &regs, sizeof(struct pt_regs) );

        unsigned long arg = va_arg( vl, long );

        // fill R0-R3 with the first 4 arguments
        if( i < 4 ){
            regs.uregs[i] = arg;
        }
        // push remaining params onto stack
        else {
            regs.ARM_sp -= sizeof(long) ;
            write( (size_t)regs.ARM_sp, (uint8_t *)&arg, sizeof(long) );
        }
    }

    va_end(vl);
```

покажите мне код

```
unsigned long call( void *function, int nargs, ... ) {
    int i = 0;
    struct pt_regs regs = {{0}}, rbackup = {{0}};

    // get registers and backup them
    trace( PTRACE_GETREGS, 0, &regs );
```

```
    for( i = 0; i < nargs; ++i ){
        unsigned long arg = va_arg( vl, long );

        // fill R0-R3 with the first 4 arguments
        if( i < 4 ){
            regs.uregs[i] = arg;
        }
        // push remaining params onto stack
        else {
            regs.ARM_sp -= sizeof(long) ;
            write( (size_t)regs.ARM_sp, (uint8_t *)&arg, sizeof(long) );
        }
    }
```

покажите мне код

```
28        regs.ARM_lr = 0;
29        regs.ARM_pc = (long int)function;
```

```
34            regs.ARM_cpsr |= CPSR_T_MASK;
35        }
36        else{
37            /* arm */
38            regs.ARM_cpsr &= ~CPSR_T_MASK;
39        }
40
41        // do the call
42        trace( PTRACE_SETREGS, 0, &regs );
43        trace( PTRACE_CONT );
44        waitpid( _pid, NULL, WUNTRACED );
45
46        // get registers again, R0 holds the return value
47        trace( PTRACE_GETREGS, 0, &regs );
48
49        // restore original registers state
50        trace( PTRACE_SETREGS, 0, &rbackup );
51
52        return regs.ARM_r0;
53    }
```

покажите мне код

```
28      regs.ARM_lr = 0;
29      regs.ARM_pc = (long int)function;
30      // setup the current processor status register
31      if ( regs.ARM_pc & 1 ){
32          /* thumb */
33          regs.ARM_pc   &= (~1u);
34          regs.ARM_cpsr |= CPSR_T_MASK;
35      }
36      else{
41      // do the call
42      trace( PTRACE_SETREGS, 0, &regs );
43      trace( PTRACE_CONT );
44      waitpid( _pid, NULL, WUNTRACED );
45
46      // get registers again, R0 holds the return value
47      trace( PTRACE_GETREGS, 0, &regs );
48
49      // restore original registers state
50      trace( PTRACE_SETREGS, 0, &rbackup );
51
52      return regs.ARM_r0;
53  }
```

покажите мне код

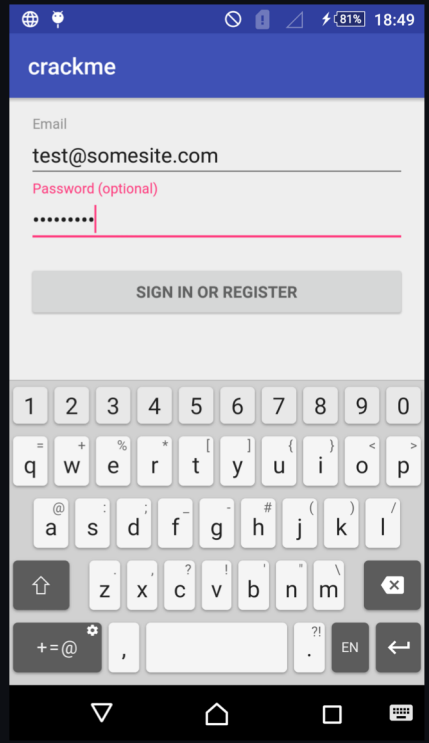
```
28      regs.ARM_lr = 0;
29      regs.ARM_pc = (long int)function;
30      // setup the current processor status register
31      if ( regs.ARM_pc & 1 ){
32          /* thumb */
33          regs.ARM_pc   &= (~1u);
34          regs.ARM_cpsr |= CPSR_T_MASK;
35      }
36      else{
37          /* arm */
38          regs.ARM_cpsr &= ~CPSR_T_MASK;
```

```
46      // get registers again, R0 holds the return value
47      trace( PTRACE_GETREGS, 0, &regs );
48
49      // restore original registers state
50      trace( PTRACE_SETREGS, 0, &rbackup );
51
52      return regs.ARM_r0;
```

```
52      return regs.ARM_r0;
53  }
```

xposed

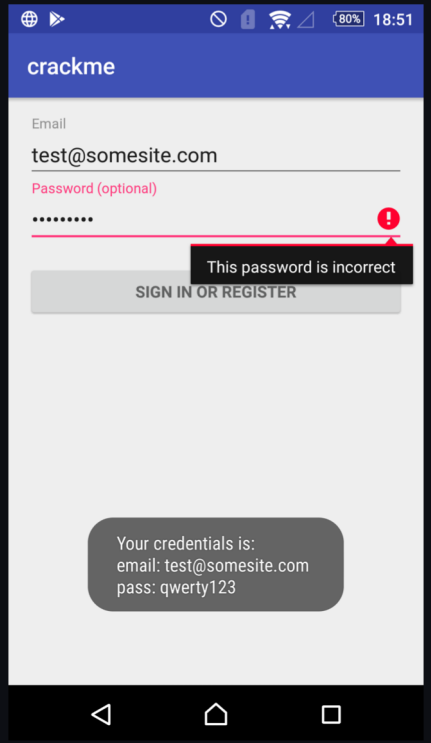
```
90      private fun checkPass(mEmail: String, mPassword: String): Boolean {
91          return CREDENTIALS.map { it.split( ...delimiters: ":") }
92                  .firstOrNull { it[0] == mEmail }
93                  ?.let { it: List<String>
94              // Account exists, return true if the password matches.
95              it[1] == mPassword
96          } ?: false
97      }
```



xposed

```
12
13  public class XposedModExample implements IXposedHookLoadPackage {
14      public void handleLoadPackage(XC_LoadPackage.LoadPackageParam lpparam) {
15          if (!lpparam.packageName.equals("ru.andrroider.apps.crackme"))
16              return;
17
18          findAndHookMethod( className: "ru.andrroider.apps.crackme.LoginActivity", lpparam.classLoader,
        methodName: "checkPass", String.class, String.class, new XC_MethodHook() {
19              @Override
20              protected void afterHookedMethod(MethodHookParam param) {
21                  String email = (String) param.args[0];
22                  String pass = (String) param.args[1];
23                  Toast.makeText(AndroidAppHelper.currentApplication(),
24                          text: "Your credentials is:\nemail: " + email + "\npass: " + pass, Toast.LENGTH_LONG).show();
25              }
26          });
27      }
28  }
```

xposed


80% 18:51
crackme
Email
test@somesite.com
Password (optional)
This password is incorrect
SIGN IN OR REGISTER
Your credentials is:
email: test@somesite.com
pass: qwerty123

способы защиты: buildTypes & Proguard

```
buildTypes {
  debug {
     minifyEnabled false
     applicationIdSuffix ".debug"
  }
  release {
     minifyEnabled true
     proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
  }
}
```

способы защиты: стек вызовов

```
try {
  throw new Exception("blablabla");
}
catch(Exception e) {
  for(StackTraceElement stackTraceElement : e.getStackTrace()) {
    Log.wtf("HookDetection", stackTraceElement.getClassName() + "->" + stackTraceElement.getMethodName());
  }
}
```

способы защиты: стек вызовов

com.example.hookdetection.DoStuff->getSecret

de.robv.android.xposed.XposedBridge->invokeOriginalMethodNative

de.robv.android.xposed.XposedBridge->handleHookedMethod

com.example.hookdetection.DoStuff->getSecret

com.example.hookdetection.MainActivity->onCreate

……

com.android.internal.os.ZygoteInit->main

de.robv.android.xposed.XposedBridge->main

dalvik.system.NativeStart->main

способы защиты

разбросать проверки на рут по всему приложению, а не все в одном месте

проверки на рут на разных уровнях (java/native)

проверки на дебаг

на разных уровнях (Java vs Native)

```
fun isBeingDebugged(): Boolean = Debug.isDebuggerConnected()
```

VS

```
JNIEXPORT jboolean JNICALL Java_com_test_example_DebuggerConnectedJNI(JNIenv * env, jobject obj) {
  if (gDvm.debuggerConnected || gDvm.debuggerActive)
     return JNI_TRUE;
  return JNI_FALSE;
}
```

способы защиты

проверки по всему приложению, а не все в одном месте

проверки на разных уровнях (java/native)

добавляйте оригинальность в способы проверки, а не просто копируйте с github/stackoverflow

способы защиты

работая над методами защиты от отладки ответьте на следующие вопросы:

можно ли обойти механизмы тривиально (например, перехватив (хукнув) одну функцию api)?

насколько сложно идентифицировать код отладки с помощью статического и динамического анализа?

нужно ли написать код, чтобы отключить защиту?

сколько времени на это нужно?

какова ваша субъективная оценка сложности обхода механизмов?

хардкорные способы

```
static boolean detect_threadCpuTimeNanos() {
  long start = Debug.threadCpuTimeNanos();

  for(int i=0; i<1000000; ++i)
    continue;

  long stop = Debug.threadCpuTimeNanos();
  return stop - start < 10000000
}
```

slides.night-labs.de

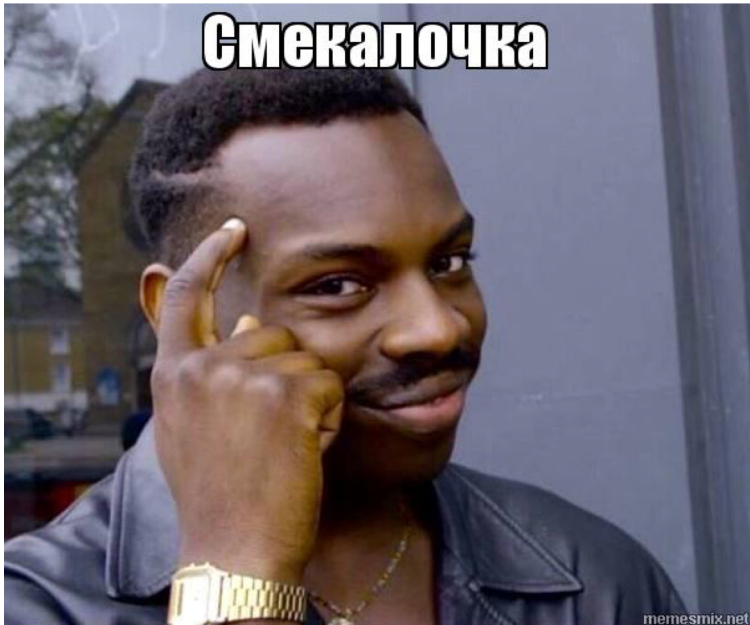
способы защиты

```
public static boolean hasTracerPid() throws IOException {
    BufferedReader reader = null;
    String tracerpid = "TracerPid";
    try {
        reader = new BufferedReader(new InputStreamReader(new FileInputStream( name: "/proc/self/status")),
: 1000);
        String line;

        while ((line = reader.readLine()) != null) {
            if (line.length() > tracerpid.length()) {
                if (line.substring(0, tracerpid.length()).equalsIgnoreCase(tracerpid)) {
                    if (Integer.decode(line.substring(tracerpid.length() + 1).trim()) > 0) {
                        return true;
                    }
                    break;
                }
            }
        }

    } catch (Exception exception) {
        exception.printStackTrace();
    } finally {
        reader.close();
    }
    return false;
}
```

github.com

способы защиты

```
void fork_and_attach() {
 int pid = fork();

 if (pid == 0)
   {
    int ppid = getppid();

    if (ptrace(PTRACE_ATTACH, ppid, NULL, NULL) == 0)
     {
       waitpid(ppid, NULL, 0);

       /* Continue the parent process */
       ptrace(PTRACE_CONT, NULL, NULL);
     }
   }
}
```



github.com/Jacks0n23

instagram.com/alexandr23

t.me/Jacks0n23