Genymobile

BE PARANOID
OR
NOT TO BE ?

Alizée PENEL

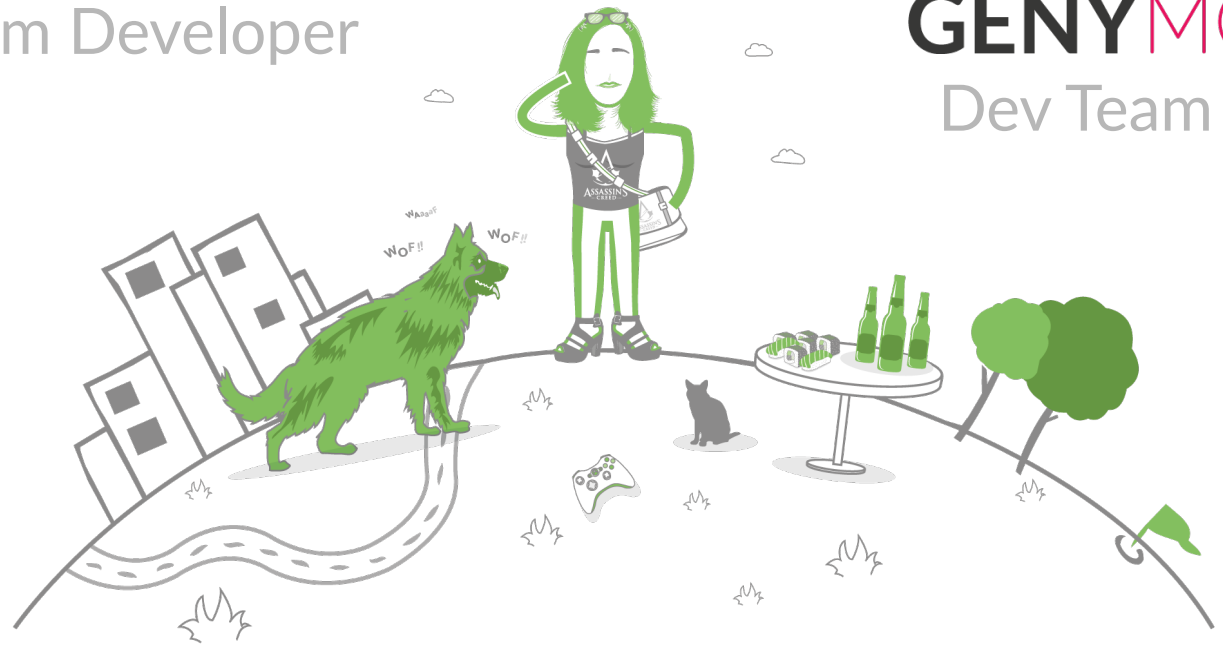
Linux and Android
System Developer

Genymobile

GENYMASTER

GENYMOTION

Dev Team Member



Agenda

Genymobile

01

Internet Permission in Marshmallow

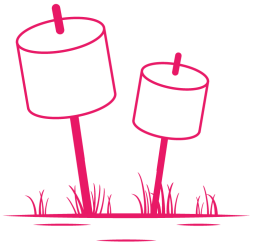
02

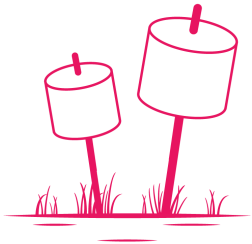
Network socket in Android OS

03

Security Aspects

INTERNET PERMISSION IN MARSHMALLOW





INTERNET PERMISSION DECLARATION

Genymobile

AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
        package="sk.vx.connectbot"
        android:versionName="1.7.1-29"
        android:versionCode="29"
        android:installLocation="auto">

        <uses-sdk android:targetSdkVersion="11" android:minSdkVersion="8" />

        <uses-permission android:name="android.permission.INTERNET" />
```

https://github.com/vx/connectbot from VX Solutions

INTERNET PERMISSION DEFINITION

Genymobile

frameworks/base/core/AndroidManifest.xml

```
<!-- Allows applications to open network sockets.
     <p>Protection level: normal
-->
<permission android:name="android.permission.INTERNET"
    android:description="@string/permdesc_createNetworkSockets"
    android:label="@string/permlab_createNetworkSockets"
    android:protectionLevel="normal" />
```

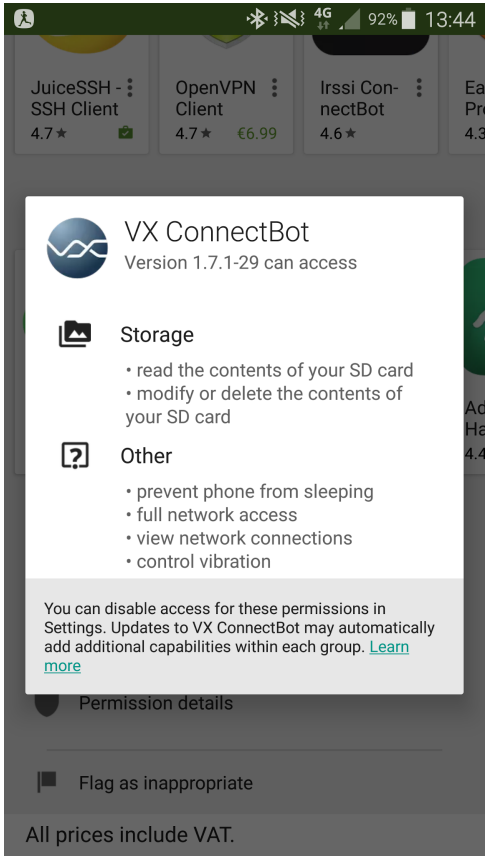
MARSHMALLOW PERMISSIONS

Genymobile

Permission are automatically granted at install time

- UI shows permissions details
- UI from Google Play, not from the system

Dangerous permissions are granted at runtime



INTERNET PERMISSION INTERNALS

Genymobile

On device : /system/etc/permissions/platform.xml

```
<permission name="android.permission.INTERNET" >
    <group gid="inet" />
</permission>
```

system/core/include/private/android_filesystem_config.h

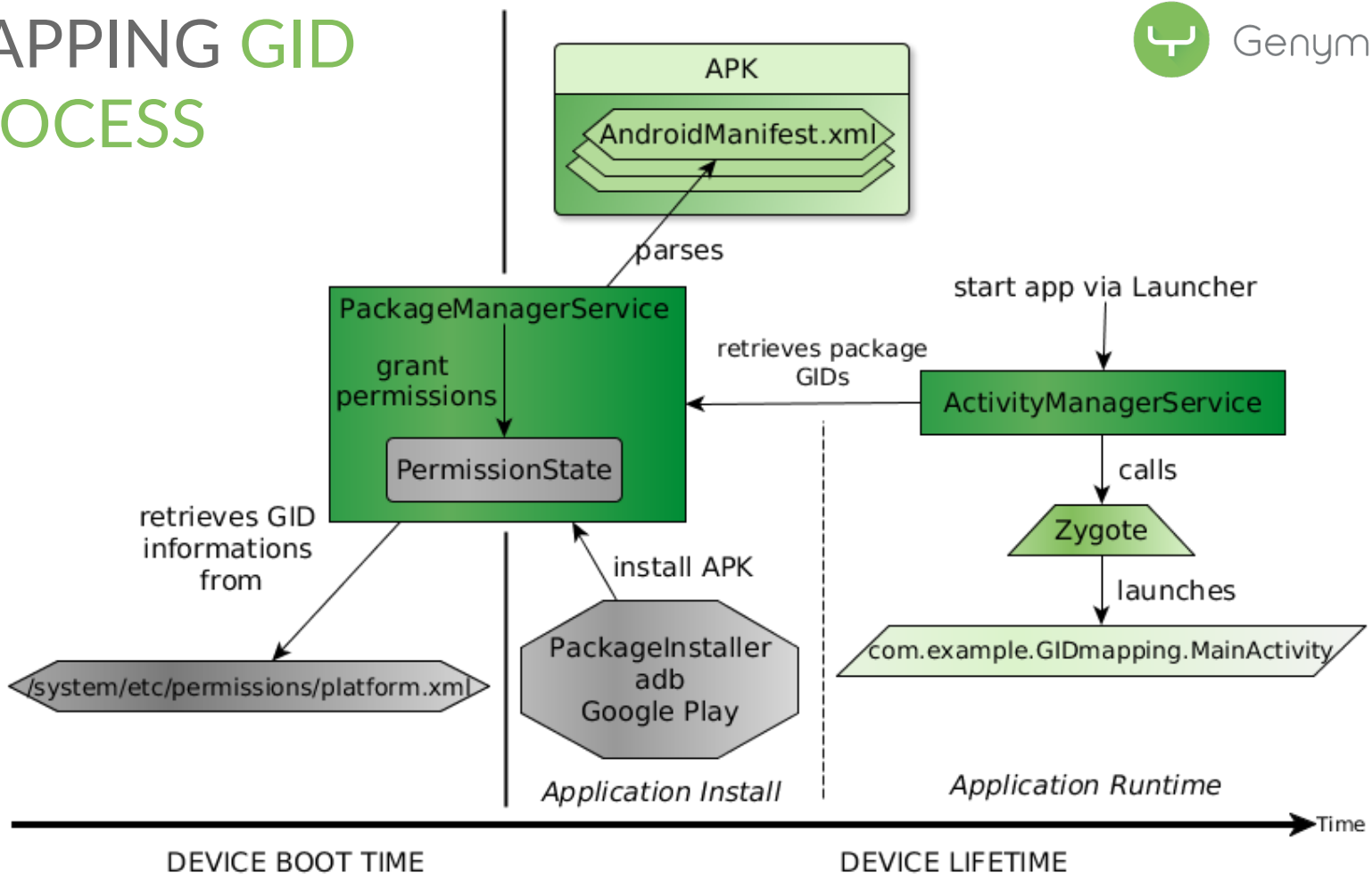
```
/* The 3000 series are intended for use as supplemental group id's only.
 * They indicate special Android capabilities that the kernel is aware of. */
#define AID_NET_BT_ADMIN  3001  /* bluetooth: create any socket */
#define AID_NET_BT        3002  /* bluetooth: create sco, rfcomm or l2cap sockets */
#define AID_INET          3003  /* can create AF_INET and AF_INET6 sockets */
#define AID_NET_RAW       3004  /* can create raw INET sockets */
```

root@genymotion:/ cat /data/system/packages.list

```
sk.vx.connectbot 10070 0 /data/data/sk.vx.connectbot default 3003
```

MAPPING GID PROCESS

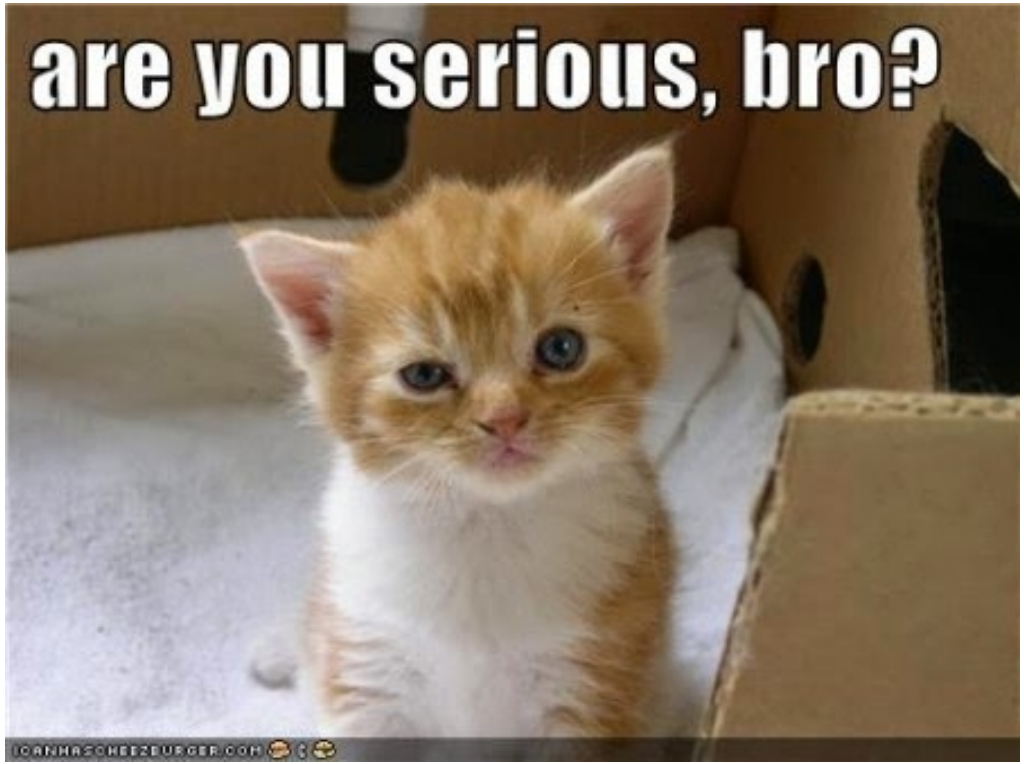
Genymobile



Genymobile

That's all ?

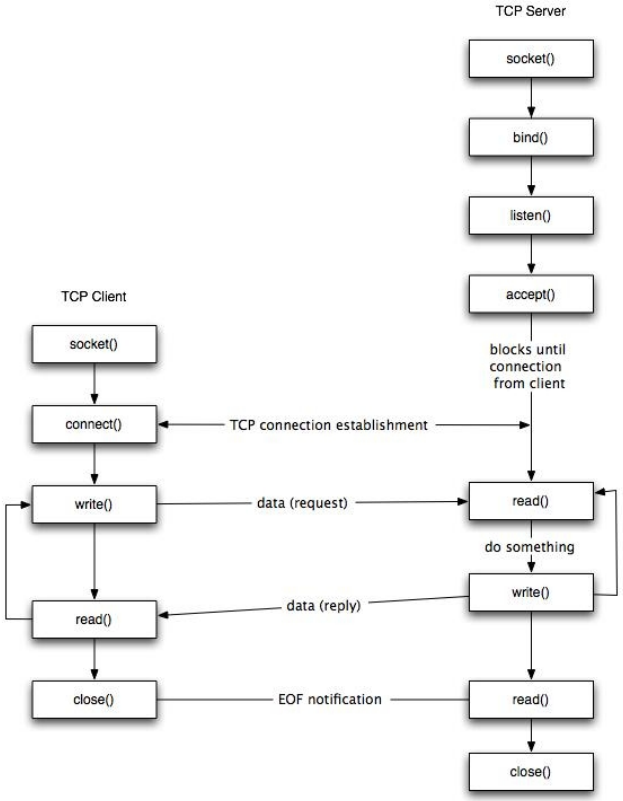
Anything is checked at the runtime ?



NETWORK SOCKETS
IN ANDROID OS

THE BASICS

Genymobile




```
package com.kynzie.talk.socketClient;

import java.net.Socket;
[...]

public class Client extends Activity {

        private Socket socket;

        private static final int PORT = 4242;
        private static final String IP = "163.5.224.242";

        @Override
        public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.main);

                new Thread(new ClientThread()).start();
        }

        public void onClick(View view) {
                [...] // Write something on the stream
        }

        class ClientThread implements Runnable {
                @Override
                public void run() {
                        try {
                                InetAddress serverAddr = InetAddress.getByName(IP);
                                socket = new Socket(serverAddr, PORT);

                        } catch (UnknownHostException e1) {
                                e1.printStackTrace();
                        } catch (IOException e1) {
                                e1.printStackTrace();
                        }
                }
        }
}
```

JAVA.NET.SOCKET CLASS

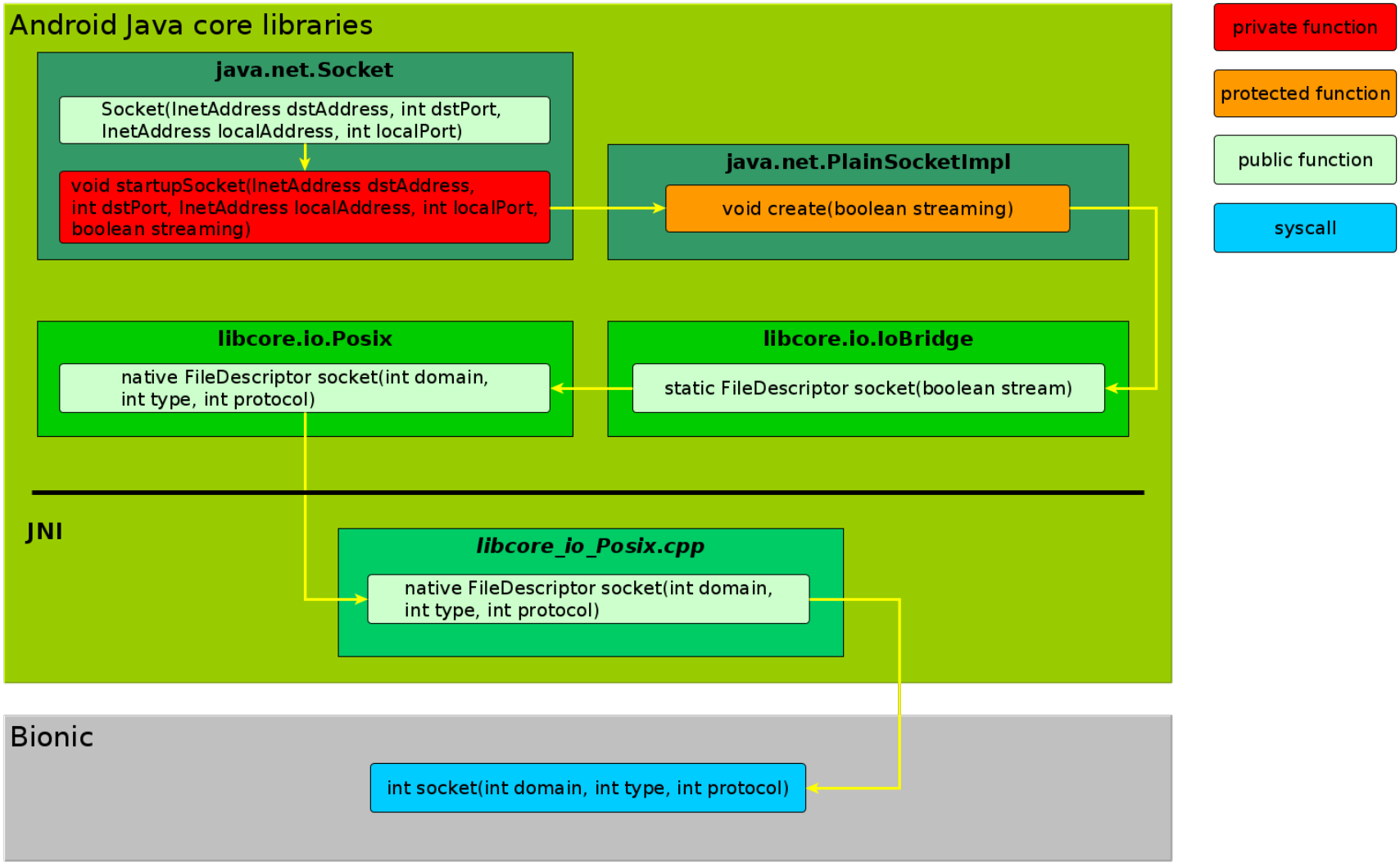
Genymobile

Any application can directly instantiate this class

Even the framework uses it

Packed in Android Java core library : core-libart.jar

Source file : libcore/luni/src/main/java/net/Socket.java


Android Java core libraries
java.net.Socket
Socket(InetAddress dstAddress, int dstPort, InetAddress localAddress, int localPort)
void startupSocket(InetAddress dstAddress, int dstPort, InetAddress localAddress, int localPort, boolean streaming)
java.net.PlainSocketImpl
void create(boolean streaming)
libcore.io.Posix
native FileDescriptor socket(int domain, int type, int protocol)
libcore.io.IoBridge
static FileDescriptor socket(boolean stream)
JNI
libcore_io_Posix.cpp
native FileDescriptor socket(int domain, int type, int protocol)
Bionic
int socket(int domain, int type, int protocol)
private function
protected function
public function
syscall

ANY PERMISSION CHECKED !?

Genymobile

SOCKET SYSCALL IN BIONIC

Genymobile

bionic/libc/bionic/socket.cpp

```
#include "private/NetdClientDispatch.h"

#include <sys/socket.h>

int socket(int domain, int type, int protocol) {
    return __netdClientDispatch.socket(domain, type, protocol);
}
```

Same type of declaration for connect and accept syscalls

NetdClientDispath, C structure of 4 function pointers on

3 syscalls (__socket, __connect, __accept4) & 1 function (fallBackNetIdForResolv)

WHAT HAPPENING IN BIONIC ?

Genymobile

As soon as bionic is loaded, the function **__libc_preinit()** is called by the dynamic linker

In **__libc_preinit()**, call to **netdClientInit()** function

The **libnetd_client.so** library is loaded by **dlopen()**

WHAT HAPPENING IN BIONIC ?

Genymobile

From **libnetd_client.so** library, bionic retrieves 4 function symbols :

- **netdClientInitSocket()**
- **netdClientInitConnect()**
- **netdClientInitAccept4()**
- **netdClientInitNetIdForResolv()**

Call them, one by one, with their respective syscall as a parameter.

NETDCLIENT LIBRARY

Genymobile

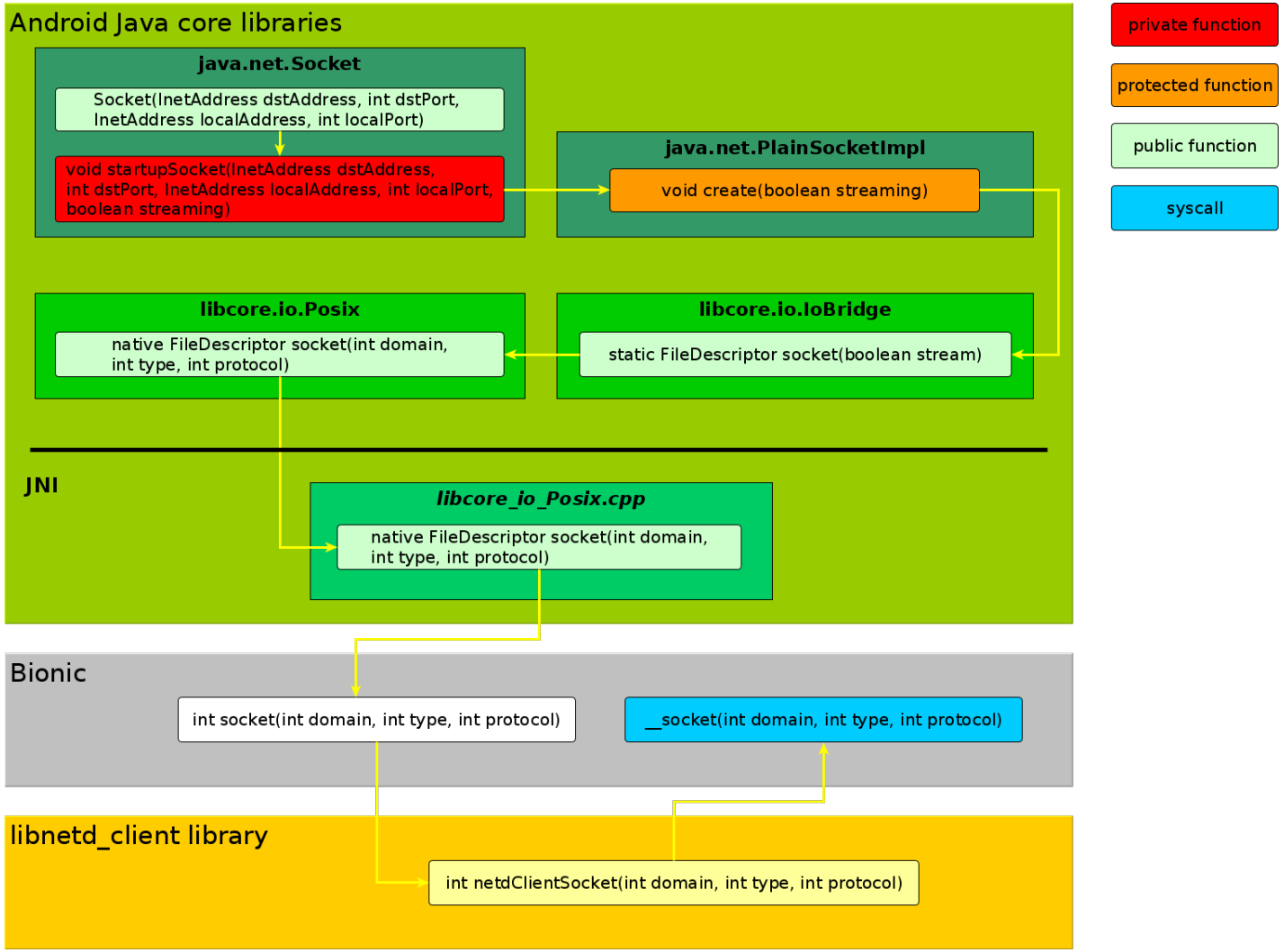
```
extern "C" void netdClientInitSocket(SocketFunctionType* function) {
    if (function && *function) {
        libcSocket = *function;
        *function = netdClientSocket;
    }
}
```

IMPACTS ON NETDCLIENTDISPATCH STRUCTURE

NetdClientDispatch structure does not contain the syscalls anymore

It points on **libnetd_client** library functions :

- **netdClientSocket()**
- **netdClientConnect()**
- **netdClientAccept4()**
- **getNetworkForResolv()**


Android Java core libraries
java.net.Socket
Socket(InetAddress dstAddress, int dstPort, InetAddress localAddress, int localPort)
void startupSocket(InetAddress dstAddress, int dstPort, InetAddress localAddress, int localPort, boolean streaming)
java.net.PlainSocketImpl
void create(boolean streaming)
libcore.io.Posix
native FileDescriptor socket(int domain, int type, int protocol)
libcore.io.IoBridge
static FileDescriptor socket(boolean stream)
JNI
libcore_io_Posix.cpp
native FileDescriptor socket(int domain, int type, int protocol)
Bionic
int socket(int domain, int type, int protocol)
__socket(int domain, int type, int protocol)
libnetd_client library
int netdClientSocket(int domain, int type, int protocol)
private function
protected function
public function
syscall

WHAT !?

Genymobile

ANDROID KERNEL

Genymobile

Android kernels have many modifications

Every Android kernel has a network option activated : Paranoid

PARANOID KERNEL OPTION

Genymobile

It restricts access to some networking features depending on the group of the calling process

include/linux/android_aids.h

```
/* AIDs that the kernel treats differently */
#define AID_OBSOLETE_000 3001  /* was NET_BT_ADMIN */
#define AID_OBSOLETE_001 3002  /* was NET_BT */
#define AID_INET         3003
#define AID_NET_RAW      3004
#define AID_NET_ADMIN    3005
#define AID_NET_BW_STATS 3006  /* read bandwidth statistics */
#define AID_NET_BW_ACCT  3007  /* change bandwidth statistics accounting */
```

SOCKET CREATION IN THE KERNEL

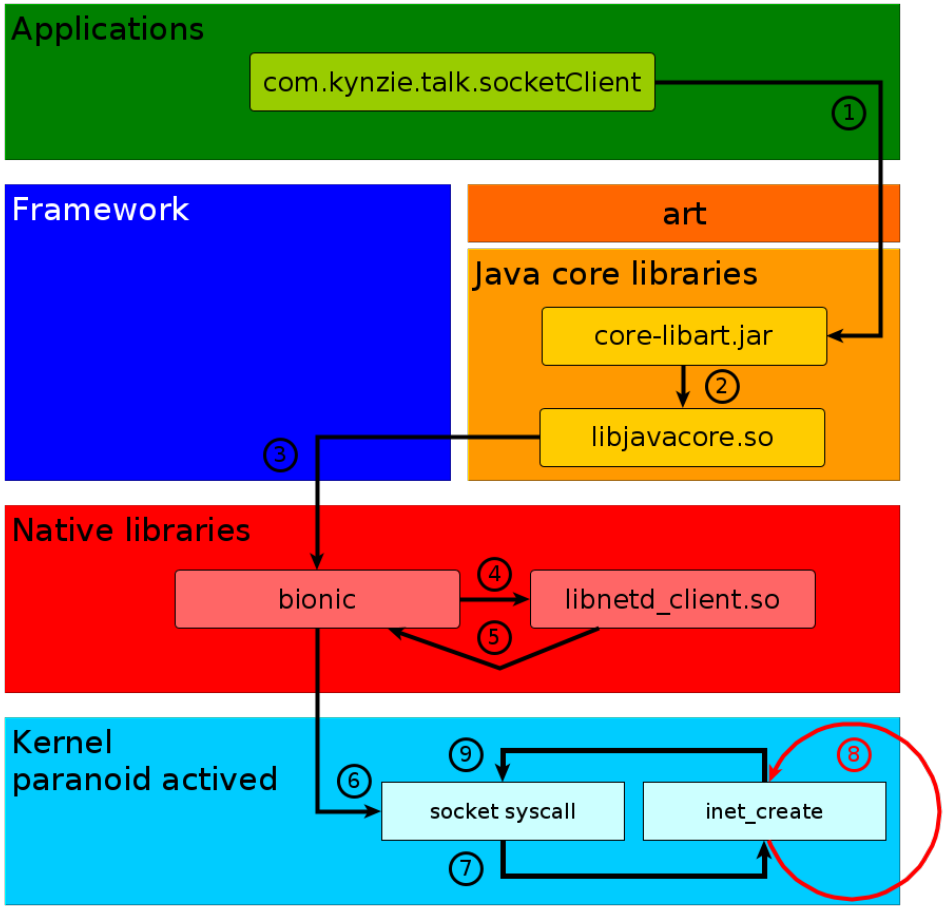
Genymobile

In net/ipv4/af_inet.c & net/ipv6/af_inet6.c, the process group is checked before creating the socket

If not allowed, return EACCES

SUMMARY

Genymobile



INTEREST OF NETDCLIENT LIBRARY AND BIONIC TRICK

Firewall marks in netd

Networks packets are flagged through a fwmark client/server mechanism

Allow packets going through iptable rules, set by the OS

In a “system case”, fwmark server checks also the permission of the process

SECURITY ASPECTS

DISCLAIMER

Genymobile

I am NOT a Security developer

Consider just the architectural aspect of the implementation

HOW TO BREAK THE SYSTEM ?

Genymobile

Internet permission

Paranoid option

Rooted devices

HOW TO BREAK THE SYSTEM ?

Genymobile

sharedUserId

- A way to share permissions between packages
- Permissions state is propagated to all packages upon changes

Other applications

Thanks for your attention !

PENEL Alizée
apenel@genymobile.com

www.genymobile.com

Genymobile

QUESTIONS ?