

# Dvmap: the first Android malware with code injection

By Roman Unuchek

Published: 2017-06-08 · Archived: 2026-04-06 00:09:20 UTC

**In April 2017 we started observing new rooting malware being distributed through the Google Play Store. Unlike other rooting malware, this Trojan not only installs its modules into the system, it also injects malicious code into the system runtime libraries. Kaspersky Lab products detect it as Trojan.AndroidOS.Dvmap.a.**

The distribution of rooting malware through Google Play is not a new thing. For example, the Ztorg Trojan has been [uploaded to Google Play almost 100](#) times since September 2016. But Dvmap is very special rooting malware. It uses a variety of new techniques, but the most interesting thing is that it injects malicious code into the system libraries – libdmv.so or libandroid\_runtime.so.

This makes Dvmap the first Android malware that injects malicious code into the system libraries in runtime, and it has been downloaded from the Google Play Store more than 50,000 times. Kaspersky Lab reported the Trojan to Google, and it has now been removed from the store.



# colourblock

Retgumhoap Kanumep Puzzle

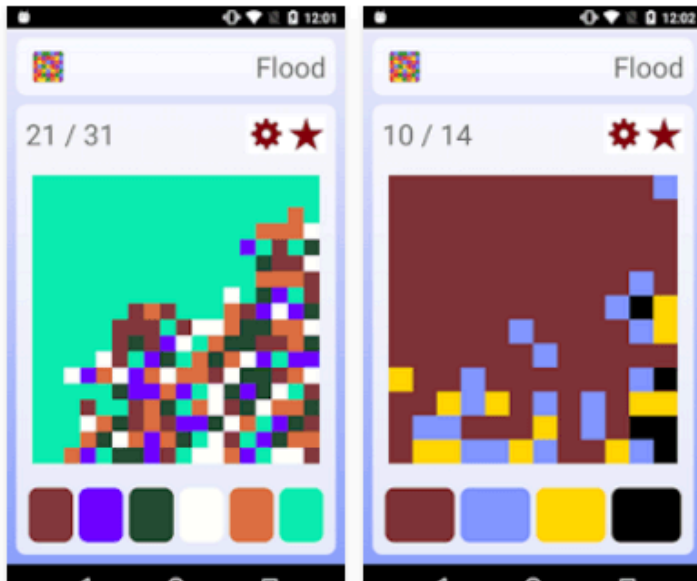
★★★★★ 75

Everyone

Contains ads

This app is compatible with all of your devices.

Installed

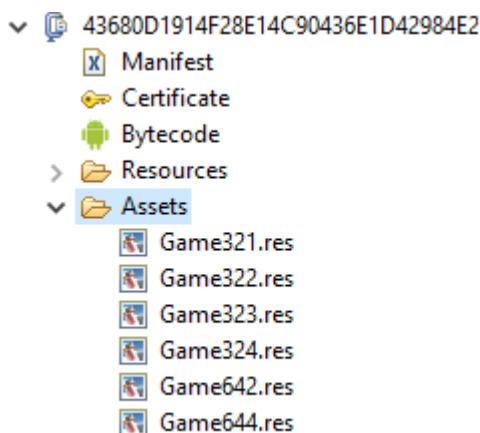


colourblock is a Simplest, Challenging, addictive puzzle game.

## *Trojan.AndroidOS.Dvmap.a on Google Play*

To bypass Google Play Store security checks, the malware creators used a very interesting method: they uploaded a clean app to the store at the end of March, 2017, and would then update it with a malicious version for short period of time. Usually they would upload a clean version back on Google Play the very same day. They did this at least 5 times between 18 April and 15 May.

All the malicious Dvmap apps had the same functionality. They decrypt several archive files from the assets folder of the installation package, and launch an executable file from them with the name “start.”



*Encrypted archives in the assets folder*

The interesting thing is that the Trojan supports even the 64-bit version of Android, which is very rare.

```
int v0 = h.e();
if(v0 == 1) {
    v2 = h.c[arg8];
    v0_1 = String.format(Locale.getDefault(), "Game32%d.res", Integer.valueOf(v2));
    v1 = "32" + v2;
    goto label_19;
}
else if(v0 == 2) {
    v2 = h.d[arg8];
    v0_1 = String.format(Locale.getDefault(), "Game64%d.res", Integer.valueOf(v2));
    v1 = "64" + v2;
label_19:
    h.c(arg6, "k135843", v1);
```

*Part of code where the Trojan chooses between 32-bit and 64-bit compatible files*

All encrypted archives can be divided into two groups: the first comprises Game321.res, Game322.res, Game323.res and Game642.res – and these are used in the initial phase of infection, while the second group: Game324.res and Game644.res, are used in the main phase.

## Initial phase

During this phase, the Trojan tries to gain root rights on the device and to install some modules. All archives from this phase contain the same files except for one called “common”. This is a local root exploit pack, and the Trojan uses 4 different exploit pack files, 3 for 32-bit systems and 1 for 64-bit-systems. If these files successfully gain root rights, the Trojan will install several tools into the system. It will also install the malicious app “com.qualcmm.timeservices.”

These archives contain the file “.root.sh” which has some comments in Chinese:

```
#### 参数4. APK所在目录(默认是MY_FILES_DIR目录)
APK_NAME=app-release.apk
SUAPK_PATH=$MY_FILES_DIR/$APK_NAME
if [ ! -f "$SUAPK_PATH" ]; then
    SUAPK_PATH=$MY_FILES_DIR/$APK_NAME
fi
if [ ! -z "$4" ]; then
    SUAPK_PATH=$MY_FILES_DIR/$APK_NAME
    if [ ! -f "$SUAPK_PATH" ]; then
        SUAPK_PATH=$4/$APK_NAME
    fi
fi
echo "SUAPK_PATH: $SUAPK_PATH"
```

Part of .root.sh file

## Main phase

In this phase, the Trojan launches the “start” file from Game324.res or Game644.res. It will check the version of Android installed and decide which library should be patched. For Android 4.4.4 and older, the Trojan will patch method `_Z30dvmHeapSourceStartupBeforeForkv` from `libdvm.so`, and for Android 5 and newer it will patch method `nativeForkAndSpecialize` from `libandroid_runtime.so`. Both of these libraries are runtime libraries related to [Dalvik](#) and [ART](#) runtime environments. Before patching, the Trojan will backup the original library with a name `bak_{original name}`.

```
_pid_t __fastcall dvmHeapSourceStartupBeforeFork(int a1, int a2, int a3, int a4)
{
    __pid_t result; // r001
    __pid_t v5; // r002
    int v6; // r002
    const char *v7; // [sp+0h] [bp-54h]@2
    int v8; // [sp+4h] [bp-50h]@2
    int v9; // [sp+24h] [bp-30h]@1
    int v10; // [sp+28h] [bp-2Ch]@1
    int v11; // [sp+2Ch] [bp-28h]@1

    v9 = a2;
    v10 = a3;
    v11 = a4;
    result = linux_eabi_syscall(__NR_fork);
    if ( !result )
    {
        v5 = linux_eabi_syscall(__NR_setsid);
        v7 = "/system/bin/ip";
        v8 = 0;
        v6 = linux_eabi_syscall(__NR_execve, "/system/bin/ip", (char *const * __attribute__((__org_arrdim(0,0))) )&v7, 0);
        result = 0;
    }
    return result;
}
```

Patched `libdvm.so`

During patching, the Trojan will overwrite the existing code with malicious code so that all it can do is execute `/system/bin/ip`. This could be very dangerous and cause some devices to crash following the overwrite. Then the Trojan will put the patched library back into the system directory. After that, the Trojan will replace the original `/system/bin/ip` with a malicious one from the archive (Game324.res or Game644.res). In doing so, the Trojan can be sure that its malicious module will be executed with system rights. But the malicious `ip` file does not contain

any methods from the original ip file. This means that all apps that were using this file will lose some functionality or even start crashing.

## Malicious module “ip”

This file will be executed by the patched system library. It can turn off “VerifyApps” and enable the installation of apps from 3<sup>rd</sup> party stores by changing system settings. Furthermore, it can grant the “com.qualcomm.timeservices” app Device Administrator rights without any interaction with the user, just by running commands. It is a very unusual way to get Device Administrator rights.

As I mentioned before, in the “initial phase”, the Trojan will install the “com.qualcomm.timeservices” app. Its main purpose is to download archives and execute the “start” binary from them. During the investigation, this app was able to successfully connect to the command and control server, but it received no commands. So I don’t know what kind of files will be executed, but they could be malicious or advertising files.

## Conclusions

This Trojan was distributed through the Google Play Store and uses a number of very dangerous techniques, including patching system libraries. It installs malicious modules with different functionality into the system. It looks like its main purpose is to get into the system and execute downloaded files with root rights. But I never received such files from their command and control server.

These malicious modules report to the attackers about every step they are going to make. So I think that the authors are still testing this malware, because they use some techniques which can break the infected devices. But they already have a lot of infected users on whom to test their methods.

I hope that by uncovering this malware at such an early stage, we will be able to prevent a massive and dangerous attack when the attackers are ready to actively use their methods.

## MD5

43680D1914F28E14C90436E1D42984E2  
20D4B9EB9377C499917C4D69BF4CCEBE

---

Source: <https://securelist.com/dvmap-the-first-android-malware-with-code-injection/78648/>