Root Scanner for Android

Root Scanner provides a comprehensive way to detect **rooted Android** devices and **emulator environments**. The plugin uses multiple detection methods to reliably identify rooted or virtualized environments, helping developers implement appropriate security measures for their applications.

> ⚠️ **Platform Supports:** ✅ Android **Api 23.0+**

Features

Root Scanner utilizes multiple detection techniques to identify rooted devices and emulators. Below is a breakdown of each detection method:

1. Root Management and Dangerous App Detection

Checks for the presence of well-known root management and dangerous apps, including:

- **Root management tools** like:
  - Magisk
  - SuperSU
  - KingRoot
  - Framaroot
- **Dangerous tools** like:
  - Lucky Patcher
  - Freedom
  - Xposed Framework
  - Cheat Engines
  - Blackmart and similar stores

2. Binary Checks

Scans for dangerous binaries in typical system locations:

- `su` binary (Superuser)
- `busybox` binary (multi-tool used in rooted environments)
- `magisk` binary (Magisk root manager)

3. System Properties Checks

Detects dangerous system configurations by inspecting properties like:

- `ro.debuggable=1`
- `ro.secure=0`

These indicate a system built for development or insecure use.

4. Writable System Paths Check

Verifies if restricted system directories are **writable**, such as:

- `/system`
- `/system/bin`
- `/system/xbin`
- `/vendor/bin`
- `/sbin`

Writable access to these indicates possible root.

5. Test-Keys Detection

Checks if the firmware is signed with **test-keys**, commonly used in non-production builds.

6. Native Binary Detection (JNI)

Uses **native C++ checks** to detect low-level traces of root by scanning binary paths directly from native code, which is harder to bypass.

7. Emulator Detection

Identifies if the app is running on a virtual environment like:

- **Nox Player**
- **BlueStacks**
- **LDPlayer**
- **Genymotion**
- **Android Studio Emulator**

The detection is based on properties such as:

- Build fingerprint starting with `generic`
- Device model containing `Emulator`, `Google SDK`, or `x86`
- Manufacturer containing `Genymotion` or `Nox`
- Product names like `google_sdk` or `nox`

RootScanner

Member	Description
static bool IsRooted { get; }	Returns `true` if the device is detected as rooted, `false` otherwise.
static bool IsEmulator { get; }	Returns `true` if the app is running in an emulator, `false` if on a real device.
static bool IsPlatformSupported { get; }	Indicates whether root detection is available on the current platform (**Android** only).

Example Usage

```
bool isRooted = RootScanner.IsRooted;
bool isEmulator = RootScanner.IsEmulator;

if (isEmulator)
{
    Debug.LogWarning("App is running in an emulator environment.");
}

if (isRooted)
{
    Debug.LogWarning("Root access detected on the device.");
}
else
{
    Debug.Log("Device appears to be secure (no root detected).");
}
```