National University of Computer and Emerging Sciences
Islamabad Campus

CY2002

Digital Forensics

Project

ANDROID ROOT DETECTOR

Submitted by: MANAHIL CHOUDHRY, ASAD KHURSHID, EHTISHAM Ul-HASSAN
Roll number: I22-1728, I221585 ,I22-1777
Date: 17 Nov, 2024

Table of Contents

Introduction

The Android Device Analysis Tool is a Python-based script designed to assist in the analysis of Android devices through ADB (Android Debug Bridge). This tool provides essential functionalities for digital forensics and device analysis, including root detection, partition information retrieval, and installed application listing. The script is intended for use by digital forensics professionals, security researchers, and Android developers who need to gather critical information from Android devices quickly and efficiently.

Details and Steps

This guide provides a step-by-step process to run the root detection script.

Prerequisites

Before starting, ensure the following are in place:

1. Python 3.x
2. ADB (Android Debug Bridge) installed and accessible from the command line
3. An Android device with USB debugging enabled

Enable USB Debugging on the Android Device

1. Open the **Settings** menu on the Android device.
2. Go to **About Phone** and tap **Build Number** multiple times (usually 7) to enable Developer Mode.
3. Return to **Settings** and open **Developer Options**.
4. Locate **USB Debugging** and enable it.
5. Confirm the warning message by selecting **OK**.

Connect the Android Device to the PC

1. Use a USB cable to connect the Android device to the PC.
2. Ensure the device prompts you to allow USB debugging access. Select **OK** to allow the connection.

Running the Script

1. Open the terminal or command prompt on your PC.
2. Navigate to the directory where the Python script is located.
3. Execute the script using Python (refer to the script's user instructions for exact commands).
4. Run the script using Python: `python main.py`
5. Navigate through the menu options to perform desired operations.

The script performs several checks and operations, including ADB connection verification, root indicator checks, partition table retrieval, and installed app listing.

Check Root Status

This feature examines the device for indicators of root access. It performs the following checks:

- Searches for root management apps (e.g., SuperSU, Magisk)
- Checks for the presence of the `su` binary
- Examines build properties for test keys
- Verifies if the system partition is mounted as read-write

The function provides a list of detected root indicators, if any, along with a warning about potential false positives or negatives.

```
Android Device Analysis Tool
1. Check root status
2. Fetch partition table
3. Fetch names of installed apps
4. Exit
Enter your choice (1-4): 1

Checking for root indicators...
This may take a few moments...

Warning: Could not perform Root Management Apps check
Warning: Could not perform Su Binary check
Warning: Could not perform Build Properties check
No root indicators found.
Device appears to be unrooted, but root could still be hidden.

Note: This script provides indicators only and may not detect all root methods.
False positives and false negatives are possible.
```

Fetch Partition Table

This feature retrieves and displays the partition information of the connected device. It performs the following steps:

- Fetches the device model name
- Executes the `df -h` command to get partition information
- Saves the partition information to a text file named after the device model
- Displays the partition information on the console

This feature is useful for understanding the storage structure of the device and identifying potential areas for further investigation.

```
Android Device Analysis Tool
1. Check root status
2. Fetch partition table
3. Fetch names of installed apps
4. Exit
Enter your choice (1-4): 2
Connected devices: 1
adb_server_is_out_of_date.__killing..._daemon_started_successfully___SM-A022F
Partition information saved to SM-A022F_partition_info.txt

Partition Information:
adb server is out of date.  killing...
* daemon started successfully *
Filesystem            Size  Used Avail Use% Mounted on
/dev/block/dm-4       2.2G  2.2G     0 100% /
tmpfs                 1.4G  2.3M  1.4G   1% /dev
tmpfs                 1.4G     0  1.4G   0% /mnt
/dev/block/dm-5       207M  201M  1.7M 100% /vendor
/dev/block/dm-6       0.9G  0.9G     0 100% /product
/dev/block/dm-7       3.9M  1.2M  2.6M  33% /odm
/dev/block/dm-8       387M   28M  352M   8% /prism
/dev/block/dm-9        23M  756K   22M   4% /optics
tmpfs                 1.4G     0  1.4G   0% /apex
/dev/block/mmcblk0p46 193M   14M  175M   8% /cache
tmpfs                 1.4G     0  1.4G   0% /mnt/sde
/dev/block/mmcblk0p2  3.8M  324K  3.3M   9% /efs
/dev/block/mmcblk0p48  43M   24K   42M   1% /omr
/dev/block/mmcblk0p50  24G   22G  1.5G  94% /data
/dev/block/dm-10      804K  776K   12K  99% /apex/com.android.tzdata@305400100
```

Fetch Names of Installed Apps

This feature lists all applications installed on the device. It:

- Executes the `pm list packages` command via ADB
- Processes the output to extract package names
- Displays the list of installed applications on the console

This feature can help identify potentially malicious or unauthorized applications on the device.

```
Android Device Analysis Tool
1. Check root status
2. Fetch partition table
3. Fetch names of installed apps
4. Exit
Enter your choice (1-4): 3
Installed apps:
adb server is out of date.  killing...
* daemon started successfully *
com.google.android.networkstack.tethering
com.samsung.android.provider.filterprovider
com.whatsapp.w4b
com.sec.android.app.DataCreate
com.android.cts.priv.ctsshim
com.samsung.android.smartswitchassistant
com.sec.vsim.ericssonnsds.webapp
com.sec.android.app.setupwizardlegalprovider
com.google.android.youtube
com.samsung.android.app.galaxyfinder
com.sec.location.nsflp2
com.sec.android.app.chromecustomizations
com.android.internal.display.cutout.emulation.corner
com.google.android.ext.services
com.android.internal.display.cutout.emulation.double
com.sec.location.nfwlocationprivacy
com.android.providers.telephony
com.sec.android.app.ve.vebgm
com.sec.android.app.parser
com.android.dynsystem
com.samsung.internal.systemui.navbar.gestural_no_hint_wide_back
```

```
Partition Information Contents:
00000000  2f 64 65 76 2f 62 6c 6f 63 6b 2f 64 6d 2d 34 00 00
00000010  74 6d 70 66 73 00 00 00 00 00 00 00 00 00 00 00 00
00000020  74 6d 70 66 73 00 00 00 00 00 00 00 00 00 00 00 00
00000030  2f 64 65 76 2f 62 6c 6f 63 6b 2f 64 6d 2d 35 00 00
00000040  2f 64 65 76 2f 62 6c 6f 63 6b 2f 64 6d 2d 36 00 00
00000050  2f 64 65 76 2f 62 6c 6f 63 6b 2f 64 6d 2d 37 00 00
00000060  2f 64 65 76 2f 62 6c 6f 63 6b 2f 64 6d 2d 38 00 00
00000070  2f 64 65 76 2f 62 6c 6f 63 6b 2f 64 6d 2d 39 00 00
00000080  74 6d 70 66 73 00 00 00 00 00 00 00 00 00 00 00 00
00000090  2f 64 65 76 2f 62 6c 6f 63 6b 2f 6d 6d 63 62 6c 6b
000000a0  74 6d 70 66 73 00 00 00 00 00 00 00 00 00 00 00 00
000000b0  2f 64 65 76 2f 62 6c 6f 63 6b 2f 6d 6d 63 62 6c 6b
000000c0  2f 64 65 76 2f 62 6c 6f 63 6b 2f 6d 6d 63 62 6c 6b
000000d0  2f 64 65 76 2f 62 6c 6f 63 6b 2f 6d 6d 63 62 6c 6b
```

Summary

The Android Device Analysis Tool provides a streamlined approach to gathering crucial information from Android devices. Its key strengths include:

- Easy-to-use command-line interface
- Multiple analysis features in a single tool
- Root detection capabilities
- Partition information retrieval
- Installed application listing

While the tool is powerful, it's important to note its limitations:

- Reliance on ADB, which requires USB debugging to be enabled
- Potential for false positives/negatives in root detection
- Limited to information accessible without root privileges

Overall, this tool serves as a valuable asset for initial device analysis in digital forensics investigations, security audits, and Android development processes. It provides a solid foundation for more in-depth analysis and can be extended with additional features as needed.