<Client Name>

<Date>

<Product Name> Security Assessment

Client Details

Company Name:

Contact Person:

Address: Address:

Email:

Telephone:

Document History

Version	Date	Author	Remark

Table of Contents

1. About Payatu

Payatu is a research-focused security testing service organization specialized in IoT and embedded products, web, mobile, cloud and infrastructure security assessments and in-depth technical security training. Our state-of-the-art research, methodologies, and tools ensure the safety of our client's assets.
At Payatu, we believe in following one's passion, and with that thought, we have created a world-class team of researchers and executors who are bending the rules to provide the best security services. We are a passionate bunch of folks working on the latest and leading-edge security technology.
We are proud to be part of a vibrant security community and don't miss any opportunity to give back. Some of the contributions in the following fields reflect our dedication and passion

nullcon - nullcon security conference is an annual security event held in Goa, India. After years of efforts put in the event, it has now become a world-renowned platform to showcase the latest and undisclosed research.
hardwear.io - Hardware security conference is an annual hardware security event held in The Hague, Netherlands. It is being organized to answer emerging threats and attacks on hardware. We aim to make it the largest platform where hardware security innovation happens.
Dedicated fuzzing infrastructure - We are proud to be one of the few security research companies to own an in-house infrastructure and hardware for distributed fuzzing of software such as browsers, client and server applications.
null - It all started with null - The open security community. It's a registered non-profit society and one of the most active security community. null is driven totally by passionate volunteers.
Open source - Our team regularly authors open source tools to aid in security learning and research.
Talks and Training - Our team delivers talks/highly technical training in various international security and hacking conference, i.e., DEFCON Las Vegas, BlackHat Las Vegas, HITB Amsterdam, Cansecwest Vancouver, nullcon Goa, HackinParis Paris, Brucon Belgium, zer0con Seoul, and PoC Seoul to name few.

We are catering to a diverse portfolio of clients across the world, who are leaders in banking, finance, technology, healthcare, manufacturing, media houses, information security, and education, including government agencies. Having various empanelment and accreditations, along with a strong word of mouth has helped us win new customers, and our thorough professionalism and quality of work, have brought repeat business from our existing clients.

We thank you for considering our security services and requesting a proposal. We look forward to extending the expertise of our passionate, world-class professionals to achieve your security objectives.

2. Project Details

2.1 Executive Summary

The Payatu security team performs security assessments on the device MDM (Mobile Device Management). The aim of this assessment was to uncover any security issues in the assessed MDM that will allow the adversary to bypass the MDM controls or uninstall the MDM from the device as well as to explain the impact and risks associated with the found issues, and provide guidance in the prioritization and remediation steps.

Security Assessment of the product MDM has been performed, considering below common security issues:

- ✓ If MDM admirative control on the device can be disabled
- ✓ If FRP and other MDM features can be abused to uninstall MDM from the device.
- ✓ If MDM can be bypassed during the firmware update mechanism.
- ✓ If device/OS is prone to any vulnerability which help to take full control of device and uninstall MDM

Overall security postures of the product are moderate, though some of the security controls/measures have not been properly thought of/implemented.

The security assessment revealed 3 critical severity security issue, 0 high severity security issue, 1 medium severity issues, and 0 low severity issues in this product. The consolidated summary of the assessment has been presented in the Executive Summary section. Additional information is contained within the Detailed Vulnerability Information section of this report.

2.2 Scope and Objective

The scope of this assessment was limited to Device provided by <<Client>>.

2.3 Project Timeline

The security assessment was performed for <<xyz>> days from <<X date>> to <<Y date>>.

2.4 Technological Impact Summary

It was identified that the Device did not implement proper crash management, malformed application and Fault Injection in the device which could lead to putting in device in firmware update mode and compromise the security of the device.

2.5 Business Impact Summary

We identified the following business impacts:

- An attacker can get crash the device OS filesystem and factory reset the device and uninstall the MDM – (Financial Loss)
- An attacker can use engineering mode to crash the kernel and put the device in serial mode which could lead to removing of MDM from the device – (Financial Loss)
- An attacker can inject EM fault in the SRAM and put the device in SPRD mode which could led to removing the MDF from the device – (Financial Loss)

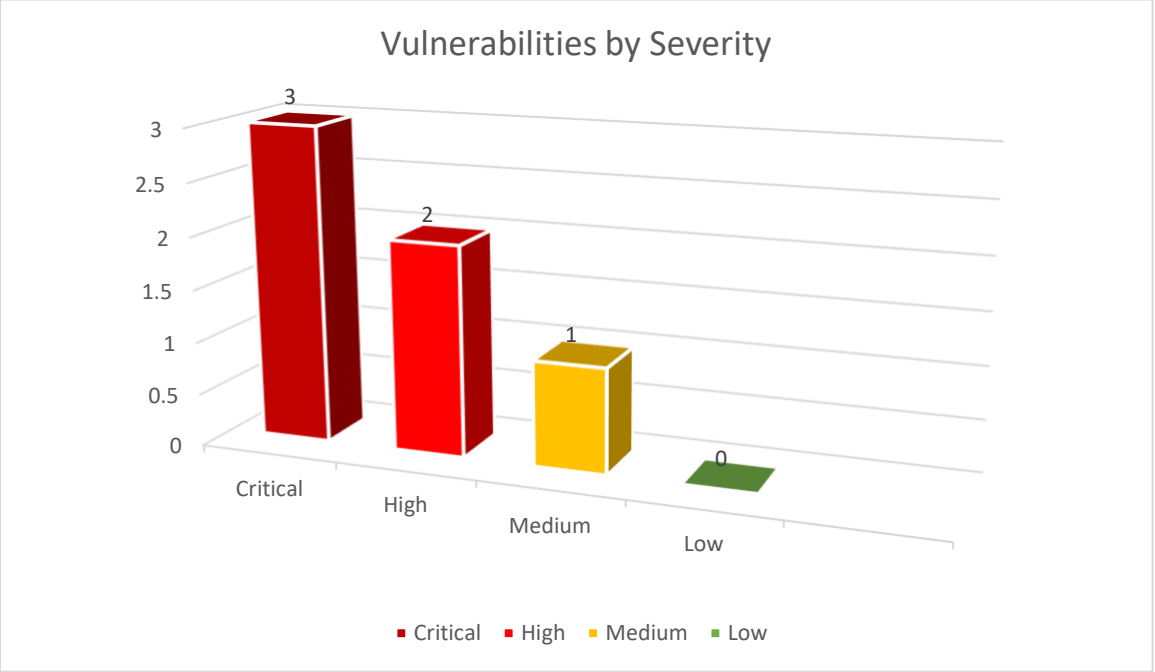
2.6 Testing Environment

To perform the device MDM test, various tools and software are used including adb, fastboot, EM Injector, Termux, SSH Server, etc.

2.7 Vulnerability Chart

The discovered vulnerabilities table and chart illustrated below, provides a snapshot view of the number and severity of issues discovered during this security assessment.

CRITICAL	This issue can impact the application severely and should be addressed immediately. Attackers can gain root or super user access or severely impact system operation.
HIGH	This issue can cause a problem like unprivileged access and should be addressed as soon as possible.
MEDIUM	This issue may pose a significant threat over a longer period of time.
LOW	This issue is more likely an information disclosure and may be an acceptable threat.



2.8 Table of Findings

Vuln ID	Finding	Severity	Status
PY-CSN-001	Factory Reset bypass by disabling core android application	Critical	Fixed
PY-CSN-002	Crashing Kernel using engineer mode to expose SPRD Mode	Critical	Not Fixed
PY-CSN-003	Crashing Kernel using EM Fault Injection to expose SPRD Mode	Critical	Not Fixed
PY-CSN-004	Ability to use browse website from the client application webview	Medium	Not Fixed

2.9 Product Strengths

During our assessment, we observed the following properties of the product that are well designed and serve towards its strengths:

- ✓ Implementation of strong enforcing device admin policies and restrictions.
- ✓ Device recovery mode is lock in the bootloader
- ✓ Device does not have an external emmc to prevent tampering of the filesystem

2.10 Product Weaknesses

The vulnerabilities below were identified and verified by Payatu during the process of this security assessment.

- Fails to restrict disabling system applications
- Fails to disable corrupt/non-compatible system calls in engineer mode
- The device is not resistant to Fault injection

Retesting should be planned after the remediation of these vulnerabilities

3. Technical Findings

3.1 PY-CSN-001: Factory Reset bypass by disabling core android application

Potential Impact: CRITICAL

Description:

During our assessment, we observed that the device OS has an option to disable most of the system application. This can be abused to disable the core system functionality of the device and leads to not loading the android and crash into recovery mode.

From recovery mode, device can be factory reset and the client app can be uninstalled or new firmware, without the client app can be installed from fastboot.

Affected Hosts: Device OS FRP Protection

Business Risk: Financial loss, as this will allow the adversary to use the restricted features of device without making complete payment.

Technical Risk: Successful exploitation of this vulnerability allows an attacker to bypass the client application restriction and use it without their control.

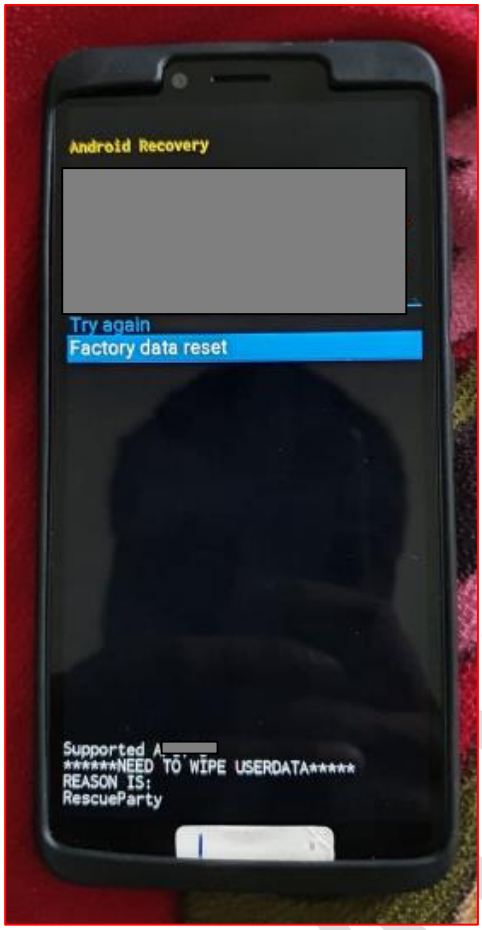
Remediation:

- Enforce Device admin policies to disable letting user disable core system applications
- Restrict the bootloader to crash to recovery if android fails to boot

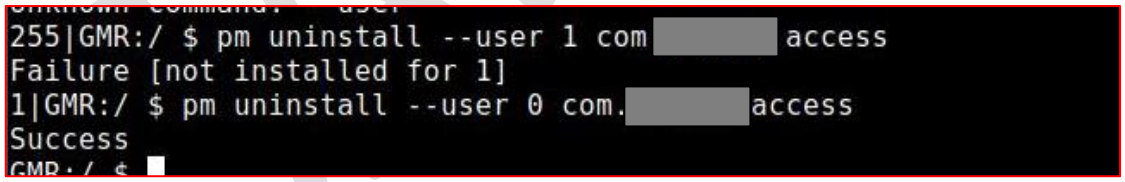
---

Steps to Reproduce:

1. Start your android in the phone
2. Go to Settings –> App Settings
3. In the right corner enable the option “Show System Applications”
4. Now disable all the application one by one.
5. Restart the phone now
6. Below screenshot shows that the android cannot be loaded and it entered recovery mode



7. Now factory reset can be performed and booted to the device.
8. Now once in the android, enable debugger option from about and enable USB debugging.
9. Use “adb shell” to gain access the device’s shell.
10. Use am uninstall –user 0 com.<<client app>>.access to uninstall the app.
11. Device will not download or enable the client application till the device is factory reset.



3.2 PY-CSN-002: Crashing Kernel using engineer mode to expose SPRD Mode

Potential Impact: **CRITICAL**

Description:

During our assessment, we observed that the device OS has an option to go to the engineer mode using <<xxxxxxxxxxxxx>>.

From the engineer mode, few system operations can be performed, there one specific option in ylog which leads to crashing of the AP kernel and puts the device in Gadget Serial Programming Mode.

Affected Hosts: Device OS Firmware Protection

Business Risk: Financial loss, as this will allow the adversary to use the restricted features of device without making complete payment. Device control loss, as the flashing of custom firmware can lead the adversary to unearth other vulerabilities.

Technical Risk: Successful exploitation of this vulnerability allows an attacker to bypass the client application restriction by reflashing a new custom firmware.

Remediation:

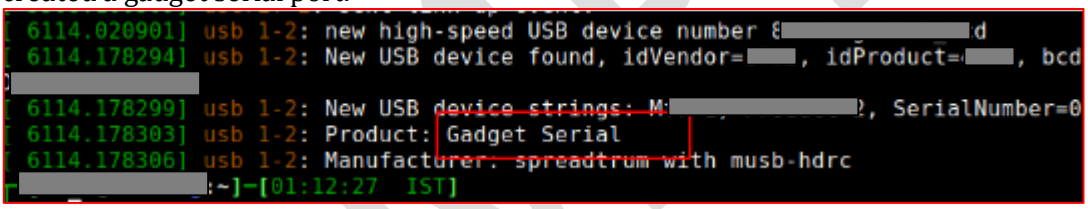
- Enforce bootloader to disable programming mode when kernel or hard fault occurs
- Restrict the sysdump functionality of the bootloader or kernel

---

Steps to Reproduce:

1. Start your android in the phone
2. Go to your dialer and type <<xxxxxxxxxxxxx>> and go to the engineer mode
3. Now go to debug tab of the engineer mode and select Ylog
4. Now select the setting in the right corner of the Ylog Window.
5. Now Select the Log/Debug setting menu.
6. Now Select “Assert CP2 Manually” and press the corresponding number,
7. Now the phone will restart and you will see a kernel crash with a sysdump screen as shown

8. Now connect the USB cable to your device, see your device manager or lsusb to check that it created a gadget serial port.



9. This Serial port is used in SPRD Research/Factory mode to flash it with a new firmware.

3.3 PY-CSN-003: Crashing Kernel using EM Fault Injection to expose SPRD Mode

Potential Impact: **CRITICAL**

Description:

During our assessment, we observe in the main Motherboard that it is using a standard <<xxxx>> RAM module from <<xxxx>>.

This type of RAM is not robust against Fault injection Attacks like EM Fault Injection Attack.

Affected Hosts: Device OS Firmware Protection

Business Risk: Financial loss, as this will allow the adversary to use the restricted features of device without making complete payment.

Technical Risk: Successful exploitation of this vulnerability allows an attacker to bypass the client application restriction by reflashing a new custom firmware.

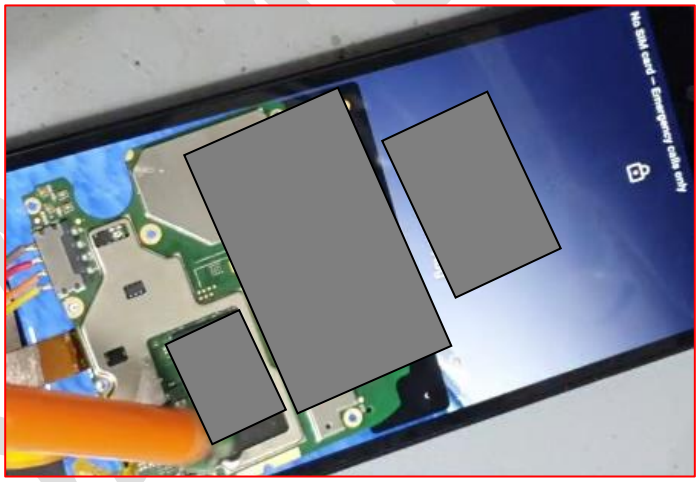
Remediation:

- Enforce bootloader to disable programming mode when kernel or hard fault occurs
- Restrict the sysdump functionality of the bootloader or kernel

---

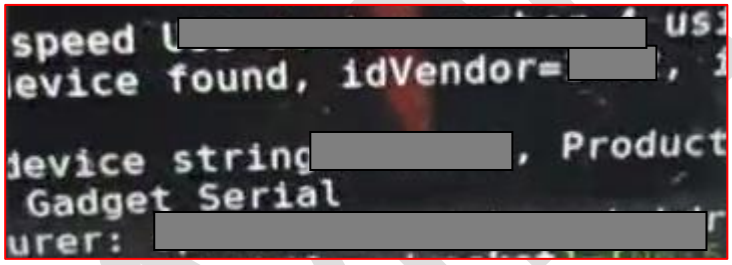
Steps to Reproduce:

1. Open the casing of the device and expose the main motherboard.
2. Remove the thermal/EM Shielding covering the Main SPRD Processor and the <<xxxx>> SRAM.
3. Start your android by turning on your phone.
4. Now inject EM fault using any EM Injector on the SRAM Module when the device is Turned On.



5. You will see flickering in the LCD screen and the device will restart and get into Kernel crash mode and sysdump will be enabled.

6. Now connect the USB cable to your device, see your device manager or lsusb to check that it created a gadget serial port.



7. This Serial port is used in SPRD Research/Factory mode to flash it with a new firmware.
Link for the Video PoC – <<>>

3.4 PY-CSN-004: Ability to use browse website from client application web-view

Potential Impact: Medium

Description:

During our assessment, we observed that the client app has a tab called <<XX>> and <<YY>> which is web-view for the client payment and client account website.

This web-view can be abused even if the client application is locked for payment to browse other website by traversing from external link provided in the client website

---

Affected Hosts: Client application Access Restrictions

Business Risk: Financial loss as this will allow the adversary to use the restricted features of device without making complete payment.

Technical Risk: Successful exploitation of this vulnerability allows an attacker to use browser functionality without accessing the core android

---

Remediation:

- Client application Webview has to be restricted to only client or partner website.
- Implement a restricted webpage to be integrated for this application

---

Steps to Reproduce:

1. Start your android in the phone
2. Once the device is locked because of payment. Client app will block all application access
3. Go to the <<XX>> or <<YY>> section and navigate to the bottom of the page
4. (This test can be performed using other external websites too)
5. Click on the <<XYZ>> and go to <<XYZ>> account of the client.
6. In the <<XYZ>> website, search for google and go to google account or alternatively other account you have access to, so you can enter a URL of the page you want to access
7. Now click on any google link and traverse to the main google search page.
8. Now you can navigate to any website from there easily and access the browsing functionality of the device even when the device is locked for payment.

4. We Prescribe

The below table provides an at-a-glance view of the remediation solution and the best practices that should be taken into consideration to improve the security posture of the scoped-in application:

Vuln ID	Recommendation & Best Practices
PY-CSN-001	• Enforce Device admin policies to disable letting user disable core system applications • Restrict the bootloader to crash to recovery if android fails to boot **Note:** It is paramount to ensure that any backup bootloader to be disabled to not enter any recovery mode
PY-CSN-002	• Enforce bootloader to disable programming mode when kernel or hard fault occurs • Restrict the sysdump functionality of the bootloader or kernel **Note:** It is paramount to ensure that any backup bootloader to be disabled to not enter any recovery mode
PY-CSN-003	• Enforce bootloader to disable programming mode when kernel or hard fault occurs • Restrict the sysdump functionality of the bootloader or kernel **Note:** It is paramount to ensure that any backup bootloader to be disabled to not enter any recovery mode
PY-CSN-004	• Client Web-view has to be restricted to only client or partner website. • Implement a restricted webpage to be integrated for this application

Looking at the types and severity of vulnerabilities found, we would recommend:

- The Disabling any recovery feature of the device in the bootloader to put the device in SPRD Mode

- Proper restriction in the UI/UX of the client application installed.

5. Appendix

5.1 Observations

In the observation section, we document any specific issue that we have observed, which, while does not directly lead to any vulnerability, maybe something that needs to be looked at from the perspective of a functional bug or configuration issue.

We have not observed anything unusual in this project from functionality or configuration prospective.

5.2 References

https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-811.pdf

https://www.nccoe.nist.gov/publication/1800-4/VolB/

5.3 Failed Test Cases

In the failed test case section, we document some of the application strengths that we have observed during our testing. This serves as a view of the best practices that are being followed by the client.

5.3.1 Enabling another Device Admin

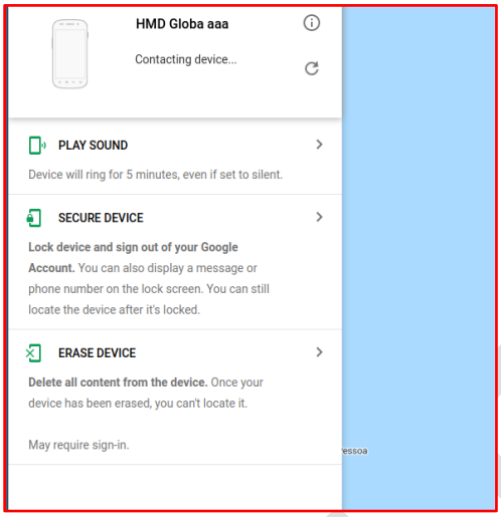
Description

Client application does not allow users to add other application to be a device admin policy, this can be abused to perform factory reset bypassing the client application device admin policies. Google's Find my phone is one example which lets user to perform a factory reset remotely

Testing Process

1. Run the Google service and install "find my phone app"
2. Login to your Google account and you will see the device name
3. Perform the Factory erase setting from the web.

4. This didn't succeed, because the client application doesn't let other apps give the factory reset permission to erase the device



5.3.2 Access to recovery mode using hardware key

Description

Most android phone has a physical key combination used to enter the recovery mode from bootloader, this is part of the bootloader, and this lets you perform factory reset and clear the client application or install custom rom.

Testing Process

1. Turn off your phone completely.
2. Press the volume up + power button.
3. It will vibrate and enter recovery mode.
4. This failed because bootloader locked this hardware combination to enter the device's recovery mode.

5.3.3 Modifying the EMMC of the device

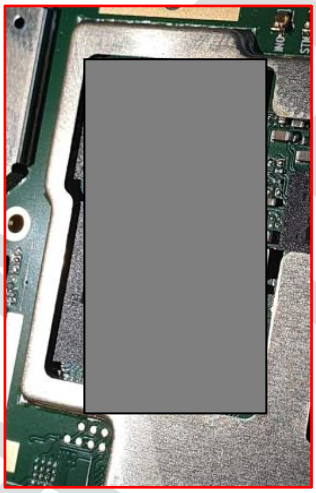
Description

Most Android device, in hardware they have the main processor, external SRAM and ROM module, the firmware resides on the ROM like Emmc or NAND Flash, these modules can be desoldered and firmware can be extracted and modified to bypass any frp or firmware locks

---

Testing Process

1. Disassemble the board and identify the external Emmc
2. Now map the pinout of the emmc or NAND flash
3. Connect it to any emmc reader or easyjtag in our case
4. Use the provided tool to extract a flash the modified firmware
5. This didn't succeed, because internal storage is part of SoC. No external storage is found.



5.3.4 Termux shell to remove the client application

Description

Even if the adb is disabled, specific application like Termux or ssh server, gives a restricted shell access, which user can use to interact with the device. Tools like am and pm can be used to disable or uninstall the application

---

Testing Process

1. Install Termux from the Google play store
2. Open Termux and install openSSH and start the SSH Server
3. Now you can connect to the SSH server from your machine and send am/pm commands to call an intent or uninstall an app
4. This was unsuccessful because, shell access to these commands requires elevated privilege which is restricted by the device admin

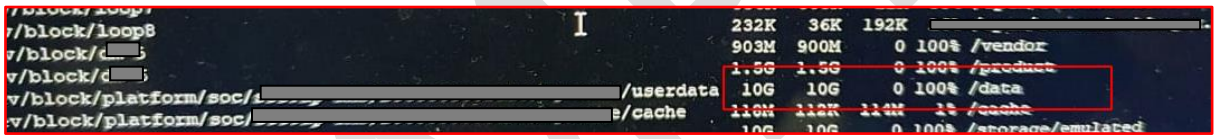
5.3.5 Storage full attack

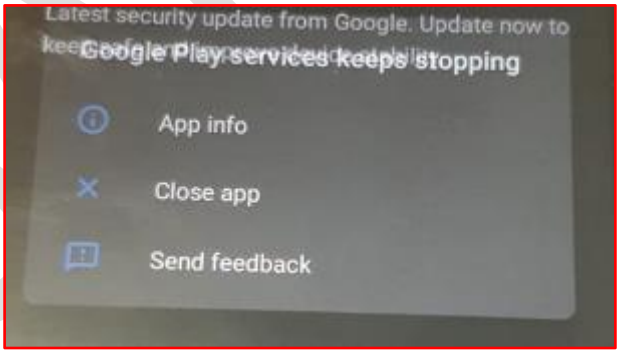
Description

Android OS stops booting if the user-data partition is completely full, you can use shell access and dd command to create large chunk of data to completely fill the data and let the android core not have space to boot and crash into recovery.

Testing Process

1. Access the shell from the Termux
2. Use “dd if=/dev/zero of=/sdcard/1 bs=1G “to create 1GB block files.
3. Repeat it till it has 0k left. Reboot the system
4. This attack was able to crash some android component but not the core ones because of the storage clear manager app which comes as part of the Android file manager. It cannot be disabled as it is system ap





5.3.6 Enabling adb from restricted shell

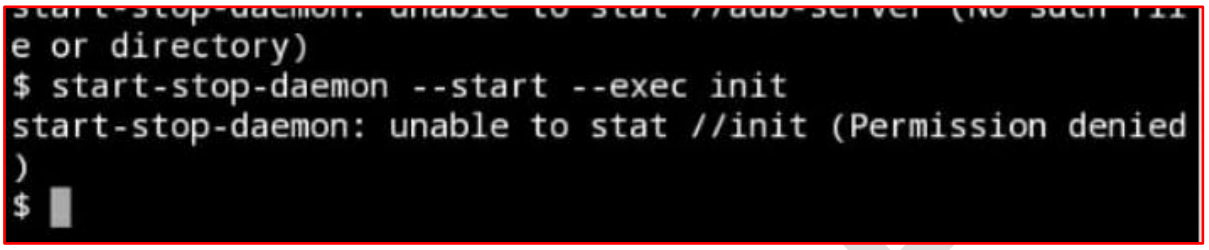
Description

From a restricted shell, adbd server can be started over tcpip this can help in getting “Shell” privilege to gain access to the recovery mode or fastboot mode and also lets you

uninstall system application

Testing Process

1. From the Termux shell, run “start-stop-service” to start the service
2. Now you can use adbd to start the adb service
3. This operation is unsuccessful because of the restricted privilege shell which does not let you start any service.



5.3.7 Using One Click Root

Description

There are few 3rd part apps which lets you root the device, without the need to tamper bootloader. Some apps like kingo root, OneClick Root and other public exploits

---

Testing Process

1. Find the apk directly from their respective website
2. Install the apk in the phone, Google play protect will warn you about it. You can agree to install
3. Now perform app-based rooting
4. This failed as the security patch is updated and rooting is not possible from the user-space application