Elghouate Zouhair

```
C:\Users\amjad>adb -s emulator-5556 root
restarting adbd as root

C:\Users\amjad>adb remount
Successfully disabled verity
virtual bool android::fiemap::ImageManagerBinder::MapImageDevice(const std::string &, const std::chrono::milliseconds &, std::string *) binder returned: Fai
led to map
[libfs_mgr] could not map scratch image
Failed to allocate scratch on /data, fallback to use free space on super
Using overlayfs for /system
Using overlayfs for /vendor
Using overlayfs for /product
Using overlayfs for /system_dlkm
Using overlayfs for /system_ext
Verity disabled; overlayfs enabled.
Now reboot your device for settings to take effect

C:\Users\amjad>adb -s emulator-5556 id
adb.exe: unknown command id

C:\Users\amjad>adb -s emulator-5556 shell id
uid=0(root) gid=0(root) groups=0(root),1004(input),1007(log),1011(adb),1015(sdcard_rw),1028(sdcard_r),1078(ext_data_rw),1079(ext_obb_rw),3001(net_bt_admin),
3002(net_bt),3003(inet),3006(net_bw_stats),3009(readproc),3011(uhid),3012(readtracefs) context=u:r:su:s0

C:\Users\amjad>adb -s emulator-5556 shell getprop ro.boot.verifiedbootstate
orange

C:\Users\amjad>adb -s emulator-5556 shell getprop ro.boot.veritymode
enforcing

C:\Users\amjad>adb -s emulator-5556 shell getprop ro.boot.vbmeta.device_state


C:\Users\amjad>adb -s emulator-5556 shell "su 0 id"
uid=0(root) gid=0(root) groups=0(root),1004(input),1007(log),1011(adb),1015(sdcard_rw),1028(sdcard_r),1078(ext_data_rw),1079(ext_obb_rw),3001(net_bt_admin),
3002(net_bt),3003(inet),3006(net_bw_stats),3009(readproc),3011(uhid),3012(readtracefs) context=u:r:su:s0

C:\Users\amjad>adb -s emulator-5556 disable-verity
Successfully disabled verity
enabling overlayfs
Reboot the device for new settings to take effect
```

```
C:\Users\amjad>adb -s emulator-5556 reboot

C:\Users\amjad>adb -s emulator-5556 remount
Not running as root. Try "adb root" first.

C:\Users\amjad>adb -s emulator-5556 root
adb: unable to connect for root: closed

C:\Users\amjad>adb root
adb: unable to connect for root: more than one device/emulator

C:\Users\amjad>adb -s emulator-5556 remount
Successfully disabled verity
Remounted /system as RW
Remounted /vendor as RW
Remounted /product as RW
Remounted /system_dlkm as RW
Remounted /system_ext as RW
Remount succeeded

C:\Users\amjad>adb -s emulator-5556 logcat -d | tail -n 200 > logcat_root_check.txt
'tail' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\amjad>adb -s emulator-5556 logcat -d | Select-Object -Last 200 > logcat_root_check.txt
'Select-Object' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\amjad>adb -s emulator-5556 shell "logcat -d | -n 200" > logcat_root_check.txt
/system/bin/sh: -n: inaccessible or not found

C:\Users\amjad>adb -s emulator-5556 shell "logcat -d | tail -n 200" > logcat_root_check.txt
```

exécute une série de commandes ADB sur un émulateur Android (emulator-5556) pour obtenir les privilèges root, désactiver la vérification de partition (verity) et remonter les partitions système en lecture-écriture. La première séquence montre un root réussi, la désactivation de verity avec l'activation d'overlays, et la confirmation du statut de démarrage "orange". Après un redémarrage, une nouvelle tentative de remontage échoue car le démon ADB n'est plus root, puis réussit après avoir rétabli la connexion root. Finalement, l'utilisateur tente de capturer les 200 dernières lignes du logcat, en corrigeant les erreurs de syntaxe liées à l'absence de commandes Windows, pour aboutir à une commande valide exécutée dans le shell Android.

```
amjad@DESKTOP-NP166KJ MINGW64 ~/diva-apk-file (main)
$ adb -s emulator-5556 install --bypass-low-target-sdk-block DivaApplication.apk
Performing Streamed Install
Success
```

L’installation de l’application de test.

Android Security

1. **Sandboxing** : Isole chaque application dans un espace sécurisé (utilisateur Linux unique), interdisant toute intrusion ou espionnage d'une application sur une autre.
2. **Modèle de permissions** : Soumet l'accès aux ressources sensibles (caméra, contacts, GPS) à l'autorisation explicite de l'utilisateur pour un contrôle total des données.
3. **Intégrité système** : Utilise le *Verified Boot* pour certifier que le logiciel n'a pas été altéré, garantissant un démarrage sain et protégé contre les modifications malveillantes.

Verified Boot (idée générale + check AVD)

1. Objectif principal du Verified Boot

Assurer que le système d'exploitation lancé est l'original certifié par le fabricant, exempt de toute altération.

- **Analogie :** C'est un gardien qui vérifie que les scellés de votre maison sont intacts avant de vous laisser entrer.

2. La "Chain of Trust" (Chaîne de confiance)

Mécanisme où chaque composant logiciel valide l'authenticité du suivant avant de lui donner le contrôle.

- **Analogie :** Un relais de gardiens où chacun vérifie l'identité du suivant avant de lui confier les clés.

3. Importance critique de l'intégrité au démarrage

Le démarrage est le socle de la sécurité ; s'il est corrompu, toutes les barrières de protection suivantes perdent leur efficacité.

- **Analogie :** Si la porte principale d'une forteresse est forcée, la solidité des coffres à l'intérieur n'a plus d'importance.

4. Analyse comparative sur AVD

- **AVD Standard (Non-rooté) :** ro.boot.verifiedbootstate = green
    - **Diagnostic :** Système intègre et certifié conforme.
- **AVD Rooté :** ro.boot.verifiedbootstate = orange
    - **Diagnostic :** Système modifié (bootloader déverrouillé), l'intégrité n'est plus garantie par le constructeur.

5. Signification des états (Codes couleurs)

```
C:\Users\amjad>adb -s emulator-5556 shell getprop ro.boot.verifiedbootstate
orange
```

- **Vert :** Intégrité totale, signature valide.
- **Jaune / Orange :** Avertissement, le système a été modifié ou le bootloader est ouvert.
- **Rouge :** Danger critique, l'intégrité est rompue ou le système est corrompu.
-

Conclusion : Le passage de l'état **GREEN** à **ORANGE** lors du rooting apporte la preuve technique que la chaîne de confiance est rompue. Le système perd sa garantie d'inviolabilité dès que ses fichiers racines sont modifiés.

AVB (Android Verified Boot)

L'AVB modernise la vérification d'intégrité en contrôlant chaque partition système via des signatures cryptographiques robustes.
Il intègre une protection contre le **rollback**, empêchant tout retour forcé vers d'anciennes versions du système porteuses de failles.
Ce dispositif garantit ainsi que l'appareil démarre toujours sur un socle logiciel authentique, sécurisé et à jour.

Définir le rooting

1.1 Définition et Contexte

L'obtention des privilèges **Root** correspond à l'acquisition de l'identifiant utilisateur **UID 0** (Super-utilisateur). Dans le cadre de ce laboratoire sur l'application DIVA, le root est indispensable pour briser le "bac à sable" (Sandboxing) d'Android qui isole normalement chaque application.

1.2 Impact sur la Sécurité du Système

- **Abolition des barrières :** Le passage en mode Root désactive les restrictions d'accès aux répertoires protégés (ex: /data/data/).
- **Modification de la confiance :** Le système de fichiers n'est plus considéré comme "intègre". L'intégrité du démarrage (Verified Boot) est compromise ou désactivée.
- **Exposition des données :** Une application vulnérable (comme DIVA) exécutée sur un système rooté permet à un attaquant (ou à un testeur) de lire des secrets normalement chiffrés ou cachés par le système d'exploitation.

1.3 Intérêt en Laboratoire de Sécurité

Le mode Root nous permet d'observer en temps réel :

1. **Le stockage local :** Accès aux fichiers .xml (Shared Preferences) et aux bases de données .db (SQLite).
2. **L'activité système :** Capture de logs détaillés via logcat sans filtrage de confidentialité.
3. **L'analyse dynamique :** Utilisation d'outils comme Frida ou Xposed pour intercepter les fonctions de l'application.

1.4 Gestion des Risques et Bonnes Pratiques

En raison des risques d'instabilité et d'exposition, les mesures suivantes ont été appliquées :

- **Isolation :** Utilisation exclusive d'un émulateur (AVD) déconnecté de tout compte personnel (Google, etc.).
- **Traçabilité :** Journalisation systématique des actions via l'exportation des logs (logcat_root_check.txt).
- **Reset (Réinitialisation) :** Utilisation de snapshots d'émulateur pour revenir à un état sain après chaque test, garantissant qu'aucune donnée malveillante ou modification persistante ne reste dans l'environnement.

Preuve technique

```
C:\Users\amjad>adb -s emulator-5556 shell id
uid=0(root) gid=0(root) groups=0(root),1004(input),1007(log),1011(adb),1015(sdcard_rw),1028(sdcard_r),1078(ext_data_rw),1079(ext_obb_rw),3001(net_bt_admin),
3002(net_bt),3003(inet),3006(net_bw_stats),3009(readproc),3011(uhid),3012(readtracefs) context=u:r:su:s0
```

Command fait avec succes