|  |  |
| --- | --- |
| **Name: Vikram Upadhya** | **Employee Code: 51381895** |
| **DU Name: ERS-PTS-MAA** | **L4: Saravanan Sugavanam** |
| **L3: Krishnamoorthy** | **L2:Sanyam** |
| **Domain: Medical V&V** | **Service Line:** |
| **Business Impact to Customer: Yes / No** | **Account Name : GE Healthcare** |
|  | |
| **Idea Heading:**  **Non Invasive/Tamper Proof Test automation for Android Devices.**  **Validation and Certification of Medical devices for the Client Hardware, Firmware and Application by mandated Federal Drug Administration needs the Device Under Test (DUT) and the Application Under Test (AUT) to be tamperproof. The Application shall need to be as close as possible to the Production release without any changes/Tamper. Non-Invasive solutions are to add a camera or a separate APK/Java Application to take screenshots from the screen.**   |  |  |  | | --- | --- | --- | | **Binder Usage** | **Advantages** | **Disadvantages** | | **No change in APK, additional method needed to take Screenshots** | **No need for Rooting** |  | | **Security Permissions and Roles** | **No need to change UID or root the Phone.**  **Secure access since only valid users know the Services list**  Context.checkPermission(permission, pid, uid) | **None** | | **Scripting Support to ensure that Files exist outside Android** | **No need to modify Production Aplication/Devices** | **None** | | **Limitations of Binder** |  | **Memory and Performance Limitations\* see caveat** |   **Caveats:**  Binders are not ideal for transferring large data streams (like audio/video) since every object has to be converted to (and back from) a Parcel  **The Client APK (AUT) and DUT are not available for testing outside their premises. A POC can be made for a similar application/APK and results compared for an effective solution.** | |
| **Idea Details:**  **The Value Generation is to have the AUT APK which has rights to take screenshots exposed as a parcelable interface and declared to Binder.**  **Interface to take screenshot needs to be declared in a .AIDL file and is exposed. The Binder interface allows for a command line trigger to get the Screenshot.**  **Binder command line arguments can be passed to take screenshots and transfer them over USB/File System**  adb shell service list  **Current Process:**  **Custom solutions exist that allow for the DUT to be monitored invasively, however the AUT cannot be modified as per the guidelines of the test procedures. Certain application and storage locations inside the OS (Android in this case) needs root permissions or rooting which modify the application behavior and compromise security.**  **Some approaches have been made to capture Text/Screen from the GPU controlled OpenGL Data using Java Custom programs but these are currently invasive.**  **Constraints:**  **The Challenge faced are 2 Fold:**  **Non Invasive Data Capture of the Screen**  **Limitations in Performance and Exact match scenario**  **Using Adb shell commands we can trigger the method to take a screenshot when needed and export the same without enabling Permissions or changing the root access of the device.**  **References:**   * [**http://www.dre.vanderbilt.edu/~schmidt/cs282/PDFs/android-binder-ipc.pdf**](http://www.dre.vanderbilt.edu/~schmidt/cs282/PDFs/android-binder-ipc.pdf)      * [**http://developer.android.com/guide/components/bound-services.html#Binder**](http://developer.android.com/guide/components/bound-services.html#Binder) * [**http://www.truiton.com/2014/11/bound-service-example-android/**](http://www.truiton.com/2014/11/bound-service-example-android/) * [**https://code.google.com/p/android-screenshot-library/wiki/DeveloperGuide**](https://code.google.com/p/android-screenshot-library/wiki/DeveloperGuide) | |