|  |  |  |
| --- | --- | --- |
| ScannerController | | |
| **Name** | ScannerController | |
| **Purpose** | To scan all side-loaded applications | |
| **Attributes** | | |
| **Name** | **Type** | **Description** |
| ApplicationStorage | Arraylist | Store an arraylist of application to be monitored |
|  |  |  |
|  |  |  |
| **Operations / Methods()** | | |
| **Name** | **Return Type** | **Description** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Scanner Boundary | | |
| **Name** | Scanner Boundary | |
| **Purpose** | For user to interact with the scan button | |
| **Attributes** | | |
| **Name** | **Type** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
| **Operations / Methods()** | | |
| **Name** | **Return Type** | **Description** |
| displayMalwareScan() | void | Display malware scan results |
| displaySideloadedApp() | void | Display all side loaded apps |
| displaySideloadedCertificates() | void | Display the status of the certificate (whether it is valid or not) |

|  |  |  |
| --- | --- | --- |
| Certificate Scanner | | |
| **Name** | Certificate Scanner | |
| **Purpose** |  | |
| **Attributes** | | |
| **Name** | **Type** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
| **Operations / Methods()** | | |
| **Name** | **Return Type** | **Description** |
| scanValidCertificate() | Boolean | Check validity of certificate, current date is check against expiry date |
| scanDirectlyInstalled() | Boolean | Check issuer specified on signing app certificate to see whether it uses Google Play App Signing. Only apps that are not signed with Google Play App signing will be scanned. |

|  |  |  |
| --- | --- | --- |
| Malware Scanner | | |
| **Name** | Malware Scanner | |
| **Purpose** |  | |
| **Attributes** | | |
| **Name** | **Type** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
| **Operations / Methods()** | | |
| **Name** | **Return Type** | **Description** |
| isInfected(File) | Boolean | Check if a file is infected with malware |
| detectSystemImg() | Boolean | Identifies any changes to the phone's system.img. Changes made means that new binaries or applications are being installed on the phone which may indicate that the malicious applications are establishing persistence. |
| detectRecoveryChanges() | Boolean | Identifies changes to phone's recovery.img, which holds the running kernel & root file system when the phone is booted into recovery mode. Changes to the recovery.img would means that the user will be installing a custom recovery.img that may have elevated privileges to areas of the phone storage that are typically protected while the phone is running in normal mode. |
| detectBootImg() | Boolean | Identifies any changes to the phone's boot.img. It then performs further analysis to determine whether the change occured in the phone's kernel or ramdisk, which contains the phone's root file system and core system files. Core startup files like init.rc can be modified , and a malicious application could add executable files to the root system that could survive a reboot. |
| detectBootloaderChanges() | Boolean | Detector identifies any changes to the phone's bootloader. In the enterprise context, observed unplanned changes to the bootloader should rarely happen. Activity that changes this would normally be deemed as malicious. |
| detectInspectAPK() | Boolean | Observes Android packages (APKs) that have been installed since the last snapshot by first running fiwalk to identify the APK file's installation directory and inode, then using icat to extract the APK file. Once the APK file has been extracted, the resulting AndroidManifest.xml file is parsed and then inspected for the presence of a BOOT\_COMPLETED even registration. If BOOT\_COMPLETED is not found, the detector will simply log the installation of a new application. This will serve as a good indicator for identifying malicious applications, |
| checkLogFileMace() | Boolean | Logger executes the forensic tool fiwalk to output DFXML, which is processed to identify any files that were added or changed on the file systems. The resulting log output is time stamped based upon the MACE times and stored in the database. |
| checkLogFLSParse() | Boolean | Logger executes the forensic tool fls from the Sleuth Kit and then processes the results to output any observed deleted files. The resulting log output is time stamped based upon the MACE times and is stored in the database. If any seemingly legitimate application contains exploit files in its assets directory and subsequently writes them to the file system, executes them, and deletes them. This logger will log the existence of any remnants of the deleted exploit files on the file system. |

|  |  |  |
| --- | --- | --- |
| PermissionScanner | | |
| **Name** | PermissionScanner | |
| **Purpose** |  | |
| **Attributes** | | |
| **Name** | **Type** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
| **Operations / Methods()** | | |
| **Name** | **Return Type** | **Description** |
| listPermission() | Void | Display the permission settings required by the application. |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| MonitorBoundary | | |
| **Name** | MonitorBoundary | |
| **Purpose** |  | |
| **Attributes** | | |
| **Name** | **Type** | **Description** |
| displayAppStats(Name: String) | void | Display usages like (RAM,CPU,Storage space) |
| displayAppTraffic(Name: String) | void | Display incoming traffic and outgoing traffic from this |
| displayEnvironmentSettings() | Void | Display OS version and tells user whether it is up to date |
| **Operations / Methods()** | | |
| **Name** | **Return Type** | **Description** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| MonitorController | | |
| **Name** | MonitorController | |
| **Purpose** |  | |
| **Attributes** | | |
| **Name** | **Type** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
| **Operations / Methods()** | | |
| **Name** | **Return Type** | **Description** |
| getCpuUsage(AppID : String) : float | float | Retrieve cpu usage by a particular app |
| getRamUsage(AppID : String) | float | Retrieve ram usage by a particular app |
| getStorageUsage(AppID : String) | float | Retrieve storage usage by a particular app |
| getCurrentOSVersion() | String | Retrieve the current OS version |
| getLatestOSVersion() | String | Retrieve the latest version of the OS |
| compareOSVersion() | String | Compare current OS version with latest OS version |

|  |  |  |
| --- | --- | --- |
| NetworkTraffic | | |
| **Name** | NetworkTraffic | |
| **Purpose** | To store incoming and outgoing network traffic information | |
| **Attributes** | | |
| **Name** | **Type** | **Description** |
| ipAddress | String |  |
| incomingTraffic: double | double | Size of incoming data |
| outgoingTraffic | double | Size of outgoing data |
| **Operations / Methods()** | | |
| **Name** | **Return Type** | **Description** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | | |
| **Name** |  | |
| **Purpose** |  | |
| **Attributes** | | |
| **Name** | **Type** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
| **Operations / Methods()** | | |
| **Name** | **Return Type** | **Description** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | | |
| **Name** |  | |
| **Purpose** |  | |
| **Attributes** | | |
| **Name** | **Type** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
| **Operations / Methods()** | | |
| **Name** | **Return Type** | **Description** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | | |
| **Name** |  | |
| **Purpose** |  | |
| **Attributes** | | |
| **Name** | **Type** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
| **Operations / Methods()** | | |
| **Name** | **Return Type** | **Description** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | | |
| **Name** |  | |
| **Purpose** |  | |
| **Attributes** | | |
| **Name** | **Type** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
| **Operations / Methods()** | | |
| **Name** | **Return Type** | **Description** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | | |
| **Name** |  | |
| **Purpose** |  | |
| **Attributes** | | |
| **Name** | **Type** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
| **Operations / Methods()** | | |
| **Name** | **Return Type** | **Description** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | | |
| **Name** |  | |
| **Purpose** |  | |
| **Attributes** | | |
| **Name** | **Type** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
| **Operations / Methods()** | | |
| **Name** | **Return Type** | **Description** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | | |
| **Name** |  | |
| **Purpose** |  | |
| **Attributes** | | |
| **Name** | **Type** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
| **Operations / Methods()** | | |
| **Name** | **Return Type** | **Description** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | | |
| **Name** |  | |
| **Purpose** |  | |
| **Attributes** | | |
| **Name** | **Type** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
| **Operations / Methods()** | | |
| **Name** | **Return Type** | **Description** |
|  |  |  |
|  |  |  |