Detection of Social Media Exploitation via 
SMS and Camera

https://doi.org/10.3991/ijim.v13i04.10521

Mohamad Adib Azhar(✉), Madihah Mohd Saudi, Azuan Ahmad, Azreena Abu Bakar
Universiti Sains Islam Malaysia (USIM), Nilai, Malaysia
jbalika@outlook.com

Abstract—Internet users all over the world are highly exposed to social media exploitation, where they are vulnerable to be targeted by this cyber-attack. Furthermore, excessive use of social media leads to Internet Addiction Disorder (IAD). Fortunately, social media exploitation and IAD can be monitored and controlled closely based on user’s mobile phone surveillance features which are camera, SMS, audio, geolocation (GPS) and call log. Hence to overcome these challenges, this paper presents five (5) Application Programming Interfaces (APIs) and four (4) permissions for SMS and camera that are mostly and widely used with the social media applications. These 9 APIs and permissions matched with 2.7% of the APIs and permissions training dataset that are related with SMS and camera. This experiment was conducted by using hybrid analysis, which inclusive of static analysis and dynamic analysis, with 1926 training dataset from Brunswick. These 9 APIs and permissions, if being misused by the attacker, could lead to privacy concerns of a mobile device. The finding from this paper can be used as a guidance and reference for the formation of new mobile malware detection technique and modeling in future.

Keywords—Social media exploitation, API, permission, SMS, camera, mobile malwares, mobile phone surveillance feature.

1 Introduction

Human emotion or desire to browse social media via mobile phone to get latest information, communicate with friends and play game, is currently becoming a trend. Unfortunately, excessive use of social media could lead to Internet Addiction Disorder (IAD) and depression. Recently, World Health Organisations (WHO) has declared gaming as one of the International Classification of Diseases (ICD-11) in year 2018. Hence, it is not impossible in future that social media addiction will be categorized as mental disorder due to its implications and impacts to serious depression and lifestyle.

In a smartphone, 5 main surveillance features which are: SMS, camera, call log, geolocation (GPS) and audio could be exploited by the attacker. They can monitor user’s movement and steal confidential information via these surveillance features. In earlier day, Short Message System (SMS) is one of the main mechanisms used by many users for communication. Until now, SMS is still being used for communication
and authentication of online banking. Apart from SMS, camera becomes as an important element in smartphone selection due to our current lifestyle. Picture can be easily disseminated to social media just in a second. Different platforms such as iOS and Android have been implemented in different smartphones and Android has been ranked as the mostly used worldwide. As a result, it is most targeted by the attackers and malwares due to its open-source distribution [1]. Malware is defined as a software that could infect devices without the owner’s consent for malicious intention and it can be categorised as virus, worm, Trojan Horse, adware, spyware, botnet or ransomware. So far, mobile botnet posed the most serious impact to the smartphone users. For an example, in August 2017, WireX botnet spreads among users from 100 countries and it has infected advertising software and launched the DDoS attacks. It hides under system processes and has been taken down from Playstore with the help from Akamai, Flashpoint and Oracle Dyn [2]. In an Android smartphone, every application has limited capability to use smartphone resources and it needs to request permission and Application Programming Interface (API) to perform any task. For an example, once a mobile application (app) is being installed, the mobile app will request a permission to use SMS and camera during first execution or during installation. Once user granted this permission, the app has the authority to send related information and request via SMS and camera.

Features such as API and permission are seen as an opportunity for exploitation [3]. Existing works by [4-12] showed the significant of API and permission usage for exploitation and malwares detection. These works used different analysis techniques such as static analysis, dynamic analysis or hybrid analysis. As for work from [12], MalDozer is proposed to detect the malwares in different of IoT devices, with API as the input. Even in 2018, works by [13-17] also applied the API and Permission in their work. The summarization of work in year 2018 can be referred in Table 1. Nonetheless, none of these works focus on social media app exploitation.

Table 1. Summarisation of related existing works

| Author | Feature | Description | Challenges |
|--------|---------|-------------|------------|
| [12]   | API     | This paper presents about malware classification. | Performance issue related with dataset. |
| [13]   | API and permission | This paper presents how can defend against poisoning attacks from malwares efficiently. | Improvement needed for feature selection and classifier. |
| [14]   | API and permission | This paper presents malware detection based on accuracy, recall and F-measure. | Performance issue related with the feature selection of the permission list. |
| [15]   | API and permission | This paper presents model based on computational processes. | Improvement for limitation of malware classification based on binary format. |
| [16]   | API     | This paper presents malware detection for anti-virus scanners evasion. | Performance issue related with training dataset. |

Though each of the existing works has it owns strength, but still lack of discussion on social media exploitation via API and permission. There are five (5) social media
applications (apps) have been selected for the experiment of our paper. These social media apps are chosen due to the significant impact to the Internet user lifestyle and privacy concerns. Therefore, this paper aims to identify API and permission that are possible to be used for exploitation specifically through SMS and camera.

This paper is organized as follows: Section 2 presents the methodology used in this research, while Section 3 describes the experiments findings carried out in this research and Section 4 includes the summary of the research work.

2 Methodology

The following Fig.1 is the illustration of the lab setup for the experiment conducted and Table II displays the software used. Prior matching step of the extracted API and permission, 1926 of dataset from Brunswick have been downloaded for training purpose [18]. 328 of APIs and permissions for mobile botnet have been reverse engineered by using hybrid analysis and being compared with the APIs and permissions extracted from the social media apps. Hybrid analysis is the combination of the static analysis and dynamic analysis. For this experiment the hybrid analysis is being used to ensure the full extraction from the apps are successfully retrieved. Only 1500 dataset from 1926 training dataset are fully functioning for the analysis. As for the testing, 5 social media apps have been selected where their names are being sanitized and displayed as anonymous in this paper to avoid any conflict of interest. These social media apps are among the top 5 in the world with highest usage.

![Fig. 1. Lab setup](image-url)
Table 2. Software Function

| Software/Hardware       | Function                                      |
|-------------------------|-----------------------------------------------|
| Genymotion              | It is used as the Android emulator.           |
| Show Java Application/| APKtool It is used to decompile APK resource file and extract Permission. |
| Java Decompiler        | It is used to extract API.                   |

Fig. 2. Overall research processes

Fig. 2 represents the summarization of the whole steps during the experiment. These 5 social media apps of APIs and permissions are being reverse engineered and analyzed, and compared with the existing extracted of 328 mobile botnets APIs and permissions. This is important to classify each of the API and permission as normal or as dataset with an opportunity for malicious exploitation (refer Fig. 3). While Fig. 4, shows an example of permission extraction for the social media app.

Fig. 3. Comparison between Permissions and APIs with mobile botnet features
3 Findings

The following are the findings of API and permission classification for SMS and camera for mobile botnet from the training dataset and possible exploitation of API and permission in social media apps.

The nominal data in Table III to Table VIII represents the feature representative in symbol. Table III depicts 190 APIs extraction names from the training dataset and Table IV displays 14 APIs that are related with SMS and camera.

| Nominal data | API |
|--------------|-----|
| **AP1 – AP12 : path: android/accounts/AccountManager** | addAccount, addAccountExplicitly, blockingGetAuthToken, getAccounts, getAuthToken, getPassword, invalidateAuthToken, peekAuthToken, removeAccount, setAuthToken, setPassword |
| **API13 – API18 : path: android/app/Activity** | |
Detection of Social Media Exploitation via SMS and Camera

| AP13  | sendBroadcast |
| AP14  | setContentView |
| AP15  | setPersistent |
| AP16  | startActivity |
| AP17  | startActivityForResult |
| AP18  | startActivityIfNeeded |

**APIs**

- `sendBroadcast`
- `setContentView`
- `setPersistent`
- `startActivity`
- `startActivityForResult`
- `startActivityIfNeeded`
- `getRecentTasks`
- `getRunningTasks`
- `killBackgroundProcesses`
- `restartPackage`

**Android APIs**

1. **Path:** `android/app/ActivityManager`
   - `getRecentTasks`
   - `getRunningTasks`
   - `killBackgroundProcesses`
   - `restartPackage`

2. **Path:** `android/app/Activity`
   - `reportFailedPasswordAttempt`
   - `reportSuccessfulPasswordAttempt`
   - `setActivePasswordState`
   - `setTimeZone`
   - `backup/BackupManager.dataChanged`
   - `Instrumentation.sendKeyDownUpSync`
   - `KeyguardManager$KeyguardLock.disableKeyguard`
   - `KeyguardManager$KeyguardLock.reenableKeyguard`
   - `KeyguardManager.exitKeyguardSecurely`
   - `NotificationManager.notify`
   - `Service.sendBroadcast`
   - `Service.startActivity`
   - `StatusBarManager.expand`
   - `WallpaperManager.setBitmap`
   - `WallpaperManager.setResource`
   - `WallpaperManager.suggestDesiredDimensions`

3. **Path:** `android/appwidget/AppWidgetManager`
   - `bindAppWidgetId`

4. **Path:** `android/bluetooth/BluetoothAdapter`
   - `cancelDiscovery`
   - `disable`
   - `enable`
   - `getAddress`
   - `getBondedDevices`
   - `getState`
   - `isDiscovering`
   - `isEnabled`
   - `listenUsingRfcommWithServiceRecord`
   - `startDiscovery`
   - `createRfcommSocketToServiceRecord`
   - `getBondState`
| AP52-AP54 | path: android/bluetooth/ |
|-----------|--------------------------|
| AP52      | BluetoothDevice;->getName |
| AP53      | BluetoothHeadset;->getBatteryUsageHint |
| AP54      | BluetoothSocket;->connect |
| AP55-AP65 | path: android/content/ContentResolver |
| AP55      | addPeriodicSync |
| AP56      | getMasterSyncAutomatically |
| AP57      | getSyncAutomatically |
| AP58      | openFileDescriptor |
| AP59      | openInputStream |
| AP60      | openOutputStream |
| AP61      | query |
| AP62      | removePeriodicSync |
| AP63      | setIsSyncable |
| AP64      | setMasterSyncAutomatically |
| AP65      | setSyncAutomatically |
| AP66-AP74 | path: android/content/Context |
| AP66      | sendBroadcast |
| AP67      | sendOrderedBroadcast |
| AP68      | sendStickyBroadcast |
| AP69      | setWallpaper |
| AP70      | startActivity |
| AP71      | startService |
| AP72      | ContextWrapper;->sendBroadcast |
| AP73      | ContextWrapper;->setWallpaper |
| AP74      | ContextWrapper;->startActivity |
| AP75-AP77 | path: android/content/pm |
| AP75      | PackageManager;->addPreferredActivity |
| AP76      | PackageManager;->clearPackagePreferredActivities |
| AP77      | PackageManager;->setComponentEnabledSetting |
| AP78      | path: android/ |
| AP78      | hardware/Camera;->open |
| AP79-AP88 | path: android/location/LocationManager |
| AP79      | addGpsStatusListener |
| AP80      | addNmea_listener |
| AP81      | getBestProvider |
| AP82      | getLastKnownLocation |
| AP83      | getProvider |
| AP84      | getProviders |
| AP85      | isProviderEnabled |
| AP86      | requestLocationUpdates |
| AP87      | sendExtraCommand |
| AP88      | setTestIdProviderEnabled |
Paper—Detection of Social Media Exploitation via SMS and Camera

```
| API | Description |
|-----|-------------|
| API89 | isBluetoothA2dpOn |
| API90 | isWiredHeadsetOn |
| API91 | setBluetoothScoOn |
| API92 | setMode |
| API93 | setSpeakerphoneOn |
| API94 | startBluetoothSco |
| API95 | stopBluetoothSco |
| API96 | MediaPlayer->start |
| API97 | MediaPlayer->stop |
| API98 | MediaRecorder->setAudioSource |
| API99 | MediaRecorder->setVideoSource |
| API100 | RingtoneManager->setActualDefaultRingtoneUri |
| API101 | getActiveNetworkInfo |
| API102 | getAllNetworkInfo |
| API103 | getMobileDataEnabled |
| API104 | getNetworkInfo |
| API105 | requestRouteToHost |
| API106 | setMobileDataEnabled |
| API107 | startUsingNetworkFeature |
| API108 | stopUsingNetworkFeature |
| API109 | NetworkInfo->isConnectedOrConnecting |
| API110 | WifiLock->acquire |
| API111 | WifiLock->release |
| API112 | addNetwork |
| API113 | disableNetwork |
| API114 | disconnect |
| API115 | enableNetwork |
| API116 | getConfiguredNetworks |
| API117 | getConnectionInfo |
| API118 | getDhcpInfo |
| API119 | getScanResults |
| API120 | getWifiState |
| API121 | isWifiEnabled |
| API122 | reconnect |
| API123 | removeNetwork |
| API124 | saveConfiguration |
| API125 | setNumAllowedChannels |
| API126 | setWifiEnabled |
```
### Paper—Detection of Social Media Exploitation via SMS and Camera

| Line | Code | Description |
|------|------|-------------|
| API127 | startScan | |
| API128-AP130 | path: android/os/PowerManager | |
| API128 | $WakeLock;->acquire | |
| API129 | $WakeLock;->release | |
| API130 | reboot | |
| API131-AP132 | path: android/os/Vibrator | |
| API131 | cancel | |
| API132 | vibrate | |
| API133-AP149 | path: android/provider | |
| API133 | Browser;->clearHistory | |
| API134 | Browser;->clearSearches | |
| API135 | Browser;->getAllBookmarks | |
| API136 | Browser;->getAllVisitedUrls | |
| API137 | Contacts$People;->addToMyContactsGroup | |
| API138 | Contacts$People;->createPersonInMyContactsGroup | |
| API139 | Contacts$People;->setPhotoData | |
| API140 | ContactsContract$Contacts;->getLookupUri | |
| API141 | ContactsContract$Contacts;->openContactPhotoInputStream | |
| API142 | Settings$Secure;->putInt | |
| API143 | Settings$Secure;->putLong | |
| API144 | Settings$Secure;->putString | |
| API145 | Settings$System;->putInt | |
| API146 | Settings$System;->putString | |
| API147 | Telephony$Sms$Sent;->addMessage | |
| API148 | Telephony$Sms;->addMessageToUri | |
| API149 | Telephony$Threads;->getOrCreateThreadId | |
| API150: path: android/speech/SpeechRecognizer | startListening | |
| API151-AP170 | path: android/telephony | |
| API151 | gsm/SmsManager;->sendMultipartTextMessage | |
| API152 | gsm/SmsManager;->sendTextMessage | |
| API153 | PhoneNumberUtils;->isVoiceMailNumber | |
| API154 | SmsManager;->copyMessageToIcc | |
| API155 | SmsManager;->deleteMessageFromIcc | |
| API156 | SmsManager;->getAllMessagesFromIcc | |
| API157 | SmsManager;->sendDataMessage | |
| API158 | SmsManager;->sendMultipartTextMessage | |
| API159 | SmsManager;->sendTextMessage | |
| API160 | SmsManager;->updateMessageOnIcc | |
| API161 | TelephonyManager;->getCellLocation | |
| API162 | TelephonyManager;->getDeviceId | |
| API163 | TelephonyManager;->getDeviceSoftwareVersion | |
| API164 | TelephonyManager;->getLine1Number | |
Paper—Detection of Social Media Exploitation via SMS and Camera

| API  | Description |
|------|-------------|
| AP165 | TelephonyManager;->getNeighboringCellInfo |
| AP166 | TelephonyManager;->getSimSerialNumber |
| AP167 | TelephonyManager;->getSubscriberId |
| AP168 | TelephonyManager;->getVoiceMailAlphaTag |
| AP169 | TelephonyManager;->getVoiceMailNumber |
| AP170 | TelephonyManager;->listen AP171-AP170  : path: com/android/internal/telephony/CallerInfo AP171 | getCallerInfo |
| AP172 | markAsVoiceMail |
| AP173-AP180 : path: java | AP173 | lang/Runtime;->exec |
| AP174 | net/HttpURLConnection;->connect |
| AP175 | net/ServerSocket;->bind |
| AP176 | net/URL;->getContent |
| AP177 | net/URL;->openConnection |
| AP178 | net/URL;->openStream |
| AP179 | net/URLConnection;->connect |
| AP180 | net/URLConnection;->getInputStream |
| AP181 : path: org/apache | AP181 | http/impl/client/DefaultHttpClient;->execute |
| AP182-AP184 : path: Cipher | AP182 | AES |
| AP183 | AES/CBC/PKCS5Padding |
| AP184 | RSA/ECB/PKCS1Padding |
| AP185 | Crypto-Cipher |
| AP186 | Get-Package-Info |
| AP187 | Get-System-Service |
| AP188 | Http-Post |
| AP189 | Obfuscation-base64 |
| AP190 | Send-SMS |

Table 4. 14 APIs extracted from training dataset that are related with sms and camera

| Nominal Data | API String (starts with : android) | Function |
|--------------|----------------------------------|----------|
| AP101        | net/ConnectivityManager;->getActiveNetworkInfo | It returns details about the currently active default data network |
| AP102        | net/ConnectivityManager;->getAllNetworkInfo | It returns connection status information about all network types supported by the device |
| AP104        | net/ConnectivityManager;->getNetworkInfo | It returns connection status information about a particular network type. |
| AP147        | provider/Telephony$Sms$Sent;->addMessage | It contains all sent text-based SMS messages in the SMS app. |
| AP151        | telephony/gsm/SmsManager;->sendMultipartTextMessage | It sends a multi-part text based SMS. |
As for Table V, it displays 5 extracted APIs from the social media apps that matched and could be associated with SMS and camera exploitation from the training dataset. While Table VI displays, 138 permissions extracted from the training dataset. Table VII presents 15 permissions that are related with SMS and camera. Table VIII presents 4 permissions extracted from the social apps that matched and could be associated with SMS and camera exploitation from the training dataset.

**Table 5.** API associated with sms and camera from social media app

| Social Media Apps | API          |
|-------------------|--------------|
| SM1               | AP78, AP177  |
| SM2               | AP78, AP82   |
| SM3               | AP78, AP107  |
| SM4               | AP78         |
| SM5               | AP78, AP86   |

**Table 6.** Permission extracted from training dataset

| Nominal data | Permission                        |
|--------------|-----------------------------------|
| Q1 – Q7: path: Access |                      |
| Q1           | Checkin_properties                 |
| Q2           | Coarse_Location                    |
| Q3           | Fine_Location                      |
| Q4           | Location_EXTRA_Commands           |
| Q5           | Network_State                      |
| Q6  | Notification_Policy   |
| Q7  | Wifi_State            |
| Q8  | path: Account         |
| Q8  | Manager               |
| Q9  | path: Add             |
| Q9  | Voicemail             |
| Q10 | path: Battery         |
| Q10 | Stats                 |
| Q11 | –Q34: path: Bind      |
| Q11 | Accessibility_Service |
| Q12 | Appwidget             |
| Q13 | Carrier_Messaging_Service |
| Q14 | Carrier_Services      |
| Q15 | Chooser_Target_Service|
| Q16 | Condition_Provider_Service |
| Q17 | Device_Admin          |
| Q18 | Dream_Service         |
| Q19 | Incall_Service        |
| Q20 | Input_Method          |
| Q21 | Midi_Device_Service   |
| Q22 | Nfc_Service           |
| Q23 | Notification_Listener_Service |
| Q24 | Print_Service         |
| Q25 | Quick_Settings_Tile   |
| Q26 | Remoteviews           |
| Q27 | Screening_Service     |
| Q28 | Telecom_Connection_Service |
| Q29 | Text_Service          |
| Q30 | Tv_Input               |
| Q31 | Voice_Interaction      |
| Q32 | Vpn_Service            |
| Q33 | Vr_Listener_Service   |
| Q34 | Wallpaper              |
| Q35 | –Q37: path: Bluetooth |
| Q35 | Same as path          |
| Q36 | Admin                 |
| Q37 | Privileged             |
| Q38 | path: Body_Sensors    |
| Q38 | Same as path          |
| Q39 | –Q42: path: Broadcast |
| Q39 | Package_Removed       |
| Q40 | Sms                   |
| Q41 | Sticky                |
| Q42 | Wap_Push |
|----|---------|
| Q43 – Q44: path: Call |
| Q43 | Phone |
| Q44 | Privileged |
| Q45: path: Camera |
| Q45 | Same as path |
| Q46 – Q48: path: Capture |
| Q46 | Audio_Output |
| Q47 | Secure_Video_Output |
| Q48 | Video_Output |
| Q49 – Q53: path: Change |
| Q49 | Component_Enabled_State |
| Q50 | Configuration |
| Q51 | Network_State |
| Q52 | Wifi_Multicast_State |
| Q53 | Wifi_State |
| Q54: path: Clear_App |
| Q54 | Cache |
| Q55: path: Control_Location |
| Q55 | Updates |
| Q55–Q57: path: Delete |
| Q56 | Cache_Files |
| Q57 | Packages |
| Q58: path: Diagnostic |
| Q59: path: Disable_Keypad |
| Q60: path: Dump |
| Q61: path: Expand_Status_Bar |
| Q62: path: Factory_Test |
| Q63–Q66: path: Factory_Test |
| Q63 | Accounts |
| Q64 | Accounts_Privileged |
| Q65 | Package_Size |
| Q66 | Tasks |
| Q67: path: Global_Search |
| Q68 – Q70: path: Install |
| Q68 | Location_Provider |
| Q69 | Packages |
| Q70 | Shortcut |
| Q71: path: Internet |
| Q72: path: Kill_Background_Processes |
| Q73: path: Location_Hardware |
| Q74: path: Manage_Documents |
| Q75: path: Master_Clear |
| Q76: path: Media_Content_Control |
|---------------------------------|
| Q77-Q78: path: Modify           |
| Q77 Audio_Settings              |
| Q78 Phone_State                 |
| Q79-Q80: path: Mount            |
| Q79 Format_Filesystems          |
| Q80 Unmount_Filesystems         |
| Q81: path: Nfc                  |
| Q82: path: Package_UsageStats   |
| Q83: path: Persistent_Activity  |
| Q84: path: Process_Outgoing_Calls |
| Q85-Q96: path: Read             |
| Q85 Calendar                    |
| Q86 Call_Log                    |
| Q87 Contacts                    |
| Q88 External_Storage            |
| Q89 Frame_Buffer                |
| Q90 Input_State                 |
| Q91 Logs                        |
| Q92 Phone_State                 |
| Q93 Sms                         |
| Q94 Sync_Settings               |
| Q95 Sync_Stats                  |
| Q96 Voicemail                   |
| Q97: path: Reboot               |
| Q98-Q101: path: Receive         |
| Q98 Boot_Completed              |
| Q99 Mms                         |
| Q100 Sms                        |
| Q101 Wap_Push                   |
| Q102: path: Record_Audio        |
| Q103: path: Reorder_Tasks       |
| Q104-Q105: path: Request        |
| Q104 Ignore_Battery_Optimizations |
| Q105 Install_Packages           |
| Q106: path: Restart_Packages    |
| Q107: path: Send_Respond_Via_Message |
| Q108: path: Send_Sms            |
| Q109-Q118: path: Set            |
| Q109 Alarm                      |
| Q110 Always_Finish              |
| Q111 Animation_Scale            |
| Q112 Debug_App                  |

https://www.i-jim.org
Q113: Preferred_Applications
Q114: Process_Limit
Q115: Time
Q116: Time_Zone
Q117: Wallpaper
Q118: Wallpaper_Hints
Q119: path: Signal_Persistent_Processes
Q120: path: Status_Bar
Q121: path: System_Alert_Window
Q122: path: Transmit_Ir
Q123: path: Uninstall_Shortcut
Q124: path: Update_Device_Stats
Q125: path: Use_Fingerprint
Q126: path: Use_Sip
Q127: path: Vibrate
Q128: path: Wake_Lock
Q129- Q138: path: Write
Q129: Apn_Settings
Q130: Calendar
Q131: Call_Log
Q132: Contacts
Q133: External_Storage
Q134: Gservices
Q135: Secure_Settings
Q136: Settings
Q137: Sync_Settings
Q138: Voicemail

Table 7. 15 Permissions extracted from training dataset that are related with sms and camera

| Nominal Data | Function |
|--------------|----------|
| Q5           | Network information. |
| Q14          | Binds with services in carrier apps |
| Q40          | Broadcast SMS notification. |
| Q87          | Reads user's contact information |
| Q88          | Reads external storage. |
| Q92          | Reads to phone state (phone number, network information, any ongoing call status and registered phone account |
| Q93          | Reads SMS. |
| Q99          | Monitors incoming MMS. |
| Q100         | Receives SMS. |
| Q101         | Receives WAP. |
| Q108         | Sends SMS. |
Q132: Writes user's contact information.
Q133: Writes external storage.
Q45: Accesses to the camera.
Q88: Reads from external storage.
Q133: Writes to external storage.
Q48: Captures video recording.

| Social Media Apps | Permission     |
|-------------------|----------------|
| SM1               | Q40, Q45, Q48  |
| SM2               | Q40, Q45, Q48  |
| SM3               | Q40, Q45, Q48, Q108 |
| SM4               | Q40, Q45, Q48  |
| SM5               | Q40, Q45, Q48  |

Table 8. Permission associated with sms and camera from social media apps

The significant of having 328 APIs and permissions (combination of Table III and VI) from the mobile botnets training dataset is, it could be used as guidance for the mobile apps developer on how the attackers could exploit the smartphone via API and permission. Furthermore, from the analysis, 29 APIs and permissions (from Table IV and Table VII) are related with SMS and camera. This represents 8.8% from the training dataset and could be used for SMS and camera exploitation. From 5 selected of social media apps, only total of 9 permissions and APIs that matched with the extracted APIs and permissions from Table IV and Table VII. This represents 2.7% from the training dataset. These APIs and permissions of SMS and camera might pose privacy and financial risks for smartphone users.

4 Conclusion

Based on the experiment conducted, it showed that social media apps could be used as the attacker’s target for SMS and camera exploitation. Since Android-based application is in open-source form, malware may camouflage itself as a legitimate mobile application. The significant finding of this paper is the identification of normal API and permission for SMS and camera and possible of API and permission SMS and API exploitation. This extracted classification can be used as input or database for the development of mobile application for detection of social media exploitation.

5 Acknowledgment

The authors would like to express their gratitude to Universiti Sains Islam Malaysia (USIM) for the support and facilities provided. This research paper is funded under grant: [PPP/USG-0216/FST/30/15116]
6 References

[1] Verma, S. Arora and P. Verma, “Android OS, its security and features”, International Journal of Recent Research Aspects, Vol. 4, Issue 3, Sept 2017, pp. 241-251.

[2] L. Thomson, “Tech firms take down WireX Android botnet”, 2017, The Register, URL: https://www.theregister.co.uk/2017/08/28/tech_firms_take_down_wirex_android_botnet/.

[3] P. R. K. Varma, K. P. Raj and K. V. S. Raju, "Android mobile security by detecting and classification of malware based on permissions using machine learning algorithms," 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, 2017, pp. 294-299. https://doi.org/10.1109/i-smac.2017.8058358

[4] Talha, K. A., Alper, D. I., and Aydin, C., “APK Auditor: Permission-based Android malware detection system”, Digital Investigation, 13, 2015, pp.1–14. https://doi.org/10.1016/j.diiin.2015.01.001.

[5] Saracino, A., Sgandurra, D., Dini, G., and Martinelli, F., “MADAM: Effective and Efficient Behavior-based Android Malware Detection and Prevention”, IEEE Transactions on Dependable and Secure Computing, 5971(c), 2016, pp. 1–1. https://doi.org/10.1109/ TDSC.2016.2536605

[6] Yerima, S. Y., Sezer, S., and Muttik, I., “High accuracy android malware detection using ensemble learning”. IET Information Security, 9(6), 2015, pp. 313–320. https://doi.org/10.1049/iet-ifs.2014.0099

[7] EMB. Karbab, M. Debbabi, A. Derhab and D. Mouheb, “MalDozer: Automatic framework for android malware detection using deep learning”, Digital Investigation, Vol. 24, 2018, pp. S48-S59. https://doi.org/10.1016/j.diiin.2018.01.007

[8] Wang, Z., Cai, J., Cheng, S., and Li, W., “DroidDeepLearner: Identifying Android malware using deep learning. 37th IEEE Sarnoff Symposium, Sarnoff 2016, 2017, pp. 160–165. https://doi.org/10.1109/SARNOF.2016.7846747

[9] Kamesh, SPN. and Priya, S., “Security enhancement of authenticated RFID generation”. International Journal of Applied Engineering Research, 9(22), 2014, pp. 5968–5974. https://doi.org/10.1002/sec

[10] Li, J., Sun, L., Yan, Q., Li, Z., Sriska-an, W., and Ye, H. “Android Malware Detection”, IEEE Transactions on Industrial Informatics, 2018, 3203(c). https://doi.org/10.1109/TII.2017.2789219

[11] Li, D., Wang, Z., Li, L., Wang, Z., Wang, Y., & Xue, Y. (2017). FgDetector: Fine-grained Android Malware Detection, 311–318. https://doi.org/10.1109/DSC.2017.13

[12] Madihah Mohd Saudi and Muhammad ‘Affi b. Husainiamer, “Mobile Malware Classification via System Calls and Permission for GPS Exploitation” International Journal of Advanced Computer Science and Applications(IJACSA), 8(6), 2017. http://dx.doi.org/10.14569/IJACSA.2017.080636

[13] Burnap, P., French, R., Turner, F., & Jones, K. (2018). Malware classification using self organising feature maps and machine activity data. Computers and Security, 73, 399–410. https://doi.org/10.1016/j.cose.2017.11.016

[14] Chen, S., Xue, M., Fan, L., Hao, S., Xu, L., Zhu, H., & Li, B. (2018). Automated poisoning attacks and defenses in malware detection systems: An adversarial machine learning approach. Computers and Security, 73, 326–344. https://doi.org/10.1016/j.cose.2017.11.007

[15] Li, J., Sun, L., Yan, Q., Li, Z., Sriska-an, W., & Ye, H. (2018). Android Malware Detection. IEEE Transactions on Industrial Informatics, 3203(c). https://doi.org/10.1109/TII.2017.2789219

[16] Yerima, S. Y., & Sezer, S. (2018). DroidFusion: A Novel Multilevel Classifier Fusion Approach for Android Malware Detection. IEEE Transactions on Cybernetics, 1–14. https://doi.org/10.1109/TCYB.2017.2777960
[17] Wang, S., Yan, Q., Chen, Z., Yang, B., Zhao, C., & Conti, M. (2018). Detecting Android Malware Leveraging Text Semantics of Network Flows. IEEE Transactions on Information Forensics and Security, 13(5), 1096–1109. https://doi.org/10.1109/TIFS.2017.2771228

[18] E. Biglar Beigi, H. Hadian Jazi, N. Stakhanova and A. A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches," 2014 IEEE Conference on Communications and Network Security, San Francisco, CA, 2014, pp.247-255. https://doi.org/10.1109/cns.2014.6997492

7 Authors

Mohd Adib Azhar is a student at Faculty of Science and Technology (FST), Universiti Sains Islam Malaysia (USIM). His interest are in malwares and computer security.

Madihah Mohd Saudi is an Associate Professor at Information Security and Assurance (ISA) Programme, Faculty of Science and Technology (FST), Universiti Sains Islam Malaysia (USIM), Malaysia. Currently she is the Chief Information Officer (CIO) at USIM and leading a group called as Cybersecurity and Systems Research Unit at Islamic Science Institute (ISI), USIM. She is also senior member for IEEE Computer Society and International Association of Computer Science and IT (IACSIT) and technical committee member on IEEE Security and Privacy and Pattern Analysis and Machine Intelligence. She has over 18 years of research experience in the fields of cyber security and data mining. She has published numerous of books, policies and academic papers in international journals related with computer security and data mining. She has also won a few international innovation awards related with cyber security. Her research interests are in cyber security, data mining and bio-inspired computing. She holds 2 professional CERT related with Cyber Security from United States and received her PhD from University of Bradford, United Kingdom in Computer Security, MSc (Software Engineering) from University of Malaya (UM), and BSc(Hons) in Computer Science from National University of Malaysia (UKM) (madihah@usim.edu.my).

Dr. Azuan Ahmad is an academician at the Faculty of Science and Technology (FST), Universiti Sains Islam Malaysia (USIM). He obtained his Ph.D and M.Sc. in Computer Science from University Teknologi Malaysia. He received his B.Sc Honours in Computer Science (Information Security Assurance) from Universiti Sains Islam Malaysia. He is the certified professional in various area including cybersecurity, computer network and various operating systems. Until today, he is actively doing research in the area of cloud security, Internet of Things (IoT), Big Data and malware research (azuan@usim.edu.my).

Azreena Abu Bakar is an a lecturer at Faculty of Science and Technology (FST), Universiti Sains Islam Malaysia (USIM). Her interest are in computer security, knowledge management and information systems (azreena@usim.edu.my).

Article submitted 2019-01-15. Resubmitted 2019-02-28. Final acceptance 2019-03-15. Final version published as submitted by the authors.