FCC TAC Cyber Mobile Device Security

11/05/2015
Version 1.0

Contents

1 Purpose

This document captures requirements and development guidelines for a security checker application. The app is intended to help consumers to configure the security settings on their personal smart phones in a quick and user-friendly way. Recommended configurations follow best industry practices while reflecting the individual security needs. The app is first launched during the initial device set up and can be re-visited later to make changes to security settings or to view the current security status of the device.

The document's target audience is FCC, OS vendors as well as any party involved in the development, provisioning/hosting, and maintenance of the security checker app. In addition to development requirements & guidelines, the document makes recommendations for FCC's consideration for any related issues that are not addressed in this document. Similarly, suggestions for mobile OS vendors are included which would further promote the usefulness and effectiveness of a security checker app.

2 Scope

This document is intended to capture general requirements for the security checker app development including user experience and architecture aspects as well as a list of the security features that need to be covered by the security checker including recommendation of security levels and default selections for each of these features.

This document will not recommend third party products but may refer to other publications that do.

The following topics are out of the scope of this document:

- User education (e.g., app permissions, rooting, etc)
- Discussion on who develops and maintains the security checker
- Discussion on who will maintain the security checker and the update frequency
- Costs associated with the security checker development/maintenance/hosting
- Liabilities associated with the security checker development/maintenance/support/hosting

3 Security Checker Architecture

3.1 Client Server Architecture

There are two possible variations of this architecture.

A hybrid model in which the client SW installed is on the smart phone and web portal users can log-in to configure security settings/policies. The portal can be accessed from any device and be used to manage multiple accounts (e.g., family members, multiple devices, etc), if desired.

A variation of this model would eliminate the need for end users to login to a web portal for managing the security settings and they can do all that on the native client on the respective device. However, the security checker operator will still be able to push any security application related updates and policies in a model similar to a MDM.

3.2 Alternative Architecture

Use of a fully native client as the security checker app. All configurations and processes run locally and are managed by the user via the client app on the device.

3.3 Considerations

List of primary pros & cons of either application architecture.

3.3.1 Pros & Cons client/server model

Pros

- Better user experience: can be used to configure multiple devices at once, can log in from any device, same experience across all platforms
- Different set of security policies and features can be relatively easy supported for various OS
-

Cons

- Hosting requirements
- New threat vectors (e.g., attack security checker portal in recommended architecture)

3.3.2 Pros& Cons native client

Pros

Native apps are more responsive, and work without network/data accessCons

- Need a native app for each supported OS
- Keeping up with OS versions
- Likely different user experience across platforms

4 Intro Questionnaire

A questionnaire or a short survey should be triggered during initial on boarding to identify the intended use of the device (e.g., email only, or intended use of sensitive apps such as on-line banking/trading). The design and questions of this questionnaire are out of the scope of this document.

The results of the survey should be used to pre-select the appropriate security level and settings in the security checker app. Different options are conceivable: a brief questionnaire to help identifying the security level for the general use of the device (from low to high security) or a more detailed questionnaire leading to a more granular configuration of the device, beyond the suggested three security levels.

As a general guideline, the total number of steps required to set up the device should be limited. There should also be an option to skip the questionnaire altogether, but still launching the security checker (in which case we recommend pre-selecting the medium security level).

5 Security Checker Development Guidelines

This section described some general development guidelines for the security checker app. These guidelines should be followed by the developer, implementer and maintainer of the app.

In general, the security checker app should make the on boarding process easy for the consumer. When launched it should give the users the option to select “no”, “Low”, “Medium” or “High” security level and based on the user selection, go ahead and configure their device accordingly. A security level will be recommended and pre-selected based on the security checker intro questionnaire.

In addition, the security checker app must expose the detected security level information (which may be a security score that is more granular than the 4-tier security levels) to applications on the device via an operating system specific inner application communication framework if available. If such a framework is unavailable, an appropriate alternate approach can be selected based on technical limitations.

If the OS installed on the device is more than X versions old, app functionality should be restricted or the security score exposed to the applications should be adjusted such that apps have limited functionality.

It must be further studied how the security score can be exposed by the security checker such that third party apps requesting the score know that the score comes from a trusted source (i.e., the legitimate security checker) and the score has not been modified en route (e.g., by a malware intercepting and altering the score).

Apps installed on the device may choose to enable or disable certain functions of the application, based on the reported security level. For example, a banking application can choose to disable transfers when the reported security level is “Low”.

To reduce the set up time and improve the overall user experience of the security checker configuration, the feature & policy selections for each security level should be presented in an abbreviated form to the users, who can then change or fine tune individual selections, if desired; before the final selection will be executed.

This document provides examples for each level and default settings. However, the setting for each of the levels can be agreed upon at the time of the security checker implementation, e.g., by surveying a sample target group.

- Required platform support: at minimum Android and iOS supporting latest OS versions
- Quick set up option with pre-defined security levels (low, medium, strong)
- Users can manually select security settings
- Every selection/setting has a pre-selected default setting either based on an intro questionnaire or a medium security level
- The security checker is automatically launched at initial device set up & after factory reset. This is deployment specific feature which can be implemented by the carriers and not necessarily by the device manufacturers.
- Users can revisit the security checker app any time to view the current security configuration and status and/or to modify security settings.
- The security checker user interface must provide an easy to grasp overview of the security settings and current device status. For example, there should be a security status screen with small number of security categories and color coding displaying the current security level. Users could then expand the settings for each category and feature to gain a deeper understand of the settings and selections for each security category on the phone.
- The security checker app can be set to launch whenever a user accesses a security-related setting. This will cover only critical security setting changes. For example, enabling USB debugging. Settings such as turning Bluetooth on and off temporarily will not trigger the launch of security checker.
- The security checker app should prominently link to resources that educate the consumer on the differences between each available options to support the consumer select the type and level of security they wish to use then walk them through the process.
- The security checker app should be available at no cost from trusted app stores (e.g., OS, OEM and/or carrier app stores)

6 OS-Based Security Features

This section lists all native security features that are supported by the mobile OS that could be of concern for a smart phone user. In particular, we list the security features that should be configurable &

enforceable by the security checker app (Section 6.1) as well as the features which likely cannot be configured/enforced but which status should be displayed by the security checker (Section 6.2). Examples for four-tier configuration and status levels (no security, low security, medium security, strong security) are provided in Section 6.3.

6.1 Configurable/Enforceable Features

This section summarized all security features that should be configurable and enforceable by a security checker app. Note that some of these features might not be supported by all considered OS or might not be configurable. Further note that the current status of any of the features mentioned in this section must be displayed in the security checker app (here the security configuration levels match the security status levels).

6.1.1 Screen lock mechanisms

The security checker app must enable the configuration of the OS-supported screen lock mechanisms.

- No security: No screen lock, swipe
- Low: e.g.; pattern,
- Medium: e.g.; PIN, simple password, face recognition
- Strong: e.g.; complex password, fingerprint, two-factor

6.1.2 Screen lock timeout

The security checker app must enable the configuration of the OS-supported screen time out values if screen lock is enabled.

- Low: e.g., set the value to the max permissible time by the OS
- Medium: e.g., set the value to the median value permissible by the OS
- High: e.g., set the value to the min permissible time by the OS

6.1.3 Security Policy Updates

Some OS, device vendors or carriers offer security updates independently from "general" OS updates. For example, Google and Samsung recently announce that they will release monthly security updates[1]. On affected phones, the security checker app should display the status of these security updates, i.e., outdated vs up to date. As for OS updates, the security checker app should provide the user with an option to download & install the latest update.

If applicable, it is recommended that automatic security updates are enabled and the security checker should support the configuration of the feature.

- Low security: outdated security patch version
- Medium security: up to date security patch version
- If applicable: high security: automated security updates available.

[1] http://techcrunch.com/2015/08/05/google-and-samsung-will-now-release-monthly-ota-android-security-updates/

6.1.4 Automatic App Updates

- Low security: Enable
- Medium/high: Disable

We recommend that automatic app updates should be disabled, because if a new app version requests additional permissions (OS specific), these should be reviewed by the user (user won't be prompted if automatic updates are enabled).

Ideally, end users would only be prompted when an update requests additional permissions.

6.1.5 Disk Encryption

If disk encryption is supported and can be turned off, then the security checker should provide different security levels for the configuration (enable/disable encryption) and display the current status on the app checker home screen.

- No/low security: disabled
- Medium/high security: enabled

6.1.6 SD Card Encryption

If SD card encryption is supported and can be turned off, then the security checker should provide different security levels for the configuration (enable/disable encryption) and display the current status on the app checker home screen.

- No/low security: disabled
- Medium/high security: enabled

6.1.7 WiFi

Smart phones typically support a wide range of WiFi settings. While providing best security, disabling WiFi is too inconvenient for most users.

We recommend the control of the following settings (if supported):

- Strong WiFi security settings: Disable automatic connection to WiFi networks & Disable WPS
- Low/medium: disable WPS
- No security: none of above settings

In addition we recommend that the security checker displays the security status of the current WiFi connection.

- No/low security: open network, WEP
- Medium/high: WPA/WPA2

6.1.8 Bluetooth

Smart phones typically support a wide range of Bluetooth settings. While providing best security, disabling Bluetooth is too inconvenient for most users.

We recommend the control of the following settings (if supported):

- No/low security: Bluetooth enabled, set to discoverable/visible, no pass code required for initial pairing
- Medium security: Bluetooth enabled & limited profile (phone audio, media audio, etc)
- High security: Bluetooth disabled OR enabled & set to non-discoverable & short visibility time & pass code required for initial pairing

6.1.9 NFC

Many modern smart phones ship with NFC ship. Since many NFC-enabled applications are security sensitive (electronic wallets being the prominent example), we recommend NFC to be disabled when not in use.

- No/low security: NFC enabled
- Medium/high: NFC disabled (enabled per use)

6.1.10 App store

Goal is to permit apps only from trusted app stores. Currently, mobile OS only accept their own trusted app stores. Ideally, FFC would vet a list of trusted app stores and phones could be configured to accepts apps from any of those app stores.

- No/low security: apps from unknown sources permitted
- Medium/high security: apps from unknown sources permitted

6.2 Non-Configurable/Non-Enforceable Features with Security Status

While some security features cannot be configured or enforced by the security checker, users will still benefit from having one place to see the security status of these features (as opposed to the various places in the current menus and settings).

While obvious, it should be noted that displaying the status of a feature requires that the security checker is able to check the status of said feature. This may require a simple call to a native API but may require additional intelligence & functionality (e.g., root detection).

6.2.1 Remote Lock, Wipe, Locate and Alarm (LWLA)

Features for remotely locking, wiping, and locating a smart phone are often bundled into a single remote management tools provided by the OS. Enabling remote device management often requires setting up a user account first, e.g., for a web portal that allows managing /using the individual remote management features.

While the registration for such a separate management account will likely not being handled by the security checker app, the app should link to any OS-supported native remote management sites. In addition, the security check should display whether remote management features have been enabled or are enforced on the device.

- No/low security: Not set up/Disabled
- Medium/high security: features are enabled/enforced

6.2.2 OS Updates

OS updates are important from a security point of view because they often include security patches. However, most OS updates from OS/device/chip vendors or carriers are often not forcefully pushed to all phones. Instead, users are informed when updates are available and/or can manually check whether updates are available.

The security checker app should 1) display whether the phone runs the latest OS update available for the phone and 2), if it doesn't, give the user the option to download & install the latest update (e.g., but linking to the respective OS menu).

- Low security: outdated OS
- Medium/high security: runs latest software

6.2.3 Developer Mode

This feature is not of interest to the average user. By default this feature is turned off in all (popular) mobile OS.

Because some of the features in developer mode may introduce additional security risks to the user, the security checker should prominently display a status indicating whether the developer mode is currently enabled on the device.

6.2.4 Portable WiFi Hotspot

If a smart phone is used as a personal WiFi hotspot, a security policy should be enforced that requires the use of a PIN or password for connecting to the hotspot.

Some carriers enforce hotspot policies (general support, connection options (WiFi, Bluetooth, USB) and security requirements (e.g., WPA2 PSK)).

At the minimum, the security checker should display whether a WiFi Hotspot is currently enabled and, if so, which security policy is enforced.

- No security: WiFi Hotspot enabled, no security
- Low/ Medium security: WiFi Hotspot enabled, PIN/password with no complexity rules enforced
- High security: WiFi hotspot disabled OR WiFi Hotspot enabled with complex password rules

It requires further study to determine whether hotspots could be enabled/disable and/or security policies be configured and enforced by a security checker app.

6.2.5 Access to Location Information

Smart phones support varies method for determining location, e.g., GPS, carrier-based and WiFi –based. While offering many benefits, location information can be used for the malicious tracking of users. At

the very minimum, users should be aware of which apps have access to location information. The security checker should display a status and which apps have access.

- No/low security: fine granular location information is available to some third party apps
- Medium security: only system apps have access to location information
- High security: no apps have access

6.2.6 Locking boot loader

Some users may unlock the boot loader of their phones to be able to install another OS version, etc. This is an advanced feature that most users likely don't care about.

The security checker should prominently display a cautionary note on the app checker home screen whenever the boot loader is unlocked.

- No/low security: boot loader unlocked
- High security: boot loader locked

In addition, if supported by the OS, the security app checker should allow the user to lock the boot loader from its menu.

6.2.7 Show camera status

Some attackers use malware to turn on the host's camera to secretly watch users and/or their environment. To not interfere with the user experience of the many apps that legitimately use the device camera(s), the security checker should not enforce turning off the camera but rather notify the user whenever the camera is on.

A non-intrusive form of notification such as an icon in the top bar on the screen is recommended. There is no security rating associated with this recommendation.

7 Third Party Security Features and Services

7.1 Malware Protection

The security checker shall check for the existence of malware protection software on the device. It should recommend use one of the **trusted** malware protection software.

Some of the items to consider are that malware is essentially non-existent on iOS and on Android it originates mostly from 3rd party stores outside of US. In addition traditional Anti Virus apps, due to sandboxing and other mobile architecture features, are ineffective at detecting real risks

Recommendations for trusted software can be made available via carrier or FCC publications.

7.2 Root/Jailbreak status

Some users may intentionally root/jailbreak their phone. In other cases, malicious apps (or other players) may root/jailbreak the phones unbeknownst to the user. In either case, the security checker should prominently display whenever a device is rooted or jail broken.

For the security checker to request this device status, a third party app or service is likely necessary.

8 Recommendations for FCC's Consideration

This section summarizes recommendations for FCC, including items that are outside of the scope of this WG but would need to be addressed for a successful deployment and acceptance of a security checker app.

Recommended immediate actions:

- Get mobile OS vendors involved for feedback and help with the execution & deployment of the security checker app

Recommended next steps:

- Recommend a focus group to develop an intro questionnaire and derive more detailed guidelines e.g., based on user research, in terms of what would be an acceptable user experience for various security levels (none, low, medium, high). This approach will decrease the initial setup time and improve overall user experience

- Form group to investigate whether recommendations are technically feasible, i.e., can be supported by considered mobile OS versions

- Define a vetting process for entities to become trusted app stores.

- Implement a two pronged approach, 1) a security checker and 2) a web based or native app acting as an educational tool which is constantly refreshed with latest recommendations

- identify suitable partners for the development, deployment and maintenance of the security checker

- create a validation team to ensure design guidelines and security requirements as outlined in this document are met

9 Suggestions for Mobile OS Vendors

This section summarizes suggestions for mobile OS vendors that would allow for a more effective security checker

- For a better user experience for security conscious end users, automatic app updates should come with an option, such that end users are only prompted when an app update requests additional permissions.

- Assist with the development & deployment of the security checker app, by enabling the security checker to 1) access the status of the security settings and features listed in this document and 2) expose a security score (or something similar) that app developers can use and leverage to control app behavior without the need to create new APIs

- If not already supported, OS vendors should incorporate FIPS compliant crypto libraries on the device

- If not already supported, enable users to select whether they want to share GPS or cellular/WiFi network based location information as separate items

Appendix A: Examples of Some Relevant OS Supported Checks

The table below lists a number of checks that are available in one or more of the considered mobile OS.

We provide these device policies as a reference for the security checker developer in the hope these information may be helpful implementing some of the security & privacy requirements and guidelines provided in this document. The list is not complete and the correctness of the provided information should be verified for any targeted OS version.

The information in this table is provided courtesy of Citrix. A "y (es)" in the table denotes that the check is supported in Citrix MDM solution for the respective OS and thus the check is supported by the OS. A "-" denotes the feature is not supported by Citrix, but the OS support for the feature has not been verified.

	iOS 8	Android 5	Windows Phone 8.1
Passcode/Screen lock			
Minimum length	y	y	y
Allow Simple Values? (Allow repeated & in sequence values)	Y	-	y
Require characters? (At least one letter)	Y	-	y
Minimum number of symbols	y	-	y
Grace period before device lock policy applied to device	y	-	-
Lock device after x minutes of inactivity	y	y	y
Passcode expiration in days (1-730 days or none)	y	y	y
Previous passwords - Password History (0-50)	y	y	y
Maximum failed passcode sign-in attempts (before full wipe)	y	y	y
Minimum number of letters required in the password:	-	y	-
Minimum number of lowercase letters required in the password:	-	y	-
Minimum number of numerical digits required in the password:	-	y	-
Minimum number of symbols required in the password:	-	y	-
Minimum number of uppercase letters required in the password:	-	y	-
Biometric Recognition	Allow Touch ID to unlock device	y	-
Disk Encryption			
Enable Encryption	n/a	y	-
require device encryption	n/a	y	y
Allow storage card	n/a	-	y

WiFi Settings			
SSID	y	y	y
Auto Join to target network?	y	-	-
Open	y	y	y
Shared	y	y	-
WEP	y	-	-
WPA/WPA2	y	y	y
Allow Wi-Fi	-	-	y
Allow manual configuration	-	-	y
NFC/Bluetooth			
Allow Blutooth	-	-	y
Allow NFC	-	-	y
App Lock			
Allow store access	-	-	y
Allow developer unlock	-	-	y
Prevent Uninstall	-	y	-
Enforce Blacklist	-	y	-
Enforce Whitelist	-	y	-
Location			
Allow location services	-	-	y
Allow search to use location	-	-	y
Poll Interval (Mins/hours/days)	-	y	-
Accuracy (Meters, Feet, Yards)	y	-	-
Report if locations services are off	y	y	-
Tracking			
Force limited ad tracking	y	-	-
Allow sending diagnostic and usage data	y	-	-
Camera			
Allow use of camera	-	-	y

Appendix B: Test Cases

The table below lists tests that the security checker app would need to execute to check or enforce the security and privacy features described in Sections 6 and 7.

The Table below is incomplete and should only serve as a guideline. Whether Android, iOS, Windows Phone and other mobile OS support these checks needs to be investigated by whoever develops the security checker app.

Security/Privacy Tests	*iOS*	*Android*	Windows Phone	Comments
	Screen Lock Mechanism (6.1.1)			

Is screen lock enabled?	check available in iOS 8 and above	check available		if the lock type can't be checked; true == medium security; false == no security
What type of screen lock is active? (value == password, PIN, swipe...)				
	Screen lock timeout (6.1.2)			
Timeout set to minimum time?				high security
Timeout set to maximum time?				low security
	Security updates (6.1.3)			
Are automatic security updates enabled?	part of OS updates?			
	Automatic App Updates (6.1.4)			
Are Automatic App Updates Disabled?		not officially supported for a 3rd party app		
	Disk Encryption (6.1.5)			
Is Disk Encryped?	always true	check available		
	External SD Card Encryption (6.1.6)			
Is SD Card Encrypted?	n/a	check available		
	Wi-Fi connection (6.1.7)			

Is WPS enabled?				
Is WiFi automatic connection enabled?				
Is active WiFi connection WPA or WPA2 protected?				
	Bluetooth (6.1.8)			
is Bluetooth enabled?		check available		
is Bluetooth discoverable?				
is pairing passcode protected?				
	NFC (6.1.9)			
is NFC enabled?		check available		
	App store (6.1.10)			
are apps from unknown sources enabled?	n/a	check available		
	Remote Lock, Wipe, Locate and Alarm (6.2.1)			
is Remote Lock enabled?				
is Remote Wipe enabled?				

is remote locate enabled?				
is remote alarm enabled?				
	OS update (6.2.2)			
Is OS version up to date?				
	Developer Mode (6.2.3)			
Is Developer Mode enabled	N/A	check available		
	WiFi Hotspot (6.2.4)			
Is Wi-Fi hotspot enabled?				
Are PIN or password enforced?				
Are PIN/password complexity rules enforced?				
	Location information access (6.2.5)			
Is location enabled?		check available		
Is GPS enabled?		check available		
Can apps access fine-grained GPS ?				

	Bootloader (6.2.6)			
is Boot loader unlocked ?	n/a			
	Camera Status (6.2.7)			
Is camera on?	check available	check available		
	Malware protection(7.1)			
is a trusted Malware protection app installed ?		check if apk is installed?		
	Device rooting/jailbroken (7.2)			
Is the device rooted/jailbroken?	Check available (native + third party checks)	check available (native plus third party cheks)		