5A-6

携帯端末におけるセキュアアプリケーション構築環境の検討

岡田　英明　　清原　良三
三菱電機（株）　情報技術総合研究所

1. はじめに

従来、携帯端末では、セキュリティおよび不具合に対して十分に管理、試験された組み込みアプリケーションを搭載することにより、端末内の機密情報の保護を行ってきた。一方で、BREW や SymbianOS 搭載のスマートフォンなど、製品出荷後の端末へのネイティブアプリケーションの追加が行なわれている例もある。これはセキュリティ上のリスクを増大させており、実際にウィルス被害が発生している。携帯端末、特に携帯電話では PC 等に比べて高度な知識を必要とすることなく、より安全に、手軽に利用できることが求められており、万全の対策をとる必要がある。

現在、このリスクに対して主流となっている防御策は、アプリケーションの動作等をしかるべき検証機関が検証し署名を付与することである[1]。すなわち技術ではなく運用による対処であり、結果としてアプリケーション作成者を限定している。本稿では、セキュリティ上のリスクとその対処技術および運用を考慮したセキュリティ対策について検討する。

2. 不正なアプリケーションによる脅威とその対策

携帯端末のソフトウェアがセキュアであるためには、第一に不正なアプリケーションのインストールあるいは侵入を防ぐことが重要である。さらに、そのような侵入が発生した場合の脅威を最小化することが必要である。主な侵入経路、脅威とその対策を示す。

以下に不正なアプリケーションの侵入経路を示す。

(A) 悪意のあるアプリケーションのインストール
(B) 悪意はないがセキュリティ上の不具合のあるアプリケーションのインストール
(C) アプリケーションの脆弱性に対する攻撃による侵入

以下に不正なアプリケーションによる脅威を示す。

(ア) メール送信等による他端末への自身の拡散
(イ) DoS 攻撃、踏み台などによる他端末への攻撃
(ウ) 端末内のソフトウェア・データの改ざん、破壊
(エ) 端末、アプリケーションの機密情報の漏洩

以下にこれらの侵入、脅威に対する、現在および近い将来に利用可能な対策を示す。

① 信頼できる署名者（携帯通信キャリアや OS、端末ベンダ等）による署名を付加したアプリケーションのみをインストール可能とする。これにより、悪意のあるアプリケーションの侵入を防止する[1]。
② 第三者検証機関等によりアプリケーションのソースコードレベルで安全性が検証されたアプリケーションのみ①の署名を行なう。これにより、不具合や攻撃による侵入を防止する。
③ サンドボックスや、機能・API レベルでのアクセス制御機能を導入する。これにより、端末内のソフトウェア・データの改ざん等を防止する[2]。
④ 端末組み込みの機密情報への直接アクセスを禁止し[3][4][5]、特定の操作を間接的に行う機密分離 API を導入する。これにより、機密情報の漏洩を防止する。例えば、端末の電話帳機能へのアクセスは許可するが、電話番号そのものの取得は許可せず、電話番号に対応するインデックスの取得のみを許可する。発信、電話番号の画面表示などの操作を行なう場合、上記インデックスを指定して対応する機密分離 API を呼び出す。
⑤ 通信量、通信先などの監視機能を導入し、想定外の通信を行なった場合にはアプリケーションの機能を停止させる。これにより、被害を最小化する。

以下に上記対策を実施したときの効果を示す。また、侵入、脅威に対して有効な対策を示す。

脅威	効果	有効な対策				
A	解決される。	①	②			
B	軽減される。	①	②	③	④	
C	軽減される。		②			
ア	メール送信などの機能（アクセス権限）を持つアプリケーションに対しては解決されないが、監視機能により軽減される。			③	④	⑤
イ	パケット送信などの機能（アクセス権限）を持つアプリケーションに対しては解決されないが、監視機能により軽減される。			③	④	⑤
ウ	アプリケーション自身のデータに対しては解決されないが、他アプリケーションやシステムに対しては解決される。			③	④	
エ	アプリケーション自身のデータに対しては解決されないが、他アプリケーションやシステムに対しては解決される。			③	④	

Study of secure application platform for mobile terimals
Hideaki OKADA, Ryozo KIYOHARA
Mitsubishi Electric Corporation Information Technology R&D Center

3. 残存する脅威への対策

残存する脅威とそれに対する対策について検討する。

- ア（他端末への拡散）、イ（他端末への攻撃）
  一部用途、例えばメールアプリケーションにおけるメール送信においては、送信前にユーザ確認を行うことにより不正な送信を発見することができる。しかし、使い勝手の観点から、ソケット通信等を含めたすべての通信に対してユーザ確認を行うことは現実的には不可能であり、不正アプリケーションによる通信をすべて防御することはできない。
  したがって、詳細なアクセス制御を可能にすることにより、被害の影響範囲を限定することが重要である。例えば、被害がそのアプリケーションを用いたサービスを提供するサーバに限定できる場合、運用による制限（ソースコードの検証など）を緩和できる。
- ウ（端末データの改ざん）、エ（機密情報の漏洩）
  例えばナビゲーション機能における GPS によるデータなど、本質的にアプリケーションに対して分離できない機密情報を扱うアプリケーションがある。このようなアプリケーションに対しては、その機密情報の内容に応じて運用による対処を変更する（ソースコードの検証を行うなど）必要がある。
  一方、メールアプリケーションにおけるアドレス帳機能など、データとしてはアプリケーション固有であっても、多くのアプリケーションに共通の機能がある。このような機能に対して機密分離を実現したライブラリ（機密分離付共通ライブラリと呼ぶ）を適用することにより、端末内の機密情報と同様に漏洩を防止することができる。

4. 対策の問題点、課題

4.1 アプリケーションの検証、署名

対策①、②は不正なアプリケーションの侵入に対しては有効であるが、一方でアプリケーションの導入・普及の障壁となる。したがって、可能な限りその適用範囲が小さくなるようにすべきである。

- ①署名機能の導入
  特定の署名者により署名されたアプリケーションのみインストールを認めることは、一般ユーザによるアプリケーションいわゆる勝手アプリの作成を不可能する。これは、情報通信プラットフォームとしての携帯端末の可能性を一部奪うことになる。
- ②ソースコード検証の導入
  ソースコードの検証は、サービス展開の遅れやノウハウ流出の危険性をもたらし、広い意味で開発コストを増大させる。

したがって、署名の有無、署名者、署名に必要な条件による、以下のアプリケーションに対する署名レベルの導入が必要である。

1）無署名アプリケーション

2）不特定機関による署名アプリケーション（携帯通信キャリアや OS、端末ベンダではなく、サービスを行なう企業や第三者機関による署名つきのアプリケーション）

3）特定機関（携帯通信キャリアや OS、端末ベンダ）による署名アプリケーション

4）ソースコード検証を伴う署名アプリケーション

4.2 機密分離機能の実現

対策④機密分離機能や機密分離付共通ライブラリの導入は、アプリケーション作成上の自由度を小さくする。一方、商用アプリケーションでは見た目も重要であり、自由度の高さが求められる。その要求に耐え得るライブラリを提供できなければ、機密分離機能の普及が妨げられることになる。したがって、実際のアプリケーションの構築において使い勝手のよい、必要十分な機能を持つライブラリを実現することが不可欠である。

5. おわりに

ネイティブアプリケーションの追加によるセキュリティリスクに対して、技術と運用の双方を考慮した対策を行なうため、主な脅威および対策を整理した。

技術による対策として、③アクセス制御、④機密分離 API、⑤通信監視を導入するとともに、機密分離付共通ライブラリ、通信に対する詳細なアクセス制御機能を導入することが必要であることを示した。このような技術による対策を導入した上で、脅威および導入のしやすさを考慮した署名レベルを導入することが必要であることを示した。

今後は、既存のアプリケーションを分析し、有効かつ必要十分なアクセス権限の定義、機密分離ライブラリの詳細機能について検討する。これにより、署名レベルの妥当性を検証するとともに、署名レベルに対応する技術対策、および署名レベルに応じて許可できるアクセス権について検討していく。

参考文献

[1]Symbian Signed,https://www.symbiansigned.com/

[2]Symbian, Platform Security - a Technical Overview, http://www.symbian.com/

[3]Douglas Kilpatrick, Privman:A Library for Partitioning Applications, In Proceedings of USENIX 2003 Annual Technical Conference, FREENIX Track, 2003.

[4]太田他,セキュア携帯機のための機密分割型アプリケーション自己監視方式, DICOMO 2005

[5]Srivaths Ravi,Security in embedded systems: Design challenges, ACM Transactions on Embedded Computing Systems, Volume 3, Issue 3, 2004