Vol.2011-CSEC-53 No.2
Vol.2011-IOT-13 No.2
2011/5/12

Android フォンのアプリ管理 ～ホワイトリスト方式～

竹森敬祐[†]　川端秀明[†]　磯原隆将[†]
窪田 歩[†]　池野潤一[††]

Android フォンの特徴は、誰もが自由にアプリケーションの開発・公開を行え、ユーザは Market プレイスからアプリケーションをダウンロード･インストールして端末の高機能化を図れる点にある。このインストールの過程で、アプリケーションが利用する機能や情報の一覧が表示されるが、ここからユーザがアプリケーションに潜む脅威を推定することは難しく、誤ってマルウェアに感染してしまう恐れがある。Android フォンのインシデントの殆どは、このアプリケーションのインストールによって生じる。そこで本研究では、安全性の求められる法人向け端末の保護策として、安全性を確認したアプリケーションだけを予めホワイトリスト化しておき、業務用端末にアプリケーションがインストールされるタイミングで、これと一致しないアプリケーションを駆除するフレームワークを提案・実装する。そして、1000 台規模の実証実験を行い、Android フォンにとっての最も大きな脅威を確実に低減できることを示す。

Restriction framework for Android application -Whitelitst-based installation-

Keisuke Takemori[†] Hideaki Kawabata[†] Takamasa Isohara[†]
Ayumu Kubota[†] and Jyunichi Ikeno[††]

Android OS is an attractive platform for a mobile phone because anyone can develop an application and upload it to a market place. Users download and install the application in order to upgrade their Android phone. When an application is installed, the user needs to check access permissions. As it is difficult to recognize potential threats of the application by referencing the access permissions, many users infect Android phones with malware via the market place. Most security incidents of Android phones occur when users download malware via the market place. In this paper, we propose a secure application management framework for business use of Android phones. Only the authorized application can be installed in the Android phone by comparing with a whitelist. In addition, we show a result of field trial using 1,000 Android phones. The framework is expected for an effective countermeasure to the business Android phones.

1. はじめに

昨今、誰もが自由にアプリケーションを開発･公開･インストールできる Android OS [1]を使った携帯電話が注目を集めている。Android OS には、携帯電話が持つ機能や情報をアプリケーションから利用するための Application Interface（API）が豊富に設けられている。よって、利便性の高いアプリケーションを実現できる点で、従来の携帯電話で実現が難しかった業務用アプリケーションの開発が促進されることから、法人での業務用端末としての利用が期待されている。

ここで安全面に注目すると、Android OS には任意のホストからパケットを受け取るデフォルトオープンな Port は無く、Windows OS で問題となっているワームへの脅威は低い。このため、法人利用で直面する脅威の多くは、情報漏洩や端末破壊を引き起こすマルウェアを誤ってインストールすることでの感染と言える。

これに対して、Android OS には Dalvik とよばれる仮想マシンのサンドボックス上でアプリケーションを隔離・実行する仕組みが設けられている。しかし、利便性の高いアプリケーションを実現するフレームワークとして、アプリケーションの素行を記したマニフェストをユーザが承認することで、サンドボックスに穴を開ける機構が設けられている。これは、アプリケーションが携帯電話の機能や情報へアクセスするための許可をパーミッションとして規定しておき、インストール時にユーザが参照して承認する仕組みである[2]。しかしパーミッションは、利用する機能や情報を大まかな単位で規定したものであり、ここから脅威を推定することは難しい。また、複数のパーミッションが組み合わされたときに実現される総合的な作用を読み取ることはさらに難しく、悪意を持ったアプリケーションに感染してしまう危険性がある。

Android Market [3]では、ユーザからの報告によって掲載されたマルウェアを迅速に排除する運用がなされている。しかし、Android フォンをユーザ好みに改造するための root 権限を奪うツールの中には、ユーザから正規の設定ツールとみなされる場合があり、掲載され続けるケースがある。また、人気のアプリケーションにボット機能を内包させたマルウェアが掲載されたこともあり[4]、駆除される前に多数のユーザがダウンロードしてしまう事故も発生している。

マルウェア対策ソフトとして、既知のマルウェアをキーに検知するブラックリスト方式がある。しかし、Android Market から駆除されている有名なマルウェアは検知できるものの、掲載中の新しいマルウェアや日本だけで販売される Android フォンを狙ったターゲット型マルウェアを検知できない問題がある。現在、世界中のアプリケー

[†] KDDI 研究所 ネットワークセキュリティグループ
KDDI R&D Laboratories Inc. Network security laboratory
[††] KDDI
KDDI

Vol.2011-CSEC-53 No.2
Vol.2011-IOT-13 No.2
2011/5/12

ション開発者が次々と Android 向けアプリケーションをリリースしており、Android Market だけでも 1,000 件／日を超えるアプリケーションが新規にリリースされる為[5]、ブラックリストのメンテナンスが追いつかない問題も潜んでいる。

そこで本研究では、安全性の求められる法人向け Android フォンの保護策として、安全性が確認されたアプリケーションだけを予めホワイトリスト化しておき、業務用端末にアプリケーションがインストールされるタイミングで、これと一致しない未承認アプリケーションを駆除する仕組みを提案する。これは、Market プレイスや SD カード、adb シェル[6]を通じてアプリケーションが端末にインストールされたことを Android のインテントの仕組みを利用して把握し、そのアプリケーション名と署名から検証することで、未承認アプリケーションを検知する。こうした承認アプリケーションのインストールや未承認アプリケーションの駆除は、操作履歴として Android フォンからセンタ局に報告することで、法人の管理者が社員に配布した業務用端末の状態を把握できるようになる。我々は、Android フォン向けクライアントツールとセンタ局サーバを実装し、1000 台規模での実証実験を行った。その結果、利便性重視の Android フォンに有効な安全策を実現できることが確認され、業務用端末として活用できる目処を得た。

2. Android フォンのセキュリティ上の脅威

表 1 Android フォンにおける脅威と比較

	従来の携帯	Android™フォン	PC（汎用OS）
OS/開発情報（脆弱性）	クローズ（限定的）	オープン（多数が公知化）	オープン *1（多数が公知化）
API（サンドボックス）	限定的（全アプリへ強制）	豊富＋ユーザ承認（全アプリへ承認型）	豊富（無し *1）
端末識別子管理	強（堅牢に保護）	弱（APIアクセス可）	ー
コンソール	無	有	有
ディスクの暗号化	無	限定的	有
アプリ販売（Market）	クローズ	オープン	オープン
管理者権限アプリ	無	無 *2	有
マルウェア感染	限定的	有（手動）	有（手動＆自動）
Webスクリプト攻撃	限定的	有	有
フィッシング	有	有	有
総合的な安全度	★★★	★★	★

最大の脅威

*1 一部例外あり
*2 Android™OSの想定外であるが、管理者権限の奪取や管理者権限を利用するアプリがある。

2.1 脅威の一覧

従来からの携帯電話や PC と Android フォンを比較しながら、セキュリティの脅威を表 1 に纏める。オープンな Android OS では、脆弱性に関する情報が公開されていること、端末機能を利用する API が豊富に用意されていること、端末識別子（International Mobile Equipment Identity：IMEI）や SIM 識別子（International Mobile Subscriber Identity：IMSI）などの ID 情報を取得する API も用意されていることなど、マルウェアへの悪用が容易である。また、Android フォンには、PC に USB 接続してコンソールから操作する adb シェルがあり、管理者権限が奪われた状態では、単なる PC として操作されてしまう。さらに、/system 領域の暗号化が図られていないこと、マルウェアへの感染が懸念されること、Web ブラウザを通じたスクリプト攻撃や、ユーザを騙すフィッシング攻撃なども懸念される。

これらの中で最大の脅威は、ユーザが誤ってマルウェアをインストールしてしまう感染である。特に、管理者権限を奪うマルウェアに感染した場合には、Android のセキュリティ機構が無効化されてしまうことで、勝手なクレジットカード決済や、他のマルウェアへの自動感染も懸念される。

2.2 マルウェア感染への導線

Android フォンがマルウェアに感染する導線は、Market プレイス、adb シェル、SD カードを利用したアプリケーションのインストールの 3 種類がある。

2.2.1 Market プレイス

多くのユーザは、アプリケーションを Market プレイスから入手してインストールしている。図 1 に、市販の Android フォンの画面上に現れる Android Market と au one Market へのアイコンを示す。Android market では、アプリケーションの公開はオープンであり、誰もが自由に開発したアプリケーションを、セキュリティに関する事前審査を経ないでアップロードできる。代わりに、ユーザからの通報によって、マルウェアを特定し削除する迅速な事後対応を図っている。よって、公開から削除までのタイムラグがあり、2011 年 3 月には正規のアプリケーションに Bot 機能が組み込まれた Droid Dream と呼ばれるマルウェアが掲載され、感染が拡がった[4]。au one Market では、アプリケーションに対する静的解析による潜在的な脅威の評価と、動的解析による顕在化した脅威の評価による事前審査がなされている。但し、マルウェアの高度化に伴い完全な検知を保証することはできないため、事後対応の仕組みも導入されている。

図 1 Market プレイスへのアイコン

Vol.2011-CSEC-53 No.2
Vol.2011-IOT-13 No.2
2011/5/12

こうした、事前審査や迅速な事後対応のなされている Market プレイスからの感染被害は少ないものの、審査のなされない提供元不明の Market プレイスを通じたマルウェア感染が懸念される。2010 年 12 月には、Geimini と呼ばれる Bot コードを忍ばせたトロイの木馬が、中国の Market プレイスを通じて配布された[7]。この基となったアプリケーションは有料であるにも関わらず、それを無料で配布していることから、感染の拡大を狙った悪意の Market プレイスと言える。

2.2.2 adb シェル

Android フォンには、PC にダウンロードしたアプリケーションを、コマンドラインから手動でインストールする adb シェルが設けられている[6]。これは、PC に Android SDK [8]と USB 接続用ドライバをインストールすることで、PC のターミナル画面から、*adb install "アプリケーション名"* でインストールできるインタフェースである。このとき、パーミッションを確認する表示はなく、誤ってマルウェアをインストールしてしまう懸念がある。

2.2.3 SD カード

Android フォン内のフォルダやファイル操作を行うファイル管理アプリケーションがある[9]。このアプリケーション画面から、SD カード上のアプリケーションをクリックすることで、Android の Installer が呼び出され、パーミッションの承認フェースを経て、インストールされる。ここでも、パーミッションからマルウェアを見抜くことは難しく、マルウェアへの感染が懸念される。

3. Android のセキュリティ機構

Android OS は、Google 社によって開発された携帯電話向けの OS であり[1]、Dalvik 仮想マシンで安全性を、パーミッション機構で利便性を制御しており、このトレードオフをユーザに委ねる特徴がある。

3.1 Dalvik 仮想マシン

Android OS は Linux 2.6 をベースに、ライブラリや Dalvik 仮想マシンが設けられている。アプリケーションを、Dalvik 仮想マシン上で隔離・実行することで、その影響を抑えている[1]。

3.2 パーミッション機構

図 2 に、アプリケーションをインストールする際に、アプリケーションがアクセスする情報や機能をパーミッションとして一覧表示して、ユーザに承認を仰ぐ機構を紹介する。"OK"をクリックすることで、表示されたパーミッションを全て承認したことになり、Dalvik のサンドボックスをパーミッションの範囲で越えることができるようになる。端末の戻るボタンをクリックすると、インストールが中止される。

図 2 アプリケーションのインストール時にユーザ承認を求めるパーミッション

図 2 は、あるピアノアプリのインストール時に表示されるパーミッションの様子を示している。このアプリは、画面上にピアノの鍵盤が現れ、押すとピアノ音が出力される。ピアノの機能を考えた場合、GPS による位置情報、携帯のステータスや ID 情報を利用する必要はなく、またインターネットへの通信機能も不要である。このように、ユーザはアプリの趣旨とパーミッションのギャップから潜在的な脅威を読み取る必要がある。実際には、本ピアノには広告機能が内包されており、ユーザに合わせたターゲット広告を表示するために、これらの情報が広告サーバへ送信されている。

Android 2.2 の API Level 7 では、Android 標準のパーミッションが約 120 個用意されている[2]。他にも、アプリケーションが独自に宣言するパーミッションを設けることもできる。

4. 既存技術と課題

4.1 マルウェア対策ソフト

Android OS 向けのマルウェア対策ソフトがリリースされている[10], [11]。これらはアプリケーションがインストールされるタイミングでの検査と、任意の時刻にディス

Vol.2011-CSEC-53 No.2
Vol.2011-IOT-13 No.2
2011/5/12

ク内をフルスキャンする検査が設けられている。

しかし、Android フォンにとってのマルウェアの定義が不明である中、情報漏洩を引き起こすアプリケーションの多くが検知の対象になっていない[12], [13]。そもそも Android OS では、機能や情報へのアクセスを、パーミッションを通じてユーザが承認する仕組みがあるため、情報漏洩マルウェアとみなさない考え方もある。また、世界中の開発者が次々とアプリケーションを開発・公開することで、検知のためのパターンファイルの作成が追いつかないブラックリスト方式の限界も懸念される。

4.2 マルウェア感染への導線の阻止

Android フォンと呼ばれる為には、Android OS に改変を加えることはできず、世界共通の OS の状態であることを保証する Compatibility Test Suite (CTS)認証を経なければならない[14]。このため、例えセキュリティ重視とは言え、Android OS が持つマルウェア感染への導線を削除することはできない。

5. 提案と実装

業務に必要かつ確実に安全なアプリケーションのみを、法人の管理者が予めホワイトリスト化しておき、これを業務用 Android フォンに配信することで、未承認アプリケーションの侵入を防ぐ。ホワイトリスト方式を提案する。実装にあたっては、Android OS に改変を加えない一般アプリケーションとして実現する。このため、法人利用における前提として、業務用端末から対策アプリケーションを削除する行為や、未認定アプリケーションの削除を拒否する行為は、社内規則で禁止されるものとし、社員は違反しないこととする。本稿では、こうした違反を管理者が把握できる機構を設ける。

5.1 構成（図 3）

① 法人の管理者は、業務に必要なアプリケーションを、Market プレイスから管理用端末にインストールする。
② このアプリケーションの機能や安全性を把握するために、実行してみる。
③ 該当のアプリケーション（パッケージとも呼ばれる）や動作ログを、安全性を評価するセンタ局へ送付する。
④ センタ局では、パッケージから得られるパーミッションに関する静的解析と、動作ログに対する動的解析を行い、それぞれ潜在脅威と顕在脅威を評価する。そして、脅威の低いアプリケーションを認定することで、ホワイトリストを作成する。
⑤ 業務用端末は、一定周期でホワイトリストをセンタ局から取得する。
⑥ 業務用端末では、3 つの導線のいずれかを通じてアプリケーションがインストールされたことを検知し、ホワイトリストと比較することで、認定の場合には何もせず、未認定の場合にはアンインストールを社員に促す。
⑦ ユーザによるインストール履歴や端末に組み込まれているアプリケーション一覧をセンタ局に送付することで、管理者は配布した業務用端末の状態を把握する。

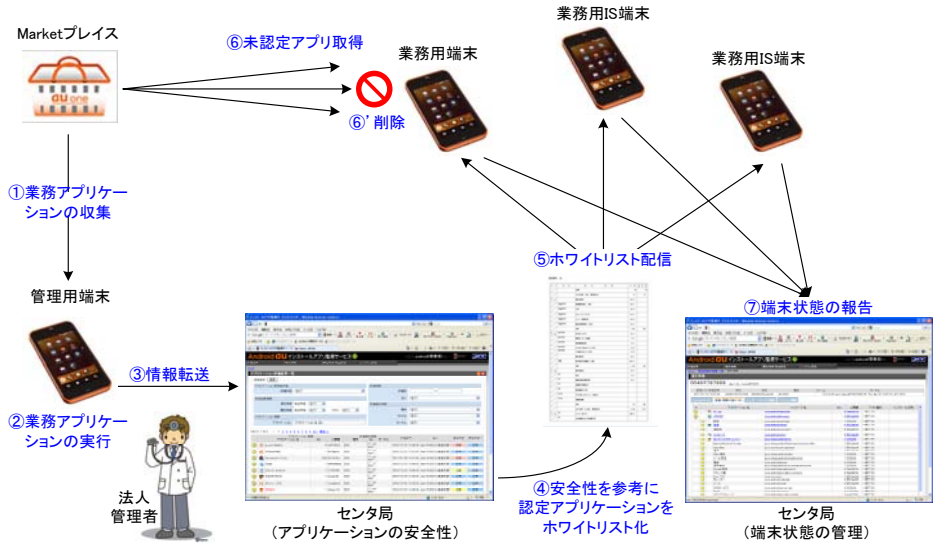

図 3 Android フォン向けアプリ管理機構　～ホワイトリスト方式～

5.2 管理用端末

管理用端末では、Market プレイスからアプリケーションを取得して、これを実行することで、静的解析に必要なパッケージと動的解析に必要な動作ログを得る。但し、Android のコピープロテクションが指定されたアプリケーションの場合、一般アプリケーションからの Read 権限の無い /data/app-private/ へ保存されるため、パッケージ情報を得ることはできず、取得可能なパーミッション情報のみをセンタ局へ送付する。動作ログとしては、アプリケーション層で取得できる Logcat Log を収集する[12], [13]。また、管理用端末を WiFi 接続し、ネットワーク側で通信パケットを取得する。

図 4 に、管理用端末で動作する本ツールの操作画面を示す。”アプリ選択”タブには、管理者が評価すべき/data/app/もしくは/data/app-private/に保存されたダウンロードアプリケーションの一覧を表示する。Android フォンにプリインストールされている /system/app/に保存されているアプリケーションは対象外とする。アプリ選択タブから、アイコンをクリックすることで、Logcat Log を収集するコマンドを実行し、アプリケーションの起動が始まる。そして、ある一定時間が経過すると、Logcat Log を停止して、パッケージ情報と Logcat Log を合わせて、実行結果タブへと移す。管理者は、評価の必要なアプリが、アプリ選択タブから無くなるまで実行し、実行結果タブに集められた結果を纏めて、センタ局へ送信する。

Vol.2011-CSEC-53 No.2
Vol.2011-IOT-13 No.2
2011/5/12

図 4 管理用 Android フォンでのアプリケーション情報収集ツール

5.3 安全性評価

センタ局では送られてきた情報を基に、潜在脅威と顕在脅威を評価する。潜在脅威は、不必要なパーミッションの付与を検知する。顕在脅威は、Logcat ログや通信パケットから、顕在化した脅威を評価する。法人の管理者は、アプリーションの潜在脅威と顕在脅威が安全であることを確認した後に、認定する。尚、解析アルゴリズムについては別の論文で扱うこととし、ここでは割愛する[12], [13]。

5.4 ホワイトリスト

アプリケーションの認定／未認定を判定するために、パッケージ名と署名データに含まれる公開鍵を用いることとした。パッケージ名は、アプリケーションに対する認証の意図であり、公開鍵は開発者に対する認証の意図である。パッケージのハッシュ値やバージョンも合わせて検証することで厳密性は高まるが、端末上での処理のため処理負荷の軽減などを考慮した。

5.5 業務用端末

未認定アプリケーションの侵入を検知すると、Installer を制御してユーザの承認なしに強制アンインストールする設計が良い。しかし、こうした自動的な制御を行うには、Android フォンで保護されるべき/system 領域への事前の組み込みが必要であり、様々な Android フォンに対応できないため、一般権限アプリケーションとして実装する。この対策アプリケーションは、センタ局にホワイトリストの更新を一定周期で確認することとし、センタ局へのアクセスが集中しないように、周期内の時刻は端末毎にランダムになるように設計した。アプリケーションの検査は、インストールを知らせる Intent Receiver を記述して、インストールされたアプリケーションとホワイトリストのマッチングを行う。もし未認定アプリケーションと判定された場合には、Installer を呼び出してアンインストールをユーザに促す(図 5)。

図 5 業務用 Android フォンでの未認定アプリケーションのアンインストール

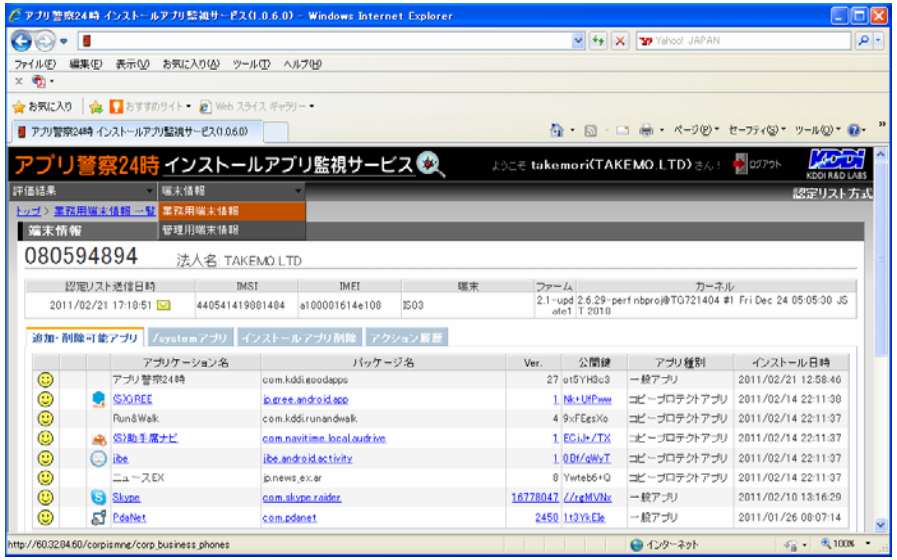

図 6 業務用 Android フォンの状態管理

Vol.2011-CSEC-53 No.2
Vol.2011-IOT-13 No.2
2011/5/12

5.6 端末状態の管理

端末にインストールされているアプリケーション一覧や、社員によるインストール操作の履歴をセンタ局へ送付することで、各端末に組み込まれているアプリケーションを把握する(図 6)。また、定期的に行われるホワイトリストのダウンロード履歴も管理する。これにより。社員が業務用端末から対策アプリケーションを削除したことや、未認定アプリケーションを端末に組み込み続ける状態を把握できるようになる。

6. 評価

6.1 削除時間

Android アプリケーションは、端末にインストールされただけでは被害を及ぼすことはなく、実行されたときに攻撃を行う。よって、アプリケーションが起動する前に、削除する必要がある。起動は、ユーザによるクリックの他に、端末の再起動や画面遷移を行うときに発行されるインテントをきっかけにでき、可能な限り迅速な削除が求められる。そこで我々は、業務用端末上でアプリケーションのインストールが完了してから、ホワイトリストとマッチングを行う時間と、アンインストールが指示されてから消去が完了するまでの時間を、以下の条件で 5 回測定した。結果を表 2 に示す。

- ホワイトリスト：10 件の登録
- 端末：IS03
- OS：Android2.1

表 2 ホワイトリストとのマッチングとアンインストール時間(msec)

	1 回目	2 回目	3 回目	4 回目	5 回目	平均(msec)
パターンマッチ	1,056	896	1,346	1,323	979	1,120
アンインストール	94	34	67	91	59	69

これより、ホワイトリストとのマッチングが 1,120msec でアンインストールが 69msec であり、アプリケーションによる被害が出る前に消去できる目処を得た。

6.2 トライアル実験

我々は、1,000 台程の市販の IS03 Android フォンに対して本機構を導入し、業務用端末として社員への配布を行った。センタ局のスペックは以下の通りである。

- DELL PowerEdge R210：CPU-Cerelon 2.26GH、メモリ 1 GB
- ブロードバンド回線：100Mbps

本環境において、ホワイトリストのダウンロード周期を 3 時間に設定し、実験を行ったところ、特にセンタ局の負荷が高まることなくホワイトリストの配布や、端末状態の収集を行えた。

配布した全ての端末に、認定アプリケーションしかインストールされていない状態を確認するとともに、数名の社員が Market プレイスから未認定アプリケーションのインストールを試みて、駆除された履歴を確認した。

ここで問題点として、ホワイトリストの更新を 1 日間行わなかった端末は、本機能がアンインストールされたとみなしたが、実際には休日に電源を落としていた場合や、電池切れで通信を行えなかったものも見られ、誤アラームを受け取った。そこで、この期間を 3 日間として再評価を行ったところ、非更新に関するアラームは低減したものの、本機能が削除された状態を把握するまでのタイムラグが大きくなる懸念がある。

7. おわりに

本稿では、法人の管理者が認めるアプリケーションを予めホワイトリスト化しておき、業務用端末側でアプリケーションをインストールするときに、未認定アプリケーションを削除するアプリケーション管理の機構を提案した。これにより、Market プレイスに次々と現れるマルウェアへの感染を防止でき、Android フォンを業務用端末化できる目処を得た。

参考文献

[1] Android OS, http://www.android.com/jp/
[2] Access permissions, http://developer.android.com/intl/ja/reference/android/Manifest.permission.html
[3] Android Market, http://www.android.com/market/
[4] Droid Dream, http://blogs.computerworld.com/17929/google_android_market_kills_droid_dream_malware_in_trojans
[5] AndroLib, http://jp.androlib.com/appstats.aspx
[6] Android Debug Bridge, http://developer.android.com/guide/developing/tools/adb.html
[7] Geimini, http://blog.mylookout.com/2010/12/geinimi_trojan/
[8] Android SDK, http://developer.android.com/sdk/index.html
[9] Astro File Manager, https://market.android.com/details?id=com.metago.astro
[10] Norton Mobile Security, http://us.norton.com/mobile-security
[11] F-Secure Mobile Security, http://mobile.f-secure.com/
[12] 磯原隆将、竹森敬祐、窪田歩、高野智秋, “Android 向けアプリケーションの挙動に注目したマルウェア検知”, IEICE, SCIS2011, 3B3-2, 2011 年 1 月.
[13] 竹森敬祐、磯原隆将、窪田歩、高野智秋, “Android 携帯電話上での情報漏洩検知”, IEICE, SICS2011, 3B3-32011 年 1 月.
[14] Android CTS, http://source.android.com/compatibility/cts-intro.html