5Z-2

Android 端末におけるルート化に関する調査

堀洋輔† 鈴木舞音‡ 上原崇文‡ 金子洋平‡ 齋藤孝道†
明治大学† 明治大学大学院‡

1. はじめに

現在，商業販売されている Android 端末に対して，Android 端末ベンダがルート権限を与えてない一般利用者が不正にルート権限を獲得する行為，いわゆる，ルート化と呼ばれる不正行為が問題視されている．

Android では，サンドボックス機能などのセキュリティ機能上，一般利用者が使用するアプリケーション（以降，アプリという）には一般権限が与えられ，アクセスできるリソースが限られている．しかし，ルート化によってルート権限を得たアプリでは，通常アクセスすることができないリソースに対してアクセスすることができるようになる．

たとえば，ルート権限を得たアプリでは，カメラのシャッター音を無効化するなどの本来 Android ではできない操作を行える．また，本来許諾のないでデザリングなどの有料機能の不正利用や有料アプリの不正コピーなどのルート権限を利用した不正行為が発生し，そのことが Android に関わる企業にとって問題になっている．Android のルート化の手法は，インターネット上に公開されていることが多く，一般利用者でも容易にルート化が行える環境にある．

ルート化の対策として，（ルート化の手段の一つである）脆弱性を排除するためのアップデートや，ルート化された端末で使用されるアプリ名の検知などの対策が行われている．しかし，これらの対策はルート化を行うことに対しての根本的な対策となってはない．

そこで，本論文では，Android 端末におけるルート化に至るまでの手法に関して調査し，いくつかの手法について取り上げる．

Investigation about rooting in an Android terminal
†Yosuke Hori
‡Maine Suzuki ‡Takafumi Uehara ‡Yohei Kaneko
†Takamichi Saito
Meiji University(†),Graduate School of Meiji University(‡)
1-1-1 Higashimita, Tama-ku, Kawasaki-shi, Kanagawa 214-8571, Japan(†)(‡)

2. Android 端末におけるルート化に関するセキュリティ

2.1. サンドボックス機能

サンドボックス機能とは，保護された領域内でプロセスを動作させることで，プロセスが不正に操作されることを防ぐセキュリティモデルである．

Android では，インストール時にアプリに固有のユーザ識別子とグループ識別子が割り振られ，そのアプリ専用のリソースが用意される．そのアプリ専用のリソースには，通常，他の識別子をもつアプリからアクセスすることはできない．また，アプリのプロセスごとに専用の Dalvik 仮想マシンが割り当てられることにより，プロセスは独立に実行するので，他のアプリの領域にアクセスすることはできない．そのため，一般利用者が使用するアプリには一般権限が与えられているので，他の識別子を持つリソースやシステムファイルへアクセスすることは基本的にできない．

2.2. データ実行防止機能

プロセス実行時に割り当てられるメモリ空間の各領域へは「読み・書き・実行」それぞれパーミッションの権限割り当てが可能であり，古い OS ではデータ領域でも実行権限を持つ．そのため，実行権限をもつデータ領域があれば，バッファオーバーフロー攻撃（以降，BoF 攻撃という）によって埋め込まれた Shell コードなどをデータ領域でも実行させることができる．そのような攻撃を防ぐために，Android 端末で主に使用される CPU アーキテクチャである ARM には，eXecute-Nerver という機能がある．これを用いることにより，ハードウエアレベルでメモリ領域をコード（実行可能）とデータ（実行不能）領域に区別する．実行不能領域にあらかじめ属性を付与することで，プロセッサはこの領域にあるコードの実行を拒否する．

2.3. ASLR 機能

たとえば，BoF 攻撃では，関数終了時に実行されるリターンアドレスや関数の呼び出しアド

レスを格納した変数の値などを上書きすることで埋め込まれたコードの実行を行う．この種の攻撃では，メモリ内の対象データのアドレスを把握する必要がある．

ASLR（Address Space Layout Randomization）機能は，プロセスに割り当てられるメモリ空間の各領域の配置をプロセスの実行時にランダムにする．ランダム化される領域はスタック領域，ヒープ領域，共有ライブラリ，及びコード領域である．ASLR 機能により，攻撃者はメモリ空間の各領域のアドレスの把握することが困難になるため，攻撃者が攻撃を成功させる確率を低くすることができる．

3. ルート化の分類

3.1. 一時的ルート化

端末の脆弱性を攻撃して不正な権限昇格を行い，一時的にルート権限を使用できる状態にすることを一時的ルート化と呼ぶ．この一時的ルート化状態では，端末の再起動を行うと，端末は利用者がルート権限を使用できない状態に戻る．そのため，再びルート権限を利用するには，再度不正な権限昇格を行う必要がある．

3.2. 恒久的ルート化

一時的ルート化の状態から，本来書き込みできないファイルを改竄することで，恒久的にルート権限を利用できる状態にすることを恒久的ルート化と呼ぶ．この恒久的ルート化の状態にするためには，一時的ルート化状態で，system ディレクトリを読み出し専用モードから読み出し及び書き込みモードに変更し，リマウントを行う．また，一般アプリがルート権限に切り替えられる環境にするため，本来用意されてないルート権限に切り替える su コマンドの追加とルート権限を使用するアプリを管理するアプリの追加を行う必要がある．

4. ルート化に至るまでの手法

一般にルート化する手法は大きく分けて 2 種類存在する．一つは端末がもつセキュリティホールを悪用する exploit コードをインジェクションベクタとする方法である．もう一つはブートローダアンロックを利用して行う手法である．ブートローダアンロックとは，一部のベンダ側から提供されるツールに含まれる一つの機能である．目的は，ベンダ側が提供したツールの機能を使用した端末をサポート対象外とし，ルート化による端末の故障を，利用者の合意の上で利用者側の責任とすることである．

4.1. 脆弱性を用いたルート化手法

攻撃に利用される Android の脆弱性は分類すると 2 種類ある．それらは，Android が持つ脆弱性と端末ベンダの個別拡張における設計ミスもしくはセキュリティホール（バグ）である．とりわけ，確保されてないメモリの参照やファイルシステムの権限設定ミスなどがある．攻撃者は，これらの脆弱性を利用して不正な権限昇格を行う．

4.2.1 シンボリックリンク攻撃を用いたルート化手法

Android には，ADB（Android Debug Bridge）と呼ばれるデバッグモードがある．通常，ADB ではルート権限を使用することはできないが，接続端末がエミュレータであれば，ルート権限を使用できるようになっている．シンボリックリンク攻撃を利用して，起動時に読み込む設定ファイル（/data/local.prop）にエミュレータ判別に使われる設定値を書き込むことにより，接続端末をエミュレータと判断させることで，ADB 接続における Shell をルート権限で使用できる．

4.2. ブートローダアンロックを用いたルート化手法

Android におけるブートローダとは，システムを立ち上げるときに使用されるソフトウエアである．ブートローダは RAM 上に作られたシステムイメージを読み込み，Linux カーネルを起動させる．

通常，商業販売されている Android 端末のブートローダはロック状態にある．ロック状態とは，ベンダに提供されたブートローダがディジタル署名で保護されたファイルのみを読み込めるなどの状態である．そこで，このロック状態を解除すれば，任意のファイルを読み込むことができるようになる．よって，攻撃者が改変したシステムファイルやカーネルをロードさせることで，ルート化するなどを行う．

5. まとめ

本論文では，Android におけるルート化に対するセキュリティとルート化に至るまでの手法に関して調査し，いくつかの手法について取り上げた．

6. 参考文献

[1] Android Developer
http://developer.android.com/index.html
[2] 齋藤孝道著，マスタリング TCP/IP 情報セキュリティホール編