Vol.2012-DPS-150 No.19
Vol.2012-CSEC-56 No.19
2012/2/29

HTTP トラフィックを利用したクラスタリングによる Android アプリケーションの分類

葛野 弘樹[†1]

Android アプリケーションにおいて，ユーザの意図しない通信として，広告配信や利用統計の取得を目的とした情報収集モジュールなどによる携帯端末の識別情報や位置情報といった端末上にあるユーザに関するセンシティブな情報の外部送信が問題視されている．この問題を解決するために，Android アプリケーションの通信を取得し，HTTP パケット送信先と HTTP パケット送信内容の類似度を利用し HTTP トラフィックおよびアプリケーションを階層的クラスタリングにより分類する手法を提案する．提案手法を，Android アプリケーション 131 個の通信ログに適用し，アプリケーションの分類と HTTP トラフィックの分類を行った．分類結果より，各クラスタに含まれる端末情報の送信傾向がユーザの意図しない通信を行うアプリケーションかの判定に利用できることを確認した．

Clustering of Android Application using HTTP Traffic

Hiroki Kuzuno[†1]

An Android's application includes advertisement modules which collect user's privacy-sensitive information and transmit it across the network. That's information will be used for targeted advertisements or user behavior statistical. However such application's behavior is not intended by user. To address these problem, we propose a new classification technique that describes applications network behavior in terms of HTTP packet destination and HTTP packet content. We provide a method for hierarchical clustering these HTTP traffic or applications into cluster that reflect similar network behaviors. Our proposed method is empirically evaluated on a dataset of 131 android's applications network flows. Results from our evaluation, which shows that our method can group a similar HTTP packet including privacy-sensitive information for detecting user's unintended leakage by application.

1. はじめに

携帯端末を中心に Google Inc. の開発する Android を搭載した機器が急速に普及している．Android では，第三者の開発した Android アプリケーション（以下，アプリケーションとする）を Android Market を通じて利用者に公開することができ，多数のアプリケーションが提供されている．

携帯端末は，ユーザが個人の専用機器として利用することから端末の識別情報や位置情報などのユーザに関するセンシティブな情報（以下，端末上の情報とする）が格納される．そのため，Android では，セキュリティとプライバシーを確保するための機構として，アプリケーション毎へのユーザ ID(UID) の割り当てによるサンドボックス化と端末上の情報へのアクセスやアプリケーション連携を制御するパーミッションフレームワークを備えている．しかし，アプリケーションによる広告配信や利用の統計取得を目的とした情報送信[12),20),26)]，DroidDream を始めとしたマルウェア[22)]，端末やアプリケーションの不備による危険性[5),17)] などの問題が指摘されている．

Android では，ユーザにアプリケーションのインストール時にパーミッションを確認することでアプリケーションがアクセスできる端末上の情報を把握できるが，アプリケーションが端末上の情報をどのような用途で用いるかは把握できないことから，アプリケーションによるユーザの意図しない端末上の情報送信が行われる可能性がある．そのため，ユーザの意図しない通信を行うアプリケーションか判断するためにアプリケーションの通信がユーザの意図した通信あるいは意図しない通信か把握することが重要な課題となる．

本稿では，そのような課題を解決するために HTTP トラフィックを利用した階層的クラスタリングによるアプリケーションおよび HTTP トラフィックの分類手法を提案する．提案手法をアプリケーション 131 個から取得した通信ログに適用し，同一あるいは類似性の高い通信を行うアプリケーションや HTTP トラフィックへ分類した．分類結果より，特定のクラスタに属する HTTP トラフィックに端末上の情報が多く含まれており，そのような HTTP トラフィックは，広告配信や利用統計を目的とした通信である可能性が高いことから，ユーザの意図しない通信を行うアプリケーションかの判断に利用できることを確認できた．

†1 セコム株式会社 IS 研究所
Intelligent Systems Laboratory, SECOM Co., Ltd.

Vol.2012-DPS-150 No.19
Vol.2012-CSEC-56 No.19
2012/2/29

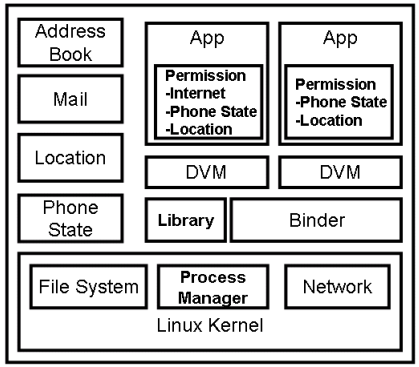

図 1 Android アーキテクチャ

2. 背景知識

2.1 Android のアーキテクチャ

Android のアーキテクチャを**図 1** に示す．Android は Linux カーネル，ミドルウェアに Binder，Library，Dalvik Virtual Machine（DVM），アプリケーション，位置情報などの端末上の情報から構成される． Linux カーネルは，プロセス管理，ファイルシステム，ネットワーク機能を上位レイヤに提供する．ミドルウェアは，アプリケーションを動作させる DVM，端末機能や端末上の情報へのアクセスのためのライブラリ，プロセス間通信のための Binder を提供する．アプリケーションは Java，C/C++で記述され，アプリケーション毎に保有するパーミッションおよびサンドボックス化において許可された権限の範囲において動作する．

2.2 パーミッションフレームワーク

Android は，アプリケーションによる外部とのネットワーク通信や端末上の情報へのアクセスなど端末リソースの利用をパーミッションにより管理し，アプリケーションがアクセスを要求した際に Binder において付与されているパーミッションとアクセス対象を確認し，アクセス可否を判断することでアクセス制御を行う．Android API Level 15 におけるパーミッションは 124 個[16] あり，端末上の情報へアクセスするためのパーミッションが多数定義されている．また，パーミッションは，開発者が独自に定義することができ，アプリケーション間通信を用いて連携する際のアクセス制御にも利用される．

3. Android アプリケーションの課題

3.1 アプリケーションの要求するパーミッション

Android では，ユーザはアプリケーションのインストール時にパーミッションを確認することで，アプリケーションが実行中に利用可能となる端末リソースを把握できる．しかし，アプリケーションがどのような用途で端末リソースを必要とするのかは把握できず，ユーザはパーミッションから用途を推測し，利用可否を判断しなければならない．

アプリケーションが要求するパーミッションとしてはネットワーク通信が最も多く[2]，ネットワーク通信と端末上の情報取得のパーミッションの組み合わせを要求しているアプリケーションも多く存在している[30]．ネットワーク通信と端末上の情報を取得するパーミッションの組み合わせを要求するアプリケーションでは，端末上の情報を外部に送信される可能性があるが，ユーザはアプリケーションの実行中に端末上の情報が外部に送信された場合でも確認することはできない．

3.2 アプリケーションによる情報送信

ネットワーク通信と端末上の情報取得のパーミッションを要求するアプリケーションによる端末上の情報の外部への送信が指摘されている[12),20]．これらのアプリケーションの多くは無料提供されており，広告収入や利用統計の提供による収益のために端末上の情報を送信していると考えられる．端末上の情報送信は，アプリケーションが内包する情報収集モジュール[29] と呼ばれる広告配信や利用統計取得を目的として提供されているライブラリにより行われている．

アプリケーション内の情報収集モジュールは，アプリケーションに付与されたパーミッションの範囲で端末リソースを利用することができる．そのため，情報収集モジュールの開発者に端末上の情報などを送信することも可能である．複数のアプリケーションが同一の情報収集モジュールを含んでいる場合は，各アプリケーションからの端末上の情報送信によりユーザの利用アプリケーション一覧や利用時間を容易に把握される恐れがある．

3.3 アプリケーションの通信内容の調査

我々は，Android マーケットの日本向けランキング上位よりネットワーク通信と端末上の情報取得のパーミッションを要求するアプリケーションを 131 個選び，端末上で実行した際にアプリケーションより送信された HTTP パケット 5477 個について調査した．調査対象とした HTTP パケットは，アプリケーションからの情報送信となる GET/POST メソッドであり，調査環境としては，Nexus S, Android 2.3.4 を用いた．

Vol.2012-DPS-150 No.19
Vol.2012-CSEC-56 No.19
2012/2/29

表 1 に HTTP パケット送信先毎の送信回数およびアプリケーション数のいずれもが多い HTTP パケット送信先を，**図 2** にアプリケーションあたりの HTTP パケット送信先数の累積度数，累積分布，**表 2** に端末上の情報の送信回数およびアプリケーション数を示す．なお，端末上の情報は，携帯機器に付与される IMEI，携帯加入者に付与され SIM カードに格納されている IMSI，SIM カードに付与される Sim Serial ID，通信キャリア名称，Android が初期化時に生成する Android ID，さらに調査対象のアプリケーションの多くで見られた ISU と指定された値とした．

表 1 HTTP パケット送信先毎の送信回数とアプリケーション数

HTTP Host Destination	# Packets	# Apps
admob.com	254	38
ad-maker.info	243	16
gstatic.com	197	42
google.com	147	17
amazonaws.com	121	7
fbcdn.net	95	7
adwhirl.com	83	15
doubleclick.net	78	30
mediba.jp	68	7
mydas.mobi	65	7
adimg.net	44	14
google-analytics.com	43	9
microad.jp	29	9
mobclix.com	28	6
adlantis.jp	27	16
flurry.com	25	17
googlesyndication.com	18	8

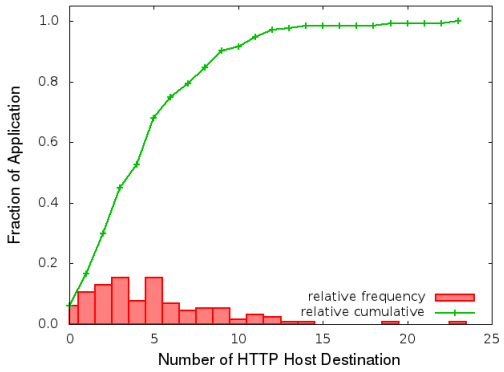

図 2 アプリケーションあたりの HTTP パケット送信先の累積度数，累積分布

表 1 より，多くのアプリケーションが同一の送信先へ HTTP パケットを送信していることを確認できた．送信先のドメイン名より広告配信，利用統計およびサードパーティのサービスを提供していると推測できる．さらに，アプリケーションの開発者が異なる場合において，送信先が同一となるのは，サードパーティの提供する API を利用している場合，アプリケーションに内包された情報収集モジュールを利用する場合であると考えられる．

図 2 より，アプリケーションあたりの HTTP パケット送信先数は 5 個以下が 70%程度，14 個以下が 90%程度となった．調査対象のアプリケーションの殆どが複数の宛先に HTTP パケットを送信していることを確認できた．調査対象のアプリケーションにおける平均 HTTP パケット送信先は 5.8 個，最も HTTP パケット送信先の多いアプリケーションで 23 個である．なお，アプリケーションの HTTP パケット送信先数は，HTTP パケットの送信回数を表していないため，送信先が少ない場合でも送信回数が多いアプリケーションが存在する可能性はある．

表 2 より，端末上の情報のうち，IMEI，通信キャリア名称，Android ID，ISU は多くのアプリケーションにより送信されていることを確認できた．IMSI および SIM Serial ID を送信するアプリケーションは 2 個のみであった．

調査結果より，調査対象アプリケーションの多くで端末上の情報を外部へ送信しており，他のアプリケーションと同一の送信先と通信していることも判明した．

4. 提案手法

4.1 HTTP トラフィックを利用したクラスタリングによるアプリケーション分類

アプリケーションによる端末上の情報の送信と複数のアプリケーションで同一の通信先が

表 2 端末上の情報の送信回数とアプリケーション数

Sensitive Information	# Packets	# Apps
IMEI (Device ID)	201	35
IMSI (Subscriber ID)	24	2
SIM Serial ID	24	2
Carrier	146	34
Android ID	221	48
ISU	511	46

Vol.2012-DPS-150 No.19
Vol.2012-CSEC-56 No.19
2012/2/29

確認できたことは，アプリケーションに内包された情報収集モジュールなどにより端末上の情報の収集が行われている可能性を示している．アプリケーションによる端末上の情報の送信は，その送信内容と送信先についてユーザが事前に同意していなければ，ユーザの意図しない通信といえる．ユーザが端末上の情報送信を防ぐためには，アプリケーションの利用にあたり，インストール時に提示されるパーミッションのみでユーザの意図しない通信を行うアプリケーションかどうかを判断しなければならず困難といえる．そのため，アプリケーションの通信がユーザの意図した通信または意図しない通信か識別し，ユーザの意図しない通信を行うアプリケーションと判断することが重要となる．

本稿では，アプリケーションの通信をユーザの意図しない通信かどうか把握し，ユーザの意図しない通信を行うアプリケーションかどうかの判断に利用するために，HTTP トラフィックに含まれる HTTP パケット送信先と HTTP パケット送信内容の類似度に基づき，アプリケーションの HTTP トラフィックおよびアプリケーション毎にクラスタリングして分類する手法を提案する．

4.2 提案手法の概要

提案手法では，アプリケーションの送信した HTTP パケットの送信先と HTTP パケットの送信内容の類似度を計算し，その類似度を基に階層的クラスタリングを適用し HTTP トラフィックおよびアプリケーションを分類する．アプリケーションによる情報の送信内容だけでなく，通信先に関してもアプリケーションおよび HTTP トラフィックを分類する際の重要な指標となると考え，HTTP パケットの送信先と HTTP パケット送信内容毎の類似度を用いて階層的クラスタリングを行う．類似度の計算および階層的クラスタリングの概要を以下に示す．

- `HTTP` パケット送信先の類似度： HTTP パケットの送信先 IP アドレス，Port 番号，HTTP ヘッダフィールドの Host ヘッダ（以下，Host ヘッダとする）を用いて，HTTP パケット間の送信先の類似度を求める．類似度として，送信先 IP アドレスは IP アドレスの上位ビットの最長一致，Port 番号は値が同一かどうか，HTTP ヘッダフィールドの Host ヘッダは文字列の距離を表す編集距離を適用し，各要素の類似度より HTTP パケット送信先の類似度とする．
- `HTTP` パケット送信内容の類似度：HTTP ヘッダフィールドの Request-Line，Cookie，Message-Body を用いて，HTTP パケット送信内容の類似度を求める．要素毎に，コルモゴルフ複雑性を利用した normalized compression distance (NCD)[7] により類似度を求め，各要素の類似度より HTTP パケット送信内容の類似度とする．
- 階層的クラスタリング： 分類対象となるアプリケーションが送信した HTTP パケットに対し，アプリケーション単位または HTTP トラフィック単位でクラスタを構成する．そして，クラスタ間の距離として HTTP パケット送信先，HTTP パケット送信内容の類似度を用い，距離の近いクラスタ毎に階層的クラスタリングを行う．クラスタリング手法は，群平均法を用いる．

4.3 HTTP パケット送信先の類似度

HTTP パケット送信先の類似度は，送信先 IP アドレス，Port 番号，Host ヘッダの類似度を用いる．二つの HTTP パケット p_x と HTTP パケット p_y における HTTP パケット送信先の類似度 $d_{dst}(p_x, p_y)$ を次のように定義する．

$$d_{dst}(p_x, p_y) = d_{ip}(p_x, p_y) + d_{port}(p_x, p_y) + d_{host}(p_x, p_y) \quad (1)$$

任意の HTTP パケット p_n の送信先の構成は $p_n = \{ip_n, port_n, host_n\}$ とし，ip_n を HTTP パケットに含まれる送信先 IP アドレス，$port_n$ を Port 番号，$host_n$ を Host ヘッダとする．なお，送信先 IP アドレスは IPv4 とする．HTTP パケット送信先の類似度に用いる送信先 IP アドレス，Port 番号，Host ヘッダの類似度を以下に示す．

- 送信先 `IP` アドレスの類似度： IP アドレスは，2^{32} のアドレス空間であり，上位から 8 ビット毎に，IP アドレスの範囲となる．IP アドレスは一定の範囲ごとに利用者に割り当てられることから，同一組織あるいはネットワークの場合は，上位ビットが共通となる可能性がある．そこで，HTTP パケット p_x，p_y において IP アドレスの類似度を $d_{ip}(p_x, p_y) = 1 - lmatch(ip_x, ip_y)/32$ と定義する．$lmatch$ は二つ IP アドレスの先頭からの最長一致ビット数を返す関数とする．$d_{ip}(p_x, p_y)$ のとりうる値の範囲は 0 から 1 である．
- `Port` 番号の類似度： Port 番号は，2^{16} の空間であり，一部の Port 番号は，特定のサービスに対し登録され使用されることが殆どである．HTTP パケット p_x，p_y において Port 番号の類似度を $d_{port}(p_x, p_y) = match(port_x, port_y)$ と定義する．$match$ は二つの Port 番号が一致すれば 0 を一致しなければ 1 を返す関数とする．
- `Host` ヘッダの類似度： Host ヘッダは，FQDN として，ホスト名とドメイン名から構成される文字列である．任意の二つの文字列の距離として編集距離を用いることとし，HTTP パケット p_x，p_y において Host ヘッダの類似度を $d_{host}(p_x, p_y) = ed(host_x, host_y)/max(len(host_x), len(host_y))$ と定義する．ed は二つの Host ヘッダの編集距離を返す関数，len は文字列長を返す関数，max は最大値を返す関数とする．$d_{host}(p_x, p_y)$ のとりうる値の範囲は 0 から 1 である．

Vol.2012-DPS-150 No.19
Vol.2012-CSEC-56 No.19
2012/2/29

4.4 HTTP パケット送信内容の類似度

HTTP パケット送信内容の類似度は，HTTP ヘッダフィールドの Request-Line，Cookie，Message-Body の類似度を用いる．二つの HTTP パケット p_x と HTTP パケット p_y における HTTP パケット送信内容の類似度 $d_{header}(p_x, p_y)$ を次のように定義する．

$$d_{header}(p_x, p_y) = d_{rline}(p_x, p_y) + d_{cookie}(p_x, p_y) + d_{body}(p_x, p_y) \quad (2)$$

任意の HTTP パケット p_n の送信内容の構成は $p_n = \{rline_n, cookie_n, body_n\}$ とし，$rline_n$ を HTTP パケットに含まれる Request-Line，$cookie_n$ を Cookie，$body_n$ を Message-Body とする．HTTP パケット送信内容の類似度に用いる Request-Line，Cookie，Meesage-Body の類似度を以下に示す．

- `Request-Line`，`Cookie`，`Message-Body` の類似度：Request-Line，Cookie，Message-Body はいずれも文字列で構成され，その値はアプリケーションやサービス毎に異なるため，コルモゴルフ複雑性を利用し，文字列の文脈に依存することなく情報量としての距離を求めることが可能な NCD を用いることとした．任意の二つの文字列への NCD は以下の式で求められる．

$$ncd(x, y) = \frac{C(xy) - min(C(x), C(y))}{max(C(x), C(y))} \quad (3)$$

$C(x)$ は文字列 x を圧縮し，その長さを返す関数とする．そして，HTTP パケット p_x，p_y において Request-Line，Cookie，Message-Body の各類似度を $d_{data}(p_x, p_y) = ncd(data_x, data_y)$ と定義する．$d_{data}(p_x, p_y)$ のとりうる値の範囲は 0 から 1 である．

4.5 階層的クラスタリング

アプリケーションおよび HTTP トラフィックの階層的クラスタリングは，クラスタの距離は群平均法と HTTP パケット送信先の類似度，HTTP パケット送信内容の類似度に基づいた HTTP パケットの距離を用いる．二つの HTTP パケット p_x と HTTP パケット p_y の距離 $d_{pkt}(p_x, p_y)$ を次のように定義する．

$$d_{pkt}(p_x, p_y) = d_{dst}(p_x, p_y) + d_{header}(p_x, p_y) \quad (4)$$

二つのクラスタ C_x と C_y の距離は，群平均法を用い以下の式で求める．

$$d_{group}(C_x, C_y) = \frac{1}{|C_x||C_y|} \sum_{p_x \in C_x} \sum_{p_y \in C_y} d_{pkt}(p_x, p_y) \quad (5)$$

N 個のアプリケーションの集合を $\mathbf{A} = \{a_i\}_{i=1..N}$，任意のアプリケーション a_i が含む M 個の HTTP パケットの集合を $\mathbf{H}(a_i) = \{p_j\}_{j=1..M}$ とした場合，アプリケーションは HTTP パケットの集合であることから，HTTP パケットに対する階層的クラスタリングの手順を以下に示す．

- 手順 1： クラスタ C_i を HTTP パケット p_i のクラスタとし，クラスタの集合を $\mathbf{C} = \{C_i\}_{i=1..M*N}$ とする．
- 手順 2： 任意のクラスタ $C_x \in \mathbf{C}$ とクラスタの集合に含まれる他のクラスタ $C_y \in \mathbf{C}, x \neq y$ に全てにおいてクラスタ間の距離 $d_{group}(C_x, C_y)$ を求める．
- 手順 3： クラスタ C_x と最も近いクラスタ C_y からなるクラスタを新たなクラスタ $C_z = \{C_x, C_y\}$ としてクラスタの集合に加え，クラスタ C_x, C_y をクラスタの集合から取り除く．
- 手順 4： クラスタ集合内のクラスタ数が 1 つになるまで，手順 2，3 を繰り返す．

5. 実験結果と評価

ネットワーク通信と端末上の情報取得のパーミッションを要求するアプリケーションの送信した HTTP パケットに対し，提案手法を用いてアプリケーションおよび HTTP トラフィックの分類を行った．実験には，3.3 節にて送信情報を調査したアプリケーション 131 個の送信した HTTP パケット 5477 個を用いた．なお，HTTP トラフィックの分類では，アプリケーションにおいて Host ヘッダが同一となる HTTP パケットを事前に集約し，提案手法による分類を行うこととした．

表 3，**表 4** に各層におけるクラスタ数とクラスタ間の距離を示す．クラスタ間の距離は小さいほど，クラスタが類似している事を表す．そのため，クラスタ間の距離の値は，評価対象の単位および階層的クラスタリングのどの層までが有効な分類であるかの指標となる．また，**表 5**，**表 6** に各層における端末上の情報の最大 F 値を示す．F 値は Precision（適合率），Recall（再現率）を組み合わせた評価尺度であり，F 値が高いほど，情報を網羅していることとノイズが少ないと言える．F 値は以下の式より求めた．

$$\text{Precisioin} = \frac{\text{\# of HTTP packet including sensitive data in the cluster}}{\text{\# of all HTTP packet}} \quad (6)$$

$$\text{Recall} = \frac{\text{\# of HTTP packet including sensitive data in the cluster}}{\text{\# of all HTTP Packet including sensitive data}} \quad (7)$$

$$\text{F-measure} = \frac{2}{(1/\text{Recall} + 1/\text{Precision})} \quad (8)$$

提案手法によるアプリケーションおよび HTTP トラフィックの階層的クラスタリングの

Vol.2012-DPS-150 No.19
Vol.2012-CSEC-56 No.19
2012/2/29

クラスタ数は，アプリケーションで 245 個，HTTP トラフィックで 9643 個となり，HTTP トラフィックの分類の方がクラスタ数が多くなった．これは，HTTP パケットを事前に集約する単位の違いによる．提案手法における階層的クラスタリングの計算量は，クラスタ数に依存するため，クラスタ数が多いほど処理に時間を要する．

表 3，表 4 より，階層的クラスタリングにおける各層でのクラスタ間の距離は，アプリケーションの分類においては 4 層，HTTP トラフィックの分類においては 5 層までは最小値が比較的小さく，クラスタを構成する HTTP パケットが近いことが分かる．

表 5，表 6 より，F 値が大きい場合，端末上の情報の各項目を含んだ HTTP パケットが階層的クラスタリングのどの層で最も網羅されかつ各項目を含まない HTTP パケットが少ないかが分かる．アプリケーションの分類では，5 層において IMEI，キャリアの F 値が，6 層において，Android ID，ISU の F 値が最も大きい．HTTP トラフィックの分類では，7 層において IMEI，ISU，9 層においてキャリア，7 または 13 層において Android ID の F 値が最も大きい．HTTP トラフィックの分類では，7 層までは，各項目の F 値がいずれも高い値となっており，端末上の情報を含む HTTP パケットのうち類似度の高い HTTP パケットがクラスタリングされていると言える．

アプリケーションの分類および HTTP トラフィックの分類結果より，HTTP パケットの集約単位の違いが，提案手法による階層的クラスタリングに大きく影響することを確認できた．階層的クラスタリングにおける各層でのクラスタ間の距離，クラスタにおける端末上の情報の最大 F 値のいずれでも HTTP トラフィックの分類はアプリケーションの分類よりも良い結果を得られている．提案手法を用いた階層的クラスタリングにより端末上の情報を送信するアプリケーションを分類するには，HTTP トラフィックの分類から得られた各クラスタに含まれる HTTP パケットから，該当する HTTP パケットを送信したアプリケーションを把握することで効果的に分類できると考えられる．しかしながら，HTTP トラフィックの分類では，クラスタ数により計算コストが増加する可能性がある．そのため，端末上の情報を送信する既知のアプリケーションも含めアプリケーションの分類を行い，既知のアプリケーションのクラスタとの距離から端末上の情報送信の有無を把握する必要がある．

6. 関連研究

Android や iOS のアプリケーションにおいて，アプリケーションの静的解析，動的解析，また端末の備える機能を利用した端末上の情報の外部送信によりアプリケーションによる情報漏洩の可能性が指摘されている[10),12),15),20),26)]．Android アプリケーションによる情報漏洩を検出し防ぐ試みとしては，Android のパーミッションフレームワークを拡張する手法[23)]，アプリケーションへの細粒度の高いアクセス制御を実現する手法[13),24)] や Android フレームワークにおいて，アプリケーションの処理する情報のフローを監視し情報漏洩を検出する手法[11)] が提案されている．しかし，アプリケーションや端末の実装不備などから，他のアプリケーションの情報フローを盗聴される可能性[5)] やパーミッションが悪用される危険性も指摘されており[9)]，アプリケーション間におけるプロセス間通信を監視する手法[14)]，端末の実装を調査する手法が提案されている[17)]．

計算機で動作するアプリケーションの構造，振る舞い，ネットワークトラフィックの類似度を利用しクラスタリングにより特徴を把握しアプリケーションを分類する手法は，マルウェアやボットネットの検出に用いられている[1),3),6),8),19),21),28)]．また，より細かくネット

表 3 アプリケーションの分類の各層におけるクラスタ数とクラスタ間の距離

Clustering Rank	Application			
	# Cluster	Min	Max	Avg
1	1	1702.571	1702.571	1702.572
2	2	348.376	2268.065	1308.221
3	4	108.5926	2135.405	800.186
4	8	42.082	1684.958	441.467
5	16	16,261	1124.349	230.937
6	32	1.028	431.450	85.837
7	62	0.303	183.609	34.280

表 4 HTTP トラフィックの分類の各層におけるクラスタ数とクラスタ間の距離

Clustering Rank	HTTP Traffic			
	# Cluster	Min	Max	Avg
1	1	3662.566	3662.566	3662.566
2	2	1656.513	2236.767	1946.640
3	4	722.731	1207.793	881.114
4	8	268.819	625.78	402.898
5	16	51.982	313.032	181.685
6	32	20.665	152.946	75.725
7	64	1.803	75.481	30.215
8	128	0.741	36.384	11.655
9	256	0.286	18.764	4.368
10	512	0.123	10.078	1.61
11	1024	0.03	4.911	0.604
12	2048	0.01	2.478	0.202
13	4012	0.01	1.277	0.111

Vol.2012-DPS-150 No.19
Vol.2012-CSEC-56 No.19
2012/2/29

ワーク上の振る舞いを把握するためにアプリケーションの通信先と通信量を個別にクラスタリングする手法[18]，HTTP パケットの統計値と文字列の類似度を個別にクラスタリングし，シグネチャを生成する手法[25] も提案されている．

先行研究では，Android アプリケーションの動的解析，静的解析を行い端末上の情報送信の有無を調査しているが，アプリケーションの類似度による分類は行われていない．提案手法では，アプリケーションにおける情報収集モジュールなどの特性に基づき，HTTP トラフィックを送信先と送信内容に分けクラスタリングを行った．クラスタリングを用いた未知の振る舞いや通信の先行研究においても，検出精度をより向上させるために，調査対象とするアプリケーションから得られる情報の構造と特性を考慮した類似度を利用しており，その特性に対しクラスタリングを適用することが重要といえる．なお，本稿で調査した Android アプリケーションによる端末上の情報送信は，情報収集モジュールによる広告配信，利用統計であると考えられる．収集された情報は，ユーザの振る舞いや環境を追跡し効果的な広告を行うターゲット広告に使用される可能性がある．ターゲット広告については，ユーザのプライバシーを考慮した広告手法も提案されているが，普及するには至っていない．[4),27)]．

表 5 アプリケーションの分類の各層における端末上の情報の最大 F 値

Clustering Rank	Application			
	IMEI	Carrier	Android ID	ISU
1	0.065	0.03	0.034	0.065
2	0.109	0.043	0.051	0.247
3	0.201	0.108	0.081	0.209
4	0.248	0.171	0.173	0.239
5	0.247	0.204	0.203	0.338
6	0.280	0.281	0.193	0.263
7	0.224	0.387	0.135	0.21

表 6 HTTP トラフィックの分類の各層における端末上の情報の最大 F 値

Clustering Rank	HTTP Traffic			
	IMEI	Carrier	Android ID	ISU
1	0.067	0.03	0.034	0.065
2	0.079	0.046	0.036	0.104
3	0.084	0.063	0.068	0.159
4	0.111	0.067	0.127	0.336
5	0.17	0.13	0.246	0.412
6	0.272	0.224	0.401	0.574
7	0.377	0.326	0.557	0.827
8	0.318	0.478	0.413	0.566
9	0.318	0.603	0.326	0.329
10	0.172	0.355	0.181	0.179
11	0.09	0.195	0.108	0.094
12	0.046	0.102	0.084	0.048
13	0.065	0.1	0.571	0.028

7. ま と め

本稿では，Android アプリケーション 131 個の通信ログより HTTP パケットの送信先および端末上の情報の送信有無を調査し，HTTP トラフィックに含まれる HTTP パケット送信先と HTTP パケット送信内容の類似度に基づきアプリケーションおよびアプリケーションの HTTP トラフィック毎に階層的クラスタリングにより分類する手法を提案した．また，調査対象としたアプリケーションに提案手法を適用し，アプリケーションの分類と HTTP トラフィックの分類を行った結果，端末上の情報を送信しているアプリケーションの把握には，HTTP トラフィックの分類が適していることを確認した．

提案手法により，アプリケーションから取得した通信ログから同一あるいは類似性の高い通信を行うアプリケーションや HTTP トラフィックへ分類できることが確認できた．これにより，端末上の情報送信を行うアプリケーションが分類対象に含まれていた場合や通信先ごとにどのようなアプリケーションが分類されたかを容易に把握でき，ユーザの意図しない通信を行うアプリケーションかの判断に利用できる．

今後は，より大量のアプリケーションの通信ログに対し提案手法の有効性を検証する予定である．さらに，HTTP トラフィック以外にアプリケーションによるユーザの意図しない通信を把握するための指標を調査し，アプリケーションの類似度を測るために利用可能であるかの検討を進めていきたい．

参 考 文 献

1) Bailey, M., Oberheide, J., Andersen, J. and Mao, Z.M.: Automated Classification and Analysis of Internet Malware, *Symposium on Recent Advances in Intrusion Detection (RAID)* (2007).
2) Barrera, D., Kayacik, H.G., van Oorschot, P. and Somayaji, A.: A Methodology for Empirical Analysis of Permission-Based Security Models and its Application to Android, *17th ACM Conference on Computer and Communicatons Security (CCS 2010)* (2010).
3) Bayer, U., Comparetti, P.M., Hlauschek, C., Krugel, C. and Kirda, E.: Scal-

Vol.2012-DPS-150 No.19
Vol.2012-CSEC-56 No.19
2012/2/29

able, Behavior-Based Malware Clustering, *Network and Distributed System Security Symposium (NDSS 2009)* (2009).
4) Bilenko, M., Richardson, M. and Tsai, J.Y.: Targeted, Not Tracked: Client-side Solutions for Privacy-Friendly Behavioral Advertising, *The 11th Privacy Enhancing Technologies Symposium (PETS 2011)* (2011).
5) Chin, E., Felt, A.P., Greenwood, K. and Wagner, D.: Analyzing Inter-Application COmmunication in Android, *The 9th International Conference on Mobile Systems (Mobisys 2011)* (2011).
6) Chung, J.Y., Park, B., J.Won, Y., Strassner, J. and Hong, J.W.: Traffic Classification Based on Flow Similarity, *IP Operations and Management, 9th IEEE International Workshop (IPOM 2009)* (2007).
7) Cilibrasi, R.: Statistical Inference Through Data Compression, PhD Thesis, Amsterdam Universtity, Amsterdam (2007).
8) Coull, S.E., Monrose, F. and Bailey, M.: On Measuring the Similarity of Network Hosts: Pitfalls, New Metrics, and Empirical Analyses, *Network and Distributed System Security Symposium (NDSS 2011)* (2011).
9) Davi, L., Dmitrienko, A., Sadeghi, A.-R. and Winandy, M.: Privilege Escalation Attacks on Android, *Proceedings of the 13th international conference on Information security (ISC 2010)* (2010).
10) Egele, M., Krugel, C., Kirda, E. and Vigna, G.: PiOS: Detecting Privacy Leaks in iOS Applications, *Network and Distributed System Security Symposium (NDSS 2011)* (2011).
11) Enck, W., Gilbert, P., Chun, B.-G., Cox, L.P., Jung, J., McDaniel, P. and Sheth, A. N.: TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones, *9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2010)* (2010).
12) Enck, W., Octeau, D., McDaniel, P. and Chaudhuri, S.: A Study of Android Application Security, *20th USENIX Security Symposium 2011* (2011).
13) Enck, W., Ongtang, M. and McDaniel, P.: On Lightweight Mobile Phone Application Certification, *16th ACM Conference on Computer and Communicatons Security (CCS 2009)* (2009).
14) Felt, A.P., Wang, H.J., Moshchuk, A., Hanna, S. and Chin, E.: Permission Re-Delegation: Attacks and Defenses, *20th USENIX Security Symposium 2011* (2011).
15) Gilbert, P., Chun, B.-G., Cox, L.P. and Jung, J.: Automated Security Validation of Mobile Apps for App Markets, *Mobile Cloud Computing and Services (MCS 2011)* (2011).
16) Google: Android Developers Manifest.permission, Google. Inc.
17) Grace, M., Zhow, Y., Wang, Z. and Jiang, X.: Systematic Detection of Capability Leaks in Stock Android Smartphones, *Network and Distributed System Security Symposium (NDSS 2012)* (2012).
18) Gu, G., Perdisci, R., Zhang, J. and Lee, W.: BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Indepentent Botnet Detection, *17th USENIX Security Symposium 2008* (2008).
19) Gu, G., Zhang, J. and Lee, W.: BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic, *Network and Distributed System Security Symposium (NDSS 2008)* (2008).
20) Hornyack, P., Han, S., Jung, J., Schechter, S. and Wetherall, D.: These Aren't the Droids You're Looking For: Retrofitting Android to Protect Data from Imperious Applications, *18th ACM Conference on Computer and Communicatons Security (CCS 2011)* (2011).
21) Ingham, K.L. and Inoue, H.: Comparing Anomaly Detection Techniques for HTTP, *Symposium on Recent Advances in Intrusion Detection (RAID)* (2007).
22) Lookout: Mobile Threat Report August 2011, Lookout Mobile Security.
23) Nauman, M., Khan, S., Alam, M. and Zhang, X.: Apex: Extending Android Permission Model and Enforcement with User-defined Runtime Constraints, *ACM Symposium on Information, Computer and Communications Security (ASIACCS 2009)* (2009).
24) Ongtang, M., McLaughlin, S., Enck, W. and McDaniel, P.: Semantically Rich Application-Centric Security in Android, *The 25th Annual Computer Security Applications Conference (ACSAC 2009)* (2009).
25) Perdisci, R., Lee, W. and Feamster, N.: Behavioral Clustering of HTTP-Based Malware and Signature Generation Using Malicious Network Traces, *7th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2010)* (2010).
26) Schlegel, R., Zhang, K., Zhou, X., Intwala, M., Kapadia, A. and Wang, X.: Soundcomber: A Stealthy and Context-Aware Sound Trojan for Smartphones, *Network and Distributed System Security Symposium (NDSS 2011)* (2011).
27) Toubiana, V., Narayanan, A., Boneh, D., Nissenbaum, H. and Barocas, S.: Adnostic: Privacy Preserving Targeted Advertising, *Network and Distributed System Security Symposium (NDSS 2010)* (2010).
28) Wehner, S.: Analyzing Worms and Network Traffic using Compression, *Journal of Computer Security*, Vol.15, No.3, pp.303–320 (2007).
29) JSSEC アプリケーション WG：スマートフォンにおけるマルウェア対策と、安全なアプリケーションの配布，日本スマートフォンセキュリティフォーラム (JSSEC)（オンライン），入手先⟨http://www.jssec.org/news/20111107.html⟩（参照 2012-01-28）.
30) 葛野弘樹：Android アプリケーションに対する情報フロー制御機構の提案，コンピュータセキュリティシンポジウム (CSS 2011) (2011).