Webトラッキング検知システムの構築とサードパーティトラッキングサイトの調査

芳賀 夢久[1]　高田 雄太[2,1]　秋山　満昭[2]　森 達哉[1]

概要：Web サイト上のユーザーの行動を分析する Web トラッキング技術は，プライバシー侵害等のリスクをはらんでいる．Web サイトはユーザーを一意に識別するために Web Browser Fingerprint と呼ばれる情報を採取する．本研究では，Web サイトの JavaScript の動的・静的解析によって Web Browser Fingerprint 採取を検知し，トラッキングサイトを網羅的かつ効率的に探索するシステムを開発した．さらに，このシステムを用いて複数サイトにリンクを張るサードパーティのトラッキングサイトの調査を行い，それらをグルーピングした後に Web トラッキングの潜在的なリスクを独自に定量化した．その結果，ほとんどの既存のトラッキング対策ツールは Web Browser Fingerprint 採取を防げないこと，主要なトラッキングサイトは約半数の人気 Web サイトからリンクされていることが明らかとなった．

キーワード：Web トラッキング，Web Browser Fingerprint

An Implementation of Web Tracking Detection System and An Investigation of Third-party Tracking Sites

Yumehisa Haga[1]　Yuta Takata[2,1]　Mitsuaki Akiyama[2]　Tatsuya Mori[1]

Abstract: Web tracking – analyzing user's behavior on the websites – pose some risks such as breach of privacy. Websites collect the information called Web Browser Fingerprint to identify the users. In our research, we developed a system which can detect fingerprinting by dynamic and static analysis of JavaScript and search tracking sites exhaustively and effectively. Furthermore, we investigated and grouped the third-party tracking sites which link to multiple websites by using this system, and uniquely quantified the potential risk of web tracking. As a result, we found that most of anti-tracking tools cannot defense fingerprinting and a major tracking site links with about a half of popular websites.

Keywords: Web Tracking, Web Browser Fingerprint

1. はじめに

急速に発展する Web 産業の中で，ショッピングサイトでの購買や動画サイトでの閲覧等，人々のオンライン上での行動は多様化の一途を辿っている．企業はそのようなインターネットユーザーの行動を追跡してその人の性格や趣味嗜好を分析し，オンライン広告やマーケティングなどの目的に活用している．このような営みのをことを一般に「**Web トラッキング**」と呼ぶ．しかしながら，この Web トラッキングはいくつかのリスクをはらんでいる．Web トラッキングによって，他人には知られたくないユーザーのオンライン上での行動が，ユーザーの意図しないところで漏えいするという、プライバシー侵害の危険性がある．また Web トラッキングは，特定のユーザーのみをフィッシングサイトやドライブバイダウンロードサイト等の悪性

[1] 早稲田大学　基幹理工学研究科
School of Fundamental Science and Engineering, Waseda University
{yumehisa.haga, mori}@nsl.cs.wasead.ac.jp
[2] NTT セキュアプラットフォーム研究所
NTT Secure Platform Laboratories
{takata.yuta,akiyama.mitsuaki}@lab.ntt.co.jp

サイトへ誘導する，攻撃の巧妙化に悪用される．これまでWebトラッキングを回避するための手法がいくつか提案されてきたが，トラッキング技術も年々高度化しており，それらを網羅的に防ぐ手段はほどんど存在しないのが現状である．

ここで，すべてのWebトラッキングを悪とみなし，無効化して良いのかという議論がある．Webトラッキングは，ユーザーに最適な広告を表示するといった恩恵を与えると同時に，前述のようなリスクもつきまとう．このことを加味して最終的にトラッキングを許容するか否かの判断は，個々のユーザーに委ねられるべきである．W3Cにおいて “Do Not Track” [1] という，ユーザーがWebサイト側に対してトラッキング拒否の意思表示ができる規格が策定されているように，ユーザーがトラッキングから身を守る権利は一般的に認められている．しかしながら，現状は一部のサイトを除いてユーザーの意志にかかわらず無制限にトラッキングが行われている．2013年には，米国にてFacebookがユーザーのネット履歴を不正に追跡したとして，Facebookが950万ドルの和解金を支払う訴訟問題にまで発展している [2].

1.1 Webトラッキングの手法

WebサイトがWebトラッキングを行う際には，ユーザーのブラウザを一意に識別する必要があるが，そのための識別子として，Cookieを利用することが多い．しかしながら，Cookieはユーザーの操作で簡単に削除・ブロックができ，さらにSame-Origin Policyによってドメインをまたいで同じCookieにアクセス出来ないため，ユーザーの追跡には限界がある．そこで近年，Cookieに代わるブラウザ識別子としてWeb Browser Fingerprint（以下，**WBF**と省略）と呼ばれる情報が使用されるようになった．WBFとは，ユーザーのブラウザの種類，画像解像度，プラグインの名前，インストール済みフォントなどの特徴点の組み合わせであり，JavaScript等を用いて容易に採取できる．Eckersley [3] によれば，JavaScriptやFlashが有効になっているブラウザにおいて，WBFを元に94.2%の確度でユーザーのブラウザを一意に特定することが可能とされている．WBFとCookieとの大きな違いとして次の3点が挙げられる．1) Same-Origin-Prilcyのような制限がなく，ドメインをまたいで追跡できる．2) ユーザーの操作で削除したり，値を変更することが困難．3) 通常利用とトラッキング目的での利用の区別が困難．このように，Webトラッキングにおいて，WBFはCookieよりも大きな脅威であるといえるため，本研究では特にWBFを使ったWebトラッキングに着目する．また，WBFを使ったトラッキングは，どのWBFを選択し採取するかによってその性能が変化する．本研究ではこのトラッキングの性能のことを「トラッキング力」と呼ぶ．

1.2 ファーストパーティサイトとサードパーティサイト

Webトラッキングを行うサイトは,「ファーストパーティサイト」と「サードパーティサイト」の二つに大別される．前者はユーザーが訪れたWebサイトそのものであり，後者はファーストパーティサイトからリンクされた，ドメインの異なる外部のサイト（または組織）のことを指す．サードパーティサイトは同時に多くの異なるWebサイトからリンクされており，複数のWebサイトにまたがって横断的にユーザーをトラッキングしている．したがってサードパーティサイトによるトラッキングはファーストパーティサイトによるそれよりも「影響度」が大きいと考えられ，今後対策を急ぐべきである．一方ファーストパーティサイトでは，本来Web画面の体裁等を整えるためにWBFを採取・参照することが多く，これを無効化することはユーザビリティの低下に繋がるおそれがある [4]．したがって，本研究では特にサードパーティサイトによるトラッキングに着目し，トラッキングを行うサードパーティサイトのことを「トラッキングサイト」と呼ぶこととする．

1.3 狙いとアプローチ

本研究の目的は，Webトラッキングに関して先行的に実態調査を行い，Webトラッキングの検知・防御手法の確立に役立てることにある．特に今回は，トラッキングサイトによるWBFを用いたトラッキングに着目する．トラッキングサイト毎に「トラッキング力」と「影響度」を算出し，Webトラッキングの潜在的なリスクを可視化する．これを達成するために，インターネット上のWebサイトを巡回し，トラッキングサイトによるWBF採取を検知するためのシステムを構築する．本研究では，JavaScriptの静的解析と動的解析を組み合わせる手法により網羅的かつ効率的にトラッキングサイトを検出するクローラを開発した．

1.4 貢献

本研究の主要な貢献を以下に示す．

- JavaScriptを用いたWebトラッキングを検知する汎用的なシステムを構築した．
- 関連しているトラッキングサイト同士をグルーピングし，それらについて複数の尺度を用いてWebトラッキングの潜在的なリスクを定量化する手法を提案した．
- トラッキングサイトのトラッキングコードの調査により，ほとんどの既存の対策ツールではWBF採取を防ぐことはできないこと、最も主要なトラッキングサイトは普段我々がアクセスするWebサイトの約半数からリンクされていることを明らかにした．

2. 関連研究

本章では，WebトラッキングあるいはWBFに関連する幾つかの研究について述べる．

Eckersley [3] らは Web トラッキングにおける WBF の有効性を初めて実証した．彼らは Fingerprint として User-Agent や HTTP Accept Header，画面解像度，タイムゾーン，ブラウザのプラグイン，インストール済みフォントなどを利用し，ユーザーを 94.2%の確度で一意に識別することができることを実証した．その後，多くの関連研究が行われ，パフォーマンス測定 [5]，JavaScript エンジン [6]，レンダリングエンジン [7]，さらには HTML5 の Canvas [8] などがユーザーを識別する Fingerprint として利用できることが明らかとなった．

Acer らは [9] [10]，ヘッドレスブラウジング技術を用いて Web サイト上で実行される JavaScript を動的に検知するシステムを構築し，Web トラッキングの実態調査を行った．その結果，多くのサードパーティ組織が WBF を用いてユーザーのトラッキングを行っていることが判明した．

磯ら [11] は，WBF を採取する Web サイトを構築し，収集データの分析を行った．約 4 ヶ月間に渡って収集した 1767 個の Fingerprint を分析した結果，それらの一部の値が時間経過に伴って変化することを明らかにした．同時に，端末に対する Fingerprint のばらつき具合（エントロピー）を算出し，どのような Fingerprint がユーザーの識別により貢献するかを示している．

本研究では Acer らのトラッキング検知システムよりも高度なシステムを開発し，Web トラッキングの実態調査を行う．さらに Web サイトの人気度や Eckersley らや磯らが調査した各 WBF の特性など，様々な指標を元に，Web トラッキングの潜在的なリスクがインターネット上にどれほど潜んでいるのかを定量的に分析する．

3. Web トラッキング検知システム

本研究では，トラッキングサイトの実態調査を行うために，WBF の採取を検知するシステムを開発した．本章では，今回開発したシステムの特徴について述べる．

3.1 Web Browser Fingerprint の種類

表 1 に，JavaScript で採取可能な WBF の一覧を示す．同時に，それぞれの WBF の端末ごとのばらつき (Entropy) を High・Med・Low，経年変化に対する耐性 (Duration) を Long・Med・Short の 3 段階で評価したものも示す．WBF の Entropy が高ければより高精度にユーザーを識別でき，Duration が高ければより長い期間ユーザーをトラッキングすることができる．これらの評価は，著者らのこれまでの調査および Eckersley ら [3] や磯ら [11] が多くのサンプルに基づいて分析を行った結果を踏まえてヒューリスティックに定めた．たとえば User-Agent の場合，ブラウザの種類やバージョンの情報が含まれているため Entropy は比較的高いが，ブラウザのバージョンアップによって頻繁に値が変化するため Duration は低くなっている．一方，Platform はユーザーの利用 OS を表現する特徴である．OS の種類は限定的であり，ユーザが頻繁に変更することはないため Entropy は低く，Duration は高くなる．

WBF の採取方法には JavaScript を用いる方法以外にも，Flash や HTTP リクエストヘッダを利用する手法があるが，JavaScript はその手軽さゆえに Web トラッキングにおいて最も多用される．したがって，本研究では特に JavaScript のトラッキングコードを調査の対象とする．

3.2 システムの概要

本研究で開発した Web トラッキング検知システムの概要について述べる．図 1 に本システムの概要図を示す．本システムの主な構成要素として，ヘッドレスブラウザ，ドライバ，アナライザがある。

ヘッドレスブラウザ: ヘッドレスブラウザとは、GUI レスのブラウザのことを指す．ヘッドレスブラウザは，内部的にページのロードや描画を行ったり，JavaScript を実行したりなど，基本的なブラウザの機能を備えており，Web 開発者がフロントエンドのテストを行う際などに活用される．本研究ではこのヘッドレスブラウザに， 3.3 節〜 3.4 節で示すような，Web トラッキング検知のための機能を実装する．このヘッドレスブラウザによって，表 1 で示した WBF の採取をすべて検知できる．また今回ベースとなるオープンソースのヘッドレスブラウザとして，HtmlUnit [12] を採用した．

ドライバ: ドライバは，ヘッドレスブラウザを並列実行させ，実行ログを人が見やすいように JSON 形式に加工するための Python で実装されたプログラムである．この，ドライバとヘッドレスブラウザを組み合わせた機構のことをクローラと呼ぶ．今回開発したヘッドレスブラウザは，1 つの Web サイトの解析を行うのに 3〜10 秒ほどかかるが，マシンスペックに応じて並列数を増やすことができ，高速に Web サイトを探索することができる．

アナライザ: アナライザは，ドライバが出力した JSON ログに対して種々の分析を行い，分析結果をデータベースに格納する．

今回，以上の一連の処理を全て自動化し，効率的な Web

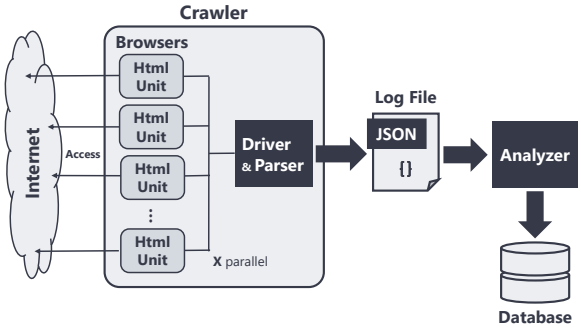

図 1 Web トラッキング検知システムの概要図

表 1　JavaScript で採取可能な WBF

Property or Method	Description	Entropy	Duration
toDataURL(), getImageData()	Canvas image data	High	Short
navigator.appCodeName	Code name of the browser	Low	Long
navigator.appName	Name of the browser	Low	Long
navigator.appVersion	Version of the brower	High	Short
navigator.userAgent	User-Agent	High	Short
navigator.mimeTypes	MIME types	Med	Long
navigator.plugins	Installed plugins	High	Short
navigator.language	Language of the brower	Med	Long
navigator.platfoorm	Platfrom of the browser	Low	Long
screen.height, screen.width	Size of screen	Med	Med
screen.availHeight, screen.availWidth	Size of screen (excluding the Windows taskbar)	Med	Med
screen.colorDepth	Bit depth of the color palette	Med	Med
screen.pixelDepth	Color resolution of the screen	Med	Med
getFontList()	Installed fonts	High	Med
getTimezoneOffset()	Time Zone	Med	Med

サイトの調査を実現した．

3.3　JavaScript の動的解析と動的解析

本研究では，前節で述べた HtmlUnit に対して改良を加え，JavaScript による WBF 採取を検知するための機能を付加した．JavaScript で WBF を採取するには，特定のオブジェクトにアクセスしたり，特定の関数を実行する必要がある．たとえば，User-Agent の情報を参照したい場合，JavaScript で “navigator.userAgent” にアクセスすれば良い．本研究では，JavaScript の動的解析と静的解析の二つのアプローチにより WBF 採取を検出する．動的解析では，ヘッドレスブラウザ内で JavaScript を実行し，WBF に関連する特定のオブジェクトへのアクセスや関数の実行をフックする．静的解析では，JavaScript のソースコードをダンプし，WBF に関連する文字列を検出する．

動的解析と静的解析を組み合わせることのメリットは大きい．動的解析は，関数の呼び出し等を厳密に評価でき，ソースコードの難読化の影響を受けないが，JavaScript の分岐処理によって実行されない命令を見落とす場合がある．実際に著者らの調査では，特定の Fingerprint が得られたか否かで条件分岐を行うトラッキングコードが見つかっている．一方，静的解析では，関数の呼び出しなどを厳密に評価できないが，JavaScript の分岐処理による影響は受けない．このように，2 つの手法の弱点を互いに補い合うことにより，より網羅的に WBF 採取を検知することができる．

3.4　難読化の解除

本システムでは，さらに JavaScript の難読化を解除する機能を追加した．難読化とは，ソースコードを人が読めないように加工することをいい，リバースエンジニアリングなどを防ぐ目的で行われる．悪性コードやトラッキングコードを隠すために難読化が用いられることも少なくない．JavaScript でよく用いられる難読化手法は，図 2 のように `eval()` 関数を利用した手法である．

```
// 難読化前
document.write("Hello, World");

// 難読化後
eval(function(p,a,c,k,e,d){e=function(c){return
c};if(!''.replace(/^/,String)){while(c--){d[c]
=k[c]||c}k=[function(e){return d[e]}];e=function()
{return'¥¥w+'};c=1};while(c--){if(k[c]){p=
p.replace(new RegExp('¥¥b'+e(c)+'¥¥b','g'),
k[c])}}return p}('0.1("2, 3");',4,4,
'document|write|Hello|World'.split('|'),0,{}))
```

図 2　JavaScript の難読化の例

ソースコードにこのような処理が施された場合，静的解析を行うことができない．しかし，`eval()` 関数の引数の中身を動的に実行し，その結果をダンプすることで，難読化前のソースコードを復元することができる．本システムにおいてもこのような機能を実装し，難読化が施された JavaScript の静的解析を可能にした．

4.　トラッキングサイトの調査

本研究では，3 章で述べたシステムを利用し，トラッキングサイトの調査を行った．本章では，トラッキングサイトのグルーピング手法，トラッキングのリスクの定量化手法，および分析結果について述べる．

4.1　調査の対象

1 章で述べたように，トラッキングサイトは，トラッキングを行うサイトの中でも，ユーザーが Web サイトにアクセスした際に，そのサイトから外部にリンクされている（リダイレクトされる）サードパーティのサイトあるいは組織のことを指す．本研究の調査の対象となるのは，Alexa [13] に登録されたアクセス数の多い人気サイト上位

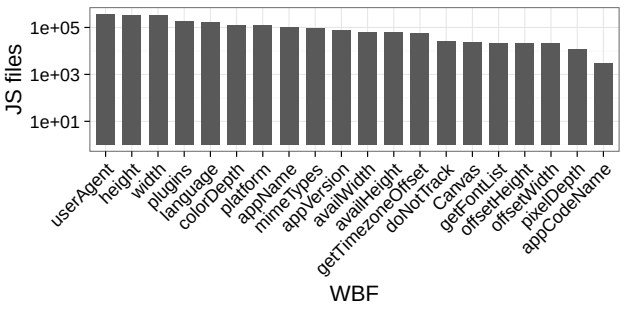

図 3 各 WBF の使用頻度

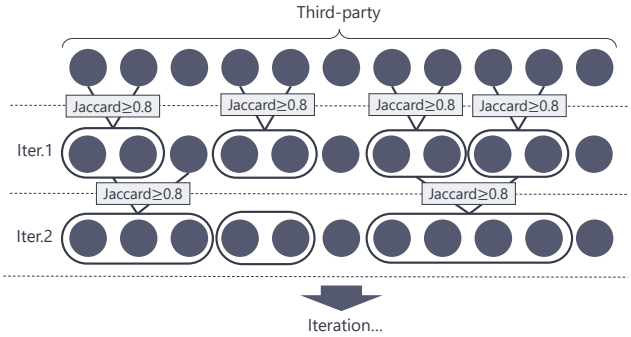

図 4 再帰的 Jaccard 計算

10 万の Web サイトにリンクされているトラッキングサイトである．それぞれのサイトについて，トップページにアクセスし，そこで実行される JavaSript を解析する．トラッキングサイトは URL のドメイン（FQDN）で識別することとし，これをトラッキングサイト名と呼ぶこととする．クロールの結果，全部で 37,812 種のトラッキングサイトが抽出され，そのうち 17,290 のトラッキングサイトが JavaScript を用いて少なくとも 1 つの WBF を採取していることが分かった．今回解析した JavaScript ファイルにおける各 WBF の使用頻度を図 3 に示す．

4.2 再帰的 Jaccard 計算によるトラッキングサイトのグルーピング

トラッキングサイトの中には，トラッキングサイト名が異なるが，同じ組織のものであるケースがある．それらは互いに裏でトラッキング情報のやり取りを行っている可能性が高い．そのような，トラッキングサイト名が異なるが中身は同じトラッキングサイトの集合のことを「トラッキンググループ」を呼ぶ．トラッキンググループが多数のトラッキングサイトの集まりであったとき，一つ一つのトラッキングサイトが小さいために巨大な組織を見逃してしまうことが考えられる．そこで本研究ではトラッキングサイトの分析に先立ち，トラッキングサイトのグルーピングを行う．

同じトラッキンググループに属するトラッキングサイトは，同じファーストパーティサイトに同時にリンクしているケースが多いということが著者らの調査で明らかになっている．したがって，トラッキングサイトがどのファーストパーティサイトにリンクしているかという情報に基づいて，トラッキングサイトをクラスタリングする．クラスタリングの手法として，再帰的に Jaccard 係数を計算するアルゴリズムを採用する．Jaccard 係数は，2 つの集合の類似度を表す数値表現のひとつである．まず，とある 2 つのトラッキングサイトの類似度を計算することを考える．一方のトラッキングサイトがリンクしているファーストパーティサイトの集合を A, もう片方のトラッキングサイトがリンクしているファーストパーティサイトの集合を B とすると，両者の類似度を示す Jaccard 係数は以下のように計算される．

$$Jaccard = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

本研究では，Jaccard 係数が 0.8 以上になったトラッキングサイト同士は同じトラッキンググループに属しているとみなす．この処理を全てのトラッキングサイトに施すと，2 つの要素を持つトラッキンググループが複数出来上がる．次に，そのトラッキンググループ同士を比較し，類似度の高いものを一つのグループにまとめ，これを繰り返していく（図 4）．するとトラッキンググループは次第に大きくなり，新たにグループ生成が起こらなくなった（収束した）段階で計算を終了する．これと同様の手法が，ドライブバイダウンロード研究の Hostname-IP Clucter (HIC) に用いられている [14].

再帰的 Jaccard 計算の結果，最大 3 つの要素を持つトラッキンググループが生成された．トラッキンググループの例として，{b.travel-assets.com, a.travel-assets.com, c.travel-assets.com}, {www.homeaway.jp, static0.homeaway.jp, static1.homeaway.jp}, {www.tribdss.com, ssor.tribdss.com, www.trbas.com} などがある．それぞれ似たようなトラッキングサイト名になっており，これらが互いに関係を持っていることが推察される．

4.3 トラッキングリスクの定量化

Web トラッキングにおいて，WBF が Web 管理者によってどのように使われているかはユーザーから見えにくく，それが本当に悪用されているかどうかがわからない．しかし，今回開発したシステムを用いて得られる情報を元に，万一ユーザーのトラッキング情報が悪用（あるいはプライバシー漏えい）された時の危険度を知ることができる．本研究では，そのような Web トラッキングの “潜在的なリスク”を定量化するために，大きく 2 つの指標を定義する．ひとつは「トラッキング力」であり，もうひとつは「影響度」である．

「トラッキング力」とはユーザーを一意に識別する精度

とそれが有効な期間の長さのことをいい，表 1 に示したような，採取する WBF の Entropy と Duration から求めた．Entropy と Duration の高い WBF をできるだけ多く採取することでよりトラッキングの性能が向上する．そこで，あるトラッキングサイトが採取した WBF の集合を $X = \{x_1, x_2, ..., x_n\}$ としたとき，そのトラッキングサイトのトラッキング力（TP: Tracking Point）を以下のように計算する．

$$TP = \sum_{i=1}^{n} (3 * Entropy(x_i) + Duration(x_i)) \qquad (2)$$

なお，WBF に対する Entropy と Duration の値は，表 1 の {High, Med, Low} と {Long, Med, Short} をそれぞれ {3 点, 2 点, 1 点 } と定めた．ユーザーを一意に特定するという Web トラッキングの特性上，Duration よりも Entropy の方が重要であるため，スコア算出時に 3 倍の重み付けをした．

もう一つの指標である「影響度」とは，トラッキングサイトがユーザーを追跡できる範囲の広さのことをいい，リンクされたファーストパーティサイトの数とそのファーストパーティサイトの Web サイト訪問者数に基づいて定めた．影響度が大きいほど，ユーザーはインターネット上のいたるところで情報を採取されることを意味する．Web サイトの訪問者数を厳密に知ることはできないが，Alexa ランキングをもとにある程度推測することができる．Alexa ランキングが上位のサイトに多くリンクしているトラッキングサイトほど影響度が大きい．あるトラッキングサイトがリンクしているファーストパーティサイトの集合を $Y = \{y_1, y_2, ..., y_n\}$ としたとき，そのトラッキングサイトの影響度（IP: Impact Point）は以下のように計算する．

$$IP = \sum_{i=1}^{n} \frac{100000 - Rank(y_i)}{1000} \qquad (3)$$

なお，Rank とは，そのサイトの Alexa ランキングを指し，最大値は 100,000 をとる．

4.4 分析結果

4.2 節で述べた手法によってグルーピングしたトラッキンググループについて、算出された TP と IP の分布を図 5 に示す。また，TP と IP の値が高かったトラッキンググループの一覧をそれぞれ表 2 と表 3 に示す．なお，表 2 と表 3 に示したトラッキンググループはいずれも一つのトラッキングサイトで構成される結果となった．

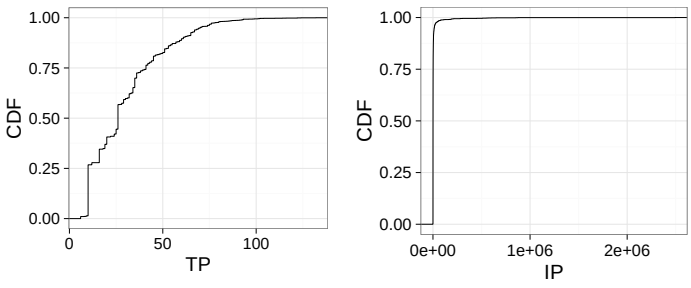

図 5 TP(右) と IP(左) の分布.

表 2 より，最も TP が高かったのは “cdn.krxd.net” の 132 点だったが，これは表 1 の WBF のほとんどを採取していることを示している．“cdn.krxd.net” は，Krux [15] という組織が運用しているドメインであり，Web サイトで収集したユーザーの情報を元にマーケティング等のサービスを提供している．表 2 に示したトラッキングサイトはいずれもアクセス解析によって広告，マーケティング，オンライン詐欺対策を行っている組織であることが分かった．

ここで，これらのトラッキングサイトによる WBF 採取が既存の対策ツールで防げるかどうかを評価する．各ツールがサポートしている WBF の情報をもとに TP を計算すると，その値がいわば各ツールの WBF 採取に対する防御性能に相当することになる．また，あるツールが防御しうるトラッキンググループの IP の累積値はそのツールの有効性 (Effectivity) を表し，有効性が高いほどユーザーはより多くの Web サイトでトラッキングを回避できることになる．今回既存の WBF 採取対策ツールとして，PriVaricator (PV) [16]，FireGloves (FG) [17]，Random Agent Spoofer

表 2 TP の高いトラッキンググループ一覧

Third-party	Path	TP
cdn.krxd.net	ctjs/controltag.js.875fd5b280a77e06def8c74a5a268e2c	132
static.fraudmetrix.cn	fm.jsv?er=0.1&t=402217	124
cse.google.com	adsense/search/async-ads.js	122
app.trustev.com	api/v2.0/TrustevJS?key=7a3c4bffd29b40cea21dc2a05446768d	116
cdn.tagcommander.com	362/tc_Aspartam_3.js	114
tags.mdotlabs.com	tracking.php?siteID=e8AJ	110
static.audienceinsights.net	t.js	104
t.qservz.com	js/pi.js	102
tags.tiqcdn.com	utag/wsjdn/wsjpages/prod/utag.56.js?utv=201510271832	102
servedby.openxmarket.asia	servedby.openxmarket.asia/w/1.0/jstag	102

表 3 IP の高いトラッキンググループ一覧

Third-party	Links	IP
www.google-analytics.com	49,963	2,493,837
pagead2.googlesyndication.com	17,586	875,754
connect.facebook.net	17,147	853,935
www.googletagmanager.com	12,226	650,517
partner.googleadservices.com	10,241	599,434
ajax.googleapis.com	11,592	566,757
www.googleadservices.com	10,474	547,918
apis.google.com	10,310	512,459
platform.twitter.com	10,099	506,008
tpc.googlesyndication.com	9,715	449,372
static.xx.fbcdn.net	6,649	303,266

(RAS) [18]，Tor Browser (Tor) [19]，FP-Block (FPB) [4] を評価対象とした．各ツールについて算出した TP の値と，WBF 採取を防御しうるトラッキンググループの割合 (Coverage)，ツールの有効性を表す値 (Effectivity) を表 4 に示す．これらのツールの中では FP-Block が最も高い防御性能を有していることが分かる．表 2 と見比べると，トラッキング力の高いトラッキングサイトに対して，FP-Block では対策可能であるが，その他のツールは防御性能が不十分であることが分かる．FP-Block は比較的新しいツールであり，その他は古くから一般に使われてきたツールであることから，Web トラッキング技術がここ最近で進化してきていることも示唆している．例えば，2012 年に Mowery [8] らによって新しく提唱された Canvas Fingerprint は近年利用されるケースが増えているが，FireGloves のような比較的古いツールは対応していない．以上のことから，ユーザーがトラッキングを回避しようとする際は慎重に対策ツールを選択する必要があり，現状では FP-Block を使用するのが最善策である．

次に, 表 3 に注目すると, IP の高いトラッキングサイトは Google，Twitter，Facebook などユーザー数の多いサービスを提供している組織であることが分かった．特に Google のアクセス解析サービスである Google Analytics [20] の IP が他のトラッキングサイトに比べ非常に高く，今回調査したファーストパーティサイトの約半数にリンクしていることが分かった．なお Google Analytics の TP は 69 点であり，前述したツールによって対策可能である．ここで Alexa ランキングの上位から順番にアクセスした際に，アクセスしたサイトに含まれる Google Analytics，Facebook，Twitter のリンクを集計し，情報伝播の様子を計測した結果を図 6 に示す．この図を見ると，影響度の大きい Google Analytics はグラフの傾きが大きく，他のトラッキングサイトに比べ Web サイトを巡回するにつれて急速にユーザー情報が伝搬（漏洩）していくのが分かる．ユーザーはトラッキングについて考える際，自分がアクセスしているサイトそのものだけでなく，その裏でリンクしているサードパーティサイトの存在を強く意識しなければならない．

トラッキンググループごとの TP と IP の関係を図 7 の散布図に示す．TP と IP とが共に大きい，つまり，グラフの右上にプロットされているトラッキンググループほど，Web トラッキングの潜在的なリスクが大きいことを表している．また，WBF の各対策ツールの TP を破線で示しており，破線よりも上にプロットがあるトラッキングサイトのトラッキングは，その対策ツールでは防ぎきれないことを示している．影響度の高い Google は，サービスの種類によって採取する WBF の組み合わせを変え，トラッキング力を調整している様子が見て取れる．

表 4　WBF 採取対策ツールの防御性能

Tool	TP	Coverage %	Effectivity
PriVaricator	19	53.3	1,345,753
FireGloves	66	96.6	9,314,979
Random Agent Spoofer	80	98.7	14,354,210
Tor Browser	99	99.8	17,014,206
FP-Block	136	100.0	17,238,061

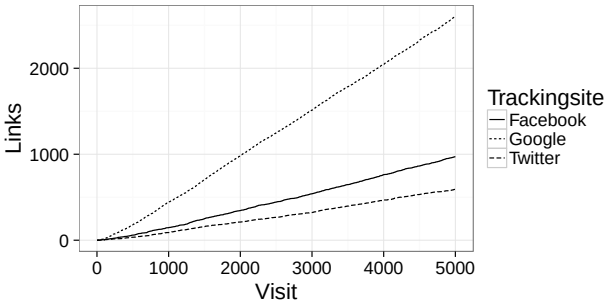

図 6　主要なトラッキングサイトの情報伝搬の様子

5. 議論

本章では，本研究の制限と今後の課題について述べる．トラッキング力の算出において，他の研究や経験を元に WBF の Entropy と Duration を 3 段階でヒューリスティックに評価したが，それらの明確な判断基準は定められていない．したがって，よりトラッキング力の妥当性を担保するために，独自に WBF を収集・分析するシステムを構築し，あらゆる WBF の Entropy と Duration を定量的に定める必要がある．

また，今回 Web トラッキングの “潜在的な” リスクを定量化したが，実際に採取された WBF がどのように使用されているかを知ることはできないため，もし WBF が悪用されたとしてもそれを検知することはできない．そこで，例えば Web サイトに自分の WBF を採取させ，全く関係ない別のサイトに移動した時に，何らか形でトラッキングに起因する挙動（ターゲット広告など）が確認できれば，自分のプライバシー情報が意図しないところで流出していることが間接的に分かる．また，トラッキング拒否の意思表示ができる Do Not Track という仕組みがあるが，これを有効にした状態で Web サイトにアクセスしたにもかかわらず WBF が採取されたり利用されたりした場合，その Web サイトは悪質である可能性が高い．

6. 結論

本研究では，複数サイトに渡ってトラッキングを行うサイトの分析を行うために，WBF の採取を検知できるシステムを開発した．このシステムは JavaScript を動的解析と静的解析（難読化解除も可能）を組み合せて評価することで，より網羅的に WBF 採取を検知できる．このシステムを利用して Alexa の人気上位 10 万サイトを調査したところ，WBF を採取する多くのトラッキングサイトを抽出し

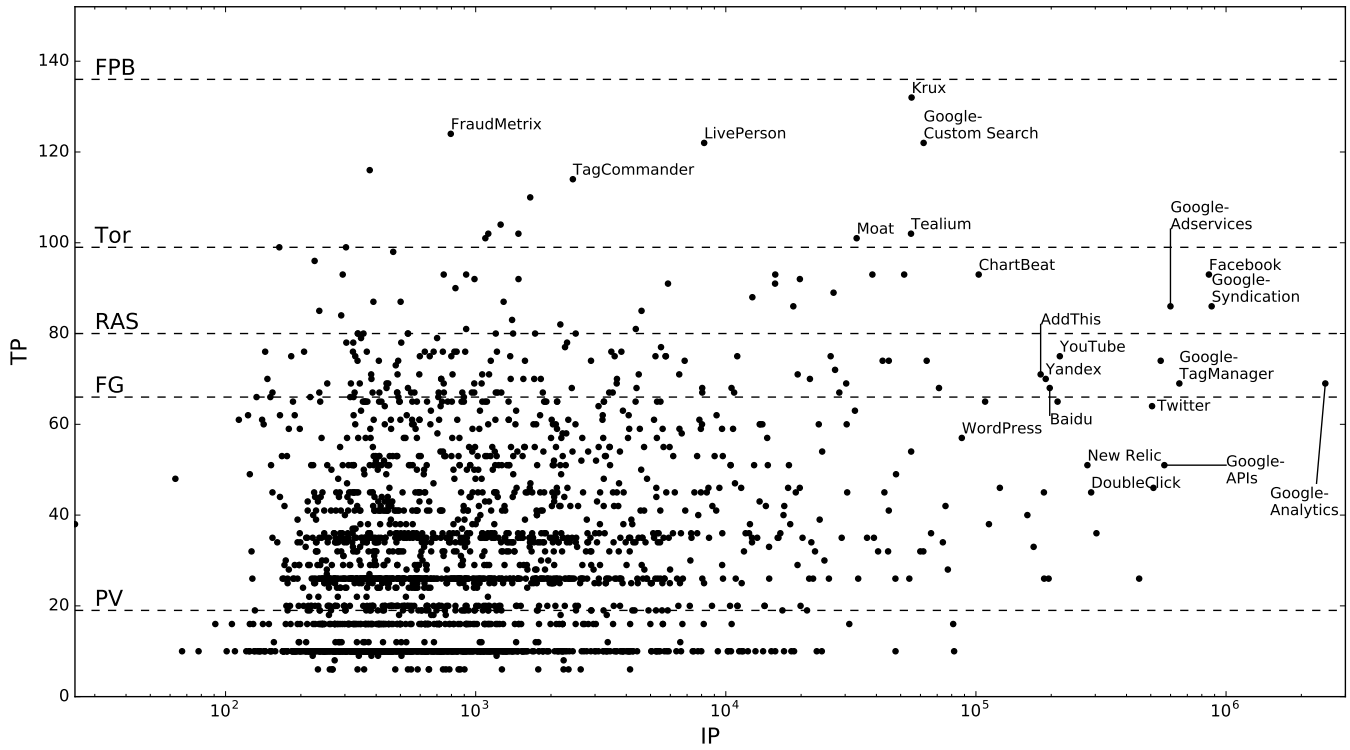

図 7 トラッキンググループの分布

た．また，トラッキングサイト同士で関連の強いものをグルーピングし，それぞれについて独自の方法で「トラッキング力」と「影響度」を算出し，Web トラッキングの潜在的なリスクを定量化した．その結果，トラッキング力の高いトラッキングは，最新のものを除く多くの対策ツールでは防御不可能であることが判明し，ユーザーは慎重に対策ツールを選択しなければならないということが分かった．さらに，影響度の高いトラッキングサイトは我々が普段アクセスする Web サイトの半数近くに対してリンクを張り巡らせていることが分かった．したがって，ユーザーは普段のオンライン上での行動の大半が一つの組織によって追跡されている可能性が高いことを常に意識する必要がある．

参考文献

[1] “Tracking Preference Expression (DNT).” `https://www.w3.org/TR/tracking-dnt/`.

[2] D. Kravets, “Facebook’s $9.5 Million ’Beacon’ Settlement Approved.” `http://www.wired.com/2012/09/beacon-settlement-approved/`.

[3] P. Eckersley, “How unique is your web browser?,” in *Privacy Enhancing Technologies (PETs)*, 2010.

[4] C. Torres, H. Jonker, and H. Mauw, “Fp-block: usable web privacy by controlling browser fingerprinting,” in *ESORICS*, 2015.

[5] K. Mowery, D. Bogenreif, S. Yilek, and H. Shacham, “Fingerprinting information in JavaScript implementations,” in *Web 2.0 Workshop on Security and Privacy (W2SP)*, 2011.

[6] M. Mulazzani, M. H. P. Reschl, M. Leithner, S. Schrittwieser, E. Weippl, and F. C. Wien, “Fast and reliable browser identification with JavaScript engine fingerprinting,” in *Web 2.0 Workshop on Security and Privacy (W2SP)*, 2013.

[7] T. Unger, M. Mulazzani, D. Fruhwirt, M. Huber, S. Schrittwieser, and E. Weippl, “SHPF: Enhancing HTTP(S) Session Security with Browser Fingerprinting,” in *In Availability, Reliability and Security (ARES)*, 2013.

[8] K. Mowery and H. Shacham, “Pixel perfect: Fingerprinting canvas in HTML5,” in *Web 2.0 Workshop on Security and Privacy (W2SP)*, 2012.

[9] G. Acar, M. Juarez, N. Nikiforakis, C. Diaz, S. Gurses, F. Piessens, and B. Preneel, “FPDetective: Dusting the Web for fingerprinters,” in *ACM Conference on Computer and Communications Security (CCS)*, 2013.

[10] G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, and C. Diaz, “The Web Never Forgets: Persistent Tracking Mechanisms in the Wild,” in *ACM CCS*, Nov 2014.

[11] 磯, 桐生, 塚本, 須, 山田, 武居, and 齋藤, “Web browser fingerprint を採取する web サイトの構築と採集データの分析,” in *CSS*, 2014.

[12] “HtmlUnit.” `http://htmlunit.sourceforge.net/`.

[13] Alexa Top Sites. `http://www.alexa.com/topsites`.

[14] W. Lee and J. Stokes, “ARROW : Generating Signatures to Detect Drive-By Downloads,” 2011.

[15] Krux. `http://www.krux.com/`.

[16] N. Nikiforakis, W. Joosen, and B. Livshits, “Privaricator: Deceiving fingerprinters with little white lies,” in *Technical Report MSR-TR-2014-26, Microsoft Research*, 2014.

[17] K. Boda, .M. Fldes, G. Gulys, and S. Imre, “User tracking on the web via cross-browser fingerprinting,” in *16th Nordic Conference in Secure IT Systems*, 2011.

[18] Random Agent Spoofer. `https://github.com/jmealo/random-ua.js`.

[19] R. Dingledine, N. Mathewson, and P. Syverson, “Tor: The second-generation onion router. technical report,” in *Naval Research Lab Washington*, 2004.

[20] Google Analytics. `https://www.google.com/analytics/`.