Vol.2019-CSEC-87 No.6
2019/12/4

ブラウザとTLSによるアンチフィンガープリンティングツールの性能評価

中村 綾花[1,a)]　大久保 隆夫[1]

概要：コンピュータにおいて，通信パケットの特徴ややブラウザから取得できる情報の特徴を利用して利用者を追跡する技術の一つにフィンガープリンティングがある．フィンガープリンティングは利用者の許可なく行うことができるため，プライバシー保護の観点から問題となっている．そのため，近年ではフィンガープリンティングを拒否したり偽装して通信できるアンチフィンガープリンティング機能を持ったツールが多く存在しているが，現在のアンチフィンガープリンティングツールの多くは偽装や拒否できるフィンガープリントの種類が限られているため，完全にフィンガープリンティングを欺けていないと考えられる．本稿では，アンチフィンガープリンティングツールがどの程度フィンガープリンティングに対して効果があるかを明らかにするため，ブラウザフィンガープリンティングとTLSフィンガープリンティングの2種類のフィンガープリンティングを行い，アンチフィンガープリンティングの性能を調査した．

キーワード：ブラウザフィンガープリンティング, TLSフィンガープリンティング

Performance evaluation of anti fingerprinting tools for browser and TLS

Nakamura Ayaka[1,a)]　Okubo Takao[1]

Abstract: Fingerprinting is one of techniques for tracking users by using characteristics of packets and characteristics of information that can be acquired from a browser. Fingerprinting can be performed without the user's permission, but it is a problem of privacyprotection. There are many tools with an anti-fingerprinting function that canrefuse fingerprinting or impersonate and communicate, but many of the current anti-fingerprinting tools havelimited types of fingerprints that can be impersonated or rejected. Therefore, it is thought that fingerprinting is not completely deceived. In this paper, in order to clarify how effective the anti-fingerprinting tool is for fingerprinting, we conducted two types of fingerprinting, browser fingerprinting and TLS fingerprinting, andinvestigated the anti-fingerprinting performance.

Keywords: Browser fingerprinting, TLS fingerprinting

1. はじめに

現在，多くのWebサイトにおいて何らかのユーザートラッキングが行われている．ユーザートラッキングを行う目的は，ユーザの行動を把握してそれぞれのユーザにあった広告を表示させるなどのマーケティングのためと言われている．

2014年にAcarらが行った研究 [1] では，Alexaによるアクセス数上位10,000サイト中の5.5%のサイトにフィンガープリンティングによるユーザトラッキングが行われていたとされており，ユーザに無断での情報収集が常態化していると考えられる．

フィンガープリンティングによるユーザトラッキングを防止するため，フィンガープリントを偽装したり取得を制限するアンチフィンガープリンティングが存在する．

[1] 情報セキュリティ大学院大学
Institute of Information Security
[a)] mgs181001@iisec.ac.jp

Firefox や Google Chrome などの主要ブラウザにおいてもデフォルトでアンチフィンガープリンティングの機能実装が進められており [2][3]，ユーザのプライバシー保護の観点から重要な技術となっている．

本稿では，アンチフィンガープリンティングの有効性を把握するため，アンチフィンガープリンティングを利用したブラウザにおいてブラウザフィンガープリンティングと TLS フィンガープリンティングを行い，どの程度フィンガープリンティングを防止できるか調査を行った．結果から，調査したアンチフィンガープリンティングツールの全てにおいて，ブラウザフィンガープリンティングと TLS フィンガープリンティングの両方に効果のあるものは存在せず，現状のアンチフィンガープリンティング技術では完全にユーザトラッキングを防ぐことが難しいことが示された．

2. フィンガープリンティング

フィンガープリンティングとは，端末から得られる情報を元に端末を一意に識別する技術である．おおまかにアクティブフィンガープリンティングとパッシブフィンガープリンティングという 2 種類の手法に分類できる．

- アクティブフィンガープリンティング
  対象端末上でプログラムを動作させることにより，能動的にフィンガープリントを取得する．
- パッシブフィンガープリンティング
  対象端末との通信パケットやログを利用し受動的にフィンガープリントを取得する．シグネチャを利用することで端末の環境を予測できる．

ここでは，アクティブフィンガープリンティングの一つであるブラウザフィンガープリンティングと，パッシブフィンガープリンティングの一つである TLS フィンガープリンティングについて解説する．

2.1 ブラウザフィンガープリンティング

ブラウザフィンガープリンティングは，クライアントとの通信時に HTTP Header から取得できる情報と，クライアントのブラウザ上で JavaScript を実行させて様々な情報を送信させることによってクライアントを識別する．

ブラウザフィンガープリンティングによく用いられる特徴情報を以下に示す．これらの情報は複合的に利用されることが一般的である．

(1) HTTP Header から取得する特徴情報

- User-Agent
  クライアントの OS やブラウザのバージョンを示す文字列．
- Accept
  クライアントがコンテンツを閲覧する際に利用するファイル形式．
- Accept-Charset
  クライアントが利用する文字コード．
- Accept-Encoding
  クライアントが利用するエンコード形式．
- Accept-Language
  クライアントが利用する言語．

(2) JavaScript の実行で取得する特徴情報

- Cookie の状態
  クライアントが Cookie の設定を有効にしているか，または無効にしているか．
- ディスプレイの解像度
  ブラウザのディスプレイの解像度．
- タイムゾーン
  ブラウザのタイムゾーン．
- ブラウザプラグイン
  ブラウザで利用されているプラグイン．
- システムフォント
  ブラウザで利用している文字フォント．
- Canvas
  ブラウザ上でグラフィックの描画を行う API．デバイスごとにレンダリングに差異が生じる．
- WebGL
  ブラウザ上で 3D オブジェクトの描画を行う API．デバイスごとにレンダリングに差異が生じる．
- AudioContext
  ブラウザ上でオーディオ処理を行う機能．デバイスごとにオーディオ生成の信号処理に差異が生じる．

Eckerslay[4] の研究では，この手法により 94.2 %のクライアントのブラウザが一意に識別できるとしており，高い識別性能を持つことが示された．

2.2 TLS フィンガープリンティング

TLS フィンガープリンティングは，TLS ハンドシェイクにおいて主にクライアントが最初に送信する Client Hello パケット内の一部をクライアントの特徴情報として利用し，クライアントを識別する．以下に識別に利用される主な情報を示す．

- Client Version
  クライアントがサポートできる TLS の最新バージョン情報．
- Cipher Suites
  クライアントがサポートできる暗号スイートのリスト．複数の暗号を示すことができる．少なくとも 1 つは指定される．
- Extensions
  クライアントが要求するその他の情報．楕円曲線暗号を利用するための追加情報や，サーバの詳細情報を要求する．

(1) Supported Groups
ネゴシエーション中に楕円曲線暗号を利用する場合，クライアントがサポートできる楕円曲線を示す．
(2) EC Point Format
ネゴシエーション中に楕円曲線暗号を利用する場合，クライアントがサポートできるポイントフォーマットを示す．
(3) その他
上記以外の追加要求．例えば，Server Name や Status Request などがある．

- GREASE[5]
現状 Google 社の開発するブラウザのみに実装されているパディング．Cipher Suites と Extension にそれぞれ 0x0A0A〜0xFAFA の値のランダムなパディングを挿入することで，古い TLS 実装のサーバが新しい Extension を処理できない不具合を検知する．

Husák らの研究 [6] では，Cipher Suites と User-Agent を関連付ける辞書を作成することにより，合計 85,250,090 回の HTTPS 通信中 99.6 %について User-Agent を推定可能であることを示された．

3. アンチフィンガープリンティング

フィンガープリンティングによるトラッキングを回避するため，アンチフィンガープリンティング技術が存在する．アンチフィンガープリンティングでは，フィンガープリントを偽装したりフィンガープリンティングを防止するものがある．

3.1 ブラウザ拡張機能のアンチフィンガープリンティング

サードパーティが提供しており，ユーザが有効化することにより機能する．以下に，アンチフィンガープリンティングの機能例を挙げる．

- ブラウザで取得する情報を偽装する
User-Agent や Canvas などを本来の情報から変更する．
- Java Script を禁止する
信頼していない Web サイトでは Java Script を実行 させないことにより，ブラウザフィンガープリンティングを防止する．

3.2 ブラウザ機能としてのアンチフィンガープリンティング

ブラウザの一つである Firefox では，ブラウザの機能の一つとしてフィンガープリンティングを防止する機能が実装されており，ユーザの設定により有効化すると利用できる [2]．また，Google Chrome においても今後の取り組みの一つとしてフィンガープリンティングの防止が挙げられている [3]．

4. 調査

アンチフィンガープリンティングツールを利用できるブラウザにおいて，通常状態とアンチフィンガープリンティングツールを利用した状態でブラウザフィンガープリンティングと TLS フィンガープリンティングをそれぞれ行い，フィンガープリントの値に変化が発生するか調査した．

4.1 調査手法

調査したブラウザは Google Chrome 78.0.3904.70（以下 Chrome）と Firefox 70.0（以下 Firefox）の 2 種類である．ブラウザの利用はクラウドでクロスブラウザテストが行えるサービスである BrowserStack 上で行った．OS は Windows 10 を利用している．フィンガープリンティングツールは既存の OSS を利用しており，ブラウザフィンガープリンティングには Fingerprintjs2[7], TLS フィンガープリンティングには Joy[8] を利用した．調査環境は図 1 に示す．

調査時の手順は以下のように行った．

(1) BrowserStack 上のブラウザからフィンガープリンティングを設置した Web サイト（以下フィンガープリントサイト）にアクセスし，フィンガープリントを取得する．
(2) ブラウザのキャッシュを削除し，調査対象のアンチフィンガープリンティングツールを有効化してからフィンガープリントサイトにアクセスし，フィンガープリントを取得する．
(3) 上記 2 つの状態のフィンガープリントの値を比較する．

4.2 調査したアンチフィンガープリンティングツール

本調査に利用したアンチフィンガープリンティングツールは，Chrome と Firefox それぞれにおいて以下の条件のどちらかを満たすものを選定した．

(1) ブラウザに最初からアンチフィンガープリンティングとして含まれているもの．
(2) 拡張機能のダウンロードサイトにおいて，機能説明にフィンガープリンティングの防止が含まれており，かつ利用しているユーザ数が 1000 人以上であるもの．

選定した結果，Firefox では 10 個，Chrome では 14 個の合計 24 個のツールが該当した．表 1 に選定したツール名とツールの持つ機能の概要を示す．条件（1）の機能は，Firefox のトラッキング防止機能が該当しているが，Chrome においては該当する機能は存在しなかった．条件（2）における一部の拡張機能は Firefox と Chrome の両方でリリースされているが，本調査ではそれぞれ別のツールとして扱っている．

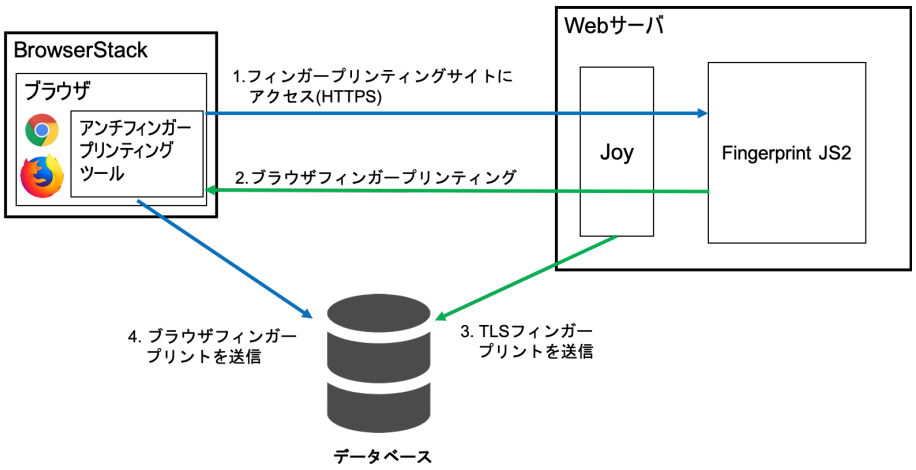

図 1　調査環境

4.3　調査結果

調査結果を表 2 に示す．ツールごとにブラウザフィンガープリントと TLS フィンガープリントの変化の可否を示している．“○”はフィンガープリントが通常状態と比べて変化したことを示し，“x”はフィンガープリントが通常状態と比べて変化しなかったことを示す．“取得不可”は，アンチフィンガープリンティングツールの利用によりフィンガープリントの取得ができなかったことを示す．

この調査結果では，本調査で使用したアンチフィンガープリンティングツールのうちブラウザフィンガープリンティングの変化の無い 2 個のツールを除いて，ブラウザフィンガープリンティングに対して偽装や取得制限を行い効果が発揮されていることがわかる．しかし，TLS フィンガープリントは全て通常状態から変化がないため，TLS フィンガープリンティングに対して偽装や取得制限は行われておらず，効果が不十分であると評価できる．

なお，フィンガープリンティングに変化の無かった 2 個のツールにおいては，5 章で考察する．

5.　考察

調査においてアンチフィンガープリンティングツールを利用していたにもかかわらずフィンガープリントに変化が発生しなかった 2 件に対し，その原因について考察する．

5.1　Firefox のトラッキング防止機能

Firefox の持つトラッキング防止機能では，ブラウザフィンガープリントと TLS フィンガープリントの両方でデフォルト状態との違いを確認することができなかった．

トラッキング機能は Firefox ではバージョン 67 から実装されており，ブラウザの設定から有効化することによってフィンガープリンティングスクリプトの実行を拒否することができる．フィンガープリンティングスクリプトの検知には Disconnect 社が収集したドメインベースのブラックリストを利用しているとされている [2]．そのため，本調査でフィンガープリントの変化が見られなかった原因として，専用の環境を新たに構築して行ったために，ドメインがブラックリストに登録されておらずトラッキング防止機能が動作しなかったと考えられる．

5.2　Chrome の Canvas Defender 拡張機能

Chrome の拡張機能の一つである Canvas Defender では，ブラウザフィンガープリントと TLS フィンガープリントの両方でデフォルト状態との違いを観測することができなかった．ただし，同じ開発者が提供する同名の Firefox 拡張機能ではブラウザフィンガープリントの変更を確認している．

調査時，Chrome 版 Canvas Defender でフィンガープリントサイトにアクセスしたとき，拡張機能に由来する JavaScript のエラーを確認した．エラーはブラウザから Local strage にアクセスできないことが影響していたとみられる．本調査では Browser Stack の仮想環境上のブラウザを利用して行ったため，Local strage にアクセスする権限がうまく付与されないために正常な動作が行われなかったと考えられる．

6.　関連研究

6.1　ブラウザフィンガープリンティングに関する研究

Nikiforakis らの研究 [9] では，User-Agent などを偽装するブラウザ拡張機能を利用するユーザについて，Navigator Object や HTTP Header で取得できる情報との比較により，何も偽装を行わないユーザと比べてフィンガープリントによる追跡可能性が高いと論じられている．

Vastel らの研究 [10] では，さらに WebGL や Canvas，ブラウザプラグイン，CSS のメディアクエリ，ブラウザ固有のエラーや関数などを利用したフィンガープリントを取得し，データベースと比較することにより偽装されたフィン

ガープリントを検出できることを示した．

6.2 TLS フィンガープリンティングに関する研究

Anderson らの研究 [11] では，TLS 通信の Client Hello パケットと Server Hello パケットの Cipher Suites と Extensions の組み合わせによるフィンガープリントを行い，通信の復号を行わずにマルウェアによる通信を高精度に識別できることを示している．

Frolov らの研究 [12] では，TLS フィンガープリントによる追跡可能性を減らすために TLS 通信において任意の Client Hello パケットを作成し，他ブラウザの TLS ハンドシェイクを模倣した通信が可能になる uTLS というライブラリを作成している．

7. まとめ

本研究では，アンチフィンガープリンティングツールのほとんどにおいて，ブラウザフィンガープリンティングに対して効果が発揮されているが，TLS フィンガープリントは全て通常状態から変化がないため，TLS フィンガープリンティングに対して偽装や取得制限は行われておらず，効果が不十分と考えられる．そのため，現状のアンチフィンガープリンティングツールを利用してフィンガープリンティング対策を行っても，ブラウザフィンガープリントと TLS フィンガープリントの両方に効果のあるものがなく，追跡される可能性が残ることが明らかになった．

今後の課題として，ブラウザフィンガープリントと共に TLS フィンガープリントやその他パッシブフィンガープリントを偽装や制限し追跡可能性を減らす方法の検討や実装が残されている．

参考文献

[1] G. Acar, C. Eubank, S. Englehardt, M. Juarez: ”The Web Never Forgets: Persistent Tracking Mechanisms in the Wild”, Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Pages 674-689.
[2] ”How to block fingerprinting with Firefox”, https://blog.mozilla.org/firefox/how-to-block-fingerprinting-with-firefox/
[3] ”Building a more private web”, https://www.blog.google/products/chrome/building-a-more-private-web/
[4] P. Eckersley: ”How Unique Is Your Web Browser?”, in Proceedings of the 2010 Privacy Enhancing Technologies Symposium, July 2010.
[5] D. Benjamin: ”Applying GREASE to TLS Extensibility” , https://tools.ietf.org/html/draft-ietf-tls-grease-02, Janualy 2019.
[6] M. Husák, M. Čermák, T. Jirsík, P. Čeleda: ”Network-based HTTPS Client Identification Using SSL/TLS Fingerprinting”, 2015 10th International Conference on Availability, Reliability and Security
[7] ”fingerprintjs2”, https://github.com/Valve/fingerprintjs2
[8] ”Joy”, https://github.com/cisco/joy
[9] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, G. Vigna: ”Cookieless Monster: Exploring the Ecosystem of Web-based Device Fingerprinting”, 2013 IEEE Symposium on Security and Privacy, May 2013
[10] A. Vastel, P. Laperdrix, W. Rudametkin, R. Rouvoy: ”FP-Scanner: The Privacy Implications of Browser Fingerprint Inconsistencies”, 27th USENIX Security Symposium, August 2018
[11] B. Anderson, S. Paul, D. McGrew: ”Deciphering Malware’s use of TLS（without Decryption）”, Journal of Computer Virology and Hacking Techniques, August 2018, Volume 14, Issue 3, pp 195–211
[12] S. Frolov, E. Wustrow: ”The use of TLS in Censorship Circumvention”, Network and Distributed Systems Security （NDSS） Symposium 2019 24-27 February 2019.

表 1 選定したアンチフィンガープリンティングツールの一覧

ブラウザ	ツール名	機能概要
Firefox	トラッキング防止機能	ブラウザにデフォルトで存在するトラッキング防止機能
	Canvas Blocker	JavaScript で取得される情報を偽装またはブロック
	Canvas Defender	Canvas の偽装
	ScriptSafe	フィンガープリンティングのブロック
	Canvas Fingerprint Defender	Canvas の偽装
	Spoof Timezone	タイムゾーンの偽装
	AntiBrowserSpy TrackingBlocker	フィンガープリンティングスクリプトのブロック，User-Agent の偽装
	Trace	複数のトラッキングスクリプトからの保護
	AudioContext Fingerprint Defender	AudioContext の偽装
	WebGL Fingerprint Defender	WebGL の偽装
Chrome	Canvas Fingerprint Defender	Canvas の偽装
	WebGL Fingerprint Defender	WebGL の偽装
	CanvasFingerprintBlock	Canvas のブロック
	AudioContext Fingerprint Defender	AudioContext の偽装
	Browser Plugs Fingerprint Privacy Firewall	複数のフィンガープリントの偽装
	Canvas Defender	Canvas の偽装
	ScriptSafe	フィンガープリンティングのブロック
	Canvas Blocker （Fingerprint protect）	Canvas の偽装
	JustBlock Security	広告のブロック，トラッキングスクリプトからの保護
	Random User-Agent	User-Agent の偽装
	Trace - Online Tracking Protection	複数のトラッキングスクリプトからの保護
	AntiBrowserSpy - TrackingBlocker	フィンガープリンティングスクリプトのブロック，User-Agent の偽装
	TunnelBear Blocker	広告と複数のトラッキングスクリプトのブロック
	Spoof Timezone	タイムゾーンの偽装

表 2 調査結果

ブラウザ	ツール名	ブラウザフィンガープリントの変化	TLS フィンガープリントの変化
Firefox	トラッキング防止機能	x	x
	Canvas Blocker	○	x
	Canvas Defender	○	x
	ScriptSafe	取得不可	x
	Canvas Fingerprint Defender	○	x
	Spoof Timezone	○	x
	AntiBrowserSpy TrackingBlocker	取得不可	x
	Trace	○	x
	AudioContext Fingerprint Defender	○	x
	WebGL Fingerprint Defender	○	x
Chrome	Canvas Fingerprint Defender	○	x
	WebGL Fingerprint Defender	○	x
	CanvasFingerprintBlock	○	x
	AudioContext Fingerprint Defender	○	x
	Browser Plugs Fingerprint Privacy Firewall	○	x
	Canvas Defender	x	x
	ScriptSafe	取得不可	x
	Canvas Blocker （Fingerprint protect）	○	x
	JustBlock Security	○	x
	Random User-Agent	○	x
	Trace - Online Tracking Protection	○	x
	AntiBrowserSpy - TrackingBlocker	取得不可	x
	TunnelBear Blocker	取得不可	x
	Spoof Timezone	○	x