Computer Security Symposium 2020
26 - 29 October 2020

# Androidにおける悪性Webサイトアクセスの可視化手法の提案とページ遷移分析

市岡 秀一[1]　川島 千明[2]　佐藤 将也[1]　山内 利宏[1,a)]

**概要**：Android において，利用者による操作の意図に反してリダイレクトにより，悪性 Web サイトへ誘導する攻撃が存在する．この攻撃では，遷移元サイトから複数の経由サイトへ遷移した後，悪性 Web サイトへ遷移することが多い．さらに，経由サイトでは，遷移先 URL の生成やブラウザの履歴の変更しており，攻撃において重要な役割を担っている．このため，経由サイトに着目して分析することで攻撃の特徴が明らかになる可能性がある．しかし，Web アクセスを可視化する従来手法では，良性 Web サイトの Web アクセスを可視化できるものの，上記のように経由サイトでの履歴の変更や通信を伴わない URL バーの変更などにより，悪性 Web サイトの解析が困難であった．そこで，本研究では，従来の Web アクセス可視化手法を拡張し，Android 上の悪性 Web サイトアクセスを可視化する新たな手法を提案する．この手法を用いて，既知の利用者の意図しない Web サイトへ遷移する Web サイトへアクセスし，ページ遷移を分析したところ，複数の遷移元サイトから共通の経由サイトや悪性 Web サイトへページ遷移する事例があることが明らかになった．また，ページ遷移の分析で発見された FQDN の中にスマートフォンを対象にしたユーザ参加型 Web 媒介型攻撃対策の実証実験により収集されたログに含まれる FQDN と一致する FQDN があり，遷移元サイトが異なる事例が見つかった．本稿では，ページ遷移の分析の結果を述べる．また，経由サイトや悪性 Web サイトの特徴を用いた検知手法の実現可能性を示す．

**キーワード**：Web 媒介型攻撃，利用者の意図しない Web サイト，Web セキュリティ，Android，Web ブラウザ

# Proposal of Method of Page Transition Visualization and Survey of Page Transitions in Android

Shuichi Ichioka[1]　Chiaki Kawashima[2]　Masaya Sato[1]　Toshihiro Yamauchi[1,a)]

***Abstract:*** There is an attack that redirects a user to an unwanted website on Android. In this attack, the user often transitions from landing website to multiple transit sites, and then to a malicious website. Therefore, it is possible to clarify the characteristics of the attack by focusing on the transit site. In this study, we analyzed the page transitions to unwanted websites and found that there were cases of page transitions to the same transit or malicious websites from multiple landing websites. Besides, some of the FQDNs discovered by the analysis of page transitions were found to be identical to the FQDNs in the experiment data collected by the demonstration experiment, and the landing sites were found to be different.

## 1. はじめに

Android 端末を対象とした攻撃の 1 つに利用者の意図しない Web サイト（以降，悪性 Web サイト）へ誘導する攻撃がある．この攻撃では，遷移元サイトから複数の経由サイトを経由した後，悪性 Web サイトへ遷移する．経

[1] 岡山大学 大学院自然科学研究科
Graduate School of Natural Science and Technology, Okayama University
[2] 岡山大学 工学部
Faculty of Engineering, Okayama University
[a)] yamauchi@cs.okayama-u.ac.jp

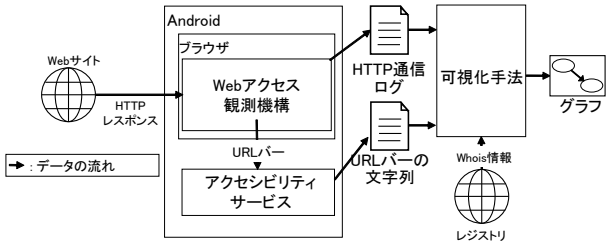


**図 1** 可視化手法のログの取得方法と可視化手法の出力

由サイトでは，悪性 Web サイトへの誘導やブラウザの履歴を変更しており，攻撃において重要な役割を担っている．また，経由サイトが存在する点は，文献 [1] で述べられている Drive-by Download 攻撃に類似している．しかし，Android 端末では，利用者の許可なくアプリケーションをインストールすることはできない．このため，Android 端末における攻撃は，偽警告画面や偽の当選懸賞画面など利用者を欺いてアプリをインストールさせることや個人情報をフォームに入力させることによって詐取することが多い．また，この攻撃では，攻撃者は遷移元サイト，経由サイトおよび悪性 Web サイトを用意する必要があり，多くの労力が必要であると推察する．しかし，経由サイトや悪性 Web サイトは頻繁に作成されている．

このため，我々は，複数の遷移元サイトから共通の経由サイトや悪性 Web サイトへ遷移するという仮説を立てた．もし，仮説が正しければ，複数の遷移元サイトからのページ遷移を分析することで，同一の経由サイトや悪性 Web サイトへ遷移する様子が観測できると推察する．また，もし，複数の遷移元サイトから同一の経由サイトや悪性 Web サイトへ遷移する事例が見つかれば，攻撃の防止に利用できる可能性がある．

本稿では，URL バーに表示された URL を利用したページ遷移の可視化手法（以降，提案手法）を提案する．この手法は，我々が [2] で提案した可視化手法を改良した手法である．次に，既知の複数の遷移元サイトから同一の経由サイトや悪性 Web サイトへ遷移する事例について述べる．最後に，遷移元サイト，経由サイト，悪性 Web サイト，および遷移を発生させるコードを含む外部ファイルの FQDN を Web 媒介型攻撃観測システムによる収集データから検索した結果を述べる．具体的には，検索の結果，収集データに含まれる FQDN と一致する FQDN が 5 件発見できた．この結果は未知の遷移元 Web サイトから悪性 Web サイトへ遷移することを防止できる可能性を示す．

## 2. 従来の Web アクセス可視化手法

### 2.1 目的と全体像

従来の Web アクセス可視化手法 [2]（以降，従来の可視化手法）は，Android における遷移元サイトから悪性 Web サイトまでのページ遷移を可視化することで分析を支援する．この手法は，読み込まれた外部ファイルや遷移の様子を可視化することで，遷移を発生させる攻撃の全容を解析者が理解するまでの時間を短縮し，経由サイトや悪性 Web サイトへ遷移を発生させるコードの特定に要する時間を減らすことを目指す．

**表 1** Web アクセス観測機構 [3] で収集可能な情報

| 項目 | 説明 |
|---|---|
| タイムスタンプ | HTTP リクエストヘッダ生成直後の時刻 |
| URL | Web コンテンツの URL |
| IP アドレス | 通信先の IP アドレス |
| HTTP リクエストヘッダ | – |
| HTTP リクエストボディ | – |
| HTTP レスポンスヘッダ | – |
| HTTP レスポンスボディ | – |

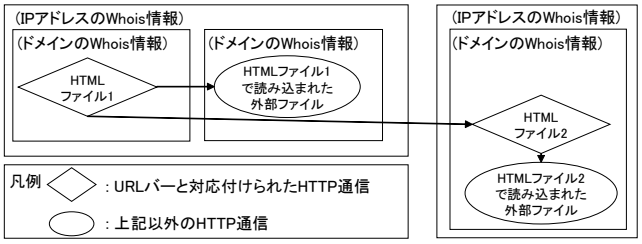


**図 2** 可視化手法の出力の概要

従来の可視化手法のログの取得方法と可視化手法の出力を**図 1** に示す．従来の可視化手法は，Android WebView における Web アクセス観測機構 [3]（以降，観測機構）を利用して収集したログ（以降，HTTP 通信のログ），およびアクセシビリティサービスを用いて収集した URL バーの文字列を利用する．観測機構で収集可能な情報を**表 1** に示す．これにより，リファラが抑制された場合に直前に URL バーに表示されていた URL が遷移元 URL である可能性が高いため，直前に URL バーに表示されていた URL に対応する HTTP 通信のログを遷移元と推定する．また，可視化手法の出力の概要を**図 2** に示す．可視化手法では，収集したログをもとに，各 HTTP 通信の遷移元を推定し，グラフでページ遷移や外部ファイルの読み込みの様子を可視化する．

また，リダイレクトが発生する悪性 Web サイトの特徴の 1 つにドメインを取得してから経過した期間を示すドメイン年齢が低い傾向がある [4]．このため，ドメイン名の Whois 情報を取得し，ドメイン取得日を収集する．これにより，膨大な HTTP 通信のログの中から攻撃に関わる可能性の高いログを効率的に見つけることができる．

### 2.2 処理の流れ

可視化手法の処理流れを以下に示す．

**(処理 1)** HTTP 通信のログに含まれるドメインと IP アドレスの Whois 情報を取得する．ドメインの Whois

情報からドメイン名，ドメイン取得日，レジストラを取得する．これにより，ドメイン年齢が低いドメインやドメインの契約で使用されたサービスが明らかになる．また，IP アドレスの Whois 情報に含まれる range（IP アドレスの範囲），descr（説明），OrgName（組織名）や Owner（所有者）を取得する．これにより，IP アドレスを管理している団体が明らかになる．

**(処理 2)** HTTP レスポンスの Location ヘッダがある場合は，Location ヘッダが指す URL を遷移先 URL と推定する．

**(処理 3)** HTTP リクエストの Referer ヘッダが指す URL を遷移元 URL として推定する．

**(処理 4)** HTTP リクエストの Referer ヘッダが抑制されたため，遷移元 URL が不明なログについて，遷移が発生する前に URL バーに表示されていた URL をもとに遷移元 URL を推定する．

**(処理 5)** 推定した遷移元 URL や遷移先 URL の関係を表したグラフを生成する．

図 2 をもとに可視化手法で出力するグラフを以下に説明する．グラフのノードは 1 つの URL に対応する HTTP 通信のログを示している．エッジは，遷移元から遷移先への流れを示す．また，内側の長方形は HTTP 通信のログに含まれるドメイン名の Whois 情報が同じノードを囲むように描画する．さらに，外側の長方形は，HTTP 通信のログに含まれる IP アドレスの Whois 情報が同じノードを囲むように描画する．図 2 は，HTML ファイル 1 が異なるドメインの外部ファイルを読み込んだ後，HTML ファイル 2 に遷移し，同じドメインの外部ファイルを読み込んでいる様子を示している．これにより，不審なドメイン名のサーバと通信していた場合に不審なノードが描画されるため，経由サイトの分析を支援できる．

## 3. Web アクセス可視化手法

本章では，従来の可視化手法を拡張した Web アクセス可視化手法について述べる．

### 3.1 従来の可視化手法の問題点

可視化手法を用いて，実際に遷移元サイトから悪性 Web サイトまでのページ遷移の分析を試みたところ，可視化手法で得られたグラフにいくつかの問題があり，分析が困難であった．可視化手法の問題点を以下に示す．

**(問題点 1)** HTTP 通信のログや可視化手法で得られたグラフのみでは，攻撃の内容を理解することが困難

HTTP 通信のログや可視化手法で得られたグラフのみでは，どのような画面が表示されているか，どのような攻撃が行われたか不明である．このため，多くの外部ファイルを読み込んでいる Web サイトや難読化された JavaScript を読み込んでいる Web サイトにおいて，HTTP 通信のログや可視化手法で得られたグラフのみでは，攻撃の内容を理解することが困難である．

**(問題点 2)** URL バーに表示される URL が変更された場合に分析が困難

JavaScript の `history.pushState()` メソッドや `history.replaceState()` メソッドにより，HTTP 通信が発生せず，URL バーの文字列や HTTP リクエストの Referer ヘッダが変更されることがある．この場合，HTTP リクエストヘッダの Referer ヘッダと遷移元 URL が異なる．このため，遷移元 URL が不正確になり，分析が困難になる．

**(問題点 3)** グラフのエッジが不正確

可視化手法では，HTTP レスポンスヘッダの Location ヘッダを利用した遷移先 URL を推定する方法，HTTP リクエストの Referer ヘッダを利用した遷移元 URL を推定する方法，および遷移の直前に URL バーに表示されていた URL を遷移元 URL として推定する方法を利用している．しかし，HTML で iframe タグが使われた場合やブラウザの新しいタブが開かれた場合に正しく遷移元を推定できない．遷移の直前に URL バーに表示されていた URL を遷移元 URL として利用する方法の信頼性が低い．

**(問題点 4)** 現実的にグラフに描画できる情報に限りがあり，分析に有用な情報の出力が困難

悪性 Web サイトへ遷移する攻撃は，PC 向けの攻撃の 1 つである Drive-by Download 攻撃に類似している．文献 [5] では，リダイレクトを含んだ特徴的な通信をもとに攻撃を検知する既存研究について述べられている．分析に有用な情報として多くの項目が提案されている．このため，多くの情報が必要である．しかし，現実的にグラフに描画できる情報に限りがあり，分析に有用な情報が可視化できない．

### 3.2 問題点への対処

以上の問題点に対処するため，可視化手法を拡張した．変更点を以下に示す．

**(変更点 1)** 収集する情報にスクリーンショットを追加

改良した可視化手法のログの取得方法を**図 3** に示す．(問題点 1) への対処として，Android OS の機能を利用して，遷移元サイト，経由サイトおよび悪性 Web サイトのスクリーンショットを収集する．これにより，出力された警告表示や Web サイトの様子を確認できるため，攻撃の内容を理解するために有用である．

**(変更点 2)** HTTP 通信が発生せず URL バーに表示される文字列が変わったことを示すノードをグラフに追加

(問題点 2) への対処として，HTTP 通信が発生せず，URL バーに表示される文字列が変わったことを示すノードを追加した．変更後のグラフのノードの

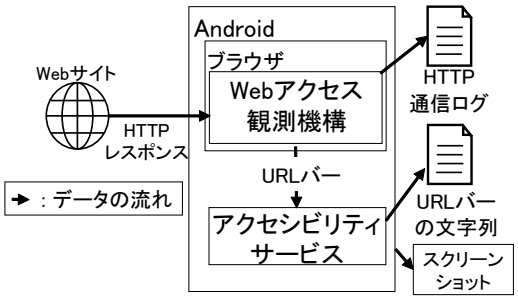


図 3　ログの収集におけるデータの流れ

表 2　ノードの形と対応するログの存在

| ノードの形 | HTTP 通信ログ | URL バーの文字列 |
|---|---|---|
| ひし形 | あり | あり |
| 楕円 | あり | なし |
| 六角形 | なし | あり |

形および対応するログの有無を**表 2** に示す．追加したノードは，HTTP 通信ひし形のノードは，発生した HTTP 通信のうち，URL バーに対応する URL が表示された HTTP 通信のログを示す．つまり，ひし形のノードは，ウィンドウオブジェクトの階層の最上位のウィンドウの URL と対応している．また，楕円のノードは，HTTP 通信のログには URL が含まれるものの，URL バーに表示されなかったことを示す．つまり，楕円のノードは，画像や JavaScript など Web ページで読み込まれる外部ファイルであることを示す．さらに，六角形のノードは，URL バーに URL が表示されたものの，HTTP 通信が発生しなかったことを示す．つまり，このノードの遷移元のノードで，URL バーや Referer ヘッダが変更されたことを示す．ここで，`history.pushState()` メソッドおよび `history.replaceState()` メソッドでは，変更後の URL は変更前と同一オリジンである必要がある [6], [7]．このため，FQDN は，遷移元のノードと同一である．これにより，URL バーの文字列や Referer ヘッダのみが変わったことをグラフに描画できる．

**(変更点 3)**　遷移元のノードを推定した方法をグラフに描画

(問題点 3) への対処として，エッジのラベルに遷移元の URL の推定方法を描画する．これは，信頼性が低いノードの解析を行う際に意識できるようにするためである．描画するエッジのラベルの説明を**表 3** に示す．これにより，グラフのエッジの信頼性を考慮した分析ができる．

**(変更点 4)**　可視化手法の出力に表を追加

(問題点 4) への対処として，スペース不足でグラフに描画できない項目を表で出力する．改良した可視化手法のデータの流れを**図 4** に示す．改良した可視化手法が表で出力する項目を**表 4** に示す．表には，グラフで描画している項目に加えて，id，URL，ステータスコード，content-type，および content-length を出力する．content-type によって，HTML ファイル，画像ファイル，CSS ファイル，JavaScript ファイルといったファイルの中がどのようなものか判別するために使用する．また，content-length は，content-type に虚偽の内容を含む場合に，正しくないことが分かる．　たとえば，JavaScript ファイルであるにも関わらず，content-length が 4 といった小さい数字の場合，JavaScript のコードを記述することは困難であるため，JavaScript では無いとみなすことができる．これにより，攻撃に関係のない HTTP 通信のログを分析する時間を短縮できる．

表 3　エッジのラベル

| エッジのラベル | 説明 |
|---|---|
| location | HTTP レスポンスの Location ヘッダにより，遷移元を推定したことを示す． |
| accessibility_service | 六角形のノードの遷移元 URL を直前に表示されていた URL と断定したことを示す． |
| referer | HTTP リクエストの Referer ヘッダにより，遷移元を推定したことを示す． |
| accessibility | URL バーに表示させた URL により，遷移元を推定したことを示す． |

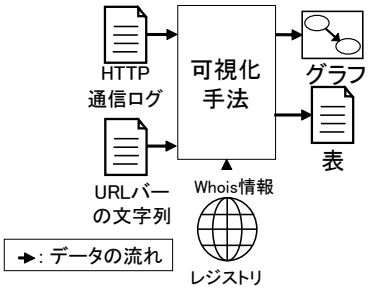


図 4　改良した可視化手法のデータの流れ

### 3.3　拡張した可視化手法による出力例

拡張した可視化手法で出力されるグラフを**図 5** に示す．図 5 では，id が 1 のノードから id が 3 である外部ファイルが読み込まれている．また，URL バーに表示される URL が変更された後，id が 4 の経由サイトに遷移している．その後，経由サイトで URL バーに表示される URL が変更された後，id が 6 の悪性 Web サイトへ遷移している．また，経由サイトや悪性 Web サイトに遷移することを示すエッジのラベルがすべて `accessibility` になっている．これは，すべての遷移で HTTP リクエストの Referer ヘッダが抑制されたことを示している．

表 4　改良した可視化手法で出力する表の項目

| 項目 | 説明 |
|---|---|
| id | グラフのノードと共通のログの識別番号 |
| タイムスタンプ | HTTP 通信のログのタイムスタンプ |
| URL | HTTP 通信のログに対応する URL または，URL バーに表示された URL |
| Location | HTTP レスポンスの Location ヘッダが示す URL |
| Referer | HTTP リクエストの Referer ヘッダが示す URL |
| ステータスコード | HTTP レスポンスのステータスコード |
| content-type | HTTP レスポンスヘッダの content-type ヘッダの情報 |
| content-length | HTTP レスポンスヘッダの content-length ヘッダの情報 |
| ノードの形 | グラフで描画されているノードの形（「ひし形」,「楕円」または「六角形」を格納） |
| IP アドレスの Whois 情報 | IP アドレスをもとに取得した Whois 情報の name および description を格納 |
| ドメイン名の Whois 情報 | ドメイン名をもとに取得した Whois 情報のレジストラおよびドメインの取得日を格納 |
| 遷移元 | 遷移元 URL に対応するログの id |
| 遷移元の推定方法 | グラフで描画されたエッジのラベルと同一 |

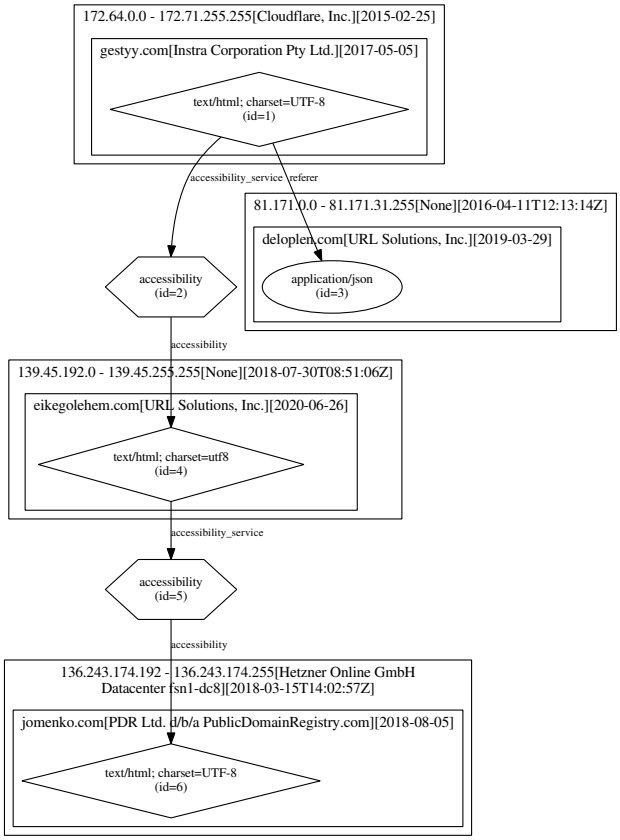


図 5　可視化手法で出力されるグラフ（抜粋）

### 3.4　期待される効果

拡張した可視化手法により，以下の効果が期待できる．

**(効果 1)**　画面表示の可視化

拡張した可視化手法では，スクリーンショットを収集する．これにより，出力された警告表示や Web サイトの様子を確認できる．スクリーンショットは，攻撃の内容を理解するために有用である．

**(効果 2)**　HTTP リクエストの Referer ヘッダの変更を追跡可能

拡張した可視化手法では，URL バーの文字列の変更を可視化する．また，遷移元のノードの推定に URL バーに表示された URL を利用する．これにより，`history.pushState()` メソッドや `history.replaceState()` メソッドによって URL バーに表示される URL や HTTP リクエストの Referer ヘッダが変更された場合に，遷移元を正しく推定できる．

**(効果 3)**　エッジの信頼性を判別可能

拡張した可視化手法では，エッジのラベルに遷移元のノードの推定方法を描画する．これにより，ノードの推定方法の信頼性を考慮した分析が可能になる．

**(効果 4)**　出力できる情報の増加

拡張した可視化手法では，グラフの他に表を出力する．これにより，可視化手法が出力できる情報が増加した．

## 4.　経由サイトの分析

### 4.1　目的

遷移元サイトから悪性 Web サイトに遷移するまでに多くの経由サイトを経由する．経由サイトでは，悪性 Web サイトで戻るボタンを使えないようにした上で，悪性 Web サイトへ遷移するといった挙動が確認されており，攻撃において重要な役割を担っている．このため，経由サイトを分析することで攻撃の特徴が明らかになる可能性がある．また，多くの遷移元サイトが存在するため，複数の遷移元サイトから共通の経由サイトや悪性 Web サイトへの遷移が確認できる可能性がある．この分析では，複数の遷移元サイトから共通の経由サイトや悪性 Web サイトへの遷移が無いかを分析する．

以上の目的を満たすために行う分析は以下の 3 つである．

(1) 異なる遷移元サイトや経由サイトから同一の経由サイ

表 5　URL の収集期間，収集された URL の数および Web アクセスのログの収集日

| | |
|---|---|
| [8] の手法で URL が収集された期間 | 2020/6/11 – 2020/7/10 |
| 収集された URL の数 | 48 個 |
| ユニークな FQDN の数 | 18 個 |
| 既知の遷移元サイトから<br>ログを収集した日 | 2020/7/29 |

表 6　Web アクセスのログを収集した環境

| | |
|---|---|
| 端末 | Android Emulator |
| OS | Android 6.0 (Marshmallow) |
| WebView | Chromium 60.0.3094.2 に観測機構 [3] を追加 |
| ブラウザ | Browser 6.0-6352195 |

トへ遷移した場合および攻撃に関わる外部ファイルを比較する．

(2) 攻撃者が作成した遷移元サイト，経由サイト，悪性 Web サイト，および遷移が発生するコードを含む外部ファイルのネットワーク構成を明らかにする．

(3) 攻撃に関わる Web サイトの IP アドレス，Web ページの名前および URL を比較する．

### 4.2　分析方法

#### 4.2.1　可視化手法を用いた HTTP 通信のログの収集および可視化

既知の遷移元サイトの URL として，文献 [8] の手法によって発見された遷移元サイトの URL を利用する．文献 [8] では，クローラを用いて Web 空間から収集した大量の HTML ファイルについて，既知の悪性 Web サイトから抽出したキーワードを用いて悪性である可能性が高い HTML ファイルを検索することで効率的にモバイル向けの悪性 Web サイトを発見する手法が提案されている．発見され，収集された URL に改良した可視化手法を適用する．URL の収集期間，収集された URL の数，Web アクセスのログの収集日を**表 5** に示す． また，Web アクセスのログを収集した環境を**表 6** に示す．

#### 4.2.2　攻撃に関わる HTTP 通信のログの抽出

改良した可視化手法を用いて，遷移元サイト，経由サイト，悪性 Web サイトおよび遷移を発生させるコードを含む外部ファイルのページ遷移の様子および外部ファイルの読み込みの様子を全て可視化する．改良した可視化手法の出力をもとに攻撃に関わる HTTP 通信のログを抽出する．抽出する手順を以下に示す．

**(手順 1)**　悪性 Web サイトの HTTP 通信のログを取り出す．

**(手順 2)**　最後に取り出した HTTP 通信のログに対応する URL を読み込むコードまたは遷移させるコードを探索する．

**(手順 3)**　遷移するコードを含む HTTP 通信のログを取り出す．

**(手順 4)**　遷移元サイトを取り出すまで（手順 2）と（手順 3）を繰り返す．

（手順 2）で JavaScript のコードを分析する際，見つかったコードが必ず，実行されるとは限らない．文献 [9] では，HTTP レスポンスボディに遷移先の URL と同じ URL が発見された場合，遷移が発生したと推定している．この文献を参考に，直接，遷移を発生させると考えられるコード以外にも，遷移が発生するコードを含む文字列が発見された場合も遷移を発生させるコードであるとみなす．これは，遷移先 URL を含む遷移を発生するコードがある場合，遷移を発生させるコードが実際に実行された可能性が高いためである．また，生成される URL に乱数を含むものがあるため，URL と思われる文字列を分析し，一部が遷移先の URL と一致した場合，遷移が発生したとみなす．

#### 4.2.3　リダイレクトの様子を示すグラフの作成

複数の遷移元サイトから同一の経由サイトや悪性 Web サイトへ遷移している様子を可視化する．可視化手法の出力で得られるグラフの概形を**図 6** に示す．図 6 のような可視化手法のグラフをそのまま結合した場合，1 つの遷移元サイトから複数の悪性 Web サイトへ分岐したのか，異なる遷移元サイトから同一の経由サイトに遷移した後，異なる悪性 Web サイトへ遷移したのか判別できなくなる．このため，可視化手法で得られるグラフを**図 7** のように変形して，1 つの遷移元サイトから悪性 Web サイトまでの遷移を分岐させずに 1 本で表現する．改良した可視化手法で得られたグラフを変形し，ノードの形および FQDN が一致するノードを同一のノードで描画する．このとき，自己ループしているエッジは削除している．また，遷移元サイトを長方形で描画し，最終的に遷移する悪性 Web サイト

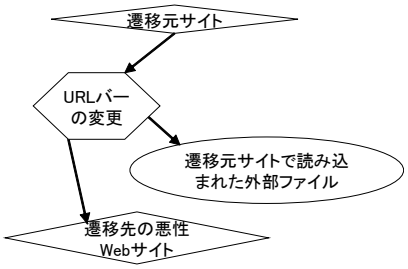


図 6　可視化手法で出力されるグラフの概形

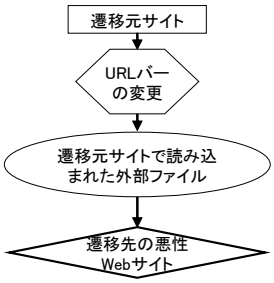


図 7　変形後のグラフの概形

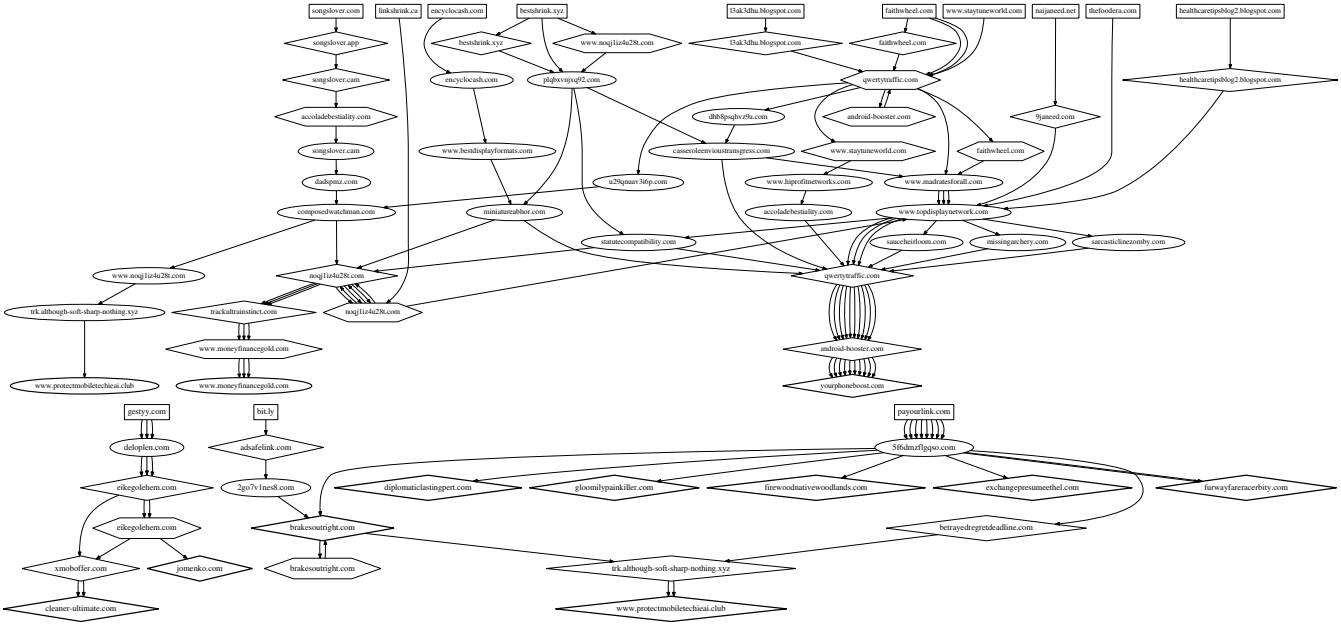


**図 8** 遷移元サイトから悪性 Web サイトまでのページ遷移を可視化したグラフ

に対応するノードを太線で描画する．

### 4.3 分析結果

分析対象の URL 48 件のうち，悪性 Web サイトに遷移する 26 件のログを分析した．また，可視化手法で生成したグラフを**図 8** に示す．長方形のノードは，遷移元サイトの FQDN を示す．

グラフは，3 つの連結グラフであった．図 8 の上半分に，10 個の異なる FQDN を持つ遷移元サイトから 3 件の悪性 Web サイトに遷移したことを示す．また，左下の連結グラフは，1 つの遷移元サイトから 2 つの悪性 Web サイトへ遷移していることを示す．さらに，右下の連結グラフは，2 種類の遷移元サイトから 7 つの悪性 Web サイトへ遷移したことを示す．

これらは，一部で共通の経由サイトを経由する．よって，複数の遷移元サイトから同一の悪性 Web サイトへ遷移する事例がある．この結果は，一部の経由サイトや悪性 Web サイトを発見することにより，他の遷移元サイトから行われる攻撃を発見できる可能性を示す．

## 5. 分析で得られた FQDN について Web 媒介型攻撃観測システムによる収集データの探索

図 8 に出現する遷移元サイトから悪性 Web サイトまでのすべての FQDN 56 個を Web 媒介型攻撃観測システム [10] による収集データから探索した．利用したログは，2020 年 7 月 1 日から 2020 年 7 月 31 日までに実験参加者から収集データが格納されるサーバにアップロードされたログである．このログの概要を**表 7** に示す．

**表 7** Web 媒介型攻撃観測システムによる収集データの概要

| | |
|---|---|
| ログの件数 | 1,015,690 |
| ユニークユーザ数 | 590 |
| ユニークな FQDN の数 | 30313 |

**表 8** 基盤から見つかった FQDN とこの FQDN にアクセスしたユニークユーザ数

| FQDN | ログの件数 | ユニークユーザ数 |
|---|---|---|
| bit.ly | 161 | 88 |
| yourphoneboost.com | 16 | 11 |
| cleaner-ultimate.com | 8 | 3 |
| eikegolehem.com | 2 | 1 |
| jomenko.com | 2 | 1 |

この結果，5 個の FQDN が収集データに含まれる FQDN と一致した．一致した FQDN と FQDN にアクセスしたユニークユーザ数を**表 8** に示す．`bit.ly` は，URL 短縮サービスであるため，攻撃の意図はないと推察する．また，`jomenko.com` は，ドメインの取得日が 2018 年であった．その他のドメインの取得日は，2020 年である．2020 年 4 月 22 日にドメインが取得された悪性 Web サイトの FQDN である `yourphoneboost.com` に 590 人中 11 人のユニークユーザがアクセスしている．これは，取得されたばかりのドメインの悪性 Web サイトであるにも関わらず，1.8%のユニークユーザが遷移元サイトから新しい悪性 Web サイトへ利用者が誘導されていることを示す．また，2020 年 7 月 11 日にドメインが取得された悪性 Web サイトの FQDN である `cleaner-ultimate.com` では，590 人中 3 人（0.5%）のユニークユーザがアクセスしている．これにより，新し

い悪性 Web サイトへ誘導するように遷移元サイトや経由サイトのコードを書き換えることが短期間で行われていると言える．

今後は，これらのログを実際に確認し，新たな遷移元サイトや悪性 Web サイトが見つかるかを調べる．

## 6. 関連研究

文献 [11] では，同じ Web サイトや似た Web サイトへ誘導する複数の Web サイトを示すグラフを生成し，このグラフを用いて機械学習により分類器を作成している．ここで，よく似た Web サイトとは，以下のいずれかの条件を満たす Web サイトのことである．

(1) ドメイン名，Web ページのタイトルおよび URL に含まれるパラメータが一致
(2) TLD (Top-level Domain)，Web ページのタイトルおよび URL に含まれるパラメータが一致
(3) ドメイン名の文字数，TLD，Web ページのタイトルおよび URL に含まれるパラメータが一致
(4) IP アドレスが一致

文献 [11] のページの遷移の流れをもとにグラフを生成する点は類似しているものの，文献 [11] は，Android を対象としていない．

文献 [12] では，HTTP 通信の解析，HTML の解析，および呼び出された JavaScipt API の観測結果を利用して，遷移の流れをグラフで表現している．また，分析に必要な情報の収集にブラウザエミュレータの HtmlUnit を使用している．この研究は，Android を対象にしていない．

文献 [13] では，多くの Web サイトで外部のリソースを読み込んでいることに着目し，疑わしいリソースを読み込んでいる Web サイトの割合を調査している．この調査の結果，73%の Web サイトで疑わしい外部のリソースを読み込んでいたと述べている．この研究では，実際に攻撃が行われる事例を調査しているわけでは無い．しかし，疑わしいリソースが攻撃を始めた場合，これらの Web サイトへアクセスした人が危険にさらされることを示している．

## 7. おわりに

既知の遷移元サイトの URL を利用して，遷移元サイトから悪性 Web サイトまでのページ遷移を分析した．これにより，複数の遷移元サイトから共通の経由サイトや悪性 Web サイトへ遷移していることが明らかになった．また，遷移元サイト，経由サイト，悪性 Web サイト，および遷移を発生させるコードを含む外部ファイルを格納していた Web サイトの FQDN をモバイル向けの Web 媒介型攻撃観測システムによる収集データから探索したところ，5 種類の FQDN が分析で見つかった FQDN と一致した．

**謝辞** 本研究成果は，国立研究開発法人情報通信研究機構 (NICT) の委託研究「Web 媒介型攻撃対策技術の実用化に向けた研究開発」により得られたものです．

### 参考文献

[1] Shibahara, T., Yagi, T., Akiyama, M., et al.: POSTER: Detecting Malicious Web Pages Based on Structural Similarity of Redirection Chains, *Proc. 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS'15)*, pp.1671–1673, (2015).

[2] 市岡秀一，佐藤将也，山内利宏：Android WebView における利用者の意図しない悪性 Web サイトへの Web アクセス可視化手法，2020 年電子情報通信学会総合大会 情報・システム講演論文集 2，p.117 (2020).

[3] Imamura, Y., Uekawa, H., Ishihara, Y., Sato, M. and Yamauchi, T.: Web Access Monitoring Mechanism for Android Webview, *Proc. Australasian Computer Science Week Multiconference*, No.1 (2018).

[4] 木村匡，佐々木良一：ドメイン情報の分析による Drive by Download 攻撃の対策の提案，情報処理学会マルチメディア，分散協調とモバイルシンポジウム 2016 論文集，Vol.2016，pp.1705–1710 (2016).

[5] 高田真資，高橋健一，川村尚生，菅原一孔：Drive-by-Download 攻撃検知項目の評価，情報処理学会コンピュータセキュリティシンポジウム 2017 (CSS2017) 論文集，Vol.2017，No.2，pp.811–816 (2017).

[6] MDN: History.pushState(), available from ⟨`https://developer.mozilla.org/ja/docs/Web/API/History/pushState`⟩(accessed 2020–08–18).

[7] MDN: History.replaceState(), available from ⟨`https://developer.mozilla.org/ja/docs/Web/API/History/replaceState`⟩(accessed 2020–08–18).

[8] 石原聖，折戸凜太郎，佐藤将也，山内利宏：モバイル向け悪性 Web サイトの探索によるブラックリスト構築手法，情報処理学会コンピュータセキュリティシンポジウム 2019 (CSS2019) 論文集，Vol.2019，pp.1025–1032 (2019).

[9] Nelms, T., Perdisci, R., Antonakakis, M., Ahamad, M.: WebWitness: Investigating, Categorizing, and Mitigating Malware Download Paths, *Proc.24th USENIX Security Symposium* (*USENIX Security 15*), pp.1025–1040, USENIX Association(2015).

[10] 山田明，佐野絢音，窪田歩，嶌田一郎，中嶋淳，吉岡克成，瀬尾浩二郎，満保雅浩，佐藤将也，松村礼央，田辺瑠偉，小澤誠一，田中翔真，梅本俊，松田壮，山内利宏，澤谷雪子：スマートフォンにおける Web 媒介型サイバー攻撃の観測機構：設計と実装，2020 年暗号と情報セキュリティシンポジウム（SCIS2020）論文集，電子媒体 (2020).

[11] Stringhini, G., Kruegel, C., Vigna, G.: Shady paths: leveraging surfing crowds to detect malicious web pages, *Proc. 2013 ACM SIGSAC conference on Computer & communications security* (*CCS' 13*), pp.133–144, Association for Computing Machinery(2013).

[12] Takata, Y., Akiyama, M., Yagi, T., Yada, T. and Goto, S.: Fine-Grained Analysis of Compromised Websites with Redirection Graphs and JavaScript Traces, *IEICE Transactions on Information and Systems*, Vol.E100.D, No.8, pp.1714–1728 (2017).

[13] Ikram, M., Masood, R., Tyson, G., Kaafar, A.M., Loizon, N., Ensafi, R.: Measuring and Analysing the Chain of Implicit Trust: A Study of Third-party Resources Loading, *ACM Trans. Priv. Secur.*, Vol.23, No.2(2020).