Vol.2021-DPS-186 No.44
Vol.2021-CSEC-92 No.44
2021/3/16

Androidにおける遷移元Webサイトから利用者の意図しないWebサイトまでの通信内容の分析

川島 千明[1]　市岡 秀一[2]　山内 利宏[2]

概要：Android において，利用者の意図に反して，偽警告画面や偽の懸賞当選画面などを表示する悪性 Web サイトへ誘導する攻撃が存在する．この攻撃では，利用者が遷移元 Web サイトにアクセスした際，自動または画面の任意の場所のタップを契機として経由 Web サイトへ遷移した後，悪性 Web サイトまで遷移する．また，悪性 Web サイトへの遷移に関わる Web サイトおよび外部ファイルの特徴として，URL が短期間で変更される点が挙げられる．このため，URL のブラックリストによる対策は難しく，ブラックリストに依存しない悪性 Web サイトへの対策が必要である．そこで，本稿では，遷移元 Web サイトから悪性 Web サイトまでの Web アクセスのログを分析し，悪性 Web サイトまでの誘導方法，および遷移の原因となったソースコードについて述べる．また，同じ遷移元 Web サイトにおける時間経過による変化について分析し，新たに発見した誘導方法や，繰り返し使われているソースコードの事例について述べる．

Analysis of HTTP Response between Landing Website and Malicious Website in Android

Chiaki Kawashima[1]　Shuichi Ichioka[2]　Toshihiro Yamauchi[2]

1. はじめに

スマートフォンなどのモバイル端末が普及している現代において，モバイル端末の利用者数は，世界人口の 67%に達したと報告されている [1]．また，2020 年には世界中の Web トラフィックの約 51%がモバイル端末から発生している [2]．モバイル端末が普及するにつれて，モバイル端末を対象としたマルウェアが増加しており，特に Android 端末を対象としたマルウェアが多く流行している [3]．

攻撃者がモバイル端末をマルウェアに感染させる攻撃の 1 つに，利用者を欺くことで，利用者自らマルウェアに感染する行動をとらせる手口がある．この手口では，リダイレクトにより，出会い系サイトや偽の警告画面を表示する Web サイトといった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．一部の利用者は，この悪性 Web サイトの指示に従って信憑性の低いアプリケーションをモバイル端末にインストールしている可能性がある．

IPA が公表した「情報セキュリティ 10 大脅威 2020」[4] によると，「不正アプリによるスマートフォン利用者への被害」が 6 位，「偽警告によるインターネット詐欺」が 9 位になっており，不正アプリや偽警告による攻撃の被害が大きいことが分かる．また，不正アプリをインストールさせる方法として，偽の警告画面からの誘導や不正な Web サイトからの誘導が挙げられている [5]．この他に，会員制の Web サイトへの入会を促す画面や出会い系サイトなどへ遷移する事例が報告されている [6]．以上より，悪性 Web サイトから不正アプリのインストールや，個人情報の詐取などにつながる可能性があるため，悪性 Web サイトの対策が必要である．

悪性 Web サイトの対策として，ブラックリストを用いる手法がある．しかし，悪性 Web サイトへの遷移に関わる Web サイトの特徴として，ドメインが取得されてからの

[1] 岡山大学 工学部
Faculty of Engineering, Okayama University
[2] 岡山大学 大学院自然科学研究科
Graduate School of Natural Science and Technology, Okayama University

経過期間を表すドメイン年齢が低いことが挙げられる [7]. 悪性 Web サイトでは短期間で新しいドメインが利用されるため，ブラックリストなどを用いた対策が難しくなっている．

そこで，本研究では，既知の遷移元 Web サイトに対して，2020 年 7 月 29 日に 1 回目の Web アクセスを行った．得られた Web アクセスログから，悪性 Web サイトへの遷移に関わった Web サイト，およびこれらの Web サイトで読み込まれる外部ファイルのソースコードを分析し，悪性 Web サイトへの誘導方法を明らかにする．誘導方法については，遷移元 Web サイトからの遷移の契機が「自動」であるか「画面タップ」であるかによって分類し，それぞれの結果を述べる．ここで，「自動」とは，遷移元 Web サイトにアクセスした際に，利用者の操作なしで経由 Web サイトへ遷移する場合を指す．また，「画面タップ」とは，遷移元 Web サイトにアクセスした際に，利用者の画面のタップにより経由 Web サイトへ遷移する場合を指す．

また，1 回目の Web アクセスから約 5 か月後の 2021 年 1 月 4 日に同じ遷移元 Web サイトに対して Web アクセスを行い，悪性 Web サイトへの遷移に関わる Web サイトおよび外部ファイルの時間経過による変化について分析した．この分析で発見した時間経過による誘導方法の変化や，繰り返し利用されているソースコードについて述べる．

2. 分析する Web アクセスログの収集方法

2.1 Android における悪性 Web サイトアクセス可視化手法

Android における悪性 Web サイトアクセス可視化手法 [8] では，Android における遷移元 Web サイトから悪性 Web サイトまでの通信内容を収集し，ページ遷移を可視化することで分析を支援する．遷移元 Web サイトから悪性 Web サイトまでの通信内容を収集するために，文献 [8] の手法を用いる．この手法を用いた通信内容の収集におけるデータの流れを**図 1** に示す．

この手法では，Android Emulator に WebView の Web アクセス観測機構 [9] をインストールすることにより，遷移元 Web サイトから悪性 Web サイトへアクセスするまでのすべての URL と HTTP 通信の通信内容を収集する．Android Emulator および Web アクセス観測機構の環境を**表 1** に示す．また，Android のアクセシビリティサービスを用いて URL バーの文字列を取得する．HTTP 通信のログと URL バーの文字列を組み合わせることにより，Web サイトが履歴を変更したことや，悪性 Web サイトへ遷移したことが判別できる．さらに 1 秒ごとにスクリーンショットを取得することで，偽警告画面の表示や偽の懸賞当選画面の表示を確認する．

この Android における悪性 Web サイトアクセス可視化手法を用いて HTTP 通信の通信内容を収集することで，悪性 Web サイトへの遷移に関わった通信内容を収集できる．この収集した通信内容を用いて，遷移元 Web サイト，経由 Web サイト，および悪性 Web サイトにおいて，繰り返し使われる共通のソースコード，および誘導方法の特徴を分析する．

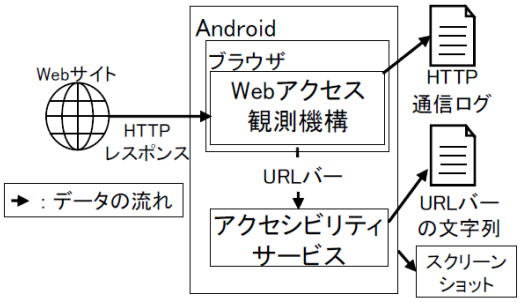

図 1 通信内容の収集におけるデータの流れ

表 1 Web アクセス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 [9] を追加
ブラウザ	Browser 6.0-6352195

表 2 URL の収集期間，収集された URL の数および Web アクセスログ収集日

[10] の手法で URL が収集された期間	2020 年 6 月 11 日–2020 年 7 月 10 日
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 Web サイトから Web アクセスログを収集した日 (1 回目)	2020 年 7 月 29 日
既知の遷移元 Web サイトから Web アクセスログを収集した (2 回目)	2021 年 1 月 4 日

2.2 通信内容の収集

分析対象の Web サイトとして，モバイル向けブラックリスト構築手法 [10] により，Twitter に投稿された URL をもとに構築されたブラックリストに登録された URL を利用した．このブラックリストに登録された URL に対し，悪性 Web サイトアクセス可視化手法を用いてアクセスし，Web アクセスログを収集した．URL の収集期間，収集された URL の数，および Web アクセスログ収集日を**表 2** に示す．既知の遷移元 Web サイトから Web アクセスログを収集した日（1 回目）と既知の遷移元 Web サイトから Web アクセスログを収集した日（2 回目）の間で経過した期間は 5 カ月である．

3. 遷移元 Web サイトから利用者の意図しない Web サイトへの誘導方法の分析

3.1 分析の目的

本研究の目的は，悪性 Web サイトへの誘導において，複

数の Web サイトで共通するソースコード，および誘導方法の特徴を発見することである．これにより，URL のブラックリストに依存しない悪性 Web サイトの対策に利用できる可能性がある．

また，発見した共通するソースコードおよび誘導方法が短期間で変更されるか否かを分析するため，悪性 Web サイトへの遷移に関わる Web サイトの時間経過による変化を分析した．これにより，悪性 Web サイトへの遷移に関わる Web サイトの特徴を明らかにすることを目的とする．

3.2 分析方法

2 章に示した Web アクセスログには，遷移元 Web サイトから悪性 Web サイトまでの遷移の過程で読み込まれる全ての HTTP 通信のログが含まれる．このため，悪性 Web サイトへの誘導において，攻撃に関わった Web アクセスログの抽出を行った．抽出する手順を以下に示す．

(1) 悪性 Web サイトの HTTP 通信のログを取り出す．
(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．
(3) (2) でコードを含む HTTP 通信のログを取り出す．
(4) 遷移元 Web サイトを取り出すまで (2) と (3) を繰り返す．

この手順により抽出した Web アクセスログに含まれる遷移元 Web サイトから悪性 Web サイトまでのソースコードについて，利用者の端末情報の取得，外部ファイルの読み込み，および Web サイトの読み込みを行っている部分を中心に誘導方法を分析した．

文献 [8] の手法では，遷移元 Web サイトにアクセスした後，15 秒後に画面のタップ操作を行っている．このため，取得したスクリーンショットにおいて，遷移元 Web サイトにアクセスしてから 15 秒以内に遷移が発生している Web サイトの遷移の契機を「自動」，15 秒後に遷移が発生している Web サイトの遷移の契機を「画面タップ」として分類する．

また，攻撃に関わった外部 JavaScript ファイルにおいて，ソースコードの難読化が行われている場合があった．この場合は，手作業により難読化の解除を行い，コード整形ツールを用いて整形している．

4. 悪性 Web サイトへの誘導方法

4.1 分析の観点

2020 年 7 月 29 日の Web アクセスログ 48 件において，遷移元 Web サイトから利用者の意図しない遷移が発生した Web アクセスログは 26 件であった．なお，利用者の意図しない遷移とは，以下の 2 つの場合を指す．

- 遷移元 Web サイトにアクセスした際，自動で経由 Web サイトに遷移した場合
- 利用者が操作できる場所と認識していない部分の画面タップにより経由 Web サイトに遷移した場合

表 3 2020 年 7 月 29 日の Web アクセスログ 48 件の分類結果

利用者の意図しない遷移	あり			なし
遷移の契機	自動	画面タップ		
外部ファイル	JavaScript		PHP	
件数	13 件	10 件	3 件	22 件
ユニークな FQDN 数	9 件	3 件	1 件	5 件

表 4 2020 年 7 月 29 日の Web アクセスログにおける悪性 Web サイトの種類

偽警告	クリーナー	偽懸賞当選	WebView の強制終了
10 件	4 件	3 件	7 件

26 件の Web アクセスログすべてで，遷移元 Web サイトで読み込まれる外部ファイルにより遷移が発生していた．このことから，遷移元 Web サイトからの遷移の契機の違いは，外部ファイルの記述の違いによるものであると推察される．このため，悪性 Web サイトへの誘導方法について，以下の 3 つの観点から分析結果を示す．

(1) 遷移元 Web サイトからの遷移の契機が「自動」の場合
(2) 遷移元 Web サイトからの遷移の契機が「画面タップ」の場合
(3) 時間経過による変化

また，**表 3** に 2020 年 7 月 29 日の Web アクセスログ 48 件の分類結果を示す．遷移元 Web サイトから利用者の意図しない遷移が発生した 26 件の Web アクセスログのうち，遷移元 Web サイトから経由 Web サイトへ「自動」で遷移する Web サイトが 13 件，「画面タップ」により遷移する Web サイトが 13 件であった．

なお，遷移元 Web サイトで読み込まれる外部ファイルの URL のパスの終端が.php の場合，外部ファイルは PHP により記述されているとみなす．PHP のプログラムはサーバサイドで実行されるため，ソースコードを確認することができない．このため，本分析においては JavaScript で記述された外部ファイルのみを分析の対象としている．

26 件の Web アクセスログにおいて遷移した悪性 Web サイトの種類を**表 4** に示す．表 4 中の「偽警告」は偽の警告画面を表示する Web サイト，「クリーナー」はクリーナーアプリのインストールを勧める Web サイト，「偽懸賞当選」は偽の検証当選画面を表示する Web サイト，および「WebView の強制終了」は Web アクセスの途中で WebView が強制終了した場合を指す．

4.2 遷移元 Web サイトからの遷移の契機が「自動」の場合

遷移元 Web サイトから「自動」で遷移する Web サイトについて，悪性 Web サイトまでの遷移の流れの例を**図 2** に示す．この事例では，遷移元 Web サイトにおいて，外部 JavaScript ファイル (A)，外部 JavaScript(B) が順に読み

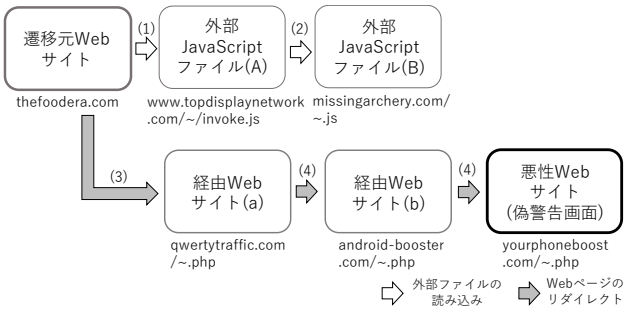

図 2 遷移元 Web サイトから「自動」で遷移する悪性 Web サイトまでの流れ

```
<script type="992914f13565dde8acf8725a-text/javascript">
    atOptions = {
        'key' : '145089d485336ff04a61703813393e4a',
        'format' : 'iframe',
        'height' : 90,
        'width' : 728,
        'params' : {}
    };
    document.write('<scr' + 'ipt type="text/javascript"
    src="http' + (location.protocol === 'https:' ? 's' : '')
    + '://www.topdisplaynetwork.com/145089d485336ff04a61703813393e4a
    /invoke.js"></scr' + 'ipt>');
</script>
```

図 3 遷移元 Web サイトから外部 JavaScript ファイル (A) を読み込む部分

込まれ，外部 JavaScript ファイル (B) によって遷移元 Web サイトから経由 Web サイト (a)，経由 Web サイト (b)，悪性 Web サイトへ順に遷移していた．

遷移元 Web サイトから「自動」で悪性 Web サイトまで遷移していた 13 件のログにおいて図 2 の外部 JavaScript ファイル (A) および外部 JavaScript ファイル (B) のソースコードはすべて一致した．ここで，ソースコード中の URL およびユーザの識別に用いられる値のみが異なる場合も一致していると判断した．このことから，異なる FQDN で同じソースコードが用いられていることが分かった．

また，遷移の契機が「自動」の場合における遷移元 Web サイトから悪性 Web サイトまでの処理流れを以下の (1)–(5) に示す．なお，以下の (1)–(4) は図 2 中の (1)–(4) と対応している．

(1) 遷移元 Web サイトにおける外部 JavaScript ファイル (A) の読み込み
図 2 の遷移元 Web サイトは，ニュースサイトのような Web サイトであった．遷移元 Web サイトにおいて，外部 JavaScript ファイル (A) を読み込むソースコードを**図 3** に示す．
外部 JavaScript ファイル (A) を読み込む部分では，以下の記述が見られた．

```
'format'='iframe'
'height':90,'width':728
```

記述に見られる 728 × 90 というサイズは，バナー広告で一般的に用いられるサイズである [11]．このため，この部分ではバナー広告用のインラインフレームを作成するスクリプトで外部 JavaScript ファイル (A) が読み込まれていると推察される．このことから，遷移元 Web サイトの広告が攻撃に利用された可能性がある．

(2) 外部 JavaScript ファイル (A) における外部 JavaScript ファイル (B) の読み込み
外部 JavaScript ファイル (A) の処理流れを**図 4** に示す．また，図 4 中に示す各処理について，**表 5** に詳細を示す．なお，外部 JavaScript ファイル (B) の URL を `JS_B` と表記している．
図 4 では，(i)，(ii)，(iii) の処理が順に実行され，(iii) が成功した場合は JavaScript ファイル (B) が読み込まれる．(iii) がエラーとなり，ブラウザの幅や高さが 0 でない場合は (iv-a) の処理，ブラウザの幅や高さが 0 の場合には (iv-b) の処理が実行される．これにより，外部 JavaScript ファイル (B) の読み込みが行われる．

(3) 外部 JavaScript ファイル (B) における経由 Web サイト (a) の読み込み
外部 JavaScript ファイル (B) では，`window.top.location` に経由 Web サイト (a) の URL を指定したスクリプトを，script タグを動的に生成して遷移元 Web サイトに書き込んでいた．このスクリプトが実行されることにより，遷移元 Web

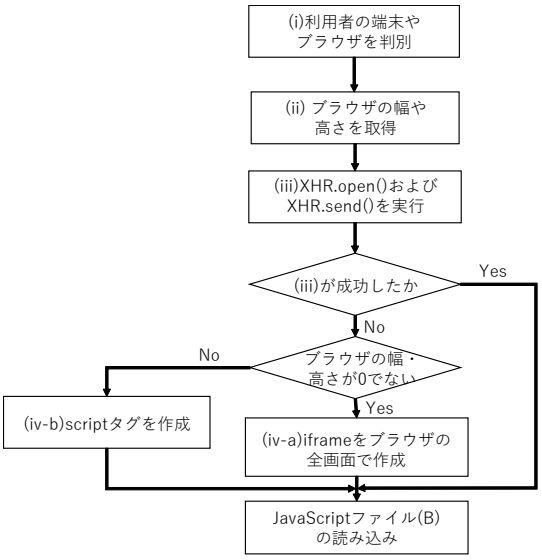

図 4 外部 JavaScript ファイル (A) の処理流れ

表 5 外部 JavaScript ファイル (A) の処理の詳細

	処理	使用するプロパティおよびソースコード例
(i)	利用者の端末や ブラウザを判別	NavigatorID.userAgent NavigatorID.platform
(ii)	ブラウザの幅や 高さを取得	window.screen.width window.screen.height
(iii)	XHRを用いて 非同期読み込み	XMLHttpRequest.open(“GET”, JS_B); XMLHttpRequest.send();
(iv-a)	インラインフレームを 作成しJSを読み込み	iframe = document.createElement(“iframe”); iframe.src = JS_B;
(iv-b)	scriptタグを作成し JSを読み込み	script = document.createElement(“script”); script.src = JS_B;

```
1   <p>ご使用の携帯
2   <script>document.write(getURLParameter('device_name'))</script>
3   の動作を加速させることができます。Cleaner のアップデートがリリース
4   され、すべての携帯に推奨されています。</p>
5
6   <p>アップデートを行わないと、ご使用の
7   <script>document.write(getURLParameter('device_name'))</script>
8   は動作が重くなり、電池の持ち時間が短くなる可能性があります。</p>
9
10  <p>今すぐ無料でアップデートして、直ちにご使用の
11  <script>document.write(getURLParameter('device_brand'))</script>
12  のクリーンアップとブーストを行ってください！</p>
```

図 5 悪性 Web サイトのソースコードの一部

サイトから経由 Web サイト (a) への遷移が発生する．

(4) 経由 Web サイト (a) から悪性 Web サイトまでの読み込み

この部分では，HTTP の Location ヘッダにより遷移していた．また，経由 Web サイト (a) および経由 Web サイト (b) の URL のパスの終端は.php であった．このため，経由 Web サイト (a) および経由 Web サイト (b) は PHP で記述されており，PHP の header 関数でリダイレクト処理が行われたと推察している．

(5) 悪性 Web サイト

悪性 Web サイトのソースコードの一部を**図 5** に示す．図 5 の 2，7，11 行目において，デバイス名やブランド名を URL のパラメータから取得し，偽警告文に表示していた．なお，悪性 Web サイトの URL のパラメータには，外部 JavaScript ファイル (A) により取得した情報を用いていると推察される．

4.3 遷移元 Web サイトからの遷移の契機が「画面タップ」の場合

本節では，遷移元 Web サイトから「画面タップ」で遷移する 13 件の Web アクセスログのうち，外部 JavaScript ファイルが原因で悪性 Web サイトに遷移した 10 件の Web アクセスログの分析結果について示す．

「画面タップ」で遷移する遷移元 Web サイトの画面のスクリーンショットを**図 6** に示す．この遷移元 Web サイトは短縮 URL を提供する Web サイトのような見た目であった．遷移元 Web サイトにおいて，画面左端中央部分の見た目ではリンクの存在しない部分をタップしたところ，経由 Web サイトに遷移した．この事例における悪性 Web サイトまでの遷移の流れの例を**図 7** に示す．この事例では，遷移元 Web サイトにおいて，外部 JavaScript ファイル (C) が読み込まれ，外部 JavaScript(C) によって遷移元 Web サイトから経由 Web サイト (c)，経由 Web サイト (d)，悪性 Web サイトへ順に遷移していた．

また，10 件のログにおいて，図 7 の外部 JavaScript ファイル (C) のソースコードは一致した．ここで，ソースコード中の URL およびユーザの識別に用いられる値のみが異なる場合も一致していると判断した．このことから，遷移元 Web サイトから「自動」で遷移する場合と同様に，異なる FQDN で同じソースコードが用いられていることが分かった．

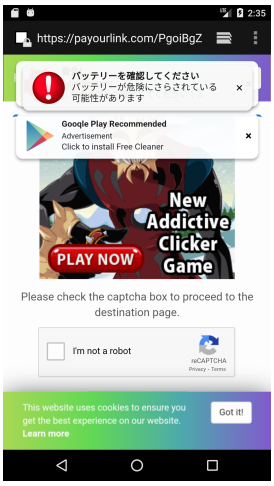

図 6 画面タップで遷移する遷移元 Web サイト

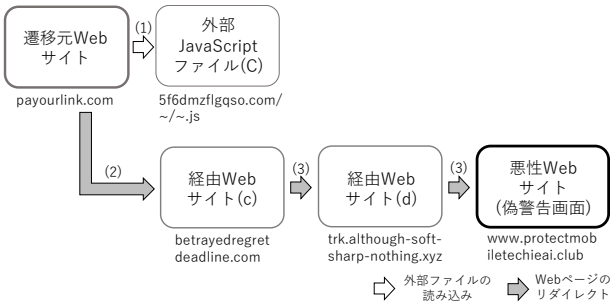

図 7 遷移元 Web サイトから「画面タップ」で遷移する悪性 Web サイトまでの流れ

遷移の契機が「画面タップ」の場合における遷移元 Web サイトから悪性 Web サイトまでの処理流れを以下の (1)–(4) に示す．なお，以下の (1)–(3) は図 7 中の (1)–(3) と対応している．

(1) 遷移元 Web サイトから外部 JavaScript ファイル (C) の読み込み

遷移元 Web サイトにおいて外部 JavaScript(C) を読み込む部分では，script タグの src 属性により，外部 JavaScript ファイル (C) が読み込まれていた．

(2) 外部 JavaScript ファイル (C) から経由 Web サイト (c) の読み込み

外部 JavaScript ファイル (C) の処理流れを**図 8** に示す．また，図 8 中の各処理について，**表 6** に詳細を示す．なお，経由 Web サイト (c) の URL を keiyu_c と表記している．

図 8 では，(i)，(ii)，(iii) の処理が順に実行され，(iii) で画面タップなどの動作を取得した際に (iv-a) の処理が実行される．また，(iii) で画面タップなどの動作が行われず，利用者がブラウザの「戻る」ボタンや「閉じる」ボタンを押下した場合，(iv-b) の処理が実行される．これにより，利用者が画面の任意の部分のタッ

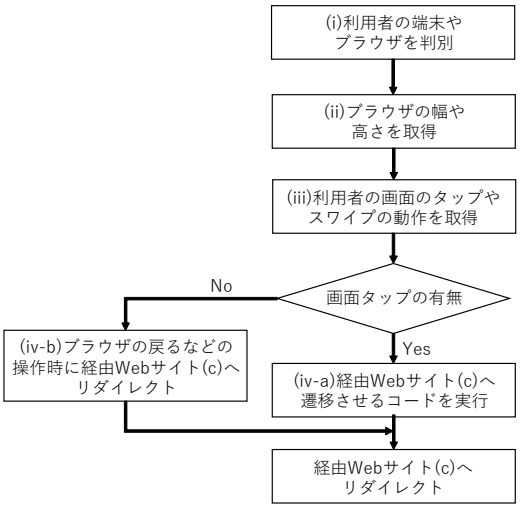

図 8 外部 JavaScript ファイル (C) の処理流れ

表 6 外部 JavaScript ファイル (C) の処理の詳細

	処理	使用するプロパティおよびソースコード例
(i)	利用者の端末や ブラウザを判別	NavigatorID.userAgent
(ii)	ブラウザの幅や 高さを取得	window.screen.width window.screen.height
(iii)	利用者の画面の タップやスワイプ の動作を取得	document.addEventListenerのイベント タイプに"touch"および"swipe"を指定
(iv-a)	(iii)の動作発生時に 経由Webサイト(c)へ 遷移するコードを実行	window.location.href = keiyu_c
(iv-b)	ブラウザの戻るなどの 操作時に経由Webサイ ト(c)へリダイレクト	window.onbeforeunload = function(){ window.location.href = keiyu_c }

プ，またはブラウザの「戻る」や「閉じる」操作を行った際，強制的に経由 Web サイト (c) へリダイレクトが発生する．

(3) 経由 Web サイト (c) から悪性 Web サイトまでの読み込み
経由 Web サイト (c) および経由 Web サイト (d) には，HTTP レスポンスボディが存在しなかった．また，HTTP レスポンスヘッダのコンテンツタイプはどちらも HTML ファイルであった．このことから，`.htaccess` などを用いた HTTP のリダイレクトが発生したと推察している，

(4) 悪性 Web サイト
悪性 Web サイトにおいては，`window.alert` により偽警告画面が表示された．また，悪性 Web サイトのソースコードから，偽警告のポップアップを閉じた場合，以下の動作が実行されることが分かった．

- 画面に新たな偽警告のメッセージを表示
- "Remove Virus Now"と書かれたボタンを表示（クリックで同ドメインの異なる Web ページへ遷移）
- 5 秒間隔のバイブレーション
- mp3 ファイルの自動再生
- カウントダウンタイマの表示

表 7 2021 年 1 月 4 日の Web アクセスログ 48 件の分類結果

利用者の意図しない遷移	あり			なし
遷移の契機	自動	画面タップ		31 件
外部ファイル	JavaScript		PHP	
件数	5 件	10 件	2 件	
ユニークな FQDN 数	3 件	4 件	1 件	10 件

表 8 2021 年 1 月の Web アクセスログ (17 件) における悪性 Web サイトの分類

クリーナー	動画再生	WebView の強制終了
7 件	3 件	7 件

なお，ブラウザの履歴も変更されるため，偽警告のポップアップ画面，およびポップアップ画面を閉じた後の画面では「戻る」ボタンを押下しても前のページに戻ることは出来ない．

4.4 時間経過による変化

4.4.1 時間経過による変化の分析

表 7 に 2021 年 1 月 4 日に取得した Web アクセスログ 48 件の分類結果を示す．表 7 において，悪性 Web サイトへの遷移を確認できたログは 17 件であった．すべての遷移元 Web サイトにおいて，読み込まれる外部ファイルが原因で遷移しており，遷移元 Web サイトからの遷移の契機が「自動」の Web サイトが 5 件，「画面タップ」の Web サイトが 12 件であった．

また，表 7 の「画面タップ」を遷移の契機として外部 JavaScript ファイルにより遷移する 10 件の遷移元 Web サイトのうち，3 件は 2020 年 7 月の Web アクセスログで遷移が発生していない Web サイトであった．このことから，悪性 Web サイトへの遷移に関わる遷移元 Web サイトにおいて，攻撃に利用した Web サイトを利用しない状態に変更し，一定期間後に再び利用している事例を確認できた．

2021 年 1 月 4 日に取得した Web アクセスログで見られた悪性 Web サイトの種類による分類を**表 8** に示す．悪性 Web サイトは，クリーナーアプリのインストールを勧める Web サイトの他に，動画再生画面のような見た目を表示する Web サイトが新たに存在した．クリーナーアプリのインストールを勧める Web サイトは，4.2 節で示した悪性 Web サイトとソースコードがほぼ一致し，FQDN が変更されていた．これにより，悪性 Web サイトにおいて，同じソースコードを用いて新たな FQDN を持つ Web サイトが作成されていることが確認できた．

以上の分析結果より，悪性 Web サイトへの遷移に関わる Web サイトにおいて，遷移の原因となる外部 JavaScript ファイルは，同じソースコードが長期間において使われていることが明らかになった．

4.4.2 新たな誘導方法の分析結果

動画再生画面のような見た目を表示する Web サイトへ

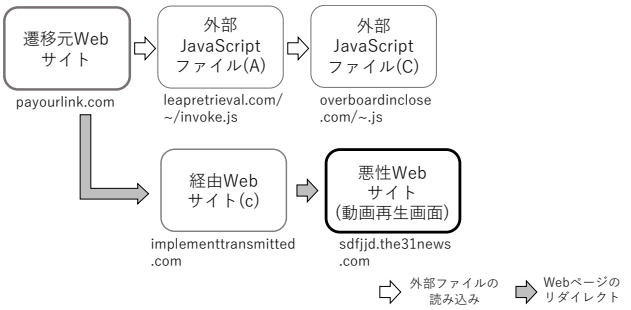

図 9 新たに発見した誘導方法の遷移の流れ

遷移した 3 件の Web アクセスログのうち 1 件において，4.2 節および 4.3 節で述べた 2 種類の誘導方法とは異なる新たな誘導方法が確認できた．新たに確認した誘導方法における遷移元 Web サイトから悪性 Web サイトまでの遷移の流れを**図 9** に示す．この事例では，遷移元 Web サイトから外部 JavaScript ファイル (A)，外部 JavaScript ファイル (C) が順に読み込まれ，外部 JavaScript ファイル (C) により，遷移元 Web サイトから経由 Web サイト (c) へ遷移していた．次に，経由 Web サイト (c) により，動画再生画面のような見た目を表示する悪性 Web サイトへ遷移していた．

図 9 で示した外部 JavaScript ファイル (A)，外部 JavaScript ファイル (C)，および経由 Web サイト (c) はそれぞれ図 2 の外部 JavaScript ファイル (A)，図 7 の外部 JavaScript ファイル (C)，および経由 Web サイト (c) と，変数名や URL の記述部分を除きソースコードが一致した．また，この事例では，遷移元 Web サイトからの遷移の契機が「画面タップ」であった．このことから，外部 JavaScript ファイル (C) は，外部 JavaScript ファイル (A) により自動で読み込まれ，外部 JavaScript ファイル (C) により「画面タップ」を契機として遷移元 Web サイトから遷移が発生したと考えられる．

5. 関連研究

文献 [12] では，Google 検索からスマートフォン等の当選詐欺サイトまで誘導される攻撃の誘導方法について調査している．文献 [12] では，Google 検索により表示された遷移元 Web サイトにアクセスすると，複数の経由 Web サイトを経由したのちに，当選詐欺の画面を表示する悪性 Web サイトへ遷移することが述べられている．また，遷移元 Web サイト内の JavaScript コードにより，遷移元 Web サイトのドキュメントに `script` タグが書き込まれることにより，JavaScript ファイルが新たに読み込まれることが述べられている．しかし，文献 [12] は，Android を対象としていない．また，時間経過による遷移元 Web サイト，経由 Web サイトおよび悪性 Web サイトの変化について，分析していない．一方で，本研究では，Twitter に投稿された URL をもとに遷移元 Web サイトを探索し，Android Emulator を用いて収集した HTTP 通信の通信内容を分析した．また，時間経過による遷移元 Web サイト，経由 Web サイトおよび悪性 Web サイトの変化について分析した．

文献 [13] では，Android アプリケーションで多く利用されている WebView における偽警告画面を表示する Web サイト，フィッシングサイト，およびコインマイニングを行う Web サイトについて脅威分析を行っている．この研究では，悪性 Web サイトにおいて，経由 Web サイト間の遷移に `window.location.href` が利用されていることが述べられているものの，遷移元 Web サイトから経由 Web サイトへの遷移の原因や，異なる遷移元 Web サイトにおいて用いられる共通のソースコードについて分析していない．一方で，本研究では，遷移元 Web サイトから悪性 Web サイトまでの遷移の全過程について，原因となったソースコードを示した．また，異なる遷移元 Web サイトにおいて共通のソースコードが利用される事例について分析した．

文献 [14] 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．この研究は，異なる Flash Player の環境によって攻撃に使用される脆弱性が異なることを示しているものの，JavaScript の静的な分析や時間経過による遷移元 Web サイト，経由 Web サイトおよび悪性 Web サイトの変化について分析していない．一方で，本研究では，JavaScript を静的に分析し，JavaScript のコードに記述された挙動を明らかにした．また，時間経過による遷移元 Web サイト，経由 Web サイトおよび悪性 Web サイトの変化について分析した．

6. おわりに

本稿では，48 件の遷移元 Web サイトの URL に対して Web アクセスのログを取得し，悪性 Web サイトへ遷移した 26 件のログにおける悪性 Web サイトへの誘導を分析した．これにより，遷移元 Web サイトからの遷移は，外部ファイルが原因となっていることを明らかにした．また，外部ファイルの記述により，遷移元 Web サイトからの遷移の契機が「自動」と「画面タップ」の 2 つに分類されることを述べた．そこで，遷移元 Web サイトからの遷移の契機で Web アクセスのログを分類し，悪性 Web サイトまで遷移する原因となったソースコードを示した．ここで，遷移の契機ごとに，異なる FQDN を持つ外部 JavaScript ファイルにおいて，ソースコードが一致することを確認した．

また，同じ遷移元 Web サイトに対して，約 5 か月後に再度 Web アクセスを行い，悪性 Web サイトへの遷移に関わる Web サイトの時間経過による変化を分析した．遷移に関わる Web サイトおよび外部ファイルのほとんどは，変数名や URL 記述部分などを除き，約 5 ヶ月前の Web アクセスログに含まれるソースコードと一致した．これにより，悪性 Web サイトへの誘導に関わる外部ファイルや Web サ

イトは，同じソースコードが長期間において用いられていることを述べた．また，攻撃に利用した Web サイトを利用しない状態に変更し，一定期間後に再び利用している事例が存在することや，1 件の新たに発見した誘導方法について述べた．

今後の課題として，悪性 Web サイトへの対策として利用できるソースコードの特徴や，ファイル名に見られる共通点などについて調査することが挙げられる．

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] DataReportal: Digital 2020: Global Digital Overview(online), available from ⟨https://datareportal.com/reports/digital-2020-global-digital-overview⟩ (accessed 2021-02-03).
[2] Clement, J: Percentage of mobile device website-traffic worldwide from 1st quarter 2015 to 2nd quar-ter 2020 (online), Statista, available from ⟨https://www.statista.com/statistics/277125/share-of-website-traffic-coming-from-mobile-devices/⟩ (accessed 2021-02-03).
[3] LIARNA LA PORTA：4 ways hackers are inltrating phones with malware on Android phones, wandera, 入手先 ⟨https://www.wandera.com/malware-on-android/⟩（参照 2021-01-26）.
[4] IPA：情報セキュリティ 10 大脅威 2020, 入手先 ⟨https://www.ipa.go.jp/security/vuln/10threats2020.html⟩（参照 2021-01-21）.
[5] カスペルスキー: スマートフォンを狙う不正アプリの脅威, 入手先 ⟨https://home.kaspersky.co.jp/store/kasperjp/ja_JP/html/pbPage.article-06/ThemeID.37143200/⟩（参照 2021-01-21）.
[6] 利穂虹希，折戸凜太郎，佐藤将也，山内利宏：Android を対象とした利用者の意図しない Web サイトの分類，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.1011–1016 (2019).
[7] 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，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).
[8] 市岡秀一，川島千明，佐藤将也，山内利宏：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，コンピュータセキュリティシンポジウム 2020 (CSS2020) 論文集，Vol.2020，pp.551–558 (2020).
[9] Imamura, Y., Orito, R., Uekawa, H., Chaikaew, K., Leelaprute, P., Sato, M. and Yamauchi, T.: Web access monitoring mechanism via Android WebView for threat analysis, *International Journal of Information Security* (2021).
[10] 石原 聖, 佐藤将也, 山内利宏：悪性 Web サイトの探索によるモバイル向けブラックリスト構築手法の実証実験データによる評価，コンピュータセキュリティシンポジウム 2020 (CSS2020) 論文集, Vol.2020, pp.21–28 (2020).
[11] 中釜啓太：【2021 年最新】GDN・YDN バナーサイズ一覧, UNIAD，入手先 ⟨https://www.uniad.co.jp/220101⟩（参照 2021-02-03）.
[12] 白井優武，岡澤佳寛，三谷和也，小林晴貴，徳永 渉，齊藤泰一：Google 検索結果から当選詐欺サイトへのリダイレクトチェーンの調査，暗号と情報セキュリティシンポジウム (SCIS2021), 電子媒体 (2021).
[13] 今村祐太，折戸凜太郎，Kritsana Chaikaew，Celia Manardo，Pattara Leelaprute，佐藤将也，山内利宏：Android における WebView の Web アクセス観測機構を利用した悪性 Web サイトの脅威分析と対策の提案，コンピュータセキュリティシンポジウム 2018 (CSS2018) 論文集，Vol.2018，pp.137–144 (2018).
[14] 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).