Computer Security Symposium 2018
22 - 25 October 2018

MITB攻撃においてコンテンツ改ざんを行う不正JavaScriptの解析手法

高田 一樹[1,2,a)]　松本 英樹[2]　邦本 理夫[2]　吉岡 克成[3]　松本 勉[3]

概要：近年，インターネットバンキング等のオンラインサービス利用者をターゲットとした，Man-In-The-Browser（MITB）攻撃による被害が社会問題となっている．MITB 攻撃は，マルウェアによってブラウザの通信内容の盗取・改ざんを行う攻撃方法であり，攻撃対象サイトのコンテンツを改ざんすることで，情報盗取や不正送金などが行われる．MITB 攻撃によるコンテンツ改ざんには，改ざんのための不正な JavaScript が用いられる．改ざんの実態を明らかにするためには，この不正 JavaScript の詳細を明らかにする必要があるが，難読化やコード量が膨大等の影響で静的な解析が困難なものが多く，デバッガ等を用いて動的に解析する必要がある．しかし，マルウェア感染環境下でオンラインサービスへ接続し不正 JavaScript の解析を行うことは，該当オンラインシステムへ何らかの悪影響を及ぼす等のリスクがある．本稿では，安全に改ざんの実態を明らかにするため極力マルウェア本体を用いることなく，不正 JavaScript の収集および動作を解析する手法の提案を行う．

キーワード：マルウェア，MITB，JavaScript，動的解析

Analysis Method of Malicious JavaScript that tampers Web Contents in MITB attack

Kazuki Takada[1,2,a)]　Hideki Matsumoto[2]　Michio Kunimoto[2]　Katsunari Yoshikoka[3]
Tsutomu Matsumoto[3]

1. はじめに

近年，インターネットバンキング等のオンラインサービス利用者をターゲットとした，Man-In-The-Browser（MITB）攻撃による被害が社会問題となっている．MITB 攻撃は，マルウェアが感染 PC の Web ブラウザに対し，コードインジェクション等の方法で入り込み，通信内容の監視や改ざんを行う攻撃手法である．この MITB 攻撃により，認証情報の盗取やインターネットバンキングにおける不正送金等の被害が発生する．現在，日本国内においても，Ursnif や DreamBot と呼ばれる MITB 攻撃を行うマルウェアの流行が確認されている [1][2].

本稿では，これらの MITB 攻撃により，インターネットバンキング等で不正送金を行うマルウェアを金融系マルウェアと呼称する．Ursnif 等の金融系マルウェアによる MITB 攻撃では，Web ブラウザが攻撃対象のサイトとの通信時に通信内容に含まれる Web コンテンツを改ざんすることで，入力フォームの改ざんや偽の入力画面の表示等が発生する．このコンテンツ改ざんには，情報盗取や不正送金を行うための機能を持つ不正な JavaScript（以下，MITB 攻撃用 JavaScript）が用いられる．MITB 攻撃におけるコンテンツ改ざんの実態を明らかにするためには，この MITB 攻撃用 JavaScript の機能を明らかにする必要が

[1] 横浜国立大学大学院環境情報学府
Graduate School of Environment and Information Sciences, Yokohama National University
[2] 株式会社セキュアブレイン
SecureBrain Corporation
[3] 横浜国立大学大学院環境情報研究院/先端科学高等研究院
Graduate School of Environment and Information Sciences, Yokohama National University / Institute of Advanced Sciences
[a)] takada-kazuki-hw@ynu.jp

ある．しかし，MITB 攻撃用 JavaScript は，難読化されているものやコード量が多いもの等が多く，静的に解析することは困難である．そこで，MITB 攻撃用 JavaScript を動的解析する必要があるが，金融系マルウェアに感染した環境でオンラインサービスに接続し解析を実施することは，該当オンラインシステムへ悪影響を及ぼすリスクや感染環境を別の攻撃の踏み台にされる危険性がある．また，MITB 攻撃用 JavaScript の解析において Web ブラウザのデバッガ機能等を用いる際に，マルウェア感染の影響で正しい解析が行えない等の問題が発生する恐れがある．また，複数の金融機関を攻撃対象にする金融系マルウェアや複数の金融系マルウェアに同時期に攻撃対象にされている金融機関への攻撃を効率的に解析する上で，マルウェア感染環境を適切に維持することは非常に手間である．さらに，マルウェアの取扱に不慣れな JavaScript 解析者がマルウェア感染環境を用いて MITB 攻撃用 JavaScript の解析を行うことは，リスクを伴うと共に解析者の精神的な負担も大きい．

そこで本稿では，MITB 攻撃用 JavaScript と攻撃対象サイトのダミー環境を用いて MITB 攻撃によるコンテンツ改ざんを再現するシステムを用いた MITB 攻撃用 JavaScript の解析手法について提案する．また，MITB 攻撃用 JavaScript を容易に収集する方法についても提案する．これらの手法を用いて，2018 年 7 月現在において攻撃を行っている金融系マルウェア 2 検体を用いて実験を行った．この結果，提案手法が MITB 攻撃におけるコンテンツ改ざんを行う MITB 攻撃用 JavaScript の解析に有効であることを示す．本研究の貢献を以下に示す．

- MITB 攻撃におけるコンテンツ改ざんに用いられる MITB 攻撃用 JavaScript の解析手法の提案を行ったこと．
- MITB 攻撃用 JavaScript の収集方法の提案を行ったこと．
- 提案手法を用いることで，金融系マルウェア本体を用いずに MITB 攻撃用 JavaScript の解析を可能としたこと．

本稿の構成は，以下の通りである．まず，2 章で，関連研究について記述する．3 章で，MITB 攻撃について記述する．4 章で，提案手法について記述する．5 章で，評価実験について記述する．最後に，6 章で，まとめと今後の課題について記述する．

2. 関連研究

関連研究について述べる．MITB 攻撃の実態調査の研究として，Rahimian らの研究 [3] がある．研究 [3] では，金融系マルウェアの静的解析手法および MITB 攻撃の実態について明らかにしている．MITB 攻撃に関して，Web ブラウザに対する金融系マルウェアによるインジェクションの手法や攻撃対象の情報等については明らかにしているが，コンテンツ改ざんや MITB 攻撃用 JavaScript については述べられていない．Boutin の研究 [4] では，MITB 攻撃におけるコンテンツ改ざんおよび MITB 攻撃用 JavaScript について詳細な調査がされている．しかし，コンテンツ改ざんを解析する手法に関しては，提案されていない．

金融系マルウェアを動的解析し，MITB 攻撃を調査する手法として，Continella らの動的解析システム Prometheus [5] がある．Prometheus は，金融系マルウェアの動的解析を行い，MITB 攻撃によるコンテンツ改ざん時の DOM 情報の変化を収集・分析するシステムである．このシステムは，コンテンツ改ざん時の DOM の変化を取得することを目的としているが，本研究では，改ざん後のコンテンツを操作した際の MITB 攻撃用 JavaScript を解析することを目的としており異なっている．

本研究に類似する研究として，瀬川らの研究 [6] がある．研究 [6] は，ダミーコンテンツを設定したサーバに金融系マルウェアに感染したマシンで接続することで MITB 攻撃の動的解析を行うシステムである．研究 [6] は，金融系マルウェアの MITB 攻撃の動的解析を目的としているが，本研究では，MITB 攻撃におけるコンテンツ改ざんに用いられる MITB 攻撃用 JavaScript の解析をマルウェアを用いずに解析することを目的としており異なる．また，研究 [6] では，攻撃対象の可能性のあるダミーコンテンツを複数用意し，感染マシンと通信を行うことで攻撃対象を特定する方法をとっている．このため無駄なダミーコンテンツの生成や攻撃対象に漏れが生じる可能性がある．これに対し，我々の提案手法では，予め攻撃対象を特定した上で解析を実施する点で優位性がある．

Web のコンテンツ改ざん時に用いられる不正 JavaScript の動的解析手法に関する研究には，柴田らの Js-Walker[7] や上川らの研究 [8] がある．これらは，いずれも難読化等の処理をされ Web コンテンツに埋め込まれた不正 JavaScript の解析に有用なシステムである．しかし，いずれも Drive By Download を引き起こす Exploit Kit に用いられる不正 JavaScript を対象としている．本研究では，MITB 攻撃によるコンテンツ改ざんで用いられる MITB 攻撃用 JavaScript を対象としており，解析の対象および目的が異なっている．

3. MITB 攻撃

MITB 攻撃について述べる．論文 [9] によると，MITB 攻撃は，認証情報等の盗取を目的とした ID 盗取型 MITB 攻撃と利用者が実行した送金処理の内容をリアルタイムで改ざんする取引内容改ざん型 MITB 攻撃の 2 種類に分類することができる．本稿における MITB 攻撃とは，ID 盗取型 MITB 攻撃を指す．

金融系マルウェアによる MITB 攻撃の概要を**図 1** に示す．MITB 攻撃は，金融系マルウェアを制御するコマン

表 1　Ursnif 攻撃設定情報の構成

構成要素名	内容
URL	攻撃対象 URL
src	改ざん対象文字列
dst	挿入コード片

ド・アンド・コントロールサーバ（以下，C&C サーバ）と MITB 攻撃用 JavaScript の配信や盗取情報の収集をするマニピュレーションサーバと言った外部サーバと連携して実行される．MITB 攻撃は，C&C サーバから取得した攻撃設定情報に従って行われる．攻撃設定情報とは，攻撃対象および改ざん方法等の攻撃方法を金融系マルウェアに設定するための情報である．本稿で実験に用いた Ursnif のコンテンツ改ざんを行う攻撃設定情報の例を**図 2** に示す．図 2 は，Ursnif の保持する暗号化された攻撃設定情報を複合し解析を行った結果である．攻撃設定情報の構成を**表 1** に示す．

MITB 攻撃に用いられる不正 JavaScript は，2 種類存在する．攻撃設定情報の挿入コード片に含まれ MITB 攻撃による改ざんで正規コンテンツに挿入される不正 JavaScript と，挿入された不正 JavaScript によってマニピュレーションサーバから取得される情報盗取や偽画面の表示等を行う機能を持つ不正 JavaScript である．本来これらを総称して MITB 攻撃用 JavaScritp と定義すべきであるが，本稿では，マルウェアによって挿入される不正 JavaScript を挿入コード片，情報盗取や偽画面の表示等を行う機能を持つ不正 JavaScript を MITB 攻撃用 JavaScript と呼称する．

我々は，文献 [10] の調査結果から MITB 攻撃を以下のステップに分割する．

Step 0. 感染：スパムメール，不正ウェブサイト等から金融系マルウェアがユーザ PC に感染する．

Step 1. 攻撃設定情報ダウンロード：金融系マルウェアは，外部の C&C サーバと通信を行い攻撃設定情報を取得する．

Step 2. Web ブラウザの通信監視：金融系マルウェアは，Web ブラウザにコードインジェクション等の方法で入り込み通信を監視する．

Step 3. 正規コンテンツの改ざん：ユーザが Web ブラウザを使用して攻撃設定情報に指定された攻撃対象 URL に接続をした際に，攻撃設定情報に従って正規コンテンツを改ざんして挿入コード片を挿入する．

Step 4. MITB 攻撃用 JavaScript の読込：Step 3 で正規コンテンツに挿入された挿入コード片が実行されることで，MITB 攻撃用 JavaScript がマニピュレーションサーバから取得され Web ブラウザ上に読み込まれる．

Step 5. ログイン情報の盗取・自動送金：MITB 攻撃用 JavaScript によって，偽画面の表示や入力された認証情報の盗取または，ユーザ PC 上で意図しない送金が発生する．

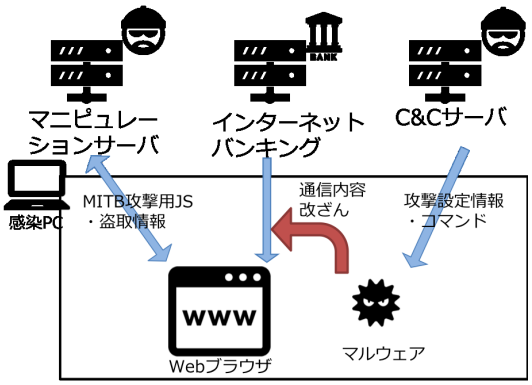

図 1　MITB 攻撃の概要

```
replace:
  URL: https://                                                    .js
  src: softpop = false;
  dst: softpop = false;(function(){function d(b){var c="/iimg
c/?c=script&r=softkey-pers&b="+encodeURIComponent("@ID@"),a=w
indow.XMLHttpRequest?new XMLHttpRequest:new ActiveXObject("Mi
crosoft.XMLHTTP");a.onreadystatechange=function(){4==a.readyS
tate&&200==a.status&&b(a.responseText)};a.open("GET",c);a.sen
d()}function e(){d(function(b){try{-1!=b.indexOf("%SERVER_URL
%")&&eval(b.replace(/%SERVER_URL%/g,"/iimgc/"))}catch(c){}})}
try{e()}catch(f){};})();
```

図 2　Ursnif の攻撃設定情報

4. 提案手法

3 章の MITB 攻撃のステップのうち Step 3 ～ 5 を再現することで，コンテンツ改ざんに利用される MITB 攻撃用 JavaScript の解析を行う手法を提案する．提案手法は，以下の 3 段階で構成される．

(1) 攻撃設定情報の分析
(2) MITB 攻撃用 JavaScript 収集
(3) MITB 攻撃用 JavaScript の動的解析

提案手法の全体概要を**図 3** に示す．4.1～4.3 に各段階の詳細について述べる．また，提案手法で用いるコンテンツ改ざん再現システム（以下，改ざん再現システム）を構築した．システムの概要を**図 4** に，システムの構成を**表 2** にそれぞれ示す．

ダミーサイトサーバ：攻撃対象サイトのダミーコンテンツを応答する Web サーバである．ダミーサイトサーバは，改ざん再現ルールに従って，ダミーコンテンツを動的に改ざんして応答する．改ざん再現ルールは，攻撃設定情報の分析結果に従って作成する．改ざん再現ルールの詳細を 4.1.1 に示す．改ざん再現ルールには，複数の改ざん方法が設定可能であり，Web ブラウザからアクセスする際の URL にパラメータを設定することで，改ざんの有無および種類を切り替えることを可能とする．また，Sinatra の after フィルタ [11] を利用して，配信するコンテンツに対し文字列の置換・挿入を行う機能（以下，文字列置換・挿入機能）を有している．

ダミーマニピュレーションサーバ：ダミーマニピュレーションサーバは，MITB 攻撃用 JavaScript を取得する通信を模擬し，サーバ内に設定した MITB 攻撃用 JavaScript

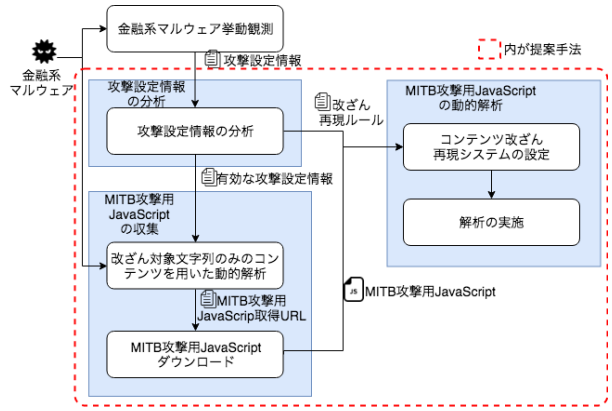

図 3　提案手法の概要

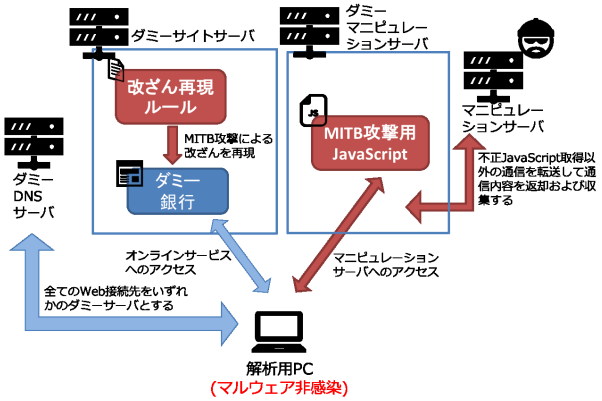

図 4　コンテンツ改ざん再現システム

表 2　コンテンツ改ざん再現システムの構成

ダミーサイトサーバ および ダミーマニピュレーションサーバ	Ruby 2.0 Sinatra 1.4.5 thin 1.6.1
ダミー DNS サーバ	dnsmasq 2.79

による応答を行う．MITB 攻撃用 JavaScript 取得以外のマニピュレーションサーバへの通信は，実際のマニピュレーションサーバへ転送し，応答を通信元に返却する．ダミーサイトサーバと同様に配信コンテンツに対する文字列置換・挿入機能を有している．

ダミー DNS サーバ：DNS クエリに対してダミーサイトサーバもしくは，ダミーマニピュレーションサーバの IP アドレスを返却する．解析用 PC の DNS サーバとしてダミー DNS サーバを設定して使用する．

解析用 PC：Web ブラウザを使用してダミーサイトサーバへ接続し，MITB 攻撃用 JavaScript の解析を行う．本稿の実験において使用した解析用 PC の構成を**表 3** に示す．

4.1　攻撃設定情報の分析

攻撃設定情報を分析することで攻撃対象および攻撃方法の情報を入手する．我々は，静的解析によって金融系マルウェアの詳細な機能を明らかにした上で，対象検体を挙動観測する調査手法 [10] を用いて金融系マルウェアの攻撃設定情報を収集している．この収集した攻撃設定情報を分析の対象とする．攻撃設定情報の分析では，攻撃対象の特定と改ざん対象文字列を含むコンテンツの存在を確認する．その後，有効な改ざん対象が存在する攻撃対象の改ざん再現ルールを作成する．改ざん再現ルールの詳細を，4.1.1 に示す．

4.1.1　改ざん再現ルール

改ざん再現ルールは，攻撃設定情報に指定された攻撃対象 URL，改ざん対象文字列，挿入コード片を元に設定する．改ざん再現ルールは，Ruby のハッシュ形式で記載する．以下に各項目の詳細と設定方法を示す．

url-pattern：ダミーサイトサーバ上の攻撃対象の URL を指定する．正規表現を用いることが可能である．実際の攻撃対象 URL がメインコンテンツの場合は，改ざん対象サイトのドメイン名または，メインコンテンツのフル URL を指定する．実際の攻撃対象 URL がメインコンテンツからリンクされたコンテンツの場合は，メインコンテンツ内に記載されている対象コンテンツへのリンク URL を指定する．

replace-src：攻撃設定情報に指定された改ざん対象文字列を指定する．正規表現を用いることが可能である．基本的に攻撃設定情報の内容を複写して使用することが可能であるが，実際の攻撃設定情報で正規表現等が用いられる場合，Ruby で解釈可能な形式への書換え，または，正規表現に一致する全文字列を記載する必要がある．

replace-dst：攻撃設定情報に指定された挿入コード片を記載する．改ざん再現システムでは，改ざんを文字列の置換のみで行うため，金融系マルウェアの用いる改ざん方法が置換以外の場合は，改ざん対象文字列 + 挿入コード片の様に文字列置換で再現可能な内容に変更する必要がある．injection-file-path とは一緒に設定できない．

injection-file-path：攻撃設定情報の内容に合わせて予め改ざんしたコンテンツのファイルを用いて応答するためのファイルパスを指定する．本設定は，挿入コード片に Ruby で置換処理を行った場合に正しく取り扱えないバイナリ文字が入っていた場合に使用する．replace-dst とは一緒に設定できない．

content-type：injection-file-path で指定されたファイルのコンテンツタイプを指定する．injection-file-path を使用する場合は必須．

4.2　MITB 攻撃用 JavaScript 収集

MITB 攻撃用 JavaScript は，4.1 で収集した攻撃設定情報の挿入コード片が実行される事で，マニピュレーションサーバから取得される．しかし，挿入コード片を単体で実行しても MITB 攻撃用 JavaScript が取得されない場合が存在する．これは，挿入コード片が挿入された際，もしくは，通信が発生した際にマルウェアによって挿入コード片の一部や通信先を動的に変換する場合が存在するためである．そこで，MITB 攻撃用 JavaScript を取得可能な URL

表 3　解析用 PC 環境

ホスト OS	macOS 10.13.6
仮想環境	VMwareFusion 8.5.10
ゲスト OS	Windows 7 Professional 32bit
Web ブラウザ	Internet Explorer 11，Google Chrome 67，Firefox 36
通信監視ツール	Fiddler2，WireShark

の収集および挿入コード片や通信先の動的な変更を確認するために改ざん対象文字列のみが存在するコンテンツを改ざん再現システムに設定して金融系マルウェアの動的解析を行う．この際，改ざん再現ルールは設定せずに解析用 PC に金融系マルウェアを感染させた状態でダミーサイトサーバへ接続する．また，改ざん再現システムと解析用 PC は安全にマルウェアを実行するため閉じたネットワーク構成としダミーマニピュレーションサーバから実際のマニピュレーションサーバへの通信転送も行わない．なお，動的解析の際に，改ざん対象文字列のみが存在するコンテンツを用いることで，コンテンツを容易に作成することが可能である．さらに，正規コンテンツに含まれる従来の通信が発生しないため MITB 攻撃用 JavaScript 取得通信のみを観測することが可能となる．

動的解析の結果，MITB 攻撃によるコンテンツ改ざんが発生し，MITB 攻撃用 JavaScript 取得通信が発生する．この発生した，通信を記録する．その後，記録した通信から MITB 攻撃用 JavaScript 取得 URL を収集し，wget 等のコマンドで MITB 攻撃用 JavaScript を取得する．

また，解析用 PC の Web ブラウザのデバッグ機能を用いて通信ログと改ざんされたコンテンツを収集する．この通信ログをダミーマニピュレーションサーバに対して発生した通信と比較することで，マルウェアによる通信先変更が行われているかを確認する．また，改ざん後のコンテンツに含まれる挿入コード片と攻撃設定情報に含まれる挿入コード片を比較することで，マルウェアによる挿入コード片の動的な変更が行われているかを確認する．通信先変更が行われた場合，マルウェア本体を用いない環境では，解析用 PC からは変更前の通信が発生するため，変更前の通信情報を用いてダミーマニピュレーションサーバと通信をするようにルーティングの設定をする．また，挿入コード片の変更が行われた場合，各ダミーサーバの文字列置換・挿入機能に置換対象と置換後の文字列を設定する．これによって，改ざん再現ルールに含まれないマルウェアによる動的な文字列の置換を再現可能とする．

4.3　MITB 攻撃用 JavaScript の動的解析

MITB 攻撃用 JavaScript の動的解析は，改ざん再現システムを使用して実施する．ダミーサイトを構築するために攻撃対象サイトのコンテンツ収集を行う．コンテンツ収集には，Google Chrome を用いる．Google Chrome のデベロッパーツールの Network パネルにおける通信モニタリングを有効にした状態で攻撃対象サイトに接続する．攻撃対象サイトの読み込みが完了した時点で，HTTP ARchive（以下，HAR）ファイルを保存する．取得した HAR ファイルを，Ruby で作成したパーサーを用いて展開する．コンテンツは Web サイトのフォルダ構成を再現した状態で展開される．このコンテンツと改ざん再現ルールをダミーサイトサーバに設定して，ダミーサイトを構築する．

解析対象の MITB 攻撃用 JavaScript は，ダミーマニピュレーションサーバの指定フォルダに配置し，MITB 攻撃用 JavaScript 取得 URL への通信が発生した際に MITB 攻撃用 JavaScript を応答する様にルーティングを設定する．また，ダミーマニピュレーションサーバの文字列置換・挿入機能を用いて MITB 攻撃用 JavaScript に対し，“sourceURL” ディレクティブを追加する．これによって，通常 Web ブラウザのデバッガ機能を用いた解析が困難な eval のソースを Google Chrome や Firefox 等のデバッガ機能で解析を行うことを可能とする．

MITB 攻撃用 JavaScript の動的解析は，解析用 PC の Web ブラウザからダミーサイトサーバに接続し，実際の操作を行うことで実施する．MITB 攻撃の攻撃対象は，多くがインターネットバンキング等のログイン画面であるため，ダミーの認証情報を入力し，ログインボタン等を押下する操作を行う．その間の UI の状態の確認および通信ログを収集する．さらに，Web ブラウザのデバッガ機能を利用して MITB 攻撃用 JavaScript の挙動を解析する．また，MITB 攻撃用 JavaScript が難読化されている場合，同様に Web ブラウザのデバッガ機能を利用することで難読化の解除された MITB 攻撃用 JavaScript を入手する．

5.　評価実験

5.1　実験方法

提案手法の有効性を評価するため金融系マルウェア 2 検体から収集した攻撃設定情報を用いて実験を行う．実験対象のマルウェアは，VirusTotal[12] から取得した Ursnif を用いる．この 2 検体は，それぞれ異なる攻撃設定情報を保有する．実験対象の概要を**表 4** に示す．実験対象の 2 検体を用いて，以下の手順で実験を行った．

(1) 攻撃設定情報の分析による攻撃対象の特定および改ざん対象の有無を調査し，改ざん再現ルールを作成する．
(2) 改ざん対象文字列のみのコンテンツを用いた動的解析を行い，MITB 攻撃用 JavaScript 取得 URL を収集する．
(3) 収集した MITB 攻撃用 JavaScript 取得 URL を用いて，MITB 攻撃用 JavaScript を取得する．
(4) 改ざん再現システムを用いて，MITB 攻撃用 JavaScript の動的解析を行う．

表 4　実験対象の概要

検体名	攻撃対象サイト数	有効攻撃対象サイト数
検体 1	5 サイト	3 サイト
検体 2	50 サイト	43 サイト

5.2　攻撃設定情報の分析

攻撃設定情報の分析結果について述べる．全攻撃対象 URL のコンテンツ取得を行い改ざん対象の有無の確認を行った．その結果判明した有効な攻撃対象サイト数を，表 4 に示す．検体 1 では，2 サイトが法人向けのインターネットバンキングであり，銀行の発行した証明書を持った使用者のみが接続可能であったため本実験では，対象外とし有効攻撃対象サイトにはカウントしていない．また，検体 2 では，攻撃設定情報の攻撃対象 URL が実在しないものや攻撃対象 URL のコンテンツ内に改ざん対象文字列が存在しないものが含まれていた．

5.3　MITB 攻撃用 JavaScript の収集

MITB 攻撃用 JavaScript の収集結果について述べる．5.2 の結果明らかになった有効攻撃対象サイトに含まれる改ざん対象文字列のみのコンテンツを改ざん再現システムに設定し，検体 1 および検体 2 を感染させた解析用 PC を用いて，ダミーサイトサーバに接続し，MITB 攻撃用 JavaScript 取得 URL の収集を行った．また，収集した全ての MITB 攻撃用 JavaScript 取得 URL を用いて wget コマンドで MITB 攻撃用 JavaScript の取得を行った．結果を，**表 5** に示す．

動的解析の結果，検体 1 で収集した MITB 攻撃用 JavaScript 取得 URL は，3 個であった．3 個の URL には，2 種類のドメインが存在しており，いずれも接続可能であった．また，3 個の URL 全てから異なる MITB 攻撃用 JavaScript を取得することが可能であった．これは，有効な攻撃対象の 3 サイト全てに対して，MITB 攻撃による情報盗取や不正送金が行われていると考えられる．

同様の動的解析を検体 2 で行った結果，収集した MITB 攻撃用 JavaScript 取得 URL は，26 個であった．これは，複数の攻撃対象に対して，同一の挿入コード片が用いられているためである．同一の挿入コード片が用いられる攻撃対象は 3 グループ，20 サイト存在した．各サイトの内容を確認したところそれぞれ，3 種類の共同インターネットバンキングシステム（以下，共同 IB システム）を使用していることを確認した．この結果から，共同 IB システムに対しては，共通の挿入コード片および MITB 攻撃用 JavaScript が使用されていることが判明した．これらの共同 IB システムを用いるサイトは，グループ毎に 1 サイトとカウントし，攻撃設定情報内で各共同 IB システム毎の先頭に登録されているものを実験対象とする．また，26 個の URL には 3 種類のドメインが存在したが，そのうち 2 つのドメインを用いる URL からは，MITB 攻撃用 JavaScript を取得するこ

表 5　MITB 攻撃用 JavaScript 収集実験の結果

検体名	MITB 攻撃用 JS 取得 URL 数	取得した MITB 攻撃用 JS 数
検体 1	3	3
検体 2	26	20

表 6　取得した MITB 攻撃用 JavaScript の攻撃対象サイト種別

サイト種別	検体 1	検体 2
銀行	3	7
EC サイト	1	1
クレジットカード会社	0	9
仮想通過取引所	0	2
フリーメールサービス	0	1

とができなかった．1 つは DNS 解決の行えない無効なドメインであった．また，もう 1 つのドメインは，URL に対してアクセス可能であったが，全て 404 エラーが返却された．この無効な URL を除いた残りの MITB 攻撃用 JavaScript 取得 URL は 21 個存在した．そのうち，1 つの URL では，応答があるものの空コンテンツが返却され MITB 攻撃用 JavaScript は取得されなかった．残りの 20 個の URL からは，それぞれ異なる MITB 攻撃用 JavaScript を取得した．これらの 20 個の攻撃設定情報に対応するサイトに対しては，MITB 攻撃による情報盗取や不正送金が行われていると考えられる．MITB 攻撃用 JavaScript が取得可能であったサイトの種別を**表 6** に示す．

また，マルウェアによる挿入コード片または MITB 攻撃用 JavaScript 取得通信時の通信先の動的な変更に関しては，検体 1 の 2 個の攻撃設定情報を除く全てで，いずれかの動的な変更が行われていることを確認した．

5.4　MITB 攻撃用 JavaScript の動的解析

MITB 攻撃用 JavaScript の動的結果について述べる．表 6 の攻撃対象サイトから本稿における実験対象を決定する．銀行（インターネットバンキング），クレジットカード会社，仮想通貨取引所に関しては，いずれもログイン画面が攻撃対象とされていた．この内，銀行は，全てのサイトを実験対象とした．クレジットカード会社，仮想通貨取引所は，各 1 サイトを実験対象とした．なお，攻撃設定情報内で各サイト種別毎の先頭に登録されていたものを選択している．EC サイトおよびフリーメールサービスはログイン画面以外を攻撃対象としていたため実験の対象外とした．

実験対象とした各サイトのダミーサイトおよび改ざん再現ルールを改ざん再現システムに設定し，動的解析を実施した．実験は以下の観点で評価を行った．

- コンテンツ改ざんを再現し，MITB 攻撃用 JavaScript の取得が行われるか．
- MITB 攻撃用 JavaScript からダミーマニピュレーションサーバへの通信が発生するか．
- MITB 攻撃用 JavaScript による情報盗取または偽画面の表示が発生するか．

- Web ブラウザのデバッガ機能を利用して，MITB 攻撃用 JavaScript の実行状況を解析することが可能か．また，難読化解除後のコードを取得することが可能か．

評価実験の結果を**表 7** および**表 8** に示す．これらの結果からすべてのダミーサイトに対してコンテンツ改ざんが再現され MITB 攻撃用 JavaScript の取得が行われることを確認した．

表 7 の結果から銀行 A，銀行 B 共に MITB 攻撃用 JavaScript からダミーマニピュレーションサーバへの通信が発生することを確認した．しかし，銀行 A のダミーサイトでは，ログイン操作を行ったものの認証情報がダミーマニピュレーションサーバにアップロードされる通信は確認されなかった．銀行 B のダミーサイトでは，ログイン操作を行った際に認証情報がダミーマニピュレーションサーバにアップロードされることを確認した．しかし，実験中に実際のマニピュレーションサーバが停止し，応答が得られず，その後の動作を確認することはできなかった．銀行 A，銀行 B いずれのダミーサイトでも偽画面等の表示は確認されなかった．Web ブラウザのデバッガ機能を用いた解析では，主に Google Chrome を用いて，MITB 攻撃用 JavaScript のマニピュレーションサーバとの通信などの実装箇所を特定して動作を解析することができた．また，難読化の施された銀行 A に対する MITB 攻撃用 JavaScript は，Web ブラウザのデバッガ機能を使用して難読化が解除された状態のコードを取得することが可能であった．

表 8 の結果から銀行 C および銀行 G を除くダミーサイトの MITB 攻撃用 JavaScript からダミーマニピュレーションサーバへの通信が発生することを確認した．同様に銀行 C および銀行 G を除くダミーサイトでログイン操作を行った際に認証情報がダミーマニピュレーションサーバにアップロードされることを確認した．また，銀行 D，銀行 G およびカード会社 A では，認証情報のアップロード後に偽画面の表示を確認した．銀行 D，銀行 H では暗証番号等の入力を，カード会社 A では，クレジットカード情報の入力を促す偽画面が表示された．**図 5** に銀行 D の偽画面を示す．Web ブラウザのデバッガ機能を用いた解析では，全ての MITB 攻撃用 JavaScript に対し，主に Google Chrome を用いて，MITB 攻撃用 JavaScript のマニピュレーションサーバとの通信などの実装箇所を特定し動作を解析することができた．また，検体 2 の MITB 攻撃用 JavaScript は全て難読化が施されていたが Web ブラウザのデバッガ機能を使用して，難読化が解除された状態のコードを取得することが可能であった．

以上の結果から改ざん再現システムを用いて MITB 攻撃によるコンテンツ改ざんを再現し，MITB 攻撃用 JavaScript の解析が行えることを確認した．また，銀行 B に関しては，検体 1 および検体 2 に共通に存在していた．この銀行 B に対し 1 つのダミーサイトを作成し，改ざん再現ルールを切り替えることで 2 つのコンテンツ改ざんを再現可能であることを確認した．

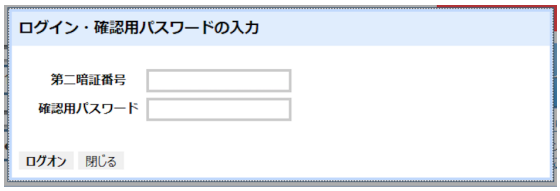

図 5 暗証番号を要求する偽画面

5.5 実験結果の考察

攻撃設定情報の分析の結果から，攻撃対象が存在しないなど有効ではない設定が存在していることが判明した．同様に，MITB 攻撃用 JavaScript 収集の結果，MITB 攻撃用 JavaScript が取得されず，有効では無い設定が存在していることも判明した．このことから，攻撃設定情報の分析後，MITB 攻撃用 JavaScript 収集を行い，MITB 攻撃用 JavaScript が取得されたサイトの改ざん再現ルールを作成することで，解析対象を限定することが可能と考える．

MITB 攻撃用 JavaScript 収集の過程において，挿入コード片または通信先がマルウェアによって動的に変更される場合が，ほぼ全ての攻撃設定情報で確認された．このことから MITB 攻撃用 JavaScript 取得 URL を収集する際にマルウェアの動的解析が必要であると考える．この際に，改ざん対象の文字列のみのコンテンツを利用することで容易に動的解析を実施することが可能であった．

MITB 攻撃用 JavaScript の動的解析の結果から，改ざん再現システムを用いて MITB 攻撃によるコンテンツ改ざんを再現可能であった．また，一部を除いて MITB 攻撃用 JavaScript とマニピュレーションサーバの通信，情報盗取および偽画面の表示を再現可能であった．なお，通信や情報盗取の動作を確認することができなかった一部の MITB 攻撃用 JavaScript に関しては，動作しない状況が正しいかを調査する必要があると考える．

検体 1 および検体 2 に共通する攻撃対象の銀行 B の実験結果から，改ざん再現ルールを用いることで，1 つのダミーサイトを利用して複数のコンテンツ改ざんを再現することが可能であることを実証することができた．

現在は，Web ブラウザのデバッグ機能を用いて手動で MITB 攻撃用 JavaScript の解析を行っている．この方法では改ざん前後の DOM 情報の差異を比較することが難しいという問題がある．DOM 情報の比較等の改ざん状況を自動で取得する仕組みを検討する必要がある．また，MITB 攻撃用 JavaScript の通信先として実際のマニピュレーションサーバに通信を転送しているが，検体 1 の銀行 B の様にマニピュレーションサーバが停止してしまうと，その後の動作を解析することができない．今後は，コンテンツ改ざんだけでなくマニピュレーションサーバを再現する必要が

表 7　検体 1 の攻撃対象コンテンツ改ざん再現実験の結果

ダミーサイト	MITB 攻撃用 JS の特徴		改ざんの再現	MITB 攻撃用 JS からの通信	情報盗取	偽画面	デバッガによる解析	難読化解除
	難読化	eval 実装						
銀行 A	有	有	○	○	×	×	可	可
銀行 B	無	無	○	△	○	×	可	対象外

○：確認された，×：確認されなかった，△：確認されたが不十分

表 8　検体 2 の攻撃対象コンテンツ改ざん再現実験の結果

ダミーサイト	MITB 攻撃用 JS の特徴		改ざんの再現	MITB 攻撃用 JS からの通信	情報盗取	偽画面	デバッガによる解析	難読化解除
	難読化	eval 実装						
銀行 B	有	有	○	○	○	×	可	可
銀行 C	有	有	○	×	×	×	可	可
銀行 D	有	有	○	○	○	○	可	可
銀行 E	有	有	○	○	○	×	可	可
銀行 F	有	有	○	○	○	×	可	可
銀行 G	有	有	○	×	×	×	可	可
銀行 H	有	有	○	○	○	○	可	可
カード会社 A	有	有	○	○	○	○	可	可
仮想通貨取引所 A	有	有	○	○	○	×	可	可

○：確認された，×：確認されなかった，△：確認されたが不十分

あると考える．その方法として，実際のマニピュレーションサーバの応答を蓄積して用いる方法と，MITB 攻撃用 JavaScript のソースコードから必要な通信結果を作成し，MITB 攻撃用 JavaScript を意図した通りに動作させる方法が考えられる．MITB 攻撃用 JavaScript の全機能を解明する上では後者がより有効であると考える．

6. まとめと今後の課題

金融系マルウェアの MITB 攻撃によるコンテンツ改ざんに用いられる MITB 攻撃用 JavaScript をマルウェア本体を用いることなく解析を行う手法について提案した．また，手法を実現するための MITB 攻撃用 JavaScript の収集方法および改ざん再現システムを構築した．これらの手法およびシステムを用いて MITB 攻撃によるコンテンツ改ざんを再現し，MITB 攻撃用 JavaScript の動的解析が可能であることを確認した．本手法を用いることで，MITB 攻撃によるコンテンツ改ざんをマルウェアを用いずに解析することが可能となる．このことは，解析を効率化するだけでなく，解析のリスクを低減し，解析者の精神的負担の軽減にも貢献すると考える．

今後，実験対象を拡大し本手法の有効性をさらに検証すると共に，システムの機能拡充を行い有効性を高めて行きたい．

参考文献

[1] 岡本勝之：拡張子 “.iqy” のファイルとは？1 日でメール 29 万通が日本国内に拡散，トレンドマイクロセキュリティブログ（オンライン），入手先 ⟨https://blog.trendmicro.co.jp/archives/19387⟩（参照 2018-08-17）．

[2] 独立行政法人情報処理推進機構セキュリティセンター：コンピュータウイルス・不正アクセスの届出状況および相談状況［2018 年第 2 四半期（4 月〜6 月）］，（オンライン），入手先 ⟨https://www.ipa.go.jp/security/txt/2018/q2outline.html⟩（参照 2018-08-17）．

[3] Rahimian, A., Ziarati, R., Preda, S. and Debbabi, M.: On the Reverse Engineering of the Citadel Botnet, *Foundations and Practice of Security* (2014).

[4] Boutin, J.-I.: The evolution of webinjects, *Virus Bulletin Conference*, pp. 25–34 (2014).

[5] Continella, A., Carminati, M., Polino, M., Lanzi, A., Zanero, S. and Maggi, F.: Prometheus: Analyzing WebInject-based information stealers, *Journal of Computer Security*, Vol. 25, No. 2, pp. 117–137 (2017).

[6] 瀬川達也，神薗雅紀，星澤裕二，吉岡克成，松本　勉：Man-in-the-Browser 攻撃を行うマルウェアの安全な動的解析手法，研究報告コンピュータセキュリティ，Vol. 2013-CSEC-61, No. 8, pp. 1–8 (2013).

[7] 柴田龍平，羽田大樹，横山恵一：Js-Walker: JavaScript API hooking を用いた解析妨害 JavaScript コードのアナリスト向け解析フレームワーク，コンピュータセキュリティシンポジウム 2016 論文集，Vol. 2016, No. 2, pp. 951–957 (2016).

[8] 上川先之，山内利宏：API 操作ログ取得による難読化 JavaScript コード解析支援システム，コンピュータセキュリティシンポジウム 2017 論文集，Vol. 2017, No. 2 (2017).

[9] 鈴木雅貴，中山靖司，古原和邦：インターネット・バンキングに対する Man-in-the Browser 攻撃への対策「取引認証」の安全性評価，　Vol. 32, No. 3, pp. 51–76 (2013).

[10] 西田雅太，太刀川剛，岩本一樹，遠藤　基，奥村吉生，星澤裕二：静的解析と挙動観測による金融系マルウェアの攻撃手法の調査，コンピュータセキュリティシンポジウム 2014 論文集，Vol. 2014, No. 2, pp. 859–866 (2014).

[11] Mizerany, B.: Sinatra, (online), available from ⟨http://sinatrarb.com/⟩ (accessed 2018-08-16).

[12] GoogleInc.: VirusTotal, (online), available from ⟨https://www.virustotal.com⟩ (accessed 2018-08-18).