情報処理学会論文誌 Vol.60 No.9 1550–1564 (Sep. 2019)

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]

受付日 2018年11月26日, 採録日 2019年6月11日

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 Web ブラウザの通信内容の盗聴・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等が行われる．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．コンテンツ改ざんの実態を明らかにするためには，この不正 JavaScript の詳細な機能を明らかにする必要がある．しかし，難読化やコード量が膨大等の原因で静的解析のみですべてを明らかにすることは困難である．そこで，動的解析を実施する必要がある．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，MITB 攻撃に用いられる不正 JavaScript を安全に動的解析するための手法を提案する．また，提案手法が複数のマルウェアによって行われる MITB 攻撃に用いられる不正 JavaScript の解析に有効であったことを示す．

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 Yoshioka[3] Tsutomu Matsumoto[3]

Received: November 26, 2018, Accepted: June 11, 2019

Abstract: Man-In-The-Browser (MITB) attacks that target users of online webservices such as Internet banking continue to be a major threat. MITB malware tampers with the contents of the web site and typically has malicious JavaScript downloaded on the target's browser to steal credentials and/or tamper transactions. In order to understand how these attacks are conducted, it is necessary to analyze the malicious JavaScript. However these scripts are often obfuscated and can consist of thousands of lines making manual static analysis difficult. On the other hand, dynamic analysis of the malicious JavaScript with an environment with actual malware infection risks negative impact on the web services and possible interruption by the attackers. In this paper, we propose a new dynamic analysis method of malicious JavaScript using an analysis environment without actual malware infection to realize more stable and less visible analysis from the attackers. We evaluate our method using in-the-wild MITB malware and show that the method is effective to most of them.

Keywords: malware, MITB, JavaScript, dynamic analysis

[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, Yokohama, Kanagawa 240–8501, Japan
[2] 株式会社セキュアブレイン
SecureBrain Corporation, Chiyoda, Tokyo 102–0094, Japan
[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, Yokohama, Kanagawa 240–8501, Japan
[a)] takada-kazuki-hw@ynu.jp

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，通信内容の盗聴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，DreamBot，Ramnit 等の MITB 攻撃を行うマルウェアの流行が確認されている [1], [2], [3].

本稿では，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 ブラウザと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 の詳細な機能を明らかにする必要がある．

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，これらが原因で静的解析のみですべてを明らかに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．MITB 攻撃の再現には，攻撃対象のオンラインサービスとの通信が必要となる．しかし，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，当該オンラインシステムへ悪影響を及ぼすリスクがある．また，金融系マルウェアには，Ursnif や DreamBot のように感染端末の操作情報等の盗取や VNC 機能による感染端末の遠隔操作の機能を有するものが存在している [4], [5]．このような攻撃機能によって，解析状況の漏洩や感染端末を別の攻撃の踏み台にされる危険性がある．さらに，金融系マルウェアによる解析妨害により，マルウェア本体や解析ツールの強制終了等が発生することで MITB 攻撃用 JavaScript の解析が行えない可能性がある．このように，マルウェア感染環境を用いた MITB 攻撃用 JavaScript の動的解析には，様々なリスクが存在している．

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うえで，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がかかる．さらに，マルウェアの取扱いに不慣れな 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，リスクをともなうとともに解析者の精神的な負担も大きい．

そこで本稿では，あらかじめ MITB 攻撃用 JavaScript を収集し，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．提案手法を用いて，2018 年 7〜10 月の期間に日本国内の金融機関等を対象に攻撃を行っている 3 種類の金融系マルウェア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 を金融系マルウェア本体を用いずに解析する手法を示したこと．
- 攻撃対象サイトの改ざん対象文字列を用いたダミーコンテンツによる金融系マルウェアの動的解析によって MITB 攻撃用 JavaScript を効率的に収集する手法を示したこと．
- 提案手法が最新の MITB 攻撃を行うマルウェア 3 種に対して有効であることを示したこと．

本稿の構成は，以下のとおりである．まず，2 章で，関連研究について記述する．3 章で，MITB 攻撃とその再現方法の検討結果について記述する．4 章で，提案手法について記述する．5 章で，評価実験および検証実験について記述する．6 章で，各実験結果の考察を記述する．最後に，7 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 [6] がある．また，日本国内における同様の研究として中津留の研究 [7] がある．研究 [6] および研究 [7] 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．これらは，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 [8] 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 [9] 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 [10] がある．研究 [10] 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 [10] 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 [10] 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，あらかじめ攻撃対象を特定したうえ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 [11] や上川らの研究 [12] 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．

攻撃が発生する環境を再現し動的解析を行う手法に関する研究には，津田らの研究 [13] がある．研究 [13] では，標的型攻撃の実態を把握するためにマルウェアの活動を安全に再現する環境およびダミーのコマンド・アンド・コントロールサーバ（以下，C&C サーバ）を構築して観測を行う手法を提案している．研究 [13] は，標的型攻撃に用いられるマルウェアの解析環境であり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 [14] 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．図 1 に示すとおり，MITB 攻撃は，初めに金融系マルウェアが攻撃設定情報を C&C サーバから受け取り，Web ブラウザの通信内容を常時監視する．その後，ユーザが攻撃対象のインターネットバンキング等にアクセスをすると攻撃設定情報の内容に従って通信内容の改ざんが発生する．この改ざんによって，マニピュレーションサーバからの MITB 攻撃用 JavaScript のダウンロードや盗取した認証

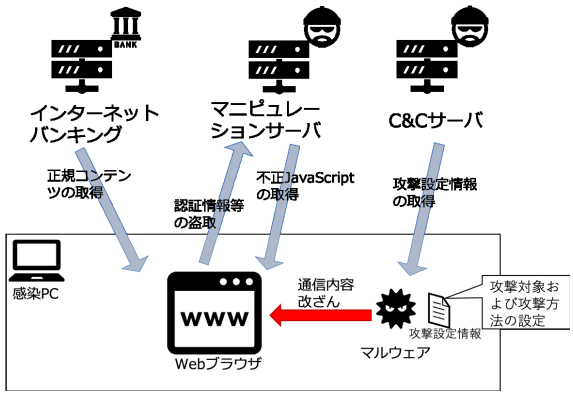


図 1 MITB 攻撃の概要
Fig. 1 Overview of MITB attack.

```
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 DreamBot の攻撃設定情報復号結果
Fig. 2 Decryption result of attack configuration of the DreamBot.

表 1 DreamBot 攻撃設定情報の解析結果
Table 1 Analysis result of attack configuration of the DreamBot.

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

情報のアップロード等が発生する．このように，MITB 攻撃は，金融系マルウェアを制御する 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いった外部サーバと連携して実行される．また，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．本稿で実験に用いた 3 種類の金融系マルウェアのうち，DreamBot 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，DreamBot の保持する暗号化された攻撃設定情報を復号し解析を行った結果である．攻撃設定情報の内容の解析結果を**表 1** に示す．このように，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．

MITB 攻撃に用いられる不正 JavaScript は，2 種類存在する．1 つは，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 である．もう 1 つは，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．本稿では，前者の不正 JavaScript を挿入コード片，後者の不正 JavaScript を MITB 攻撃用 JavaScript と呼称する．なお，稀に挿入コード片に情報盗取等の機能を持ちマニピュレーションサーバから不正 JavaScript 取得を行わない場合が存在する．この場合は，挿入コード片内に存在する MITB 攻撃用 JavaScript を提案手法の解析対象とする．

我々は，文献 [15] および論文 [16] 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 の読み込み：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．

3.1 MITB 攻撃用 JavaScript

MITB 攻撃用 JavaScript の機能について述べる．研究 [8] および論文 [16] の調査結果から MITB 攻撃用 JavaScript は，一般的に以下の攻撃機能を持つことが考えられる．

(1) 情報盗取機能
　攻撃対象サイト内の盗取対象情報の input タグや button タグ等の改ざんによる認証情報等の盗取機能．

(2) 偽画面表示機能
　正規のログイン画面では要求されない情報の入力を要求する偽の入力画面等の表示機能．

(3) 自動送金機能
　MITB 攻撃用 JavaScript がインターネットバンキングと通信をして感染 PC 上から不正送金を行う機能．

本研究では，MITB 攻撃用 JavaScript の持つ，これらの攻撃機能の挙動を解析することを目的とする．これらの攻撃機能により，どのような操作を行った際に，どのような情報が盗取されるのか，どのような画面が表示されるのか，どのような通信が発生するのかの挙動を解析することが主たる目的である．

本稿では，解析対象の MITB 攻撃用 JavaScript が，これらの攻撃機能を保有すると仮定し，動的解析を行う．なお，(3) 自動送金機能に関しては，インターネットバンキングログイン後の画面で行われるものである．インターネットバンキングログイン後の画面の構築には，銀行口座の開設が必要であり，本稿では，ログイン画面に攻撃対象を限定して行っているため，解析の対象外とする．

3.2 MITB 攻撃再現方法の検討

MITB 攻撃用 JavaScript を解析するためには，MITB 攻撃によるコンテンツ改ざんを再現する必要がある．MITB 攻撃の際に発生する金融系マルウェアによる Web ブラウザへのインジェクションおよびコンテンツの改ざんは，Web ブラウザの通信レイヤ（通信を行う DLL 内の API 等）に対して行われる．これによって，改ざんされたコンテンツをブラウザエンジンが解釈し動作することで MITB 攻撃が成立する．よって，ブラウザエンジンがコンテンツを読み込む前にコンテンツを改ざんすることで，金融系マルウェアによるコンテンツ改ざんを再現することが可能である．また，複数の Web ブラウザに対してインジェクションおよびコンテンツの改ざんを行う金融系マルウェアにおいて，いずれの Web ブラウザに対しても同一の攻撃設定情報が用いられ，改ざん後のコンテンツが同一になることを確認している．

金融系マルウェア本体を用いずにブラウザエンジンがコンテンツを読み込む前にコンテンツ改ざんを再現する方法として，以下が考えられる．

- Web ブラウザ内で通信内容を改ざんする
  - コードインジェクション等を実施するツール（以下，疑似マルウェアツール）による改ざん
  - ブラウザ拡張による改ざん
- Web サーバであらかじめ改ざんしたコンテンツを配信する

疑似マルウェアツール，ブラウザ拡張は，金融系マルウェアと同様に Web ブラウザ内でコンテンツを改ざんするため再現度が高いと考えられる．しかし，これらは，Internet Explorer・Google Chrome・Firefox 等のインジェクション対象の Web ブラウザごとに個別に実装をする必要が生じる．また，Web ブラウザや OS のバージョンアップ等にともないメンテナンスを必要とする可能性がある．さらに，疑似マルウェアツールは，API Hook 等の高い技術力を必要とする．このように，Web ブラウザ内での再現手法は，開発やメンテナンスにかかるコストが大きく運用が困難と考えられる．

ブラウザエンジンは，通信レイヤでどのようなデータを扱っているかを関知せず通信の結果のみを解釈し動作するため，Web ブラウザの外部から受け取るデータが改ざんされていた場合も通信レイヤでコンテンツが改ざんされた場

合と同様の挙動を示すと考えられる．よって，Web サーバからあらかじめ改ざんしたコンテンツを配信することで，コンテンツ改ざんを再現することが可能である．Web サーバでコンテンツ改ざん再現をすることで，Web ブラウザごとの開発やメンテナンスをする作業が基本的に不要となる．また，特定の Web ブラウザに限定されることなく，MITB 攻撃によるコンテンツ改ざんを再現し解析することが可能となるため汎用性が高い．

以上の検討結果に基づいて，次章以降で Web サーバでコンテンツ改ざんを再現する手法を用いた提案手法について述べる．また，マルウェア感染環境を用いた解析結果と提案手法を用いた解析結果を比較する実験を行うことで，提案手法を用いて MITB 攻撃を再現可能であることを検証する．

4. 提案手法

提案手法は，3 章の MITB 攻撃のステップのうち Step 3〜5 を 3.2 節の検討結果を用いて再現することで，MITB 攻撃によるコンテンツ改ざんを発生させ，その際の MITB 攻撃用 JavaScript の挙動を解析することを可能とするものである．提案手法の全体概要を**図 3** に示す．図 3 は，提案手法を構成する手順と各手順へのインプットとアウトプットの流れを示したものである．なお，本稿の提案手法は，図 3 のうち赤点線の枠内である．図 3 内の金融系マルウェア挙動観測の手法に関しては，論文 [16] を参照されたい．

図 3 のとおり，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

提案手法では図 3 に示したとおり，攻撃設定情報の分析と MITB 攻撃用 JavaScript 収集により，攻撃対象サイトの特定および MITB 攻撃用 JavaScript の収集を行う．その後，分析結果および MITB 攻撃用 JavaScript を用いて，本手法のために構築した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）を用いて，MITB 攻撃用 JavaScript の解析を行う．4.1 節で改ざん再現システムについて述べる．また，4.2〜4.4 節で提案手法の各段階の詳細について述べる．

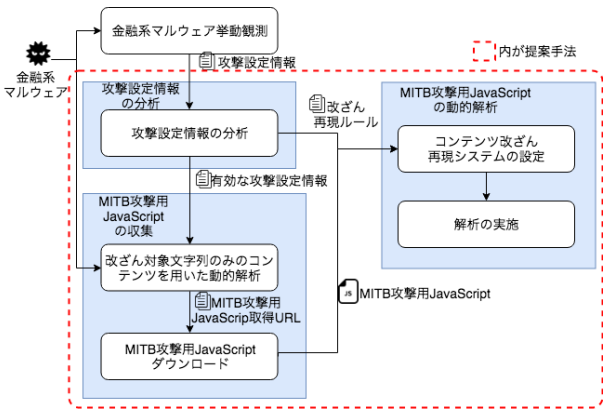


図 3 提案手法の概要
Fig. 3 Overview of the proposed method.

4.1 コンテンツ改ざん再現システム

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．図 4 の改ざん再現システムは，ダミーサイトサーバとダミーマニピュレーションサーバの 2 つのダミーサーバを中心に構成される．解析者は，解析用 PC でダミーサイトサーバに接続して解析を行う．ダミーサイトサーバは，改ざん再現ルールに従ってあらかじめ改ざんしたコンテンツを返却する．その後，解析用 PC 内の Web ブラウザ上で改ざんコンテンツが動作することで，ダミーマニピュレーションサーバとの通信が発生する．この通信に対し，ダミーマニピュレーションサーバは，MITB 攻撃用 JavaScript の返却や実マニピュレーションサーバへの通信の転送を行う．また，システム内で発生する Web アクセスはすべてダミー DNS サーバによって 2 つのダミーサーバのいずれかとなる．これによって，解析用 PC の Web ブラウザ上で MITB 攻撃が再現される．なお，図 4 内のマニピュレーションサーバに関しては，攻撃者の利用するサーバであり，本システム用に構築したものではない点に注意されたい．改ざん再現システムの構成要素について以下で解説する．

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

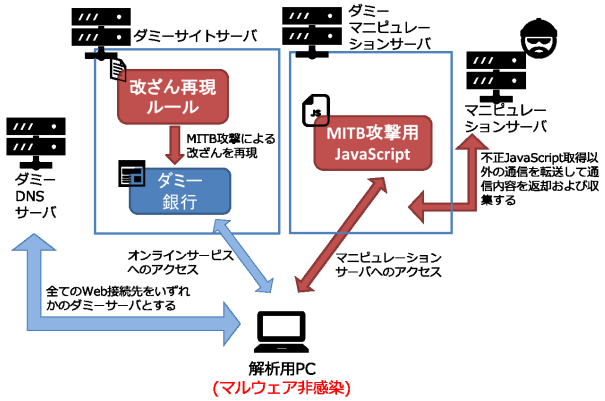


図 4 コンテンツ改ざん再現システム
Fig. 4 Content tampering reproduction system.

表 2 コンテンツ改ざん再現システムの構成
Table 2 Composition of the content tampering reproduction system.

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

表 3 解析用 PC 環境
Table 3 Environment of analysis PC.

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

ざん再現ルール（詳細は，4.1.1 項を参照）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 [17] の after フィルタを利用して，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：攻撃者のマニピュレーションサーバを模擬し，MITB 攻撃用 JavaScript を取得する通信に対し，サーバ内に設定した MITB 攻撃用 JavaScript を送り返す．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．この機能を利用して，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 Web ブラウザで動的評価される JavsScript を Web ブラウザのデバッガ機能で解析を行うことを可能とする．

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．

4.1.1 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，挿入コード片をもとに設定する．改ざん再現ルールは，Ruby のハッシュ記法で記載する．以下に各項目の設定内容を示す．

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．

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．

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 + 挿入コード片のよう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 とは排他である．

injection-file-path： 攻撃設定情報の内容に合わせてあらかじ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等が入っていた場合に使用する．replace-dst とは排他である．

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である．

4.1.2 ダミーサイトの構築手法

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（以下，Chrome）を用いる．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．

4.2 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，論文 [16] の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うえで，対象検体を挙動観測する調査手法を用いて金融系マルウェアの攻撃設定情報を観測・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，有効な改ざん対象を MITB 攻撃用 JavaScript 収集の対象とする．また，攻撃対象の改ざん再現ルールを作成する．

4.3 MITB 攻撃用 JavaScript 収集

MITB 攻撃用 JavaScript は，攻撃設定情報内の挿入コード片が実行されることで，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 の

収集と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金融系マルウェアを感染させた解析用 PC で，Internet Explorer 11（以下，IE11）を用いて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である．

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，通信ログを記録する．その後，通信ログから 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．

4.4 MITB 攻撃用 JavaScript の動的解析

改ざん再現システムを使用して，MITB 攻撃用 JavaScript の動的解析を実施する．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，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，以下の手順で解析を実施する．

(1) 各ダミーサイトのログイン画面に解析用 PC の Web ブラウザで接続
(2) ダミーの認証情報を入力し，ログインボタンを押下

上記操作時の通信ログを Fiddler で収集，UI の状態を目視で確認する．また，Web ブラウザのデバッガを立ち上げた状態で同様の操作を行い，難読化が解除された MITB 攻撃用 JavaScript の取得および MITB 攻撃用 JavaScript のステップ実行等によるコード解析を行う．

5. 実験

5.1 評価実験

提案手法の有効性を評価するため 3 種類の金融系マルウェアから収集した攻撃設定情報を用いて評価実験を行う．実験対象のマルウェアは，VirusTotal [18] から取得し，Ursnif，DreamBot および Ramnit を用いる．これらは，事前に論文 [16] の調査手法で観測を行い攻撃設定情報を収集している．この 3 検体は，それぞれ異なる攻撃設定情報を保有する．実験対象の概要は，**表 4** を参照．実験対象の 3 検体を用いて，提案手法による MITB 攻撃用 JavaScript の動的解析を行った．

5.1.1 MITB 攻撃用 JavaScript 動的解析の手順および評価基準

MITB 攻撃用 JavaScript の動的解析の実施手順および評価方法について述べる．

攻撃対象サイトのうちログイン画面を対象に動的解析の評価を実施する．動的解析は，4.4 節に述べた手順に従って行う．その際，各ダミーサイトへのアクセスは，解析用 PC の 3 種類の Web ブラウザすべてで実施する．なお，Web ブラウザのデバッガ機能を用いた解析は，Chrome のデバッガ機能でのみ実施する．

動的解析結果の評価基準は，以下のとおりである．

コンテンツ改ざん初期動作の再現：改ざん再現システムであらかじめ埋め込まれた挿入コード片の実行による MITB 攻撃用 JavaScript の読み込みと実行が確認されるか．

表 4 攻撃設定情報の分析および MITB 攻撃用 JavaScript 収集結果
Table 4 Result of attack configuration analysis and malicious JavaScript collection.

検体名	マルウェア名	攻撃設定情報の分析		MITB 攻撃用 JavaScript 収集結果	
		攻撃対象サイト数	有効攻撃対象サイト数	攻撃 JS 取得 URL 数	取得した攻撃 JS 数
検体 1	Ursnif	5	3	3	3
検体 2	DreamBot	50	43	26	20
検体 3	Ramnit	16	16	14	16（2）

（）内は，挿入コード片内に攻撃 JS が含まれるもの

MITB 攻撃用 JavaScript からの通信確認：MITB 攻撃用 JavaScript からマニピュレーションサーバへの通信が発生するか．

情報盗取機能による攻撃動作の確認：MITB 攻撃用 JavaScript の動作により，正規インプットフィールドへ入力した ID，パスワード等の情報がマニピュレーションサーバへアップロードされる動作が確認されるか．

偽画面表示機能による攻撃動作の確認：MITB 攻撃用 JavaScript の動作により，追加情報盗取用の偽画面が 1 つでも確認されるか（銀行であれば，第二暗証番号等の決済認証情報・その他カード会社等であればクレジットカード情報，仮想通貨取引所であれば二段階認証のパスコード等）．

Web ブラウザのデバッガ機能を用いたコード解析：MITB 攻撃用 JavaScript のエントリーポイントを特定し，Web ブラウザへの読み込み完了時点からステップ実行による挙動の解析が可能か．また，MITB 攻撃用 JavaScript が難読化されていた場合に，難読化を解除したコードを特定して取得およびステップ実行による挙動の解析が可能か．

5.2 攻撃設定情報の分析結果

各検体の攻撃設定情報の分析結果を，表 4 に示す．表 4 から検体 1 および検体 2 では，あらかじめ設定された攻撃対象サイトに対して有効攻撃対象サイトが減少している．これは，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．本来これらは，対象とすべきであるが，コンテンツを入手することが不可能であったため本実験では，対象外とした．

検体 2 では，攻撃設定情報の攻撃対象 URL が存在しないものが 5 サイト，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 2 サイト含まれていたため解析対象から除外した．

検体 3 では，すべての攻撃対象サイトが有効な攻撃対象であった．

5.3 MITB 攻撃用 JavaScript の収集結果

5.2 節の結果明らかになった有効攻撃対象サイトを対象に，MITB 攻撃用 JavaScript 取得 URL および MITB 攻撃用 JavaScript の収集を行った．結果を，表 4 に示す．また，MITB 攻撃用 JavaScript が取得可能であったサイトの種別を**表 5** に示す．

5.3.1 検体 1 の MITB 攻撃用 JavaScript の収集結果

表 4 の結果から，検体 1 で収集した MITB 攻撃用 JavaScript 取得 URL は，3 個であり，3 個すべての URL から異なる MITB 攻撃用 JavaScript を取得することが可能であった．

5.3.2 検体 2 の MITB 攻撃用 JavaScript の収集結果

表 4 の結果から，検体 2 で収集した MITB 攻撃用 JavaScript 取得 URL は，26 個であり，有効攻撃対象サイト数より大幅に減少している．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，グループごとに 1 サイトとカウントし，攻撃設定情報内で各共同 IB システムごとの先頭のものを今後の実験対象とした．

また，取得された 26 個の URL のうち 20 個から，それぞれ異なる MITB 攻撃用 JavaScript を取得した．なお，MITB 攻撃用 JavaScript が取得できなかった URL は，DNS 解決ができないドメインやアクセス可能でも 404 エラー等が返却されるものであった．

5.3.3 検体 3 の MITB 攻撃用 JavaScript の収集結果

表 4 の結果から，検体 3 で収集した MITB 攻撃用 JavaScript 取得 URL は，14 個であり，14 個すべての URL から異なる MITB 攻撃用 JavaScript を取得することが可能であった．

なお，MITB 攻撃用 JavaScript 取得通信が発生しなかった 2 つの挿入コード片の内容を確認したところ，他の挿入コード片と比べてコード量が多く，情報盗取用と思われる偽画面の html コンテンツが存在していた．このことから，2 個の挿入コード片には，MITB 攻撃用 JavaScript が含まれると考えられる．よって，検体 3 では，16 個すべての攻撃対象に対して異なる MITB 攻撃用 JavaScript を取得した．

5.3.4 マルウェアによる挿入コード片または通信先の動的変更

マルウェアによる挿入コード片の動的な変更に関しては，検体 1 および検体 2 で，挿入コード片内の “@ID” という文字列をマルウェアの保有する ID と思われる文字列に置換する処理が確認された．また，検体 3 でも挿入コード片内の “<%IDBOT%>” という文字列をマルウェアの

表 5 取得した MITB 攻撃用 JavaScript の攻撃対象サイト種別
Table 5 Types of targeted sites to collected malicious JavaScript.

サイト種別	検体 1	検体 2	検体 3
銀行	2	7	0
EC サイト	1	1	2
クレジットカード会社	0	9	13（1）
仮想通貨取引所	0	2	0
フリーメールサービス	0	1	0
Web ポータル	0	0	1（1）

（）内は，挿入コード内に攻撃 JS が含まれるもの

表 6 検体 1 の攻撃対象コンテンツ改ざん再現実験の結果
Table 6 Results of content tampering reproduction experiment of Sample 1.

ダミーサイト	MITB 攻撃用 JS の特徴		改ざん初期動作の再現	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	JS の動的解釈						
銀行 A	有	有	○	○	×	×	可	可
銀行 B	無	無	○	△	○	×	可	対象外

有：該当特徴有り，無：該当特徴無し ○：確認された，×：確認されなかった，△：確認されたが不十分
可：該当の操作またはデータ入手が可能，不可：該当の操作またはデータ入手が不可能

表 7 検体 2 の攻撃対象コンテンツ改ざん再現実験の結果
Table 7 Results of content tampering reproduction experiment of Sample 2.

ダミーサイト	MITB 攻撃用 JS の特徴		改ざん初期動作の再現	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	JS の動的解釈						
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

有：該当特徴有り，無：該当特徴無し ○：確認された，×：確認されなかった，△：確認されたが不十分
可：該当の操作またはデータ入手が可能，不可：該当の操作またはデータ入手が不可能

表 8 検体 3 の攻撃対象コンテンツ改ざん再現実験の結果
Table 8 Results of content tampering reproduction experiment of Sample 3.

ダミーサイト	MITB 攻撃用 JS の特徴		改ざん初期動作の再現	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	JS の動的解釈						
カード会社 A	有	有	○	○	○	○	可	可
EC サイト A	有	有	○	○	○	○	可	可
Web ポータル A	無	無	○	○	○	○	可	対象外

有：該当特徴有り，無：該当特徴無し ○：確認された，×：確認されなかった，△：確認されたが不十分
可：該当の操作またはデータ入手が可能，不可：該当の操作またはデータ入手が不可能

保有する ID と思われる文字列に置換する処理が確認された．通信先の動的な変更に関しては，検体 1 および検体 2 で通信先の動的な変更が行われていることを確認した．これらの，挿入コード片の置換および通信先の変更を改ざん再現システムの設定に用いた．

なお，通信先の動的変更に関しては，挿入コード片からの通信先 URL に含まれる文字列をマニピュレーションサーバに変更するルールが攻撃設定情報に含まれることが判明した．

5.4 MITB 攻撃用 JavaScript の動的解析結果

表 5 からログイン画面が攻撃対象とされたサイトのうち，銀行は全サイトを，その他の企業は攻撃設定情報の先頭にある 1 サイトを選定して解析対象とする．解析対象は以下のとおりである．

検体 1：銀行 2 サイト
検体 2：銀行 7 サイト，クレジットカード会社 1 サイト，仮想通貨取引所 1 サイト
検体 3：クレジットカード会社 1 サイト，EC サイト 1 サイト，Web ポータル 1 サイト

各解析対象のダミーサイトを用いた改ざん再現システムによる動的解析の結果を**表 6**，**表 7**，**表 8** に示す．各表の，「改ざん初期動作の再現」，「MITB 攻撃用 JS からの通信」，「情報盗取」，「偽画面」は，5.1.1 項の対応する評価基準の内容が確認されたか否かで「○」，「×」判定を行った．なお，検体 1 の銀行 B においてのみ「MITB 攻撃用 JS からの通信」で MITB 攻撃用 JavaScript から通信が発生したもののマニピュレーションサーバからの応答が得られない状態が発生したため不十分と判断し「△」とした．また，各表の「デバッガによる解析」，「難読化解除」は，5.1.1 項の「ブラウザのデバッガ機能を用いたコード解析」で期待される操作を行うことが，可能であったか否かを確認した．これらの結果からすべてのダミーサイトでコンテンツ改ざんの初期動作が再現可能であった．また，いずれ

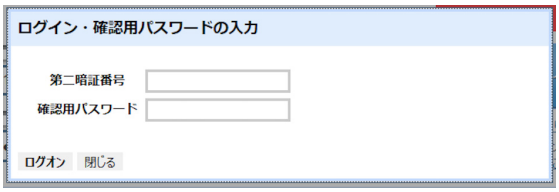


図 5 暗証番号を要求する偽画面
Fig. 5 Fake PIN code input screen.

の MITB 攻撃用 JavaScript も実行開始時にマニピュレーションサーバと通信が発生することを確認した．

Chrome のデバッガ機能を用いたコード解析もすべてのダミーサイトで，MITB 攻撃用 JavaScript の読込時点からステップ実行による挙動解析が可能であった．また，解析の結果，マニピュレーションサーバとの通信等の実装箇所を特定して動作を解析することができた．難読化された MITB 攻撃用 JavaScript の解除については，難読化が施されている MITB 攻撃用 JavaScript を使用した全ダミーサイトで難読化を解除したコードを特定し，取得およびコード解析の対象とすることができた．

5.4.1 検体 1 のコンテンツ改ざん再現結果

表 6 の結果から，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等は確認されなかった．

一方，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．なお，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．

これらの結果から，銀行 A のダミーサイトでは，情報盗取機能および偽画面表示機能のいずれの挙動も確認することができなかった．銀行 B のダミーサイトでは，情報盗取機能の挙動を確認し，この間の通信ログから通信内容の分析および Web ブラウザのデバッガ機能を用いてこの間の MITB 攻撃用 JavaScript コードをステップ実行することでコード分析をすることが可能であった．しかし，その後のマニピュレーションサーバの応答が得られなかったため，動作を継続して解析できなかったことで，偽画面表示機能については確認することができなかった．

5.4.2 検体 2 のコンテンツ改ざん再現結果

表 7 の結果から，すべての銀行のダミーサイトでログインボタン押下時に認証情報のアップロードが行われることを確認した．さらに，銀行 C，銀行 D，銀行 G，銀行 H では，認証情報のアップロード完了後に暗証番号等の入力を求める偽画面が表示された．例として，**図 5** に銀行 D の偽画面を示す．また，カード会社 A では，ログイン操作時に認証情報の盗取は行われず，ログインボタン押下時にクレジットカード情報の入力を求める偽画面が表示された．この偽画面にダミーのカード情報を入力すると，入力済みのログイン認証情報およびカード情報が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．

これらの結果から，検体 2 のすべてのダミーサイトで情報盗取機能の挙動を確認し，この間の通信ログから通信内容の分析および Web ブラウザのデバッガ機能を用いてこの間の MITB 攻撃用 JavaScript コードをステップ実行することでコード分析をすることが可能であった．偽画面表示機能に関しては，銀行 B，銀行 E，銀行 F および仮想通貨取引所 A のダミーサイトを除くすべてのダミーサイトで偽画面表示機能の挙動を確認し，この間の通信ログから通信内容の分析および Web ブラウザのデバッガ機能を用いてこの間の MITB 攻撃用 JavaScript コードをステップ実行することでコード分析をすることが可能であった．

5.4.3 検体 3 のコンテンツ改ざん再現結果

表 8 の結果から，すべてのダミーサイトで，情報盗取機能および偽画面表示機能の挙動を確認し，この間の通信ログから通信内容の分析および Web ブラウザのデバッガ機能を用いてこの間の MITB 攻撃用 JavaScript コードをステップ実行することでコード分析をすることが可能であった．いずれも，検体 2 のカード会社 A の動作と同様に，ログイン操作時に認証情報の盗取は行われず，ログインボタン押下時にクレジットカード情報の入力を求める偽画面が表示された．この偽画面にダミーのカード情報を入力すると，入力済みのログイン認証情報およびカード情報がマニピュレーションサーバにアップロードされる点も検体 2 のカード会社 A の動作と同様であった．なお，Web ポータル A では，MITB 攻撃用 JavaScript が挿入コード片に含まれるが他の MITB 攻撃用 JavaScript と同様に解析することが可能であった．

5.4.4 検体間で共通する攻撃対象サイトについて

攻撃対象サイトのうち銀行 B が検体 1 および検体 2 に，カード会社 A が検体 2 および検体 3 に攻撃対象として共通して存在している．これらの攻撃対象サイトに対し 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複数のコンテンツ改ざんを再現可能であることを確認した．

5.5 検証実験

改ざん再現システムによる改ざん再現および解析の評価結果の正当性を検証するために検証実験を行った．検証実験は，改ざん再現システムの改ざん再現ルールを無効にした状態で，各検体に感染させた解析用 PC を用いて，5.1.1 項の手順に従って MITB 攻撃用 JavaScript の動的解析を行った．解析の際，ダミーマニピュレーションサーバは有効にしている．これは，検体 1 のマニピュレーションサーバが停止したため本物のマニピュレーションサーバを

表 9 検体 1 による改ざん実験の結果
Table 9 Results of content tampering by Sample 1.

ダミーサイト	MITB 攻撃用 JS の特徴		改ざん動作	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	JS の動的解釈						
銀行 A	有	有	○	○	×	×	不可	可
銀行 B	無	無	○	△	○	×	可	対象外

有：該当特徴有り，無：該当特徴無し ○：確認された，×：確認されなかった，△：確認されたが不十分
可：該当の操作またはデータ入手が可能，不可：該当の操作またはデータ入手が不可能

表 10 検体 2 による改ざん実験の結果
Table 10 Results of content tampering by Sample 2.

ダミーサイト	MITB 攻撃用 JS の特徴		改ざん動作	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	JS の動的解釈						
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

有：該当特徴有り，無：該当特徴無し ○：確認された，×：確認されなかった，△：確認されたが不十分
可：該当の操作またはデータ入手が可能，不可：該当の操作またはデータ入手が不可能

表 11 検体 3 による改ざん実験の結果
Table 11 Results of content tampering by Sample 3.

ダミーサイト	MITB 攻撃用 JS の特徴		改ざん動作	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	JS の動的解釈						
カード会社 A	有	有	○	○	○	○	可	可
EC サイト A	有	有	○	○	○	○	可	可
Web ポータル A	無	無	○	○	○	○	可	対象外

有：該当特徴有り，無：該当特徴無し ○：確認された，×：確認されなかった，△：確認されたが不十分
可：該当の操作またはデータ入手が可能，不可：該当の操作またはデータ入手が不可能

使用して解析が行えない攻撃対象が存在するためである．また，Ursnif や DreamBot は情報盗取や遠隔操作の危険性があり，閉じたネットワークで解析を行うためである．ダミーマニピュレーションサーバの文字列挿入・置換機能は，解析補助のための “sourceURL” ディレクティブを追加する以外の設定を無効にしている．

なお，攻撃者のマニピュレーションサーバが停止した検体 1 に関しては，評価実験中に発生した通信ログを用いてダミーマニピュレーションサーバにおいて可能な限り，実際の通信をエミュレートしている．

また，事前の静的解析の結果からすべての検体で，IE11，Chrome，Firefox の 3 種類の Web ブラウザに対してインジェクションし MITB 攻撃を行うと考えられたが，検証実験の過程で検体 1 のみが Chrome にインジェクションしないことが判明した．このため，検体 1 のみ IE11，Firefox の 2 種類の Web ブラウザを用いて改ざんの状況を確認した．また，Web ブラウザのデバッガ機能を用いた MITB 攻撃用 JavaScript の解析についても，2 種類の Web ブラウザのいずれかのデバッガ機能を用いて JavaScript の解析が行えるかの確認を行った．

5.6 検証実験結果

検証実験の結果を**表 9**，**表 10**，**表 11** に示す．各表の各項目に対する評価方法は，評価実験と同一である．これらの結果を評価実験の表 6，表 7，表 8 の結果と比較すると，検体 1 の銀行 A の MITB 攻撃用 JavaScript に対して Web ブラウザのデバッガ機能によるコード解析が行えなかった点を除いて，改ざん再現システムを用いた評価結果と同様の結果となった．検体 1 の銀行 A の MITB 攻撃用 JavaScript に対してコード解析が行えなかった点については後述する．

改ざんの内容や発生した通信に関して，詳細を確認した結果について述べる．改ざんによりコンテンツに挿入された挿入コード片および読み込まれた MITB 攻撃用 JavaScript

の内容に改ざん再現システムとマルウェアによる改ざんで違いは確認されなかった．また，挿入コード片およびMITB 攻撃用 JavaScript から発生する通信に関しては，5.3.4 項で確認された通信先の動的変更による違いを除いて発生しなかった．また，情報盗取機能および偽画面表示機能の挙動解析に関しても評価実験と同様の結果となることを確認した．以上の結果から，改ざん再現システムを用いて金融系マルウェアによる MITB 攻撃によるコンテンツ改ざんを正確に再現可能であると考えられる．

検体 1 の銀行 A の MITB 攻撃用 JavaScript に対して Web ブラウザのデバッガ機能によるコード解析が行えなかった理由について述べる．検体 1 の銀行 A に対する改ざんでは，MITB 攻撃用 JavaScript に “sourceURL” ディレクティブを挿入したにもかかわらず，IE11 および Firefox では JavaScript の動的解釈の影響を受け，MITB 攻撃用 JavaScript の位置を特定することができなかった．このため，MITB 攻撃用 JavaScript の読込時にエントリーポイントを特定しステップ実行することができなかった．

6. 考察

6.1 攻撃設定情報の分析および MITB 攻撃用 JavaScript 収集の有効性

5.2 節の検体 2 のように攻撃設定情報に有効ではない設定が存在していることが判明した．同様に検体 2 では，5.3.2 項の結果でも，MITB 攻撃用 JavaScript が取得されないものが存在していることも判明した．このように，攻撃設定情報の分析，MITB 攻撃用 JavaScript 収集の結果から有効な解析対象を特定することが可能と考える．

また，5.3.2 項における共同 IB システムのように共通で用いられる MITB 攻撃用 JavaScript が存在している．このように MITB 攻撃用 JavaScript が共通して用いられる場合，複数の攻撃対象から 1 つを選定して解析することで解析対象を限定することが可能であると考える．

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，ほぼすべての攻撃設定情報で確認された．このうち，通信先の変更については，攻撃設定情報に通信先変更の設定が存在することが確認された．このことから，攻撃設定情報の分析結果によっては，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不必要な場合も存在すると考えられる．しかし，挿入コード片の変更については，攻撃設定情報に設定が存在しないため，マルウェアによる文字列の動的な変更を確認するためには，動的解析が有効であると考える．

6.2 改ざん再現システムの有効性

5.4 節の結果から，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 の解析を行うことが可能であると考える．また，ダミーマニピュレーションサーバにおいて MITB 攻撃用 JavaScript に “sourceURL” ディレクティブを追加することで，Web ブラウザで動的に解釈される JavaScript を容易に Web ブラウザのデバッガ機能を用いて解析することを可能とした．

また，5.4.4 項の結果から，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った．このことから，改ざん再現システムを用いることで，複数のマルウェアで用いられる MITB 攻撃用 JavaScript の解析を効率的に実施することが可能であると考えられる．

6.2.1 改ざん再現システムによる改ざんの正当性について

改ざん再現システムの評価実験結果，表 6，表 7，表 8 とマルウェア感染環境を用いた検証実験結果，表 9，表 10，表 11 は，同等であり，改ざん状況や通信の内容にもあらかじめ想定されたマルウェアによる動的な通信先変更以外の差分は確認されなかった．このことから，改ざん再現システムを用いた MITB 攻撃による改ざんの再現は，実マルウェアを用いて解析を行った場合と同等の結果を得ることが可能と考えられる．

また，表 9 の銀行 A では，Web ブラウザのデバッガ機能による解析が行えなかった結果に対し，表 6 の銀行 A では，解析を実施することが可能であった．これは，改ざん再現システムでは，マルウェアの動作に影響を受けないため Chrome を用いた解析が可能であった．しかし，検証実験では，検体 1 が Chrome にインジェクションしなかったため IE11 および Firefox による解析を試みたが期待した解析を行うことができなかった．このように，改ざん再現システムを用いた場合，本来はマルウェアによるインジェクション対象ではない Web ブラウザ等のツールを用いて解析を行える優位性がある．

6.3 金融系マルウェア本体を使用しないメリットおよびデメリット

改ざん再現システムを用いることで，マルウェア本体を用いない最大のメリットとして，“解析環境の秘匿” および “安全な解析の実施” があげられる．これは，検証実験の実施において可能な限り実環境に近づけるため，攻撃設定情報に含まれる攻撃対象ドメインや日本国内の IP アドレスへの接続を制限したうえで検体 2 に感染した解析用 PC をインターネットに接続して解析を実施した．解析中に，解析用 PC 上で MITB 攻撃用 JavaScript の内容をコピーした際にマルウェアにインジェクションされている Web ブラウザおよび explorer.exe が即座に強制終了するという現象が発生した．また，解析に使用した IP アドレスからの接続をマニピュレーションサーバから拒否されるという現象が発生した．これは，検体 2 の持つクリップボード情報

のアップロード等の機能により解析環境であることが判明し，解析を妨害されたものと考える．このように，解析環境であることが攻撃者に露見した場合にマルウェアの停止や攻撃者サーバへの通信遮断等により，解析を継続不可能となる状況に陥ることがある．そこで，改ざん再現システムを用いることで，マルウェアによる環境情報アップロード等が発生しないことにより，解析環境を秘匿することが可能となると考えられる．

上記は，解析を妨害された事例であったが，マルウェア感染環境を用いる場合に安全に配慮した環境であっても，マルウェアから意図しない通信が発生した結果として，別のマルウェアへの再感染やバックドアによって踏み台として利用されるリスクはつねに存在する．よって，改ざん再現システムを用いることは安全に解析を行ううえでも有効であると考えられる．

また，マルウェアを用いた検証実験では，OS のフリーズ，Web ブラウザの強制終了等が時折発生した．マルウェア感染環境は，挙動が不安定になることが多いため，改ざん再現システムを用いることは，安定した解析環境を提供するメリットがあると考えられる．

マルウェア本体を用いないデメリットとしては，“改ざん再現ルール作成のオペレーションミス” および “マルウェア本体と MITB 攻撃用 JavaScript が密結合した攻撃へ対応できない” が考えられる．

“改ざん再現ルール作成のオペレーションミス” では，解析者が改ざん再現ルールの作成を誤った場合に正しい解析が行えないという問題がある．実際に，本稿の基となった論文 [19] において評価実験の検体 2 の銀行 C および銀行 G の改ざん再現実験で，改ざん再現ルールに誤りがあり，MITB 攻撃用 JavaScript が正しく読み込まれなかったという問題が生じ調査を必要とした．この問題は，改ざん再現システムでの再現結果に不審な点が見られた場合，検証実験で用いたようにマルウェア感染環境とダミーサイトを用いた解析を行う方法で検証することが可能である．

“マルウェア本体と MITB 攻撃用 JavaScript が密結合した攻撃へ対応できない” では，現在，マルウェア本体は，MITB 攻撃用 JavaScript を読み込むための挿入コード片の挿入が主な役割であり，挿入コード片や通信先の動的変更も 5.3.4 項で述べたように，わずかな変更を行うのみであるため，改ざん再現システムを使って MITB 攻撃用 JavaScript のみを実行することが可能である．しかし，攻撃者が解析を妨げる等の目的でマルウェアと MITB 攻撃用 JavaScript が密に連携することで成立する攻撃を実施することは可能である．このような攻撃が行われた場合，現在の提案手法では，解析を行うことができない．このような攻撃が行われた場合は，改ざん再現システムでマルウェアをエミュレートする方法の検討や検証実験で行ったようにマルウェア感染環境を用いた解析を実施する必要があると考えられる．

6.4 提案手法の課題

6.4.1 改ざん前後のコンテンツの比較について

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 JavaScript のプロパティ情報等の差異を比較することが難しいという問題がある．JavaScript のプロパティ情報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．

6.4.2 攻撃機能の解析が行えない解析対象について

実験の結果，検体 1, 2 において，情報盗取機能および偽画面表示機能の挙動解析が行えない MITB 攻撃用 JavaScript が存在した．本稿では，MITB 攻撃用 JavaScript が解析対象の攻撃機能を保有すると仮定し，挙動解析ができたか，できなかったかの判定を行っている．しかし，解析対象とする攻撃機能の挙動を確認できなかった MITB 攻撃用 JavaScript は，これらの機能を保有していないことや特定の条件下でのみ動作するといった可能性が考えられる．動的解析のみでは，解析対象の動作した結果しか知り得ないため，今後，難読化解除後の MITB 攻撃用 JavaScript 内のコールフロー，組み込み関数やライブラリ関数等の利用状況，HTML リソースの有無等を自動的に分析することで解析対象の機能の有無や発動条件を明らかにする機能を検討する．また，分析結果に従って環境を変更して解析対象の機能を強制的に動作させることを可能とする必要がある．なお，該当機能の挙動解析が行えた対象においても同様に発動条件等によって異なる挙動をする場合等を解析できていない可能性が考えられる．このように，該当機能の挙動解析が行えた対象に対して発動条件等を変更して網羅的に動作させるためにも必要な機能であると考えられる．

6.4.3 マニピュレーションサーバとの通信再現について

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 のよう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あると考える．その方法として，検証実験で用いたように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 を意図したとおりに動作させる方法が考えられる．これは，MITB 攻撃用 JavaScript の全機能を解明するうえでは後者がより有効であると考える．なお，この全機能とは，マニピュレーションサーバの応答によって，盗取情報や偽画面の内容が変わる等の挙動が変化する実装や，マニピュレーションサーバの指示に従って送金を行う自動送金機能を持つ MITB 攻撃用 JavaScript を想定している．

7. 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 を安全に動的解析する手法について提案した．また，提案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．

本稿では，情報盗取機能・偽画面表示機能等の想定される攻撃機能の解析およびデバッガ機能によるコード分析・難読化の解除・通信ログの収集を解析の目的として，改ざん再現システムを用いて動的解析を行った．その結果，ログイン画面において通常のログイン操作を行った際に発動する情報盗取および偽画面表示の攻撃機能を再現し挙動解析することが可能であることを確認した．このように，提案手法を用いて 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．また，マルウェア感染環境を用いた検証実験を行うことで，改ざん再現システムによる再現結果が正しいことを検証した．提案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させ，解析者の精神的負担の軽減にも貢献すると考える．

しかし，攻撃機能が発動しない解析対象も存在している．これに対しては，解析対象内の該当機能の有無や発動条件等を分析し，その結果に従って環境を変化して，該当機能を強制的に動作させる機能の実現を目指す．これは，攻撃機能の挙動解析が行えている対象においても解析の網羅性を向上させるために必要な機能である．

また，今回は，攻撃対象をログイン画面に限定して実験を行っているため，ログイン後の画面を攻撃対象とする自動送金機能やログイン画面以外を攻撃対象とする EC サイトおよびフリーメールサービス向けの MITB 攻撃用 JavaScript について提案手法の有効性の検証を行う必要がある．

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ともに，システムの機能拡充を行い有効性を高めていきたい．

謝辞 本研究の一部は，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」によって得られた．

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] 岡本勝之：国内クレジットカード 12 社が標的，日本に矛先を向ける「RAMNIT」，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/15252⟩ （参照 2018-11-03）．

[4] 吉川孝志，菅原 圭：オンラインバンキングマルウェア「DreamBot（Ursnif/Gozi）」の今，MBSD Blog（オンライン），入手先 ⟨https://www.mbsd.jp/blog/20180607.html⟩ （参照 2018-10-30）．

[5] 吉川孝志，菅原 圭：隠された（見えない）デスクトップに潜む脅威とその仕組み，MBSD Blog（オンライン），入手先 ⟨https://www.mbsd.jp/blog/20180914.html⟩ （参照 2018-10-30）．

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

[7] 中津留勇：Fight Against Citadel in Japan, JPCERT/CC 分析センター（online），入手先 ⟨http://www.jpcert.or.jp/present/2014/20140218CODEBLUE-Citadel_ja.pdf⟩（参照 2018-11-03）．

[8] Boutin, J.-I.: The Evolution of Webinjects, *Virus Bulletin Conference*, pp.25–34 (2014).

[9] 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).

[10] 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，情報処理学会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).

[11] 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).

[12] 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).

[13] 津田 侑，神薗雅紀，遠峰隆史，安田真悟，三浦良介，宮地利幸，衛藤将史，井上大介，中尾康二：標的型攻撃のシナリオ再現環境の構築，情報処理学会研究報告コンピュータセキュリティ，Vol.2013-CSEC-61, No.18, pp.1–6 (2014).

[14] 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，金融研究，Vol.32, No.3, pp.51–76 (2013).

[15] 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).

[16] 高田一樹，岩本一樹，遠藤 基，奥村吉生，岡田晃市郎，西田雅太，吉岡克成，松本 勉：静的解析と挙動観測を組み合わせた金融マルウェア長期観測手法の提案，情報処理学会論文誌，Vol.59, No.12 (2018).

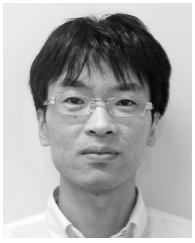
[17] Mizerany, B.: Sinatra (online), available from ⟨http://sinatrarb.com/⟩ (accessed 2018-08-16).

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

[19] 高田一樹，松本英樹，邦本理夫，吉岡克成，松本 勉：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，コンピュータセキュリティシンポジウム 2018 論文集，Vol.2018, No.2, pp.1008–1015 (2018).

高田 一樹 （学生会員）

2003年日本大学工学部情報工学科卒業．2005年同大学大学院博士前期課程修了．2014年株式会社セキュアブレインに入社．主に不正サイトの検知・分析，マルウェアの静的・動的解析，不正送金対策システムの研究開発に従事．2017年横浜国立大学大学院環境情報学府入学．電子情報通信学会会員．



松本 英樹

2004年九州大学工学部電気情報工学科卒業．2006年同大学大学院システム情報科学府知能システム学専攻修士課程修了．同年電子情報通信学会九州支部学術奨励賞受賞．2016年より株式会社セキュアブレインにて不正送金対策システムの研究開発に従事．

邦本 理夫

2000年早稲田大学理工学部情報学科卒業．2002年同大学大学院理工学研究科情報科学専攻修士課程修了．2015年より株式会社セキュアブレインにて不正送金対策システムの研究開発に従事．2019年情報セキュリティ大学院大学博士前期課程入学．

吉岡 克成 （正会員）

2005年横浜国立大学大学院環境情報学府情報メディア環境学専攻博士課程後期修了．博士（工学）．同年独立行政法人情報通信研究機構で研究員として勤務．2007年より横浜国立大学学際プロジェクト研究センター特任教員（助教）．2011年横浜国立大学大学院環境情報研究院准教授．マルウェア解析やネットワーク攻撃観測・検知等のネットワークセキュリティの研究に従事．2009年文部科学大臣表彰・科学技術賞（研究部門），2016年産学官連携功労者表彰総務大臣賞，2017年情報セキュリティ文化賞をそれぞれ受賞．

松本 勉

1986年東京大学大学院工学系研究科電子工学専攻博士課程修了．工学博士．同年より横浜国立大学勤務．現在，同大学・環境情報研究院教授および先端科学高等研究院情報・物理セキュリティ研究ユニット主任研究者および産業技術総合研究所サイバーフィジカルセキュリティ研究センター長．CRYPTREC暗号技術検討会座長，日本学術会議連携会員を兼任．情報・物理セキュリティの研究教育に1981年より従事．この間，日本銀行金融研究所客員研究員，独カールスルーエ大学客員教授，日本学術振興会学術システム研究センター専門研究員，国際暗号学会IACR理事等を歴任．暗号学国際会議ASIACRYPT，暗号と情報セキュリティシンポジウムSCIS等の創設に貢献．電子情報通信学会業績賞，第5回ドコモ・モバイル・サイエンス賞，第4回情報セキュリティ文化賞，2010年文部科学大臣表彰・科学技術賞等受賞．