# 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

西田 雅太　太刀川 剛　岩本 一樹
遠藤 基　奥村 吉生　星澤 裕二

株式会社セキュアブレイン 先端技術研究所
102-0083 東京都千代田区麹町 2-6-7 麹町 RK ビル 4F
{masata_nishida, tsuyoshi_tachikawa, kazuki_iwamoto}@securebrain.co.jp
{motoi_endo, yoshio_okumura, yuji_hoshizawa}@securebrain.co.jp

**あらまし** 近年，金融機関の利用者を攻撃対象としたマルウェアによる被害が急増している．これらのマルウェアは，自身の挙動を変化させるために設定情報を動的に外部から取得する．このため，マルウェアの攻撃対象や攻撃手法を知るためには，マルウェア本体の解析のみならず設定情報の入手と解析が必要となる．本稿では，金融機関の利用者を攻撃対象とする代表的なマルウェアの実行コードと設定情報を静的解析し，その攻撃手法を調査する．また当該検体を一定期間動作させ設定情報の変化を観測する．これらの調査により，金融機関の利用者に対する攻撃がマルウェア本体だけでなく，変化する設定情報を利用した複雑な枠組みで構成されることを示す．

# Research on Attack Methods of Online Banking Malware Based on Static Analysis and Behavior Observation

Masata Nishida　Tsuyoshi Tachikawa　Kazuki Iwamoto
Motoi Endo　Yoshio Okumura　Yuji Hoshizawa

Advanced Research Laboratory, SecureBrain Corporation
Kojimachi RK Building 4F 2-6-7 Kojimachi, Chiyoda-ku, Tokyo, Japan
{masata_nishida, tsuyoshi_tachikawa, kazuki_iwamoto}@securebrain.co.jp
{motoi_endo, yoshio_okumura, yuji_hoshizawa}@securebrain.co.jp

**Abstract** The incidence of online banking malware has been increasing. These malware would obtain configuration settings from C&C servers and change their own behavior based on those settings. To understand a malware's attack methods and targets, we would need to analyze not just the malware on its own but also its corresponding configuration. In this paper, we will analyze an online banking malware and its configuration. We will also observe this malware's behavior to know more about the configuration changes. Based on this analysis, we can then show the complexity of an online banking malware's attack framework.

## 1 はじめに

近年，金融機関の利用者を攻撃対象としたマルウェア（以降，金融系マルウェア）が増加し，それに伴う不正送金被害額が増加している[1]．金融系マルウェアに感染した利用者が攻撃対象のインターネットバンキングサイトにアクセスすると，Man In the Browser（MITB）攻撃とい

われる手法によりブラウザ上のコンテンツが改ざんされ，アカウント情報などが窃取される．

金融系マルウェアのなかには，攻撃対象などの情報を内包せず，外部サーバなどから設定情報を取得し，その設定情報により挙動を変えるものが存在する．

設定情報により動的に挙動が変わるマルウェアの場合，マルウェア本体の静的解析ではマルウェアが持つ機能しか把握が出来ない．マルウェアが実際に行う攻撃の内容を知るためには，設定情報も含めた解析を行う必要がある．

そこで本稿では，実際の金融系マルウェアの実行コードと設定情報を共に静的解析し，金融機関の利用者に対する攻撃手法を調査する．また，検体の挙動を 1 ヶ月に渡り観測し，設定情報の変化の様子を調査する．

これらの調査により，金融機関の利用者に対する攻撃がマルウェア本体だけでなく，変化する設定情報を利用した複雑な枠組みで構成されることを示す．

# 2 解析対象のマルウェア

本稿では，金融系マルウェアとして最近猛威を振るっている VAWTRAK や Neverquest[2] などと呼ばれるマルウェアを解析対象とする．表 1 に解析対象マルウェアの概要を示す．検体は著者らが独自に入手したものである．以下，本検体を検体 A と呼ぶ．

**表 1 解析対象検体概要**

| | |
|---|---|
| ファイルサイズ | 264,817 bytes |
| 入手日 | 2014/5/29 |
| ファイル形式 | PE(DLL) |

# 3 金融系マルウェアの解析

## 3.1 検体の静的解析

### 3.1.1 検体概要

検体 A は DLL 形式であり，単独では実行できないため，regsrv32 コマンドを使って実行する．

検体 A はアンパック処理後，起動中の他のプロセスのメモリを VirtualAllocEx で取得して実行コードを展開し，CreateRemoteThread により実行する．これらの処理により実行コードを既存のプロセスに注入する．新規にプロセスが起動した場合にも，そのプロセスに対して実行コードを注入する．

検体 A は Command & Control(C&C)サーバと通信し，実行すべきコマンドや設定情報を取得する．また，検体 A は FTP クライアントソフトのアカウント情報の窃取やバックドアなどの機能を持つが，本稿では金融機関の利用者に対する攻撃に関連した機能を中心に解析を行う．

### 3.1.2 MITB 攻撃

検体 A は，ブラウザが通信で利用する API をフックすることによって MITB 攻撃を実現する．

#### 3.1.2.1 API フック

検体 A は実行コードの注入先プロセスがブラウザだった場合，ブラウザが通信で利用する API のフック処理を行う．

**表 2 ブラウザプロセスの API フック**

| ブラウザ(DLL) | 対象 API |
|---|---|
| **Internet Explorer** (wininet.dll) | InternetWriteFile<br>InternetSetOptionW<br>InternetSetOptionA<br>InternetReadFileExA<br>InternetReadFile<br>InternetQueryOptionW<br>InternetQueryOptionA<br>InternetQueryDataAvailable<br>InternetConnectW<br>InternetConnectA<br>InternetCloseHandle<br>HttpSendRequestW<br>HttpSendRequestA<br>HttpSendRequestExW<br>HttpSendRequestExA<br>HttpQueryInfoA<br>HttpOpenRequestW<br>HttpOpenRequestA<br>HttpEndRequestW<br>HttpEndRequestA |
| **Firefox** (NSPR4.DLL NSS3.DLL) | PR_Write<br>PR_Read<br>PR_Close |
| **Chrome** (chrome.dll) | Kernel32.dll LoadLibrary への API フックによりchrome.dll の読込時に関数テーブルから通信用 API のアドレスを調べてパッチをあてる. |

これにより，ブラウザの通信の改ざんを行う．Internet Explorer，Firefox，Chrome の 3 つのブラウザが MITB の対象となる．攻撃対象の各ブラウザにおいて API フックが行われる DLL および API を表 2 に示す．

#### 3.1.2.2 API フックによるブラウザ通信の操作

検体 A は，利用者がブラウザで特定 URL にアクセスした際に，API フックにより次のような操作を行う．

- 特定 URL への通信の遮断
- 受信データの外部サーバへの送信
- 受信データの文字列置換
- 接続先 URL の変更

改ざんの対象となる URL や改ざん方法・内容は，C&C サーバから取得する設定情報に記載されている．改ざん方法の詳細については「3.1.4.2 設定情報のフォーマット」で述べる．

### 3.1.3 C&C サーバとの通信

検体 A は C&C サーバと通信を行う．検体内部には複数のホスト名がハードコーディングされている．検体は，これらのホストからコマンドや設定情報を取得したり，ホストに対して検体が収集した情報を送信したりする．検体 A には 9 つのドメインと 6 つの IP アドレスがハードコーディングされていた．

HTTP プロトコルの POST リクエストによってコマンドの取得が行われる．その際，リクエストには実行されている環境ごとに異なる ID や，設定情報の復号に用いられるシード値（3.1.4.2 で後述）がパラメータとして含まれる．

検体と C&C サーバとの通信は，ブラウザのプロセスが起動したときおよび，その後 10 分に 1 回の頻度で行われる．

### 3.1.4 C&C コマンドと設定情報

#### 3.1.4.1 コマンドデータフォーマット

C&C サーバから HTTP リクエストによって取得できるデータのフォーマットを図 1 に示す．データは $n$ 個（$n \geqq 0$）のコマンドで構成される．データは”ok”という文字列ではじまり，その次にコマンド数が記述される．続いてそのコマンド個数分のコマンドが配置される．コマンドは，コマンド種別（Cmd type），データ長（Cmd length），データ（Cmd data）で構成されている．

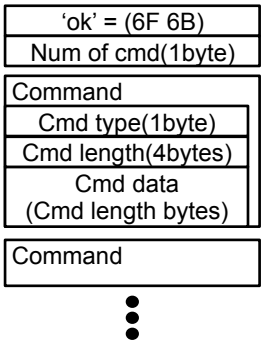


**図 1 コマンドデータフォーマット**

図 1 のコマンドのコマンド種別の一部を表 3 に示す．

**表 3 コマンド種別（一部抜粋）**

| 種別 | コマンド |
|---|---|
| 0 | 設定情報 |
| 1 | 外部プログラムの実行 |
| 2 | ダウンロード&実行 |
| 6 | プロセスリストの取得 |
| 7 | Cookie の削除 |
| 14 | マルウェアの更新 |
| 20 | ファイル送信 |

コマンド種別が 0 のとき，データは設定情報を表す．設定情報には MITB 攻撃の改ざん対象，改ざん方法などが記述される．

#### 3.1.4.2 設定情報のフォーマット

コマンドに含まれる設定情報は簡易的な暗号化が施されている．データの先頭 4byte がシード値になっており，このシード値と排他的論理和演算やビット演算を組み合わせて復号する．復号できたデータはさらに aPLib[3]で圧縮されており，これを展開すると図 2 の構造の設定情報を得ることが出来る．

設定情報は以下の 3 つのセクションで構成されている．

- 改ざん設定セクション（図 2 Injection Command）
- 文字列セクション（図 2 Some String）
- URL 文字列セクション（図 2 URL String）

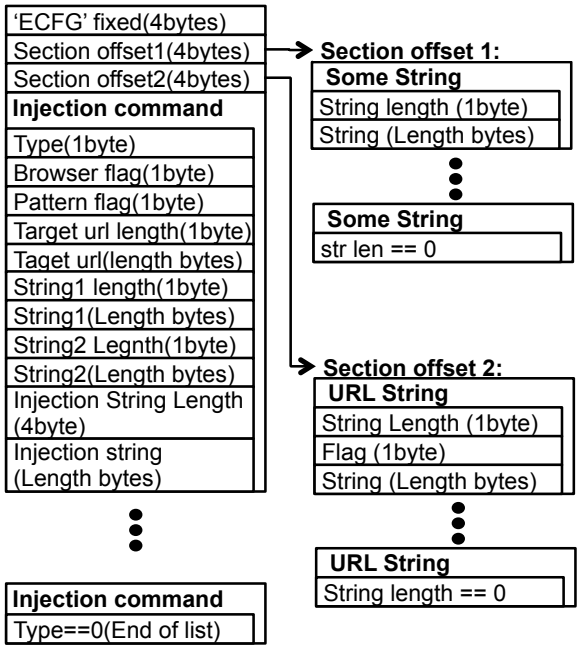


**図 2 設定情報データフォーマット**

本稿では 3 つのセクションのうち MITB 攻撃に使用される改ざん設定セクションのみに注目する. 改ざん設定セクションは複数の改ざんコマンドで構成される. 改ざんコマンドのフォーマットを表 4 に示す. このうち String1, 2 および Injection String は, 改ざんの条件や改ざん後の文字列などを表現するのに用いられる.

**表 4 改ざんコマンドフォーマット**

| フィールド | 説明 |
|---|---|
| Type | 改ざん種別 |
| Browser flag | 対象ブラウザフラグ<br>IE, Firefox, Chrome |
| Pattern flag | 文字列の比較方法<br>strstr, regexp, strcmp |
| Target URL | 対象 URL |
| String1 | 条件文字列 1 |
| String2 | 条件文字列 2 |
| Injection String | 挿入（変更）文字列 |

次に改ざんコマンドのコマンド種別の一部を表 5 に示す.

改ざんコマンドはコンテンツ文字列の置換以外に, HTML コンテンツの保存, 接続先 URL の変更などのコマンドも存在する.

**表 5 改ざん種別（一部抜粋）**

| 種別 | 意味 |
|---|---|
| 0 | データの終了 |
| 1 | 通信の終了 |
| 3 | 受信データの保存 |
| 8 | ページ全体の置換 |
| 10 | String1 から String2 までを置換 |
| 12 | String1 でパターンマッチによる置換 |
| 13 | String1 の前に挿入 |
| 14 | String1 の後に挿入 |
| 17 | 通信先 URL の置換 |

## 3.2 金融機関への攻撃手法の調査

検体 A を仮想環境上で動作させ取得できた設定情報を解析し, 実際の攻撃手法について調査する. 本節では, 2014 年 7 月 10 日に C&C サーバから取得できた設定情報（以降, CFG-0710 とする）を復号し, 解析を行う.

### 3.2.1 改ざん対象

CFG-0710 では国内の銀行 5 行とカード会社 5 社のサイトが改ざん対象になっており, 国内の金融機関の利用者のみが標的になっている.

また, 1 つの金融機関に対し, 複数の改ざんコマンドが定義されていた. 表 6 に改ざん対象と, 改ざんコマンドの数を示す.

**表 6 改ざん対象と改ざんコマンド数**

| 対象 | 対象数 | 改ざんコマンド数 |
|---|---|---|
| 銀行 | 5 | 19 |
| カード会社 | 5 | 13 |
| 計 | 10 | 32 |

### 3.2.2 改ざん手法

CFG-0710 における, MITB 攻撃による主な改ざん内容を以下に示す.

- JavaScript コード片の挿入
- 不正送金注意喚起の div 要素の非表示化
- 不正送金注意喚起ページをリダイレクトでスキップ
- 受信した HTML の保存・送信

CFG-0710 では，改ざん対象ページにアクセスすると受信した HTML データの head 要素の直後に JavaScript のコード片が挿入される．このコード片は情報の窃取などを行う攻撃の本体となる JavaScript を攻撃者のサーバから呼び込むためのコードとなる．挿入される JavaScript コード片の例を図 3 に示す．

```
<script jve=1>(function(){try{var e="/bmhnno/?c=script&v=3";var t="%user_id%";if(t.length!=9){e+="&b="+encodeURIComponent(t)}var n=document.getElementsByTagName("head")[0];var r=document.createElement("script");if(r&&n){r.jve=1;r.src=e;n.appendChild(r)}}catch(i){}})()</script><
```

**図 3 改ざん対象に挿入されるコード片例**

図 3 のスクリプトを実行することによってブラウザの DOM に挿入される要素を図 4 に示す．

```
<script jve=1 src="/bmhnno/c=script&v=3...">
```

**図 4 コード片により DOM に挿入される要素**

### 3.2.3 マニピュレーションサーバとの通信

CFG-0710 には，表 7 に示す改ざんコマンドが存在する．この設定は，HTTP のリクエスト URL に”/bmhnno/”を含む場合，接続先の URL を String2 文字列で置き換える．

**表 7 通信先を変更する改ざんコマンド**

| | |
|---|---|
| Cmd type | 17 (通信先 URL の置換) |
| Pattern flag | strstr |
| Target URL | /bmhnno/ |
| String1 | .*/bmhnno/(.*) |
| String2 | https://crono■■■.com/$1 |
| Injection String | なし |

図 4 の要素は，もとの HTML を取得したホストに対してスクリプトを要求する要素に見えるが，表 7 の改ざんコマンドによって，URL が攻撃者のサーバへのリクエストに書き換えられる．このリクエストにより，偽画面の表示や情報の窃取など実際の攻撃行う JavaScript（以降，攻撃 JavaScript）が取得できる．

この攻撃者のサーバは攻撃 JavaScript を配信するだけでなく，攻撃 JavaScript が窃取したアカウント情報のアップロード先となる．また，利用者のブラウザ上から自動送金を行う（3.2.5 で後述）際の振り込み先口座情報の配信も行う．本稿では，このサーバを C&C サーバと区別するためにマニピュレーションサーバと呼ぶ．表 7 の String2 に記載された URL はマニピュレーションサーバを表す．

マニピュレーションサーバとの通信の様子を簡略化した図を図 5 に示す．

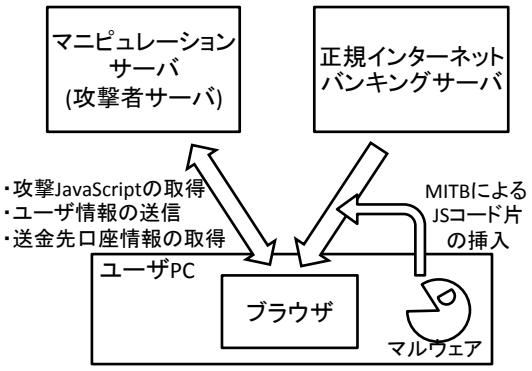


**図 5 マニピュレーションサーバとの通信**

### 3.2.4 攻撃 JavaScript

攻撃 JavaScript は，インターネットバンキングサーバが配信した正規のログインフォームに利用者が入力した情報をマニピュレーションサーバに送信する機能がある．正規のフォームに本来存在しない PIN コードなどのフィールドを追加して認証情報を盗むパターンも見られた．

1 つの金融機関に対する攻撃 JavaScript は 1 種類で，同じインターネットバンキングサイトでは複数のページで同じ攻撃 JavaScript が読み込まれる．攻撃 JavaScript は URL や HTML の構成要素から表示されているページを判定し，ページに適合した改ざんや情報の窃取を行う．

CFG-0710 では，インターネットバンキングの利用者への攻撃に用いられるマニピュレーションサーバは 1 つであった．マニピュレーションサーバは，HTTP リクエストのリファラでリクエスト元の金融機関を区別し，それぞれの金融機関に対応した攻撃 JavaScript を配信する．

### 3.2.5 ブラウザ上からの半自動送金処理

攻撃 JavaScript には，マニピュレーションサーバから指示のあった口座に振り込みを半自動

的に行う機能が存在する.

金融機関の利用者がインターネットバンキングサイトにログインし，口座残高などが表示されるページに遷移した際に，攻撃 JavaScript は残高などの情報をマニピュレーションサーバに送信する. そのレスポンスに振り込み先口座情報や振り込み金額が含まれていた場合に，攻撃 JavaScript は自動送金を開始する. 振り込み処理の実行中は図 6 のようなプログレスバーが表示され，利用者による操作を抑止する.

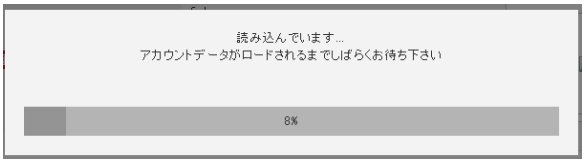


**図 6 自動送金時のプログレスバー表示**

プログレスバーが表示されている間，攻撃 JavaScript は XMLHttpRequest を使って正規インターネットバンキングサーバに対して，振り込みに必要な HTTP リクエストを逐次発行する. その際，ワンタイムパスワードなどの入力が必要になると，偽の入力画面を表示して利用者に入力を促し，振り込み処理を実行する.

自動送金機能は攻撃対象となっている銀行 5 行のうち，3 行で確認された.

# 4 マルウェアの挙動観測

## 4.1 観測方法

### 4.1.1 観測期間・環境

検体 A を表 8 の環境で動作させ挙動の観測を行った.

**表 8 観測環境**

| | |
|---|---|
| 仮想環境 | KVM |
| ホスト OS | CentOS 6.5 |
| ゲスト OS | Windows7 (32bit) |
| ブラウザ | Internet Explorer 8 |

観測は下記の期間で実施した.

- **観測開始**: 2014/7/10 10:00
- **観測終了**: 2014/8/18 23:59

### 4.1.2 マルウェア感染環境の構築と観測

regsrv32 コマンドを使って検体 A を実行し，仮想環境上の Windows 端末に検体を感染させる. 次に，ブラウザを起動して C&C サーバから設定情報を取得することで感染環境を構築する.

検体 A はブラウザのプロセスから 10 分に 1 回の頻度で C&C サーバからコマンドを取得する. そこで，ブラウザを起動したままの状態にして，Wireshark を用いてパケットを収集する.

収集したパケットを解析し，C&C サーバとの通信のみを抽出する. 静的解析の結果に基づいて作成したコマンドと設定情報のデコーダを使ってコマンドおよび設定情報の抽出を行う.

## 4.2 観測結果

### 4.2.1 観測結果概要

表 9 に挙動観測結果の概要を示す.

**表 9 観測結果概要**

| 観測項目 | 回数 |
|---|---|
| C&C へのコマンド要求 | 5812 |
| 通信を行ったホスト数 | 9 |
| 検体の更新 | 1 |
| 設定情報の更新 | 10 |

### 4.2.2 検体の更新

7/14，マルウェアのアップデートコマンドにより，検体がアップデートされた. 更新された検体については詳細な解析は行っていないが，VAWTRAK の亜種であることが確認されたため，更新された検体で観測を継続した. 以下，本検体を検体 B と呼ぶ.

### 4.2.3 設定情報の更新

観測期間中に設定情報の更新は 10 回観測された. 設定情報の更新の概要を表 10 に示す. なお，表 10 の ID は設定情報を識別するために著者が付与した.

設定情報の更新時間や更新間隔には規則性は見られなかった. また，更新があった直後に再び更新があるケースが 3 回観測された.

表 10 設定情報の更新状況

| ID | 更新日時 | 更新内容 |
|---|---|---|
| CFG-0715a | 2014/07/15 06:12 | カード会社 20 社への攻撃追加 |
| CFG-0715b | 2014/07/15 07:03 | 機能してない一部の攻撃削除 |
| CFG-0718a | 2014/07/18 01:50 | 一部銀行の改ざんスクリプトの修正 |
| CFG-0718b | 2014/07/18 03:50 | 地銀 11 行への攻撃追加 |
| CFG-0729 | 2014/07/29 19:38 | JS の改ざんコード追加 |
| CFG-0731 | 2014/07/31 22:45 | JS の改ざんコード追加 |
| CFG-0808 | 2014/08/08 21:01 | 古い設定情報へ回帰 |
| CFG-0812 | 2014/08/12 18:24 | 7/31 相当の設定に戻る |
| CFG-813a | 2014/08/13 19:37 | マニピュレーションサーバのホストの変更 |
| CFG-0813b | 2014/08/13 19:47 | カード会社に対する攻撃が削除 |

#### 4.2.3.1 攻撃対象の変化

観測開始時の設定情報 CFG-0710 では，5 社のカード会社が攻撃対象となっていたが，カード会社攻撃に使われていたマニピュレーションサーバは停止しており，実際の被害は発生していなかった．設定情報 CFG-0715a では，既存の 5 社を含む国内カード会社 20 社が攻撃対象となった．この変更では 5 つの新規マニピュレーションサーバを利用していた．ただし，これらのサーバは数日ほどですべて停止した．その後，CFG-0813b の更新によりカード会社に対する攻撃が削除された．

一方，CFG-0718b では，地銀 11 行が攻撃対象として追加された．攻撃には，3.2.3 で示した既存のマニピュレーションサーバが利用されていた．マニピュレーションサーバから 11 行に対する攻撃 JavaScript を取得してみたところ，全て同一の JavaScript であった．攻撃 JavaScript 内に金融機関ごとに挙動を変えるようなコードも存在するが，基本的な攻撃は同一であり，新たに攻撃対象となった地銀 11 行が同一または類似のインターネットバンキングシステムを使用しているのではないかと推測できる．

#### 4.2.3.2 マニピュレーションサーバの変化

CFG-0813a では，銀行利用者に対する攻撃を行うマニピュレーションサーバのホストが変更された．その際，変更前のマニピュレーションサーバは停止したわけではなく，継続して稼働していた．

また，ホストの変更前後で配信される攻撃 JavaScript にほとんど変化が見られなかったことから，2 つのサーバは同一の攻撃者によって運用されていることが推測できる．

#### 4.2.3.3 特定 JavaScript の無効化処理

CFG-0729，CFG-0731 では，ある銀行の特定 JavaScript に対する改ざんが行われていた．改ざん対象の JavaScript は難読化され 1 行で記述されており，CFG-0729 では，先頭に"//"の文字列挿入し，全体をコメントアウトして無効化する処理がなされていた．また，CFG-0731 では，同 JavaScript の URL に対するリクエストの HTTP コネクションを閉じるような処理に変更された．攻撃者がなんらかの目的で対象の JavaScript を無効化しようとしていたのではないかと推測できる．

### 4.2.4 リバースプロキシ利用の可能性

8/1 から 8/8 にかけて複数のサーバに対する通信が連続的に失敗する事象が観測された．

検体 B は C&C サーバへの通信が失敗すると検体にハードコーディングされている別のホストに対して通信を行う．

一方，通信が失敗するときの C&C サーバからの応答 HTTP ステータスコードは 502，504 であった．502 は"Bad Gateway"を，504 は"Gateway Timeout"を表す．

図 7 に観測期間中の C&C サーバの応答 HTTP ステータスコードの推移を示す．

通信失敗時のステータスコードが Gateway 系のエラーであることや，複数のサーバで同時期に通信が失敗していることから，検体にハードコーディングされているホストはリバースプロキシサーバとして動き，実際の C&C サーバはリバースプロキシの上位に存在する構成になっていることが考えられる．上位の C&C サーバがダウ

ンするなど通信できなくなった場合にリバースプロキシが 502, 504 の応答になったのではないかと推測できる.

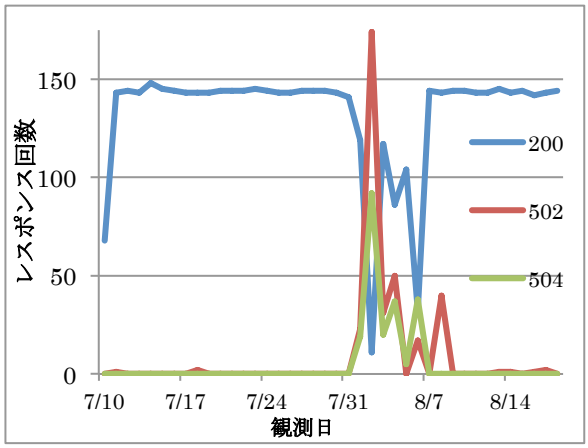


**図 7 C&C サーバの応答ステータスコード**

図 8 に観測した現象から推測できるサーバの構成図を示す.

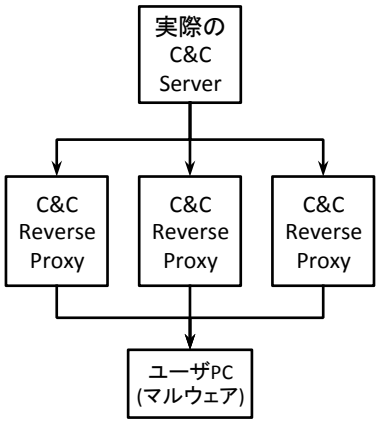


**図 8 推測される C&C サーバの構成**

## 5 考察

「3.2 金融機関への攻撃手法の調査」で示したとおり, 解析対象の検体では, 実際の攻撃手法・対象は設定情報に記述されている. また, 実際の攻撃はさらにマニピュレーションサーバから取得した攻撃 JavaScript により実行されていた. このことから, マルウェア本体は攻撃を行うためのプラットフォームとして動作していると考えることが出来る. マルウェア本体が更新されなくても, 設定情報やマニピュレーションサーバから配信される JavaScript が更新されることにより, 攻撃を変化させることが可能なフレームワークで動作している.

「4 マルウェアの挙動観測」では 1 ヶ月以上の期間, 設定情報の更新状況を観測した. この間, 攻撃対象, マニピュレーションサーバのホスト, 改ざん内容など複数の変化が観測され, 金融機関の利用者に対する攻撃が常に変化していることが分かった.

今回の観測ではマルウェアを実際に動作させることで設定情報の変化などを取得したが, C&C サーバやマニピュレーションサーバのプロトコルを詳細に解析することによって, 各サーバに対して直接リクエストを発行して情報の取得を行うことも可能である.

## 6 おわりに

本稿では, 金融系マルウェアを設定情報と合わせて解析を行った. そして, 設定情報の変化の観測を実施した. これにより, 金融機関の利用者に対する攻撃が, C&C サーバやマニピュレーションサーバを用いた複雑な枠組みの中で行われている現状を示した. また, 攻撃を正確に把握するために, マルウェア本体のみならず, その攻撃のフレームワーク全体を解析することが重要であることを示した.

今後は他のマルウェアについても設定情報などの取得と観測に取り組んでいきたい.

**謝辞** 本研究の一部は, 総務省情報通信分野における研究開発委託「国際連携によるサイバー攻撃の予知技術の研究開発」により行われた. ご協力頂いた皆様に, 謹んで感謝の意を表する.

## 参考文献

[1] 平成 25 年中のインターネットバンキングに係る不正送金事犯の発生状況等について, http://www.npa.go.jp/cyber/pdf/H260131_banking.pdf
[2] 新たな標的を狙う Neverquest の進化形, http://www.symantec.com/connect/ja/blogs/neverquest-0
[3] aPLib, http://www.ibsensoftware.com/products_aPLib.html