

SpyEye 解析レポート

SpyEye vs FFRI Limosa



Fourteenforty Research Institute, Inc.

株式会社フォティーンフォティ技術研究所



目次

目次	2
著作権	3
免責事項	3
更新履歴	4
文書情報	4
1. はじめに	5
2. SpyEye の概要	6
2.1. 動作概要	7
2.2. システムへのインストール	7
2.3. 感染活動	8
2.4. 他プロセスへの影響	9
2.5. SpyEye 隠ぺいルーチン	9
2.6. HTML 改ざんルーチン	10
3. SpyEye vs FFRI Limosa	11
3.1. HTML インジェクション	11
3.2. スクリーンキャプチャ	12
4. まとめ	14



著作権

当文書内の文章・画像等の記載事項は、別段の定めが無い限り全て株式会社フォティーンフォティ技術研究所（以下、フォティーンフォティ）に帰属もしくはフォティーンフォティが権利者の許諾を受けて利用しているものです。これらの情報は、著作権の対象となり世界各国の著作権法によって保護されています。「私的使用のための複製」や「引用」など著作権法上認められた場合を除き、無断で複製・転用することはできません。

免責事項

当文書は AS-IS (現状有姿)にて提供され、フォティーンフォティは明示的かつ暗示的にも、いかなる種類の保証も行わないものとします。この無保証の内容は、商業的利用の可能性・特定用途への適応性・他の権利への無侵害性などを保証しないことを含みます。たとえフォティーンフォティがそうした損害の可能性について通知していたとしても同様です。また「この文書の内容があらゆる用途に適している」あるいは「この文書の内容に基づいた実装を行うことが、サードパーティー製品の特許および著作権、商標等の権利を侵害しない」といった主張をも保証するものではありません。そして無保証の範囲は、ここに例示したものだけに留まるものではありません。

また、フォティーンフォティはこの文書およびその内容・リンク先についての正確性や完全性についても一切の保証をいたしかねます。

当文書内の記載事項は予告なしに変更または中止されることがありますので、あらかじめご了承ください。



SpyEye 解析レポート – SpyEye vs FFRI Limosa

更新履歴

2013-02-01 Ver.1.0 岡野 友輔

文書情報

発行元 : 株式会社フオティーンフオティ技術研究所

連絡先 : 株式会社フオティーンフオティ技術研究所

sales@fourteenforty.jp

〒150-0013

東京都渋谷区恵比寿 1 丁目 18 番 18 号 東急不動産恵比寿ビル 4F

1. はじめに

近年、オンラインバンキングの利用者を狙ったマルウェアとして SpyEye が流行しています。下表は「2011 年下半期 Tokyo SOC 情報分析レポート¹」より引用した SpyEye 検体と C&C サーバー間の通信件数の推移です。

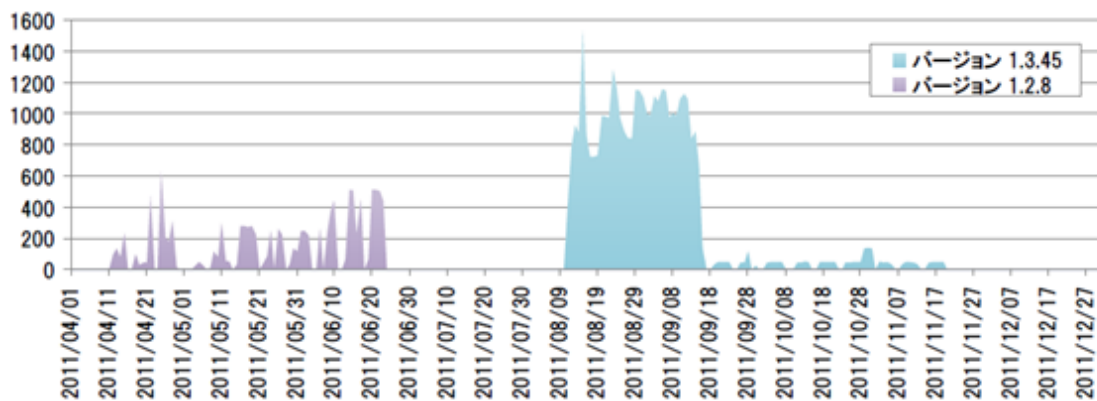


図 1-1 感染した SpyEye による C&C 通信検知件数の推移

上記から SpyEye のバージョン毎に感染が広がる時期があることがわかります。今後も SpyEye のバージョンアップなどで感染が拡大する可能性があります。

¹ http://www-935.ibm.com/services/jp/its/pdf/tokyo_soc_report2011_h2.pdf

2. SpyEye の概要

SpyEye の構成は、下記の 3 つに分けられます。

- ・ マルウェア本体 … 感染活動を行う実行ファイル
- ・ ビルダ― … マルウェア本体生成ツール
- ・ C&C サーバー … 感染したコンピュータの管理・データ収集用

C&C サーバーは Web UI を備えており、様々な操作を Web 上から行うことができます。マルウェアの本体は機能・設定をカスタマイズできるように、下図のようなビルダ―と呼ばれる GUI のツールを用いて生成します。



図 2-1 SpyEye ビルダ―GUI

SpyEye 解析レポート – SpyEye vs FFRI Limosa

ビルダーには複数の設定項目が用意されており、項目の有効・無効で生成するマルウェア本体に付加する機能をカスタマイズできます。また、これ以外にもプラグインを用意することによって、より高度なカスタマイズを行うことができます。

我々の部署では、SpyEye ビルダーを利用し、このビルダーで生成した検体を解析しました。

2.1. 動作概要

SpyEye の動作のイメージは下記ようになります。

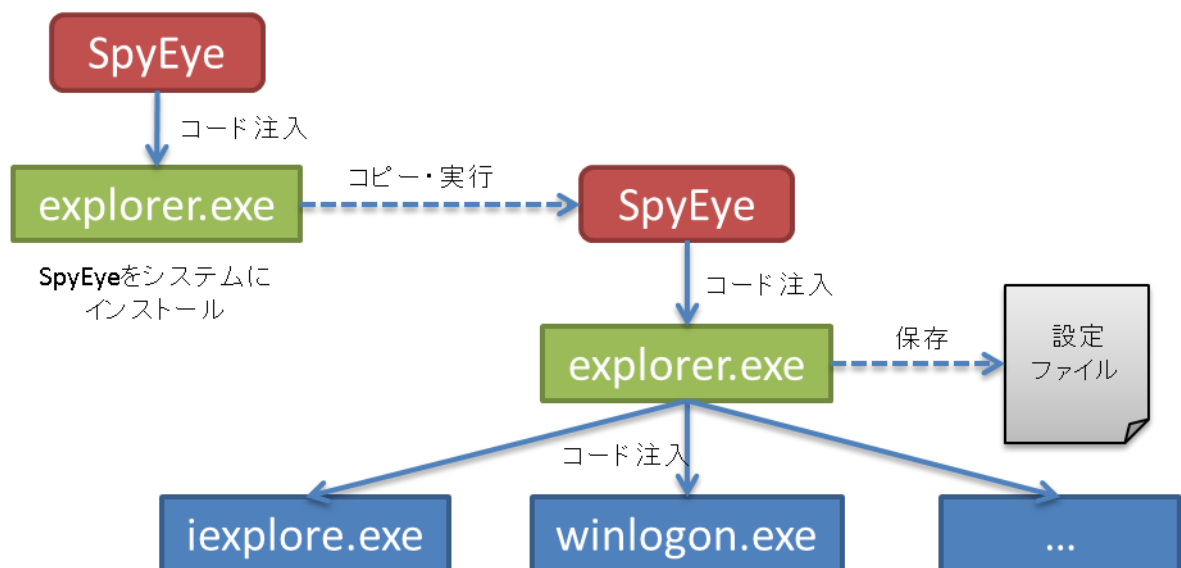


図 2-1 SpyEye 動作イメージ

2.2. システムへのインストール

SpyEye は、自身の実行ファイルに埋め込まれた複数のリソースデータをロードします。リソースは下記の 5 つが用意されています。

- C1: インストール時の設定項目が記載
- C2: 設定ファイル + config.bin データ



SpyEye 解析レポート – SpyEye vs FFRI Limosa

- C3: config.bin 展開用パスワード
- SC1:コード注入用データ
- SC2:情報収集用データ

次に、CreateToolhelp32Snapshot を利用して実行中のプロセス一覧を取得し、explorer.exe に SC1 のコードを注入します。コード注入の手法としては、OpenProcess で explorer.exe のハンドルを取得し、そのハンドルで識別されるプロセスのメモリ領域を NtAllocateVirtualMemory で確保し、確保したメモリに NtWriteVirtualMemory を使用して SC1 のコードを書き込みます。一般的にマルウェアがコード注入を行う際は、この後に CreateRemoteThread を利用しますが、SpyEye は NtClose の先頭アドレスを書き換え、その後に CloseHandle を呼ぶことで注入したデータに制御を移しています。

注入したコードでは自身のコピーを作成し、オリジナルファイルを削除します。最後にコピーしたファイルを実行します。以上で explorer.exe に注入したコードのスレッドは終了します。

2.3. 感染活動

コピーしたファイルが実行されると、検体内部に持っている暗号 ZIP された設定ファイルをフォルダに保存します。次にリソースデータ C3 のパスワードで ZIP ファイルを展開し、内容を取得します。その後、再度 explorer.exe にコードを注入します。

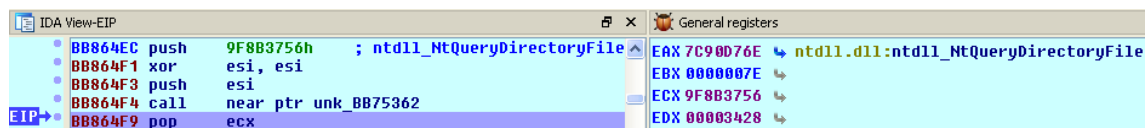
再度注入されたコードでは、設定ファイルに書かれた内容に従って処理を実施し、特定のシステムプロセスを除いたすべての起動中のプロセスにインストール時と同じ手法でコード注入を行います。

2.4. 他プロセスへの影響

explorer.exe に 2 度目に注入されたコードでは、他のプロセスにコード注入を行います。プロセスに注入されたコードでは、SpyEye 検体をシステムから隠ぺいするためのルーチンとブラウザ上で表示される HTML を改ざんするルーチンが呼び出されます。

2.5. SpyEye 隠ぺいルーチン

このルーチンは、実行された SpyEye 検体をシステムから隠ぺいするための処理を行うルーチンです。次のようにハッシュ化された数値を引数にアドレス取得ルーチンを読み出すことで API の先頭アドレスを取得します。



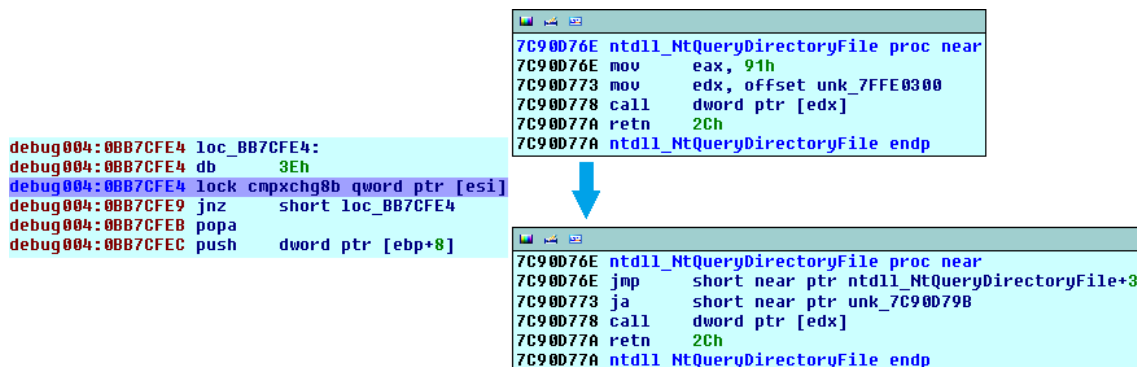
```

IDA View-EIP
BB864EC push 9F8B3756h ; ntdll_NtQueryDirectoryFile
BB864F1 xor esi, esi
BB864F3 push esi
BB864F4 call near ptr unk_BB75362
EIP-> BB864F9 pop ecx

General registers
EAX 7C90D76E ntdll.dll:ntdll_NtQueryDirectoryFile
EBX 0000007E
ECX 9F8B3756
EDX 00003428
  
```

図 2-5-1 隠ぺいルーチン

取得したアドレスに対してデータを上書きするルーチンを読み出し、API の先頭アドレスを改ざんします。実際に書き換えているのは下記の処理です。



```

debug004:0BB7CFE4 loc_BB7CFE4:
debug004:0BB7CFE4 db 3Eh
debug004:0BB7CFE4 lock cmpxchg8b qword ptr [esi]
debug004:0BB7CFE9 jnz short loc_BB7CFE4
debug004:0BB7CFEB popa
debug004:0BB7CFEC push dword ptr [ebp+8]

7C90D76E ntdll_NtQueryDirectoryFile proc near
7C90D76E mov eax, 91h
7C90D773 mov edx, offset unk_7FFE0300
7C90D778 call dword ptr [edx]
7C90D77A retn 2Ch
7C90D77A ntdll_NtQueryDirectoryFile endp

7C90D76E ntdll_NtQueryDirectoryFile proc near
7C90D76E jmp short near ptr ntdll_NtQueryDirectoryFile+3
7C90D773 ja short near ptr unk_7C90D79B
7C90D778 call dword ptr [edx]
7C90D77A retn 2Ch
7C90D77A ntdll_NtQueryDirectoryFile endp
  
```

図 2-5-2 処理図



SpyEye 解析レポート – SpyEye vs FFRI Limosa

API の改ざんを行う対象としては、例えば NtQueryDirectoryFile や NtEnumerateValueKey、NtResumeThread などがあります。これらを改ざんすることによって、API フックを仕掛け、SpyEye 検体の痕跡（例えば、コピー後の実行ファイルの保存フォルダなど）がシステムから閲覧できなくなります。

2.6. HTML 改ざんルーチン

このルーチンは、複数のネットワーク系 API を改ざんし、設定ファイルで設定された情報の搾取やブラウザが表示する HTML を上書きするための処理を行うルーチンです。

次に、フォルダに保存された設定ファイルの内容を読み取ります。また、HTML インジェクション用の設定を行います。その後、前述したアドレス取得ルーチンを呼び出し、取得した API のアドレスの先頭を改ざんします。

API の改ざんを行う対象としては、例えば HttpOpenRequestA(wininet.dll)や InternetReadFile(wininet.dll)、send(ws2_32.dll)などが挙げられます。これらの API はブラウザが通信をするうえで基本的には必ず利用する API です。よって、これらを改ざんして API フックを仕掛けることによって、設定ファイルで設定された情報を搾取したり、ブラウザに表示する内容を書き換えたりすることが可能になります。



3. SpyEye vs FFRI Limosa

今回の解析に利用した SpyEye 検体を FFRI Limosa(以後 Limosa と表記)で防御できるか検証しました。Limosa は弊社が 2012 年 11 月にリリースした製品で、ID・パスワードの搾取や新しい脅威である MITB (Man in the Browser) 攻撃から Web ブラウザを保護するシステムです。

検証の結果、Limosa が有効になっている Internet Explorer では SpyEye が標準で実施できる攻撃²を防御できることが確認できました。

² SpyEye はこれら以外にも、プラグインを別途購入することで、クレジットカード情報の搾取や SOCKS5 による遠隔操作を行うことができます。

3.1. HTML インジェクション

本検体は、「yahoo.co.jp」という文字列を含む URL に対しては、ブラウザに表示される HTML の title 要素の内容を「xxx:ttt」を付加して表示します。図 3-1 のように、Limosa が有効な IE (図下の緑枠で囲まれたウィンドウ) ではタイトルの改ざんが発生していません。今回は単純化のため、タイトル要素のみを改ざんしましたが、SpyEye 検体はクレジットカード番号の入力を求めるポップアップ画面を表示するなどの高度な改ざんも可能です。

なお、SpyEye はサーバー側のサイトを改ざんするわけではないため、検証に利用したサイト自体は安全であり、本書の読者がアクセスしても問題ありません。

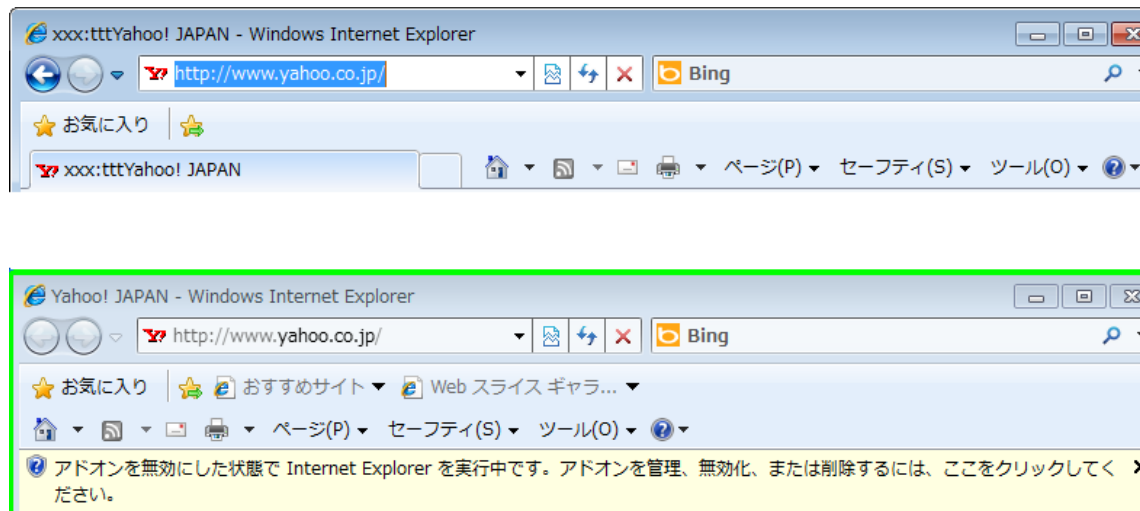


図 3-1 ブラウザ画面

SpyEye は HTML インジェクションを行うにあたり、はじめに IE プロセスへコード注入をします。その後、ネットワーク API をフックして HTML インジェクションを行います。Limosa はこのようなコード注入を防御するために、マルウェアからの Web ブラウザプロセスへのメモリ操作を制限することで、HTML インジェクションを阻止しています。

3.2. スクリーンキャプチャ

本検体は、「yahoo.co.jp」という文字列を含む URL にアクセスし、発動条件を満たすと、一定時間スクリーンキャプチャを取得して C&C サーバーにデータを送信します。Limosa が有効な IE では、スクリーンキャプチャは取得されず、C&C サーバーにもデータは送信されません。

下図は Limosa が有効でない IE で取得されたスクリーンキャプチャを、C&C サーバーの WebUI から閲覧した様子です。



図 3-2 スクリーンキャプチャ

SpyEye はスクリーンキャプチャを取得する発動条件として、マウスクリックイベントを利用しています。つまり、ブラウザがクリックされた時を起点としてスクリーンキャプチャを取得し始めます。Limosa はイベントを横取りすることによってスクリーンキャプチャが取得されることを防いでいます。

Limosa には、上記以外にもブラウザを保護するための複数のロジックが実装されています。



4. まとめ

SpyEye ビルダーで生成した検体を解析した結果、SpyEye の感染活動を把握することができました。Limosa による検証の結果、SpyEye が標準で行うことのできる攻撃からブラウザを防御可能であることが確認できました。そのため、今回 SpyEye に関して検証した範囲では、Limosa を導入することで安全にオンラインバンキングを利用できると考えられます。