ゴール分析を用いたマルウェアの機能整理と一考察

東　結香†　　猪俣 敦夫†　　藤川 和利†

†奈良先端科学技術大学院大学
〒630-0192 生駒市高山町 8916-5

yuka-h@is.naist.jp, {atsuo, fujikawa}@itc.naist.jp

あらまし 本論文では、攻撃者の要求を分析することにより、特定の目的達成のために活動を行うマルウェアの持つ機能整理を試みた。具体的には、攻撃者による視点に着目し、攻撃者自身の目的から分析するため、ソフトウェアとしてのマルウェアが持つべき機能要件を限定できる可能性を示した。また、Zbot等の多量の亜種が存在している現状であるが、様々な機能追加を施されている検体に関しては、攻撃者にとっても大量生産が可能等で扱いやすいマルウェアあり、今後も機能の追加が検討されている可能性が高い。このようなマルウェアに対して提案する手法を適用することにより、攻撃者の目的に合致する機能のうち未実装のものが新たに追加される機能である見当をつけることが可能である。

study of malware's function is organized using goal-oriented requirements analysis method

Yuka Higashi†　Atsuo Inomata†　Kazutoshi Fujikawa†

†Nara Institute of Science and Technology
8916-5 Takayama, Ikoma-cho, Nara 630-0192, JAPAN
yuka-h@is.naist.jp, atsuo@itc.naist.jp, fujikawa @itc.naist.jp

Abstract This paper discusses sort function of malware using Requirement analysis. Malware is kind of software, so it is very difficult to define malicious. I tried to sort malware's function by analyzing attacker's purpose. We were able to confirm the possibility of limiting the functional requirements as malicious software. In addition, Some types of malware, it may be possible to predict the features existing technologies that are added.

1　はじめに

ここ数年、多種多様のマルウェアにより様々なセキュリティインシデントが引き起こされている。これらのマルウェアに感染した場合、発見や完全な駆除が困難であるため、可能な限り感染自体を食い止める必要がある。また、攻撃者は目的遂行のため、様々なソフトウェアのセキュリティホールや

情報科学技術を駆使しセキュリティ対策製品へ対抗しており、攻撃者による攻撃を検知してから対策を進めていては、被害の拡大をおさえることは困難である。マルウェアは文字通り悪意のあるソフトウェアであるが、機能や影響は個々の検体の実装による部分が大きい。また、その機能の一部は正規のソフトウェアでも使用されるため、振る舞いによる検知では多くの過検知が生じてしまうと考えられる。先手を打つ策略をとったとしても、情報が分散しており、すべての可能性に対処するにはコスト面の考慮も必要になる。新たな機能実装や進化があって初めて対応が始まるのが現実である。

そこで、本論文では、攻撃者の目的よりマルウェアの機能をトップダウンで分解し、現在実装されている機能の整理を行うことにより、他の発生系統のマルウェア感で起きる可能性があると考える。具体的には、ソフトウェア開発と同様に攻撃者の要求や目的システムの実装レベルまで、ゴール分析を用い分解・整理を試みる。そして、ユースケースとして分解した結果を Zbot ファミリーの持つ機能を用い考察を与える。

本論文の構成は以下の通りである。2 章では関連研究として、マルウェアの機能を整理する際に用いられるフレームワークに関する研究についてまとめる。3 章にて、ゴール分析手法の KAOS 法を用いたマルウェアの機能整理について提案を行う。4 章では、本提案を Zbot に当てはめてスケースを提示すとともに考察を与え、5 章にて本論文のまとめと今後の展望について述べる。

2 関連研究

本章では、マルウェアの機能や特徴を整理する手法に関する関連研究について述べる。

2.1 MACE

MACE(Malware Attribute Enumeration and Characterization)[1]はマルウェアの属性や機能着目した表現形式を制定しており、MITRE のプロジェクトの1種である。MACE では大きく3層に分割して記載される。

1. High-Level

本階層では、マルウェアの持つ機能を抽象的に定義が要素そして用いられる。具体的には、【持続性】や【自己防御機能】といったものが利用されている。この抽象的な機能を次の中間層及び最下層で具体化していく。

2. Medium-Level

High-Level と Low-Level の間をつなぐ層である。High-Level で定義された機能がどのように動作するのかという部分の定義を担っている。High-Level での定義が持続性であれば、どのように持続性を持たせているか、たとえば悪意あるバイナリのインスタンス化といった記述が入る。

3. Low-Level

本階層では、マルウェア自体が何をするのかに焦点を当てている。例えば、レジストリ操作やファイル生成といったものが要素として用意されている。現状 API ベースで定義されるものが多いが、概念としてはマルウェアがコンピュータに与える影響（変更や削除、アクセス）があれば、幅広い記述を認める。

2.2 OpenIOC

米国の MANDIANT 社によって作成された[2]である。Ediot や Reader が要されており、解析者等は容易にプロファイルを作成することが可能である。MACE との大きな違いは非常にシンプルな構造であることである。要素の内容自体は MACE の Low-Level と同等の粒度での記載である。基本的に階層構造は AND または OR の識別子のみであり、どの階層でもどの要素を配置することが可能である。OpenIOC は共通形式の制定とともに、プロファイルをセキュリティ対策製品に読み込ませることにより、同一のマルウェアを見つけるという部分にも目的が置かれていると考えられる。

2.3 Malware Ontology

オントロジーとは、ある対象の世界を、一貫性をもったモデル化とするために、共通概念や規約を提供するものである[3]。オントロジーで重要視さ

れているものは概念(要素)間の関係である。通常の階層化構造の場合、上下の要素間の関係性は包含関係程度である。オントロジーでは is-a 関係をはじめとして複数の関係が適宜されている。また、要素に対する制約もかけることが可能であるため、現実を忠実に明示できるとされており、様々なサイバーインシデントへの応用が検討されている。Swimmer[5]は、マルウェアの分類に特化したオントロジーを構築した。彼は機能に着目したオントロジーと、マルウェアの命名や定義に着目した 2 種のオントロジーの構築について報告した。この用語は一般的に解析者をはじめとした情報セキュリティに携わる関係者では使われていたものの、一貫した定義や関係性の中では使用されていなかったと述べられている。

上述した関連研究より、フレームワークや表現形式は用語や定義の統一が主な目標であると考える。解析者や企業内のセキュリティ対策部門(社内 CERT 等)間で情報共有を円滑に行うことが可能となると考えられているためである。用語や定義の統一だけでなく、マルウェア開発者の視点より、要求要件を分析することによって、攻撃に対して先手を打てる可能性があると考える。

3 要求分析による手法

本稿では、攻撃者の視点で要求分析を行い、目標達成に必要な機能を洗い出すことを目的とする。洗い出した機能に対して具体的なコードや手法を埋めていくことにより、今後どのような手法が考えられるのかといった検討や、マルウェアとしての系統は異なるが、同一の目的で配布されているマルウェアに対する対策の検討が可能となると考えられる。一方、要求分析の手法は複数存在するが、本稿では KAOS 法[7]を適用する。KAOS 法は、他の手法と異なり、ある一定の表記規則にならい記述すると、数学的に検証が可能である[7][6]。また、ある状態に遷移する際の責任範囲によるモデルの分割が可能であるため、状況に応じて必要なモデルのみを使用することが可能である。

3.1 KAOS 法

ゴールから必要な要件を分析する手法の 1 種である KAOS 法を使用する[7]。KAOS 法では以下に述べる 4 つのモデルが一般に利用される。必要な用語を Tab.1 に示す。また、モデルの概念を Fig.1 に示す。

Table 1:KAOS の用語

Goal	目的を表す要素 この時点では、抽象的な文言でもよい
Entities	独立した受動的オブジェクトであり、それ自身は操作できない
agents	独立した能動的なオブジェクトであり、操作できるもの
associations	関係性やエンティティの状態変異を定義することが可能
Concern	上位 Object の要求を満たすための要求をリンクするための関係

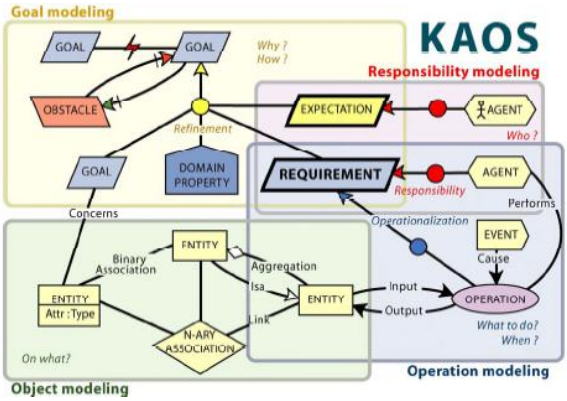

Figure 1:KAOS の概念図

① ゴールモデル

目標を達成するために、相互に関連する事象を系統的に表現しているモデルである。要求パターンという有向グラフを各要素に対して構築する。基本的には、抽象的な目標を達成するために必要な要素を記述する。各要素の責務エージェントまで分解できたところでゴールモデルは完成する。Fig.1 に示された平行四辺形が各目標であり、太枠の平行四辺形のものは責務エージェントが定義できたものである。そして、ゴールモデルの末

端の要素、即ち最も分解された Requirement である。

② 責務モデル

どのエージェントがどのゴールに責務を持っているかを表現したモデルである。本モデルは構築するというよりも、ゴールモデルを構築することに付随して生成されるものである。責務モデルは責務を持っているエージェントに焦点を当て、エージェントがどの目標に対して責務を持っているかを明確にする。

③ オブジェクトモデル

ゴール分析で定義されたRequirementは概念的なものが多い状態である。そこで、オブジェクトモデルではより具体的に分析を行う。分析を行うに当たり、上位の分析によって生成したオブジェクトには、受動能動やオブジェクト間の関連性が存在する。それらの関係性を踏まえて分析、記述される。最終的に個数関係(1 対 N)や包含関係も記載することが可能である。

④ 操作モデル

Agent が要素を満たすために必要な全ての行動、即ち Agent によって実行される操作について説明するためのモデルである。Operatorとしては、Input や Output、それによって引き起こされるイベントを定義している。このモデルによって機能の具現化が完了する。

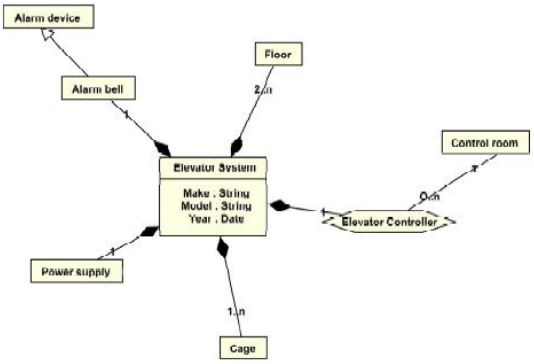

Figure 2 :オブジェクトモデルの例

今回、情報搾取を目的としたマルウェアを対象に、ゴールモデルを作成し、一部オブジェクトモデルや操作モデルに関しても言及する。責務モデルに関しては、作成する対象外とする。

3.2 マルウェアに対するゴールモデル

本論文では、情報搾取に関するマルウェアに対してゴールモデルの構築を行う。現状、情報搾取を行うマルウェアに実装または求められている機能がまとめられたリソースは存在しない。また、その挙動をどの程度細分化するかの検討がなされていないため、ゴールモデルの構築のみ行うこととする。情報搾取を行うためには以下の 2 つの概念が存在すると考えられる。

① 情報収集

情報収集という目的を達成するということは、長期間ユーザや防御サイドに見つからずに、有用な情報が取得することが重要だと考えられる。長期間情報を収集し続けるためには、ユーザの挙動に関わらず稼働していることが求められる。その目的を達成するためには自動実行のロジックが必要となる。また、ユーザや防御サイドに見つからずに稼働し続けるためには、耐ウイルス対策ソフトの仕組みや解析自体を困難とする仕組みが必要となる。本稿では、マルウェアの持つ一般的な防御機能として以下の 5 つの方法をあげる。

1. 耐ウイルス対策ソフト
2. 耐 VMware
3. 暗号化
4. 難読化
5. インジェクション

有用な情報を取得するという目的を達成するためには、可能な限り特権に近い権限であり、何らかの仕組みなしには見ることができない通信(ブラウザで処理される通信等)などの情報取得が望ましい。

② 情報送出

情報を攻撃者に送出する際には気づかれないように送出することが重要である。送出方法として、物理デバイスとネットワーク環境が存在する。しかし、情報収集時における気づかれないという目的と比較すると優先度は落ちる。ある程度必要なデータを収集したうえで攻撃者に送出するため、ユ

ーザが気付いたとしても攻撃者の目的達成には
あまり影響はないと言える。

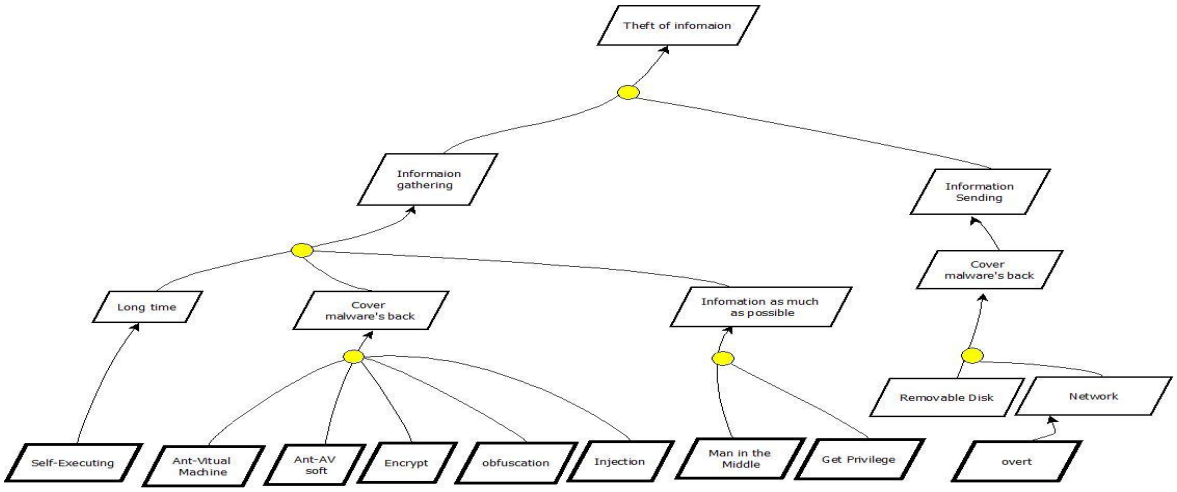

Figure 3:情報搾取を行うマルウェアのゴールモデル

4　ユースケースと考察

ユースケースでは Zbot を取り扱う。Zbot は一部ソースコードが公開状態となっているマルウェアであり、多くのサイバーインシデントを引き起こしているマルウェアファミリーである。活動自体は 2005 年頃から確認されており、日本でも近年大きな被害を出している。当時、オンライン銀行や SNS、電子メールのアカウント情報を収集する目的で設計されていたといわれている。2007 年-2013 年頃には Banking Trojan という命名から推定されるように、銀行のアカウント取得に特化した活動が確認されている。2011 年にソースコードが公開されているため [8]、攻撃者の使用したい機能を追加しやすいマルウェアだと考えられる。

4.1　Zbot

Zbot は長期間にわたり、様々な進化を遂げてきている。その亜種は、citadel[9]や gameover[10]として知られた有名な亜種のみならず、数多くの亜種が確認されている[11]。本稿では、主に 1.x 系と 2 系の間での変化と Zbot ファミリーでの大きな変化について公開情報をもとに情報の整理を行った。今回、分析するにあたり、各セキュリティベンダレポート等[12-15]を主に用いた。

Zbot は当初は他のボットとは変わらない機能を持つマルウェアであったが、様々な改良が加えられた。2012 年後半より日本でも頻繁に発見されているものは、Zbot 本体のみでは全容を解明することが困難なものが多い。Zbot は動作する際、以下 3 つのファイルが必要となる。

1．Zbot 本体の実行ファイル
2．Config ファイル
　Botnet へ加入するために必要な情報や、マルウェア自身のアップデートのための情報、攻撃対象の銀行の URL 等が含まれる
3．Drop Server 情報
　収集した情報を送付するサーバ群

Zbot の持つ一般的な機能を以下にあげる。
・自己複製機能 / 自己消去機能
・レジストリ変更
・Internet Explore の設定変更
・他のプロセスへのインジェクション
・Config ファイルのダウンロード及び実行
・API フックによる情報取得
・C&C への情報送付

4.2 Zbot における特異点

情報搾取を目的としたマルウェアという観点で Zbot 進化を分析する。はじめに、Zbot における特異点を時系列にて整理をしたものを Tab.2 に示す。また、コードの系統的に大きな変化があった 1.x 系～2 系では、機能面でも非常に大きな変化があった。こちらに関しては主にフックしている WindowsAPI について引用している[17]。

Table 2:Zbot の進化

年月	特徴
2007 年 9 月頃	Config によって動作をするものが確認される[18]
2009 年	Man-in-the-browser を用いた挙動が公開される[19]
2010 年初頭	Zbot の改良版(2.0 系)が確認される[14][15][17] ・Firefox における Man in the Brower ・スクリーンショットやキーストロークの取得 ・Config ファイルの暗号化手法（より複雑に）
2011 年	64bit 版の Zbot を確認[20]
2012 年初頭	Citadel を確認[10l] 暗号化に AES の使用を確認
2012 年初頭	P2P の使用を確認 Zbot Gameover[21]
2013 年	Tor を用いたものを確認[20]
2013 年秋	ランサムウェアの感染に利用される[22]

4.3 ユースケースの適用

今回のゴールモデルは情報搾取を目的として構築している。はじめに、情報収集について適用する。情報収集を達成するための目的として前述の 3 要素について検討する。

Table 3目的に対する Zbot の実装

目的	Zbot での実装
長期的な実行	自動実行機能 レジストリ変更によって実現
気づかれずに実行	自己消去及び複製機能 他のプロセスへのインジェクション アクセス先等の情報の暗号化（Config ファイル） 本体コードの難読化
多くの情報を取得	特権の取得 該当なし 中間（アプリケーション等）での取得 API フックにより実現 －ブラウザ －各種クライアント －キーストロークやスクリーンショット

4.4 考察

本論文では、ゴールモデルのみを構築し、実際のマルウェアの変化に当てはめて以下に考察する。Zbot の持つ基本的な機能と各ゴールの比較をしたところ、ほぼ分析した要求を満たすソフトウェアであることが判明した。しかし、Zbot の特徴の 1 つである Config ファイルによる感染の進行に関しては検討できない状況であった。これは、スタートが情報搾取というゴールから分析を行ったことが理由である。感染ルーチンにも各マルウェアの特徴が表れるもの存在するため、根本のゴールを 1 段抽象的なものを検討すべきである。

また、具体的な手法をオブシェクとモデル及び操作モデルで記述することにより、具現化が可能である。例えば、レジストリ変更についてもオブジェクトとしてレジストリを定義することにより、それぞれの目的に対して、どのようなレジストリ変更が生じるかを整理できると考える。同一の目的を持つ異なる系統のマルウェアで取られた手法を効率よく他の系統のマルウェアの対策に利用できる可能

性があると考えられる。

一方、ネットワーク周りとしては、TCP/IP→P2P⇒Tor と変化をとげている。また、Zbot2 系のソースコードには IPv4 および IPv6 デュアルスタックの記載が存在する。これらの変化は決して予想できないものではないと考える。P2P についてはここ数年、Tor は特に近年、犯罪に利用されるネットワークとして世間を賑わせたネットワークである。また、IPv4 および IPv6 についても毎年枯渇による移行が進むという話も出てきている。犯罪に利用する際は、できるだけ送信先と犯罪者本人と結びつかない通信路を選択するという考えを前提とするならば、マルウェアの通信環境として利用することは容易に想定できる。

次に、Zbot2 系で取り入れられた Firefox に対する Man in the Browser も想定の範囲だと考えられる。元々、Zbot1 系でも IE を対象とした MITB 攻撃は実装されており、その後の展開として IE の次にシェアの高い Firefox を対象とすることは想定の範囲内だと言える。

また、Config 暗号化に関しては、Zbot1 系から 2 系へ変化する際にロジックが複雑に変化していた。これについては、予測が非常に難しいと判断される。例をあげるならば、RC4 の使用が確認された時点で、RC4 より強力な暗号化が使用されることは予測可能であるが、どのようなロジックでどのような暗号化方式を使用するかを予期することは難しい、という理由からである。

4.5 今後の課題

ゴールモデルの各要素の粒度を揃える必要がある。今回の分析では、【気づかれずに実行】という要素の下位要素の粒度が細かすぎるということが判明した。実際に、一般に使用されている用語や MACE の用語を用いてモデルの構築を行った。そこで、中間の要素を設けることにより、何を目的とした機能であるかをより正確に分析可能であると考えられる。なお、昨秋に Zbot をランサムウェアのトリガーとして利用する検体が確認されている。ゴール分析の根本であるゴールが変わる可能性を考慮したモデル（多層化など）を検討すべきだと考える。

はじめに述べたとおり、マルウェアはソフトウェアの 1 つでもあり、実装次第でどのような形態にもなりうる。要求分析によるマルウェアの整理は、攻撃者側に目的（金銭目的、社会活動等）があれば、その目的より実際のコンピュータの挙動に落とし込むことが可能であるが、目的が見えないものの場合対応ができない。

また、要求分析を用いたマルウェアの必要要件を洗い出すコストについても無視できないと考えられる。Zbot のような発生系統的にグルーピングされているマルウェアをモデルとして、進化の過程学習した上で、その他の発生統計的に少数派のマルウェアに適応していくことにより、比較的少ないコストで網羅性を持った整理ができると考えられる。

5 総括

本稿では、ゴール分析を用いたマルウェアの機能整理について、Zbot を例にあげて考察を与えた。攻撃者の目的を分析することにより、未だ課題は存在するものの、マルウェアに必要な機能については検討することができた。この整理をオブジェクトモデル及び操作モデルまで作成し、他の目的（成りすましなど）に対しても同様の整理を行うことにより、特定の目的を持つマルウェアの要件を定義できると考えられる。

今後の課題として、各要素の粒度の統一と扱った情報搾取を目的とする Zbot に対するオブジェクトモデル及び操作モデルをもとに 1 モデル作成のコストや規模感を図る。昨今、マルウェアの API シーケンスやコールブロック、オペコードをもとにマイニングを行い分類するという研究が進められている。ソフトウェアとしての系統は比較的高い精度で分類可能であるが、ノイズとなる情報が多いため機能に着目した分類は困難であった。今後、目的別で必要な要件を操作モデルまで落とし込んだ結果と系統分類したものを組み合わせ、マイニングで得たものに機能的な意味合いを持たせることができる可能性があると考える。

参考文献

[1] DESIREE BECK et al, THE MAEC LANGUAGE OVERVIEW,2014 http://maec.mitre.org/about/docs/MAEC_Overview.pdf

[2] Mandiant, OpenIOC, http://www.openioc.org/

[3] 溝口理一郎、他オントロジー工学基礎論－意味リンク，クラス，関係，ロールのオントロジー的意味論－ 人工知能学会誌, Vol.14, pp.1019-1032, 1999

[4] OBRST, Leo; CHASE, Penny; MARKELOFF, Richard. Developing an Ontology of the Cyber Security Domain. In: STIDS. 2012. p. 49-56.

[5] Swimmer, M. Towards An Ontology of Malware Classes. January 27, 2008. http://www.scribd.com/doc/24058261/Towards-an-Ontology-of-Malware-Classes.

[6] 山口奈津子，要求工学手法の知識要素の分析,2009 , http://www.seto.nanzan-u.ac.jp/ise/gr-thesis/it/proc/2009/04mt120.pdf

[7]Objectiver ： "A KAOS Tutorial, 2007 http://www.objectiver.com/fileadmin/download/documents/KaosTutorial.pdf

[8]Visgean/Zeus,2011 、 https://github.com/Visgean/Zeus

[9]Citadel Trojan Malware Analysis, Jason Milletary ,2012 http://botnetlegalnotice.com/citadel/files/Patel_Decl_Ex20.pdf

[10] ANDRIESSE, Dennis, et al. Highly resilient peer-to-peer botnets are here: An analysis of Gameover Zeus. In: *Malicious and Unwanted Software:" The Americas" (MALWARE), 2013 8th International Conference on*. IEEE, 2013. p. 116-123.

[11] S21sec、ZeuS timeline http://securityblog.s21sec.com/

[12] Aditya Balapure, Botnets Unearthed – The ZEUS BOT, 2013, http://resources.infosecinstitute.com/botnets-unearthed-the-zeus-bot/

[13] McAfee, McAfee Labs Threat Advisory PWS-Zbot,2014,https://kc.mcafee.com/resources/sites/MCAFEE/content/live/PRODUCT_DOCUMENTATION/23000/PD23030/en_US/McAfee_Labs_Threat_Advisory_PWS-ZBot.pdf

[14] File-Patching ZBOT Variants ZeuS 2.0 Levels Up, TrendLabs, 2010, http//www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp__file-partching-zbot-varians-zeus-2-9.pdf

[15] FILES, Financial Malware. *ONLINE BANKING FRAUD MITIGATION*. PhD Thesis. Delft University of Technology.

[16] Nicolas Falliere et al. , Zeus: King of the Bots , 2009, http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/zeus_king_of_bots.pdf

[17] Karthik Selvaraj , A Brief Look at Zeus/Zbot 2.0, 2010, http://www.symantec.com/connect/ja/blogs/brief-look-zeuszbot-20

[18] Alexander Gostev, Vitaly Kamluk , Malicious code evolution: July - September , 2007,http://securelist.com/analysis/quarterly-malware-reports/36178/malicious-code-evolution-july-september-2007/

[19]SonicWALL Security Center, Zeus Trojan Family, 2009, https://www.mysonicwall.com/sonicalert/searchresults.aspx?ev=article&id=132

[20]P.Paganini, Detected 64-bit ZeuS banking trojan using Tor network, 2013, http://securityaffairs.co/wordpress/20409/cyber-crime/zeus-banking-trojan-64bit.html

[21] 010 - Crime - GameOver Zeus (with P2P and DGA) –trojan,2012 ,http://contagioexchange.blogspot.jp/2012/03/010-crime-gameover-zeus-with-p2p-and.html

[22] K.Alintanahi,ランサムウェア Crypto Locker、オンライン銀行詐欺ツール「ZBOT」を経てコンピュータに侵入 , http://blog.trendmicro.co.jp/archives/8017