マルウェア作成者の知識・技能を主眼としたインテリジェンスの収集法の提案

村上 弘和 [1,2,3*]　西垣 正勝 [1]

概要：サイバー犯罪の調査に不可欠なインテリジェンス情報は，サイバー攻撃に関連するあらゆる痕跡がソースになる．マルウェア自体も，サイバー攻撃者が残した重要な痕跡の一つである．インテリジェンス情報の収集を目的としたディープマルウェア解析のうち，マルウェア作成者の特徴に焦点を置いた情報収集法を確立することは本研究の 1 つの重要な目的である．マルウェア作成者の特徴のうち，知識・技能に焦点を置いた情報の収集法を提案する．具体的な例を用いてマルウェア作成者の知識・技能に関する情報がどのようなものかを示す．本稿で述べる抽出法を利用することにより，マルウェア作成者の能力およびそれに基づいたマルウェア作成者の特徴情報を収集することができる．この方法によって，マルウェア作成者の持つ能力や，将来を含めたサイバー攻撃の意図および手法を推察するための材料となる情報を得ることができる．また，複数のマルウェアに対しこの方法で抽出した情報を比較分析することにより，マルウェア作成者の同一性，非同一性を推察する情報として利用でき，サイバー攻撃者に関する一つのインテリジェンスになると考えられる．

キーワード：ディープマルウェア解析，脅威インテリジェンス

The method for collecting knowledge and skill of malware authors

Hirokazu Murakami[1,2,3,*]　Masakatsu Nishigaki [1]

Abstract: All evidence related to cyberattacks is a source of intelligence that is essential for cybercrime investigations. Malware is one of the most important evidence left by cyber attackers. One of the key objectives of this research is to establish information collecting methods, within the context of deep malware analysis for intelligence collecting purposes, that focus on identifying characteristics of malware creators. Among the features of malware authors, we propose the information collecting method that focuses on knowledge and skills. To demonstrate what kind of information pertains to the knowledge and skills of malware authors, it is presented specific examples. By using the extraction methods described in this paper, it is possible to collect information on the capabilities of the malware authors and their characteristic information based on that. This method provides information that can be used to infer the capabilities of malware authors and the intentions and methods of cyber attacks, including future ones. By analyzing some malware with this method, it is possible to show the identity or non-identity of malware authors and it is considered to be one of the intelligence about cyber attackers.

Keywords: Deep malware analysis, Threat intelligence

1. 序　論

マルウェアはサイバー攻撃で使用される手段の一つであり，マルウェアにもサイバー攻撃者に関するインテリジェンスが含まれる. このインテリジェンスには, 少なくとも，マルウェア使用者とマルウェア作成者の情報が含まれると考えられる．サイバー攻撃者のインテリジェンス収集のためのディープマルウェア解析の理論については，SCIS2022 で既に発表済みである[1]．マルウェア解析で見つけることができる情報と出現が予想されるマルウェア作成者・攻撃者像の関係およびレイヤーのイメージを図 1 に示す．この理論が実用できることを示すためには，ディープマルウェア解析法の具体化が必要であると考える．図 1 に示した内容のうち，マルウェア作成者のコーディング癖(図 1(d))の収集法については CSS2022 で既に発表済みである[2][3]．本論文では，マルウェア作成者の知識・技能に関するインテリジェンスの収集法を詳述する(図 1(c))．マルウェア作成者の知識･技能のインテリジェンスが必要な理由として，そもそもマルウェアによるサイバー攻撃は，攻撃者にその意図があり, かつ能力がある場合においてのみ発生し得る，ということが挙げられる．もし意図が無ければサイバー攻撃という行動自体を起こさず，能力が無ければ実現していないか，痕跡を残す前に失敗してしまう．マルウェア作成者の知識・技能の分析は，サイバー攻撃者の意図および能力のうち, 能力に主たる焦点を当てた分析であるといえる．マルウェア使用者の思想・目的（使用者の意図）(図 1(a))，マルウェア作成に関する設計・技術的ポリシー（作成者の意図）(図 1(b))はマルウェア攻撃者の意図にあたり，これら

1 静岡大学　創造科学技術大学院
Graduate School of Science and Technology, Shizuoka University
2 東京電機大学　サイバーセキュリティ研究所
Cyber Security Lab, Tokyo Denki University
3 株式会社 CyCraft Japan
CyCraft Japan Corporation.
* murakami.hirokazu.22@shizuoka.ac.jp

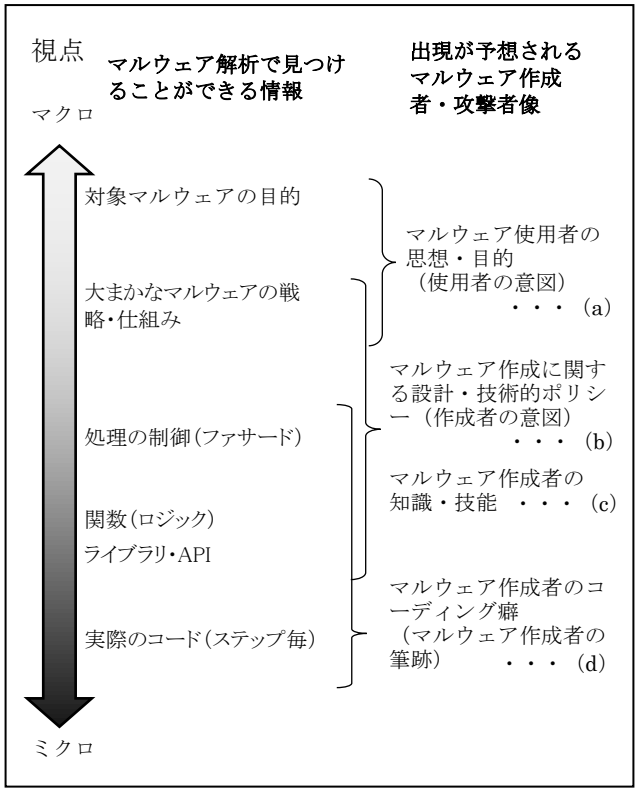

図 2　マルウェア使用者及び作成者の分析のレイヤーのイメージ

も含めて考察することで，そのマルウェアに関するサイバー攻撃の脅威をより正しく把握することができるようになると考えられる．このインテリジェンスは，アクティブ・サイバー・ディフェンスなどのサイバー攻撃の対策や考察に利用できることが予想される．マルウェア作成者の知識・技能がマルウェアに含まれる理由として，以下の理由が考えられる．

- 仮に実行結果が同じプログラムでも，実装コードが違うということが散見される．これは，単純な好みだけではなく，処理効率やマルウェア検知対策など，様々な技術的理由により，コードに細工をした結果である場合があると考えられる．(図 1(c))
- マルウェアに実装されているという事実は，マルウェア作成者にその技術に対する知識を有していることの証拠である．また，実装するだけの技術力があるということの証拠でもある．
- 仮に知識があったとしても，実装する技術力がなければ，マルウェアの設計に含むことはできず，マルウェアにそのコードが表れることも無い．また，知識が無ければ，そもそも設計に含まれることは無い(図 1(b))．
- 知識・技能が不十分であれば，高度な実装方法はできず，仮にマルウェアとして検知がされやすいとしても，マルウェア作成者はその方法しか選択できない．一方，知識・技能が高ければ，より効率的かつマルウェアとして検知され辛い洗練されたコーディングが出現すると考えられる．

図 1 に示されているように，マルウェア作成者の知識・技能は，「大まかなマルウェアの戦略・仕組み」から「関数（ロジック）」のインテリジェンスに該当する．マルウェア作成者の知識・技能は，マルウェアで使用する技術や実装の可否に影響を及ぼすため，マルウェア作成に関する設計・技術的ポリシーに影響を与える．また，知識・技能はマルウェアの実装内容にも影響を及ぼすため，マルウェア作成者のコーディング癖にも影響を与える．

マルウェア作成者の知識・技能は，マルウェアのコーディング癖と同様に，マルウェア作成者の特徴の表れである．このため，複数のマルウェアからマルウェア作成者の知識・技能のインテリジェンスを抽出し，比較することによって，コーディング癖ほど明確ではないものの，マルウェア作成者の同一性を判定する一要素になるのではないかと考える．これにより，作成者を視点としたマルウェア分類法の確立，マルウェア作成者が所属する組織の同定など，新たなサイバーセキュリティインテリジェンスの醸成が期待される．

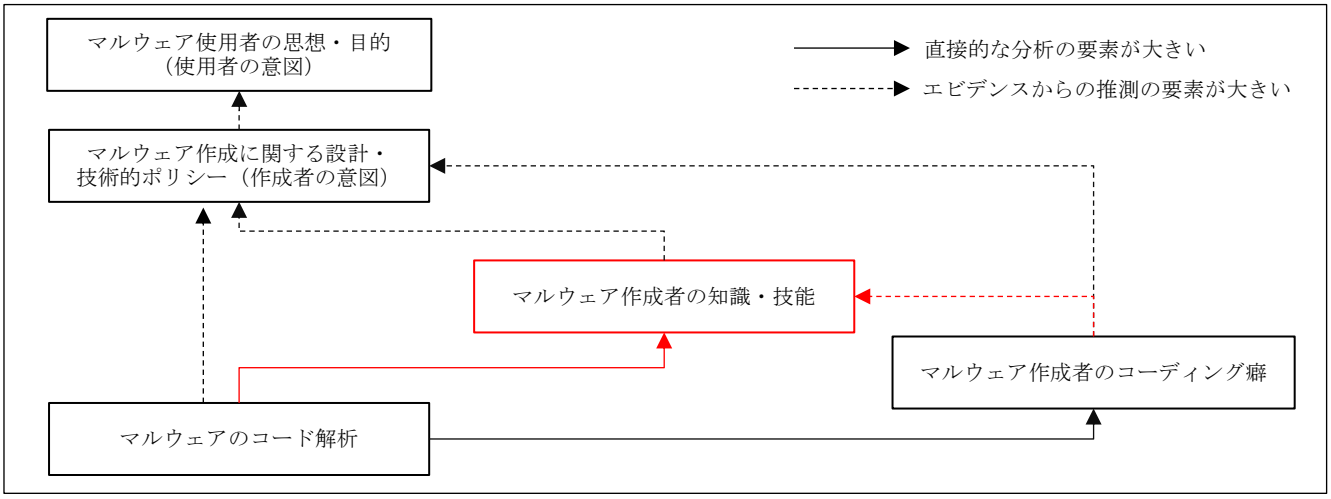

図 1　マルウェア作成者・使用者のインテリジェンスの分析の関係性

特に，マルウェアの知識・技能は，今後取りうる攻撃の能力を示す指標の情報源になり得ると考える．

マルウェア作成者の知識・技能を抽出するためには，マルウェア作成のためにどのような技法を使ったか，という観点で分析することが重要である．これを実現するためには，解析された関数や機能から技術的な特徴を洗い出し，それを利用した背景を考察する． また，採用した技術や実装方法から，どのような知識があるかを考察する．この分析をするための情報源として，マルウェアそのものの解析結果のほか，マルウェアのコーディング癖の分析結果も有用な情報源として利用できる．これは，マルウェア作成者の知識・技能の分析は，以前提案したマルウェア作成者のコーディング癖の分析が終わった後，その情報も含めて行うことが有用であることを示している．解析の対象と情報源の関係を図 2 に示す．本論文では，実際のマルウェアを用いて，マルウェア作成者の知識・技能を抽出する方法を試行し，マルウェアのコードやコーディング癖の分析から目的の情報を抽出できるかを調査，検討する．本論文の分析を通じ，マルウェア作成者の知識・技能の分析が可能であり，有用なインテリジェンスになり得ることを確認する．

2. マルウェア作成者の知識・技能の分析のための基本的な原理

マルウェア作成者の知識・技能の分析をするための基本的な考え方を示す．マルウェアは，設計やポリシーを元にコーディングされることが考えられる．この実装方法の優劣の判定や実装理由などを技術的な観点で考察することにより，作成者の知識・技能を推定することが可能だと考えられる．現在では，世の中に存在するプログラムに関する全ての知識を習得することは難しく，必ず知識の「偏り」が生じる．そのような偏りから，作成者がどういった分野が得意で，どういった分野が不得手なのかを推定することが可能と考えられる．仮に AI によるマルウェア作成が行われたとしても，学習対象の選択やチューニングといった「偏り」による差が生じると予想され，プロファイルすることは可能だと予想される．マルウェアには通常のアプリケーションのプログラムには存在しない「検知・解析回避」のための技術が使われることが多いという特徴があり，最近ではそれが大きなウェイトを占めているという傾向がある．このことから，「調査を必要とするレベルのマルウェア」が発見されたこと自体が，作成者の「知識・技能」を示す一つの指標になる．「知識・技能」を利用してマルウェア作成を実現できたという事実のほか，それが容易に発見できるものだったか，もしくは非常に発見が困難だったかといった発見理由の考察も，「知識・技能」を推し量るための一つの要素であると考える．

マルウェア作成者の知識・技能の分析では，「どのようにして」実装し，機能を実現したかについて焦点を当てることで可能になる．実際のコード，あるいはコードの意味の理解から，何を目的とした機能かを合理的に判定する．そして，その実装の技術的な正確性や難易度から能力を推測し，機能の内容からどのような知識があるかを推測する．

知識・技能の分析に用いる情報源の質は，図 1 に示した通り，マルウェアの分析結果のうち大まかな戦略や仕組みから，その実現のための制御（ファサード），個別の機能（ロジック）にあたるレイヤーの情報に注視する．これは，マルウェアを俯瞰的に分析してどのような技術が使われているかを把握しつつ，その根拠や証拠として下層の情報を用い，確実性を高めていく方法になる．一方，些末なコードの違いは，知識・技能に関わらないことも多く，知識・技能の分析においては，あまり注視する必要はないと考える．例えば，マルウェアから暗号化処理が見つかったとする．この処理を分析してその暗号に関してマルウェア作成者の知識・技能を分析する場合，暗号計算をしているロジックを分析して暗号の種類を調べる[4]．この情報から，マルウェア作成者がどの程度暗号の知識があるのかを推測することになる．また，ファサードを分析して実装の妥当性や実装に必要な知識を推測することになる．この中で，例えば使用した暗号鍵の情報をゼロクリアしている処理があった場合，「暗号鍵を漏洩しないようにする知識がある」という情報が得られる．このときのゼロクリアの方法も様々あり，コーディング癖としては特徴となるが，暗号処理の知識および実装の能力の分析においては，ゼロクリアでいずれの方法を用いても，暗号の知識・技能の評価は変わらない．

知識・技能の分析のデータの情報源は，図 2 に示した通りである．マルウェアのコード解析結果から直接情報を得られるほか，マルウェア作成者のコーディング癖の分析結果から推測のための情報を得ることができる．情報源は，いずれかの一方から得ることも，両方の情報源から得ることもできる．マルウェア作成者のコーディング癖の分析結果を利用できる理由は，同じ結果のために複数の実装方法が考えられる場合であっても，より洗練された方法や，知識が無ければ選択や実装をし得ないような方法が見つかることがあり，それらの痕跡から知識・技能を推測することが可能と考えられるためである．また，分析された知識・技能は，マルウェア作成に関する設計・技術的ポリシーの分析に利用できる情報となる．いずれの情報源においても，技術的な観点で合理的な推測をすることが必要である．これは，優秀な技法・実装の発見により高い能力があることを推測だけでなく，稚拙，または不合理な実装，エラー判定および処理の不足，バグの発生状況から能力の低さが見て取れる場合もある，ということである．

マルウェアの解析結果から直接情報を得られる場合では，マルウェアの中で使われている技術的な特徴を抽出し，インテリジェンス情報とする．また，使っている技術からど

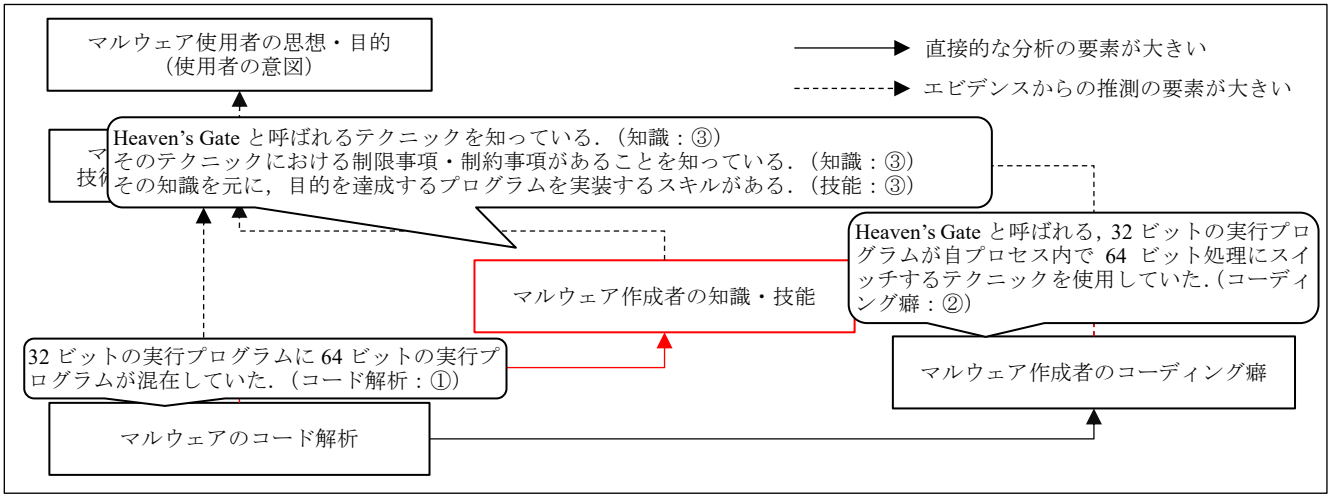

図 3 Heaven's Gate 技法の利用における知識・技能のインテリジェンスに関する関係

のような知識があるかを推測する．例えば，Windows のマルウェアで AES の CBC モードの暗号化を Windows API を使って実装していた場合，Windows API による暗号化実行を実装できる技能がある，ということが分かる．また，そこから AES 暗号についての知識がある程度以上あることが推測できる．

マルウェアのコーディング癖の解析結果から情報を得られる場合では，特に特徴量の高い実装があった場合の方法に技術力を推測する情報があった場合，その情報をインテリジェンス情報とする．また，その方法を選択したという事実から，どのような知識があるかを推測する．例えば，メモリクリアをしているが，一般的なライブラリ関数ではなく，特定の条件で軽量かつ高速で動作する方法で実装しているコーディング癖が見つかっている場合，そのような実装を選択して実装できる技能があることが分かる．また，それがソフトウェアの動作速度向上や解析回避に有用であることを知っているという知識があることを示す．一方，マルウェアのコーディング癖のエラー処理が不十分だった場合やバグは，知識・技能の低さの指標になり得る．

分析の手順は，プログラム全体のフローが判明したら，そのプログラムで用いられている具体的なテクニック，マルウェア使用者の「思想・目的」やマルウェア作成者の「設計・技術的ポリシー」の関与が低いレベルでのマルウェア作成者の知識の影響，プログラムの構造上に対する実装の妥当性を検証するという方法を取る．特に，主たる目的の機能の実装方法や利用しているテクニックは合理的か，それぞれの処理は妥当か，無駄な処理がどの程度あるか，技巧的に見て特筆すべき点はあるか，といった点に注目して分析する．マルウェア作成者に「どのような知識があるか」，「どのような技能があるか」という観点で洞察することがこの観点での分析の重要な点となる．

3. マルウェア作成者の知識・技能の判定例

実際の機械語のマルウェアを逆アセンブルしたコードを用いて，マルウェア作成者の知識・技能に着眼した分析する方法を例示する．この分析を行う前提条件として，対象のマルウェア全体を可能な限り解析しておくことが前提となる．また，マルウェアのコーディング癖も分析が完了していることが望ましい．これらの情報の欠落は，分析の正確性に大きな影響を及ぼす．本論文では，2020 年春ごろに見られた Emotet[5]として分類されていたマルウェアを対象として分析例を示す．

3.1 Heaven's Gate の利用

分析対象のマルウェアでは，Heaven's Gate[6] と呼ばれるテクニックを使用していることが確認された．これは，予約された特定のセレクタを使用してコールまたはジャンプ命令を実行することにより，32 ビットのプロセスから 64 ビットの機械語命令を実行，あるいはその逆を行うことができる方法の利用を指す．これにより，自動的な解析システムであれば，32 ビットとして解析していた場合，64 ビットのコードを正しく認識できず，解析が失敗する可能性が考えられる．また，解析者が解析している場合でも，この技法を知らなければ，解析そのものが行き詰ることが考えられる．これは，マルウェア開発者としては大きなメリットであるため，解析回避の技法として利用されたと考えられる．この Heaven's Gate の利用について焦点を当て，マルウェア作成者の知識・技能を分析する．

Heaven's Gate の利用に関するマルウェア作成者の知識・技能に着目した分析の関係を図 3 に示す．分析の情報源の 1 つはマルウェアの解析結果である．この情報から Heaven's Gate の利用の痕跡を分析する．まず，表層解析や初期の動的解析から，32 ビットアプリケーションであり，実行時も 32 ビットアプリケーションとして実行される．また，機械語命令も 32 ビットであることが確認できる．一方，難読化

解除されたメモリ内のデータには，64 ビットの機械語命令が含まれていることが確認された．また，セレクタ 0x0033 を指定し，far call 命令するコードが確認された．これらは，マルウェアそのものから得られた直接的な証拠である(①).

図 3 で示した分析のもう 1 つの情報源は，マルウェアのコーディング癖の分析結果である．マルウェアのコーディング癖では，セレクタ 0x0033 を指定し，far call 命令するコードを Heaven's Gate の利用と判定し，特徴として抽出していた．これは，マルウェアのコードに対し，コーディング癖という観点での分析による評価が加味された情報となる(②).

これらの情報を元に，マルウェア作成者の知識・技能の観点で考察することで，インテリジェンス情報の 1 つが得られる．知識面で考察すると，そもそも「Heaven's Gate と呼ばれるテクニックを知っている」という知識があることが分かる(知識 : ③). また，この方法を使うことで解析を困難にすることができ，アンチフォレンジック技術として利用が可能という知識もあった，と推測することが可能ではないかと考えられる．この知識は，マルウェアの検知を回避したいというマルウェア作成者の意図に関わる設計・ポリシーにも影響を及ぼしたということも予想される．Heaven's Gate を利用した場合，コーディングにおける制限があることも知られている．このマルウェア作成者は，この知識もあったと推測できる(知識 : ③). なぜなら，マルウェアは動作するものとして完成しており，その知識無しではこのテクニックを実装し得ない，という事実から推測可能である．この知識は，マルウェア設計における制限事項となるため，マルウェアの設計に影響を及ぼしたと考えられる．また，この制限事項はプログラムの実装に大きく影響するため，マルウェアを作成する過程でコーディング癖にも影響を及ぼしたと考えられる．このマルウェアからこのテクニックが動作する状況で発見されたことは，Heaven's Gate に関する知識をもとにプログラムを実装し得る能力があったことの証拠となる(技能 : ③). 複雑な技法を実装する能力は，マルウェアのコーディング癖に大きな影響を及ぼすと考えられる．認知度が低く，かつ複雑なコーディング能力を要する技法を実装したという事実は，マルウェア作成者の知識・技能を推測する上で非常に大きな情報になると考えられる．

3.2 関数単位の暗号化の利用

分析対象のマルウェアでは，関数単位の暗号化というテクニックを使用していることが確認された[7]．これは，関数のコードを xor により暗号化し，実行直前に復号化して実行し，関数が終了してリターンした直後に再暗号化する技法である．これにより，静的解析時だけでなく，プログラムがメモリに展開された時にもコードが暗号化されている状態となる．この場合，メモリフォレンジック技術を使っても，実行中の関数以外の悪意あるコードは暗号化されているため，全容把握や検知が困難になる．マルウェア開発者としては，メモリフォレンジックによる検知，回避のために大きなメリットがあるため，解析回避の技法として利用されたと考えられる．この関数単位の暗号化の利用について焦点を当て，マルウェア作成者の知識・技能を分析する．

関数単位の暗号化の利用に関するマルウェア作成者の知識・技能に着目した分析の関係を図 4 に示す．分析の情報源の 1 つはマルウェアの解析結果である．この情報から関数単位の暗号化の利用の痕跡を分析する．この技法でマルウェアから直接取得できる痕跡は，暗号化や復号化を制御する仲介関数のコードである．暗号化された関数が実行される場合，直接関数が呼び出されるのではなく，仲介する関数が呼び出される．仲介する関数は，最初に呼び出し元の関数を暗号化する．次に，関数の ID となるパラメータから実行される関数を識別し，関数のアドレスを取得する．さらに，xor による関数の復号化処理と関数内のアドレス調整処理を行う．これらが完了後，関数をコールする．当該関数が実行を完了して仲介する関数にリターンすると，

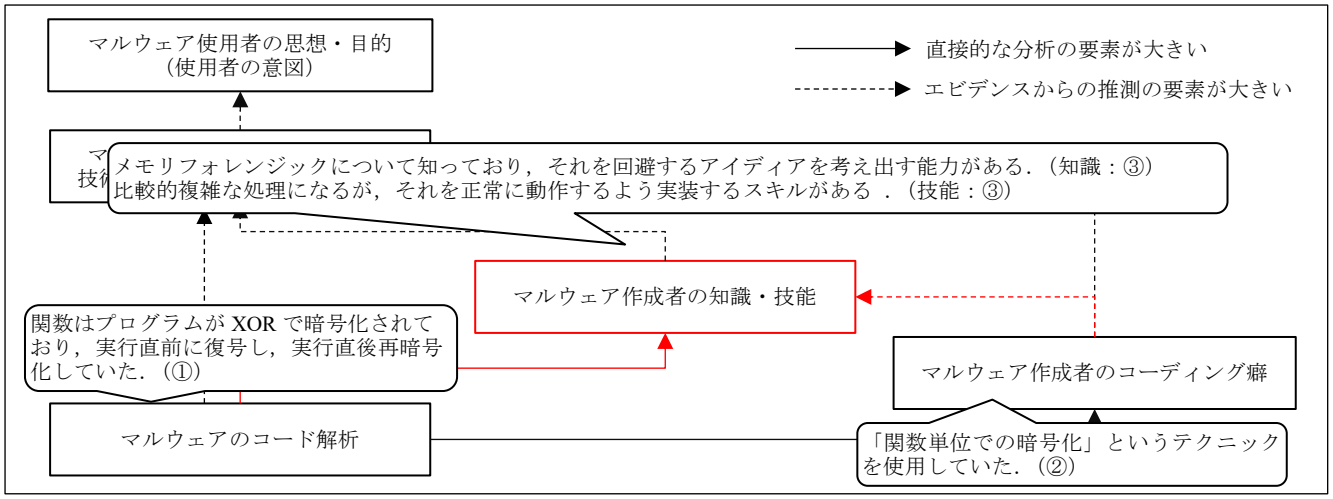

図 4 関数単位の暗号化技法の利用における知識・技能のインテリジェンスの関係

関数を xor で再暗号化する．その後，呼び出し元の関数を復号化し，呼び出し元にリターンする．この仲介関数は，マルウェアそのものから得られた直接的な証拠である(①).

図 4 で示した分析のもう 1 つの情報源は，マルウェア作成者のコーディング癖の分析結果である．マルウェアのコーディング癖では,仲介関数を用いて呼び出し元の暗号化，呼び出す関数の復号化および実行，関数のリターン後の関数の再暗号化，呼び出し元関数の復号と呼び出し元関数へのリターンまでの一連のコードについて,「関数単位の暗号化」というテクニックだと判定し，特徴として抽出していた．これは，マルウェアのコードに対し，コーディング癖という観点での分析による評価が加味された情報となる(②).

これらの情報を元に，マルウェア作成者の知識・技能のインテリジェンスを得る．知識面で考察すると，この技法を使った理由として，メモリフォレンジックについてある程度知識があり，メモリフォレンジックではどのような弱点があるかを知っており，その回避策としてこの技法を考案または流用するに至ったと考えられる(知識 : ③). また，一般的なプログラムでは，ある程度バグがあっても動いてしまうことはあるが,この技法ではこの処理が失敗すると，プログラムが即座にクラッシュしてしまう可能性が高い．このため，この複雑な処理を，より緻密かつ正確な実装の必要性がある．このプログラムが実装され，実際に動作しているという事実は，相応のプログラムのコーディングスキルがあることを示す(技能 : ③). この知識・技能分析の結果もまた，マルウェア作成者に関する能力を示す一つのインテリジェンスになると考えられる．

4. 結　論

マルウェアの解析結果やコーディング癖の分析結果を情報源とし，技術的な点に着目してその能力を分析することで，マルウェア作成者の知識・技能のインテリジェンス情報を抽出できると考える．本論文では，それぞれの情報源からの情報の抽出のための考え方と方法を示し，実際のマルウェア解析情報を利用して具体的な事例を示した．この結果から，マルウェアからマルウェア作成者の知識・技能に着眼したディープマルウェア分析情報を抽出し得ることを確認することができたと考える．この方法を実践し，さらに発展させることにより，マルウェアの作成者の能力を把握し，マルウェアの筆跡鑑定や将来的な攻撃の予測などに利用できる情報を得られると考える．サイバー攻撃者に関するインテリジェンスが増えることにより，サイバー攻撃に対する調査，対策に寄与することに期待したい．

課題として，実際にこの方法を用いて，どの程度マルウェア作成者のインテリジェンスが引き出せるかを検証する必要がある．また，本論文に示した観点以外にもマルウェア作成者の知識・技能を抽出できる観点が無いかを研究する必要がある．新たな観点が追加されれば，マルウェア作成者のインテリジェンスの質・量ともに向上すると考えられる．他の課題として，マルウェアの知識・技能の分析における考察の評価方法が挙げられる．分析者は，明確な証拠や客観的な観察の元，論理的な理由を付けて分析結果を示すことになるが，そこにはどうしても分析者の主観的価値が残りやすい．マルウェア作成者の知識・技能の場合，知識については，実装が確認された技術を根拠にする，あるいは正しくない実装を根拠にすることで，知識の有無について根拠を持って述べやすい．一方，能力については，評価にあたり分析者自身の能力と比較して判定される傾向が予想される．これは，能力が高い，低いという価値観が，根本的に何かと比較した相対的な評価であることに起因する．そのため，技術力の高い分析者の場合，能力の評価が低いとされることが多くなることが予想され，他方経験が浅い分析者の場合，能力の評価が高いとされることが多くなることが予想される．このため，より客観的なスコアリングの価値基準を定める，などといった方法を考え，分析結果が分析者によってばらつかないようにする方法を考案する必要性が考えられる．

今後の研究では，実際のマルウェアに対しこの観点を用いた分析を行い，対象のマルウェアのマルウェア作成者の知識・技能の抽出を行いたい．そして，得られたインテリジェンスをどのように利用できるかを検証したいと考える．

参考文献

[1] 村上弘和, 西垣正勝, “サイバー攻撃者のインテリジェンス収集のためのディープマルウェア解析”, 2022 年暗号と情報セキュリティシンポジウム(SCIS2022), 2022.
[2] 村上弘和, 西垣正勝, “マルウェア作成者のコーディング癖の収集法の提案”,コンピュータセキュリティシンポジウム 2022 (CSS2022), 2022.
[3] 村上弘和, 西垣正勝, “マルウェア作成者のコーディング癖の収集例：メモリのゼロクリアの観点から”,コンピュータセキュリティシンポジウム 2022 (CSS2022), 2022.
[4] J.A.ブフーマン, 林芳樹,”Einfühlung in die Kryptographie 暗号理論入門”,シュプリンガー・ジャパン, 2003.
[5] ”Emotet”, https://attack.mitre.org/software/S0367/, The MITRE Corporation, 2023.
[6] George Nicolaou, “Knockin’ on Heaven’s Gate – Dynamic Processor Mode Switching”, Reverse Code Engineering, 2012.
[7] Hirokazu Murakami, “Malware Function-based encryption technique”, SANS Institute, 2022.