サイバー情報共有イニシアティブ（J-CSIP） 運用状況 [2019 年 4 月～6 月] 《 付録 》 ～オープンソースの解析ツール「Ghidra（ギドラ）[1]」の紹介～

2019 年 7 月 26 日
IPA（独立行政法人情報処理推進機構）
セキュリティセンター

はじめに

サイバー攻撃は、あらゆる企業・組織に対して試みられている。このような状況の中、実被害が発生したか否かに関わらず、自組織（あるいは ISAC[2]等の会員組織）に対して試みられた攻撃について、分析し、その情報を蓄積し、更に可能ならば他組織と情報共有していくことには一定の意義があるものと考えられる。例えば、その攻撃が広く無差別に行われたと思われるものの一部であるのか、または意図的に攻撃対象が絞られたもの（標的型攻撃）であるのかによって、被害組織での対応の優先度や扱いは異なるだろう。また、自組織や関連業界が過去に受けた攻撃と関係があるか（連続性があるか）といった点についても、分析・蓄積・共有を重ねた結果、判明することがある。分析には様々な観点や方法があり、その一つとして、攻撃に用いられたウイルス等の不正ファイルの解析がある。

IPA セキュリティセンターでは、J-CSIP の運用をはじめとして、サイバー攻撃の情報を分析するため、必要に応じてウイルス等の解析を行っている。ウイルス解析は、一般的に、表層解析、動的解析、静的解析といった手法が用いられ、各手法に応じて様々なツールが存在する。これらの作業は、環境整備、人的資源、工数等の問題から、簡単に実践できるような性質のものではないと思われるが、一部組織の CSIRT[3]や ISAC 等では、取り組みを進めているところもある。

ウイルス解析に関する動向として、2019 年 3 月 6 日（日本時間）、米国の国家安全保障局（NSA）より、オープンソースのリバースエンジニアリングフレームワークである「Ghidra」が公開されたことが挙げられる。商用製品のような保証やサポートを受けることはできないが、無償で使用できることから、分析に使用するツールの選択肢として検討範囲に入ってくるだろう。Ghidra には静的解析のための機能が含まれており、これをウイルス解析に使用する場合の長所や短所、また、広く使われている類似ツールとの比較の観点から、同様のリバースエンジニアリングツールである IDA Pro[4]との機能的な異同の概要を確認した。本書では、参考情報として、その結果を報告する。

---

[1] Ghidra（NSA）
https://www.nsa.gov/resources/everyone/ghidra/
本書では、Version 9.0.2 を対象とする。

[2] Information Sharing and Analysis Center（ISAC、アイザック）。同じ業界の民間事業者同士でサイバーセキュリティに関する情報を共有し、サイバー攻撃への防御力を高めることを目指して活動する民間組織。

[3] Computer Security Incident Response Team（CSIRT、シーサート）。組織内の情報セキュリティ問題を専門に扱う、インシデント対応チーム。

[4] IDA:About（Hex-Rays）
https://www.hex-rays.com/products/ida/index.shtml
本書では、Version 7.2.181105 を対象とする。

逆アセンブル機能

同一のウイルスに対して両ツールにて逆アセンブルを行い、結果を比較したところ、命令の表現に一部違いはあるが、同等の逆アセンブルコードが出力されることが確認できた（図 1）。現時点で確認した限りにおいては、Ghidra の逆アセンブル機能を静的解析に活用することは十分可能であると考える。

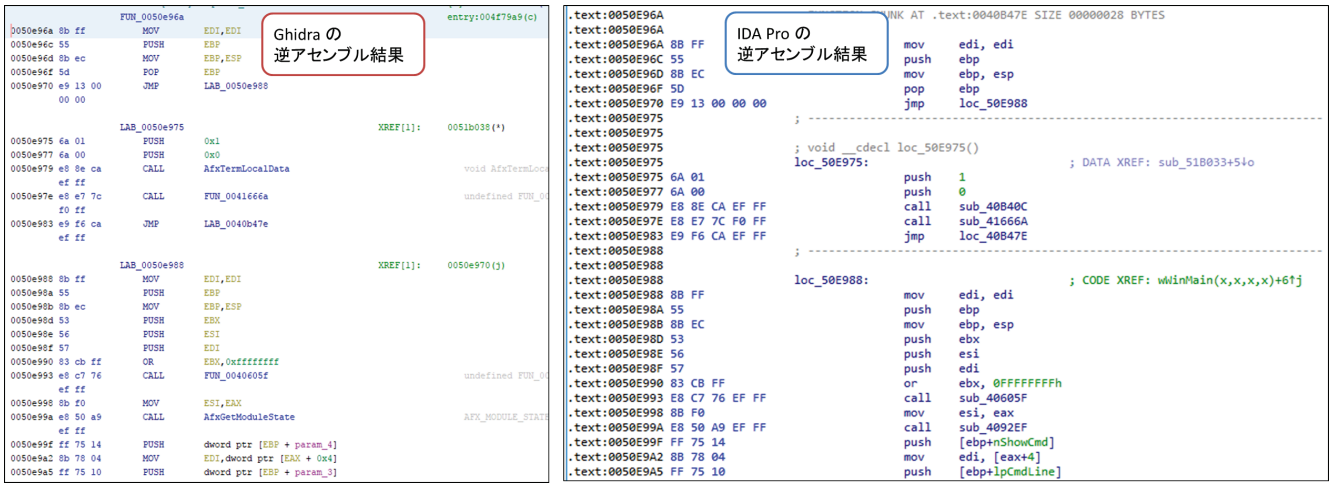

図 1 Ghidra と IDA Pro で逆アセンブルした結果の比較（同一ウイルスの同一箇所）

機能の差異

両ツールで機能に大きく差異を認めた 3 点について説明する。なお、この他にも機能差や使用感の違い等を感じる部分はあったが、現時点では十分な評価を行っているわけではないため、本稿ではこれ以上の言及は行わない。

デバッガ機能

IDA Pro と Ghidra の両方とも、逆アセンブルされたコード等から静的解析を行う機能があるが、IDA Pro には標準でデバッガ機能が付属しているため（図 2）、ウイルスの動的解析手法を同一ツール内で併用することができる。Ghidra にはデバッガ機能はないが、ウイルス解析等で使用するデバッグツール（OllyDbg[5]や x64dbg[6]等）を併用することにより、この機能差はカバーすることができるものと考える。

[5] OllyDbg
http://www.ollydbg.de/

[6] x64dbg
https://x64dbg.com/

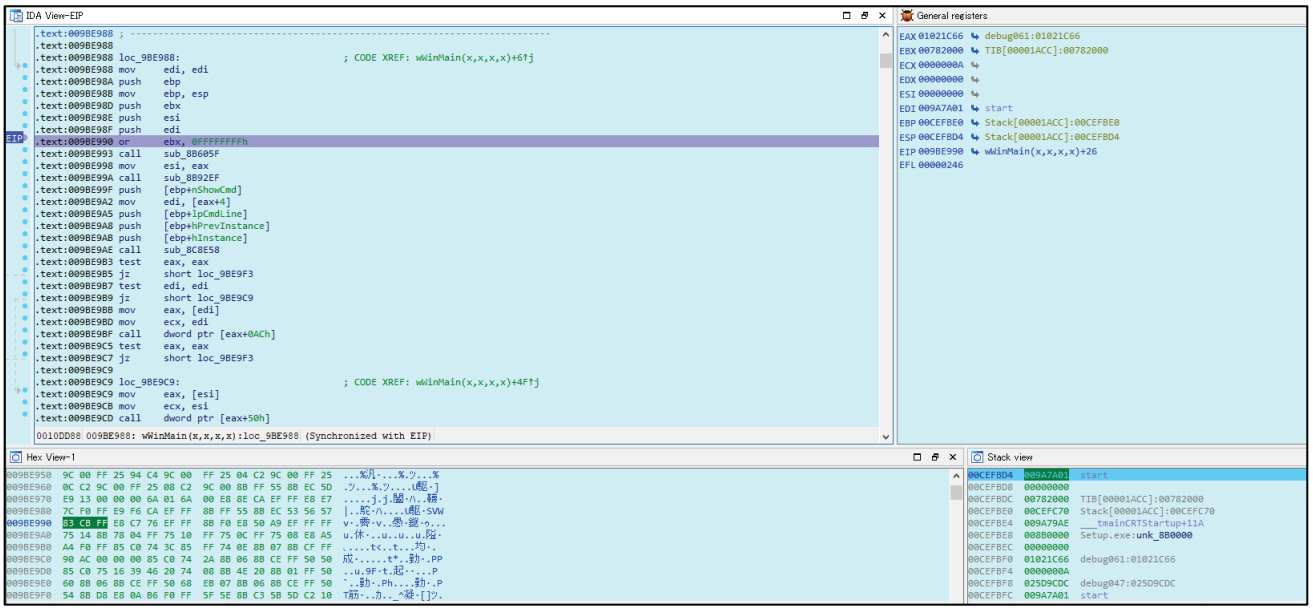

図 2 IDA Pro のデバッガ実行画面

デコンパイル機能

IDA Pro にはデコンパイル機能は標準では付属していないため、別途デコンパイラ機能のライセンスを追加購入する必要がある。一方、Ghidra にはデコンパイル機能が付属しており、逆アセンブルされたコードに加え、デコンパイルされたコードを参照することができる。Ghidra の画面上では、逆アセンブル結果とデコンパイル結果を並べて表示することができ（図 3）、例えば、上下にスクロールしたり、カーソルで選択した箇所が、両方のウインドウで同期して表示される。この並列表示の機能を利用すれば、両方の結果を見比べながら静的解析作業を進めることが期待できる。

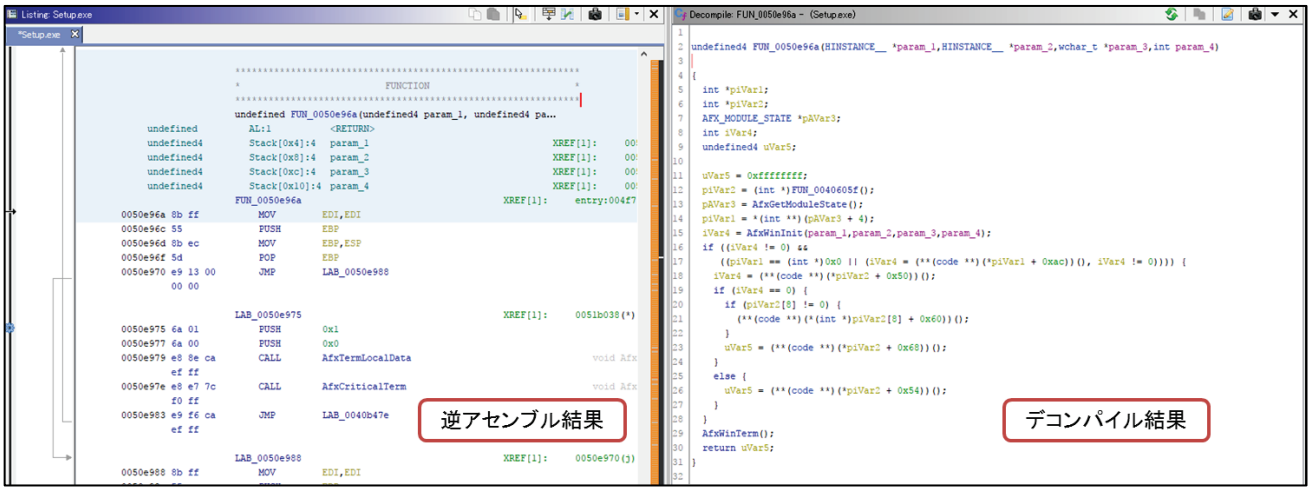

図 3 Ghidra で逆アセンブル結果とデコンパイル結果を並べて表示している画面

コード比較機能

逆アセンブルしたコードの比較（例えば 2 つのウイルスのコードを比較し類似点を確認する等）を行うためには、IDA Pro では第三者が提供しているプラグイン（BinDiff[7]等）を導入する必要があるが、Ghidra にはコード比較機能が付属している（図 4）。

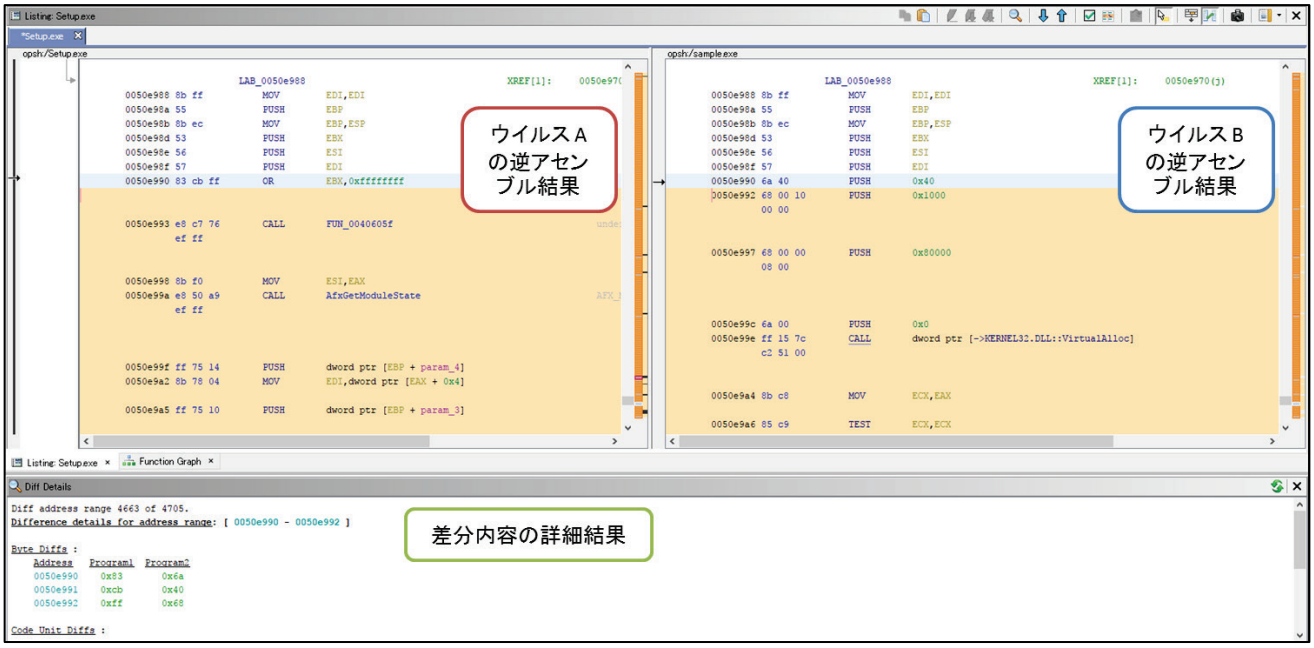

図 4 Ghidra のコード比較機能の画面

[7] zynamics BinDiff（zynamics）
https://www.zynamics.com/software.html

Ghidra のデコンパイル機能

前述の通り Ghidra のデコンパイル機能は無償で利用でき、局面によってはウイルスの解析に役立つことが期待できるため、本項で 2 つ例を示す。

条件分岐によるネストが深いコードや、多重ループのコードの解析

条件分岐やループとなっている処理が if 文や while 文等としてブロック構造で表示（図 5）され、コードの処理内容を理解しやすくなる場合があると思える。

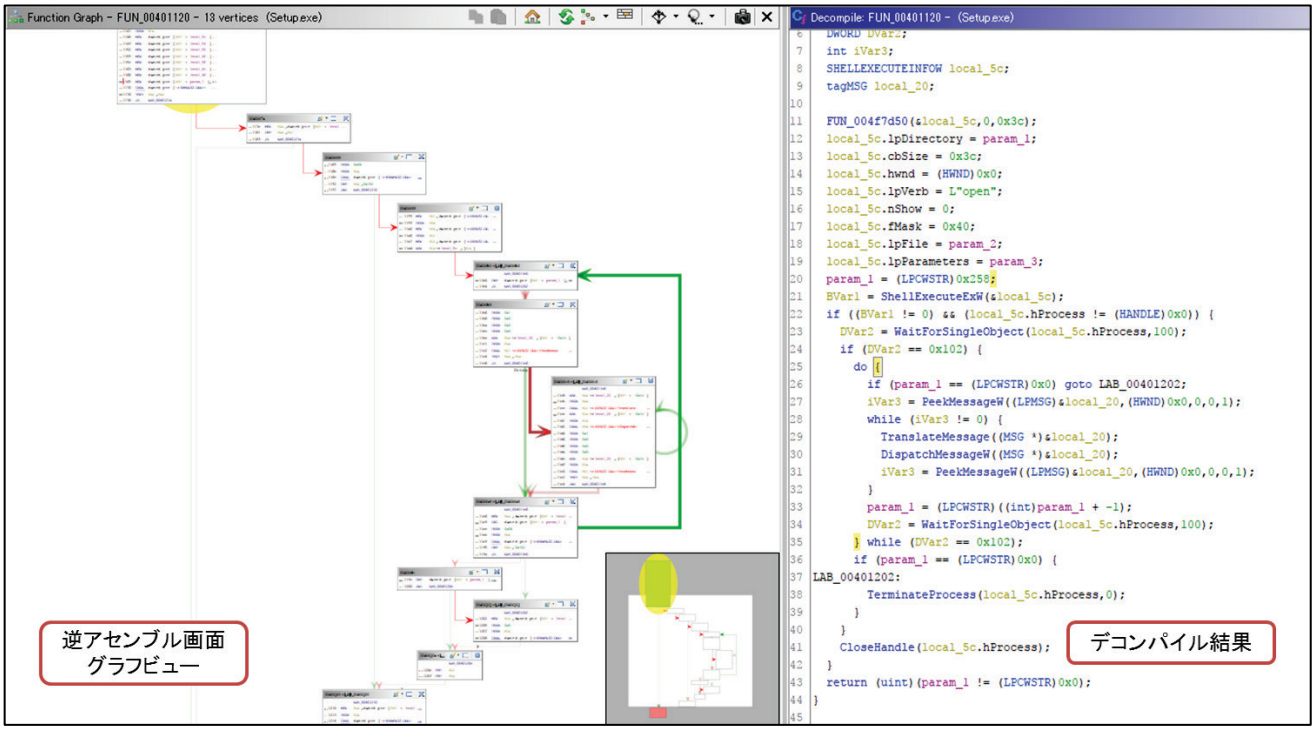

図 5 条件分岐とループを含むコードの解析の例（Ghidra）

算術演算、論理演算、シフト演算の処理の解析

INC、DEC、XOR 等の演算命令が演算子（+、−、^等）による表現（図 6）となり、暗号化・復号処理など複雑な演算を行うコードを理解しやすくなる場合があると思える。

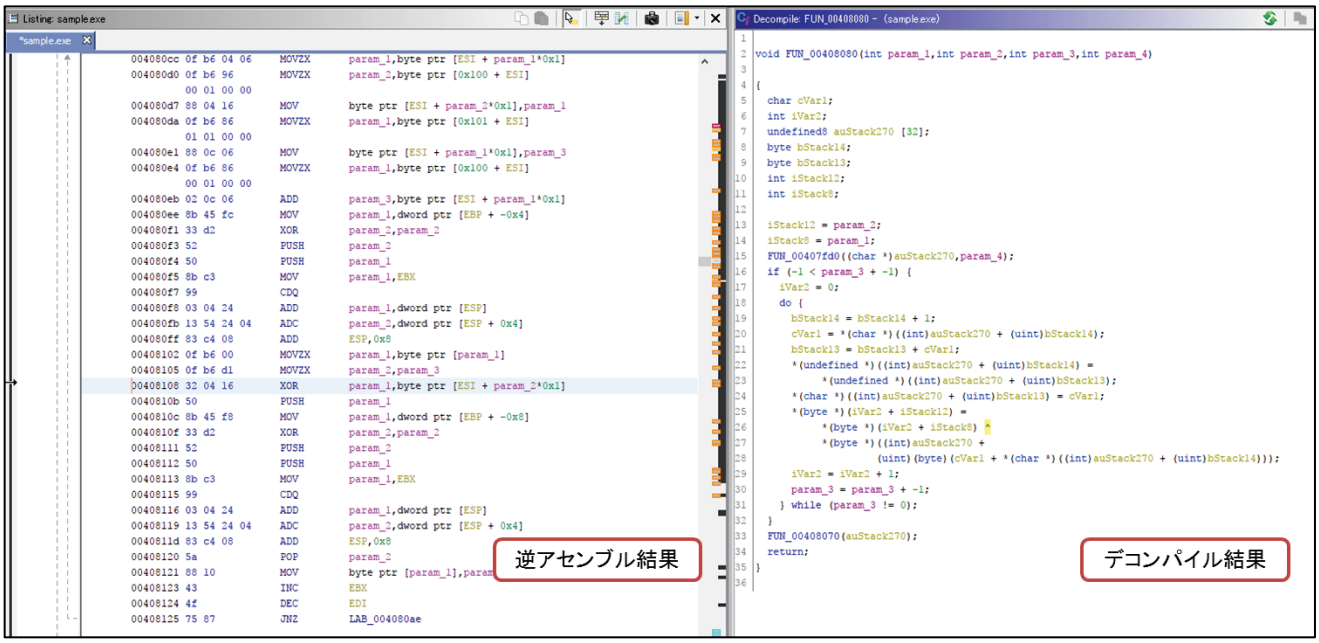

図 6 暗号化処理を行っている部分の解析の例（Ghidra）

おわりに

Ghidra は、公開から日が浅く、リサーチャーをはじめとする利用者による情報が少ないこと等から、信頼性の評価はまだ難しいが、上述の例を含め試験的に数個のウイルスを調査した範囲では、逆アセンブルの機能に大きな遜色はなく、デコンパイル機能は、上記の通り解析作業を十分支援しうると思われる。

また、Ghidra と IDA Pro のどちらが優れているかという観点から見るものではなく、例えば、用途に応じて相互の機能を併用／補完しつつ、ウイルス解析を行っていくといった使い方も考えられるだろう。

本書の内容について

本書執筆： 伊藤 博康