Computer Security Symposium 2018
22 - 25 October 2018

Ethereumブロックチェーン上に潜むマルウェアなどの定量的リスク分析

佐藤 哲平[1]　今村 光良[1,2]　面 和成[1]

概要：近年，社会基盤の新しい技術としてブロックチェーンが注目を集めている．暗号通貨は，ブロックチェーンを基盤としてパブリックに動作するシステムであるため，その振る舞いに対して，多くの研究結果が報告されている．先行研究では，Bitcoin に格納されるデータに着目し，悪意あるユーザーにより，違法なコンテンツが格納される，ブロックに対するポイズニング攻撃の問題を指摘している．本研究では，より多くのデータがブロックに格納可能な，Ethereum ブロックチェーンに対する，ポイズニング攻撃の実態について調査し，潜在するリスクについて分析した結果を報告する．

キーワード：ブロックチェーン, 暗号通貨, Ethereum, マルウェア

A Quantitative analysis of the embedded risks on Ethereum blockchain

Teppei Sato[1]　Mitsuyoshi Imamura[1,2]　Kazumasa Omote[1]

Abstract: In recent years, Blockchain has attracted attention as a new technology of social infrastructure. Since cryptocurrency is a system that works publicly on the basis of the blockchain technology, many research results have been reported for system behaviors. In previous research, focusing on the bitcoin blockchain which stored data, points out the poisoning attack on a block that malicious user stores illegal content. In our works, we survey the actual situation of the poisoning attack on the Ethereum blockchain, which can store more data in the block, and report the result of analyzing the embedded risk.

Keywords: Blockchain, cryptocurrency, Ethereum, malware

1. はじめに

近年ブロックチェーンを基にした技術の発展は著しく，さまざまな特徴をもつ暗号通貨が開発され利用されている．多種多様な暗号通貨の普及によりただの決済手段としてだけでなくスマートコントラクトを利用したり，匿名性を保って決済を行うことなどが可能となっている．

その一方でブロックチェーンや暗号通貨に対する攻撃手法も多数発見されている．Matzutt らの研究では Bitcoin のブロックチェーンに対して任意のデータの埋め込みが可能であることが明らかにされた [7]．これはブロックチェーンに対して悪意のあるデータを埋め込むことによる攻撃が可能であることを示唆しており，実際に Bitcoin ブロックチェーンには違法な可能性のあるファイルが複数埋め込まれていたことが報告されている．ただしこれらのファイルが埋め込まれていた領域は意図して設計された領域ではなく，ファイルを埋め込んだトランザクションをマイナーがブロックには含めないなどの対策をとることが可能である．よって Bitcoin のブロックチェーンにデータを埋め込むことができる 1 つの領域のサイズは，データを埋め込むことを意図して設計された領域の 80byte が最大である．これに対して Ethereum のトランザクションにはスマートコン

[1] 筑波大学
〒 305-8573 茨城県つくば市天王台 1-1-1
University of Tsukuba
Tennodai 1-1-1, Tsukuba, 305-8573 Japan
[2] 野村アセットマネジメント株式会社
〒 103-0027 東京都中央区日本橋 1 丁目 11-1
Nomura Asset Management Ltd.
1-11-1, Nihonbashi, Chuo-ku, Tokyo 103-8260, Japan

トラクトのコードなどに利用される任意のデータを埋め込むことのできる領域が存在しており，その領域のサイズは数百 kB 程度と Bitcoin よりも大きいサイズである．つまり Ethereum ではこの大きな領域を利用した埋め込み型の攻撃が可能である．

ブロックチェーンに対する違法または悪意のあるファイルの埋め込みはポイズニング攻撃の一種に位置づけることが可能であり，従来の公開データベースに対するポイズニング攻撃 (DNS キャッシュなど) よりも対応が困難である．なぜならブロックチェーンのデータの取り消しや修正が不可能である特性からポイズニングされたブロックチェーンを元に戻すことは不可能であり，またブロックチェーンに埋め込まれた情報は P2P ネットワークを通じてネットワーク参加者に共有されるため多くに影響を与えることになるからである．

そこで本論文では，大きいデータの埋め込みが可能な領域を用いて Ethereum のブロックチェーンに対して現状でどのようなファイルが埋め込まれているのかを調査し，ポイズニング攻撃のリスクについて分析を行った．その結果として Ethereum のブロックチェーンには少なくとも 77 のファイルが埋め込まれていることがわかり，その中には好ましくない内容の画像ファイルやマルウェアが含まれていたことを明らかにした．またブロックチェーンに対するポイズニングによる攻撃シナリオについて議論を行った．

2. 関連研究

2.1 ブロックチェーンに対する埋め込み型の攻撃

暗号通貨・ブロックチェーンに関わる攻撃としては，すでに使用した通貨をもう一度使用する 2 重支払いや，マルウェアなどによって計算資源を窃取してマイニングを試みる攻撃，ネットワークの計算量の大部分を独占することでブロックの採掘を独占することなどを可能とする 51%攻撃などが代表的である．

Matzutt らの研究では Bitcoin のブロックチェーンに対するデータの埋め込みについて調査している [7]．この研究では，Bitcoin のブロックチェーンに任意のデータを埋め込む手法を調査し，データ埋め込みの可能な領域として意図して設計された Bitcoin の Script の一命令である OP_RETURN による 80byte の領域とブロックのマイナーが自由なデータを書き込むことができる Coinbase トランザクションによる 96byte の領域が存在していること，また意図されたものではないが 100kB 程度のデータを埋め込むことのできる手法が存在すること，そしてそれらの手法を応用してユーザがブロックチェーンの構造やデータの埋め込み手法を意識せずにブロックチェーンにデータを埋め込むことのできるデータ埋め込みサービスも存在することを示した．さらにブロックチェーンにデータを埋め込むことの利点と危険性について議論し，実際に Bitcoin ブロックチェーンに対して埋め込まれたデータについて調査した．この結果として児童ポルノへのリンク集など多くの国で違法な可能性の高いファイルやプライバシーを侵害する内容を含んだファイルが Bitcoin のブロックチェーンに埋め込まれていたことを明らかにした．

ブロックチェーンに違法なファイルや悪質なファイルを埋め込むことによりブロックチェーン自体が違法・悪質なものとなってしまう可能性があり，これは一種のポイズニング攻撃であると言える．公開データベースに対するポイズニング攻撃として広く知られているものとして DNS キャッシュポイズニング [9] があるが，これは攻撃の検知後にデータの削除や修正が可能である．しかしブロックチェーンの場合，ブロックチェーンのデータの取り消しや修正が不可能であるという特性から同様の対応が不可能である．またブロックチェーンに埋め込まれた情報は P2P ネットワークを通じてネットワーク参加者に拡散されるため多くに影響を与えることが可能な攻撃である．

ただし Bitcoin において意図して設計されたデータ埋め込み可能な領域は OP_RETURN による領域と Coinbase による領域であり，また Coinbase に関してはマイナーのみが書き込むことができるので，一般のネットワーク参加者は 1 つの領域に対して最大 80byte のデータしか埋め込むことができない．また大容量な領域を利用可能な意図されていない埋め込み手法を用いているトランザクションは正式なものではないためマイナーがブロックに含めないという対応をとることが可能である．一方で Ethereum ではトランザクションにデータを埋め込む領域が用意されており，そのサイズは Bitcoin で許可された 80byte よりも大きな領域であるためこの領域を悪用した攻撃が可能であると考えられる．図 1 に Ethereum ブロックチェーンにおけるポイズニング攻撃の概要を示す．

2.2 Ethereum に対する攻撃

スマートコントラクトという形でブロックチェーン上でプログラムを実行することができる Ethereum では，通常のプログラムと同様にスマートコントラクトの実装による脆弱性の他に，ブロックチェーンを用いていることによる特有の脆弱性が存在している．

Atzei らの研究ではこの Ethereum のスマートコントラクトにおける攻撃について調査を行っている [1]．この論文ではスマートコントラクトの脆弱性を Solidity によるもの，EVM code によるもの，ブロックチェーンによるものの 3 つのレベルに分類して説明した上で，それらの脆弱性を用いた攻撃について Solidity で記述されたコントラクトを例にあげて示している．

この論文で挙げられた攻撃はスマートコントラクトの脆弱性を用いてそのコントラクトの利用者や管理者に対して損害を与えるものであった．一方本論文で対象とする悪意

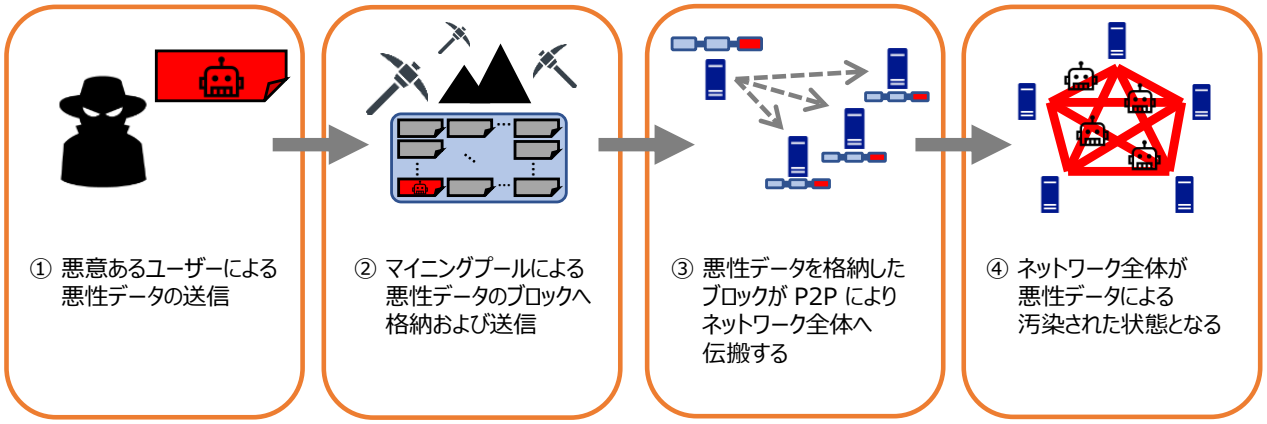

図 1　Ethereum ブロックチェーンにおけるポイズニング攻撃の概要

のあるデータ埋め込みによる攻撃はそのブロックチェーンを使用したシステムの利用者全体やシステム自体に影響を与えることが可能であるため，ブロックチェーンに対するより直接的な攻撃であると言える．

3. Ethereum について

Ethereum とはスマートコントラクトを実行する分散型プラットフォームである [3]．Ethereum には Externally Owned Account(EOA) と Contract Account(CA) という 2 種類のアカウントが存在している．EOA は秘密鍵によって管理され，EOA と紐付く秘密鍵を持つユーザはその EOA によって基軸通貨である ether を送ったりコントラクトの生成，実行などができる．一方で CA はそれぞれがコントラクトのコードを持っており，それを実行する際などに使用される．

Ethereum 上で実行されるスマートコントラクトは，Ethereum ネットワーク上で実行される Ethereum Virtual Machine(EVM) と呼ばれる仮想マシンによって実行される．EVM が実行するのは EVM code と呼ばれるバイトコードだが，高水準言語によってコントラクトを記述してそれを EVM code にコンパイルすることができるプログラミング言語が用意されており，比較的容易にスマートコントラクトを実装することが可能となっている．EVM code を生成するための高水準言語として代表的なものには Javascript に似た文法でコントラクトを記述することができる Solidity[5] がある．Solidity などの高水準言語で記述されたコントラクトはコンパイラによって EVM code にコンパイルされ，EOA が送信先アドレスを Null にしたトランザクション (Contract Creation Transaction) に EVM code を格納して送信することでブロックチェーン上で CA が作られることになる．このコントラクトを実行するためには，生成された CA に対してコントラクトの入力をもったトランザクションを EOA から送信することが必要となる．

Ethereum では，ether の送金やコントラクトの生成・実行のためにトランザクションを送信する際には他の通貨と同様に手数料がかかる．Ethereum の手数料は gas という単位であり，gas はそのトランザクションの種類やコントラクトの実行時に使用される命令の種類，init/data 領域に書き込まれているデータのサイズなどによって機械的に決定される．gas はトランザクションの送信時ではなく，トランザクションが実行された時点で決定されるため，ユーザはトランザクションの送信時にトランザクションの実行に使用される gas の上限値を指定する gasLimit という値を指定することになる．これによってコントラクトの不具合等によって想定していた以上の gas が使用されてしまうようなことを防ぐことができる仕組みとなっている．またユーザはトランザクションの送信時に gasPrice という値を設定する．これは 1gas が何 ether にあたるかを送信するユーザが決定するためのフィールドとなっている．例えばトランザクションの実行に 50000gas が必要なトランザクションを送信する際にユーザが gasPrice を 30Gwei に指定していた場合，ユーザがマイナーに支払う手数料は $50000 \times 30\text{Gwei} = 0.0015\text{ether}$ と求められる．ただし wei とは ether の最小単位であり，$1\text{wei} = 10^{-18}\text{ether}$ である．よってこの gasPrice の値を大きくすることでマイナーに優先的に処理を行わせ，ブロックに加えられるまでの時間を短くすることが可能となる．

Ethereum のブロックにはトランザクションだけでなくブロックごとにも gasLimit というフィールドが存在している．これはブロックに入ったトランザクションによって消費される gas の総和の上限を示しており，ブロックを生成したマイナーらの投票によってこの値は変化している．2018 年 8 月 18 日現在でのブロックの gasLimit は約 8,000,000gas となっている．

4. Ethereum ブロックチェーンの任意データが埋め込み可能な領域

Ethereum ブロックチェーンには以下の 2 つの任意データ埋め込みが可能な領域が存在する．

- extraData 領域
- init/data 領域

本章ではこれらそれぞれの領域の特徴について説明する．

4.1 extraData 領域

extraData 領域は Ethereum のブロックのヘッダにあり任意の byte 列を指定することができる領域である．ただしブロックヘッダの領域であるためその領域にデータを埋め込むことができるのはそのブロックのマイニングに成功したマイナーのみであり，また extraData 領域に埋め込むことのできる byte 列は最大で 32byte と決められている．

4.2 init/data 領域

Ethereum においてコントラクトを生成するために使用する Contract Creation Transaction には，EVM で実行される EVM code が埋め込まれている必要がある．この EVM code が埋め込まれる領域が init と呼ばれる領域である．

またコントラクトの実行のために EOA から CA にトランザクションを送信する際など，アカウント間でデータの送信を行う場合はトランザクションの data と呼ばれる領域にそのデータを埋め込んで送信することになる．

init 領域と data 領域は両方とも Ethereum のトランザクションの一領域であり同様の性質を持っており一つのトランザクションにはどちらか一方しか存在していないため，本論文ではこの 2 つの領域についてひとまとめにして扱う．

Solidity などコントラクト実装のための高水準言語のコンパイラから出力された EVM code は 16 進数文字列の形をしている．ウォレットと連携して Solidity などのソースコードから直接コントラクトをデプロイできる IDE も存在しているが，IDE などを用いないでコントラクトをブロックチェーン上にデプロイする場合，その 16 進数文字列を Ethereum ウォレットのアプリケーションを通してトランザクションの init 領域に埋め込んで EOA によってそのトランザクションを送信するような方法をとる．コントラクトの実行時などアカウント間でデータを送信する際もトランザクションの data 領域に同様の手順でデータを埋め込む．そのため多くの Ethereum ウォレットはトランザクションの init/data 領域に書き込むデータを 16 進数文字列の形で受け取りそれを送信することが可能となっている．よってこの機能を用いることで任意のデータを Ethereum ブロックチェーン上に埋め込むことができるため，ファイルなども 16 進数の状態に変換することで埋め込むことが可能となる．

さらにこの init/data 領域にはサイズの制限がない．しかし事実上の制限は存在する．3 章で述べたように init/data 領域に埋め込んだデータのサイズが大きいほど支払う gas は大きくなり，あるブロックに含まれるのがそのトランザクションのみだったとしてもそのトランザクションが使用する gas はそのブロックの gasLimit を超えることが許されていない．そのためブロックの gasLimit がトランザクションの init/data 領域に埋め込むデータサイズのボトルネックになっている．

ブロックの gasLimit が 8,000,000gas とした場合，少なくとも 100KB から，最大で 2MB 程度のデータの埋め込みが可能である．

5. Ethereum ブロックチェーンの調査・分析

本章では，4 章で示した任意データが埋め込み可能な領域に実際にどのようなファイルが埋め込まれているのか調査した調査方法とその結果を示す．本論文で行った調査の対象は Ethereum ブロックチェーンである．Ethereum プロトコルの Official Implementation の一つである Go Ethereum[4] のクライアント Geth を利用してブロックチェーンの同期を行い，ローカルに保存されたブロックチェーンに対して分析を行った．

5.1 extraData 領域の調査・分析

4.1 節で述べたように，extraData 領域はブロックの採掘に成功したマイナーのみが指定できる領域であるため，その領域に任意のデータを埋め込む際の難易度が高い．またこの領域は一つのブロックにつき 32byte しかないためファイルを埋め込むためには十分でなくコストが非常に高い．

5.1.1 方法

Geth によってローカルに保存されたブロックチェーンから調査範囲のブロックのヘッダに関する情報を抽出し，その中の extraData 領域の情報を 16 進文字列とそれを ascii として変換した文字列の形でそれぞれ出力する．

調査範囲は Ethereum ブロックチェーンの Height 0 ～ 5,800,000 である．

5.1.2 結果

調査の結果，ascii に変換できなかったものが多数見受けられ，ascii に変換できたものの中には以下の 2 つの種類の文字列が多くを占めていることがわかった．

- Geth のバージョン情報
- マイニングプールの情報

Geth のバージョン情報は，Geth を使用してマイニングする際に extraData に書き込む情報を指定しない場合にデフォルトで書き込まれるものやそれを模倣したものであるとみられ，例えば”010802/geth/go1.9.4/linux”のように，Geth 自体のバージョン情報以外に Geth が動作している

表 1　調査範囲のトランザクションの init/data 領域のサイズ (byte)

最大	716,805
最小	0
合計	13,574,001,814

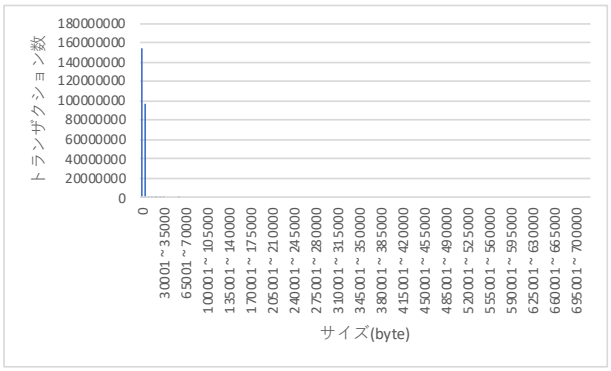

図 2　トランザクションの init/data 領域のサイズの分布

Go 言語のバージョンや OS の情報が書き込まれている．

マイニングプールの情報は，マイニングプールの名称やマイニングプールの Web サイトのドメインのようなものが多く，自分のマイニングプールにおいて採掘が成功しているということを示すことによって宣伝効果を狙ったものであると推測できる．

5.2　init/data 領域の予備調査

init/data 領域に埋め込まれたファイルの調査 (本調査) の前に調査対象がどの程度の量であるかを見極めるために init/data 領域のサイズを調査する予備調査を行った．これに関してはファイルが埋め込まれたトランザクションではなく期間内のすべてのトランザクションを対象としている．

5.2.1　方法

Geth によってローカルに保存されたブロックチェーンから調査範囲のブロックに含まれるトランザクションに関する情報を抽出し，init/data 領域のサイズを取得してその最大・最小値や分布を求めた．

調査範囲はブロックチェーンの Height 0 ～ 5,800,000 である．

5.2.2　結果

調査範囲ブロックに含まれるトランザクションは全部で 251,684,775 あり，それらのトランザクションの init/data 領域のサイズの最大値・最小値・合計値を表 1 に示す．また init/data 領域のサイズの分布を図 2 に示す．

この結果からは init/data 領域に 100kB 単位のデータが入ったトランザクションが存在しており，送金のみを行う init/data 領域のデータが 0byte のトランザクションが約 61%を占めていることがわかった．

5.3　init/data 領域の調査・分析

5.2 節の予備調査からある程度大きいサイズのデータを init/data 領域に持つトランザクションが存在していることがわかったため，それらのトランザクションの中にファイルが埋め込まれたものがあるかを調査した．

5.3.1　方法

まず Geth によってローカルに保存されたブロックチェーンから調査範囲のブロックに含まれるトランザクションの init/data 領域のデータを抜き出す．init/data 領域に書き込まれている 16 進数の状態のデータをそれぞれ以下の手順でファイルカービングの手法を用いて分析した．

(1) 16 進数の状態のデータをそのままバイナリファイルとして保存する．
(2) そのファイルをファイルカービングツールに入力として与える．
(3) ファイルの埋め込みが検知された場合はそのトランザクションの情報を記録する．

ファイルカービングとはファイルの種類ごとのシグネチャ情報を検索してファイルを探す手法であり，init/data 領域に対してファイルが埋め込まれている場合この手法によって検知が可能である．今回検知の対象としたファイルは jpg, gif, png, bmp, avi, exe, mpg, wav, riff, wmv, mov, pdf, ole, doc, zip, rar, html の 18 種類である．また今回はデータに何らかのエンコード・暗号化がされていたり，分割されて埋め込まれているようなファイルは調査の対象から除外している．

調査範囲はブロックチェーンの Height 0 ～ 4,230,740 である．

5.3.2　結果

まずは発見したトランザクションの init/data 領域に埋め込まれたファイルについて述べる．

調査範囲のブロックに含まれるトランザクションに対して 77 のファイルの埋め込みが確認できた．表 2 にその内訳を，表 3 に埋め込まれていたファイルのサイズの平均や最大最小などの統計量を示す．ファイルの種類に関しては jpg と png の画像ファイルが大部分を占めていることがわかる．画像ファイルの内容としては集合写真や人物写真やイラストや模様などの画像でありその内容に違法性がないと思われるファイルが大部分であったのに対し，好ましくない内容の画像ファイルも複数見つかった．

exe ファイルは 3 つのファイルが見つかったが，それらのファイルの MD5 ハッシュ値を以下に示す．

(1) c9a31ea148232b201fe7cb7db5c75f5e
(2) c1e5dae72a51a7b7219346c4a360d867
(3) c9a31ea148232b201fe7cb7db5c75f5e

ハッシュ値が同じであることから 1 つめと 3 つめの exe ファイルは同一のものであることがわかる．これらのハッシュ値を先行研究 [2] でも評価指標の一つとして用いられているオンラインのマルウェアスキャンサービスである

表 2　Ethereum ブロックチェーン埋め込まれていたファイルの内訳

jpg	42
png	23
html	7
exe	3
pdf	2
合計	77

表 3　埋め込まれていたファイルのサイズ (byte) に関する統計量

平均	14348
中央値	10829
最小	108
最大	60647
標本数	77

VirusTotal[*1]で検索した結果，1 つめと 3 つめのファイルは検出率が 55/65 と高い割合であったためマルウェアであると判断した．また W32.Duqu という名称で報告されている [10] マルウェアの一部のファイルとハッシュ値が一致していることを確認した．2 つめの exe ファイルも同様に VirusTotal で検索した結果，58/65 という高い検出率であったためこちらのファイルもマルウェアであると判断した．

またこれらの 3 つの exe ファイルを埋め込んだのは同一のアカウントで，exe ファイルを埋め込んだトランザクション 3 つを約 6 分の間に送信していることが分かった．

次にファイルが埋め込まれたトランザクションとそれを送信したアカウントについて結果を述べる．

調査範囲のトランザクションに埋め込まれたファイル数が 77 だったのに対し，ファイルを埋め込んだトランザクションを送信したアカウントは 62 であった．つまり複数のファイルをブロックチェーンに埋め込んだアカウントが存在しており，最大で 1 つのアカウントが 6 のファイルを埋め込んでいることが確認できた．

ファイルが埋め込まれたトランザクションの送信元アカウントと送信先アカウントの関係については表 4 のような分布になった．多くのアカウントがファイルを埋め込む際に自分に対して送っているのに対して，他のアカウントに対して送っているものも一定数存在しており，Contract Creation Transaction に埋め込んでいるものも 1 例存在していた．ただし，本論文では送信元と送信先が同一である，つまり自分に対して送られるトランザクションを self send トランザクションと呼ぶこととする．

6. 考察

6.1 調査結果について

Matzutt らの研究 [7] では多数のファイルが発見されその中には違法な可能性の高いファイルが見つかっている．

[*1] VirusTotal: `https://www.virustotal.com/`

表 4　ファイルが埋め込まれたトランザクションの送信元アカウントと送信先アカウントの関係

送信元と送信先が同一 (self send トランザクション)	58
送信元と送信先が異なる	18
送信先が Null(Contract Creation Transaction)	1

本研究の調査においても init/data 領域からはそれを不特定多数が見られる状態にすることは違法であるとみられる画像ファイルが発見された．また init/data 領域からは 3 つのマルウェアも発見された．

これらの違法な可能性の高いファイルが Ethereum の P2P ネットワークを通じてその参加者と共有されたということであり，またこれらのファイルを削除することはブロックチェーンの性質から非常に困難である．

6.2 マルウェアを埋め込んだアカウントの振る舞い

5.3.2 項で述べたように，Ethereum のブロックチェーンにはマルウェアとみられる exe ファイルが 3 つ埋め込まれておりそれらのファイル埋め込みは同じアカウントによるものだった．このアカウントのその他のトランザクションにおいて特徴的な振る舞いが見られたためそれについて考察する．

当該アカウントは合計で 10 のトランザクションを送信しており，それらのトランザクションの特徴を以下に時系列に示す．

(1) アカウント A からの送金トランザクション
(2) アカウント B への送金トランザクション
(3) ∼ (8) self send トランザクション (data 領域にデータあり)
(9) アカウント C への送金トランザクション
(10) アカウント C からの空のトランザクション

この内 (3) ∼ (8) のデータが埋め込まれた self send トランザクションに注目する．まず (3),(4) のトランザクションの data 領域にはそれぞれランダムと見られるバイト列が埋め込まれており，(3) は 4.1kB，(4) は 20.5kB であった．次に (5) のトランザクションの data 領域には 8KiB の壊れた png の画像がうめこまれていた．なおこの png 画像は壊れていたためファイルカービングの際には検知から漏れている．(6) ∼ (8) のトランザクションの data 領域には 5.3.2 項で示したようにそれぞれ 1 つずつマルウェアと見られるファイルが埋め込まれていた．

この一連の流れから次のようなストーリーが推測できる．このアカウントはデータを埋め込むために必要な手数料である ether を他のアカウントから受け取った上でまずは 2 つのサイズの違うランダムなデータを用いて埋め込み可能なデータサイズを確認，その後 png の画像データを使ってファイルの埋め込みが可能かどうかを確認し，最後にマルウェアという悪性なファイルの埋め込みが可能かどうかを

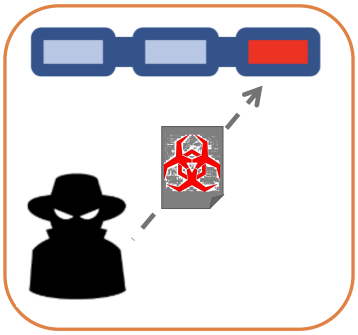

図 3　ブロックチェーンに悪意のあるデータを埋め込む攻撃者

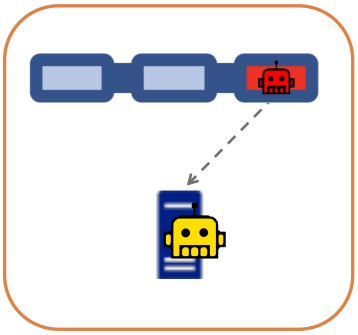

図 4　ブロックチェーンに埋め込まれたマルウェアを取得する別のマルウェア

確認した．この一連の振る舞い自体が攻撃を意図したものであるかは定かではないが，攻撃のために任意ファイルの埋め込みと悪性ファイルの埋め込みの可能性を確かめるために試行を繰り返しているということは考えられる．

6.3　ブロックチェーンへのポイズニングを利用した攻撃シナリオ

本節では，ブロックチェーンに任意のデータを埋め込むことが可能な領域が存在した場合に可能となると想定されるポイズニング攻撃の攻撃シナリオを示す．

6.3.1　ブロックチェーンを基盤としたシステムへの DoS 攻撃/暗号通貨の価格操作

ブロックチェーンはそのデータが P2P ネットワークを介して参加者に共有されるため，悪質なデータを埋め込まれた場合それはそのネットワークに参加するすべてのノードにおいてダウンロードされ，保持されることになる．そしてそれらのデータをブロックチェーンから削除することは非常に困難である．

攻撃者はブロックチェーンに対して違法な内容のファイルを埋め込み，それを何らかの方法でそのブロックチェーンを基にするシステムの利用者を含む大勢に周知することによって，そのシステムへのネガティブな印象を与えサービスを妨害する (DoS) ことが可能である．さらにそのシステムが暗号通貨であればこれによりその暗号通貨の価格を下落させることが可能となる．よってこの方法で攻撃者は意図的に暗号通貨に価格差を生じさせることが可能であり，空売りなどの方法で投資を行うことで利益を得ることが可能となる．この攻撃のイメージを図 3 に示す．

6.3.2　ブロックチェーンをマルウェアの通信先とする攻撃

ブロックチェーンは P2P ネットワークで共有されたデータをそれぞれのノードがローカルに保存している必要がある．またそれらのデータはネットワークに参加することで誰でも取得することが可能である．この特徴を利用することで，マルウェアの C&C 通信や，ドロッパーと呼ばれるマルウェアのダウンロード元としてブロックチェーンを悪用することが可能と考えられる．この攻撃のイメージを図 4 に示す．

攻撃者はマルウェアへ与えたい情報や，ドロッパーにダウンロードさせるマルウェア自体をブロックチェーンに埋め込む．攻撃対象に侵入後のマルウェアがブロックチェーン上に埋め込まれた情報を取得する方法としては以下の 3 つが考えられる．

- マルウェア自体がブロックチェーンの P2P ネットワークに参加して情報を取得する
- すでに侵入先に存在するブロックチェーンのデータから情報を取得する
- ブロックチェーンの情報を公開している Web サイトを利用して情報を取得する

まずマルウェア自体がブロックチェーンの P2P ネットワークに参加して情報を取得する方法は，文字通りマルウェアがノードとしてブロックチェーンのネットワークに参加して，通常のノードと同様の通信を行って P2P ネットワークを通じてブロックの情報を取得する．この手法は，企業などのネットワークなどではブロックチェーンを共有するための通信が許可されていない場合や通常使用されない通信のため容易に検知されてしまう可能性が高いという欠点がある．ただし暗号通貨のノードが動作しているサーバなど，その通信がすでに使用されているような環境である場合通信による検知は難しく，またこの手法ではマルウェアが情報を取得するだけでなくトランザクションを送信することによって外部に情報を送信することもブロックチェーンを介して行うことが可能である．

すでに侵入先に存在するブロックチェーンのデータから情報を取得するという方法は侵入先がブロックチェーンネットワークのノードとして動作している場合にのみ可能な方法である．ノードが動作している場合ブロックチェーンのデータはローカルに保存されているのでその情報をそのまま利用することができる．この手法はマルウェアが情報を受け取るために通信を発生させる必要がないという特徴がある．例えば暗号通貨のウォレットが動作しているサーバを標的としたマルウェアなどであれば，この手法を用いることでネットワーク通信による検知が困難となる．

ブロックチェーンの情報は公開情報であるため，ノードとしてブロックチェーンのネットワークに直接参加して情報を得る他に，ブロックチェーンの情報を公開している Web サイトを利用して情報を得ることも可能となっている．このような Web サイトは複数存在しており，例えば Etherscan[*2]などがある．ブロックチェーンの情報を公開している Web サイトを利用して情報を取得する方法とは，この Web サイトにマルウェアがアクセスすることでブロックチェーンの情報を取得するというものである．この手法の特徴として，企業などのネットワークにおいても HTTP(S) の通信は許可されていることがほとんどであるほか，HTTPS の通信であれば取得している詳細な情報が暗号化されており検知がしにくいことなどが挙げられる．さらにこの手法の最大の利点は，通常の悪性でない Web サイトであるブロックチェーンの情報を公開しているサイトをマルウェアの通信先とすることにより，その Web サイトを閲覧してブロックチェーンの情報を調べている通信とマルウェアによる通信の見分けがつきにくいということである．

ただしこれらの手法には共通した攻撃者側の欠点も存在しており，それはブロックチェーンに埋め込んだ情報は取り消したり変更したりすることが不可能であるため，例えばマルウェアをブロックチェーンに埋め込んでいることが明らかになってしまえばマルウェア自体の解析を逃れることができないという点である．ただし埋め込んだマルウェアなどが今回の我々の調査のような手法によって明らかになってしまうことを防ぐために，複数のトランザクションに分割することや，何らかのエンコード・暗号化を施すことで発見を遅らせることが可能であると考えられる．

また 4.2 節では理論的に埋め込み可能なサイズを示したが，Shahzad らの研究 [8] において調査されたマルウェアのサイズを基にすると，88%程度のマルウェアが 1 つのトランザクションに埋め込み可能であることがわかる．

7. まとめ

本研究では，ブロックチェーンにユーザが任意のデータを埋め込むことが可能であるという特徴に着目し，その特徴を用いたポイズニング攻撃の可能性について調査を行った．Ethereum のブロックチェーンには実際に複数のファイルが埋め込まれており，その中には悪質な内容の画像ファイルやマルウェアとみられるものもあった．これによりブロックチェーンへのポイズニング攻撃が実際に可能であることがわかり，実際に攻撃のための試行を行っているように見えるアカウントを発見した．またブロックチェーンへのポイズニング攻撃の攻撃シナリオとして，違法なファイルを用いた暗号通貨の価格操作や，ブロックチェーンをマルウェアの通信先として利用する可能性について議論を行った．

ブロックチェーンに対する悪意あるファイルの埋め込みへの対策として，Matzutt らの研究 [6] ではデータが埋め込まれたトランザクションをフィルタリングする手法や大きいデータを埋め込んだ場合の手数料を上げることで埋め込むコストを上げる手法などが提案されているが十分なものであるとは言えないため，今後も議論が必要である．

[*2] Etherscan: `https://etherscan.io/`

参考文献

[1] Atzei, N., Bartoletti, M. and Cimoli, T.: A Survey of Attacks on Ethereum Smart Contracts (SoK), *Principles of Security and Trust* (Maffei, M. and Ryan, M., eds.), Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 164–186 (2017).

[2] Chatterjee, R., Doerfler, P., Orgad, H., Havron, S., Palmer, J., Freed, D., Levy, K., Dell, N., McCoy, D. and Ristenpart, T.: The Spyware Used in Intimate Partner Violence, *2018 IEEE Symposium on Security and Privacy (SP)*, Vol. 00, pp. 993–1010 (online), DOI: 10.1109/SP.2018.00061.

[3] Ethereum: Ethereum Project, `https://www.ethereum.org/` Accessed: 2018-07-25.

[4] Ethereum: Go Ethereum, `https://geth.ethereum.org/` Accessed: 2018-08-09.

[5] Ethereum: Solidity, `https://github.com/ethereum/solidity` Accessed: 2018-07-29.

[6] Matzutt, R., Henze, M., Ziegeldorf, J. H., Hiller, J. and Wehrle, K.: Thwarting Unwanted Blockchain Content Insertion, *2018 IEEE International Conference on Cloud Engineering (IC2E)*, pp. 364–370 (online), DOI: 10.1109/IC2E.2018.00070 (2018).

[7] Matzutt, R., Hiller, J., Henze, M., Ziegeldorf, J. H., Müllmann, D., Hohlfeld, O. and Wehrle, K.: A Quantitative Analysis of the Impact of Arbitrary Blockchain Content on Bitcoin (2018).

[8] Shahzad, F. and Farooq, M.: ELF-Miner: using structural knowledge and data mining methods to detect new (Linux) malicious executables, *Knowledge and Information Systems*, Vol. 30, No. 3, pp. 589–612 (online), DOI: 10.1007/s10115-011-0393-5 (2012).

[9] Son, S. and Shmatikov, V.: The hitchhikers guide to DNS cache poisoning, *International Conference on Security and Privacy in Communication Systems*, Springer, pp. 466–483 (2010).

[10] Symantec: W32.Duqu The precursor to the next Stuxnet, `https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_duqu_the_precursor_to_the_next_stuxnet.pdf` (2011).