Computer Security Symposium 2020
26 - 29 October 2020

ブロックチェーンの汚染による悪用リスクの考察

木村 圭吾[1,a)]　面 和成[1,2]

概要：近年，情報通信技術の発展に伴い，情報システムの新たなプラットフォームとしてブロックチェーンの普及が広がってきている．ブロックチェーンが従来のサーバ中心のシステムと異なる点として，耐改ざん性と高可用性が挙げられる．これらの性質から，ブロックチェーンを基盤とするシステムはデータの改ざんが不可能であるだけでなく，システムダウンが生じず，強固なシステムを構築することが可能であるという特徴がある．しかし，このブロックチェーンの長所は攻撃者に対しても有利に働く．ブロックチェーンにはトランザクションを介して任意のデータを埋め込むことが出来る領域が存在する．この改ざん不可能かつ消失しないデータ領域を利用して悪性データを保管したり，ボットネットの C&C チャネルの構成に利用可能であったりするというリスクが先行研究でも提示されている．本研究では，このようなブロックチェーンの汚染がもたらすポイズニング攻撃のリスクについて分析し，実際に汚染されたブロックチェーンを用いた攻撃の容易性について実証とともに考察する．

キーワード：Blockchain，汚染データ，C&C チャネル，ボットネット，Ethereum

Consideration of abuse risk due to blockchain pollution

KEIGO KIMURA[1,a)]　KAZUMASA OMOTE[1,2]

Abstract: In recent years, blockchain has become widespread as a new platform for information systems. The blockchain differs from the server-centric system in terms of tamper resistance and high availability. Due to these characteristics , the blockchain system can build a robust system in which data cannot be tampered with and system down does not occur. But the strengths of this blockchain also benefit criminals. There is a region in the blockchain where arbitrary data can be embedded via transactions. Previous research has also presented the risk that malicious data can be stored using this data area that cannot be tampered with and that cannot be lost, and that it can be used for the construction of C&C channels for botnets. In this research, we analyze the risk of poisoning attacks caused by such blockchain pollution, and discuss the easiness of the attack using the polluted blockchain together with the demonstration.

Keywords: Blockchain,pollution data,C&C channel,botnet,Ethereum

1. はじめに

近年，情報通信技術の発展に伴い情報システムの新しいプラットフォームとしてブロックチェーンの普及が広まってきている．ブロックチェーンは 2008 年にサトシ・ナカモトによって提案された論文 [1] の中で，Bitcoin などの暗号資産の基盤技術として開発された．ブロックチェーンは当初はトランザクションを介した暗号資産の取引内容を保存しておくことを目的とした分散型台帳として用いられていた．これがいわゆる Blockchain1.0 時代と言える [2]．その後，ブロックチェーンにはスマートコントラクトが導入され，ブロックチェーン上で動く様々なアプリケーションや情報システムの開発が可能となった．スマートコントラクトではブロックチェーンの分散性やコンセンサスア

[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] 情報処理通信機構
National Institute of Information and Communications Technology, Japan
[a)] s1913551@s.tsukuba.ac.jp

ルゴリズムを応用し，信頼されていないユーザ同士が第三者信頼機関を介することなく互いにデータをやりとりしたり，トランザクションを送信したりすることが可能である．ブロックチェーンを用いた分散アプリケーションはDApps (Decentralized Application) と呼ばれ，中央管理者を必要としない運営が可能となっている．また，ブロックチェーン技術はその特性を活かし，今では医療や IoT，ソフトウェア開発などの様々な分野においてその基盤を担っている．このように，スマートコントラクトの登場によるDApps の普及が Blockchain2.0 時代 [2] であると言える．

このようなスマートコントラクトをサポートしている暗号資産プラットフォームとして最も有名なものに Ethereum が挙げられる．Ethereum は Solidity などのプログラミング言語を用いて任意のスマートコントラクトを開発することが出来る．さらに，スマートコントラクトのコードなどを埋め込むために，Ethereum のトランザクションには任意のデータを埋め込める領域が存在している．このように Ethereum はユーザに対し自由度の高いスマートコントラクトの開発を可能にしている．

このような自由度の高いスマートコントラクトを開発出来る利点は，攻撃者に対しても有利に働く．ブロックチェーンやスマートコントラクトの普及が広まるにつれて，それを悪用した新しい攻撃手法が発見されている．その一つとして，ブロックチェーンの汚染問題 [3], [4] が挙げられる．先述の通り，トランザクションには任意のデータを埋め込むことが出来る領域が存在する．その領域を悪用して，ブロックチェーンに悪性データを埋め込むことが可能である．佐藤らの研究 [3] では，Ethereum ブロックチェーンに対して調査を行い，ブロックチェーン内に実際に悪性データが格納されていることを確認している．このようなブロックチェーンに対する悪性ファイルの埋め込みは一種のポイズニング攻撃として位置付けることが可能であり，ブロックチェーンの改ざん不可能性などを踏まえると，従来の公開データベースへのポイズニング攻撃に比べて対応が困難であるということも述べられている．

ブロックチェーンに対するポイズニングは，ブロックチェーン内の被害だけに収まらない．ブロックチェーン技術を利用したマルウェアの存在や，ボットネットの C&C (Command&Control) チャネルとしてブロックチェーンを悪用する手法などが既に提案されている [5], [6], [7], [8], [9], [10]．ブロックチェーンに対してポイズニング攻撃が可能であることに加え，分散性質からくるシステムダウン耐性や，耐改ざん性がボットネットやマルウェアに対し従来を超えるテイクダウン耐性の実現や，さらに堅牢な C&C チャネルの構成を可能にし，より進化したマルウェアやボットネットの構築を可能にしている．このようにブロックチェーンへのポイズニング攻撃は，ポイズニングの被害だけに収まらず，ボットネットやマルウェアといったさらなるリスクへの入り口となりうる．

そこで本研究では，ブロックチェーンのポイズニングによるボットネットの実現方法，及びそこから始まる攻撃シナリオについて検討する．ブロックチェーンを基盤としたボットネットが具体的にどのようにブロックチェーンを悪用するのかについて考察する．ブロックチェーンがポイズニングされたときに，それを攻撃に悪用することはどの程度容易であるのか，どのようなアプローチで実現できるのかについて，テスト環境で実際に検証しながら考察した．結果として実行環境に依存することなく，容易に汚染データの格納，取り出しが可能であることを明らかにした．これは攻撃者がブロックチェーンを悪用して攻撃を行うことのハードルが低いことを示している．さらに今回想定した攻撃シナリオに対する基本的な対策についても検討した．今回行なった検証から，ブロックチェーンへの汚染データの格納だけでなく，取り出しの部分に対しても対策を講じる必要があることが明らかになった．本論文の貢献は以下である．

- ブロックチェーンの汚染がどのような被害に繋がるのかについての具体的な 4 つのシナリオを整理した．
- ブロックチェーンに格納された悪性データを取得・悪用するのがどの程度容易であるかを示し，ポイズニングが引き起こす攻撃のリスクが大きいことを示した．
- ポイズニングが発生した後のフィルタリング対策ついて基本手法とその限界を示した．

2. 関連知識

2.1 Ethereum のデータ領域について

Ethereum は現在スマートコントラクトをサポートしている暗号資産プラットフォームで一番広く用いられているものである．Ethereum では ether という暗号資産が用いられる．Ethereum ではプログラミングにより自由にスマートコントラクトを開発しデプロイすることが出来る．Ethereum には EOA (Externally Own Account) と CA (Contract Account) という 2 つのアカウントが存在する．EOA は秘密鍵によって管理されていて，対応する秘密鍵を持つユーザはその EOA を用いて ether のやり取りやスマートコントラクトの生成，実行などを行うことが出来る．CA はコントラクトのコードを保持していて，コントラクトの実行時に用いられる．スマートコントラクトは EVM (Ethereum Virtual Machine) という仮想環境上で実行される．スマートコントラクトは Solidity と呼ばれる専用のプログラム言語を用いて実装されるが，この生成されたコードは EVM 上で実行できるバイトコードの形式にコンパイルされて実行される．こうしてコントラクトを実装してデプロイすることによって CA が生成され，以降この CA のアドレスを用いてコントラクトにアクセスし，実行していくことになる．

Bitcoin や，Ethereum のトランザクションには任意のデータを格納できる領域が存在する．Matzutt らの研究 [4] では Bitcoin のブロックチェーンに対して任意データの埋め込みが可能であることが示された．そして実際に Bitcoin ブロックチェーンに悪性なデータが埋め込まれていたことが報告されている．しかし，それらはデータを埋め込むことを意図されていない領域へ埋め込まれていた．通常 Bitcoin においてデータを埋め込める領域として意図されているのは，最大 80byte の OP RETURN 領域のみである．それ以外の領域にデータが埋め込まれているトランザクションについては，悪意あるトランザクションであると判断し承認しないようにするなどの対策を行うことが可能であるとされている．

Ethereum のトランザクションにはスマートコントラクトのコードなどに使用される任意のデータ埋め込み領域が存在し，その容量は Bitcoin よりも大きく，数百 kB 程度である．このことから，Ethereum は Bitcoin よりもポイズニング攻撃を行う上で有利であることが考えられる．Ethereum の任意データ領域は ExtraData 領域と init/data 領域の 2 つが存在する．ExtraData 領域とは Ethereum ブロックのヘッダに存在し，任意のバイト列を指定することが出来る．しかしこの領域はブロックのヘッダに存在するため，データを埋め込めるのはそのブロックをマイニングしたマイナーのみである．また，データサイズも最大 32byte という上限が存在する．コントラクトの生成に使用する Contract Creation Transaction には EVM で実行するバイトコードが含まれ，このバイトコードを含む領域が init 領域となる．また，コントラクトを実行するために EOA から CA にトランザクションを送信する時などに，データを埋め込む領域が data 領域と呼ばれる．これらの領域には ExtraData 領域のようなサイズの上限は存在しない．これらは同様の性質を持っていて，1 つのトランザクションにはこれら 2 つのどちらかしか含まれないので，これら 2 つの領域は本研究ではまとめて考える．今回ポイズニングに用いられるリスクが高いのはマイナー以外でも利用することが出来る init/data 領域であると考えられるため，これを中心に議論する．

2.2 Ethereum へのポイズニング攻撃について

前項で述べたように，Ethereum トランザクションにはユーザが任意のデータを埋め込むことが出来る領域が存在する．この領域は通常ではトランザクションに必要なデータなどが格納されるが，攻撃者がブロックチェーンを汚染するために悪用することも十分想定される．佐藤らの研究 [3] では，実際に Ethereum ブロックチェーンの ExtraData 領域と init/data 領域に対して，どのようなデータが含まれているかの調査を行なっている．その結果，調査範囲の Height 0〜4,230,740 のブロックに含まれるトランザクションに対し 77 のファイルが格納されていたことが判明した．それらのファイルの内容については大部分が画像ファイルで占められており，多くは違法性のないものであったが，中にはプライバシーを侵害するような悪性な画像データも存在した．さらに 3 つの exe ファイルが発見されたと示されており，それらのハッシュ値をオンラインのマルウェアスキャンサービスである VirusTotal[*1] で検索した結果，検出率が高くマルウェアである可能性が高いと判明した．このように，ブロックチェーンに対してトランザクションを介して悪性ファイルやマルウェアを格納することが可能であり，既に実際に攻撃が行われているということが先行研究からも判明している．この事実は，ブロックチェーンポイズニングに対して早急に対策を練る必要があることを示唆している．

3. 既存研究

3.1 ブロックチェーンに関連したマルウェア

この章では関連する既存研究について説明する．近年マルウェアやボットネットなどは進化し続けており，最近の研究ではブロックチェーン技術を悪用したマルウェアに関するリスクが多く提案されている．ブロックチェーン技術を応用したマルウェアに関する研究において代表的なものとして Ali らの研究 [5] がある．この研究はボットネットの C&C 通信にブロックチェーンを用いるという提案を行なっている最初の研究である．この研究では Bitcoin のトランザクションを用いてコマンドを伝搬する方法について 4 つの方法を示している．ボットネットとはマルウェアに感染したマシンによるネットワークであり，ボットマスターからの命令を受け取って実行する．このボットネットにとって重要となるのが，各ボットマシンに命令を転送するための C&C チャネルである．多くの場合ボットネットはこの C&C チャネルが発見され，乗っ取られることで破壊される．つまり，いかに C&C チャネルを秘匿し，破壊されにくくするかということがボットネットにとって課題となる．そのような背景からも，ボットネットの C&C チャネルをブロックチェーン関連の技術を用いて構築するという研究が盛んに行われている．

Majid らの研究 [6] では，Ethereum のスマートコントラクトとして C&C チャネルを構成するという提案がされている．C&C 通信に必要なロジックをスマートコントラクトとしてプログラミングし，ボットマスターのアドレスやコマンドを変数に持ち，スマートコントラクトからのトランザクション発行という形で通信を行っている．この手法の利点としては，スマートコントラクトとして構成するためテイクダウンのリスクが低く，ボットマスターからボットへの通信だけでなく，ボットからボットマスターへの

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

アップストリーム通信もトランザクションを用いて実現できるという，双方向通信が可能である点が挙げられる．しかし，ボットネットに参加するボットがトランザクションを発行する必要があるため，トランザクション発行コストがかかるという問題が挙げられる．

Swenny らの研究 [7] では，プライベートブロックチェーンのスマートコントラクトとして C&C チャネルを実現している．プライベートブロックチェーンで実装することで法的機関に対するアクセス制御などが可能であるが，いくつかのボットがテイクダウンされるとプライベートブロックチェーンを維持できなくなる危険性がある． Pletinckx らの研究 [8] では，Bitcoin のトランザクションに C&C サーバに接続するための情報を隠しておき，各ボットはそれを見て Tor ネットワークに隠れている C&C サーバに接続するという方法が提案されている．この手法の利点として，従来までのような DGA を用いた C&C サーバ発見手法と異なり，ネットワーク上に不自然な痕跡が残らないという点が挙げられる．Baden ら [9] は，Ethereum の Whisper というプロトコルを用いて C&C 通信を行う手法を提案している．他にも様々な通信手法を持ったボットネットやマルウェアが提案されていて，ブロックチェーン技術が攻撃者に対しても有益なものになっているという事実が確認できる [10] ．これらの提案の共通事項として，ブロックチェーン技術を用いてボットネットを構成することで，従来と比較し，さらに堅牢でテイクダウンの危険性が低いボットネットを構築可能になっているという点が挙げられる．ブロックチェーンの特性はボットネットをより強固なものにするためにかなり有利に働くことが分かる．

3.2 ブロックチェーンへの攻撃

ブロックチェーンの悪用によるボットネット構成についてのリスクも重大だが，ブロックチェーン自体への攻撃についてもその対策を検討する必要がある．Ahmed らの研究 [11] では，マイニングプールに参加しているマイナーに対して攻撃を行うことで善良なマイナーにペナルティを与え，プール全体のマイニング速度を低下させるような攻撃が提案されている．他にもブロックチェーンへのポイズニングは，プライバシーの侵害につながるケースや，特定のサービスなどに対するネガティブなイメージを与えるような情報を埋め込み，サービスを妨害する攻撃 (DoS 攻撃) などに繋がるケースも考えられる．当然ブロックチェーン内に埋め込まれたデータを意図せずにダウンロードしてしまい，マルウェアに感染する危険性も考えられる．Cheng らの研究 [12] では，Ethereum のノードをハニーポットとして設置し，6 ヶ月の間ハニーポットへの攻撃を観測した．結果，1072 の異なる IP アドレスから 308 ミリオン以上のリクエストを観測したと述べられている．またこの研究から，ブロックチェーンネットワーク上には RPC ポートが何のアクセス制限もなく解放されている脆弱な Ethereum ノードが存在していることも示されていて，攻撃対象となりうるノードは多く存在していることが確認されている．

4. ブロックチェーン汚染によるリスクについて

4.1 想定する攻撃シナリオ

ここまで述べてきた通り，ブロックチェーンを悪用したボットネットの構築に関するリスクは数多く提案されており，実際にブロックチェーンは汚染されているということが先行研究から分かっている．そこで本研究では，ブロックチェーンの汚染を介した攻撃について一連のシナリオとしてまとめた．先行研究でブロックチェーンに悪性データを格納することが可能であることは分かっているので，本研究では，どのような想定で攻撃者がどのようなデータを格納するのか，そしてどのように取り出せるのかについて，一連のシナリオに沿って考察する．

今回はある不正プログラム A があるコンピュータに感染し，そして異なる不正プログラム B がブロックチェーンに格納されているとし，不正プログラム A はブロックチェーン内の不正プログラム B を取り出したいというシナリオを想定する．不正プログラム A は一般的なマルウェアであったり，ボットネットにおける感染ボットだと考えられる．それに対し不正プログラム B はボットマスターからの命令コマンドであったり，ダウンロードすることで感染するマルウェアであると想定する．今回はより具体的にシナリオを考えるため，不正プログラム A をボットネットに所属する各ボット，不正プログラム B をボットマスターが配布するボットへの命令コマンドと仮定して進める．この想定は既存研究の章でも述べたように，ブロックチェーンを基盤としたボットネットの構築について様々なアプローチが先行研究において提案されているため，最も考慮すべきリスクであると考えたためである．図 1 には今回提案する攻撃シナリオの概要について示している．ボットマスターは各ボットに配布する命令コマンドをトランザクションを介してブロックチェーンに格納する．その際にトランザクションの init/data 領域に格納するために HEX データ化する．そして各ボットはブロックチェーンに記載されたトランザクション情報から，init/data 領域に格納されている命令コマンドの HEX データを JSON-RPC 経由で取得する．そして取得した HEX データを復元することで命令コマンドを入手し，実行する．ボットは，ブロックチェーンに格納されている命令コマンドを入手するためにいくつかの手段を取る．今回は不正プログラム A が感染した先のマシンの環境をいくつかのパターンに分類して，それぞれについて不正データの取得方法を実証と共に提案する．

まずブロックチェーンには全世界に公開され，共通で使用されているパブリックブロックチェーンと，特定の参加

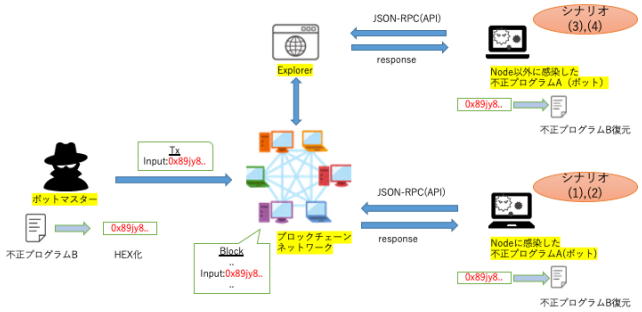


図 1　攻撃シナリオの概要

表 1　今回想定する実行環境のパターン

想定パターン	プライベート	パブリック
ノード	攻撃シナリオ (1)	攻撃シナリオ (2)
ノードでない	攻撃シナリオ (3)	攻撃シナリオ (4)

者の間で運営され，非公開となっているプライベートブロックチェーンの 2 種類がある．今回はパブリックブロックチェーンとプライベートブロックチェーンのそれぞれについて攻撃シナリオを想定した．今回パブリックチェーンだけでなく，プライベートチェーンについても考える理由としては，将来的にブロックチェーンの活躍の場は増え，特定の組織内で運用されるシステムやサービスと言った形でのプライベートブロックチェーンの普及も増加すると考えられるためである．また，攻撃者の実行環境についても複数考慮した．理由としては攻撃の容易性を示すに当たって，特定のアプリケーションや環境に依存することがないことを確認するためである．今回はネットワークを構築し，マイニングを行なったりブロックチェーンとやりとりができる Ethereum ノード (geth ,Parity など) がインストールされたマシンとインストールされていないマシンの 2 つの仮定を考えた．つまり，攻撃者の実行環境がパブリックまたはプライベートの 2 パターン，さらにその環境において，感染先がノードであるか否かという 2 パターンの合計 4 パターンの攻撃者の実行環境が考えられる．従って今回提案したシナリオについて，4 つの実行環境で実証を行った．プライベートブロックチェーンにおいてノードに感染している場合を攻撃シナリオ (1)，パブリックブロックチェーンにおいてノードに感染している場合を攻撃シナリオ (2)，プライベートブロックチェーンにおいてノードでないマシンに感染している場合を攻撃シナリオ (3)，パブリックブロックチェーンにおいてノードでないマシンに感染している場合を攻撃シナリオ (4) としている．表 1 に今回想定する 4 つの実行環境についてまとめる．

4.2　攻撃シナリオの実証

4.2.1　攻撃シナリオ (1)

不正プログラム A がプライベートブロックチェーンに接続している Ethereum ノードに感染している場合に，ブロックチェーンに格納されている不正プログラム B を取得する方法について検討する．Ethereum ノード (geth や Parity など) に感染した場合は，ノードの提供する機能である RPC モードを用いて自身を RPC サーバとして起動することでブロックチェーンとやり取りすることが可能である．今回はコマンドライン上から Ethereum ノードである geth を RPC モードで起動し，さらに別のコマンドウィンドウから JSON-RPC のリクエストを HTTP リクエストとして geth に対して送信した．今回は curl コマンドを用いてリクエストを送信し，結果として geth を介してレスポンスを得ることができた．使用した curl コマンドを以下に示す．

```
curl -X POST http://localhost:8545/
-H "Content-type:application/json"
--data '{"jsonrpc":"2.0","method":
"eth_getTransactionByHash","params":[
"0x...(TXhash)"],"id":1}'
```

このように，ノードを RPC モードで立ち上げることによって簡単にブロックチェーンから目的のデータを取得することが可能である．

4.2.2　攻撃シナリオ (2)

不正プログラム A がパブリックブロックチェーンに接続している Ethereum ノードに感染している場合に，ブロックチェーンに格納されている不正プログラム B を取得する方法について検討する．まず Ethereum には，全世界で共通に使われ公開されているメインネットの他に，開発したスマートコントラクトのテストなどに使用するためのテストネット，限られた者のみが参加している非公開のプライベートネットが存在する．今回はパブリックブロックチェーンにおける検証として，Ethereum のテストネットの一つである Rinkeby において検証を行なった．今回はテストネットを用いているが，メインネットでも同じ結果を得ることが出来る．この場合も，Ethereum ノードである geth を RPC モードで起動することで RPC サーバとして機能させることが出来るので，別のコマンドウィンドウから RPC サーバとして起動した geth に向けて HTTP リクエストを送信することでレスポンスを得ることが出来た．使用した curl コマンドを以下に示す．

```
curl -X POST http://localhost:8545/
-H "Content-type:application/json"
--data '{"jsonrpc":"2.0","method":
"eth_getTransactionByHash","params":[
"0x...(TXhash)"],"id":1}'
```

このように，パブリックブロックチェーンの場合も同様にノードを介して HTTP リクエストを送信することにより，コマンドライン上で容易にブロックチェーン内のデー

タを取得することが可能である．

4.2.3 攻撃シナリオ (3)

不正プログラム A がノードでないマシンに感染している場合に，プライベートブロックチェーンから不正プログラム B を取得する方法を検討する．ノードでないマシンとは，geth などのノードソフトウェアがインストールされていないマシンを意味する．この場合は Explorer という，ブロックチェーン内のトランザクション情報を公開している web サイトからデータを取得することが出来る．所属しているプライベートネットにおいて Explorer が存在している場合は，その web サイトからブロックチェーン内のデータを取得することが出来る．

4.2.4 攻撃シナリオ (4)

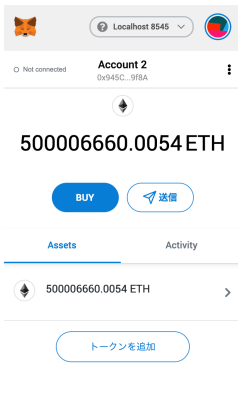
不正プログラム A がノードでないマシンに感染している場合に，パブリックブロックチェーンから不正プログラム B を取得する方法を検討する．今回はパブリックブロックチェーンとして，Ethereum のテストネットの一つである Rinkeby を使用した．この場合も，Explorer を介してデータを取得することが出来る．Ethereum の Explorer として Etherscan[*2]という Explorer が存在し，メインネットだけでなく，Rinkeby などのテストネットのブロックチェーンの情報についても閲覧することが出来る．また Etherscan には API が用意されており，会員登録を行うことで API キーを取得でき，HTTP リクエスト経由で API を使用することでコマンドライン上でデータを取得することも可能である．以下は Etherscan の API を用いてデータを取得する際に使用した curl コマンドを示している．

```
curl 'https://api-rinkeby.etherscan.io/api
?module=proxy&action=
eth_getTransactionByHash&txhash=
"0x...(TXhash)"&apikey=XXXXXX(your API Key)
```

このように，Etherscan を用いて web ブラウザ上から，あるいは API を用いてコマンド上から簡単に情報を取得することが出来る．

ここまで示してきた 4 つのシナリオのいずれも，curl コマンドによる HTTP 通信を介して簡単にブロックチェーンの情報を取得することが出来た．今回の実証で重要なことは，特別なソフトウェアや権限などを必要とせず実行出来ている点である．今回前述の 4 パターンの想定を用意し実証を行うことで，マルウェアが感染した先の環境に依存することなく攻撃が容易に可能であるということを示した．結果として簡単なコマンドのみでデータの取得を行えているため，感染先の環境に依存せず，ブロックチェーンを C&C チャネルとして用いることが可能であると言える．

*2 Etherscan: https://etherscan.io/



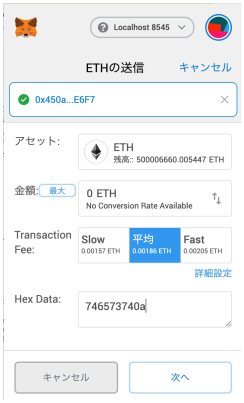


図 2 プライベートネット接続　図 3 データ埋め込み

5. ブロックチェーン悪用についての基本的な対策手法

5.1 ブロックチェーンへの汚染データの格納について

ブロックチェーンへの汚染データの格納については，佐藤らの研究 [3] において容易に行えることが示されている．今回の実証についてもこの先行研究に倣ってデータの格納を行なった．今回はプライベートネットを構築し，そこに Ethereum ウォレットの一つである METAMASK を接続した．そして METAMASK を通して送金を行う際に，オプションとして init/data 領域に hex データを挿入した．今回は全世界で使用されているパブリックチェーンに影響を及ぼすのは望ましくないと考え，汚染データの格納から対策の実証についてはプライベートブロックチェーンで行った．以下にその手順について示す．

(1) METAMASK を用いてプライベートネットワークに接続．(図 2)
(2) 用意したアカウント同士で送金を行う．その際に，option 部分にあるデータ領域に hex データを挿入する (図 3)．
(3) 送金を行う．
(4) Explorer[*3]から先ほど送金されたトランザクションを確認する．挿入された hex データも確認できる (図 4)．

なお図 4 にはプライベートネット用の Explorer において悪性データがトランザクション内に格納されていることを確認できる様子を示している．

5.2 ブラックリストによる汚染データのフィルタリング

前章の実証から，ブロックチェーンの汚染は更なる攻撃へと繋がる危険性が提示された．ブロックチェーンに対し悪性データを格納すること，及びその格納された悪性データを取り出すことは容易であり，攻撃者にとってブロックチェーンはかなり利用しやすい攻撃対象となっている．佐藤らの研究 [3] ではブロックチェーンのポイズニング攻撃を防ぐために，コントラクトの作成に関する部分について対

*3 https://github.com/carsenk/explorer

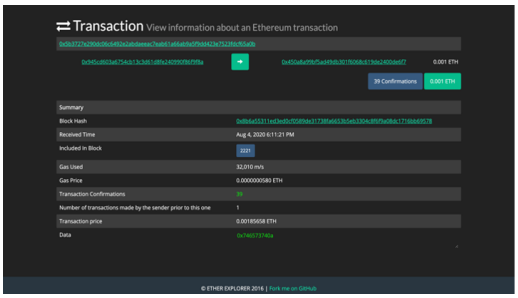


図 4　Explorer により確認

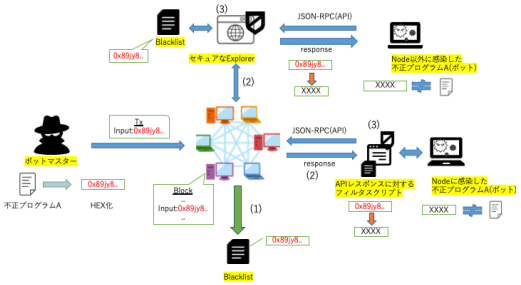


図 5　ブラックリストによるフィルタリングの流れ

策を提案している．しかし，現状においてブロックチェーンのポイズニングが既に起こっている点，ブロックチェーンに格納されたデータは容易に取り出すことが出来る点などを踏まえると，汚染データの格納に対してだけでなく，取り出しについても対策を討つ必要があると言える．

そこで本論文では，ブロックチェーンに格納された汚染データの取り出しを制限するような対策を提案する．今回提案するのはトランザクションハッシュに対するブラックリストを用いたフィルタリング機能である．まず，ブロックチェーンの中に含まれているデータの中で，実際に汚染データがどれであるかは事前に明らかであると仮定する．これは佐藤らの研究 [3] において行われた調査の中で，Ethereum トランザクション内のデータを VirusTotal にかけることでマルウェアである可能性が高いと判断しているように，トランザクション内のデータが既出のウィルスであるかどうかはある程度把握できると考えられるためである．この事実を利用し，Ethereum トランザクション内のデータを調査することにより，悪性データが格納されたトランザクションをリストアップすることが出来る．そうして悪性データのブラックリストを作成することでフィルタリングを行うことが可能になる．

先行研究ではボットマスターと思われるアドレスをブラックリスト化することで C&C 通信を防ぐという対策が提案されている [5]．しかし Ethereum などの暗号資産のアドレスは誰でも簡単に幾つでも作成することが出来，アドレスは本人と直接結びつくものではない．従ってボットマスターにアドレスを次々に新規作成されれば，アドレスによるフィルタリングでは C&C 通信を防げない．これに対して今回はトランザクションによるフィルタリングを行っている．従ってフィルタリングされたトランザクション内の悪性データに関しては，どのアドレスであろうとアクセス出来ないので，確実に C&C 通信を防ぐことが出来る．

提案する対策の具体的な流れを図 5 に示す．まずブロックチェーンに格納されているトランザクション情報から，トランザクションの init/data 領域に格納されているデータを確認することが可能である．図 5 の手順 (1) において，init/data 領域のデータを VirusTotal などに登録されているマルウェアと比較して一致したものや，実際に汚染データであると判明しているものについてブラックリストとしてリスト化する．そして作成したブラックリストを用いてフィルタリングする機能をセキュアな Explorer や API を実行するコマンドのラッパースクリプトとして実装する．そうして実装したセキュアな Explorer やラッパースクリプトを介して手順 (2) のように API を用いてリクエストを送信しレスポンスを受け取ると，フィルタリング機能によりレスポンス内の init/data 領域がリストアップされた悪性データと一致する場合は，手順 (3) においてレスポンスを書き換えそれをマスクする．このようにユーザの元に悪性データが渡るのを防ぐ．

今回の研究では，シェルスクリプトを用いてこのようなフィルタリング機能を簡易的に実装した．しかしトランザクションのフィルタリングはブロックチェーンコミュニティ全体に適応すると，自由度を下げてしまいコミュニティに影響を与えることになる．今回提案したフィルタリング手法についても，全てのユーザに適応されるように，ブロックチェーンのデータを取得する API に組み込むことが出来れば理想であるが，既に世界中で運用されているパブリックブロックチェーンの仕組みを大幅に変更することは現実的に難しい．従って，よりセキュアなオプションという位置付けで作成し，善良なユーザはこのフィルタリングを用いて JSON-RPC を用いることで攻撃を受けるリスクを削減するという形での普及が現実的であると考えられる．

5.3　対策手法の限界

提案した対策手法にはフィルタリングを挟むことによってデータ取得 API の実行時間が遅くなってしまうという問題がある．今回は実証として簡易的なフィルタリングを行なった．ブラックリストに記載するトランザクションハッシュの数を増やしていって，実行にかかる時間を計測した．その結果を図 6 に示している．図からも分かる通り，ファイル数が増えると実行時間は増加していき，後半は指数的に実行時間が増加している．今回の実証では 10 万件が最大だが，現在 Ethereum の Rinkeby テストネットには 2020 年 8 月 17 日時点で 59994516 件のトランザクションが存在している．しかし現状では悪性データはブロックチェーン

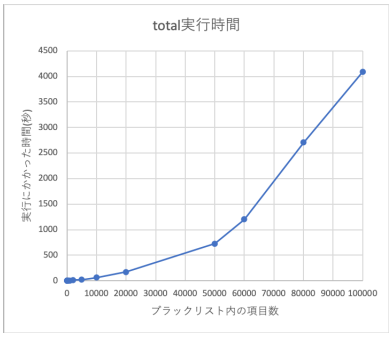


図 6 フィルタリングの件数と実行時間の関係

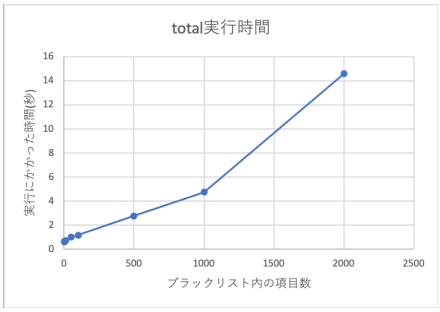


図 7 フィルタリングの件数と実行時間の関係拡大図

の内のごく少数である．図 7 には図 6 におけるトランザクション数が 2000 件までの部分を拡大表示した．この図から分かるように，フィルタリングの実行時間が 1 秒以内の範囲では約 50 件，10 秒以内の範囲では約 1500 件程度のブラックリストサイズのフィルタリングが可能である．つまり，悪性トランザクションの中でもより致命的なものに絞ってフィルタリングすれば件数は少なくなり，実行時間もほとんど掛からずフィルタリング出来ると考えられる．しかし悪性トランザクションは増加していくと考えられるので，実行時間が指数的に増える本手法はいずれ限界が来ると考えられる．従ってより効率的な対策が必要である．このように提案したフィルタリングによる対策は，トランザクション内の悪性データ取得を制限することが可能であるが，ユーザビリティを著しく低下させる危険性がある．また，コミュニティの自由度を下げることになるため，このような対策を全てのユーザに強制することは出来ない．今後ブロックチェーンの汚染による攻撃の脅威が大きくなっていく中で，先行研究で提案された汚染を防ぐようないわゆる入口対策に加え，今回提案した汚染データの取り出しの対策のような出口対策も検討していく必要がある．

6. まとめ

今回の研究では，ブロックチェーンの汚染により攻撃に用いられるリスクを具体的なシナリオを提示することで示した．ブロックチェーンの汚染が具体的にどのような攻撃に使用されうるのかについて詳細に検討し，4 つの場合分けを行なった．そして 4 つのシナリオ全てにおいて，特別なソフトウェアや権限なしに実行環境に依存することなくブロックチェーンから悪性データを取得出来ることを示した．このようにブロックチェーンに格納された汚染データを取り出すのが容易であることを実証とともに示すことで，ブロックチェーン汚染によるリスクの大きさを示した．また本研究ではこのようなポイズニングの脅威に対する基本的な対策とその限界を示した．今後のブロックチェーンのポイズニングによる被害の拡大は大きな問題であるので，有効な対策を迅速に検討する必要がある．

参考文献

[1] S. Nakamoto, 2012. ”Bitcoin: A peer-to-peer electronic cash system” https://bitcoin.org/bitcoin.pdf, Oct,2008.

[2] X. Li, P. Jiang, T. Chen, X. Luo, Q. Wen, ”A survey on the security of blockchain systems,” Future Gener. Comput. Syst. (2017). http://dx.doi.org/10.1016/j. future.2017.08.020.

[3] T. Sato, M. Imamura, K. Omote, ”Threat Analysis of Poisoning Attack against Ethereum Blockchain,”Proc. WISTP2019,pp.139–154,Dec.2019.

[4] R. Matzutt, M. Henze, J. H. Ziegeldorf, J. Hiller,and K. Wehrle, ”Thwarting Unwanted Blockchain Content Insertion,” 2018 IEEE International Conference on Cloud Engineering (IC2E), pp.364–370,April 2018

[5] S. T. Ali, P. McCorry, P. H.-J. Lee, and F.Hao,”Zombiecoin: Powering Next-Generation Botnets with Bitcoin,”Financial Cryptography and DataSecurity, eds. by M. Brenner, N. Christin, B. Johnson, and K. Rohloff, pp.34-48, Springer Berlin-Heidelberg, Berlin, Heidelberg, 2015

[6] N. A. I. Majid A. Malaika,Omar Al Ibrahim,”Botract: Abusing Smart Contracts and Blockchains for Botnet Command and Control,”https://www.omprotect.com/wp-content/uploads/2017/12/BotDraftPapev1.pdf,2017 ,accessed: 2019-02-13

[7] J. Sweeny,”Botnet resiliency via private blockchains,” SANS Institute Information Security Reading Group, 2017,https://www.sans.org/reading-room/whitepapers/covert/paper/38050

[8] S. Pletinckx, C. Trap, and C. Doerr, ”Malware coordination using the blockchain: An analysis of the cerber ransomware,” in 2018 IEEE Conference on Communications and Network Security, CNS 2018, Beijing,China, May 30 - June 1, 2018. IEEE, 2018, pp. 1–9.

[9] M .Baden, C. F. Torres, B. B. F. Pontivero, R. State: ”Whispering botnet command and control instructions,” In: 2019 Crypto Valley Conference on Blockchain Technology (CVCBT). pp. 77–81. IEEE (2019)

[10] L. Bock, N. Alexopoulos, E. Saracoglu, M. Muhlhauser, E. Vasilomanolakis ”Assessing the Threat of Blockchain-based Botnets, in 2019 APWG Symposium on Electronic Crime Research(eCrime),Pittsburgh,PA,USA,USA,13–15 Nov. 2019

[11] M. Ahmed, J. Wei, Y. Wang, E. Al-Shaer ”A Poisoning Attack Against Cryptocurrency Mining Pools / Lecture Notes in Com-puter Science 11025, 2018, pp. 140–154.

[12] Z. Cheng, X. Hou, R. Li, Y. Zhou, X. Luo, J. Li, and K. Ren, ”Towards a first step to understand the cryptocurrency stealing attack on ethereum,”in Proc. 22nd Int. Symp. Res. Attacks, Intrusions Defenses, 2019, pp. 47–60.