

動的な解析を用いたマイニングスクリプト検知手法の提案

Proposal of detection for mining script using dynamics analysis

飯田良[†] 猪俣敦夫[†] 柿崎淑郎[†] 廣瀬幸[†]

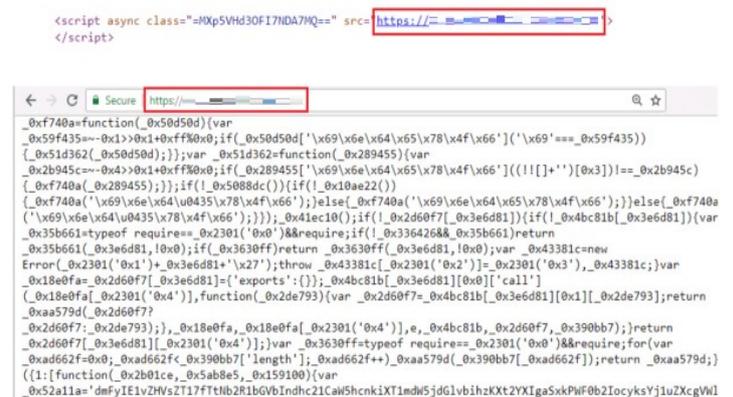
概要 マイニングスクリプトを Web ブラウザ上に用いることで、広告収入の代替とする使用法が普及し始めている。一方で、この技術を悪用した攻撃が台頭している。また、これらに対する対策としてブラックリストを用いて遮断する方法があるが、ソースコードの難読化などの点から十分な手法とは言い難い。そこで本研究では、CPU の時間ごとの使用率に着目し検知を試みる。実験の結果、対象サイトがどの程度 CPU のリソースを消費しているかが分かった。

キーワード Cryptojacking, CryptoMining, JavaScript

1. はじめに

法定通貨に代わる決済や送金的手段として、仮想通貨が使用されている。これらは、既存の通貨とは異なり、取引時間に制約が少ないことや、送金時にかかる手数料の低額さが利点となっている。仮想通貨を入手する方法として、法定通貨との換金のほかにマイニングと呼ばれる手法が存在している。従来ではマイニングを行うために処理能力が高い PC が必要とされてきた。しかしながら、近年、一般的な PC の処理能力でマイニングが可能になる仮想通貨が登場している。さらに、Web ページ上でマイニングを行うツールとしてマイニングスクリプトも登場し始めている。このマイニングスクリプトが様々な分野で用いられている。例えば Web ページでの収入源としての利用である。従来 Web ページはオンライン広告を用いて収入を得ていた。しかしこれだとユーザビリティの損失が問題になっていた。そこで、ユーザビリティを兼ねた持続的なコンテンツの提供手段としてマイニングスクリプトの使用が普及し始めている。これにより、Web 製作者が特定の Web サイトにマイニングスクリプトを埋め込むことでマイニングが可能となる手軽さが利点である。一方で、これらの技術を悪用した攻撃手法が台頭している。この攻撃手法は Cryptojacking と呼ばれている。Cryptojacking に対する対策方法としては、ブラックリストを用いた静的解析などが挙げられる。しかしこの手法だけで Cryptojacking を完全に遮断できるとは言いがたい。最近では、オープンソースの難読化ツールを使用して、対象の Web ページが難読化されているからである。図 1 は実際に確認することができた、難読化されている Web ページの一部である。そのほかの懸念として、ブラックリストを回避する新種のマイニングスクリプトの普及も上げられる。このことから、ブラックリストを用いた静的な解析だけでは課題が残る。

そこで、本研究ではマイニングサイトの判定方法として、動的な解析手法を提案する。本提案では CPU の時間ごとの使用率に着目する。ユーザが Web ページを訪れた際に、CPU の計測を開始する。その後特定の条件を満たしたものをマイニングサイトと判定する。その後判定された Web ページの通信を解析することにより、実際にマイニングが行われているかどうかを判定する。



```
<script async class="HXp5Vhd30FI7HDA7WQ==" src="https://.../...>
</script>

..._0xf740a=function(_0x50d50d){var
_0x59f435=..._0x1>0x1+0xff%0x0;if(_0x50d50d[...])...
..._0x51d362=function(_0x289455){var
_0x2b945c=..._0x4>0x1+0xff%0x0;if(_0x289455[...])...
..._0xf740a(_0x289455);};if(!_0x5088dc()){if(!_0x10ae22())
..._0xf740a(_0x69\x6e\x64\u0435\x78\x4f\x66');}else(_0xf740a(
..._0x69\x6e\x64\u0435\x78\x4f\x66');});_0x41ec10();if(!_0x2d60f7[_0x3e6d81])if(!_0x4bc81b[_0x3e6d81])(var
_0x35b661=typeof require==_0x2301('0x0')&&require;if(!_0x33642688_0x35b661)return
_0x35b661;_0x3e6d81,0x0);if(_0x3630ff)return _0x3630ff(_0x3e6d81,0x0);var _0x43381c=new
Error(_0x2301('0x1')+_0x3e6d81+'x27');throw _0x43381c[_0x2301('0x2')]=_0x2301('0x3'),_0x43381c);var
_0x18e0fa=_0x2d60f7[_0x3e6d81]='exports':{};_0x4bc81b[_0x3e6d81][0x0]['call']
(_0x18e0fa[_0x2301('0x4')],function(_0x2de793){var _0x2d60f7=_0x4bc81b[_0x3e6d81][0x1][_0x2de793];return
_0xaa579d(_0x2d60f7?
_0x2d60f7:_0x2de793);},_0x18e0fa,_0x18e0fa[_0x2301('0x4')],e,_0x4bc81b,_0x2d60f7,_0x390bb7);return
_0x2d60f7[_0x3e6d81][_0x2301('0x4')];var _0x3630ff=typeof require==_0x2301('0x0')&&require;for(var
_0xad662f=0x0;_0xad662f<_0x390bb7['length'];_0xad662f+=_0xaa579d(_0x390bb7[_0xad662f]);return _0xaa579d;
}({1:[function(_0x2b01ce,_0x5ab8e5,_0x159100){var
_0x52a11a='dmFyIF1vZHVzT17fTtN2R1bGVbIndhc21CaW5hcnkiXT1mdW5jdGlibihzKXt2YXlIgaS5kPwF0b2IocysyYj1uZCgVWl
```

図 1 難読化されたマイニングスクリプト [1]

2. 背景

2.1 マイニング

マイニングとは、仮想通貨の分散型台帳に取引を追加するプロセスのことである [2]。また、「いつ」「誰が」「いくら取引したのか」を改ざんされることの無いように記録することが目的である。追加する取引のまとまりをブロックと呼ぶ [3]。ブロックを正しく生成するためには、特定の要件を満たすハッシュ値を添付しなければならない。このハッシュ値を求めるためには、多量の計算を行い、一致するハッシュ値を求める必要がある。そして、このハッシュ値を

[†] 東京電機大学
Tokyo Denki University

最も早く見つけた人に仮想通貨で報酬が与えられる。これにより、報酬の獲得ため、高性能な GPU や暗号化に特化した構成を用いて、短時間での数学的問題を解決する必要があった。しかし、一般的な PC の処理能力のほうがより効率的にマイニングが行える仮想通貨も登場している。

2.2 Monero

Monero とは 2014 年に Bitcoin の代替として誕生した仮想通貨である。Monero は、送金や決済時において複数人で署名するため、個人を特定することが非常に困難になる特徴が挙げられる。また、Monero には Bitcoin やほかの仮想通貨に比べて多額の取引や承認時間の高速化が可能である。仮想通貨と法定通貨の交換よりも仮想通貨同士の交換のがより高速に行えることから、Monero は Bitcoin などの別の仮想通貨に換金されている。このように Monero は、決済の即時性、決済金額や取引情報の秘匿性から、ダークウェブ市場などでも使用されている。

2.3 Coinhive

Coinhive は 2017 年 9 月に導入された、Web サイトでの使用を目的とした、仮想通貨マイニングサービスの一種である。これは、導入済みのサイトにアクセスしたユーザーの CPU の一部またはすべてを使用してマイニングを行う。これにより Web サイト管理者は Web サイトから広告を削除し、その代わりに Coinhive を実行させることで、ユーザー側には、広告非表示によるユーザービリティの向上が、管理者側は広告費の代替手段として利用することが可能になる。また、Coinhive は Monero のマイニングを行っているため、取引を追跡することが困難である特徴がある。表 1 はインターネット上の機器の情報等を取得するための検索エンジンである Censys を用いて coinhive の使用量を検索した結果である。この結果からわかるように、2017 年から 2018 年の一年間でおおよそ 20 倍以上増加していることが確認された。また、国別にみても、2017 年度ではアメリカやドイツなどの国々が top5 に入っているが、近年では、それらの国々に代わり、ブラジルやインドなどの国々で爆発的な増加が確認されている。

表 1 censys での調査結果上位 5 つ[4]

2017 年 11 月		2019 年 1 月	
アメリカ	604 件 (38.4%)	ブラジル	9416 件 (26.74%)
フランス	217 件 (13.8%)	インドネシア	5031 件 (14.29%)
ドイツ	139 件 (8.84%)	インド	4023 件 (11.43%)
オランダ	78 件 (4.96%)	アメリカ	1427 件 (4.05%)

ロシア	78 件 (4.96%)	イラン	1307 件 (3.71%)
...		...	
合計	1573 件	合計	35211 件

2.4 Cryptojacking

ユーザーの同意なしに仮想通貨のマイニングをするためにブラウザに攻撃する技術のことを Cryptojacking と呼ぶ。図 2 は、Cryptojacking の一連の流れを示している。この図では 4 段階で説明している。それぞれ、

- (1) 攻撃者が Web サイトに攻撃するし、マイニングスクリプトを挿入する。
- (2) ユーザーが被害を受けた Web サイトにアクセスする。
- (3) ユーザーに通知することなく、攻撃者に代わってマイニングを行う。
- (4) 新たなブロックがブロックチェーンに加わると、攻撃者は報酬として仮想通貨を受け取る。

の 4 つである。

図 3 は 2016 年からのマイニングシステムのマルウェアのグラフである[5]。これによると、2016 年後半から出現し始めたマイニングマルウェアは 2018 年度になり、急激に増加した。第 3 四半期までに、去年と比較して、40 倍以上に増加した。経済的な被害例としては、オープンソースの CI(Continuous Integration)ツールである” Jenkins ” が挙げられる。攻撃者が” Jenkins ” のサーバにマイニングマルウェアをインストールし、利用者に実行させていた。総額として 3 億 2000 万円程度の収入があったと考えられている[6]。また、IoT(Internet of Things)デバイスなどの処理能力の低いものに対しても攻撃の対象になり始めている。理由としては、今後ますます流通量が増加することが見込まれるため、1 台当たりが少規模のマイニングでも取得額は膨大になりうるからだと考えられる。また、IoT デバイスだけではなく、MacOS に対応したマイニングマルウェアも登場している。図 3 は国別に調査した、全てのマルウェアのうちどれだけ割合でマイニングシステムのマルウェアだったかを示す図である。これによると中東やアフリカ地域では、マイニングマルウェアがランサムウェアよりも被害が出ていることがわかる[7]。

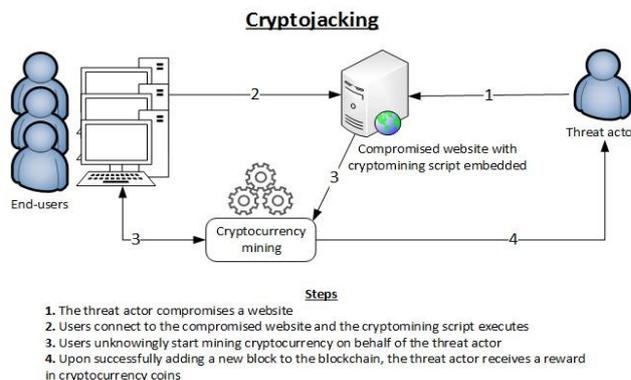


図 2 マイニングスクリプトを悪用事例[8]

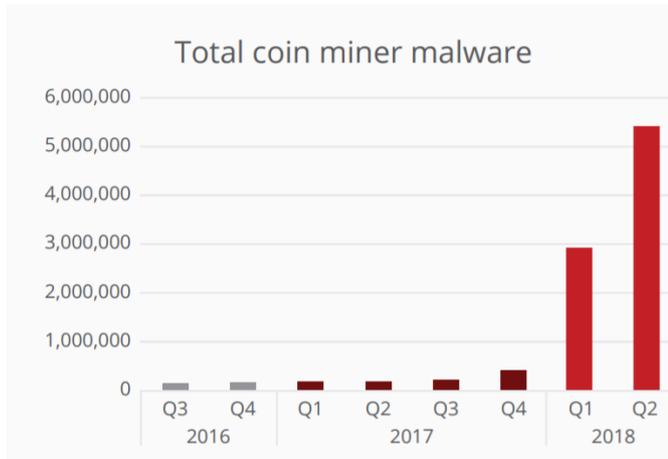


図 3 coinminer の推移

表 3 マイニングマルウェアの国別推移

国名	全マルウェアのうちマイニングが占める割合 (2016-2017)	全マルウェアのうちマイニングが占める割合 (2017-2018)
アフガニスタン	27.28%	29%
エチオピア	25.29%	31%
ウズベキスタン	24.57%	18%
タジキスタン	22.34%	21%
ザンビア	20.79%	18%
トルクメニスタン	19.71%	24%

3. 関連研究

3.1 Cryptocurrency mining

Jian らはブロックチェーンベースでの暗号通信方式について着目した。彼らはこの方式では公平性が担保されていないことに着目し、仮想通貨の支払いの領収書 (cryptocurrency payment for a receipt) との交換を公平にするための解決策を提案している[9]。また Mauro らの研究では Bitcoin に対してセキュリティとプライバシー双方を網羅する調査を行った。また、Bitcoin についての脆弱性を取り上げ、その解決案と実現可能性について調査した。そして、Bitcoin のプライバシーの脅威について紹介し、既存のプライバシー保護の解決策を説明する[10]。Yu らはブロックチェーンのセキュリティを脅かすかもしれない量子コンピューティングを用いた攻撃手法に対する対策案を提案している[11]。

しかし、これらの研究は、仮想通貨の運用等に対するセキュリティであり、今回題材としている Cryptojacking に対

しての直接的な解決法ではない。

3.2 Detecting malicious JavaScript

本研究の対象である Cryptojacking は主に JavaScript コード分析のための技術がベースとなっている。既存の研究では、悪質な JavaScript の判別に関して、静的な解析、または動的な解析が用いられている。Jsand らの研究では、4 つの異なる特徴を抽出した。彼らは drive-by-downloads-attack を行う JavaScript マルウェアサンプルを検討するために Naive Bayes アプローチを用いた。静的解析では Curtsinger らがプログラムの AST(Abstract Syntax Tree)に関連する機能から良性、悪性を判定するツールを発表した。具体的には、いくつかの固有の特徴量を利用することに着目した。Zarrasetal らは広告の安全性と悪性のコンテンツがユーザにどのように被害を与えるかについて調査した。ただし、これらのアプローチはすべてマイニングに対して直接適用できない課題がある。

3.3 Cryptojacking

Shayan らの研究では Coinhive を用いた Monero のマイニングに対しての最近の動向について調査している[12]。Jan らはマイニングサイトの特定に辿り、Fingerprinting を用いた提案を行った[13]。Geng らの研究では、Cryptojacking の 2 つの固有の特性に着目した。ひとつはハッシュであり、もう 1 つはスタックである。ブラウザ内での関数の実行時間に着目し、それぞれに設けた閾値を超えたものに対してマイニングの判定を行った[14]

これらの関連研究にも課題が見受けられる。Shayan らの研究だが、彼らは、最新の動向の調査を行ったにすぎず、具体的な検知手法の提案を行っていなかった。本研究では CPU の時間ごとの使用率を調査することにより、対象の Web サイトがどれだけ CPU を消費しているかを計測する。

4. 提案方式

4.1 目的

提案方式は、静的な解析に依存することなくマイニングサイトの判定を行うことである。また、マイニングされていた場合どの程度影響があるのかの判定をする。提案方式の目的を達成するために、動的な解析を用いた実験を試みる。具体的には CPU の時間ごとの使用率を用いて、対象の Web サイトがマイニングされているかの判定を行う。上記によって、Web ページがマイニングされている可能性があるかどうか、さらにその Web ページを訪れたときに、どの程度 CPU が消費されているかどうかの判別が可能になる。

4.2 全体像

提案方式の処理の流れを図 4 に示し、その詳細を以下で

述べる。

- (1) Web サイトからのサンプルデータ収集
- (2) 一定時間、対象の Web ページにアクセスし、CPU の使用率を計測。
- (3) 計測結果から、アクセスしたサイトがマイニングを行っているかどうか判定。条件に合致したものがあれば対象の Web サイトの URL を保存。



図 4 提案方式の処理流れ

4.3 課題

図 4 の実験を満たすための課題は以下の 4 つがあると考えた。

(課題 1) サンプルデータの収集

すべての Web ページに対して本提案手法を行うことが困難なため、一定数のデータを用いて実験を行う必要がある。

(課題 2) 計測時間の決定

Web ページにアクセスした後、CPU を計測する時間を検討する必要がある。

(課題 3) 閾値の検討

どの程度の CPU 使用率だった場合に、マイニングを行っているかを判断する閾値を検討する必要がある。

(課題 4) 判定されたものの正誤評価

実際にマイニングと思いきリストの正誤を評価する必要がある。

4.4 課題への対処

4.4.1 サンプルデータの収集

先行研究内で Geng らが提供した URL のデータリストを使用する。しかし、2019 年現在、URL のデータリストの一部がすでにアクセスできないページが一定数存在されると考えられた。そこで、本研究では、データリスト内にある各 URL に対してリクエストを送信した。そして、レスポンスとして 200 番を返したもののだけでドメインリストを再度データセットを製作した。

4.4.2 計測時間の決定

実際に Web ページにアクセスしてからにマイニングを実際に行うまでの時間差があると考えられたからである。本実験では対象の Web ページの計測時間は 15 分間とした

4.4.3 閾値の決定

閾値の決定では図 5 を参考にした。図 5 は 2018 年 8 月 2 日から 8 月 22 日の間にマイニングサイトの調査を行った結果のグラフである[15]。この調査では約 25 万サイトの中からコード上にマイニングスクリプトの文字列が現れるかどうかを判定した。そして、文字列が検出されたものに対して再度アクセスを行い Web サイトのマイニングに関する情報を取得した。結果としてマイニングされていた Web ページ 203 個検出された。この表から、設定として、一番多かったのは 100%，次いで 70% 台であることがわかった。これより本提案での閾値は 70% とする。そして、検知条件は以下 2 つとした。判定は 2 つの条件の内いずれか 1 つでも満たすものがあつた場合、その URL をログファイルに保存する。

- ① 時間ごとの CPU 使用率の最大値が 70% 以上であり、その値が 10 秒以上続いた場合
- ② 時間ごとの CPU 使用率の差分の最大値が 70% 以上の場合

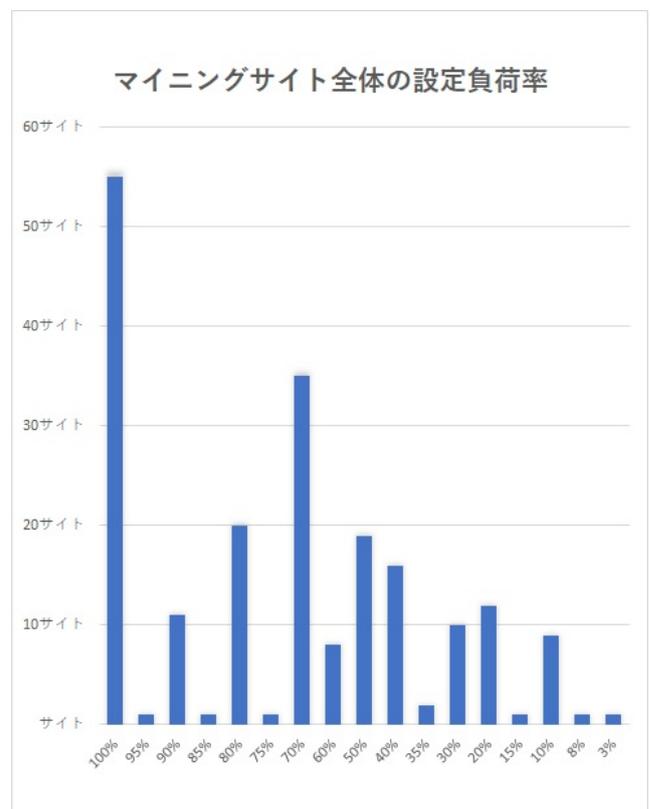


図 5 マイニングページの CPU 負荷率の調査結果

4.4.4 判定されたものの正誤評価

閾値を決定して、条件に合致したものをマイニングサイトと判定する場合、精度に影響が出ると考えられた。そこで、本提案では、条件を満たしたとき実験中に計測された CPU の使用率の最大値とその URL をログデータとして保存する。そして、取得したログデータを用いて再度対象の Web ページにアクセスを試みる。その間、通信内容を計測し、通信内容の解析結果から特定の文字列を検出した場合にマイニングサイトと判断する。本実験ではマイニングスクリプトとして広く用いられている coinhive.com のみを対象とする。

4.5 期待される効果

動的な解析を用いて、アクセスした Web ページがマイニングしているかどうかの判別を可能にする。さらに、静的な解析に依存しておらず、新種のマイニングスクリプトの登場や、難読化の影響を受けない。また、リアルタイムで CPU を計測しているので、マイニングしているかどうかだけでなく、実際にどの程度 CPU が使用されているかがわかる。

5. 実験

4.4 章で述べた課題への対処を考慮に入れて以下の実験を行う。

(1) サンプルデータの生存確認

Geng らが適用したデータから、実験開始時点までに生存している URL だけを抽出する。

(2) CPU 使用率の測定

サンプルデータの URL アクセスし、CPU の使用率を計測する。4.4.3 節で示した判定条件に合致した際には、その URL と CPU の使用率をログファイルに追記する。

(3) 通信内容の調査

4.4.4 節で述べた再検証を行うために、ログファイルに記されている URL に対して、通信内容を取得する。取得した通信内容から、文字列'coinhive.min.js'を検索する。通信内容を解析するために tshark を用いる。tshark はコマンドラインを用いて指定したネットワークインターフェイス上を通過するネットワークパケットをキャプチャして分析するツールである。尚、本実験では tshark で取得した PCAP ファイルを、CSV ファイルに変換したうえで文字列"coinhive"が存在するか判定を行った。

本実験では、異なる環境での実験を行うために、複数の PC を用いて実験を行った。表 4,5,6 は本実験で用いた実験環境である。また、本実験で用いたプログラミング言語は python3.6 である。

表 4 実験で用いた PC1

OS	Windows10,64bit
CPU	AMD®Ryzen7 2700x Eight-Core Processor 3.70GHz
メモリ	16.0GB RAM

表 5 実験で用いた PC2

OS	Windows10,64bit
CPU	AMD®Phenom(tm)II x2 550 Processor 3.10GHz
メモリ	12.0GB RAM

表 6 実験で用いた PC3

OS	Windows10,64bit
CPU	Intel®core(im) i7 2640M CPU@2.80GHZ
メモリ	8.0GB RAM

6. 結果

6.1 実験(1)

2018 年 11 月 29 日に取得を行った。結果として 2770 個のサンプルデータのから 975 個のデータサンプルを取得することに成功した。

6.2 実験(2)

2018 年 11 月 29 日～12 月 12 日までかけて(2)を行った。今回、提案手法の条件に適合したサイトは 46 個だった。また、図 6 は提案した閾値に合致したものの CPU 使用率ごとの個数のグラフである。これより、今回のデータでは CPU の消費量 100%のマイニングが一番多く、次いで 80% 台だった。しかし、本実験では 3 つの PC を用いて実験を行った、実験結果は完全には一致しなかった。最もマイニングしているとして判定していたのは表 4 の PC だった。一方で、表 6 の PC が最もマイニングかどうかを判定できなかった。

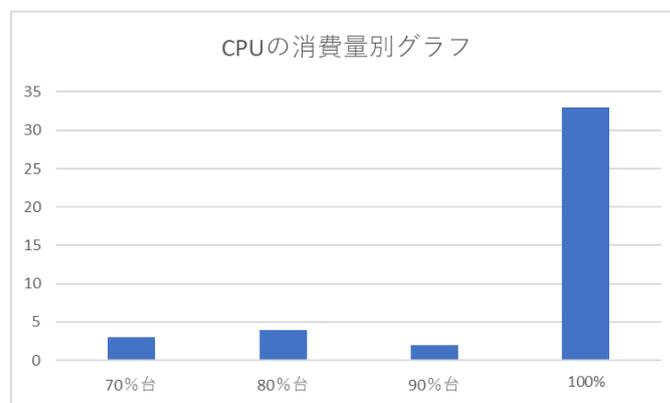


図 6 CPU 使用量別グラフ

6.3 実験(3)

実験(2)で生成できた 46 個の URL リストに対して実験(3)を行った。(3)で実際に Coinhive が文字列として表示されたのは 11 個だった。また、各 CSV ファイルの中でどれだけ文字列が出現するかに差があった。最も多くの文字列が見つかったもので 29 回であった。また、最も少ないもので 3 回であった。表 5 は文字列検索をかけ方ものの一つである。また、ユーザ側にマイニングをするかどうか確認を取る 'authedmin.com' の文字列が含むものは今回の CSV ファイルから一つも検出されなかった。

frame.number	col.info
345	Standard query 0xe1b A coinhive.com
368	Standard query response 0xe1b A 104.20.209.59 A 104.20.208.59
475	Standard query 0x3048 A ws030.coinhive.com
1292	Standard query 0x9005 A ws021.coinhive.com
1293	Standard query response 0x9005 A 37.187.167.69
1574	Standard query 0xe9fe A ws017.coinhive.com
1575	Standard query 0x0000 A wpad.local, QM" question"
1603	Standard query 0xc813 A ws025.coinhive.com
1604	Standard query response 0xc813 A 37.187.167.72
1816	Standard query 0xe50f A ws013.coinhive.com
1817	Standard query response 0xe50f A 37.187.167.21
1840	Standard query 0x4a57 A ws016.coinhive.com
1841	Standard query response 0x4a57 A 37.187.167.30
2074	Standard query 0xe068 A ws019.coinhive.com
2075	Standard query 0x0000 A wpad.local, QM" question"
2314	Standard query 0xd918 A ws007.coinhive.com
2315	Standard query 0xd918 A ws007.coinhive.com
2316	Standard query response 0xd918 A 37.187.165.207
2330	Standard query 0x9ef9 A ws028.coinhive.com
2332	Standard query 0x9ef9 A ws028.coinhive.com
2334	Standard query response 0x9ef9 A 37.187.167.83
2554	Standard query 0xb583 A ws008.coinhive.com
2555	Standard query response 0xb583 A 37.187.165.207
2583	Standard query 0xd075 A ws012.coinhive.com
2584	Standard query response 0xd075 A 37.187.166.108
2837	Standard query 0x8f15 A ws001.coinhive.com
2838	Standard query response 0x8f15 A 217.182.164.14
2914	Standard query 0x1d63 A ws027.coinhive.com
2915	Standard query response 0x1d63 A 37.187.167.83
23888	Standard query 0xea62 A ws009.coinhive.com
23889	Standard query response 0xea62 A 37.187.165.210
23934	Standard query 0x090e A ws010.coinhive.com
23935	Standard query response 0x090e A 37.187.165.210
24580	Standard query 0x58f6 A ws005.coinhive.com

表 5 文字列検索結果

7. 考察

2770 個のうちおよそ 30%である 975 個の Web ページが残った。このことから、マイニングスクリプトの疑いのあるサイトの生存期間が通常のサイトと比較して短いことが推測される。文献[14]では、およそ 20%のサイトが 9 日以内にアクセスが不能になっていることから、今回の結果も十分に妥当であるといえる。

実験(2)では、閾値として 70%を用いて測定を行った。結果として、全体に 5%程の検知であった。これより、4 章で述べた提案手法で検知が可能であることを示した。一方で、検出数に対して課題が残ったと考える。3 台の計算機で、同様の実験を行って検出数に大差がなかったため PC の処理性能の差に問題があったとは考えにくい。このため、文献[15]で示した閾値を再度検討する必要があると考える。しかしながら、閾値を下げ、判定条件を緩和することで、誤検知が誘発される可能性がある。これについては、複数の閾値を用いて検証を行い、改めて考察する必要がある。

実験(3)では、文字列「Coinhive」が各ログファイルに複数回出現していた。これにより、都度 coinhive.com へのアク

セスが行われていたといえる。また、通信解析内容から、Web ページにアクセスしてから一定時間経過後、対象の文字列が検出されている。このことから、アクセスしてから一定時間経過したのちにマイニングを行うような Web サイトも登場していると考えられる。ただし、上記の評価は文字列「Coinhive」だけでしか評価していない。提案手法をより厳密に評価するために「Coinhive」以外の文字列を予め調査しておき、通信解析ファイルに対して、文字検索を行う必要がある。上記については今後の課題とする。

8. 終わりに

本研究では CPU の用いた動的な解析によりマイニングサイトの判定を行った。これにより、Web ページを訪れた際にどれだけリソースを消費されているかどうかを知ることができた。しかし精度の観点からは、誤検知率の高さからこれだけでは判定が不十分だといえる。今後は誤検知率の低減方法を考える必要がある。併せて、Geng らが行ったように、ドメインリストを定期的に更新することで、マイニングされていると思しきサイトの生存情報を収集する必要がある。

参考文献

- [1] “New Frontiers In Cryptojacking” . <https://blog.qualys.com/securitylabs/2018/12/17/new-frontiers-in-cryptojacking> (参照 2019-01-21).
- [2] “採掘 (マイニング)” . <https://bitflyer.com/ja-jp/glossary/mining> (参照 2019-01-21).
- [3] “what is cryptocurrency mining?” . <https://www.itpro.co.uk/digital-currency/30249/what-is-cryptocurrency-mining> (参照 2019-01-21).
- [4] “Censys で日本国内における Coinhive の使用状況を調べてみた” . <https://ninoseki.github.io/security/2017/11/13/cryptojacking-in-japan.html> (参照 2019-01-21).
- [5] “McAfee Labs Threats Report” . <https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-sep-2018.pdf> (参照 2019-01-21).
- [6] “2018 年にサイバー犯罪者が狙う最大の標的は「仮想通貨の発掘? ” . <https://blog.trendmicro.co.jp/archives/17083> (参照 2019-01-21).
- [7] “KSN Report:Ransomware and malicious cryptominers 2016-2018” .https://media.kasperskycontenthub.com/wp-content/uploads/sites/58/2018/06/27125925/KSN-report_Ransomware-and-malicious-cryptominers_2016-2018_ENG.pdf (参照 2019-01-21).
- [8] “Cryptojacking - Cryptomining in the browser” <https://www.enisa.europa.eu/publications/info-notes/cryptojacking-cryptomining-in-the-browser> (参照 2019-01-21).
- [9] J. Liu, W. Li, G. O. Karame and N. Asokan, "Toward Fairness of Cryptocurrency Payments," in *IEEE Security & Privacy*, vol. 16, no. 3, pp. 81-89, May/June 2018.
- [10]M. Conti, E. Sandeep Kumar, C. Lal and S. Ruj, "A Survey on Security and Privacy Issues of Bitcoin," in *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3416-3452, Fourthquarter 2018.
- [11] Y. Gao, X. Chen, Y. Chen, Y. Sun, X. Niu and Y. Yang, "A Secure

- Cryptocurrency Scheme Based on Post-Quantum Blockchain,"
in *IEEE Access*, vol. 6, pp. 27205-27213, 2018.
- [12] Shayan Eskandari, Andreas Leoutsarakos, Troy Mursch, Jeremy Clark "A first look at browser-based Cryptojacking" in *IEEE SECURITY & PRIVACY ON THE BLOCKCHAIN (IEEE S&B) 2018 University College London (UCL), London, UK*
- [13] Jan R uth, Torsten Zimmermann, Konrad Wolsing, Oliver Hohlfeld "Digging into Browser-based Crypto Mining" in *IMC '18: Internet Measurement Conference*
- [14] Geng Hong, Zheming Yang, Sen Yang, Lei Zhang, Yuhong Nan, Zhibo Zhang, Min Yang, Yuan Zhang, Zhiyun Qian, Haixin Duan, "How You Get Shot in the Back: A Systematical Study about Cryptojacking in the Real World," *CCS '18 Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* Pages 1701-1713.
- [15] "Coinhive 利用サイトを探してみ
た" .<https://techblog.securesky-tech.com/entry/2018/09/10/> (参照
2019-01-21)