複数のプロキシを使用した経路変更Moving Target Defenseによるマルウェア検知・隔離

井上 幸紀[1,a)]　小出　洋[1]

概要：近年のインターネットの普及により社内ネットワークを導入する企業が増加している．こうした社内ネットワークは大きな利便性をもたらすものの, こうした内部ネットワークを標的とするサイバー攻撃やウイルスもまた増加している．さらに，一つのターゲットに狙いを絞り複数の段階を経て長期的かつ計画的に攻撃を行う標的型攻撃による被害も報告されている．こうした標的型攻撃では，攻撃者は標的のネットワーク内部に侵入してもすぐに攻撃を実施するわけではなく，更なる攻撃のため標的ネットワークの構成の把握やバックドアの設置，管理者権限への昇格などを試みる場合が多い．そのため内部に侵入したマルウェアを監視して検知や隔離を行うことにより被害を小さくすることができる．本研究では複数のプロキシノードを切り替えて経路を変化させる Moving Target Defense(MTD) を用いたシステムを構築し, ネットワーク内のマルウェアを検知, 隔離するということを目的とする．この手法によって内部ネットワーク内に侵入し攻撃を行うマルウェアを検知してサイバー攻撃からシステムを防御することが可能になる．本論文では提案したシステムのモデルを構築し, マルウェア検知と隔離について性能を評価した．その結果, 提案手法が有効であることが確認できた．

Detection and isolation malware by dynamic routing Moving Target Defense with proxies

Abstract: With the spread of the Internet in recent years, more and more companies are introducing in-house networks. Although these internal networks provide great convenience, cyber attacks and viruses targeting these networks are also increasing. In addition, there have been reports of advanced targeted attacks(APT) that target a single target and carry out attacks in multiple phases in a long-term. In such targeted attacks, the attackers do not immediately launch an attack after penetrating the target network, but often attempt to understand the configuration of the target network, install a backdoor, or gain administrative privileges for further attacks. Therefore, it is possible to reduce the attacked damage by monitoring, detecting, and quarantining the malware that has entered the network. In this study, we constructed a system using Moving Target Defense (MTD), which changes the route by switching multiple proxy nodes. The purpose of this system is to detect and isolate malware in the network. This method enables us to detect malware that invades and attacks the internal network, and to protect the system from cyber attacks. In this paper, we construct a model of the proposed system and evaluate the performance of malware detection and isolation. As a result, we confirmed the effectiveness of the proposed method.

Keywords: Moving Target Defense, Network Security, Malware, Intrusion Detection

1. はじめに

近年のインターネットの普及により, 社内ネットワークを導入する企業も増加している．社内のパソコンやプリンターなどの OA 機器, さらにはデータベースやサーバなど

[1] 情報処理学会
IPSJ, Chiyoda, Tokyo 101–0062, Japan
[†1] 現在，九州大学情報基盤研究開発センター
Presently with Research Institute For Information Technology, Kyushu University
[a)] inoue.kouki.882@s.kyushu-u.ac.jp

を一つのネットワークとして構成することでデータのやり取りや閲覧が容易となる．一方で本来限られた社員しかアクセスできないはずのローカルネットワークに悪意を持った外部の人間が侵入し, サイバー攻撃を行うことで顧客情報の漏洩などが発生する事例も多々発生している．

ランサムウェアや情報漏洩といったサイバー攻撃が増加する一方で，近年では標的型攻撃 (APT) と呼ばれる種類の攻撃も増えつつあり問題となっている [2][3]．標的型攻撃では従来のサイバー攻撃と異なり一つのターゲットに狙いを絞り，ターゲットのシステム内部へ侵入するために高度なテクニックが用いられる．またシステム内部に侵入してからも長い時間潜伏し，バックドアの設置やシステム構成の把握を行った上で攻撃を行う [4]．最近では大企業や官公庁を狙った標的型攻撃による被害も報告されており，国家や政治のレベルでも問題となっている [1]．こうした標的型攻撃ではいくつかの段階を経て，最終的な目標へのサイバー攻撃が行われる [6].

まず初めにインターネットや SNS を用いた事前調査と攻撃準備が行われる．次にフィッシングメールや水飲み場型攻撃などを通じて社員のパソコンへのウイルス感染が行われる．ターゲットの内部ネットワークに侵入した攻撃者はすぐに攻撃を実施するわけではなく，更なる攻撃のため標的ネットワークの構成の把握やバックドアの設置，機密情報の窃取，管理者権限への昇格などを試みる場合が多い．こうして攻撃の基盤を整えた上で最終的な目標 (データベースなど) へ攻撃を行う．

このような標的型攻撃ではマルウェアは数ヶ月以上の期間，システム内部に潜伏して様々な活動をおこなう [5]．そのため潜伏期間中にネットワーク内部のマルウェアを検知して駆除する事ができれば，大きな被害を出す事なくシステムを防御することができる．本研究では社内ネットワークにこうしたマルウェアが侵入しており, 更なる侵入拡大を狙って潜伏しているという状況を想定する．複数のプロキシを使用した MTD(Moving Target Defense) によるルーティングの変更により, マルウェアを検知・隔離する．

1.1 本研究の目的

上記で説明した方法により, 社内ネットワークなどに侵入し活動を行うマルウェアの挙動を検知し, プロキシの切り替えによってマルウェアをハニーポットへ誘導して隔離する．これによりマルウェアの検知, 隔離を行い, ネットワーク内に侵入したマルウェアの駆除とシステムの防御を助けることが目的である．第四章で述べる評価実験の結果によりマルウェアの検知, 隔離を行えることが確認でき, 提案手法が有効であることがわかった．

1.2 MTD とは

MTD は”Moving Target Defense”の略であり, サイバー攻撃のターゲットとなるシステムの設定を動的に変更することにより攻撃の成功確率を小さくできるという技術である [7] [8]．変更するべき部分は例えばシステムの IP アドレスや，動作するメモリの番地など多岐に渡りそれによって様々な種類の MTD が存在する．大まかな分類としては

(1) エンドポイント（IP アドレスや MAC アドレスなど）
(2) パケット転送（ルーティングノードやノード間のリンク）
(3) システムの動作環境（オペレーティングシステム, 命令セット, メモリ番地など）
(4) システムのソフトウェア（ソースコードの変更やエンコード）

のように分類される [8].

サイバー攻撃の防御策としては，システム内に存在する脆弱性を修正するためにパッチを当てるなどの処理が必要であるが，近年では未公開の脆弱性を利用して攻撃を行うゼロデイ攻撃なども盛んであり，このような場合はサイバー攻撃を防ぐことは難しい．MTD は脆弱性自体を修正するものではないが，動的なシステムの変更によってサイバー攻撃を難しくして攻撃の成功確率を小さくし，攻撃成功までの時間を稼ぐことでゼロデイ攻撃に対する有効な対抗策となりうる．

本研究では (2) のパケット転送部分について, クライアント-サーバ間に複数のプロキシを介在させて, 使用するプロキシを変更することで動的に経路を変化させる MTD を使用する．

1.3 本論文の構成

本論文の以下の構成は次のようになる．
第 2 章では関連研究と, 本研究の位置づけについて述べる．
第 3 章では本研究で使用する提案手法について説明する．
第 4 章では本研究のモデルについて評価実験を行う．
第 5 章では得られた実験の結果について考察を行う．
最後に第 6 章で本研究の結論と今後の展望を述べる．

2. 既存の研究と本研究の位置づけ

2.1 既存の研究

1.2 で説明したように,MTD には様々な手法があり大まかには 4 つに分類することができる．本論文に関連する研究としては, まず DYNAT[9] がある. この DYNAT ではネットワークを行き来するパケットのヘッダを変更することで IP アドレスと TCP ポートを動的に変更する．サーバ内に侵入した攻撃者はネットワークを流れるパケットを盗聴することでシステム構成を把握しようとするが，この手法によってそれが困難になる．上の分類では (1) エンドポイントでの MTD にあたる．Readactor[11] は (3) システムの動作環境レベルの MTD で, 特別なコンパイラを使ってコード

を生成することでソフトウェアが動作するメモリ番地を変更する．これにより return-oriented programming(ROP) 攻撃を防ぐことができる．(4) システムソフトウェアのレベルの研究としては NOMAD[10] がある．NOMAD では Web ページにおいて Html の name/id 要素を動的に変更することで, 悪意を持った Web ボットからの攻撃を防御できる. 同じく Web ボットへの対策である CAPTCHA と異なり, この手法では正常なユーザに認証などの不便さを強いることなく防御を行うことができる．

自分の提案手法と同じく, 複数のプロキシを使ったネットワークレベルの MTD も存在する. これらの手法では MTD を用いて DDoS 攻撃を防ぐことを目的としている．[12] では複数プロキシを使った MTD によって外部からの DDoS 攻撃を防ぐための MOTAG というシステムを実装している．この MOTAG では正規のユーザはまず認証サーバに

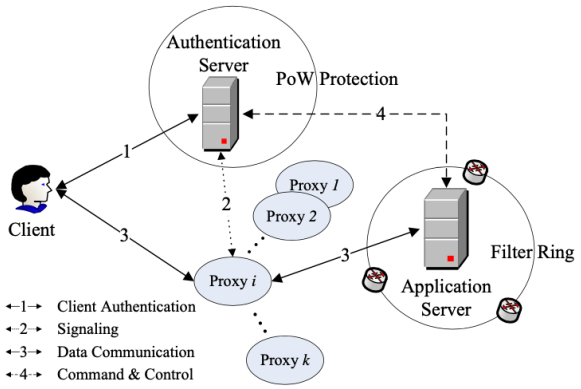

図 1 MOTAG のシステム構成 [12]
Fig. 1 MOTAG's Model.

登録, 認証してアクセスするプロキシの IP アドレスを得る. これによって正規ユーザは複数あるプロキシの中から正しいプロキシに接続して Web アプリケーションを利用できる. 一方で攻撃者は正しいプロキシを知らないため Web アプリケーションへ接続できず, よって DDoS 攻撃を行うことも出来ない．プロキシへの攻撃を検知した場合には, 動的にプロキシを変更することにより DDoS 攻撃から正規のユーザを保護し内部の攻撃者を発見することができる．また Web アプリケーション側ではフィルタを実装しプロキシ-サーバ間で軽量な通信プロトコルを採用することにより,DDoS 攻撃によるサーバダウンを防いでいる．

[13] は上記の MOTAG を発展させてクラウドや CDN での使用に適したシステム（DoSE) を実装することにより, 複数のプロキシを用いることによるコストを小さくし，より安価に DDoS 攻撃への対策を行えるようにしている．

[14] では，MOTAG や DoSE に対して有効な攻撃策となる，”Proxy harvesting attack”を提唱している．Proxy harvesting attack では攻撃者は正規ユーザになりすまし，認証サーバに対して認証を行うことで正規プロキシの IP アドレスを手に入れる．この IP アドレスを受け取るフェーズを何度も繰り返すことによって，システムで使われる IP アドレスを収集する事ができる．これによって正規プロキシとして使われうる IP アドレスを知る事ができるため，これらに DDoS 攻撃を行う事で動的なプロキシ切り替えを無効化する事ができる．

既存の複数プロキシベース MTD に対して有効である Proxy harvesting attack に対して，[14] では”BIND-SPLIT”という手法を提案した．”BIND-SPLIT”は BIND と SPLIT という二つの仕組みから構成される．BIND は正規ユーザと割り当てるプロキシをマッピングすることで管理する仕組みで，同じユーザに対して何度も正規プロキシが割り当てられるのを防ぐ．SPLIT では，あるプロキシで攻撃が検知された際に，そのプロキシに対して割り当てられたユーザを二つのグループに分けてそれぞれのグループに新たなプロキシを割り当てる．攻撃者は割り当てられた新たなプロキシに対して再び攻撃を行うため，SPLIT による割り当てを再度行う．これを繰り返すことによって攻撃者を特定する事ができる技術である．これら提案手法を組み合わせることで Proxy harvesting attack を防止し，さらに内部にサイバー攻撃への協力者がいた場合であっても，それを特定して攻撃を防止することができるということを [14] では論じている．

2.2 本研究の位置付け

上記で説明した関連研究では DDoS 攻撃への対策という目的のためにシステムが構成されていた．一方で本研究においては外部からの DDoS 攻撃に対する防御ではなく，内部ネットワークのマルウェア検知と隔離を目標としている．そのため本研究では関連研究のシステム構成を参考にしつつ, サーバ側のフィルタや認証サーバの Proof-of-Work によるユーザ認証などを省略し, 内部ネットワーク内でのマルウェア検知を主眼においたシステムを提案する．さらに [14] で提案された手法である BIND-SPLIT を取り入れることで，検知と隔離の性能を高めた．

3. 提案手法

本章では, 複数のプロキシを使用した経路変更 MTD についての概要と, 本実験で使用するシステムについての説明を行う．

3.1 複数のプロキシを使用した経路変更 MTD

社内ネットワーク内において, クライアントがネットワーク内の Web サーバにアクセスする場面を想定する．一方でマルウェアもネットワーク内に侵入しており同じく Web サーバにアクセスして偵察や攻撃を行おうとしていると仮定する．この時クライアント-サーバ間にプロキシを設置

し, リバースプロキシサーバとして動作させることで実際の Web サーバの IP アドレスを知られることなく, サービスを提供することができる. しかしこのままではリバースプロキシサーバを経由してマルウェアに攻撃を受ける可能性がある. そこで, このリバースプロキシサーバを複数設置して, その中で一つのプロキシのみが Web サーバへとアクセスできるようにする. 正規のクライアントは予め管理サーバに登録して認証することで, 正しいプロキシのアドレスを入手し Web サーバを利用できる. この正しいプロキシを本論文では正規プロキシと呼ぶ. 一方でマルウェアは正規プロキシを知らないので, 総当たりでプロキシへランダムにアクセスするしかない. これによって攻撃にかかる時間を増大させて攻撃の成功確率を減少させることができる. もし正規プロキシに対するマルウェアのアクセスが確認された場合は, 動的にプロキシを切り替えることにより Web サーバへのアクセスを回避することもできる. 本実験では更に Web サーバと同じ働きをするハニーポットを導入し, 正しくないプロキシの接続先をハニーポットに設定することでマルウェアの隔離及び動作の観察も行うことができるようになる. そこで本論文では, 正規ユーザがアクセスする Web サーバを正規サーバ, マルウェアが誘導されるサーバをハニーポットと呼ぶことにする.

3.2 動的なプロキシの変更

上記のシステムによってマルウェアの検知と攻撃に対する防御を行うことができる. しかし例えばネットワーク内部の協力者による手助けなどによりマルウェアが正規プロキシを選択し正規サーバへ攻撃を仕掛けることも考えられる. 対策として本システムでは, マルウェアによる攻撃が検知された場合には動的にプロキシ設定の変更を行う. 具体的な手法として複数のプロキシの中からランダムに次の正規プロキシを選択し, 接続先をハニーポットから正規サーバへと変更する. 次にそのプロキシのアドレスをクライアントに通知し, 元の正規プロキシの接続先をハニーポットへと変更する. これを動的に行うことでマルウェアに気づかれることなく攻撃先をハニーポットへと変更させることができ, 正規サーバへの攻撃を中断させてマルウェアを隔離する事ができる.

3.3 BIND-SPLIT によるマルウェア検知

関連研究の [14] では”BIND-SPLIT”という手法により, ユーザとプロキシをマッピングして, プロキシを効率よく割り当てることで攻撃者を特定する事ができた. 本研究でも本手法を取り入れることでマルウェア検知の性能を高めることに成功した. BIND-SPLIT によるマルウェア検知は図 3 のようになる. 攻撃を検知した際にユーザを二つのグループに分けてプロキシを割り当てるというフェーズを繰り返すことで複数のユーザの中から攻撃者を絞り込む事ができる.

3.4 実装

システムの全体図は図 2 のようになる. 本実験では Vagrant を使って, MAC OS 上に Linux のディストリビューションの一つである Ubuntu の仮想マシンを立ち上げてシステムを構築した. 本システムの実装を行うには複数のプロキシやサーバ, クライアントに対して別々の IP アドレスを振り分ける必要がある. しかしそれぞれに対して仮想マシンを立てた場合, プロキシ数の増加に伴って非常に大きなマシンパワーが必要となってしまう. そこで本研究では Linux カーネルのコンテナ関連技術の一つである Namespace を活用してシステムを実装した. これによって一つの Linux カーネル上に隔離された複数のネットワーク名前空間を作る事ができ, 複数の仮想マシン立ち上げによるオーバヘッドを大幅に削除する事ができた. 実験で使用するプロキシ群やクライアント, マルウェアや正規サーバは Python で実験用のプログラムを自作して, それぞれのネットワーク名前空間に IP アドレスを割り当てて動作させている. 今回の実験で使用した OS や Python のバージョンは表 1 の通りである.

表 1 使用した環境とバージョン
Table 1 Environments and Version.

環境	バージョン
OS	Linux 5.4.0-73-generic
ディストリビューション	Ubuntu 20.04
Python	3.8.10

4. 評価

本章では, 提案したシステムの有効性を確かめるために行った評価実験とその結果について述べる.

4.1 実験 1: 複数プロキシによるマルウェアの検知

システム内にマルウェアが侵入し, 更なる攻撃の拡大のためにネットワーク内の正規サーバに対して攻撃を試みている状況を考える. 攻撃からの防御とマルウェア検知のため, 複数のプロキシを起動させてその中の一つのみが Web サーバへ接続できるように設定する. 正規のユーザは正規プロキシを知っている一方で, マルウェアはそれを知らないのでランダムにプロキシを選択する. マルウェアが正規サーバへアクセスできる確率, つまり正規プロキシを選択する確率は起動しているプロキシ数に反比例して小さくなると予想される. 本項では以上のことを確かめるための実験について述べる.

4.1.1 実験方法

マルウェアは起動しているプロキシの中からランダムに

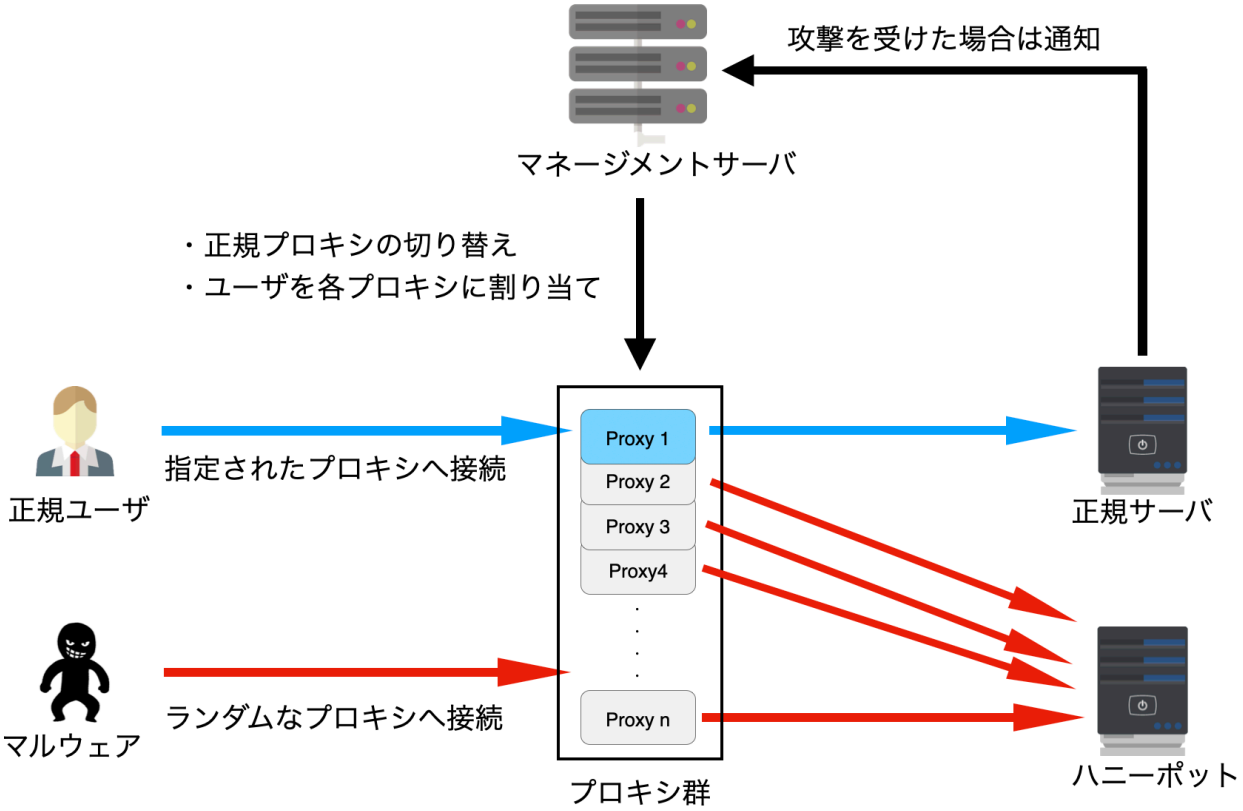

図 2 提案システム
Fig. 2 Proposed System.

接続するプロキシを選択してアクセスする．これを 10000 回行い, その中でマルウェアが正しいプロキシを選ばなかった回数, つまりマルウェアをハニーポットへ誘導した回数を記録する．総プロキシ数を 2, 4, 8, 16, 32, 64 個と変化させて, それぞれの場合について上記の試行を行った．プロキシ総数を N としたとき, マルウェアが正しくないプロキシを選ぶ確率 P は

$$P = 1 - \frac{1}{N}$$

となる．

4.1.2 実験結果

実験の結果は表 2 のようになった．

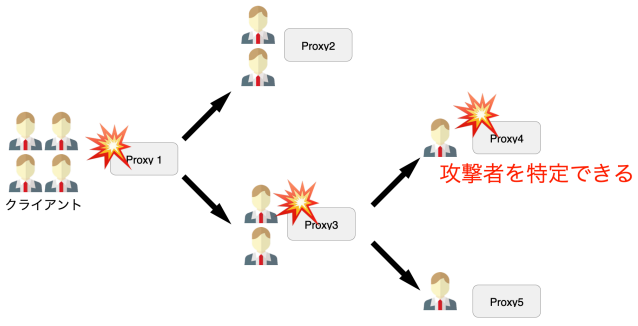

図 3 BIND-SPLIT によるマルウェアの特定
Fig. 3 Detecting Malware by BIND-SPLIT.

4.2 実験 2: BIND-SPLIT によるマルウェアの検知

内部ネットワークに侵入したマルウェアが何らかの方法によってマネージメントサーバで認証を行い，正規プロキシの IP アドレスを入手したとする．そのため正規サーバは正規プロキシを経由してマルウェアから攻撃を受けたため，マネージメントサーバにアクセス元のプロキシのアドレスを通知した．この時, 同じプロキシに複数人の正規ユーザも接続しており，どのユーザが攻撃者かどうかを判別できないとする．本実験ではこのような状態の時に BIND-SPLIT によってユーザに対して新たなプロキシを割り当て，これを繰り返す事でマルウェアを検知できる事, また正規ユーザは問題なく正規サーバにアクセスし続けられる事を確かめた．

表 2 実験 1 の結果
Table 2 Experiment1's Result.

プロキシ総数	検知できた確率	P の理論値
2	0.4963	0.5
4	0.7558	0.75
8	0.8738	0.875
16	0.9325	0.9375
32	0.9657	0.96875
64	0.9854	0.984375

4.2.1 使用するモデル

本実験では以下の 3 つのモデルを使用する．
プロキシ (64 個)
クライアント（1〜32 人)
マネージメントサーバ (1 個)
マルウェア (1 個)

4.2.2 実験方法

実験は以下の手順で実施した．
①クライアントとマルウェアを同じ正規プロキシへ割り当て，クライアントは割り当てられたプロキシに接続し，マルウェアは攻撃を行う．
②正規サーバは攻撃を検知するとマネージメントサーバに対して，どのプロキシから攻撃が行われたかを通知する．
③マネージメントサーバは通知を受けると，BIND-SPLIT によってランダムに新たな正規プロキシを二つ選びクライアントとマルウェアを割り当てる．
④クライアントおよびマルウェアは正規プロキシにアクセスし，それぞれ新たな正規プロキシの割り当てを受ける．
⑤クライアントとマルウェアは割り当てられた新たなプロキシにアクセスする．マルウェアは再び攻撃を行う．
⑥攻撃元のプロキシに割り当てられたユーザが一人になるまで②〜⑤を繰り返す．これによってマルウェアを特定できる．

上記の試行をクライアントの数を 1〜32 人と変化させた上で実験を行った．
ここでクライアントとマルウェアの総数を N とすると，本実験ではマルウェアは一つなので N は 2〜33 となる．
プロキシ割り当ての回数を d とすると，初めに一つプロキシがあり，新たに必ず二つのプロキシを割り当てるので d は二分木の深さとなる．マルウェア検知までに要するプロキシの総数 M とすると，

$$M = 2 * d + 1$$

また d の最大値は

$$\lceil \log_2 N \rceil$$

となるため，M の最大値

$$M_{max}$$

は

$$M_{max} = 2 * \lceil \log_2 N \rceil + 1$$

になると予想される．

4.2.3 実験結果

実験の結果，クライアントの数に関わらずマルウェアを検知することに成功した．また正規ユーザも新たに割り当てられたプロキシへ接続することで，正規サーバへ引き続きアクセスする事ができた．ここでマルウェア検知までに要したプロキシの総数と，プロキシ割り当て回数は図**図 4** のようになった．プロキシ割り当て回数と要したプロキシ総数は，事前の予想通りの値となっていることがわかる．

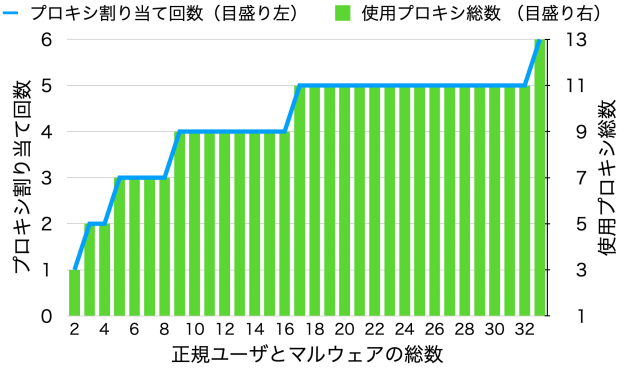

図 4 実験 2 の結果
Fig. 4 Experiment2's Result.

5. 考察

本章では, 第四章で行った実験の結果と考察について述べる．

5.1 実験 1 の考察

実験 1 の結果より, マルウェアを検知できる確率（正しくないプロキシを選ぶ確率）はプロキシ総数に反比例して小さくなり，実験の結果は概ね理論値と一致するということがわかる．そのため更に使用するプロキシの数を増やすことでマルウェアを検知できる確率も向上するが，その分起動やプロキシ切り替えに掛かる時間と必要になるマシンパワーは大きくなる．
現在の実装ではプロキシの起動や切り替えは同期処理で行っているため，プロキシ数の増大に従い時間も長くなる．そこでスレッドプールを利用して並列処理にする事で，掛かる時間をある程度は短縮することができると考えられる．
次に必要なマシンパワーであるが，本実装では Namespace を利用してシステムを構築したため，それぞれのプロキシに対して別々の仮想マシンを立てる方法に比べると大幅に必要なリソースを削減する事ができた．しかしそれでもシステムを大規模にした場合は一台のマシンでは動かせなくなる可能性が考えられる．そこでクラウド環境を利用して，今まで一つのマシン上で動かしていたプロキシやマネージメントサーバを別々のマシンで動かすという手法が考えられる．これによってシステム規模を拡大しプロキシ数を増やす事ができる一方で，マシン間での安全な通信方法についても考慮する必要がある．

5.2 実験 2 の考察

実験 2 の結果から，BIND-SPLIT によってユーザとプロキシをマッピングした上で，新たなプロキシ割り当ての繰り返しによりマルウェア検知を行える事がわかった．また必要となるプロキシの総数は事前の予想通りの結果となった．同じプロキシに接続しているユーザ数を N とした時マルウェア検知に必要なプロキシ数は O(log N) で増加するため，内部ネットワークを対象としていることもあり接続ユーザ数の増加にも十分対処できると考えられる．一方で本実験ではマルウェアの数を一つと仮定して実験を行った. 実際には複数のマルウェアから同時に攻撃を受ける可能性も十分考えられるため，それらに対処できるようにする必要がある．

6. おわりに

本論文では複数のプロキシを使用することによって経路変更 MTD を実装し, 社内ネットワーク内に侵入したマルウェアの検知や隔離を行うという手法を確認した．いくつかの実験を行い性能を評価した結果, 使用するプロキシ数に応じてマルウェア検知の確率は高まり，またプロキシ切り替えによって複数のユーザの中からマルウェアを検知できるということがわかった．これらより提案手法が社内ネットワーク内でのマルウェア駆除やシステム防御に有効であることがわかった．今後の課題として，本論文では検知したマルウェアをハニーポットへと誘導することでシステムからの隔離を行うに留めたが，このハニーポットを有効に使うことで単なる隔離のみならずマルウェアの動作を観察することも可能であると考えられる．これによって未知のマルウェアであっても攻撃手段などを知る事ができ，これからのシステム防御に役立てられると思われる．今後はクラウド等を使ってプロキシ数の増加を容易に行えるようなシステムの開発・構築を行い，またマルウェアの動作を観察するためのハニーポットについても研究を進め，マルウェア検知・隔離・動作の観察の性能強化を行いたいと考えている．

謝辞 システム実装において Namespace 名前空間などのコンテナ関連技術についてご教授いただきました寺嶋 友哉先輩にこの場を借りてお礼申し上げます．本研究は，科研費基盤（C) JP18K11295, 国立研究開発法人科学技術振興機構（SICORP), 日立システムズの補助を受けている．

参考文献

[1] IPA 独立行政法人 情報処理推進機構: 標的型サイバー攻撃の脅威と対策, IPA 独立行政法人 情報処理推進機構, page 2 (2013)
[2] IPA 独立行政法人 情報処理推進機構: 情報セキュリティ 10 大脅威 2020, IPA 独立行政法人 情報処理推進機構, pages 22-24 (2020)
[3] IPA 独立行政法人 情報処理推進機構: 情報セキュリティ 10 大脅威 2021, IPA 独立行政法人 情報処理推進機構, pages 6-9 (2020)
[4] Chen Ping and Desmet Lieven and Huygens Christophe: A study on advanced persistent threats, IFIP International Conference on Communications and Multimedia Security, pages 63-72 (Springer) (2014)
[5] Li Meicong and Huang Wei and Wang Yongbin and Fan Wenqing and Li Jianfang: The study of APT attack stage model, 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), page 1-5 (IEEE) (2016)
[6] IPA 独立行政法人 情報処理推進機構: 「新しいタイプの攻撃」の対策に向けた設計・運用ガイド 改訂第 2 版, page 10-15, IPA 独立行政法人 情報処理推進機構 (2011)
[7] Zhuang Rui and DeLoach Scott A and Ou Xinming: Towards a theory of moving target defense, Proceedings of the First ACM Workshop on Moving Target Defense, page 31-40 (2014)
[8] Cheng Lei and Hong-Qi Zhang and Jing-Lei Tan and Yu-Chen Zhang and Xiao-Hu Liu: Moving Target Defense Techniques: A Survey, Security and Communication Networks, (2018)
[9] Dorene Kewley and Russ Fink and John Lowry and Mike Dean: Dynamic Approaches to Thwart Adversary Intelligence Gathering, BBN Technologies, A Verizon Company (2001)
[10] S. Vikram and Chao Yang and Guofei Gu: NOMAD: Towards non-intrusive moving-target defense against web bots, 2013 IEEE Conference on Communications and Network Security (CNS), page 55-63 (2013)
[11] Crane Stephen and Liebchen Christopher and Homescu Andrei and Davi Lucas and Larsen Per and Sadeghi Ahmad-Reza and Brunthaler Stefan and Franz Michael: Readactor: Practical code randomization resilient to memory disclosure, 2015 IEEE Symposium on Security and Privacy, page 763-780 (IEEE) (2015)
[12] Quan Jia and Kun Sun and Angelos Stavrou: MOTAG: Moving Target Defense Against Internet Denial of Service Attacks, (Conference Paper) (2013)
[13] Wood Paul and Gutierrez Christopher and Bagchi Saurabh: Denial of service elusion (DoSE): Keeping clients connected for less, 2015 IEEE 34th Symposium on Reliable Distributed Systems (SRDS), page 94-103 (IEEE) (2015)
[14] Sridhar Venkatesan and Massimiliano Albanese and Kareem Amin and Sushil Jajodia and Mason Wright: A Moving Target Defense Approach to Mitigate DDoS Attacks against Proxy-Based Architectures (2016)