ネットワークレベルのMTDによるWebアプリケーションのサイバー攻撃からの防御

桝本 武志[1,a)]　小出 洋[2,b)]

概要：本研究の目的は，ネットワークレベルにおける Moving Target Defense（以下，MTD）により，Web アプリケーションをサイバー攻撃から防御することである．Web アプリケーションにおけるセキュリティでは，ファイアウォールや IPS/IDS，WAF などによって攻撃を防ぐ方法が存在する．しかし，一般に，脆弱性が全くないシステムを構築することは不可能であると言われており，これらを適用したとしても全ての攻撃を防ぐことは困難である．この問題を解決するためのアプローチとして，比較的新しい防御手法である MTD を用いる研究が存在する．
MTD は，コードレベルやアプリケーションレベル，ネットワークレベルなど，システムのあらゆるレベルにおいて，攻撃対象となる領域を動的に変更する手法である．MTD は，攻撃の到達可能性自体を低下させるため，いかなる攻撃手法に対しても一定の効果を発揮することが可能であると考えられる．本研究では，ネットワークレベルにおける MTD を Web アプリケーションに適用する構成として，Web アプリケーション内のアカウント情報と，MTD におけるアドレス変更の情報を結びつけるという新しい手法を提案する．さらにその性能を実験により評価することで，この手法の有効性および有用性を確認した．

Protecting Web Applications from Cyber Attacks with Network Level MTD

1. はじめに

1.1 本研究の目的

本研究の目的は，ネットワークレベルにおける MTD により，Web アプリケーションをサイバー攻撃から防御することである．

現在，様々なサービスが Web アプリケーションを利用して提供されている．中には，銀行口座の預金操作や，物の売買が可能なものが多数存在する．そのような Web アプリケーションでは，利用者の個人情報や取引情報，クレジットカード情報など，流出すると利用者に多大な損害を与えてしまうような情報を扱うことが多い．そのため，Web アプリケーションを通してサービスを提供する場合，その安全性は最も重要な性能の 1 つである．

Web アプリケーションの攻撃方法には，SQL インジェクションや CSRF（クロスサイトリクエストフォージェリ），XSS（クロスサイトスクリプティング）など，多くの手法が存在する．[1] 対策として，ファイアウォールや IPS（Intrusion Prevention System）/IDS（Intrusion Detection System），WAF（Web アプリケーションファイアーウォール）などによってセキュリティを強化することが可能であるが，これにより全ての攻撃を完全に防ぐことは困難であり，近年においても多くの脆弱性が発見されている．[2] 本研究の目的は，この問題をネットワークレベルにおける MTD により解決することである．

MTD は，コードレベルやアプリケーションレベル，ネットワークレベルなど，システムのあらゆるレベルにおいて，脆弱性を含み攻撃対象となる領域を動的に変更する手法である．ネットワークレベルでは，攻撃対象となるサーバの IP アドレスおよびポート番号を頻繁に変更することで，攻撃からサーバを防御する．攻撃者は，攻撃対象の IP アドレスおよびポート番号を知らなければ攻撃することが不可能であるため，これにより攻撃の成功確率を低下させ

[1] 九州大学 工学部 電気情報工学科 計算機工学課程
Kyushu Univrsity，Fukuoka，819-0385，Japan
[2] 九州大学 情報基盤研究開発センター
Kyushu Univrsity，Fukuoka，819-0385，Japan
[a)] masumoto.takeshi.222@s.kyushu-u.ac.jp
[b)] koide@cc.kyushu-u.ac.jp

ることが可能である．MTD は，攻撃の到達可能性自体を低下させるため，既知の手法による攻撃だけでなく，今後現れるいかなる攻撃手法に対しても一定の効果を発揮することが可能であると考えられる．本研究では，ネットワークレベルにおける MTD を Web アプリケーションに適用するための構成を 1 つ提案し，その性能を実験により評価した．評価実験の結果，この手法により，許容可能なコストで Web アプリケーションのセキュリティを改善することができることがわかった．

1.2 本論文の構成

第 1 章では，本研究の目的を述べた．第 2 章では，既存研究をいくつか紹介し，本研究との関連性について述べる．第 3 章で，ネットワークレベルにおける MTD を Web アプリケーションに適用する構成を提案する．第 4 章で，その性能を評価するために行った実験の概要を述べ，第 5 章で実験の結果とそれに対する考察を述べる．

Web アプリケーションにネットワークレベルの MTD を適用する場合，攻撃者からは攻撃対象のサーバを隠す必要があるが，同時に，可用性を保つために正規のクライアントによるサーバへのアクセス経路を確保する必要がある．提案する構成は，この問題を解決するだけでなく，Web アプリケーション内のアカウント情報と，MTD におけるアドレス変更の情報を結びつけるという新しい手法により，より高いセキュリティを提供するものである．一般に，Web アプリケーションは，インターネット上でサービスを提供するものであるため，MTD において変更する IP アドレスはグローバル IP アドレスである必要がある．しかし，本研究では，単に，有効な構成の提案とその性能評価を行うことが目的であるため，攻撃対象であるサーバのプライベート IP アドレスを変更することで，擬似的な実験を行う．

最後に第 6 章でまとめを行う．

2. 既存研究と本研究の関連性

本章では，ネットワークレベルにおける MTD に関する既存研究，および Web アプリケーションへの適用に関する既存研究の概要と，本研究との関連性について述べる．

2.1 SDN を用いたネットワークレベルにおける MTD

ネットワークレベルの MTD において，Software-Defined Networking（以下 SDN）を用いる方法 [3], [4], [5] が存在する．この方法では，コンピュータの実際の IP アドレスを固定したまま，それぞれの実 IP アドレスに割り振る仮想 IP アドレスを頻繁に変更する．SDN を用いることにより，サーバで動かすアプリケーションと MTD の実装を分離することができるため，MTD の展開や実装のカスタマイズを容易に行うことができるという利点がある．

本研究では，Web アプリケーション内のアカウント情報を，ネットワークレベルの MTD におけるアドレス変更の情報と結びつける新しい手法を提案する．SDN は単にネットワークにおけるルーティングを提供する仕組みであるため，SDN を用いた方法ではこの手法の実装は困難である．そのため，本論文で提案する構成では SDN を使用せず，サーバのアドレスを実際に変更することで MTD を実現する．

2.2 ネットワークレベルにおける MTD の有効性に関する研究

ネットワークレベルにおける MTD の有効性に関する研究には，一定の IP アドレスとポート番号の空間における，アドレスを変更する間隔と攻撃成功確率の関係を示したもの [6] や，攻撃回数と攻撃成功確率の関係を示したもの [7] が存在するが，いずれもシミュレーションによるものであり，実際に実装および実験してデータを得たものではない．本研究では，実際に，MTD を適用した簡単な Web サイトを実装し，それに対して攻撃を行う．より理論値との比較が容易な条件でデータを解析することで，ネットワークレベルにおける MTD の有効性を確認する．

2.3 MTD の Web アプリケーションへの適用

リソースプール内に存在するリソースの組み合わせを動的に変更する手法 [8] では，バックエンドリソースプールに，システムを構成する要素である Web サーバとデータベースサーバをそれぞれ複数用意し，使用するリソースの組み合わせを動的に変更する．これにより，攻撃者による偵察および攻撃の難易度を大幅に向上させる．

WebMTD[9] と呼ばれる手法では，サーバサイドのコードブロック（`PHP`）や，クライアントサイドのコードブロック（`JavaScript`）および `HTML` の要素に属性を追加し，その値を変更することで，Web アプリケーションにおけるコードインジェクション攻撃を阻止する．

Web アプリケーションに固有のレイヤー（プレゼンテーション層，アプリケーション層，データ層）への MTD の適用に関する論文 [10] では，Web アプリケーションのネットワークレベルにおいても MTD を適用可能であることを示唆しているものの，実際の適用方法などは示していない．

本研究では，ネットワークレベルにおける MTD を Web アプリケーションに適用する構成として，Web アプリケーション内のアカウント情報と，MTD におけるアドレス変更の情報を結びつけるという新しい手法を提案する．

3. 提案手法

本章では，ネットワークレベルにおける MTD を，Web アプリケーションに適用するに際して解決すべき問題について述べ，それを解決する手法を 1 つ提案する．提案する手法は，Web アプリケーション内のアカウント情報を，

MTD におけるアドレス変更の情報と結びつける新しい手法である．

3.1 ネットワークレベルにおける MTD の Web アプリケーションへの適用

ネットワークレベルにおける MTD を Web アプリケーションに適用する上で注意すべきことは，可用性を保つため，攻撃者からサーバを隠す必要があると同時に，正規のクライアントによるサーバへのアクセス経路を確保する必要があるということである．この問題を解決するためには，何らかの方法により，正規のクライアントがサーバのアドレスを知っている状態を維持しなければならない．本研究では，クライアント自身から変更後のアドレスを指定する方法を提案する．

3.2 提案手法の概要

本研究で提案する手法は，Web アプリケーション内のアカウント情報と MTD におけるアドレス変更の情報を結びつけることによって，Web アプリケーションのセキュリティを向上させる．

この手法では，2 つのサーバを用いる．MTD を適用するサーバ（以下メインサーバ）と区別して，もう 1 つのサーバを認証サーバと呼ぶことにする．

アカウント登録

ユーザはまずこの認証サーバにアクセスし，アカウント登録を行う．認証サーバでは，2 段階認証など（本研究では電話番号認証を利用），高いセキュリティレベルの認証方法によりユーザを認証する．したがって，アカウント登録では，そのために必要な情報（電話番号など）も登録する．

アドレスの変更

ユーザが，初めてメインサーバにアクセスする際や，自身に割り当てられたアドレスを忘失してしまった場合は，認証サーバにアクセスして，自身のアカウントに割り当てるアドレスを指定する．指定したアドレスからメインサーバにアクセスし，自身のアカウントでログインすることで Web サービスを利用する．ユーザはログイン後，ログアウトするまでの間に次に割り当てるアドレスを指定することで，以降は認証サーバにアクセスしなくても，継続してメインサーバにアクセスすることができる．

ログイン

各ユーザは，自身のアカウントに割り当てられたアドレス以外のアドレスからはログインすることができない．

3.3 実装

一般に，Web アプリケーションは，インターネット上でサービスを提供するものであるため，MTD において変更する IP アドレスはグローバル IP アドレスである必要がある．しかし，本研究では，単に性能評価を行うことが目的

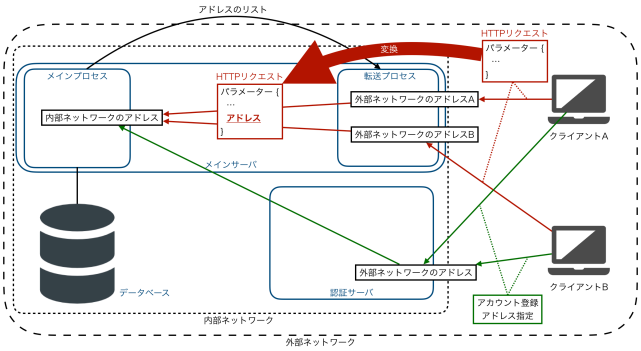

図 1 提案手法の構成
Fig. 1 Configuration of the proposed method

であるため，メインサーバのプライベート IP アドレスを変更することで，擬似的な実験を行う．

この手法では，2 つのネットワークを用いる．以降では，本来はインターネットを想定しているネットワークを外部ネットワークと呼び，もう 1 つのネットワークを内部ネットワークと呼ぶ．

メインサーバー

メインサーバには，2 つのプロセスが存在する．1 つは，クライアントにサービスを提供するための Web アプリケーションである．以降は，これをメインプロセスと呼ぶ．もう 1 つは，クライアントから受け取った HTTP リクエストに変更を加えたものを，メインプロセスに送信するプロセスである．以降は，これを転送プロセスと呼ぶ．

転送プロセス

メインプロセスが指定したすべてのアドレスにおいて，クライアントからの HTTP リクエストを受け付ける．クライアントから HTTP リクエストを受信すると，そのリクエストのパラメーターに address を追加し，値を，リクエストを受信したアドレスとした上で，それをメインプロセスに送信する．メインプロセスから返ってきたレスポンスを，そのままクライアントに送信する．

メインプロセス

メインプロセスにおいて使用する IP アドレスは内部ネットワーク内の IP アドレスであり，クライアントや攻撃者は直接アクセスすることはできない．メインプロセスは，転送プロセスと認証サーバを通してクライアントの HTTP リクエストを受け取る．転送プロセスを通して受信したリクエストについては，リクエスト内の address パラメーターの値と，そのクライアントのアカウント情報により，アクセス制限などの制御を行う．認証サーバを通して受信したリクエストについては，すでに，2 段階認証などの高いセキュリティレベルの認証方法により，ユーザを認証済みであるため，メインプロセスでは制御は行わず，アカウント登録およびアドレスの割り当てを行う．

認証サーバー

2 段階認証などの高いセキュリティレベルの認証方法に

よりユーザを認証した上で，アカウント登録およびアドレスの割り当てに関するリクエストを受け取る．クライアントから受信したリクエストから，適切な HTTP リクエストを生成し，メインプロセスに送信する．メインプロセスから返ってきたレスポンスから，適切な HTTP レスポンスを生成し，それをクライアントに送信する．本手法では，認証サーバにおいては高いセキュリティにより守られていることを前提とする．

4. 実験

本章では，第 3 章で提案した手法の有効性を確認するために，セキュリティ面，および応答速度の面について，実験により性能の評価を行った．セキュリティの性能についての実験を評価実験 1，応答速度の性能についての実験を評価実験 2 として概要を以下に示す．

4.1 評価実験 1

本項では，第 3 章で提案した手法における，セキュリティの性能を評価するために行った実験の概要を述べる．提案した手法では，クライアントそれぞれに別のアドレスを用意し，セッションベースでそれぞれのアドレスの変更を行う．実験を行う際，この手法を適用したサーバをそのまま攻撃の対象として使用した場合，構成が複雑であるため評価が困難となる．そこで，本研究では，ネットワークレベルの MTD を非常に単純な構成で適用した Web サーバを別に実装し，それに対して攻撃を行うことにより間接的に本手法の有効性を示す．

4.1.1 実験方法

実験を構成する要素である，攻撃の対象となるサーバ（以下メインサーバ）および攻撃者について説明する．実験は，メインサーバが利用可能なアドレスの数を変更して，13 回行った．

メインサーバー

メインサーバは，GET リクエストを受信すると，ある特定の文字列をボディとするレスポンスを返す．IP アドレスおよびポート番号は一定時間 T ごとに変更（ホッピング）するが，変更前と同じアドレスは使用しないようにした．

攻撃が失敗したときのコネクションエラーには，

- Connection refused
- Host is down
- No route to host
- Connection reset by peer
- Connection timed out

などがあり，それぞれのエラーによって（特に Connection timed out は）1 回の試行にかかる時間が異なる．そのため，攻撃者の攻撃速度は一定ではない．そこで，理論値との比較を容易化するため，T には，攻撃者が確実に全てのアドレスを調べることが可能な時間を設定した．

攻撃者

攻撃者は，攻撃対象のサーバが利用可能な，IP アドレスとポート番号の空間，および Web ページの脆弱性を知っているということを前提とした．

攻撃について

攻撃者は次の（1）～（4）を，アクセスを試みた回数（以下攻撃回数）が 100000 回になるまで繰り返し行う．

(1) 攻撃対象のサーバが利用可能な IP アドレス，ポート番号に対して順番にアクセス，GET リクエストを送信
(2) HTTP レスポンスを受信し，そのボディが，メインサーバが送信した文字列と一致した場合，攻撃成功とみなす
(3) 攻撃回数，攻撃成功回数をカウント
(4) 攻撃が成功した場合は，正しい HTTP レスポンスが返ってこなくなるまで同じアドレスにアクセスし続ける（攻撃成功回数にはカウントしない）

4.1.2 理論値

実験 1 における，攻撃成功確率の理論値を求める．

利用可能なアドレスの数（{IP アドレスの範囲 }×{ ポート番号の範囲 }）を N，アドレスが変更されるまでの間における最大攻撃可能回数を k とする．攻撃者は全てのアドレスを調べることが可能であるため，攻撃成功回数は常に 1 回であり，攻撃成功回数の期待値は $successes_E = 1$ である．また，メインサーバは，アドレスを変更する際，以前と同じアドレスは使用しないため，$k = N - 1$ である．したがって，攻撃回数の期待値 $attacks_E$ は次の式で表される．

$$\begin{aligned} attacks_E &= \sum_{i=1}^{k} \left(i \times \frac{1}{N-1} \right) \\ &= \frac{k\,(k+1)}{2\,(N-1)} \\ &= \frac{N\,(N-1)}{2\,(N-1)} \\ &= \frac{N}{2} \end{aligned}$$

以上より，攻撃成功確率の理論値 P_{th} は，次の式で表される．

$$P_{th} = \frac{successes_E}{attacks_E} = \frac{2}{N} \tag{1}$$

4.2 評価実験 2

第 3 章で提案した手法を用いた場合，セキュリティ面における性能が向上する一方で，ユーザは転送プロセスを介してメインプロセスにアクセスする必要があるため，応答速度の面においてはパフォーマンスが低下すると考えられる．本項では，第 3 章で提案した手法における，応答速度の性能を評価するために行った実験の概要を述べる．

4.2.1 実験方法

本研究では，POST メソッドによりデータを送信した時の応答にかかる時間 t は，送信するデータの大きさを d と

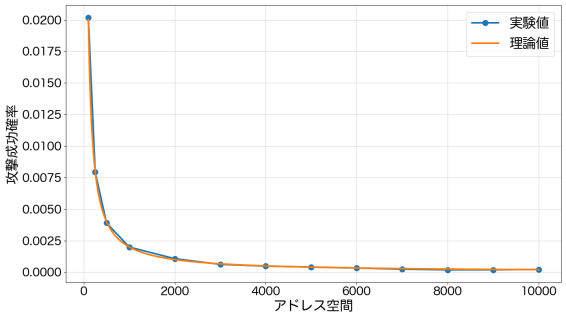

図 2　理論値と実験値の比較
Fig. 2　Comparison between theoretical and experimental values

して，定数 m，b を用いて次の式で表されると仮定した．

$$t = \mathrm{m}d + \mathrm{b} \tag{2}$$

(2) 式における m と b の値を比較することによって，応答速度の性能を比較することができる．転送サービスを介してメインプロセスにアクセスする場合と，直接メインプロセスにアクセスする場合において，それぞれ，POST メソッドにより，10MB のデータを送信した時の応答時間と，0MB のデータを送信した時の応答時間を取得した．それぞれ 100 回送信し，1 回の送信にかかった応答時間の平均を算出した．

直接アクセスする場合の m および b の値をそれぞれ $\mathrm{m_d}$，$\mathrm{b_d}$，転送サービスを介してメインプロセスにアクセスする場合の m および b の値をそれぞれ $\mathrm{m_p}$，$\mathrm{b_p}$ とする．このとき，直接アクセスする場合において，送信するデータが，10MB の時の応答時間を t_{dten}，0MB の時の応答時間を t_{dzero} とし，転送サービスを介してアクセスする場合において，送信するデータが，10MB の時の応答時間を t_{pten}，0MB の時の応答時間を t_{pzero} とすると，$\mathrm{m_d} : \mathrm{m_p}$ および $\mathrm{b_d} : \mathrm{b_p}$ は以下の式で表される．

$$\mathrm{m_d} : \mathrm{m_p} = (t_{dten} - t_{dzero}) : (t_{pten} - t_{pzero}) \tag{3}$$

$$\mathrm{b_d} : \mathrm{b_p} = t_{dzero} : t_{pzero} \tag{4}$$

5.　実験結果と提案手法の評価

本章では，第 4 章で概要を述べた実験の結果とそれに対する考察について述べる．

5.1　実験 1 の結果

実験の結果得られた攻撃成功確率のデータと，4.1.2 の計算結果をグラフで表したものとを比較すると，図 2 のようになった．横軸は利用可能なアドレスの数（{IP アドレスの範囲 }×{ ポート番号の範囲 }）であり，縦軸は攻撃成功確率である．図 2 から分かるように，実験で得たデータが理論値と非常に近い値となった．

5.2　実験 2 の結果

直接送信した場合，10MB のデータを 1 回送信するのにかかった時間の平均は，0.069639 秒であり，0MB のデータを 1 回送信するのにかかった時間の平均は，0.011915 秒であった．転送サービスを介した場合は，10MB のデータを 1 回送信するのにかかった時間の平均は，0.082821 秒であり，0MB のデータを 1 回送信するのにかかった時間の平均は，0.013545 秒であった．(3) 式および (4) 式より，$\mathrm{m_d} : \mathrm{m_p}$ および $\mathrm{b_d} : \mathrm{b_p}$ を求めると，以下のようになる．

$$\begin{aligned} \mathrm{m_d} : \mathrm{m_p} &= (0.069639 - 0.011915) : (0.082821 - 0.013545) \\ &= 0.057724 : 0.069276 \\ \mathrm{b_d} : \mathrm{b_p} &= 0.011915 : 0.013545 \end{aligned}$$

したがって，提案手法を用いた場合，用いなかった場合と比較して，(2) 式における m と b の値は，それぞれ 1.2001 倍，1.1368 倍になることが確かめられた．

5.3　提案手法の評価

実験結果から，Web アプリケーションに，第 3 章で提案した手法を適用することにより，許容可能なコストでセキュリティの性能を改善することが可能であることが確かめられた．また，本手法では，正規のクライアントは，自身のアカウントに割り当てられたアドレスからメインサーバにアクセスし，ログインすることで Web サービスを利用する．初めてメインサーバにアクセスする際や，自身に割り当てられたアドレスを忘失してしまった場合は，認証サーバにアクセスすることで自身のアカウントに割り当てるアドレスを指定することができ，以降は，ログイン後，ログアウトするまでの間に次に割り当てるアドレスを指定することで，継続してメインサーバにアクセスすることができる．これにより，Web アプリケーションにおける，正規のクライアントに対する可用性が保たれているため，本手法は有用であると言える．

6.　おわりに

6.1　本研究の主たる成果

本論文では，Web アプリケーションにネットワークレベルにおける MTD を適用する構成を 1 つ提案し，その性能を評価した．提案した手法は，Web アプリケーション内のアカウント情報と，MTD におけるアドレス変更の情報を結びつけるという新しい手法である．その性能を評価した結果，通信速度におけるパフォーマンスのわずかな低下はあるものの，Web アプリケーションのセキュリティを改善することができることを確認した．また，攻撃者から攻撃対象のサーバを隠すと同時に，正規のクライアントによるサーバへのアクセス経路を確保しなければならないという要件を満たしていることから，本手法は有用なものであると考えられる．

6.2 今後の課題

本論文では，ネットワークレベルにおける MTD により Web アプリケーションを保護する構成を提案したが，その構成は，認証サーバのセキュリティが担保されていることを前提としたものであった．今後は，よりセキュリティの高い手法を考案していくとともに，その実装をさらに工夫することで，いくつかの部分を簡単にカスタマイズ可能なものにしていく．最終的には，既存の Web アプリケーションフレームワークに組み込むことができる形にすることで，Web アプリケーションにおいて，誰もが簡単にネットワークレベルにおける MTD を利用することができるようにしたいと考えている．

謝辞 本研究を進めるにあたり，多大なるご指導並びにご教授を賜りました，小出洋教授に心から感謝致します．

本研究を進めるにあたり，ご意見並びにご助言を賜りました，九州大学サイバーセキュリティセンター学術研究員藤岡福資郎氏に心より感謝致します．

本研究を進めるにあたり，多くの討論の場において有益なご意見を賜りました，小出研究室の先輩方，そして同期の皆様に心より感謝致します．

本研究の一部は日立システムズの支援を得ています．

参考文献

[1] 独立行政法人情報処理推進機構 セキュリティセンター 安全なウェブサイトの作り方（改訂第 7 版）, 入手先 ⟨https://www.ipa.go.jp/files/000017316.pdf⟩（2020 年 2 月 4 日閲覧）, Mar 2015.

[2] 独立行政法人情報処理推進機構 セキュリティセンター: ソフトウェア等の脆弱性関連情報に関する届出状況 [2019 年第 3 四半期（7 月〜9 月）], 入手先 ⟨https://www.ipa.go.jp/security/vuln/report/vuln2019q3.html⟩（2020 年 2 月 4 日閲覧）, Oct 2019.

[3] P. Kampanakis and H. Perros and T. Beyene: *SDN-based solutions for Moving Target Defense network protection*, Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, pages 1-6, June 2014.

[4] Douglas C. MacFarland and Craig A. Shue. The sdn shuffle: *Creating a moving- target defense using host-based software-defined networking*, Proceedings of the Second ACM Workshop on Moving Target Defense, MTD ’ 15, pages 37-41, New York, NY, USA, 2015. Association for Computing Machinery.

[5] Jafar Haadi Jafarian and Ehab Al-Shaer and Qi Duan: *Openflow Random Host Mutation: Transparent Moving Target Defense Using Software Defined Networking*, Proceedings of the First Workshop on Hot Topics in Software Defined Networks, HotSDN ’12, pages 127-132, New York, NY, USA, 2012. Association for Computing Machinery.

[6] Rui Zhuang and S. Y. Zhang and Scott A. DeLoach and Xinming Ou and Anoop Singhal: *Simulation-based Approaches to Studying Effectiveness of Moving-Target Network Defense*, NIST, June 2012.

[7] D. P. Sharma and J. Cho and T. J. Moore and F. F. Nelson and H. Lim and D. S. Kim: *Random Host and Service Multiplexing for Moving Target Defense in Software-Defined Networks*, ICC 2019 - 2019 IEEE International Conference on Communications (ICC), pages 1-6, May 2019.

[8] Z. Jingzhe and F. Xuewei and W. Dongxia and M. Liang: *Web service applying moving target defense*, 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), pages 640-645, June 2018.

[9] Amirreza Niakanlahiji and Jafar Haadi Jafarian: *Webmtd: Defeating web code injection attacks using web element attribute mutation*, Proceedings of the 2017 Workshop on Moving Target Defense, MTD ’ 17, pages 17-26, New York, NY, USA, 2017. Association for Computing Machinery.

[10] M. Taguinod and A. Doupé and Z. Zhao and G. Ahn: *Toward a Moving Target Defense for Web Applications*, 2015 IEEE International Conference on Information Reuse and Integration, pages 510-517, Aug 2015.