Computer Security Symposium 2017
23 - 25 October 2017

攻撃状態遷移に基づくペネトレーションテストの自動化に関する考察

三浦　紘弥 †　　三村　守 †　　田中　秀磨 †

† 防衛大学校
239-8686 神奈川県横須賀市走水 1-10-20
em56030@nda.ac.jp

あらまし 本稿では，ペネトレーションテストの手順を状態遷移図にモデル化し，そのモデルを用いた安全性評価指標を提案する．この安全性評価指標により，システム管理者等に対し，サイバー攻撃の影響を容易に説明することが出来る．さらに，提案モデルの一部である公開サーバへの侵入を自動化するツールを実装した．このツールでは，攻撃対象のサーバの OS 及びアプリケーションに基づき，エクスプロイトを自動選択・実行する．検証実験の結果，実装したツールにより効率的にペネトレーションテストが可能であることを確認した．

Study on automating a penetration test based on an attack state transition

Hiroya Miura†　　MamoruMimura †　　Hidema Tanaka†

†National Deffence Academy.
1-10-20 Hashirimizu, Yokosuka city, Kanagawa prefecture 239-8686, JAPAN
em56030@nda.ac.jp

Abstract This paper models an attack state transition of penetration test procedure and proposes a security performance indicator using the attack state transition. The security performance indicator helps the systems manager to understand extent of the damage from cyber attacks. Moreover, we implement an automation tool that can infiltrate into a public server. This tool executes automatically exploits based on the OS and the software of the target public servers. The result of verification experiments shows that the tool can perform penetration tests efficiently.

1　はじめに

近年，情報通信技術は目まぐるしく発展が進み，情報セキュリティの重要性は日毎に増している．各種業務やインフラストラクチャの制御にコンピュータシステムが使用される機会が増え，これらの情報システムに対するマルウェアの脅威が顕著化してきている．情報システムの安全性を評価する手段の 1 つとして，ペネトレーションテストが挙げられる．ペネトレーションテストとは，評価対象へ実際に攻撃を実施し，脆弱性を洗い出す手法である．しかしながら，こうしたサイバー空間の防護に携わる人材が現在不足の傾向にあることが問題となっている．これらの背景から，ペネトレーションテストの効率性の向上や，省人化するために，ペネトレーションテストを自動化することは有効な手段の 1 つであると考える．

通常，ペネトレーションテスト実施後に情報

システムの脆弱性をシステム管理者へ報告する．しかしながら，情報システムに対してサイバー攻撃がどれほどの影響力を与えるかを理解させることは，システム管理者の識能によっては難しい場合がある．本稿では，ペネトレーションテストで実施する攻撃の状態遷移図をモデル化し，そのモデルに応じてネットワークの安全性を平易に評価できる安全性評価指標を提案する．また，提案するモデルのうち，公開サーバへの侵入を自動化するツールを実装する．このツールを実装することにより，効率的にペネトレーションテストが実行出来るとともに，少人数で多数の端末のペネトレーションテストが可能となり，省人化にも貢献出来ると考えられる．

本稿の構成について説明する．第２節は関連研究について説明し，関連研究と本研究との相違点を明らかにする．第３節は，状態遷移図を用いた情報システムの安全性評価の要領を説明する．また，公開サーバへの侵入を実行するための２つのアルゴリズムを説明する．第４節は，第３節で説明したアルゴリズムを用いて公開サーバへの侵入を自動化するツールを実装して，仮想ネットワークで効果を確認した．第５節は，実験の結果を考察し，研究の成果を明らかにする．第６節で本稿のまとめを記述する．

2　関連研究

本節では，ペネトレーションテストの自動化に関する関連研究等を示す．Naval Postgraduate School では，ペネトレーションテストを自動化する手法として，DIPR(Detect Identify Predict React) 自動化モデルを提案している [1]．この自動化モデルは，攻撃対象に対して探索プログラムを実行し，探索結果に基づき，攻撃可能な exploit を分析・実行させるものである．DIPR 自動化モデルでは，エクスプロイトの有効性，エクスプロイトの用途及び発見されにくさを考慮して優先順位を決定している．これに対し，本稿で実装したツールは，エクスプロイトのランクや，攻撃対象の OS やアプリケーションによってエクスプロイトに優先順位を付けて実行する．

他のペネトレーションテストの自動化ツールとしては APT2（An Automated Penetration Testing Toolkit）がある [2]．APT2 は複数のセキュリティスキャンツールからのスキャン結果の情報を取得し，その情報から使用可能であるエクスプロイトを実行する．APT2 はスキャナの実行結果から，攻撃可能と判断されるエクスプロイトを実行するが，優先順位を付けて攻撃を実行する機能が無い点が本稿のツールとの違いである．

もう１つの自動攻撃ツールとして，Sn1per が挙げられる [3]．Sn1per は，ターゲット端末の脆弱性をスキャン・列挙する自動脆弱性スキャナである．Sn1per にはエクスプロイトに優先順位をつけて実行する機能が無く，開放されたポートをスキャンし，各種プロトコルの脆弱性を列挙する機能がある．これに対し，本稿で実装したツールは，エクスプロイトに優先順位を決定するほか，攻撃対象の脆弱性は列挙せず，攻撃対象のバナー情報及び OS の種類を列挙する．この点が Sn1per と本稿で実装したツールの相違点である．

紹介したこれらの関連研究等はドメインコントローラサーバの管理者権限奪取までの攻撃を想定に入れていないが，本提案は，内部ネットワークのドメインコントローラサーバの管理権者限の奪取を最終目標とする．

3　提案手法

3.1　攻撃手法の区分

本節では，攻撃手法の区分について説明する．攻撃手法を「受動的攻撃」と「能動的攻撃」に区分する．

本稿では，能動的攻撃とは，ユーザが何らかの操作を行わなくとも，攻撃者の能動的に実施する操作によって実現される攻撃のことを言う．能動的攻撃の例を挙げれば，Conficker というマルウェアが挙げられる．Conficker は Windows のソフトウェアの欠陥を利用し，管理者のパスワードを辞書攻撃して，攻撃対象に不正アクセスするというものである．この攻撃手法は，攻撃対象の操作を必要とせず，攻撃者が一方的に

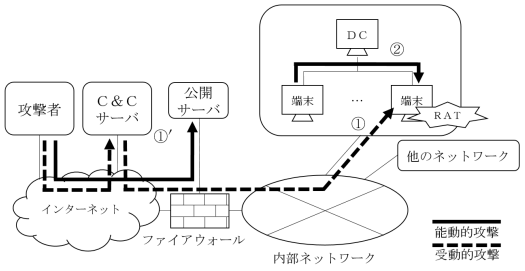

図 1: ネットワークモデル及び攻撃経路

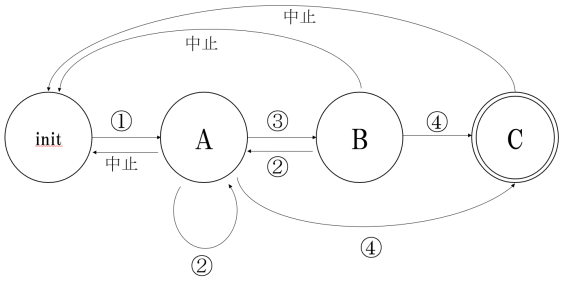

図 2: 内部ネットワークの攻撃時における状態遷移図

実行できるため，能動的攻撃であるといえる．

受動的攻撃とは，ユーザの何らかの操作を契機として行われる攻撃のことをいう．例を挙げるならば，ドライブバイダウンロード攻撃（DbD 攻撃）が挙げられる．DbD 攻撃は，攻撃者が悪意のある Web サーバを開設，または攻撃対象の Web サイトの改ざん等をして，脆弱性のあるアプリケーションやＯＳを使用するユーザが，その Web サーバにアクセスするようリンクを送付する等して誘導する．もし Web サーバに，脆弱性のある端末がアクセスしたならば，その端末にマルウェアを強制的にダウンロード及びインストールさせる．その端末にインストールされたマルウェアは，感染した端末と攻撃者の端末との間に不正な通信を強制的に確立させる．DbD 攻撃は，ユーザが攻撃者の悪意ある Web サーバにアクセスすることが攻撃成功のために必要な引き金となる動作となる．受動的攻撃と能動的攻撃の相違点は，攻撃成功の条件として攻撃対象の行う操作が必要である点である．

3.2 各攻撃段階における状態遷移

本節では，内部ネットワークのドメインコントローラサーバ（以下ＤＣと表記）の攻撃を目的とする状態遷移図と，公開サーバの管理者権限の奪取を目的とする攻撃の状態遷移図に区分して説明する．これらの状態遷移図は，ペネトレーションテスト終了後にシステム管理者に対するシステムの脆弱性の説明を容易にすることを目的としたものである．また，この状態遷移図は次節の安全評価指標の説明で使用される．

図 1 は本提案で想定するネットワークモデルである．攻撃者はインターネット上に存在し，攻撃者から見てファイアウォールの先に内部ネットワークがある．内部ネットワーク内には DC と複数のクライアントの端末がある．クライアントの端末のＩＰアドレスはＮＡＴの使用により外部から確認することはできない．また，図中の ①等の数字は，以下で説明する図 2 や図 3 の数字に対応している．また，図中の矢印は公開サーバや内部ネットワーク内のクライアント端末に対する攻撃経路を示している．

内部ネットワーク攻撃時における状態遷移図のノード

図 2 は，内部ネットワークへの侵入からＤＣの管理者権限の奪取に至るまでの攻撃を状態遷移図にして示したものである．図 2 の各ノードの説明をする．ノード init は内部ネットワークのいずれの端末にも初期潜入していない初期状態を表す．ノードＡは，内部ネットワークのいずれかのクライアント端末の初期潜入に成功した状態を表す．ノードＢは，侵入した端末のローカルの管理者権限を奪取した状態を表す．ノードＣは，DC の管理者権限を奪取した状態を表す．また，この状態遷移図は，ＤＣの管理者権限の奪取を目的としているため，ノードＣ遷移後，状態遷移は終了する．

内部ネットワーク攻撃時における状態遷移図のエッジ

図 2 を用いて，内部ネットワーク攻撃時における状態遷移図のエッジについて説明する．①は初期状態から，内部ネットワークのいずれかの端末へ侵入する動作を表す．攻撃者は標的型メール攻撃等の受動的攻撃を用いて，ＲＡＴ

(Remote Access Tool) と呼ばれるマルウェアをインストールさせる．ＲＡＴは事前に攻撃者が準備したＣ＆Ｃサーバに感染した端末をアクセスさせる．攻撃者は，このＣ＆Ｃサーバを通じて内部ネットワークの端末に侵入することができる．②はあるクライアントの端末から他のクライアントの端末への横の移動を示す．②は，Pass-The-Hash 等の手法によって実行される．③は，侵入した端末のローカルの管理者権限を奪取する動作を意味する．④は，DC の管理者の端末に侵入した後，DC 管理者権限を奪取する動作を意味する．③及び④はそれぞれ，MS14-058 や MS14-068 等の脆弱性を用いて行われるか，Mimikatz 等のツールを用いて能動的に攻撃される．MS14-058 とは，Windows のカーネルモードドライバの脆弱性を利用してローカル権限を昇格する手法である．MS14-068 とは，Windows の Kerberos 認証の脆弱性を利用して，リモートから任意のドメインアカウントへの権限昇格を行う手法である．内部ネットワーク攻撃時の状態遷移図における各エッジの主な手段を表 1 に示す．①及び②の攻撃経路を図 1 に示す．

公開サーバ攻撃時における状態遷移図のノード

図 3 は，公開サーバへの侵入から攻撃対象の管理者権限昇格に至るまでの攻撃を状態遷移図にして示したものである．図 3 の各ノードの説明をする．ノード init は，公開サーバに侵入していない初期状態を表す．ノードＡは，攻撃対象とする公開サーバへの侵入に成功した状態を表す．ノードＢは，公開サーバの管理者権限を奪取した状態を表す．本攻撃は，攻撃対象の管理者権限の昇格を目的としているため，ノードＢが受容状態となり，ノードＢに遷移後，状態遷移は終了する．

公開サーバ攻撃時における状態遷移図のエッジ

図 3 を用いて，公開サーバ攻撃時における状態遷移図の各エッジについて説明する．①’は，初期状態から公開サーバへ侵入する動作を表す．攻撃者は，攻撃対象の公開サーバに対して nmap 等のセキュリティスキャナを使用し，開放されたポートや，公開サーバの使用する OS 及びインストールされているアプリケーションの情報を収集する．収集された情報を基に，公開サーバの使用する OS やアプリケーションの脆弱性を利用し，バッファオーバーフロー等の手法を用いて攻撃者の任意のコードを実行させ，攻撃対象の公開サーバに侵入する．②’は公開サーバに侵入後，公開サーバの管理者権限を奪取する動作を表し，表 1 の③と同様の手段で実施される．公開サーバ攻撃時の状態遷移図における各エッジの主な手段を表 2 に示す．又，①’の攻撃経路を図 1 に示す．

表 1: 内部ネットワーク攻撃時における状態遷移図のエッジ

エッジ	手段の例
①	標的型メール攻撃，DbD 攻撃
②	Pass-The-Hash,Pass-The-Ticket, RDP
③	MS14-058Exploit,MS15-078Exploit,SDB UAC Bypass
④	MS14-068Exploit, Mimikatz(Golden Ticket,Silver Ticket),psecec

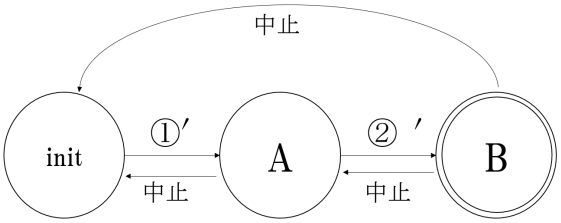

図 3: 公開サーバの攻撃時における状態遷移図

3.3 安全性評価要領

本節では，各状態遷移図を用いて安全性評価指標を説明する．安全性評価指標の目的は，情報システムに関する識能が乏しい管理者に対しても，サイバー攻撃が及ぼす影響を容易に説明できることを目的としている．安全性評価指標を前節の状態遷移図毎区分して説明する．

まず，内部ネットワーク内への攻撃に対する安全性評価指標について説明する．内部ネットワーク攻撃時における安全性評価指標は，レベ

表 2: 公開サーバ攻撃時における状態遷移図のエッジ

エッジ	手段の例
①’	MS06-068,CVE2017-7668Exploit
②’	MS14-058Exploit,MS15-078Exploit,SDB UAC Bypass

ル 0 からレベル 3 までの危険度に区分される．ペネトレーションテストの結果，侵入が不可である場合，レベル 0 とする．初期潜入のみ可能であった場合，レベル 1 とする．侵入後，ローカルの管理者権限の奪取が可能である場合，レベル 2 とし，DC の管理者権限の奪取が可能である場合をレベル 3 とする．

次に，公開サーバへの攻撃に対する安全性評価指標について説明する．内部ネットワークの攻撃時と同様に，公開サーバ攻撃時における安全性評価指標はレベル 0 からレベル 2 までの危険度に区分される．ペネトレーションテストの結果，侵入に失敗し，状態遷移がノード init にいるとき，レベル 0 とする．一般権限の不正アクセスが可能であった場合，レベル 1 とする．不正アクセス後，公開サーバの管理者権限を奪取できた状態をレベル 2 とする．

この状態遷移図と安全性評価指標を用いることによって，システムの管理者に対し，サイバー攻撃の及ぼす影響をレベルで平易に説明ができる．

3.4 公開サーバへの侵入テスト

状態遷移図のうち，公開サーバへの侵入を自動で実行するツールを実装する．本節では，**AlgorithmA** 及び **AlgorithmB** の 2 つのアルゴリズムを説明する．各アルゴリズムは，以下に示す要領で優先度点数をエクスプロイトに加点し，優先度点数の高い順にエクスプロイトを実行する．**AlgorithmA** は，攻撃対象の使用する OS 及びアプリケーションのエクスプロイトに優先度点数を加点する．次にエクスプロイトのランクに応じて優先度点数を加点し，優先度点数の高いエクスプロイトから順に実行するというものである．**AlgorithmB** は，アプリケーションの脆弱性より OS の脆弱性を優先してエクスプロイトを選択・実行する．攻撃対象が使用するアプリケーションに一致するエクスプロイトは優先度点数を減点し，ＯＳが一致するものは優先度点数を加点する．次にランクに応じて優先度点数を加点し，優先度点数の高いエクスプロイトから順に実行するというものである．

この 2 つのアルゴリズムは，metasploit framework 及び nmap を用いて実装した．metasploit framework とはエクスプロイトコードの作成・実行を行うペネトレーションテスト用オープンソースプロジェクトである．metasploit framework には約１５００種類ものモジュールが格納されている．モジュールには，それぞれ攻撃対象とするアプリケーションや OS がある．また，metasploit framework には resource コマンドと呼ばれるコマンドがあり，これを実行することによって，resource script に記述された任意のコマンドを自動で実行することができる．

nmap とは，他のコンピュータに対して OS や，インストールされたアプリケーションの検出，ポートスキャン等を行うセキュリティスキャナである．

AlgorithmA

AlgorithmA は当初，nmap の実行によって検出された OS によって優先して実行するエクスプロイトを選り分ける．次に nmap を用いてバナー情報を収集し，そのバナー情報から，ターゲットに対して適応可能であると見積もられるエクスプロイトを優先して実行する．細部は次の 5 つのステップで実行する．

Step1　metasploit framework 内のエクスプロイトコードのパス及びランクの記載されたリストを取得し，配列に代入する．

ランクとは，metasploit framework の運営会社である，Rapd7[5] が設定した，各エクスプロイトコードの評価である．ランクは Excellent，Great，Good，Normal，Average，Low の 6 種類がある．ランクは，Excellent に近ければ，攻撃の成功率は高まり，Low に近いほど攻撃の成功率は下がる．エクスプロイトコードのパスの構造について

例を用いて捕捉説明する．

e.g) unix／ftp／vsftpd__234__backdoor

エクスプロイトコードのパスは，/で三つに区切られた構造をしている．エクスプロイトコードのパスの左端の/で区切られている部分は，攻撃対象のＯＳを表す．中央の/で区切られている部分は脆弱性を持つアプリケーション名が記載されている．右端の/で区切られる部分は攻撃の要領や攻撃の効果を簡潔な名称で表現したもの，もしくは microsoft の脆弱性番号等が記載される．

Step2　nmap のコマンド”nmap -sV –script =banner ＜ターゲットのＩＰアドレス＞ -oG ＜ログファイルの保存先のパス/ログファイル名＞”を実行し，攻撃対象の端末のバナー情報及び攻撃対象が使用する OS の情報を取得し，ログファイルとして保存する．

Step3　各エクスプロイトコードにランクに応じて，優先度点数を加点する．優先度点数は，エクスプロイトコードの実行順序を決めるための点数である．優先度点数の高いエクスプロイトコードを先に実行する．

ランクに応じた優先度点数の加点要領は，Excellent ならば 0.6 点，Great ならば 0.5 点，Good ならば 0.4 点，Normal ならば 0.3 点，Average ならば 0.2 点，Low ならば 0.1 点を加点する．

Step4　**Step2** で検出されたＯＳの検索を行い，一致するものに優先度点数２点を加点する．次に，**Step2** で検出したログファイルのバナー情報から，攻撃対象がインストールしているアプリケーションを検出する．検出されたアプリケーション名を，**Step1** で取得したエクスプロイトコードのパス名のリストから検索する．検出されたアプリケーション名に一致するエクスプロイトコードに 1 点の優先度点数を加点する．**Step3**～**Step4** で，各エクスプロイトの優先度点数は，OS のみ一致していた場合は，２点，アプリケーションと OS が一致していた場合は 3 点が配点される．優先度点数の整数部が同点であるエクスプロイトは，ランクによって小数点第一位に優先度点数を加点し，優先順位付けする．こうして配点された優先度点数に基づき，**Step1** で取得したリストを降順にソートし，配列に代入する．

Step5　優先度点数の高い順にエクスプロイトコードを実行する．攻撃が成功するか，またはエクスプロイトコードを全て実行した時点で攻撃を終了する．攻撃成功とは，こちらの任意のコマンドを攻撃対象に対して実行させることが可能となった状態を意味する．

次に，nmap によって検出された OS のエクスプロイトを優先的に選択・実行する **AlgorithmB** について説明する．

AlgorithmB

AlgorithmB と **AlgorithmA** の相違点は，アプリケーションの脆弱性を利用するエクスプロイトの優先順位を下げ，OS の脆弱性を利用するエクスプロイトの優先順位を上げるように設計した点である．**AlgorithmB** の細部は次の５つのステップで説明する．

Step1～**Step3** は **AlgorithmA** に同じ．

Step4　**Step2** で検出されたＯＳの検索を行い，一致するものに優先度点数２点を加点する．次に，**AlgorithmB** と同様の要領で，バナー情報から，攻撃対象がインストールしているアプリケーションを検出する．アプリケーションの脆弱性を利用するエクスプロイトの優先順位を下げ，OS の脆弱性を利用するエクスプロイトを優先して実行させるため，検出されたアプリケーション名に一致するエクスプロイトコードに優先度点数 1 点を減点する．

Step5　**AlgorithmA** の **Step5** と同様の要領で実施する．

AlgorithmA 及び **AlgorithmB** での優先度点数の加点要領を表 3 に示す．

表 3: 優先度点数加点要領

条件	AlgA	AlgB
ＯＳが一致	2.0 点	2.0 点
アプリケーションが一致	1.0 点	-1.0 点
ランクが Excellent である	0.6 点	0.6 点
ランクが Great である	0.5 点	0.5 点
ランクが Good である	0.4 点	0.4 点
ランクが Normal である	0.3 点	0.3 点
ランクが Average である	0.2 点	0.2 点
ランクが Low である	0.1 点	0.1 点

4 実験

4.1 実験環境

本研究では，攻撃用端末として Kali Linux を，攻撃対象として，metasploitable 2 [4] を使用した．また，攻撃用端末及び攻撃対象の端末はそれぞれ仮想環境（ＶＭＷａｒｅ　ver 10.0）において構築し，各端末は 1 対 1 のホストオンリー接続で接続した．また，ホストの端末の環境は，Windows10Home，実装メモリ（RAM）8GB，CPU は IntelCorei7(3.40GHz) である．

4.2 実験手法

本研究の目的は，公開サーバへの侵入を効率的に自動化し，ペネトレーションテストの効率化及び省人化に資することである．実験の目的は，**AlgorithmA** 及び **AlgorithmB** を実装し，最適な手順を検討することである．

まず各アルゴリズムを実装し，それぞれ 30 回実行する．この時，実行時間やエクスプロイトの試行回数等のデータを記録する．終了条件は，攻撃が成功するか，もしくは実行可能な全てのエクスプロイトが失敗したときである．

4.3 実験結果

AlgorithmA 及び **AlgorithmB** をそれぞれ 30 回実行した結果，表 4 及び表 5 の結果を得ることが出来た．表 4 は実験結果を示したものである．**AlgorithmA** の試行回数 1 回であり，**AlgorithmB** の試行回数 120 回で攻撃に成功することが出来た．また，検出されたアプリケーションは表 4 に示す 4 つであり，検出された OS は Linux だった．攻撃対象のアプリケーション及び OS に一致するエクスプロイトは 7 個であり，OS のみ一致するエクスプロイトは 567 個だった．この OS に一致するエクスプロイトとは，Linux を攻撃対象とするエクスプロイトのことである．表 5 は各アルゴリズムを 30 回実行したときの実行時間を示したものである．**AlgorithmA** は **AlgorithmB** よりも早く攻撃に成功することが出来た．

表 4: 実験結果

項目	結果
試行回数 (攻撃の成否)	AlgoA:1 回 (○) AlgoB:120 回 (○)
検出された app 及び OS	apache，mysql，postgresql，vsftpd，Linux
app･OS が一致する exploit 数	7 個／ 1560 個
OS のみ一致する exploit 数	567 個／ 1560 個

表 5: 実行時間

項目	AlgA	AlgB
平均値	2 分 6 秒	11 分 1 秒
最大値	2 分 7 秒	12 分 41 秒
最小値	2 分 5 秒	9 分 28 秒
中央値	2 分 6 秒	11 分 32 秒

5 考察

本節では，**AlgorithmA** と **AlgorithmB** のどちらが効率的にペネトレーションテストを出来るかを考察する．本実験では，攻撃対象からバナー情報が取得できる環境で実験を実施した．この実験環境下においては **AlgorithmA** は **AlgorithmB** に比べて試行回数が少ないため，**AlgorithmA** の方が効率的な攻撃を実行出来ると考えられる．次に攻撃対象からバナー

情報を入手出来ない場合を考察する．この場合，**AlgorithmA** と **AlgorithmB** の実行するエクスプロイトの順番は同じになるので，どちらのアルゴリズムを使用したとしても結果はほぼ同じになると考えられる．しかしながら，攻撃対象にバナー情報が検出されない設定を施した場合，アプリケーションの脆弱性をついた効果的なエクスプロイトを優先的に実行されない．そのため，バナー情報が検出されなかった場合，公開サーバが使用することが予想されるアプリケーションを考慮したエクスプロイトを予めリストの上位に設定することで，攻撃を早期に成功させることが出来ると考えられる．次に，検出された OS が windows だった場合を考察する．windows を攻撃対象とするエクスプロイトは 1000 個程度であり，Linux を攻撃対象とする時より 2 倍近く多い．その為，優先的に実行するエクスプロイトを絞る必要があるため，**AlgorithmA** が **AlgorithmB** より効率的であると考えられる．

本ツールの実用性について考察する．攻撃に成功するエクスプロイトがなかった場合，1500 ものエクスプロイトを全て試行するため，多くの時間が必要となる．今回の実験では DbD 攻撃等，公開サーバへの攻撃に直接関係のない攻撃をも実行しているので，攻撃に失敗する場合，本ツールの実用性は欠ける．しかしながら，攻撃の目的にそぐわないエクスプロイトの実行を省略する処理を追加することにより，実用性を持たせることが可能になると考えられる．

6　まとめ

本稿では，ペネトレーションテストの手順を状態遷移図にモデル化し，そのモデルを用いた安全性評価指標を提案した．この安全性評価指標を用いることによって，システム管理者対してサイバー攻撃が社内等ネットワークに及ぼす影響を平易に説明することが可能となった．また，本稿で実装したツールを使うことによって，ペネトレーションテストをより効率化・省人化することが出来る．今後の課題は，本ツールの攻撃の終了条件の改良を行い，本ツールの実用性を向上させるとともに，内部ネットワークへの初期潜入及び，DC の管理者権限奪取までの一連のペネトレーションテストを自動化することである．

参考文献

[1] Deborah E. Goshorn：Automation of cyber penetration testing using the detect,identify, predict, react intelligence autmation model（Online）,avairable from (http://www.dtic.mil/dtic/tr/fulltext/u2/a590032.pdf)

[2] ”automated penetration toolkit” (https://github.com/MooseDojo/apt2)

[3] ”Automated Pentest Recon Scanner” (https://www.darknet.org.uk/2017/05/sn1per-penetration-testing-automation-scanner/)

[4] ”Metasploitable is an intentionally vulnerable Linux virtual machine” (https://sourceforge.net/projects/metasploitable/files/)

[5] ”RAPID7” (https://www.rapid7.com/)